# FAULT ANALYSIS AND RELIABILITY MONITORING SYSTEM (FARMS)

Nishamali Chandrika Madupani Allis Attanayake
149204M

Faculty of Information Technology
University of Moratuwa
June 2017

# FAULT ANALYSIS AND RELIABILITY MONITORING SYSTEM (FARMS)

Nishamali Chandrika Madupani Allis Attanayake
149204M

Dissertation submitted to the Faculty of Information Technology, University of Moratuwa, Sri Lanka for the partial fulfilment of the requirements of the Degree of Master of Science in Information Technology

June 2017

# Declaration

We declare that this thesis is our own work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

N. C. M. A. Attanayake

Date: 16th June 2017

Supervised by

Saminda Premarathne

Date: 16th June 2017

*To my parents, Husband Wijekoon and two daughters Viyathma and Sesathma*

*for their encouragement and love*

# Acknowledgment

I would like to express my profound gratitude and sincere appreciation to my supervisor, Mr. Saminda Premarathne for his constant guidance, constructive suggestions, continuous encouragement and moral support during all phases of the research program. Your impressive knowledge, technical skills and human qualities have been a source of inspiration and a model for me to follow.

I am grateful to Land Settlement Department for providing me funding and necessary study leave for my Master of Science degree at University of Moratuwa.

I am also thankful to Mr. K. A. K. Ranjith Dharmapala, Commissioner General, Land Title Settlement Department and Mrs. P. M. H. Priyadarshani, Commissioner, Land Title Settlement Department for their help given to me in many ways to make this course a success.

I sincerely thank and highly value the support given by the laboratory staff of Computer Lab in Information Technology Faculty, University of Moratuwa.

I would like to thank my colleagues in the MSc course in 2014/2015 (batch 8). I will never forget and I really appreciate the friendly environment you all created during my MSc degree course at University of Moratuwa.

Last but not least, I would like to express my deepest gratitude and love to my parents, my husband Wijekoon, two daughters Viyathma and Sesathma. Your love, invaluable support, patience, motivation and encouragement helped me to reach this target under tiring circumstances.

**Abstract**

*Ceylon Electricity Board (CEB), established by an act of Parliament in 1969, is the statuary body responsible for most of electricity power generation, Transmission and major part of distribution of electricity in Sri Lanka. At present 98.5% of the population has access to electricity. Once the electrification is completed, attention would be mostly paid to power supply reliability.*

*Power supply reliability could be defined as the availability of power supply at a given time duration with adequate quantity and good quality. In order to improve power supply reliability and maintain the reliability level at specified level, it needs comprehensive monitoring and analysis of power supply failures in the electricity network including individual consumer premise.*

*At present CEB has adopted a hard copy based power supply failure analysis system which has many drawback such as lack of information about the past and current power supply failures, process is handlled by ad-hoc method, information flow is difficult to identify, longer time for power failure record searching and processing, report preperation is difficult and time consuming etc.*

*In order to rectify the drawbacks in the present system, Failure Analysis and Reliability Monitoring System (FARMS) has been developed in this project. The proposed system was a client server model based system for handling power supply failures in the consumer service center of Ceylon Electricity Board. The main database of the FARMS is created by using the Ms SQL Server 2008, because it supports client/server approach and provide multi user access for the system. This provides performance wise much better with the Microsoft based systems and compatible with other third party controls.*

*The proposed system has many features which has eliminated almost all the drwabacks in the prsesent system. The System has been evaluated developing a software and validated for actual collected data.*

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Prolegomena

The power sector in Sri Lanka is organized under the Ministry of Power and Renewable Energy. Ceylon Electricity Board (CEB), established by an act of Parliament in 1969, is the statuary body responsible for most of electricity power generation, Transmission and major part of distribution of electricity in Sri Lanka. At present 98.5% of the population has access to electricity from national electricity grid and there are about 5.65 million consumers provided electricity [1].

The CEB is mainly divided into three divisions, Generation, Transmission and Distribution. The Distribution division is operated under four Distribution Divisions based on the geographical demarcations and consumer mix. According to [1], in the year 2015, the peak power supply demand was about 2,283 MW and the energy demand was 13,090 Gigawatt -hours (GWh). The demand grows at a rate of 6% per year. The total revenue from the electricity sales in the year 2015 was Rs. 188 billions.

The main objective of CEB is to provide adequate power supply with better reliability. In this regard, the CEB needs to maintain its electricity network to provide specified reliability level while upgrading the same for catering the growing demand. Thus the electricity failures should be minimized to maintain supply at higher reliability.

## 1.2 Power Supply failures and Reliability

Electricity is one of the essential needs of people at present. Most of the day to day activities of the people are linked with electricity. Power supply failure to very short duration is not expected and even an electronic weighing scale in a small boutique affects the power supply failure leading to completely stop of activities in the boutique. In order to maintain power supply reliability [2], supply interruption frequency and the duration should be minimized, ideally to 0 and practically to very low value.

In order to improve power supply reliability and maintain the reliability level at specified level, it needs comprehensive monitoring and analysis of power supply failures in the electricity network including individual consumer premise. This could be achieved through power failure analysis and monitoring the power supply reliability.

### 1.3 Motivation

The power supply to the consumers has to be maintained at very high level so that day to day activities of the consumers are not disturbed due to power supply failures. Ceylon Electricity Board (CEB) and Lanka Electricity Company (LECO) (Pvt) Ltd are the electricity distribution utilities in Sri Lanka. 90% of the consumers, amount to 5.65 millions are served by CEB while the balance 10% is served by LECO.

Electricity consumers expect better quality and more reliable power supply to fulfil their basic needs. Hence, supply failures are not expected and it has to be minimized. However, in practice power supply interruption can not be brought to zero but it could be maintained at a very low value. Quality power supply with higher reliability involves lots of investments in the expansion and reinforcement of electricity network. In order to maintain power supply reliability at specified level, supply failures have to be monitored and analyzed. This needs development of a database management system. At present this subject is not addressed properly and if such a system could be developed, it would be a great contribution. This factor drives me to select this topic for my research in the Master of Science in IT.

### 1.4 Problem Statement

The Distribution division of CEB is divided into four Regions. The four regions consist of 12 administrative provinces and each province comprises of three or more areas depending on the consumer density in the province. Each area consists of two or more Consumer Service Centers (CSC). Consumer services like providing new service connection, electricity line maintenance, power supply failure restoration etc. are done at the consumer service center. Each CSC is headed by a technical officer called Electrical Superintendent.

IT facilities have been introduced to CSCs with the development of Service Connection Data Management system which handles the service connection application processing and consumer account generation. Power supply interruption reporting mechanism has been introduced a few years ago. However, there is no proper way to analyze each power supply failure, which gives the basis for power supply reliability improvement. If such system could be developed it gives lot of information such as most frequent cause of failures, the weakest area of the network, the average time for failure restoration, average number of failures per consumers etc.

### 1.5 Hypothesis

The hypothesis of this research is that the development of power supply Failure Analysis and Reliability Monitoring System (FARMS) could improve the power supply reliability to the consumers.

### 1.6 Objectives

The main objective of the research is

- to design data management system for power supply failure analysis and reliability monitoring
- to develop computer software for power supply failure analysis and reliability monitoring for the use of Consumer Service Centers in Ceylon Electricity Board

### 1.7 The approach for Database System Design and Development

Dataflow diagrams depicted in Figure 1.1& Figure 1.2 describe the approach for system design and database development for power supply failure analysis and reliability improvement system.

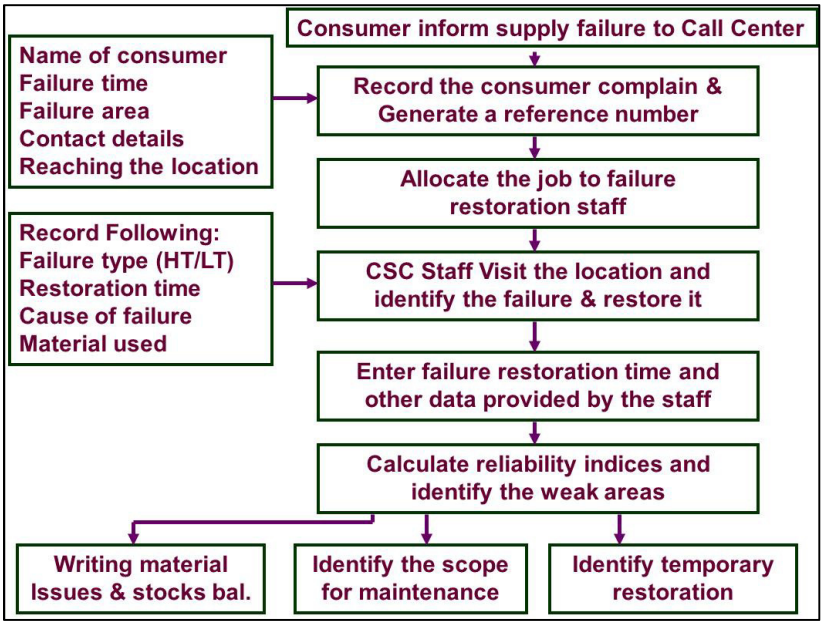



Figure 1.1: Dataflow diagram for the development of computer software for failure analysis

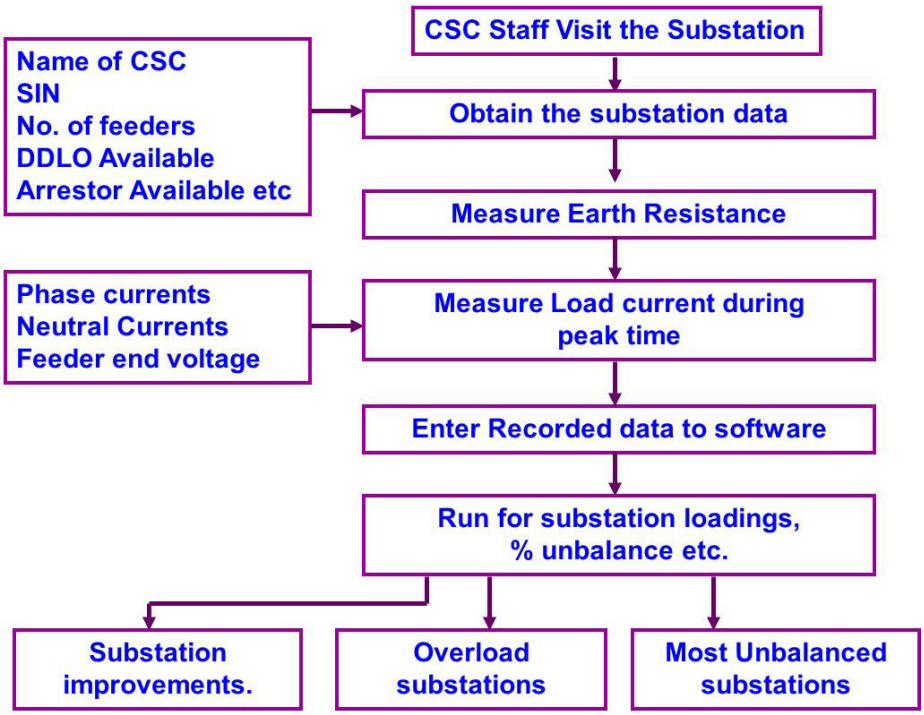



Figure 1.2: Dataflow diagram for the substation load data processing

### 1.8 The Structure of the Thesis

Chapter 1 (this chapter) of the thesis gives a comprehensive description of the overall project described in this thesis. It gives brief introduction to the project, motivation, problem statement, hypothesis, objectives of the project and the approach of the project. Chapter 2 critically reviews literature on power supply reliability and factors affecting power supply reliability. Further, the early developments, modern trends and future challenges of the improvement of power supply reliability have been described in the chapter 2. Chapter 3 is about the existing manual method of power supply failure analysis, data collection methodologies, format used for data collection, classification of power supply failures and determination of reliability indices. Chapter 4 has been dedicated to comprehensive approach to database design and development of computer software for the analysis of power supply failures for reliability improvement. Chapter 5 describes the database design for Failure Analysis and Reliability Monitoring System (FARMS). Chapter 6 is dedicated to implementation of software, selection of technology used. Chapter 7 is about the evaluation of the designed database software using collected actual data obtained from recorded hardcopy base registers. Chapter 8 is dedicated to discuss results obtained from the developed software. Chapter 9 concludes the research with note to future work.

### 1.9 Summary

This chapter gave an overall picture of the entire project presented in this thesis. As such the background, motivation, problem definition, hypothesis, objectives of the project and the approach of the project have been described. Next chapter presents a critical review of the literature on power supply reliability and factors affecting power supply reliability.

--

# Chapter 2

# Literature Review

## 2.1 Introduction

Chapter 01 gave a comprehensive description of the overall project described in this thesis. This chapter provides a critical review of the literature in relation to developments and challenges in Fault Analysis and Reliability Monitoring System (FARMS). For this purpose, the review of the past researches have been presented under three major sections, namely, early developments, modern trends and future challenges. At the end, this chapter defines the research problem as the analysis of power failure to evaluate power supply reliability to the customers attached to Ceylon Electricity Board by way of developing supply failure database.

## 2.2 Early Developments

Power supply reliability in the early stage of power system was not big concern since only a very few house hold (less than 5%) had access to electricity. Further dependability of equipment on electricity was at minimum level and most of the appliances were mechanical driven or the manual operation where human intervention was required to operate the equipment. The most focus was to supply electricity to household. Therefore, individual customer especially on heavy electricity consumers was mostly focused to address the power supply reliability.

Probability of failure and statistical analysis were used in making the quantitative calculations of the reliability and availability of the three power distribution systems [3]. In this regard, a simple radial system had been analyzed and it was found that an average forced hours downtime per year was between 5.9 to 9.0 times larger than a secondary-selective system. Three radial distribution systems analyzed were simple radial system, primary-selective system, and secondary-selective system and this study only considered forced outages of the electrical equipment and did not quantify no of consumers affected and expected energy not served due to power supply failure. Equipment outage rate and repair time relationship had been defined as follows.

$$Availability = \frac{Hours\ available}{Total\ hours} \quad (2.1)$$

$$Unavailability = 1 - Availability \quad (2.2)$$

$$Forced\ unavailaility = \frac{Forced\ hours\ down\ time}{Total\ hours} \quad (2.3)$$

$$Forced\ unavailability = Failure\ rate\ (\lambda) * Average\ down\ time\ per\ failure(r) \quad (2.4)$$

Where λ =No. of failures per year or hour, r = average time to repair

The paper [3] has given values for λ and r for different components and provided a method for calculating same parameters when components are in series and parallel. Further, it was concluded that 13.8 kV power source was the most contributor to forced outage downtime per year with 0.963 failures per year. A significant improvement can be made in both the frequency of outage and the forced hours downtime per year by having two independent sources of electric utility power at 13.8 kV. However, getting a value for λ and r required large no. of past data.

In [4], reliability consideration on cement plant power distribution had been addressed. Even the definition of the power supply reliability was different. Here [4] the power supply reliability has been defined as "the probability that a component or system will operate satisfactorily in a given environment for a stated period of time". Thus it is clear that according to this definition, the reliability was based on the operation of the equipment and did not tell about number of consumers affected and the amount of energy not served. Further, 12 recommendations have been given to follow in designing, operation and maintenance of the power supply. These recommendations includes equipment selection, operation environment, supply voltage drop, protection system etc. It was highlighted in this paper that maintenance staff has mistaken attitude that electrical apparatus was different from production machineries and would work under any condition. However, it was noted that many electrical apparatus were failed due to lack of

maintenance. Since this paper was focused on individual customer based power supply reliability and it does not address the present day power supply reliability.

Local Generating Facilities in the Reliability Evaluation of Power Distribution Systems were discussed in [5]. The paper describes models and computational techniques that have been developed to translate the conceptual ideas of local generation into practical analysis which evaluates the five reliability indices for each load-point of interest. The five reliability indices are Expected Failure Rate X, Average Outage Time r, Expected Annual Outage Time U, Average Load Disconnected L, Average Energy not Supplied E. A computational algorithm which evaluates the load and energy supplied to the load-point of interest after a failure event caused by either total or partial loss of continuity has occurred were described in the same paper. The algorithm considers the system topology, the loading profile of the busbars, the component reliability data, and the station data (staffing arrangements, operating times, priority order, switching time, unit capacity). The algorithm included different ways of generating unit failure, a generating unit may fail to operate when required because of starting or running failures, a unit may fail to start because it is out on maintenance or the appropriate breaker fails to close or the unit fails to start. This paper too was focused on the individual equipment failure in the power supply system and did not address the system as whole. However, majority of early development was on this area and later it was developed on the affected consumer basis.

The study carried out in [3] has been extended to seven distribution network in [6]. Field experience was matched with theoretical calculations for frequency and average duration of outages in order to predict levels that can be expected from each type of system. The systems are then ranked by reliability and their ability to supply what has come to be known as “clean” power. The result in [6] provides a basis of comparison for the distribution engineer when choosing the system design suitable for the particular type of load to be served. In this paper it was concluded that before selecting a distribution system, it should be carefully evaluated to determine whether it is capable of supplying the desired degrees of reliability and power quality. The reliability of a well planned and well designed grid network distribution system is several hundred times better than the

primary or secondary selective systems, distributed grid network systems with multiple stepdown transformers and multiple secondary cables per phase provide the highest degree of voltage regulation. Since this paper too based on the equipment failure rate, customer impact could not be determined using the approach given in this paper.

Reliability and the factors affecting it, to unveil any mysteries that still prevail with regard to reliability have been described in [7]. The paper has incorporated customer related reliability indices which formed the basis for evaluating reliability in the modern electrical power network. The reliability indices defined are given below.

SAIFI: the system average number of interruptions per customer per year
CAIFI: the customer average number of interruptions per customer affected per year
SAIDI: the system average interruption duration per customer served per year
CAIDI: the customer average interruption duration per customer interruption, the ratio of the total number of customer hours that service was available during a year to the total customer hours demanded
ASUI: the ratio of the total number of customer hours that service was unavailable during a year to the total customer hours demanded.
AENS: the average energy not supplied per customer served per year

In the same paper [7], as briefly discussed, reliability levels are interdependent with economics since increased investment is necessary to achieve increased reliability or even to maintain it at current and acceptable levels. This concept creates the incremental cost of reliability characteristic shown in Figure 2.1.

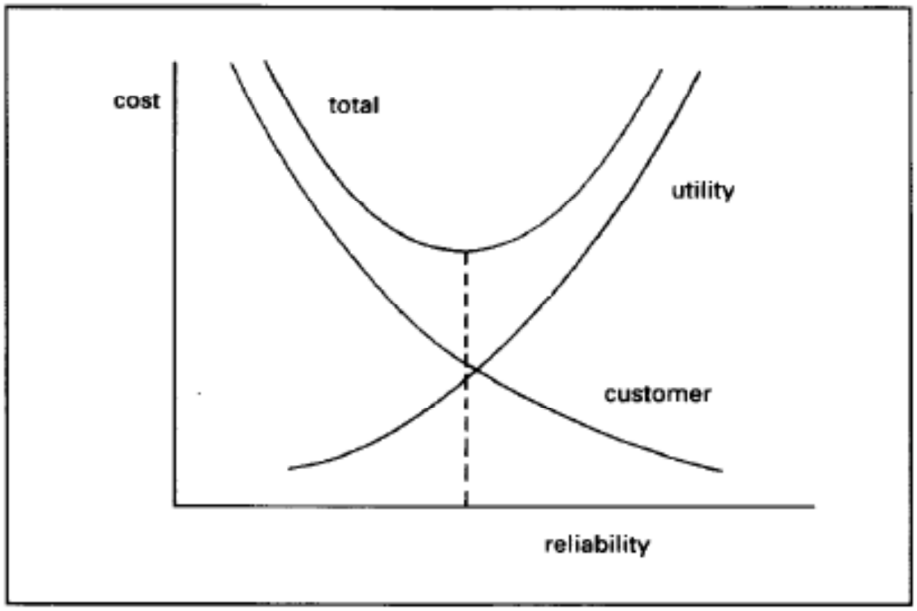



Figure 2.1: Reliability and the Cost Comparison

Although the concept of optimum level of reliability may be an ultimate and worthy goal, it can only be achieved best in slow steps. Many distribution systems are still designed according to deterministic standards. These views are changing quite significantly and there is now a positive awareness of the need to assess system design alternatives in a probabilistic sense. There is also a rapidly growing appreciation, inside and outside the industry, of the need to account for customers' expectations and their assessment of the worth of supply. Since the latter cannot be objectively assessed without adequate and objective reliability measures, we expect the two aspects, reliability and worth of supply, to become of significant importance in the very near future i.e. at present.

The costs and reliability of power distribution systems are beginning to receive as much attention as those of power generation and transmission systems. Especially in developing countries where the growing power demand provides needs for substantial expansion of power distribution systems, the long term benefit brought by using computerized optimization tools are tremendous. In [8], a new approach for the system optimization of power distribution systems is presented. Here the, distribution system reliability is modelled in the optimization objective function via outage costs and costs of

switching devices, along with the nonlinear costs of investment, maintenance and energy losses of both the substations and the feeders. The optimization model established is multi-stage, mixed-integer and nonlinear, which is solved by a network-flow programming algorithm. A multi-stage interlacing strategy and a nonlinearity iteration method are also designed. Supported by an extensive database, the planning software tool has been applied to optimize the power distribution system of a developing city. The constraints for the objective function established in this paper include Kirchhoff's current law, power capacity limits, voltage drop limits, radial configuration constraint and reliability indices. For the approach presented in this paper, the availability and accuracy of the necessary data were very important in supporting the models. With more and more attention in utilities attracting to the distribution reliability modeling and computerized planning, a more complete and accurate database can be constructed in the near future, through progresses in distribution load forecasting, outage data collecting, operational cost accounting and many other aspects. The drawback in this method is reliability is a function of the cost of electricity and it does not provide failure affected consumer based indices.

### 2.3 Modern Trends

Most of the power supply reliability evaluations discussed in Section 2.2 were based on equipment failure rates, supply restoration time, outage costs etc. However, the modern trend is to express the power supply reliability in terms of customer based indices.

Power supply reliability [2] evaluation techniques which are applied in electricity distribution system planning studies and operation have been described in [9]. Three main topics which have been discussed in the literature were reviewed for accommodating them into modern planning process. This method is good for initial stage of reliability improvement because it is simple to adopt and based on what was practiced in the past. However, this method has many draw backs since plans cannot be implemented as per the schedule. Thus in the operation stage, it does not give the reliability level used for planning process.

The worked by Bhargava, Murty and Sreevani [10] has incorporated weather effect into evaluation of power supply reliability evaluation. The weather conditions play a significant role on the reliability of a given power system leading to frequent incidence of failures to overhead system and their effectiveness. More than 50% of the power supply interruptions are due to bad weather and inclusion of weather effect into evaluation model makes it more robust. Two states weather model has been used and series of case studies were performed. The numerical results have shown that the load point failure rates are immune to variations in the percentages of failures occurring in bad weather and load point unavailability and average outage durations are directly influenced. The limitation of this method is how accurate one can predict the weather condition and the accuracy of weather model has substantial influenced on the final results.

Determination of root causes of power system faults are very important since once identified remedial actions could be implemented such that no faults would be originate again due to that cause. Kehindde and his colleges have presented a case study on the evaluation of faults in 11 kV distribution system [11]. In this study, it was highlighted that main cause of failure was due to falling of trees to the electricity lines and the authors has recommended adopting a clear right of way clearing policy. The approach in this study was based on the recorded data at the time of failure. Thus a proper and accurate failure record database should be maintained to get desired results.

In order to relate investment costs to the resulting levels of supply reliability, supply reliability costing technique have been used by Prasad and Ram Das [12]. Customer interruption costs (CIC) are used as substitute in the assessment of reliability worth in electric power systems. Customer interruption damage function has been developed for all customer categories in order to determine key indices. The results presented in this paper are useful for the electric power utilities, designers and planners in the decision-making stage. However, customer interruption cost depends not only on the energy not served but also several other factors like consumer category, purpose of use of electricity etc. Thus estimating such costs involve lots of assumptions.

The power supply failure depends on the power distribution network configuration. If the network has flexibility to reconfigure in an event of a failure in a part of the network, it could minimize the number of affected consumers due to the failure. In [13], a method for reliability improvement of power distribution system via feeder reconfiguration has been described. The work presented here is developed based on a linearized network model in the form of decoupled load power flow and linear programming model in which current carrying capacities of distribution feeders and real power constraints have been considered. The optimal open/close status of the sectionalizing and tie-switches are identified using an intelligent binary particle swarm optimization based search method. The probabilistic reliability assessment is conducted using a method based on higher probability order approximation. Several case studies are carried out on a small 33-bus radial distribution system, which is extensively used as an example in solving the distribution system reconfiguration problem. Further, the effect of embedded generation has been considered in one case scenario. Even though the method described in this paper was suited for modern power distribution systems, the increase of reconfiguration flexibility needs to install many switching devices into system and system reinforcements. Thus an economic evaluation has to be carried out to justify the investment against the achieved reliability.

### 2.4 Future Challenges

Power supply reliability is one of the most important contributor to economic development of a country. Modern shophisticated equiment may not tollerate power supply voltage deviation from specified value even to very small duration in the scale of milli seconds. Therefore, power supply reliability has always been of utmost importance during design, operation, and planning of electric power systems. In the recent decades, reliability of power systems has become more critical as the daily lives of people highly depend on electricity. According to the U.S. Energy Information Administration (EIA), over the past three decades, the consumption of electricity by appliances and electronic device of U.S. residential customers has almost doubled [14].

The Information Technology at present has devloped to very high level. With this development, many equipment in the power supply network could be remotely operated additing comunication facilities to the equipment by means of modern electrnoic devices. Thus, the isolation of faulty devices could be done within very short period of time compared to time taken for manual operation in the early days by visiting to the site. Adding communicatiuon to equiment is adding some smartness to the equimnet for flexible operation. For an example, utilities at present installe smart meters which could be read the meter reading at the billing center and print the bill. Thus it is clear that power supply relaibility could be improved tremedously with the use of modern IT facilities and smartness of the equipment to commumicate remotely.

In [14], a smart grid approach has been presented to improve future power system reliability. According to this paper, smart grid incorporates communication and control technologies to provide more efficient and reliable electricity to customers. The infrastructure of such power system will allow the customers to generate and store electricity, and use that in case of an outage or disconnection from the utility. Therefore, the outage of power from the utility side does not necessarily result in loss of electricity. The method to calculate the reliability indices, such as SAIFI and SAIDI, based on a Monte Carlo simulation has been explained and results were provided for case studies with residential, commercial, and industrial customers, renewable generation, and battery systems.

The information which could be dervied from the above paper is that imrovement of power supply relaibility adopting communication technology to the power system equipment is a future challenge.

### 2.5 Problem Statement

The literaure review stated above has been identified various problems related to power supply reliability. Eeach section 2.2, 2.3 and 2.4 presented the achievements and the limitations of the key finndings in each case presented in research papers. Even though many methods have been available for power supply reliability evaluation and

improvement, a minimum attention has been given to address the issues in Sri Lanka context. In this research, it is intended to addresse this issues and make contribution to develop method for power failure analysis and relaibility improvement in CEB distribution network.

The Distribution division of CEB is divided into four Regions. The four regions consist of 12 administrative provinces and each province comprises of three or more areas depending on the consumer density in the province. Each area consists of two or more Consumer Service Centers (CSC). Consumer services like providing new service connection, electricity line maintenance, power supply failure restoration etc. are done at the consumer service center. Each CSC is headed by a technical officer called Electrical Superintendent.

IT facilities have been introduced to CSCs with the development of Servie Connection Data Mamagenement system which handles the service conenction application processing and consumer account generation. Power supply interruption reporting mechanism has been introduced a few years ago. However, there is no proper way to analyse each power supply failure, which is gives the basis for power supply reliability improvement. If such system could be developed gives lot of informtion such as most frequent cause of failures, the weakest area of the network, the average time for failure restoration, average number of failures per consumers etc.

### 2.6 Summary

This chapter presnted a comprehencive literatiure review on the power supply reliability assessment and method for improving the power supply relaibility. The chapter focussed on ealy development, modern trends and future challenges for power supply realiability assessment and improvemnent. Next Chapter will discuss the existing manual method for power supply failure data handling and analysis.

-

# Chapter 3

# Existing Manual Method for Power Failure Analysis and Reliability Monitoring

## 3.1 Introduction

The power sector in Sri Lanka is organized under the Ministry of Power and Renewable Energy. Ceylon Electricity Board (CEB), established by an act of Parliament in 1969, is the statuary body responsible for most of electricity power generation, Transmission and major part of distribution of electricity in Sri Lanka. At present 98.5% of the population has access to electricity from national electricity grid and there are about 5.65 million consumers are provided electricity.

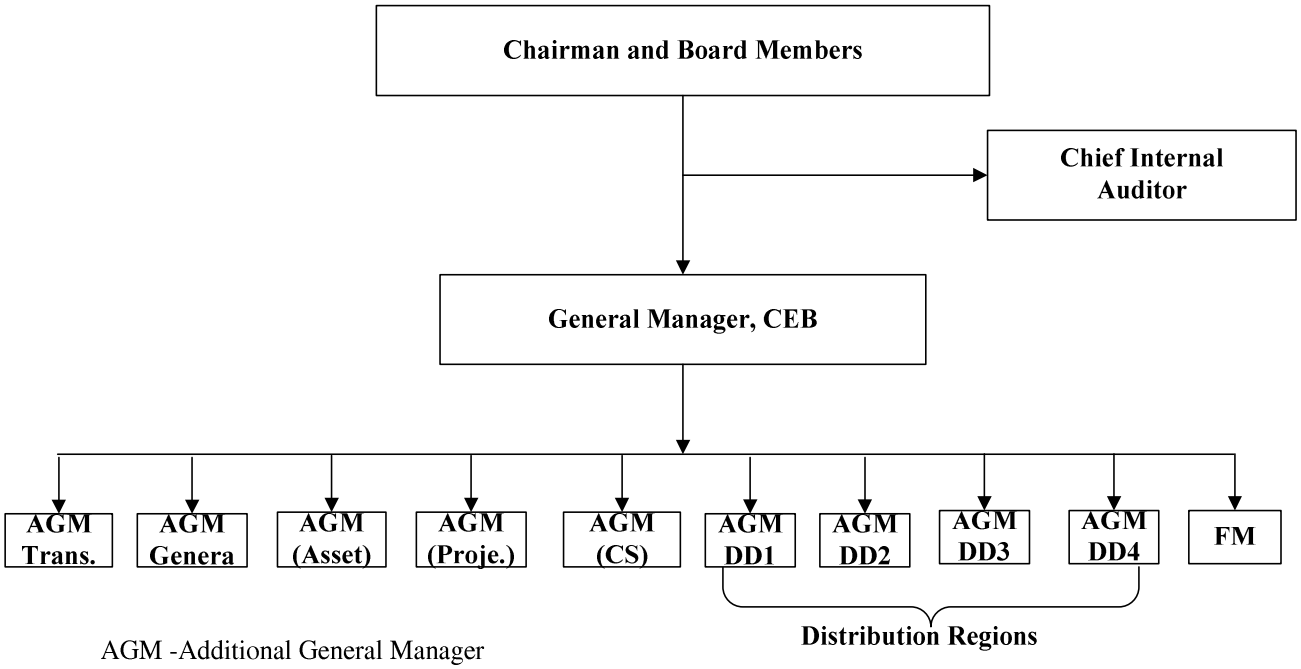
The main objective of CEB is to provide adequate power supply with better reliability. In this regard, the CEB needs to maintain its electricity network to achieve specified level of reliability while upgrading the same for catering the growing demand. Thus the electricity failures should be minimized to maintain supply at higher reliability. In order maintain the reliability level at specified level, it needs comprehensive monitoring and analysis of power supply failures in the electricity network including individual consumer premise. This could be achieved through power failure analysis and monitoring the power supply reliability.

This project focusses on the development of supply failure database, analyze the data for calculating reliability levels and identify the areas of the network to be reinforced.

## 3.2 CEB Organization structure

The top level hierarchy of CEB is given in Figure 3.1. The generation sector and transmission sector of CEB are headed by two Additional General Managers while distribution sector which has the interface to customer was divided into four distributions as given in Figure 3.2. Each distribution division has led by an Additional General Manager. The distribution is again divided into provinces and provinces are divided into

areas for making administration. The network operation and maintenance are handled by consumer service centers (CSC). Divisional Structure is shown in Figure 3.3




AGM -Additional General Manager
DD -Distribution Divisions
CS-Corporate Strategy
Trans. -Transmission
Genera. - Generation
Proje. -Project
FM - Finance Manager

Figure 3.1: Organization Structure (top level) of CEB

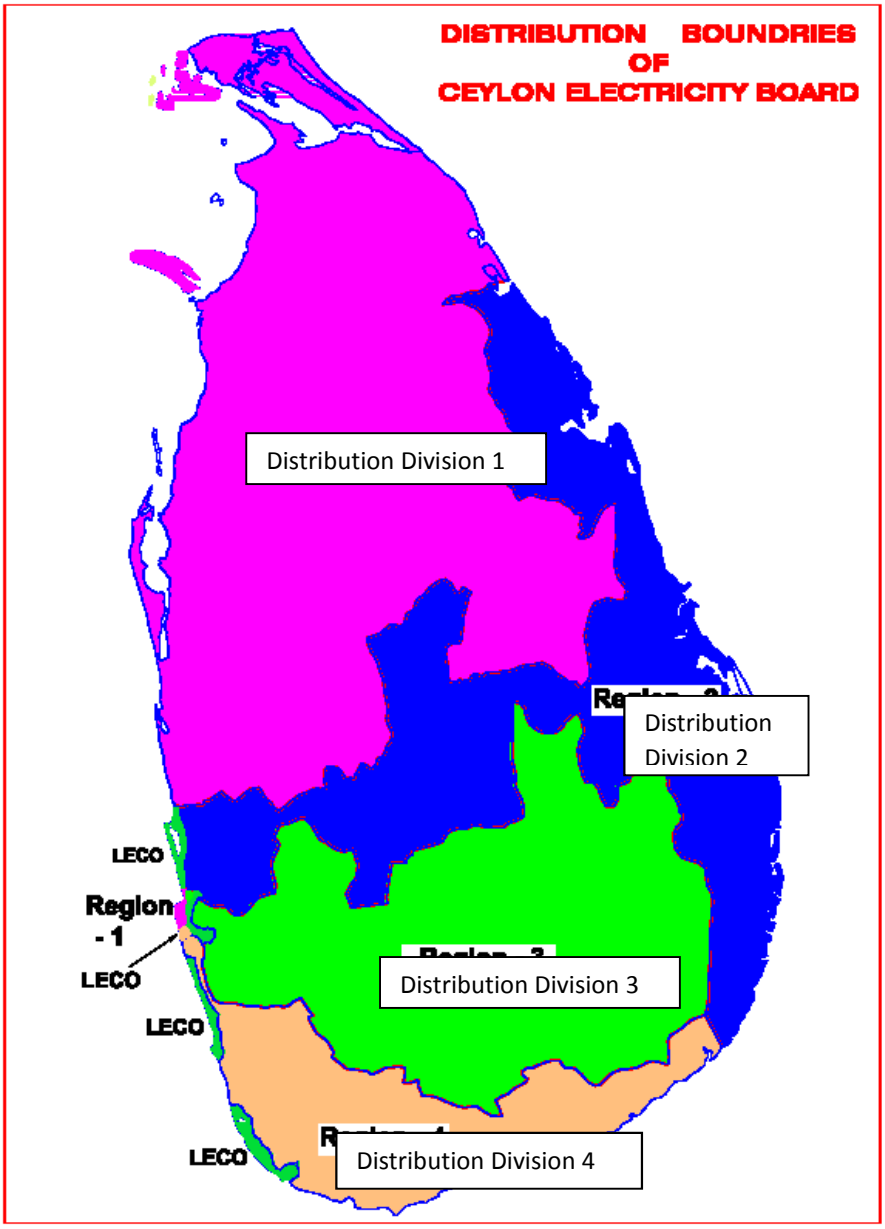



Figure 3.2: Geographical Demarcation of Distribution Divisions

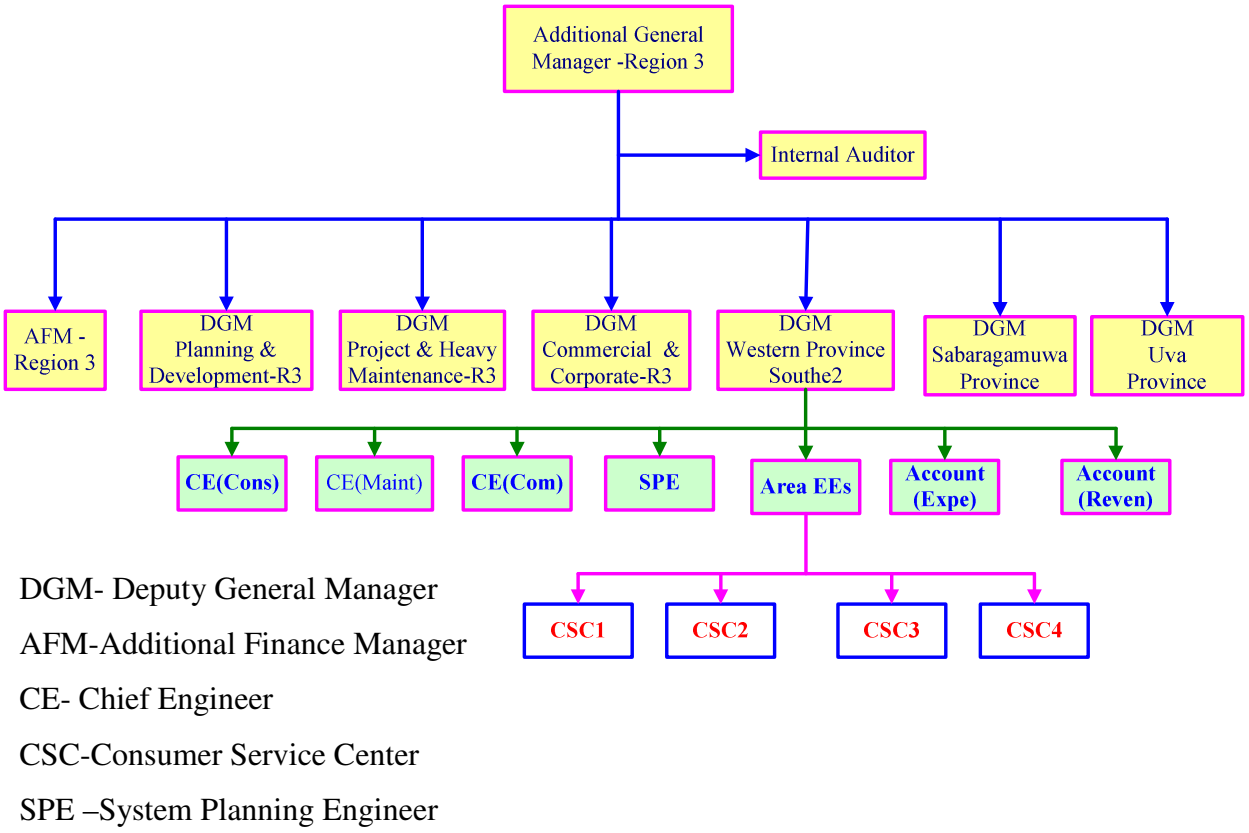



Figure 3.3: Present Divisional Structure of CEB

### 3.3 Existing Failure Analysis and Reliability Monitoring System

The customer interface to CEB for supply failure, new connection etc is the Consumer Service Center. The function of consumer service center could be described as in Figure 3.4.

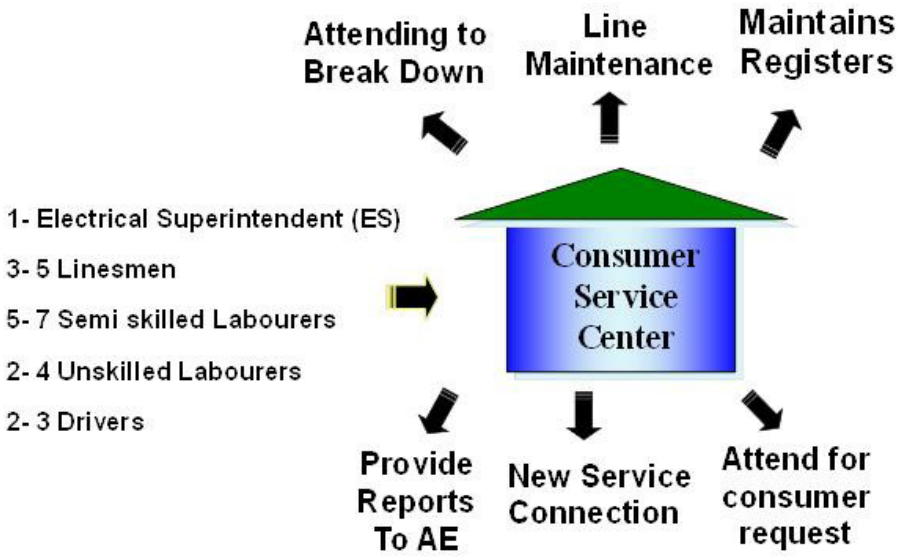



Figure 3.4: Functions of CSC and Staff allocation

It should be noted that one of the major function of CSC is attending to supply failure (Supply Breakdown). When there is an electricty supply failure, the consumer will inform the failure to call ceters established for power supply failure reporting. Then the call center will pass this information to relevent Consumer Service Center with the infomration like, time of failure, location, contact details of the person who passed the message regarding the failure, the way to reach the location and type of failure if he knows about it. After recording the failure, the technical staff at the duty will be allocated the supply failure for the restoration. Then the CSC staff will visit the location and identify the cause of failure and location of failure. Once the failure is restored, the CSC staff will record the material used, time of restoration and other relevant information. This infomarion is then recorded in the register called breakdown Register and also pass the message to call center on failure restoration. The information flow diagram describe above is given in Figure 3.5

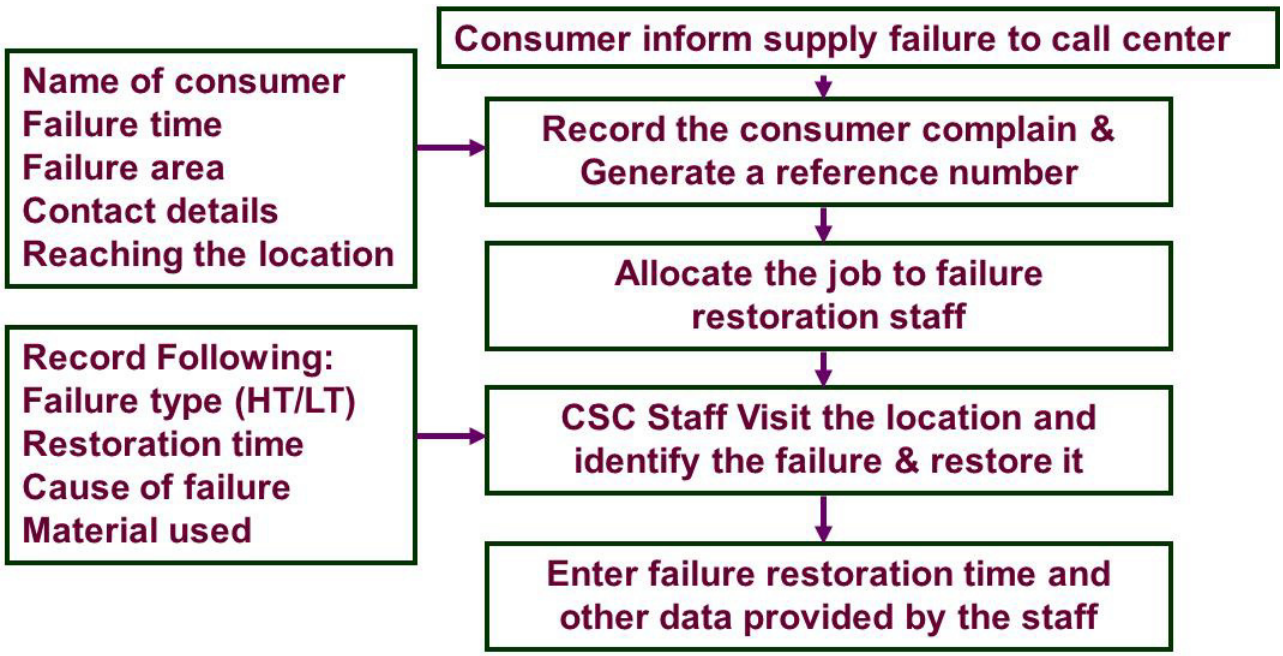



Figure 3.5: Information Flow Diagram of Supply Failure Restoration Process

At present failure restorations were recorded manually and in hard copy data registers. Hence, the technical staff as well as other staff were loaded with documantaory work and it takes long time for record keeping and retriving for report and other purposes. Also the register was not mainatined properly due to several reason and it was very hard to get clear understanding of the failures in the netwrok using the same recorded data. Thus, preperation of maintenance program, upgrading network and rehabilitaion could not be done in an optimized manner. The problems in the present manual system could be summarised as follows.

**Problems in the present manual system**

(1) Has to maintain Registers for breakdown recording and reporting

- A large book is used for breakdown register

(2) Longer time for breadown record searching and processing

- Finding records is very dificult in the breakdown register

(3) Report preperation is difficult and time consuming

- Reports are prepared in different formats

- Same recorsd are repeating several times

(4) No proper identification of electricity network for immediate maintenance

- Finding most breakdon area, type of breakdown etc are difficult

(5) Diffucult ot trace temporary and permanently completed breakdown

- Some breakdowns are restored teporary basis
- They need to be made permanent

(6) Still electricity supply relaibility is not quantified

- No information is recorded on consumers affected to breakdown.
- Calculating indices require this infromation

(7) Difficult to get input for material forecast

- Material Requirement is seperately addressed
- Thus proper material management is not done

(8) Data could not be utilized for system improvements

### 3.3.1 Breakdown Register Data

As explained in the previous section, the breakdown (supply failures) data have been entered in a Register called breakdown register. Following fields (columns) were seen in the register

(1) Number

(2) Name and address of person who complained the breakdown

(3) Time of breakdown

(4) Location of the breakdown

(5) Type of breakdown

(6) Reasons for breakdown

(7) Restoration time

(8) Remarks

When the data in the register was analyzed, it was noted that breakdown could be categorized into 4.

(1) HT Line (High Voltage) Breakdown

The failure occurred in the high voltage (33 kV and 11 kV) lines

(2) LT Line (Low Voltage) Breakdown

The failure occurred in the low voltage (400 V) lines

(3) Substation Breakdown

Here the failure occurred in distribution substations

(4) Service connection breakdown

The failure occurred in service connection wire interrupting supply to single consumer

As per the breakdown register, possible reasons have been identified as below

(1) Way leaves
(2) Fallen trees at a large distance
(3) Lose jumper connection
(4) Inadequate space between conductors and entanglement of conductors
(5) Burnt transformer bushings
(6) Defect lightning arresters and insulators
(7) Birds and Animals
(8) Fuse burnt
(9) Circuit breaker tripped
(10) Underground cable faults
(11) Sabotage
(12) Vehicle accidents
(13) Broken poles
(14) Tail wire broken down
(15) Aging of fuses
(16) Bad weather
(17) Others

Some pictures obtained for above reasons are given below

Figure 3.6: Wayleaves

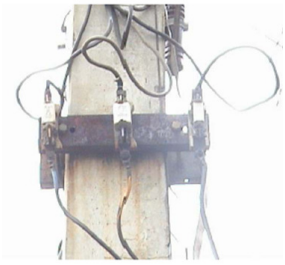


Figure 3.7: Damaged Fuses

Figure 3.8: Improper Connections

### 3.3.2 Electricity Network

Fig. 3.9 depicts the schematic diagram of the electricity network arrangement. It starts from power generation at certain voltage and step up for transmission voltages. Then the voltage was stepped down for distribution to consumers. The transmission voltages are 220 kV and 132 kV while the distribution voltages are 33kV, 11 kV and 400 V.

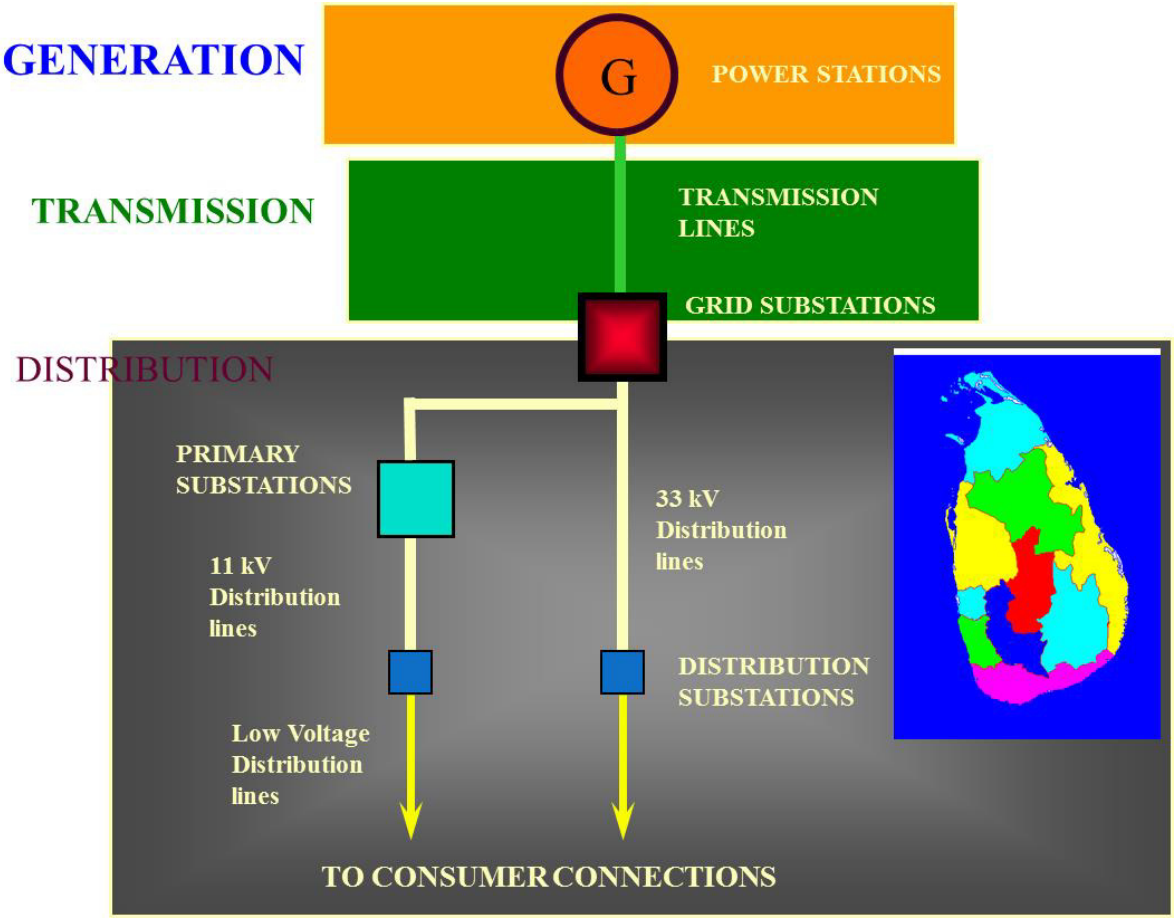



Figure 3.9: Electricity Network Arrangement

### 3.4 Summary

This chapter presented a brief dscription on the power supply utility, the CEB, its organization structure and the place where the supply failure matters were handled. Further, power supply failure recording system, analysis and report preperation approach at present were discussed. Moreover, it was highlighted the main drawnacks in the present manual method for power supply failure analysis and reliability monitoring. Next Chapter will discuss the proposed database design and development of computer software for the analysis of power supply failure and reliability improvement.

--

# Chapter 4

# System Design Approach for Failure Analysis and Reliability Monitoring System (FARMS)

## 4.1 Introduction

In chapter 3, it was described the drawbacks in the present manual system to handle power supply failure data. It was noted that recorded data was not fully used for power supply failure analysis and calculating reliability indices due to processing of records in the manual system. In this context development of computer database for failure data input, processing, report generating, output for reliability indices, netwrok improvements etc is a timely requirement.

In the first step, it is required to gather information on the recorded data and categorization of data for input design. The data could be categorized based on type of failure, voltage level, medium voltage, low voltage and cosumer premises etc. The electricity network is spread geographically over a wide area. A medium voltage feeder or electricty line will supply power to several distribution transfoemers while one distribution transformer will feed power to several low voltage lines with several consumers. Therefore the network too has to be arranged based on certain criteria. Following section will describe the approach in the database design.

## 4.2 Hypothesis

The hypothesis of this research is that the development of power supply Failure Analysis and Reliability Monitoring System could improve the power supply reliability to the consumers.

## 4.3 Requirements Analysis and Specification

This is the very first step of a system development methodology. The main requirements expecting from the system by the users and what are the boundaries to be covered by the system have to be clearly defined. Requirement analysis and specification can be divided

in to several phases such as Gathering requirements, system requirement specification, relevant constraints etc.

### 4.3.1 Gathering Requirements

Before start the system development, it is necessary to gather the requirements of the new system. System requirement definition activity is intended to discover the requirements. In the software requirement analysis, it involves consultations with customers and end users.

### 4.3.2 Coarse grain functional requirements

The basic function that the system should provide, must be defined. These are set out at an abstract level rather than in detail. Detailed functional requirement specification takes place at the sub system design level. For example, in the air traffic control system, this requirements activity would probably identify the need for a flight plan database. The flight plans of all aircraft entering the controlled space are entered in this database. However, the details of the database probably do not affect the functioning of other sub systems.

### 4.3.3 System properties

These are the non functional emergent system properties. These properties may be as availability, performance, safety, and so on. These non functional system properties affect the requirements for all sub systems in the process.

### 4.3.4 Characteristics that the system must not exhibit

It is sometimes as important to specify what the system must not do as it is to specify what the system should do. For example, in an air traffic control system, it might be specified that the system should not present the controller with too much information. This requirement implies that tests for information overload should be carried out. An important part of the requirements definition phase is to establish a set of overall objectives, which the system should meet. These should not be expressed in terms of the system's functionality but should define why the system is being procured for a particular environment.

### 4.3.5 Techniques Used to gather the information

Before going to start system development, there are things to study about the system. User will have good knowledge about the system. But system development team doesn't know what the system will do. Therefore they should collect the information about the system. That is called information gathering. There are some techniques to collect data. Following techniques have been used to gather information in the development of database system to analyse power supply failure and relaibility improvement.

- Observations
- Interviews
- Reference of related documents

In sometimes technique or techniques can be different depend on the selected system and availability of resourses. When the existing system is not available, they don't have to observe. When the users are not available, they don't have to interview

**Observations**

Observations are gathered by system analysis team or person from locations where the system related process are happening. The develoment team should understand how those process should be identified, when the existing process was statred etc. Once the system development team undestood the process, they could identify what is the problem in existing system and what should be the proposals in new proposed system. In this development, the process has been carried out in the consumer service center of CEB. Thus a visit has been made to CEB consumer Service Center to look into the present process and gather first hand infronmation and identify the drawbacks.

**Interviews**

This is also another method of information gathering and it is one of the most successful information gathering method. Those members who are involved in the process have good awareness of their day to day activities. The developer can individually meet the users and study what they are doing and ask many questions. Further, interviews could be

done having questionairs or preparing forms and filling them. A friendly environment has to be created before interviewing the people.

The types of questioners that could be raised are given below.

- What are the biggest problems you have faced in the manual system?
- What is your role in the faialure analysis data recording system?
- What are the problems you have faced when you are doing job?
- How do you do this section?
- What are the mistakes you have done and when you have done mistakes?
- How do you find the correct standard rates for the appropriate place?
- How do you identify the breakdwon types?
- How do you prepare report on failure data in CSC?
- How do you categorize the breakdonw into several types?
- What features you is expected from the new system?

Interview method can apply when the users are available and users have very good knowledge regarding their jobs. When the predefine process is not available or process is not clear when the user is changed, this method is very useful.

**Reference of Related Documents**

Reference of related document means that the system developer can gain information using the avaialble standard document which are used in the process at Consumer Service Center (CSC). It was noted that some procedures have been documented in the CSC. Using this document, the developer can get an idea regarding the existing process.

What are the reference documents for power supply failure analysis?

- Breakdown register
- Material price list
- Standard Construction Rates
- Cost Recovery Estimates

- Way bill book
- Material Requisition book

### 4.3.6 Identify Problems in the existing Manual System

- Lack of information about the past and current power supply failures.
- Process is handlled by ad-hoc method.
- Information flow is difficult to identify.
- Process will be changed depending on the place and person.
- Standards are not followed correctly.
- Technical people will involve clerical work regularly.
- Longer time for breadown record searching and processing
- Has to maintain Registers for breakdown recording and reporting
- Report preperation is difficult and time consuming
- No proper identification of electricity network for immediate maintenance
- Diffucult ot trace temporary and permanently completed breakdown
- Still electricity supply relaibility is not quantified
- Difficult to get input for material forecast
- Data could not be utilized for system improvements

## 4.4 Users of the System

There are four types of user categories. Activities of user category will be explained below.

The foure user categories are given below.

a. Entry User
b. Validate User
c. Approval User.
d. Admin User.

**a. Entry user**

This is the user who will enter the data in to the database system. To start any process there should be data. Data entry part is the main part and most improtant part of the

system. If the data is not properly enetered, the information gained from this data is wrong. There will be number of data entry users in the CSC.

**b. Validate User**

Validate user will involve to validate process. When the data entry user entered data in to system, that data could be validated by validate user. To finalize one process, there should be confirmation to move to next process. That will be done by the validate user. As an example, when the detaile estimate is prepared for power supply failure restoration, there is a process to recommend that detail estimate.

**c. Approval user**

Approval user will have full control to stop the process or implement the process. In sometimes approval user will be change depend on the finacial limits.

**d. Administrator**

Administrator has the full control of the system. He has the authority to submit privileges of all functions of the Power Supply Failure Analysis and Reliability Monitoring System (FARMS). He can define users, user functions, system security and database backups.

### 4.5 Input

The system can accept data through computer conencted to the system. A client server architecture has been adopted. Any user accessing the system should go through an authentication process. The system has been registered for house keeping.

### 4.6 Output

The output of the system could be obtained in different formats as defined based on the requirement. These output could be either soft copy format or hardcopy format.

### 4.7 Forecasting and Analysis

The output of the system is based on the past failure data and it could be used to analyse nature of power failure, most important reason for power failure, vulnarable area of electricity network, vulnarable devices like transformers, circuit breakers etc for failures

in the future, componnents replaced in quantity wise for restoring supply failures. Further, based on the analysis, it could be used for forcasting of probable area for supply failures in future, forecatsing of distribution transmformers which are subjected to fail in future. Moreover, it could be forcasted the ternds in supply failure in reason wise. Also, materials used for supply retoration could be forecatsed using the output of the failure analysis.

### 4.8 Process

The system executes a user authentication process before allowing a person to use the facilities of the system. This process goes verfication of the uer name and the password and the access will be limited based on the type of user.

### 4.9 Features of the Software

Following features are avaialbel in the designed software.

- Ability handle supply failure up to restoration
- Identify the weakest area in the power supply system
- Identify the level of power supply reliability in two voltages, high voltage and low voltage.
- Monitor nature of failure and cause of failure
- Eliminate duplication of work
- Estimate reliability indices for each zone
- Indicates measures to be taken
- Provide substation loading and unbalanced feeders
- Can generate MI reports
- Provide data for maintenance and augmentation plan

### 4.10 Sumary

This chapter presented the approach for the developement of database system to Failure Analysis and Reliability Monitoring System (FARMS). It was foucussed on rqeuirement analysis, identification of the problem in the existing system, user categories, input and output and features in the developed system.

# Chapter 5

# Design of System Model for FARMS

## 5.1 Introduction

Previuos chapter gave full picture of the approch in the database system design. This chapter describes the design of database System for Failure Analysis and Reliability Monitoring System (FARMS). A client sever architechture as in figure 5.1 with multi level security will be used..

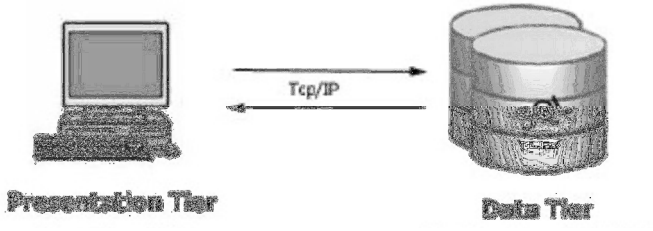



Figure 5.1: A client Server Architecture

Database will be installed in a database server while Application will be installed in a client machines. Since there are many locations that could log into the system, a client server architecture was used for this system. In the client server architecture, application will run in the client machine. Backend will be act as database. Main server is a database server. The entire request will do by the application from the client end. Processes depend on the client request will be handled in the database server. Only the result will be coming to the client machine from the server. All the validation and logics are handled by the client machine. In this way any number of client machines can be connected to the database server at a time.

Here the the data categorization, network arrangements, dataflow diagrams, Entity Relation diagram, database sytem etc will be discussed in detail

### 5.2 Network Arranagement and Inormation flow

As explained in previous section, it was noted that data has to be arranged in certain order for data collection and database system design. In this regard, the electricity network was divided as given below for data processing and analysis. The distribution substations have been uniquely identified its Substation Identification Number (SIN). The consumers could be tagged to this substation. This SIN has been included in the consumer electricity bill. Figure 5.2 depicts the network sectioning giving section numbers and division for zones based on the geograpghical spread of the feeder. It is known fact that electricty lines are running in to different geographical area like, through paddy fields, jungle areas, heavilly populated areas etc. Thus zones could be divided using such criteria. In the case of sectionining, it could be done based on network equipment, sectionalization of the network with switching arrangement etc.

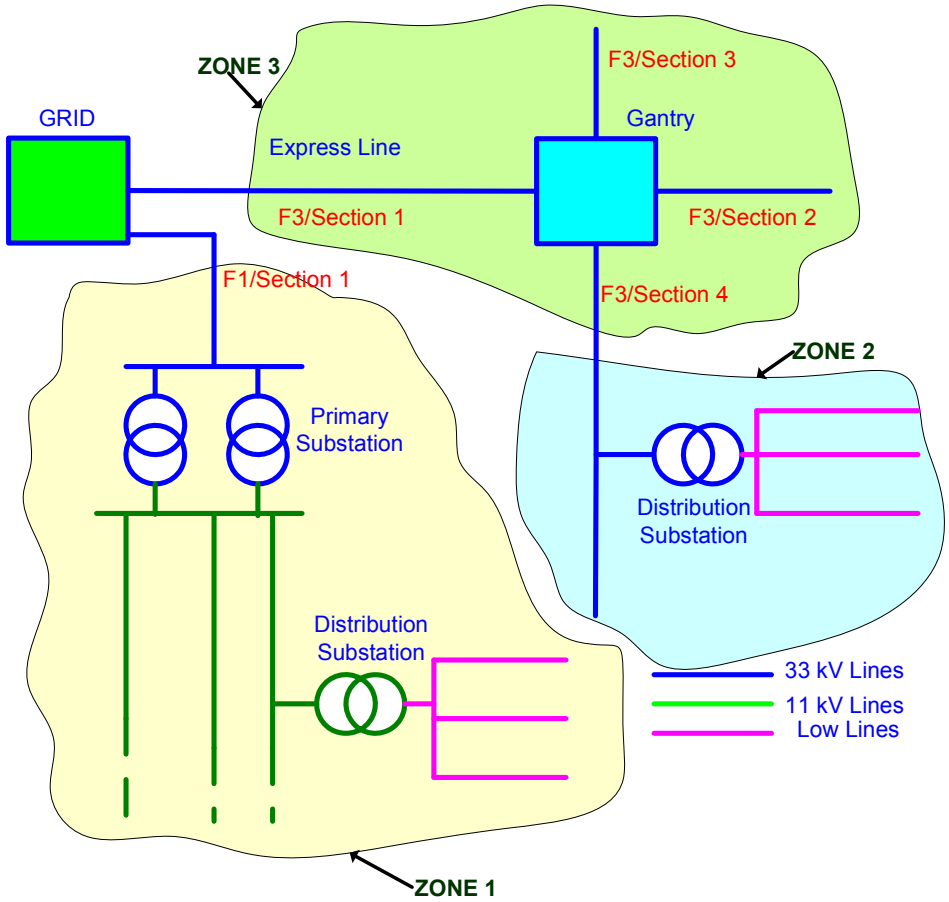



Figure 5.2: Network arrangement for sectioning and division into zones

In oredr to model a system, it is required to identify the sources, processes, flow of information etc. Thus sequence of information for handling electricity failure has been depicted in figure 5.3 while Context Diagram(CD) and Data Flow Diagram (DFD) for FARMS are shown in figure 5.4 and figure 5.5 respectovely.

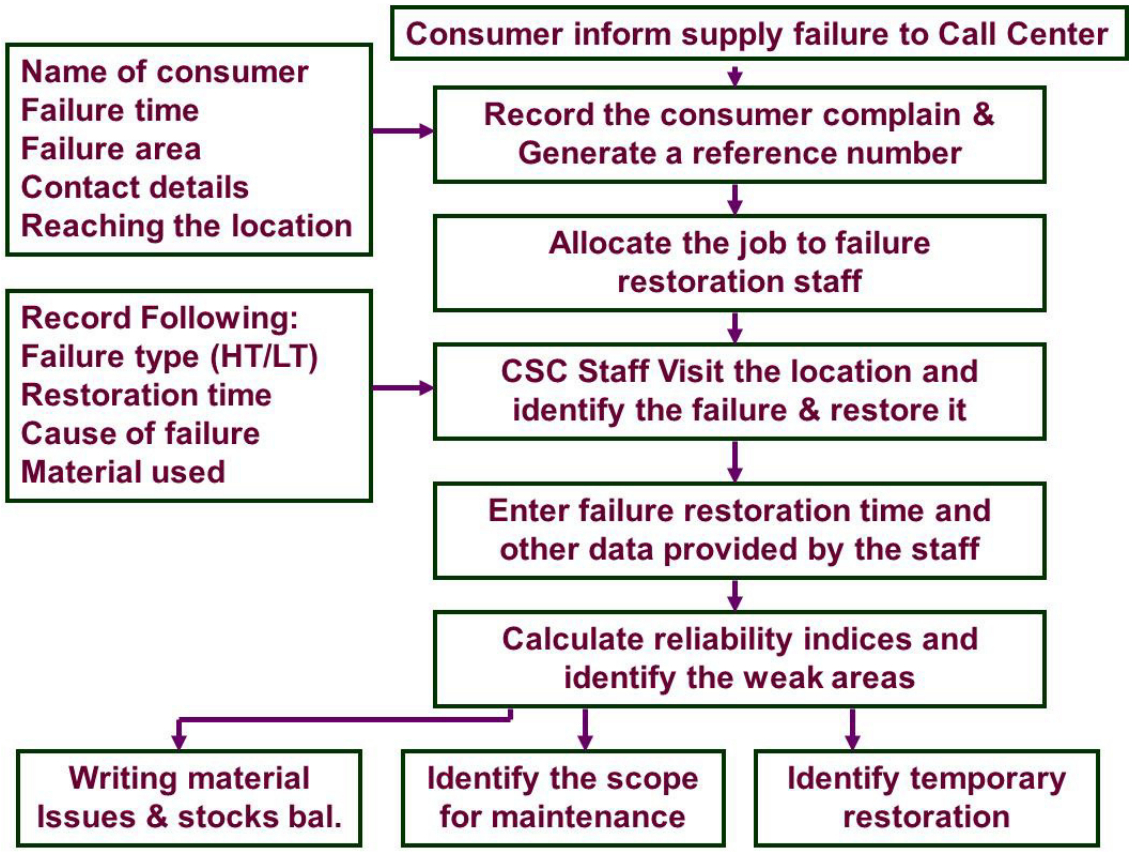



Figure 5.3: Dataflow diagram for the development of computer software for failure analysi

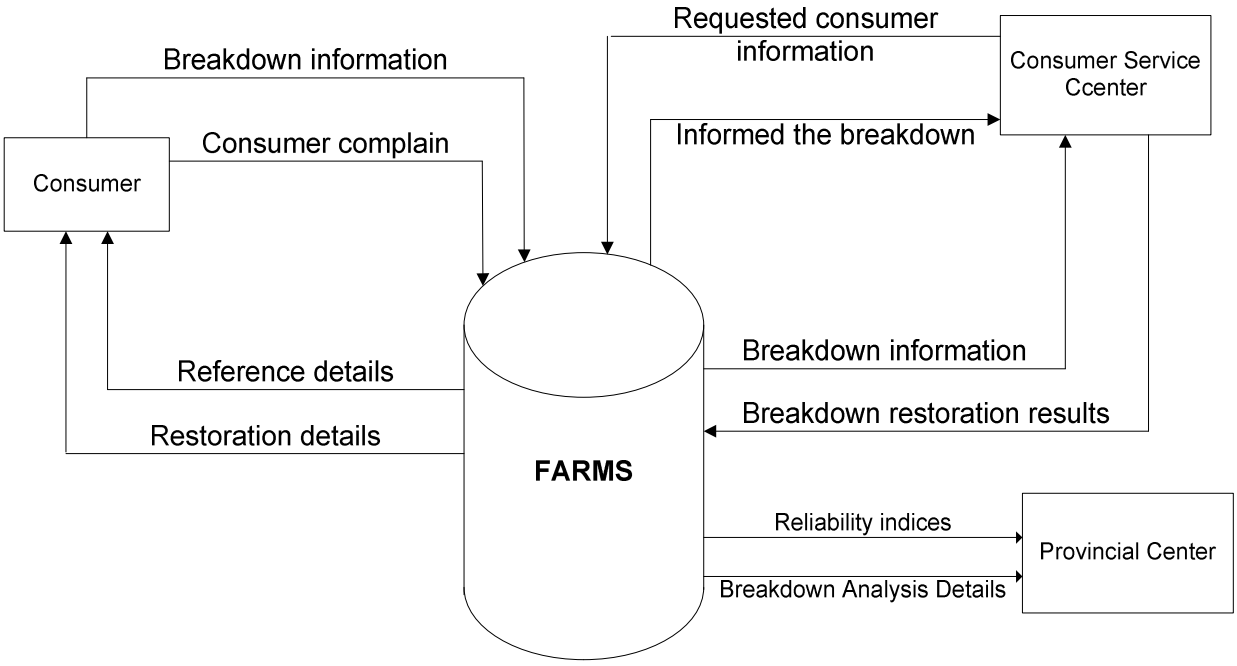



Figure 5.4: Context Diagram for FARMS

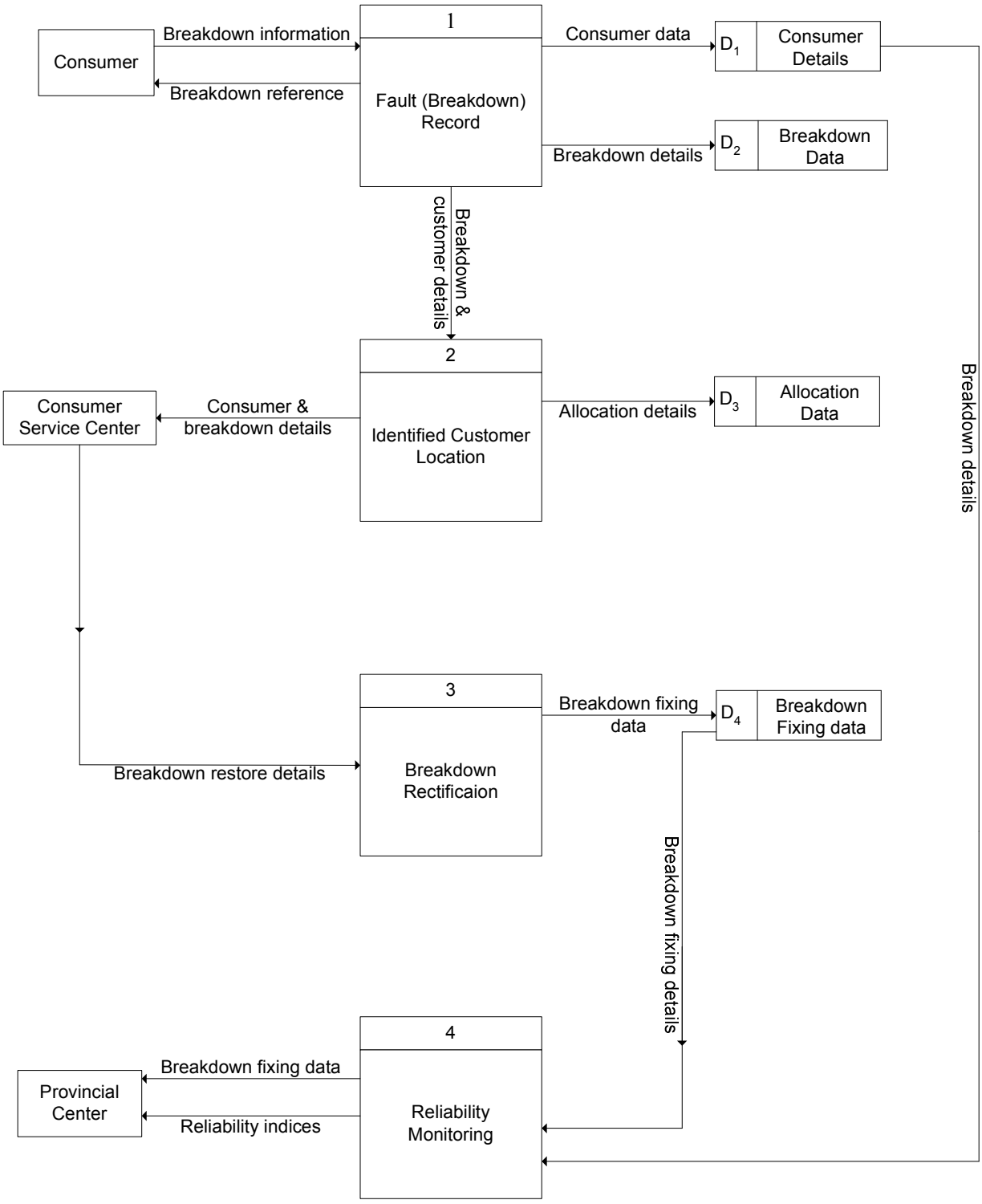



Figure 5.5 Data Flow Diagram for FARMS

### 5.3 Entity Relationship Diagram (ERD)

In an information system, data is taken, processed and represented based on the user requirement. Thus in an information system data should be stored in a well structured manner and a database has been developed to store data. In the database development, an entity relationship diagram was used to build the concepts of logical database. An entity relationship diagram is a data modeling technique that creates a graphical representation of the entities, relationships between entities of an information system. The three main concepts of ERD are:

i. Entity
ii. Relationship
iii. Cardinality

**Entity:** The entity is an object, place or an event for which same related data is grouped. When the entity is considered, same related data will be grouped in the single entity. For an example, in an information system for business, entities would include not only suppliers, but the supplier name, contact details, and invoices, deliveries as well. The entity is represented by a rectangle and labeled with a singular noun.

**Relationship:** The relationship is the interaction between the entities. A relationship captures how entities are related to one another. In the example above, the supplier makes a delivery, so the word "makes" defines the relationship between that instance of a supplier and the delivery or delivery that they will make. A relationship may be represented by a diamond shape or by drawing the line connecting the two entities. In either case, verbs are used to label the relationships.

**Cardinality:** The cardinality defines the relationship between the entities in terms of numbers. An entity may be optional: for example, a supplier will deliver one or more items. One item may deliver one supplier or more suppliers or mandatory: for example, there must be at least one item deliver by a supplier.

There are several different types of cardinality notation; commonly used notation is crow's foot notation, In crow's foot notation, a single bar indicates one, a double bar indicates one and only one. There are three main cardinal relationships, one to one, expressed as 1:1; one to many, expressed as 1: M and many to many, expressed as M: N. The Entity Relationship Diagram for the FARMS is given in figure 5.6.

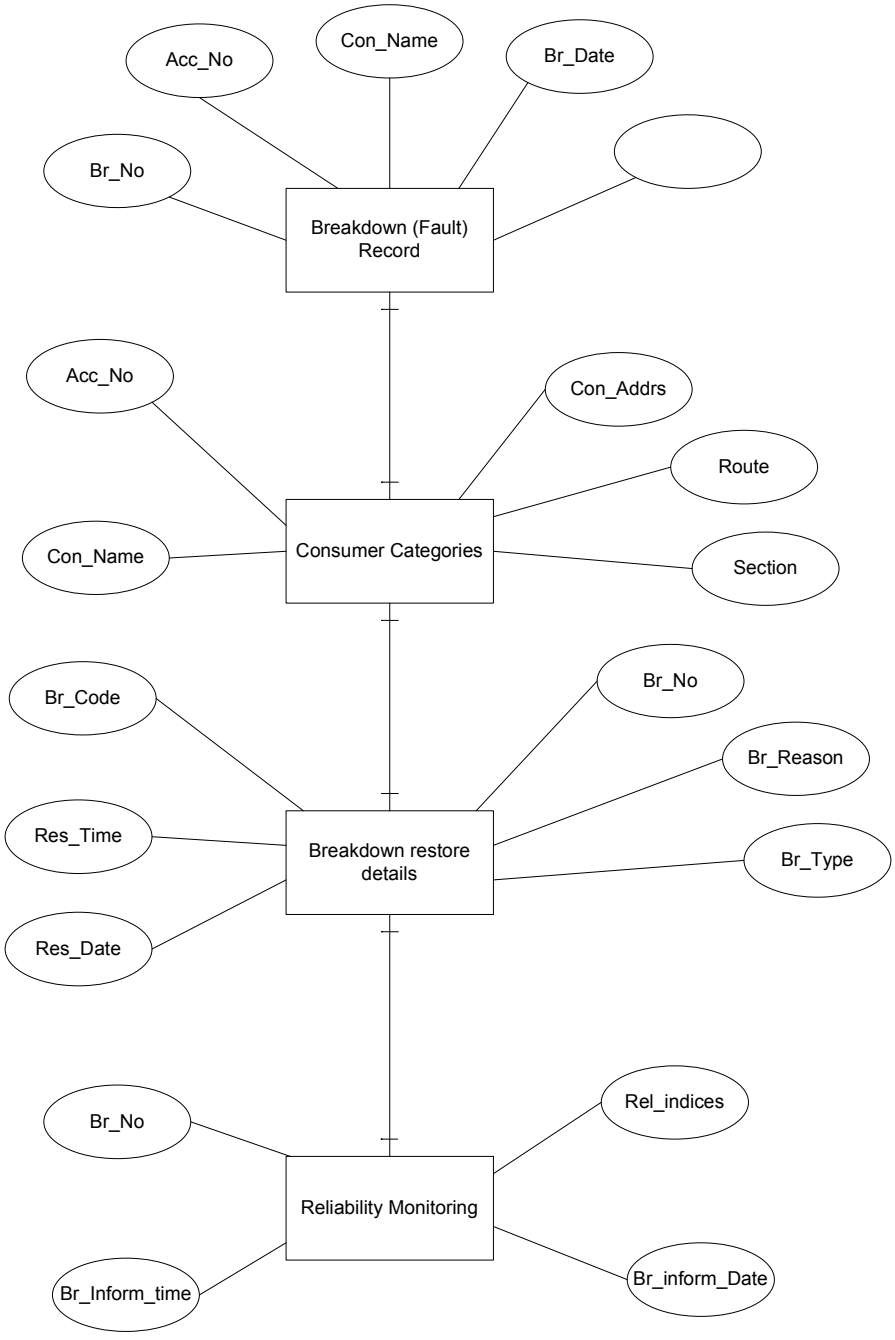



Figure 5.6: Entity Relationship Diagram (ERD)

### 5.4 Database Structure

Before designing the database structure, there should have clear idea of the whole system to be proposed. Further, it should be clearly identified the data flows, processors, inputs etc. First it is required to identify what are master data in the system. Master data means, those data not changing day by day. This will be changing periodically. As an example, standards rates will be changed annually.

After defining the master data, transaction data should be defined. Transaction data means, when the transaction happened data will be changed. In the dataflow diagram, it cleary shows how the data is used within each and every process and how the interaction between entities. Therefore dataflow digram will be converted in to our main database structure. At the first instantce, it may not be able to create an accurate complete database. Once the core database structure is made, it could be modified with minor modification to the database to meet our requirement. The main database of the FARMS is created by using the MS SQL Server 2008, because it supports client/server approach and provides multi user access for the system. This provides performance wise much better with the Microsoft based systems and compatible with other third party controls. Fig. 5.7 shows the database architechture. The system is so designed that server will be located at the area office and each consumer service center could access the same server from remote end as given in figure 5.7. The table structure of the database has been developed considering dataflow diagrams and the same has been depicted in Fig. 5.8.

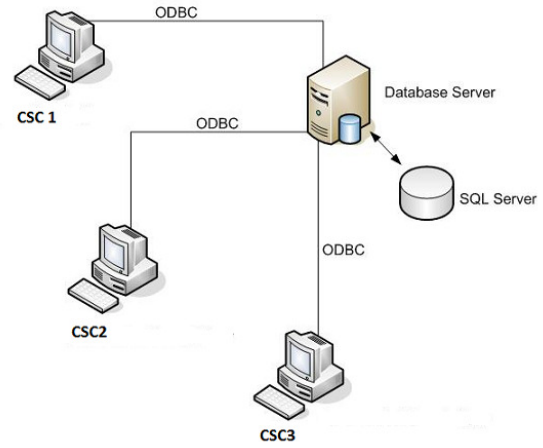



Figure 5.7: Proposed Database Architecture

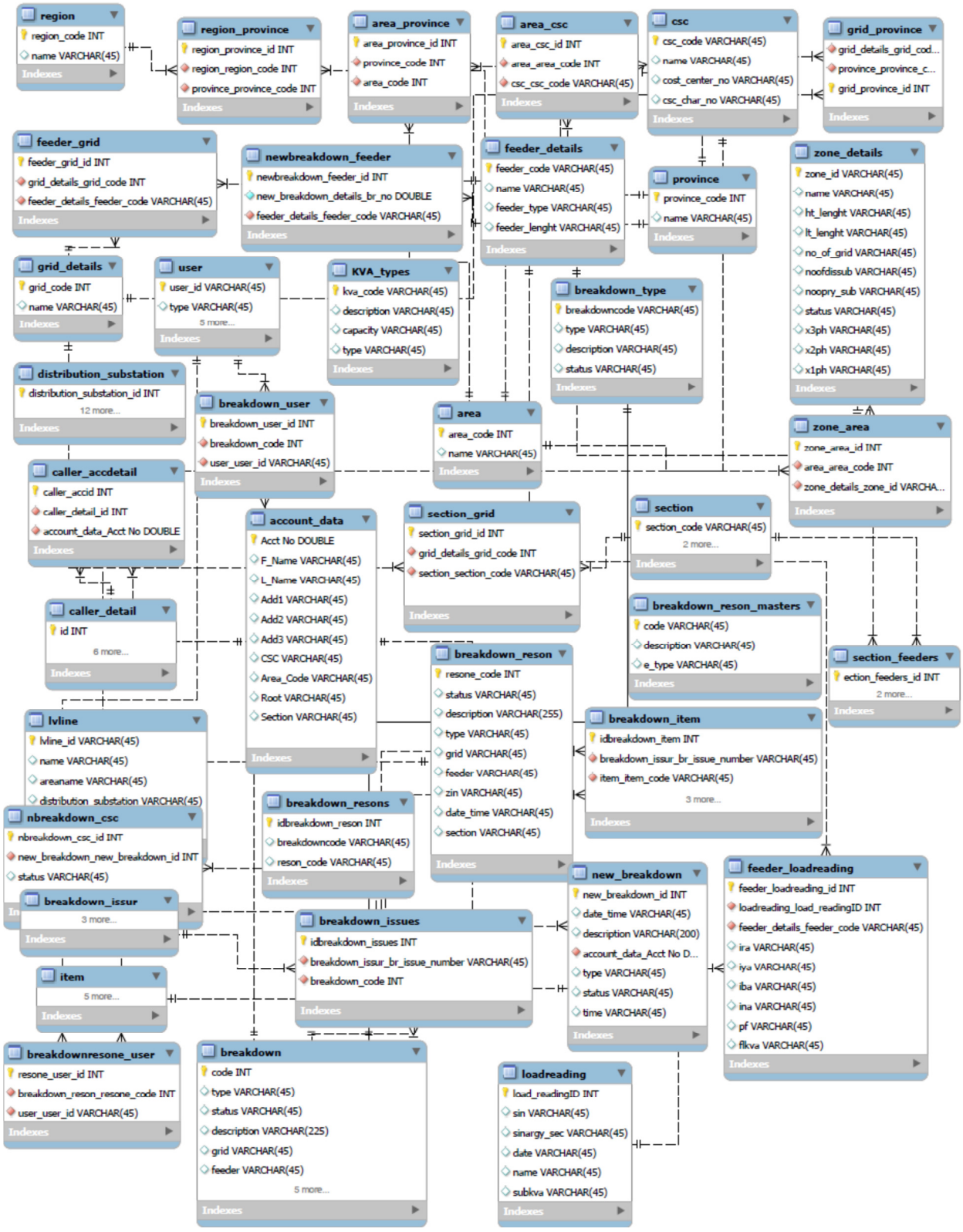



Figure 5.8: Table structure of the Database (EER Diagram)

### 5.4.1 Database & Tables

Tables can be categorized as master data and transaction data. Master data are the data which will be kept constant or updated once in a year or a given frequency (Example, regional data, area data, CSC data, material price etc.) while transaction data will change from transaction to transaction.

**Master Data tables**

**Regions or Divisions**

The distribution of electricity through out the country is managed by five Regions or divisions and four regions belong to CEB and the other one belongs to subsidiary company of CEB, Lanka Electricity company (LECO). These regions have divided based on the geographical nature. In the region table these information are stored.

**Provinces**

Each region is divided into two or three provinces for better operation and maintenance of the electricity network. In the province table Provinces details are stored.

**Areas**

Each province is divided into several areas for the proper management of electricity services. In the area table, information related to area is stored.

**Consumer Service Center (CSC)**

CSC is the place where the service connection, supply failure restoration, low voltage network operation and maintenance are carried out. Each area is divided into two or more CSCs and CSC information is stored in the CSC table.

**Material item**

In this table materials used for electricity network restoration and installation are stored. The item type, prices etc. will be updated annually.

**Breakdown types**

Possible types of breakdown are classified and stored in this table for the use in transactions.

**Breakdown Reasons**

Possible breakdown reasons are classified and stored in this table.

**Zone**

The geographical area of CSC is divided into several zones based on certain criteria like vegetation, paddy filed area etc.

**Sections**

Electricity network is divided into serval sections based certain criteria and it will be useful to identify most weak area of the network.

**KVA Type**

Transformer kVA is stored in this table

**Grid Substation**

Grid substation is the one which provide bulk electricity obtained from transmission network to distribution network. Grid substation details are stored in this table.

**Feeders**

There are several 33 kV lines emanating from a grid substation and these lines are called feeders. Details of such feeders are stored in this table.

**Distribution Substations**

In the distribution substations high voltage (33 kV or 11 kV) is stepped down to low voltage for consumer use. Such substation details are stored in this table.

**LV Line Details**

Many low voltage lines are emanating from a distribution substations for delivering electricity to consumers. These line details are stored in this table.

**Consumer accounts**

Account details of consumers are stored in this table and it will be an image of actual consumer accounts refreshed in daily or weekly basis.

**Transaction Tables**

**New breakdown**

An electricity supply failure details are stored in this table.

**Breakdown issue**

Material issued for breakdown restoration is stored in this table.

**Caller account**

Account details of caller are stored in this table

**Load reading**

Substation load readings are stored in this table.

### 5.5 Development Methodology

The processes and procedures are well defined in the existing manual system and hence, System Development Life Cycle (SDLC) methodology has been used for the FARMS development. This methodology is very robust and adopted in system development. Here, the development is divided into several stages and development is carried out one by one. There could be parallel development as well as series development.

There are many advantages as well as disadvantages in the SDLC methodology as given below.

Advantages

- All the processes will clearly identify.
- Complete system will be delivered.
- More time consuming means, Increase the system quality.
- End users will be involved from the beginning to end. User will be more satisfied.
- Best use for the more complex system development.

- Low project risk

**Disadvantages**

- More time consuming.
- End user will not have a system until whole project is completed.
- When the requirements are changing, this methodology is not suitable.
- When the whole project is not matched with the user requirements, discard the whole system and do it again from the beginning.

### 5.6 Development Tools

Development tool is one that is used to do design and develop the proposed system. Main development tool is computer language that will be used to write the computer instruction to understand the computer. There are number of languages to write the computer instructions. Those are Java, C#, C++, C, Visual basic etc. In the development application discussed in this thesis, Java with NetBeans IDE has been used. This software is an object oriented programming language and it has very good and attractive graphical user interface (GUI).

Microsoft SQL Server Management Studio (SSMS) is used as database management tool for MS SQL Server 2008 since SSMS is freeware and it is a better tool with a GUI that Windows OS users can straight away find. Further, it has an innovative feature called Object Explorer that allows the user to browse, select, and perform tasks on database objects. It's the only tool you need for creating and administrating Microsoft SQL Server databases.

### 5.7 Summary

This chapter presented the database design including the database architechture. This covers data categorization, network rearrangement, database system design, dataflow diagrams for desiging table structure, database architechture etc. The next chapter will focusses on implementation of FARMS.

# Chapter 6

# Implementation of FARMS

## 6.1 Introduction

In the previous chapter, it was described the databse design for implementation of proposed software ststem called FARMS. This chapter will be dedicated to the implementation of the designed database architecture. This covers the implementation of each part of the system architechture.

The implemented system is a process that will take data in to the system, process the data and output data what the user is required and when the user is required his own way. There should be facility to input and output/display data. To input and to output, there should be media. That media is called interface. User interface is the main component in a system that will communicate between system and user. Therefore numbers of interfaces are available in FARMS. Interface will be categorized as report format and screens.

## 6.2 Requirements Associated with user Interface Designing

FARMS will be developed to handle supply failure process in a Consumer Service Center in Ceylon Electricity Board under distribution divisions. This system will be implemented in different Consumer Service Centers attached to Area office of the CEB. There are many number of users attached to the system. Therefore user friendliness and simplicity is very important for the FARMS for successful impmetation. To achieve this purpose, user interfaces should be simple, understandable and readable. Designed user interface has been discussed in following sections. There are several screens to interact the user for different tasks. Coding of main components have been appended in appendix I.

### 6.3 User Interaction with the system

There are number of user types involved to the FARMS. Those user types have different authorization and all the users have to log onto the system with their user name and password.The administrator will be defining the user names and passwords for each system users. When the user logs in to the system, main menu will be displayed. Depend on the user defined levels; user can access the main menu functions.

Main menu has been divided in to Data, transaction and reports. Data section will be representing master data of the system. Transaction part will be representing brekdown entries, materail issues and breakdown restoration. Those sections are the most important section of the FARMS. Each subsection has simpy represented the process in the supply failure which is in line with manual system. The users those who have good experience in manual system could easily undesrtand and work on the system. System flow also same with the manual system. Report are very much essential for management decision and they can have in many formats. Reports will be generated based on the enterded transaction data.

#### 6.3.1 Main User Interfaces

Main Window or the home screen is the default interface of the Fault Alarming and Monitoring System (FARMS). This default interface has been categorize as follows. The main user interface has been depicted in figure 6.1.

- **Main Menu**

  In here, it will display all the functions that are available in FARMS and it Navigates place of the whole system.

- **Main Window**

  Main window is the background of the FARMS in every time when the system is running. All the current working interfaces are loaded in to this area.

- **Icons**

  Icons could be used to access directly, the assigned task and could reduce time acessing for the same using the main menu.

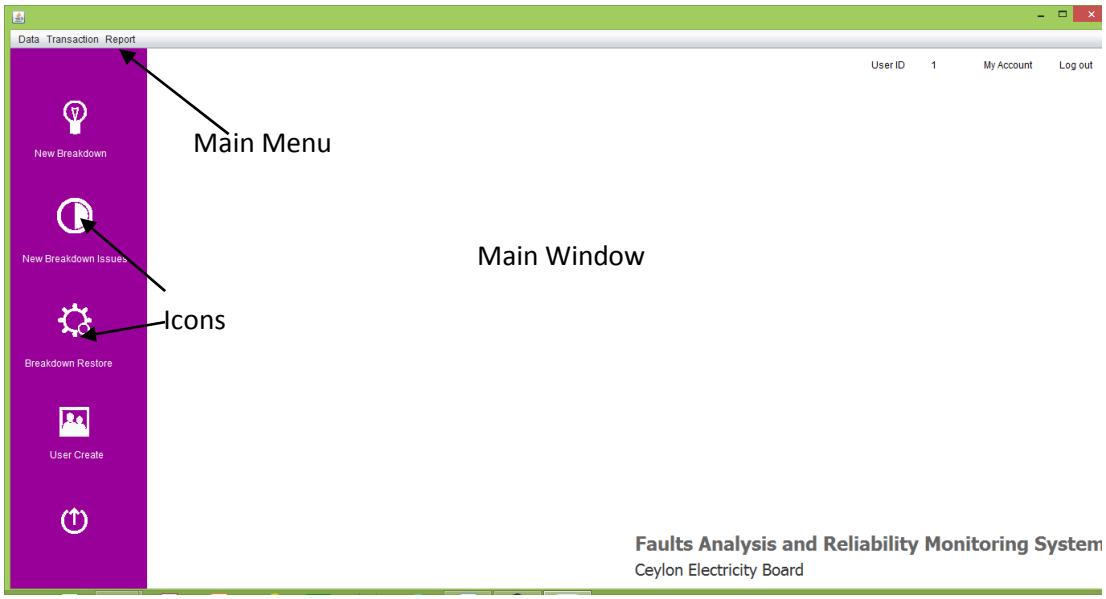



Figure 6.1: Main user interface

### 6.3.2 Accessing system

User authontication is required for eneter into the FARMS. New users are creaed by administrator giving a user name and password to all users. Using the correct user name and password allocated to each user he/she can login into the system using the login window. Login is the one of the security method in FARMS. Login interface in FARMS is shown in figure 6.2.

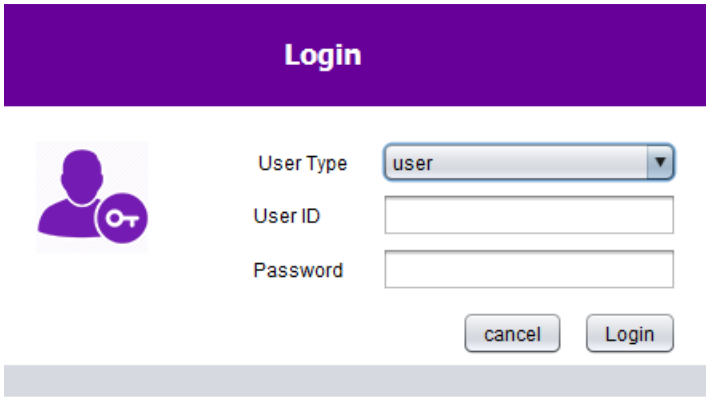



Figure 6.2: Login screen of FARMS

### 6.3.3 Data

Data has been categorized as follows master data and breakdown data. Master data is the one that initaillay require to run the syetm. Breakdown data are specific to breakdown and entered in advance for calling the same in the transaction stage. This will reduce typing the same in each and every transaction. Figure 6.3 describes menu for data.


Figure 6.3:Data menu of FARMS

Some of the essential master data entries are given below with the aid of the screenshot images of the developed system.

**Regions**

Regional details are entered to the FARMS through this interface. Entered regional data will display in the grid area. When the new record or new region is going to enter, clear the text controls by clicking the new and enter the region code and region name in the region inter face. Figure 6.4 describe the window for Regional data

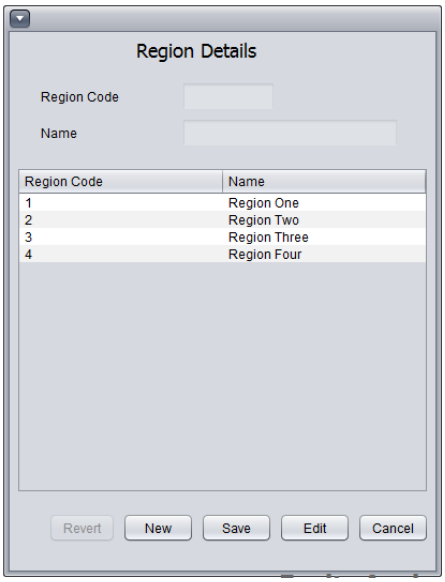



Fig. 6.4:Screen for enetring Region data in FARMS

Figure 6.5 to 6.19 displays the screens for province, area, CSC, users, Item, breakdown data, breakdown type, new breakdown, Zone, kVA details, grid, feders, sections, distribution subtation and LV feeder details.

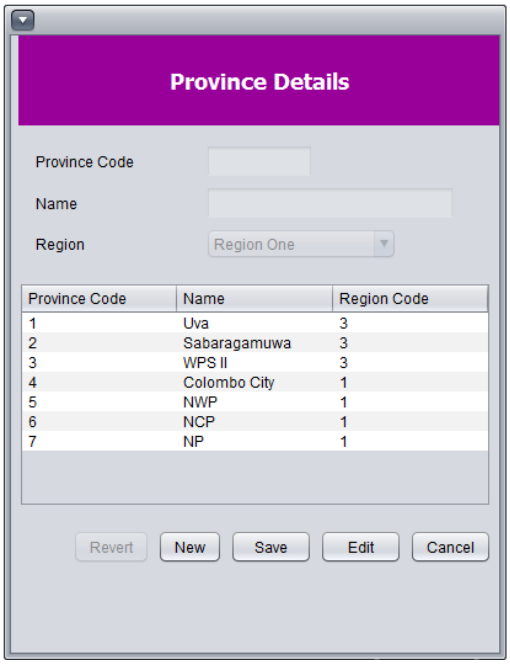



Figure 6.5: Province details screen

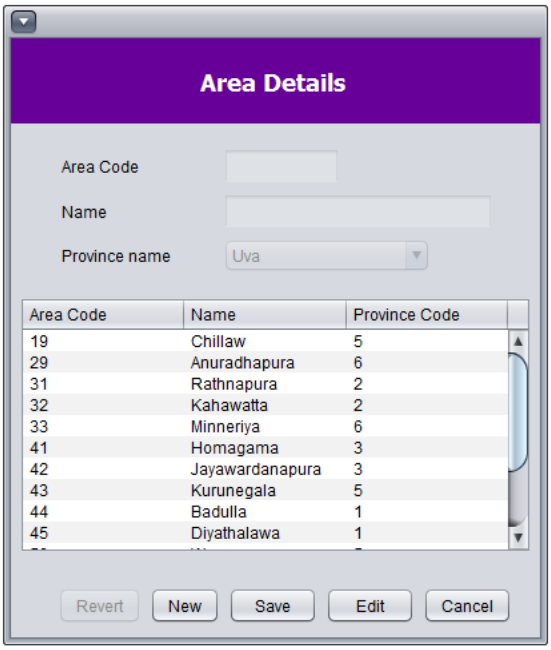



Figure 6.6: Area details screen of FARMS

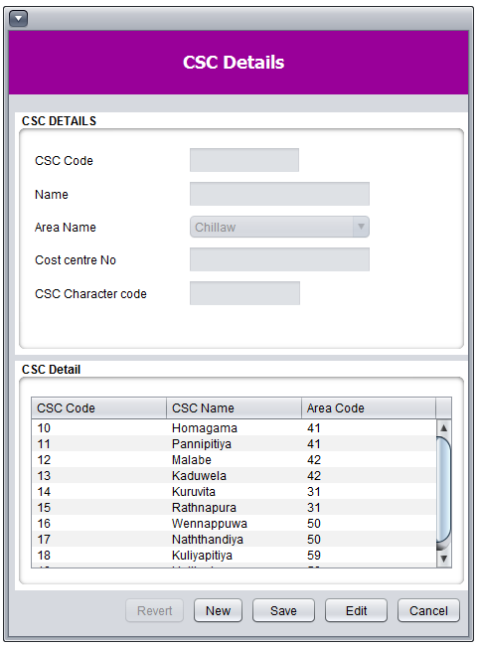



Figure 6.7: CSC details screen of FARMS

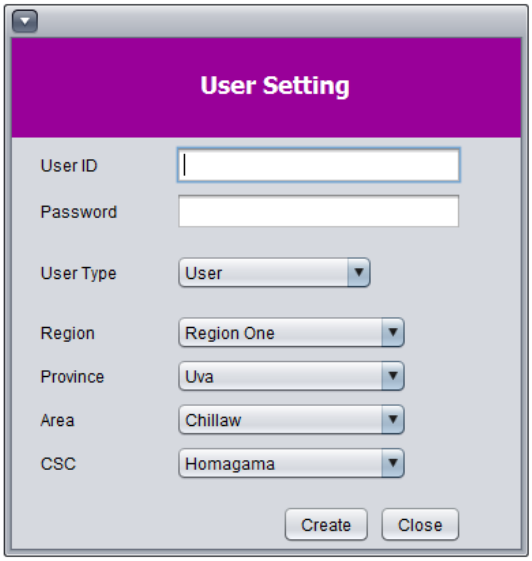



Figure 6.8: User setting screen of FARMS

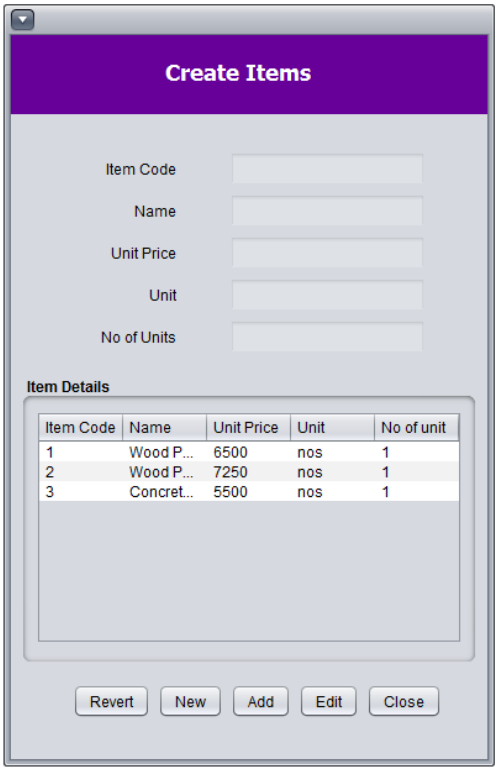



Figure 6.9 : Item entry/update screen of FARMS

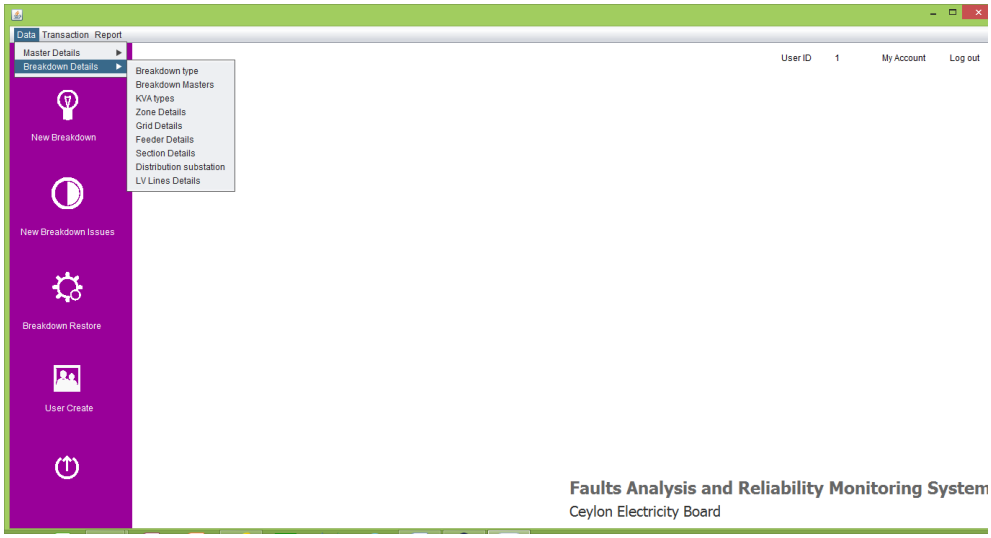



Figure 6.10: Entering breakdown data

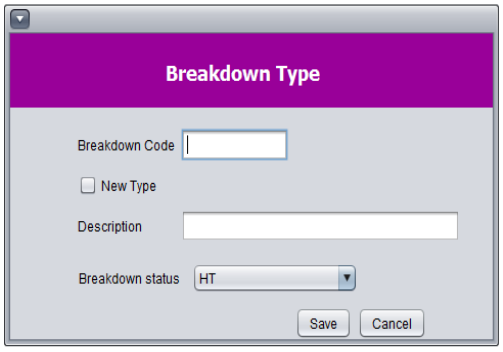



Figure 6.11: Entering breakdown Type

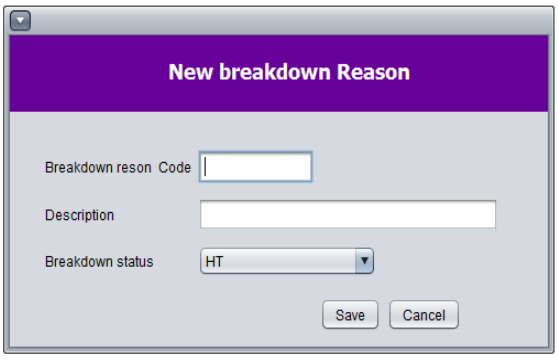



Figure 6.12: Entering breakdown Rreasons

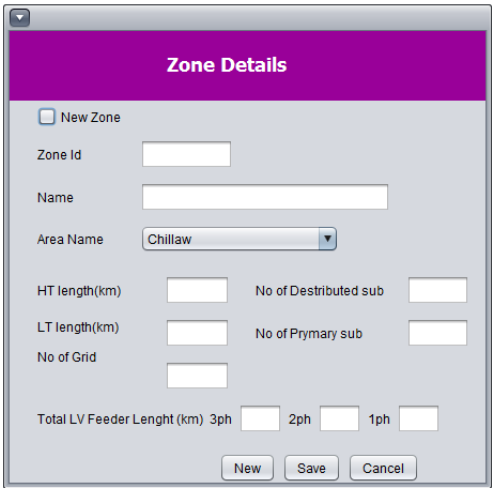



Figure 6.13: Entering Zone details

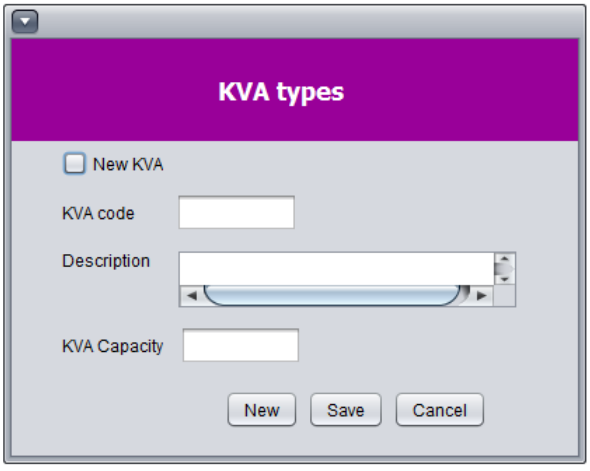



Figure 6.14: Entering susbtation kVA

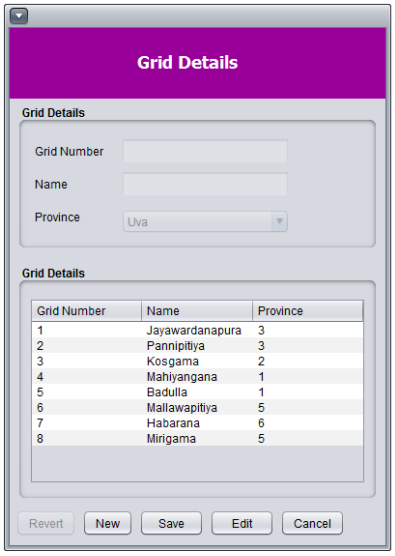



Figure 6.15: Entering Grid details

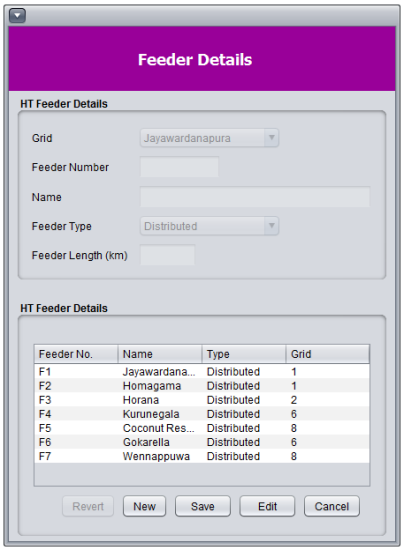



Figure 6.16: Entering Feeder data

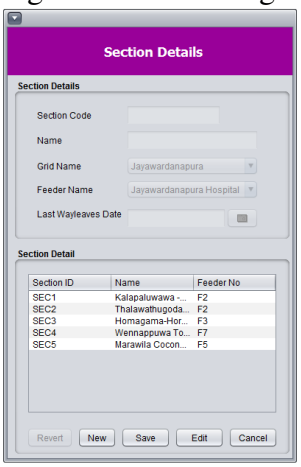



Figure 6.17: Entering Section details

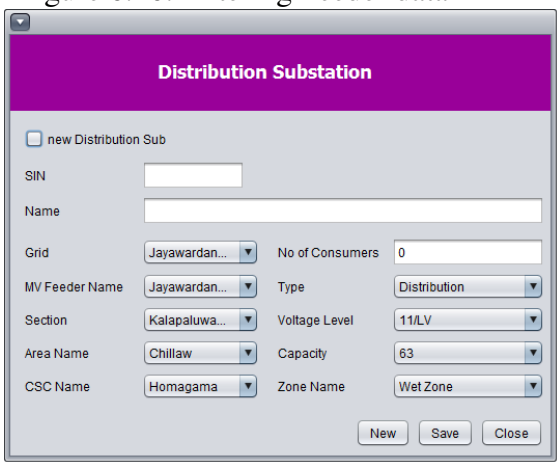



Figure 6.18: Entering subtation data

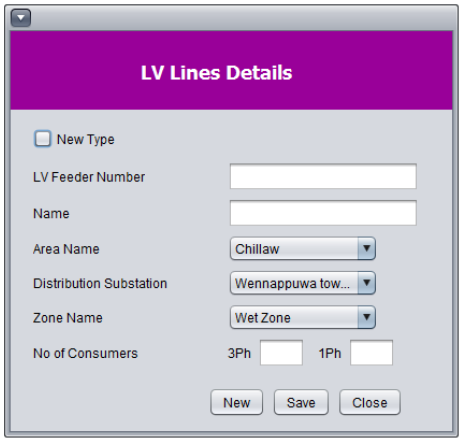



Figure 6.19: Entering LV line feeder data

All the master details are very important for the transaction process. These data shall be called each and every transaction discussed below and in some case to get single output, large number of master data will be used.

### 6.3.4 Transaction

In the transaction section, main activities or the processes are handled. As discussed above, master data will be changed periodically and the transaction data will be changed in each transaction entry. In the FARMS transaction part has been divided into new breakdown, breakdown issue, load readings and breakdown restore. The program screen shots have been appended below for main transactions related activities. Screen shots for transactions are displayed in figure 6.20 to 6.24.

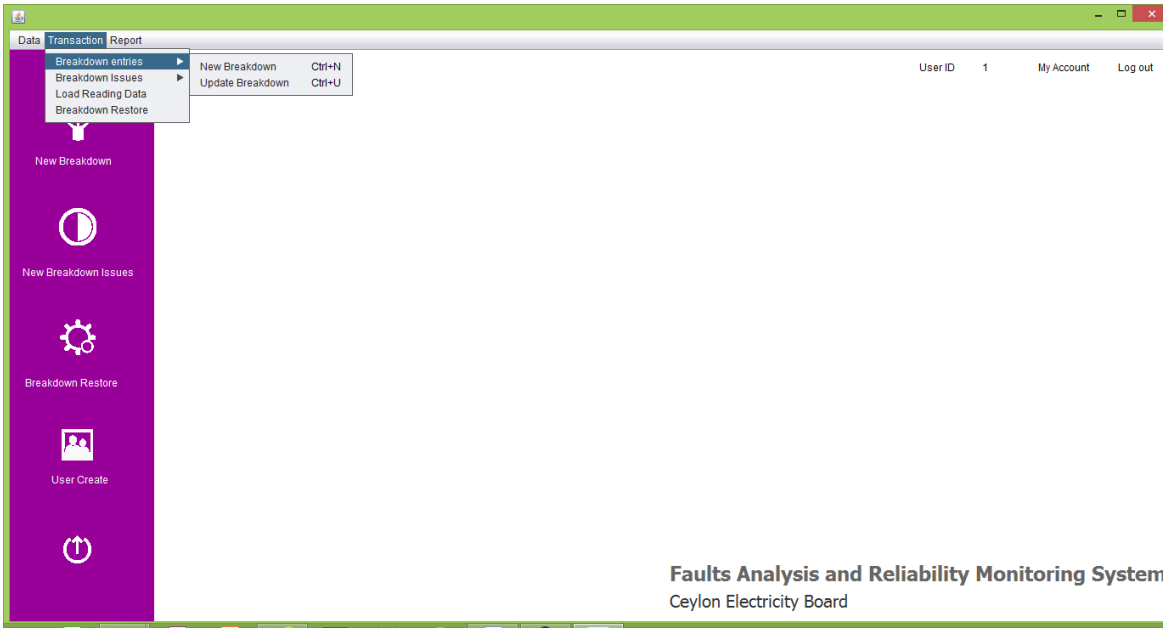



Figure 6.20: Transaction Menu in FARMS

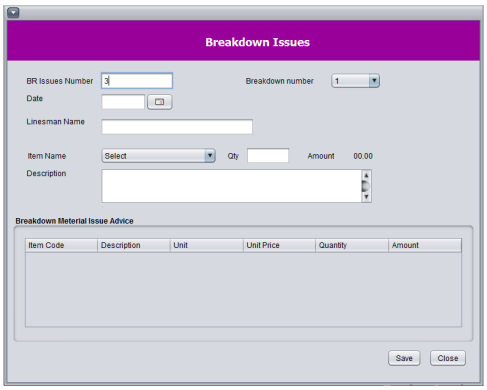



Figure 6.21: Breakdown isuue screen

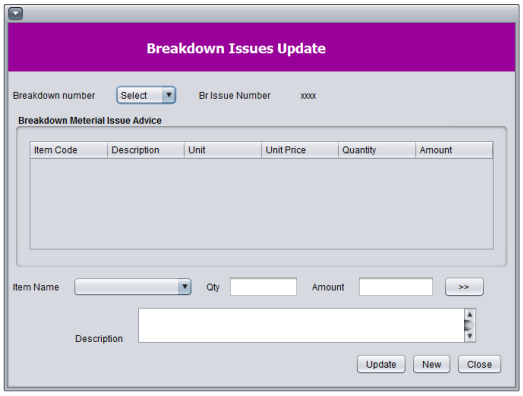



Figure 6.22: Breakdown isuue update screen

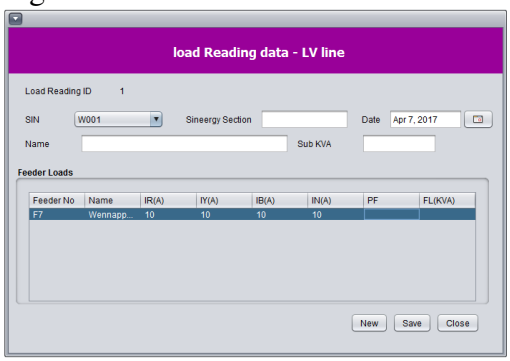



Figure 6.23: Load reading data entry

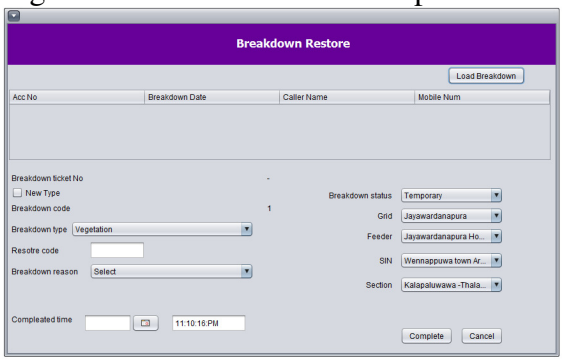



Figure 6.24: Breakdown Restore screen

### 6.3.5 Reports

All the reports that the format are given for management information (MI) are included in the report section. Further, new reports could be accommadated with the change of coding of software program since data is already entered in the database and it needs to be extracted in the given format. MI reports commonly used are given in figure 6.25 to figure 6.29.

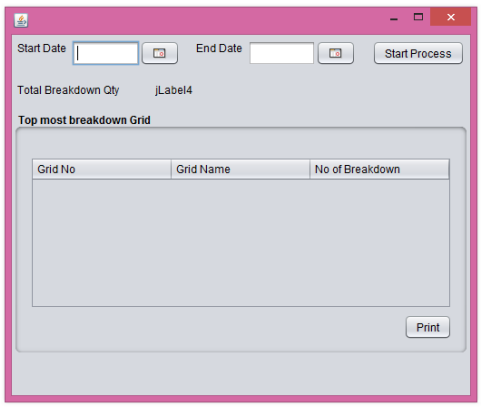



Figure 6.25 Report for top most breakdown grid

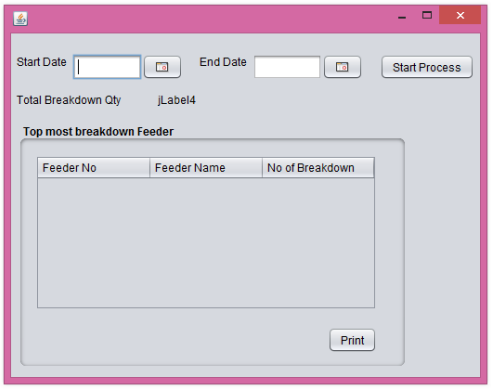



Figure 6.26: Report for top most breakdown feeder

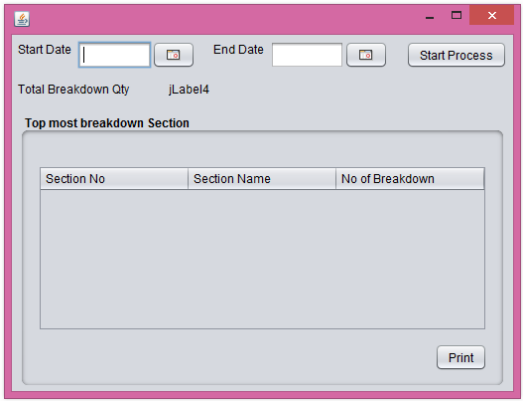



Figure 6.27: Report for top most breakdown section

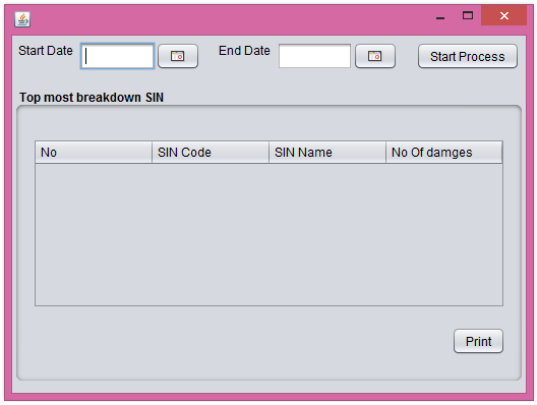



Figure 6.28: Report for top most breakdown SIN

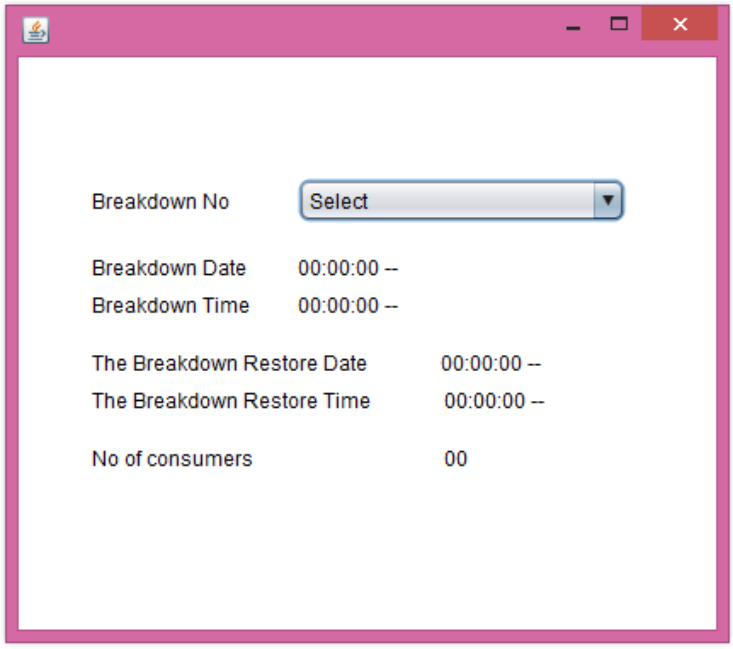



Figure 6.29: Detail breakdown report

### 6.3.6 Exit from the System

User can exit from the system by clicking the exit icon or close the main window. To exit from the existing interface click the button “Close”.

## 6.4 Implementation of the Server

Microsoft SQL Server 2008 is used as back end in Failure Analysis and Reliability Monitoring System (FARMS). Back end is software that will store the data whatever the

data used by front end. The database connection time is very low and it supports to fast data accessibility between front end and back end. MS SQL server also has good features to manipulate data in a database. Data insertion, updating, data navigation and deletion methods are very fast. That MS SQL server also has their own indexes. That will help fast operations.

All the database system should have recovery procedure as well. Because database is corrupted, database will not working properly and systems will be terminated. When the system is online, user will be faced troubles. When it happens, database system should recover within very short time and system should run. MS SQL server has well organized backup and recovery system within MS SQL server RDBMS. MS SQL server maintains the log files also. That will helps to correct the incorrect transaction as well.

## 6.4 Implementation Environment

To implement the FARMS, all the requirements should be satisfied. Requirements can be categorizes as follows. One is Hardware requirements and other one is software requirements. Required software specification and required hardware specification are as follows.

### 6.4.1 Hardware Specification

| Hardware | Client | Server |
|---|---|---|
| Processor | 1 gigahertz (GHz) or faster | 2 gigahertz (GHz) or faster |
| RAM | Minimum of 512 MB 2 | GB or above |
| Hard disk Space | 500 MB of available hard disk space | 15 GB of available hard disk Space |
| Graphics | Microsoft DirectX 9 graphics support with 1,024 x 768 | Screen Resolution 1,024 x 768 screen resolution |

**Software Specification for Client**

Microsoft Windows XP or above

**Software Specification for Server**

Windows Server 200x Rx or above

### 6.5 Summary

This chapter presented detail implemtation of the proposed software which incorporated main user interface, implementation of main server and client architechture. Finally, harware and software requirement for the software implementaion have been specified.

--

# Chapter 7

# Evaluation of FARMS

## 7.1 Introduction

This chapter discusses the evaluation of the FARMS developed to handle supply failure data in a consumer service center of the CEB. The evaluation will be done by devloping software system and actual collected data will be input to the system. The evaluation is to analyse whether the hypothesis can be sustained. Here it will be discussed the testing of system and validation of system data.

## 7.2 Testing

Testing is very important key area in system development. Testing process is the main key of making good quality system. Logical and any other errors can be identified by doing testing. Testing can be done in different stages in system development. Mainly testing can be implemented just after finishing the development of some module or section. When the system errors are high, end user will reject the system. Therefore before going to the end user environment, system should be tested and finalized the system bugs.

### 7.2.1. Test Plan

Before going to test the system, test plan should be prepared. Test plan is how testing process is implemented. When making the test plan, there should be deep knowledge about the system. Based on this it makes successful test plan to cover A to Z possibilities. If the testing process is handled in an ad-hoc manner, some conditions may be missed.

### 7.2.2 Approaches of System Testing

There are number of testing methods available for system to test. Some methods are covering some kind of errors. The main testing methods are

- Unit Testing

- Black Box Testing
- White Box Testing
- Entire System Testing

### 7.3 Features to be tested

Following features of the softeware have to tested.

- Entry/Update Feature validation
- View/Print Feature validation
- Ability to handle supply failure up to restoration
- Identify the weakest area in the power supply system
- Identify the level of power supply reliability in two voltages, high voltage and low voltage.
- Monitor nature of failure and cause of failure
- Estimate reliability indices for each zone
- Indicates measures to be taken
- Provide substation loading and unbalanced feeders
- Can generate MI reports
- Provide data for maintenance and augmentation plan

### 7.4 Data Collection

Several formats have to be introduced for data colelction which make easier to input data to the developed system. In the existing manual system there is no such formats for collecting data and it was noticed that there were many inaccuaracies in recorded data. The formats were designed so that the recorder will record the data with minimal effort.

### 7.5 Data Analysis

Data shoul be validated for standard data and other types of data. It could be evaluated for known cases. Before entering the data sepcially system related data should be properly analysed for any incompatibilities.

### 7.4 Summary

This chapter presented the evalaution of the FARMS. It includes testing of the software system, test methods, features to be tested etc.

--

# Chapter 8

# Supply Failure Analysis and Forecasting

## 8.1Introduction

Development and implementation of FAMRS software have been discussed in previous chapters. Software has been tested for data validation, transaction and calculation of various parameters like substation loading, reliability indices etc. Existing data recorded in the breakdown registers of Homagama CSC have been entered for the analysis and generating MI reports. Output of the software based on the entered data has been discussed in following sections.

## 8.2 Detail breakdown analysis

Reports have been generated from the software to identify the category of breakdown due to various reasons during the time period from January 2015 to September 2015. The results obtained for high voltage, low voltage and single consumer supply failure have been depicted in table 8.1 to 8.3 while figure 8.1 to figure 8.4 depicts monthly breakdown variation for high voltage, low voltage and single consumers.

Table 8.1: High voltage breakdown from 1st January to 30th September 2015

| **Breakdown Reason** | **Jan.** | **Feb.** | **March** | **April** | **May** | **June** | **July** | **August** | **Sept.** |
|---|---|---|---|---|---|---|---|---|---|
| Vegetation | 0 | 0 | 0 | 0 | 1 | 2 | 2 | 0 | 0 |
| Branches Coming from Distance | 0 | 0 | 0 | 0 | 3 | 1 | 3 | 3 | 2 |
| Burnt Jumpers, Conductors Due To Loose Connections | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Loose Span and Entanglement | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ABC Tripping | 0 | 0 | 0 | 0 | 0 | 11 | 1 | 4 | 0 |
| Due To Broken Poles | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| Burnt Tail Wires | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Fuse Blown | 1 | 0 | 1 | 1 | 2 | 5 | 6 | 10 | 1 |
| Wires Broken | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Service Wire Burnt | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Bad Weather | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Others | 0 | 0 | 0 | 0 | 1 | 2 | 0 | 1 | 0 |
| **Total** | **2** | **0** | **1** | **1** | **9** | **21** | **12** | **20** | **4** |

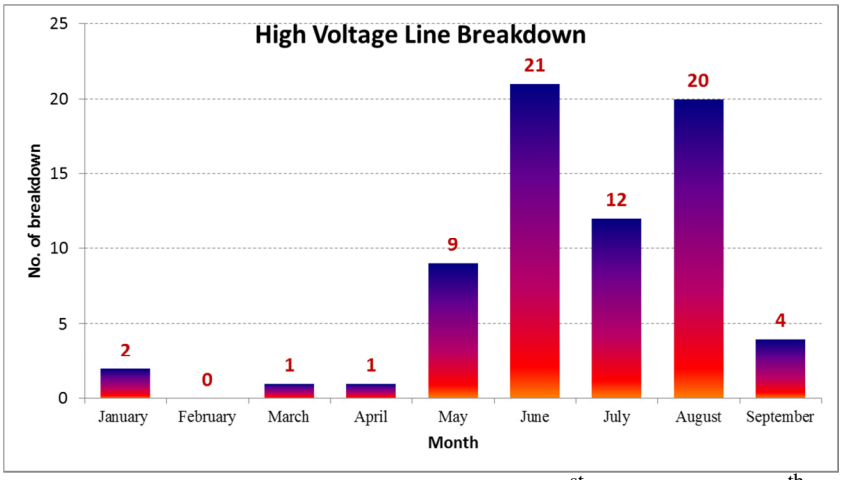



Figure 8.1 High voltage breakdowns for the period 1st Jan. 2015 to 30th Sept. 2015

Table 8.2: Low voltage breakdown for the period 1st January to 30th September 2015

| **Breakdown reasons** | **Janu.** | **Feb.** | **March** | **April** | **May** | **June** | **July** | **August** | **Sept.** |
|---|---|---|---|---|---|---|---|---|---|
| Vegetation | 12 | 17 | 3 | 11 | 5 | 6 | 2 | 0 | 0 |
| Branches Coming From Distance | 17 | 7 | 2 | 11 | 19 | 33 | 27 | 11 | 21 |
| `Burnt Jumpers & Conductors Due To Loose Connections | 13 | 4 | 14 | 33 | 52 | 49 | 29 | 41 | 27 |
| Loose Span And Entanglement | 5 | 0 | 5 | 0 | 3 | 6 | 3 | 11 | 1 |
| ABC Tripping Fuses Blown | 1 | 2 | 6 | 16 | 71 | 66 | 43 | 31 | 49 |
| Transformer Failures | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Consumer Faults | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Due To Broken Poles | 0 | 0 | 0 | 0 | 5 | 1 | 4 | 0 | 1 |
| Burnt Tail Wires | 5 | 3 | 2 | 0 | 0 | 2 | 1 | 0 | 1 |
| Aging of Fuses | 6 | 4 | 4 | 34 | 14 | 15 | 8 | 4 | 0 |
| Loose Connection at Pole | 24 | 19 | 24 | 83 | 195 | 1 | 10 | 1 | 12 |
| Damage/Defect Service Wires | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Formation of Oxide | 15 | 9 | 3 | 7 | 2 | 1 | 0 | 1 | 0 |
| Due to Animals and Birds | 85 | 67 | 38 | 37 | 30 | 33 | 14 | 8 | 4 |
| Fuse Blown | 2 | 5 | 3 | 5 | 2 | 0 | 0 | 3 | 5 |
| Wires Broken | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Service Wire Burnt | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Bad Weather | 1 | 2 | 1 | 3 | 2 | 1 | 0 | 0 | 1 |
| Sabotage | 1 | 0 | 0 | 0 | 1 | 2 | 0 | 0 | 0 |
| Accident due to vehicles | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Others | 10 | 5 | 8 | 17 | 17 | 8 | 1 | 6 | |
| **Total** | **197** | **146** | **113** | **257** | **419** | **224** | **142** | **117** | **123** |

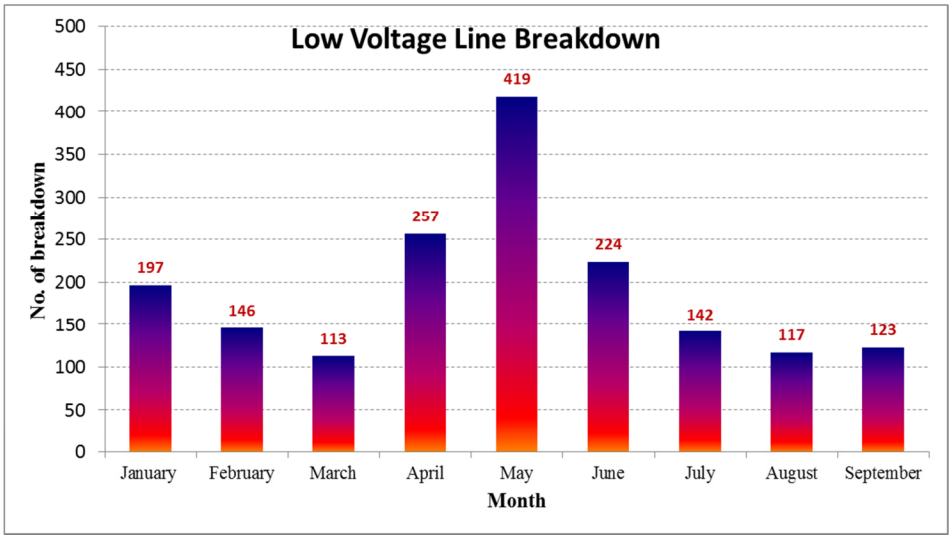



Figure 8.2 Low voltage breakdown for the period 1st January to 30th September 2015

Table 8.3: Single consumer breakdown for the period 1st January to 30th September 2015

| **Reasons for breakdown** | **Jan.** | **Feb.** | **March** | **April** | **May** | **June** | **July** | **August** | **Sept.** |
|---|---|---|---|---|---|---|---|---|---|
| Vegetation | 4 | 0 | 1 | 7 | 2 | 3 | 2 | 0 | 0 |
| Branches Coming From Distance | 7 | 3 | 4 | 5 | 17 | 24 | 12 | 9 | 1 |
| Consumer Faults | 28 | 26 | 22 | 78 | 127 | 63 | 55 | 47 | 50 |
| Due To Broken Poles | 0 | 1 | 0 | 0 | 0 | 2 | 0 | 0 | 0 |
| Burnt Tail Wires And Cables | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Service Wire Broken | 15 | 12 | 25 | 71 | 96 | 69 | 90 | 90 | 78 |
| Meter Burn | 6 | 5 | 3 | 28 | 27 | 12 | 4 | 1 | 10 |
| Meter Terminal Burnt | 6 | 12 | 9 | 12 | 19 | 16 | 17 | 7 | 21 |
| Fuse Cutout Burnt | 20 | 15 | 12 | 51 | 61 | 34 | 33 | 40 | 42 |
| Fuse Cutout Terminal Burnt | 30 | 23 | 14 | 35 | 43 | 25 | 11 | 18 | 23 |
| Loose Connection at Meter | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Loose Connection at Cutouts | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Loose Connection at Pole | 8 | 13 | 8 | 8 | 24 | 154 | 142 | 79 | 109 |
| Damage/Defective Service Wires | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Formation of Oxide | 4 | 6 | 1 | 8 | 9 | 14 | 8 | 4 | 21 |
| Sabotage | 0 | 1 | 0 | 0 | 1 | 3 | 1 | 0 | 1 |
| Bad Weather | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| Others | 5 | 5 | 5 | 20 | 34 | 23 | 27 | 30 | 38 |
| **Total** | **133** | **122** | **104** | **323** | **460** | **443** | **402** | **325** | **395** |

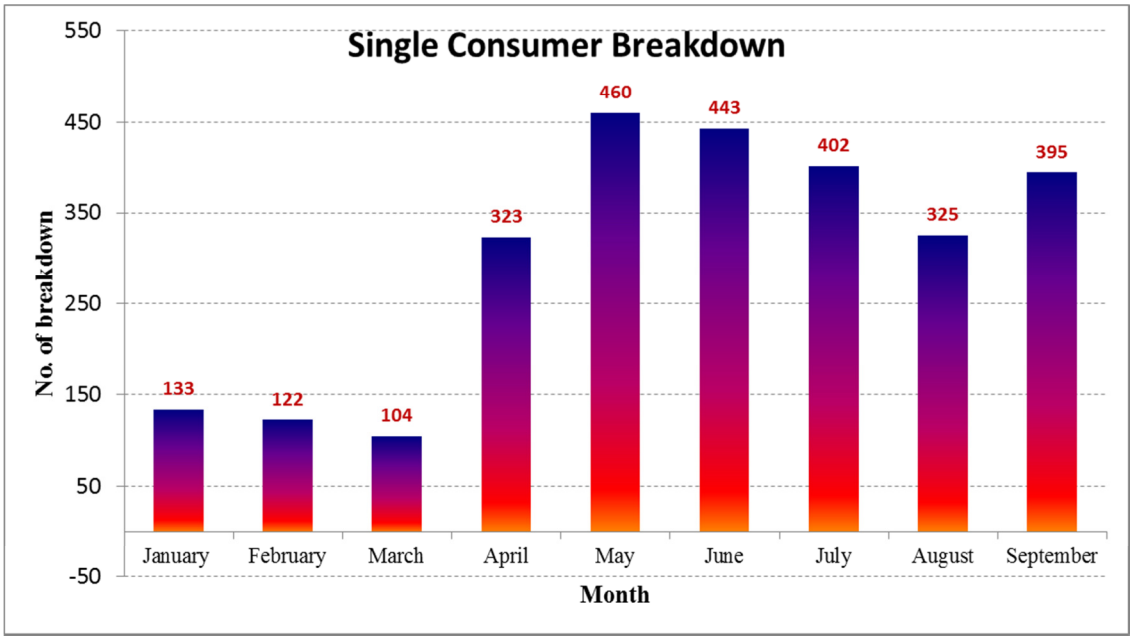



Figure 8.3: Single Consumer breakdown for the period 1st Janu. to 30th Sept. 015

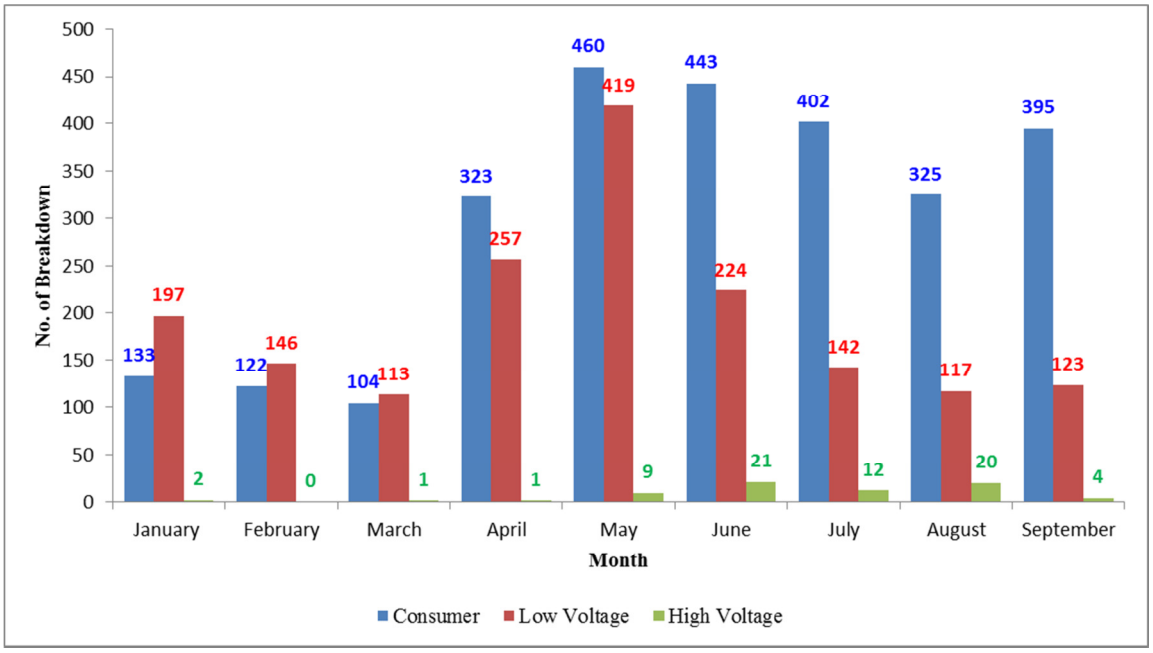



Figure 8.4: Breakdown in different categories for the period 1st Janu. to 30th Sept. 2015

The result shows that most of the high voltage breakdown has taken place during the month of May and June and most of breakdown has occurred due to fuse blown and vegetation. Thus one can analyze in detail the root cause for fusing and implement a plan for removing vegetation of the lines. This information is very vital and it could not be obtained from the hardcopy based breakdown record system. Therefore, this analysis provides information for maintenance activity schedule. Further, number of high voltage breakdown should be minimized since one failure affects large number of consumers. Thus, rectification by identifying the root cause with further analysis for the breakdown in this manner, leads to improve the supply reliability.

Further it can be seen from Fig. 8.4 that no. of low voltage breakdown is very much higher than that of high voltage breakdowns. Most of the breakdowns have occurred due to vegetation, loose jumper connection and loose connection at pole. Thus, reason for fuse blowing needs to be investigated. It could be due to aging of fuses, bad quality of fuses, wrong sizing of fuses. Therefore, further investigation could be carried out to find root cause to supply failures there by reducing the no. of low voltage breakdowns. In low voltage supply breakdown, affected no. of consumers is less compared to high voltage breakdown.

No. of single consumer breakdowns are very much higher than that of high voltage and low voltage breakdowns. However, all three cases, breakdowns are more during May, June July. Thus one can analyze further, the reason for the increase. According to table 8.3 and Fig. 8.3, most of the single consumer breakdown has occurred due to consumer faults, service wire broken down and loose connection at pole. Detail analysis shows that most of the consumes complaint for supply failures due to their internal faults like operating trip switch due to earth leakage in house. Thus when a complaint made, the call receiver can educate consumer to see whether trip switch has been operated before making any complain.

It could be extracted month of highest no. of breakdown for different categories, high voltage, low voltage and single consumer. Figure 8.5 to 8.7 shows that highest no. of high

voltage breakdown that has occurred in month of June while the same for low voltage and single consumer is the month of May. Now, further analysis could be done to get the information on the reason that has led to highest breakdown. For high voltage the reason is fuse blown while for low voltage and single consumer breakdown reasons are loose connections at poles. Thus we can narrow down to one or two reasons for devising a migratory activity.

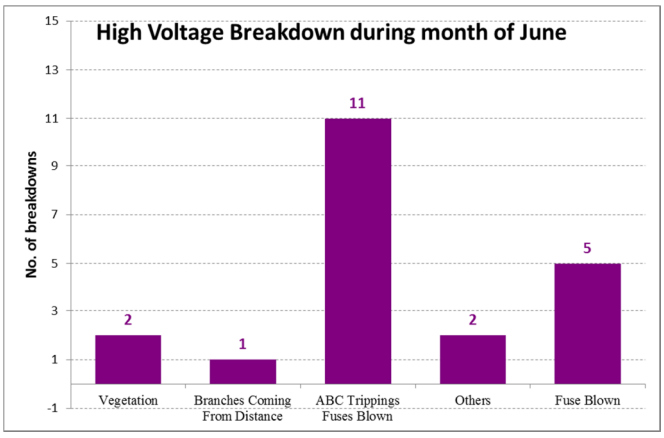



Figure 8.5: Reason wise high voltage breakdown during month of June

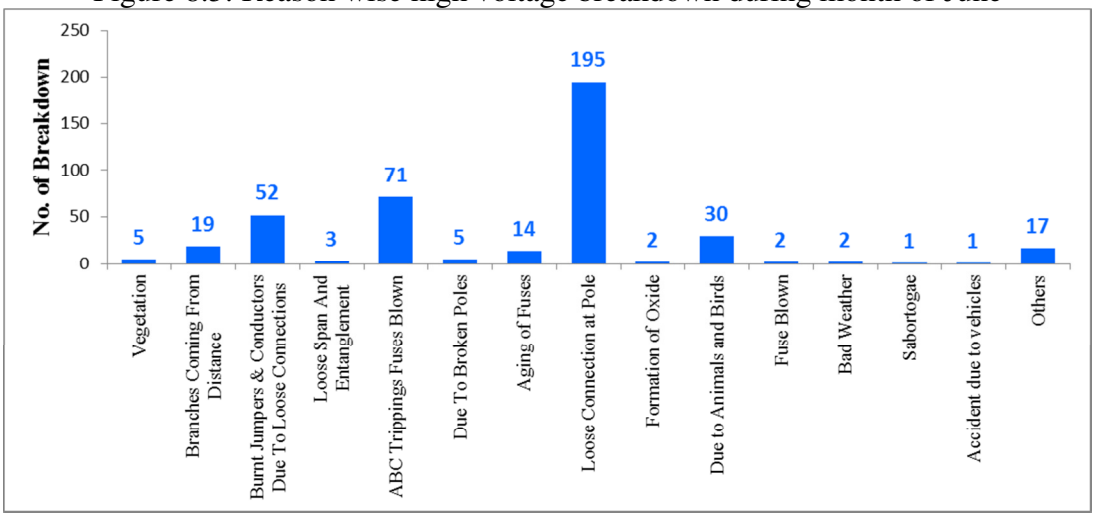



Figure 8.6: Reason wise low voltage breakdown during month of May

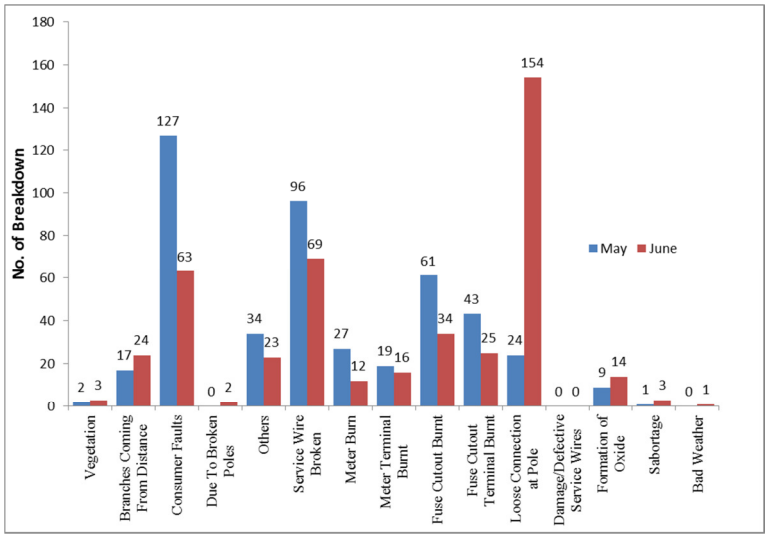



Figure 8.7: Reason wise low voltage breakdown during month of May

### 8.3 Substation wise breakdown analysis

Report could be generated to analyze substation which has highest no. of breakdown during the given period of time. Table 8.4 depicts the no. of breakdown in a substation for the period 1st January 2015 to 30th September 2015 sorted for the substation with highest breakdown substation while Fig. 8.8 shows 10 substations with highest recoded breakdowns. It is clear from table 8.4 and Fig. 8.8 that Sarasavi substation has 35 no. of breakdown during period specified and it is followed by Dolahena substation. Therefore, this substation could be considered as most vulnerable substation for more breakdowns in future and there is a chance that transformers in this substation could be burnt out. Hence, detail investigation has to be done and analysis could be carried out to rectify the problems so that breakdown in future could be minimized. Thus this type of information is vital for planning maintenance activities and improves supply reliability. Further, extracting such information from hardcopy based manual system is not possible since it may need to have large effort.

Further, table 8.5 and Fig. 8.9 depicts the sections in the network which have highest number of breakdown. According to the information in the table 8.4 and Fig. 8.9, From Pannipitiya GSS to Hathlahogoda DDLO section of high voltage line has 28 no. of breakdown during the given period. Therefore, proper attention should be given to this section with top priority for carrying out maintenance activities.

Table 8.4: Top most breakdown substation during the period 1st Janu. to 30th Sept. 2015

| SIN | Name of Sub | No. of Breakdown |
|---|---|---|
| 41/H004 | Sarasavi | 35 |
| 41/H088 | Dolahena | 33 |
| 41/H015 | Unapandura | 28 |
| 41/H024 | Maithredasa MW | 26 |
| 41/H068 | Maha Katuwana | 26 |
| 41/H009 | Walawwa | 26 |
| 41/H055 | Uduwana | 26 |
| 41/H053 | Magammana | 24 |
| 41/H051 | Niyandagala | 24 |
| 41/H095 | Kajugahayatadeniya | 24 |
| 41/H007 | Galawilawatta | 23 |
| 41/H032 | Welipillewa I | 22 |
| 41/H126 | Market Sub | 21 |
| 41/H079 | Sothosa Sub | 21 |
| 41/H047 | Katuwana | 17 |
| 41/H018 | Town Sub - Homagama | 17 |
| 41/H089 | Suwapubudugama | 16 |
| 41/H062 | Brahmanagama | 15 |
| 41/H052 | Diyagama | 14 |
| 41/H117 | Kandanawatta | 13 |
| 41/H023 | Gabadawatta I | 13 |

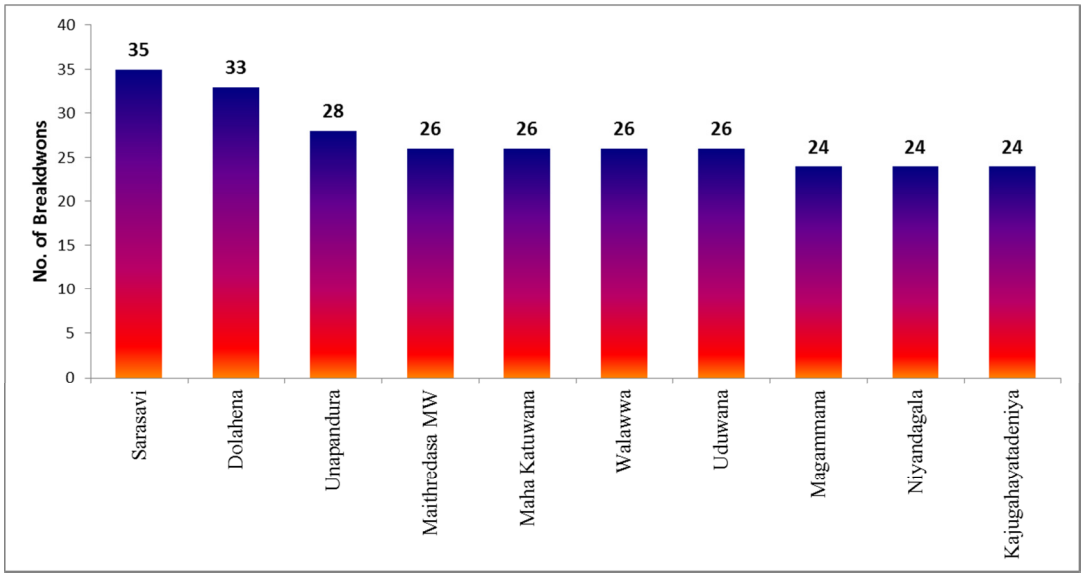



Figure 8.8: 10 substations that have highest no. of breakdowns

Table 8.5: Top most breakdown sections

| Section ID | Section Name | No of Breakdown |
|---|---|---|
| PP/F6/S402 | Hathlahagoda D.D.L.O. | 28 |
| PP/F6/S204 | Mawathgama D.D.L.O. | 16 |
| PP/F6/S403 | Diyagama D.D.L.O. | 12 |
| PP/F6/S202 | Nuwan D.D.L.O | 12 |
| PP/F6/S401 | Deepangoda D.D.L.O. | 12 |
| PP/F6/S1L1 | C.T.B Section | 6 |
| PP/F6/S201 | Walawwa D.D.L.O to Malapalla | 5 |
| PP/F6/S404 | Pelendagoda D.D.L.O. | 4 |
| PP/F6/S302 | A.B.S. Panagoda | 2 |
| PP/F6/S1R1 | Kirigampamunuwa Temple | 2 |
| KG/F2/HOMA | Welipillawa Section | 1 |
| PP/F6/S001 | Gantry Section to CTB ABS | 1 |

### 8.4 Foresting of Supply Failures

Based on the supply failure analysis discussed above, it could be used for forecasting of HT breakdown, LT breakdwon, single consumer breakdwon, vulnarable area of electricity network, supply failure in reason wise. Further, materials used for supply failure restorations could be forecatsed and it will be very useful for material planning. However, in order to make such forecasts, it is required to enter 3 to 4 years supply failure data to the developed software system.

### 8.5 Summary

This chapter presented supply failure analysis using the output of the FARMS. It was highlighted that software could be utilized for maintenance planning, material forecasting, and identification of network vulnerable areas.

--

# Chapter 9

# Conclusion

## 9.1 Introduction

In the previous chapter it was discussed about the results obtained from database using the developed software, FARMS. In this chapter, a conclusion will be given based on the achievement, features and other related information. Finally a recomemnadtion will be given for further work.

## 9.2 Conclusion

Electricity is one of the essential needs of people at present. Most of the day to day activities of the people are linked with electricity. Power supply failure to very short duration is not desirable by the people. Therefore, it is essential to maintain power supply failures to very minimum level thereby improving power supply reliability.

In order to improve power supply reliability and maintain the reliability level at specified level, it needs comprehensive monitoring and analysis of power supply failures in the electricity network including individual consumer premise. This could be achieved through power failure analysis and monitoring the power supply reliability.

The main organization for most of power generation, transmission and most of power distribution in Sri Lanka is Ceylon Electricity Board (CEB). However, CEB is still adopting manual system for power failure analysis. There are several drawbacks in the manual system such as lack of information about the past and current power supply failures, process is handlled by ad-hoc method, information flow is difficult to identify, longer time for breadown record searching and processing, it has to maintain registers for breakdown recording and reporting, report preperation is difficult and time consuming, no proper identification of electricity network for immediate maintenance, diffucult to trace temporary and permanently completed breakdown.

In order to rectify the issues in the manual failure analysis system, a database system has been designed and developed to handle failure data up to the power failure restoration. Microsoft SQL Server 2008 was used as back end in Failure Analysis and Reliability Monitoring System (FARMS). Back end is the software that will store whatever the data used by front end. Database connection time is very low and that will support to fast data accessibility between front end and back end. MS SQL server also has good features to manipulate data in a database. Data insertion, updating, data navigation and deletion methods are very fast.

The main features of the developed system are ability to handle supply failure up to restoration, identify the weakest area in the power supply system, identify the level of power supply reliability in two voltages, high voltage and low voltage, monitor nature of failure and cause of failure, estimate reliability indices for each zone, indicates measures to be taken, provide substation loading and unbalanced feeders and could generate management information report easilly.

The developed system was tested using with actual data collected from CSC of CEB. It was noted that system achieved deired performnces. The accuracy of the system depends fully on the accuracy of the collected data. Thus feeding accurate data to system is a great challenge in this process. Further there could be some system limitation when the size of the databse is becoming large. Thus the database system should be updated as and when required.

### 9.3 Future work

Comunication is being developed to very high level. Many devices in the power system like generators, transformers, switches, meters come with communication attached. Thus operation of these devices from remote location could be easily done. Further, most of the household applicances could be on or off attaching sensing devices with comunication faclities. Thus management of house load in future coul be done using a simple mobile application. Thefore incoprating these features into developed software may have to be done in future.

## 9.4 Summary

This chapter prseented a conclusion of the research carried out in this thesis. It was highlighted that developed system could perform better for the analysis of supply failure data and monitoring supply reliability. Adoting smart grid technology with the communication developemnt to proposed system could be future challenge.

--

## References


[1] Ceylon Electricity Board, "Statistical Digest 2015", pp 2

[2] Roy Billinton and Ronald N. Allan, "Reliability Evaluation of Engineering Systems: Concepts and Techniques", Springer (first published January 1st 1983), Second Edition, 1992.

[3] Charles R. Heising, "Reliability and Availability Comparison of common Low-Voltage Industrial Power Distribution Systems", IEEE Transaction on Industry and General Applications, Vol. IGA-6. No. 5, September/October 1970, pp 416-424.

[4] J. ALACCHI, "Reliability Considerations in Cement Plant Power Distribution, IEEE Transaction on Industry Applications, Vol. IA-15, No. 2, March/April 1979.

[5] E. N. Dialynas, R. N. Allan, "Local Generating Facilities in the Reliability Evaluation of Power Distribution Systems", 86 WM 058-2, November 1986, pp.62-67

[6] R. C. Settembrini , J. R. Fisher, N. E. Hudak, "Reliability and Quality Comparison of Electric Power Distribution Systems", IEEE transaction on Power System, Vol. 1, pp – 704-712

[7] Ron Allan and Roy Billinton, "Power system reliability and its assessment: Part 3 Distribution systems and economic considerations", IEEE Power Engineering Journal, pp 185-192

[8] Yifan Tang, "Power Distribution System Planning with Reliability Modeling and Optimization, IEEE Transactions on Power Systems, Vol. 11, No. 1, February 1996, pp -181-189

[9] T. Lantharthong, N. Phanthuna, "Techniques for Reliability Evaluation in Distribution System Planning", International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering Vol:6, No:4, 2012

[10] C.Bhargava1, P.S.R Murty2, P.Sreevani3, "Reliability Assessment of Radial distribution system incorporating weather effects", International Journal of Engineering Research and Development, Volume 6, Issue 7, April 2013, PP. 45-51)

[11] Kehinde Olusuyi, Ayodele Sunday Oluwole, Temitope Adefarati, Adedayo

Kayode Babarinde, “A fault analysis of 11kv distribution system (a case study of ado Ekiti electrical power distribution district), American Journal of Electrical Power and Energy Systems 2014, vol. 3, pp 27-36.

[12] J. Manikya Rao, P.V.N. Prasad2 and G. Tulasi Ram Das, “Customer Outage Cost Evaluation in Electric Power System”, Vol. 5, August 201, pp. 88- 96.

[13] Salem Elsaiah, Mohammed Benidris and Joydeep Mitra, “Reliability Improvement of Power Distribution System Through Feeder Reconfiguration”, IEEE transaction PMAPS, Vol. 1, pp 978-984, 2014.

[14] Salman Kahrobaee, Sohrab Asgarpoor, “Reliability Assessment for Smart Grid and Future Power Distribution Systems”, IEEE Conference on Technologies for Sustainability (Sus tech), pp 160-164

[15] Ceylon Electricity Board, “ Standard Construction Rate 2016”, a CEB internal publication

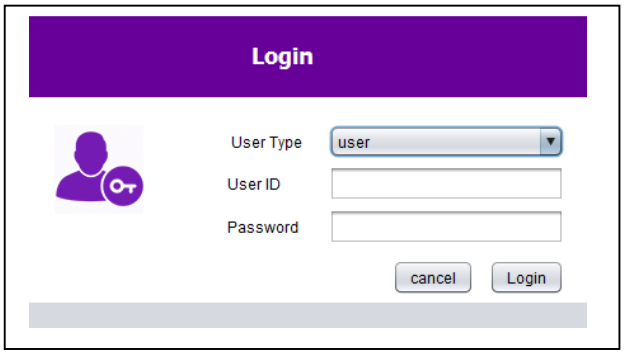
[16] Ceylon Electricity Board, “Breakdown Reregister for Consumer Service Center”, Homagama Consumer Service Center

[17] Stewart Venit, “Introduction to Programming: Concepts and Design”, Scott Jones, California, USA

[18] B. A. Forouzan, “ Data Communications and Networking”, 4th edition, Mc Graw Hill Education India Pvt Ltd.

## Appendix I: Software Coding

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package ceb;

import static com.sun.corba.se.spi.presentation.rmi.StubAdapter.request;
import connection.DB;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.JOptionPane;
import pages.userAccount;


public class login extends javax.swing.JFrame {

    /**
     * Creates new form login
     */
    public login() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        jPanel2 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        txtuserid = new javax.swing.JTextField();
        jLabel4 = new javax.swing.JLabel();
        txtpw = new javax.swing.JPasswordField();
        jButton1 = new javax.swing.JButton();
        jLabel6 = new javax.swing.JLabel();
        jComboBox1 = new javax.swing.JComboBox();
        jButton2 = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setUndecorated(true);

        jPanel1.setBackground(new java.awt.Color(255, 255, 255));
```



```
        jPanel2.setBackground(new java.awt.Color(102, 0, 153));

        jLabel1.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
        jLabel1.setForeground(new java.awt.Color(255, 255, 255));
        jLabel1.setText("Login");

        javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(175, 175, 175)
                .addComponent(jLabel1)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );
        jPanel2Layout.setVerticalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(20, 20, 20)
                .addComponent(jLabel1)
                .addContainerGap(22, Short.MAX_VALUE))
        );

        jLabel2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/icon/users.png"))); // NOI18N

        jLabel3.setText("User ID");

        jLabel4.setText("Password");

        txtpw.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                txtpwActionPerformed(evt);
            }
        });

        jButton1.setText("Login");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        jLabel6.setText("User Type");

        jComboBox1.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "user", "admin",
"super admin" }));

        jButton2.setText("cancel");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
            jButton2ActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
    jPanel1.setLayout(jPanel1Layout);
    jPanel1Layout.setHorizontalGroup(
        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jButton2)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                    .addComponent(jButton1))
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addGap(20, 20, 20)
                    .addComponent(jLabel2)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 66,
Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(jLabel3)
                            .addComponent(jLabel4))
                        .addGap(24, 24, 24))
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
                        .addComponent(jLabel6)
                        .addGap(21, 21, 21)))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                    .addComponent(jComboBox1, 0, 185, Short.MAX_VALUE)
                    .addComponent(txtuserid)
                    .addComponent(txtpw))))
            .addContainerGap(18, Short.MAX_VALUE))
    );
    jPanel1Layout.setVerticalGroup(
        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 22,
Short.MAX_VALUE)
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel1Layout.createSequentialGroup()
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel6)
                    .addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel3)
                    .addComponent(txtuserid, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel4)
                    .addComponent(txtpw, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
              .addComponent(jLabel2))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
              .addComponent(jButton1)
              .addComponent(jButton2))
            .addContainerGap())
      );

      javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
      getContentPane().setLayout(layout);
      layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jPanel1,                                    javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
      );
      layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
          .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
          .addGap(20, 20, 20))
      );

      pack();
      setLocationRelativeTo(null);
    }// </editor-fold>

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
      String type = jComboBox1.getSelectedItem().toString();

      if (type == "user") {
        uderlogin();
      }
      if (type == "admin") {
```

```
            uderlogin();
        }
    }

    private void txtpwActionPerformed(java.awt.event.ActionEvent evt) {
        String type = jComboBox1.getSelectedItem().toString();

        if (type == "user") {
            uderlogin();
        }
        if (type == "admin") {
            uderlogin();
        }
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
      this.dispose();
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(login.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(login.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(login.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(login.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
        }
        //</editor-fold>
```

```
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new login().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    public static javax.swing.JButton jButton1;
    public static javax.swing.JButton jButton2;
    public static javax.swing.JComboBox jComboBox1;
    public static javax.swing.JLabel jLabel1;
    public static javax.swing.JLabel jLabel2;
    public static javax.swing.JLabel jLabel3;
    public static javax.swing.JLabel jLabel4;
    public static javax.swing.JLabel jLabel6;
    public static javax.swing.JPanel jPanel1;
    public static javax.swing.JPanel jPanel2;
    public static javax.swing.JPasswordField txtpw;
    public static javax.swing.JTextField txtuserid;
    // End of variables declaration

    void uderlogin() {
        String userid = txtuserid.getText();
        String pw = new String(txtpw.getPassword());

        try {
            Statement s = DB.getcon().createStatement();
            ResultSet rs = s.executeQuery("SELECT * FROM `user` WHERE `user_id`='" + userid + "' and
`password`='" + pw + "'");
            if (rs.next()) {
                this.dispose();
                home h = new home();
                h.jLabel7.setText(rs.getString("user_id"));
                String x = rs.getString("user_id");

                h.setVisible(true);
            } else {
                JOptionPane.showMessageDialog(this, "password and user id not matching");
            }
        } catch (Exception e) {
            e.printStackTrace();
        }

    }
}
```

CEB.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package ceb;

/**
 *
 * @author MyLaptop
 */
public class CEB {

   /**
    * @param args the command line arguments
    */
   public static void main(String[] args) {
      // TODO code application logic here
   }

}
```

UserDetails.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package ceb;

/**
 *
 * @author MyLaptop
 */
public class userDetails {
   public userDetails(String userid) {

      }
}
```

Connection

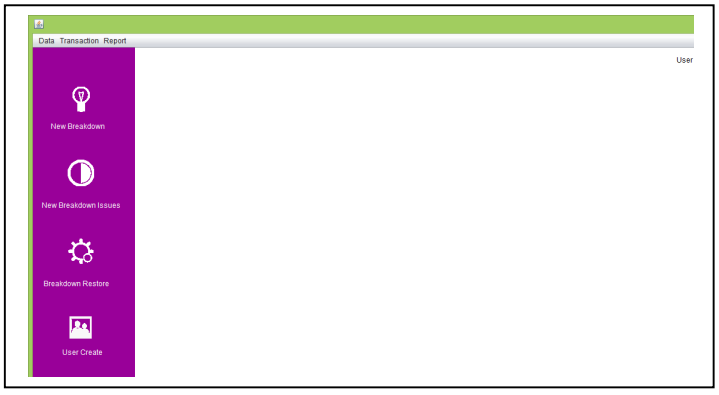
DB.Java

```
/*
```

```
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package connection;

import java.sql.Connection;
import java.sql.DriverManager;

/**
 *
 * @author MyLaptop
 */
public class DB {
   static Connection c;

   public static Connection getcon() throws Exception {
      if (c == null) {
         Class.forName("com.mysql.jdbc.Driver");
         c = DriverManager.getConnection("jdbc:mysql://localhost/ceb", "root", "");
      }
      return c;
   }
}
```

---

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package ceb;

import Report.time_report;
import Report.top_most_feeder;
import Report.top_most_grid;
import Report.top_most_section;
import Report.top_most_sin;
import connection.DB;
import java.sql.ResultSet;
import java.sql.Statement;
import pages.CSC;
import pages.add_item;
import pages.areas;
import pages.breakdownIsuarUpdate;
import pages.breakdown_issues;
import pages.breakdown_reason;
import pages.breakdown_type;
import pages.create_resons;
```



```
import pages.distribution_substation;
import pages.feeder_details;
import pages.grid_details;
import pages.kva_types;
import pages.load_reading_data;
import pages.lvlineDetails;
import pages.newbreakdown_details;
import pages.provinces;
import pages.region;
import pages.section_details;
import pages.update_breakdown;
import pages.user;
import pages.userAccount;
import pages.zone_details;

/**
 *
 * @author MyLaptop
 */
public class home extends javax.swing.JFrame {

    /**
     * Creates new form home
     */
    public home() {
        initComponents();

    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jDesktopPane1 = new javax.swing.JDesktopPane();
        jPanel1 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jButton6 = new javax.swing.JButton();
        jLabel7 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();
        jButton7 = new javax.swing.JButton();
        jPanel2 = new javax.swing.JPanel();
        jButton1 = new javax.swing.JButton();
        jLabel3 = new javax.swing.JLabel();
        jButton2 = new javax.swing.JButton();
        jLabel4 = new javax.swing.JLabel();
```

```
jLabel5 = new javax.swing.JLabel();
jButton3 = new javax.swing.JButton();
jButton4 = new javax.swing.JButton();
jLabel6 = new javax.swing.JLabel();
jButton5 = new javax.swing.JButton();
jMenuBar1 = new javax.swing.JMenuBar();
jMenu1 = new javax.swing.JMenu();
jMenu5 = new javax.swing.JMenu();
jMenuItem4 = new javax.swing.JMenuItem();
jMenuItem5 = new javax.swing.JMenuItem();
jMenuItem6 = new javax.swing.JMenuItem();
jMenuItem7 = new javax.swing.JMenuItem();
jMenuItem8 = new javax.swing.JMenuItem();
jMenuItem20 = new javax.swing.JMenuItem();
jMenu6 = new javax.swing.JMenu();
jMenuItem1 = new javax.swing.JMenuItem();
jMenuItem23 = new javax.swing.JMenuItem();
jMenuItem3 = new javax.swing.JMenuItem();
jMenuItem9 = new javax.swing.JMenuItem();
jMenuItem10 = new javax.swing.JMenuItem();
jMenuItem11 = new javax.swing.JMenuItem();
jMenuItem12 = new javax.swing.JMenuItem();
jMenuItem17 = new javax.swing.JMenuItem();
jMenuItem18 = new javax.swing.JMenuItem();
jMenu2 = new javax.swing.JMenu();
jMenu7 = new javax.swing.JMenu();
jMenuItem13 = new javax.swing.JMenuItem();
jMenuItem14 = new javax.swing.JMenuItem();
jMenu8 = new javax.swing.JMenu();
jMenuItem15 = new javax.swing.JMenuItem();
jMenuItem16 = new javax.swing.JMenuItem();
jMenuItem21 = new javax.swing.JMenuItem();
jMenuItem27 = new javax.swing.JMenuItem();
jMenu3 = new javax.swing.JMenu();
jMenuItem19 = new javax.swing.JMenuItem();
jMenuItem24 = new javax.swing.JMenuItem();
jMenuItem25 = new javax.swing.JMenuItem();
jMenuItem26 = new javax.swing.JMenuItem();
jMenuItem22 = new javax.swing.JMenuItem();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
setMinimumSize(null);

jDesktopPane1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jPanel1.setBackground(new java.awt.Color(255, 255, 255));

jLabel1.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
jLabel1.setForeground(new java.awt.Color(51, 51, 51));
jLabel1.setText("Ceylon Electricity Board");
```

```
        jLabel2.setFont(new java.awt.Font("Tahoma", 1, 24)); // NOI18N
        jLabel2.setForeground(new java.awt.Color(102, 102, 102));
        jLabel2.setText("Faults Analysis and Reliability Monitoring System");

        jButton6.setText("My Account");
        jButton6.setContentAreaFilled(false);
        jButton6.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton6ActionPerformed(evt);
            }
        });

        jLabel7.setText("jLabel7");

        jLabel8.setText("User ID");

        jButton7.setText("Log out");
        jButton7.setContentAreaFilled(false);
        jButton7.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton7ActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap(604, Short.MAX_VALUE)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
                        .addComponent(jLabel8)
                        .addGap(32, 32, 32)
                        .addComponent(jLabel7)
                        .addGap(45, 45, 45)
                        .addComponent(jButton6)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jButton7)
                        .addGap(50, 50, 50))
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(jLabel2)
                            .addComponent(jLabel1))
                        .addGap(48, 48, 48))))
        );
        jPanel1Layout.setVerticalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jButton6)
                    .addComponent(jLabel7)
                    .addComponent(jLabel8)
                    .addComponent(jButton7))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 570,
Short.MAX_VALUE)
                .addComponent(jLabel2)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel1)
                .addGap(19, 19, 19))
        );

        jDesktopPane1.add(jPanel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(170, 0, 1240, 680));

        jPanel2.setBackground(new java.awt.Color(153, 0, 153));

        jButton1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/icon/new_breakdown.png")));
// NOI18N
        jButton1.setContentAreaFilled(false);
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        jLabel3.setForeground(new java.awt.Color(255, 255, 255));
        jLabel3.setText("New Breakdown");

        jButton2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/icon/new-breakdown-
issur.png"))); // NOI18N
        jButton2.setContentAreaFilled(false);
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });

        jLabel4.setForeground(new java.awt.Color(255, 255, 255));
        jLabel4.setText("New Breakdown Issues");

        jLabel5.setForeground(new java.awt.Color(255, 255, 255));
        jLabel5.setText(" Breakdown Restore");

        jButton3.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icon/breakdown_reson.png"))); // NOI18N
        jButton3.setContentAreaFilled(false);
        jButton3.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
            jButton3ActionPerformed(evt);
        }
    });

    jButton4.setIcon(new javax.swing.ImageIcon(getClass().getResource("/icon/user.png"))); // NOI18N
    jButton4.setContentAreaFilled(false);
    jButton4.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton4ActionPerformed(evt);
        }
    });

    jLabel6.setForeground(new java.awt.Color(255, 255, 255));
    jLabel6.setText("User Create");

    jButton5.setIcon(new javax.swing.ImageIcon(getClass().getResource("/icon/logout.png"))); // NOI18N
    jButton5.setContentAreaFilled(false);
    jButton5.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton5ActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
    jPanel2.setLayout(jPanel2Layout);
    jPanel2Layout.setHorizontalGroup(
        jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel2Layout.createSequentialGroup()
            .addContainerGap(15, Short.MAX_VALUE)
            .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel5, javax.swing.GroupLayout.DEFAULT_SIZE, 155, Short.MAX_VALUE)
                .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
                        .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 100,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(31, 31, 31))
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
                        .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 140,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addContainerGap()))))
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(50, 50, 50)
            .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jButton3, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(49, 49, 49)
            .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 72,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jButton4, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jButton5, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(0, 0, Short.MAX_VALUE))
    );
    jPanel2Layout.setVerticalGroup(
        jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(60, 60, 60)
            .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 50,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(jLabel3)
            .addGap(45, 45, 45)
            .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 50,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(42, 42, 42)
            .addComponent(jButton3, javax.swing.GroupLayout.PREFERRED_SIZE, 50,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(38, 38, 38)
            .addComponent(jButton4, javax.swing.GroupLayout.PREFERRED_SIZE, 50,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 50,
Short.MAX_VALUE)
            .addComponent(jButton5, javax.swing.GroupLayout.PREFERRED_SIZE, 50,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(68, 68, 68))
    );

    jDesktopPane1.add(jPanel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 170, 680));

    jMenuBar1.setBackground(new java.awt.Color(153, 0, 153));
    jMenuBar1.setForeground(new java.awt.Color(255, 255, 255));

    jMenu1.setText("Data");
```

```
        jMenu1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenu1ActionPerformed(evt);
            }
        });

        jMenu5.setText("Master Details");
        jMenu5.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenu5ActionPerformed(evt);
            }
        });

        jMenuItem4.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_R,
java.awt.event.InputEvent.ALT_MASK));
        jMenuItem4.setText("Region");
        jMenuItem4.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenuItem4ActionPerformed(evt);
            }
        });
        jMenu5.add(jMenuItem4);

        jMenuItem5.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_P,
java.awt.event.InputEvent.ALT_MASK));
        jMenuItem5.setText("Provinces");
        jMenuItem5.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenuItem5ActionPerformed(evt);
            }
        });
        jMenu5.add(jMenuItem5);

        jMenuItem6.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_A,
java.awt.event.InputEvent.ALT_MASK));
        jMenuItem6.setText("Areas");
        jMenuItem6.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenuItem6ActionPerformed(evt);
            }
        });
        jMenu5.add(jMenuItem6);

        jMenuItem7.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_C,
java.awt.event.InputEvent.ALT_MASK));
        jMenuItem7.setText("CSC");
        jMenuItem7.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenuItem7ActionPerformed(evt);
            }
        });
```

```
        jMenu5.add(jMenuItem7);

        jMenuItem8.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_U,
java.awt.event.InputEvent.ALT_MASK));
        jMenuItem8.setText("Users");
        jMenuItem8.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenuItem8ActionPerformed(evt);
            }
        });
        jMenu5.add(jMenuItem8);

        jMenuItem20.setText("Metirial Item");
        jMenuItem20.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenuItem20ActionPerformed(evt);
            }
        });
        jMenu5.add(jMenuItem20);

        jMenu1.add(jMenu5);

        jMenu6.setText("Breakdown Details");

        jMenuItem1.setText("Breakdown type");
        jMenuItem1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenuItem1ActionPerformed(evt);
            }
        });
        jMenu6.add(jMenuItem1);

        jMenuItem23.setText("Breakdown Masters");
        jMenuItem23.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenuItem23ActionPerformed(evt);
            }
        });
        jMenu6.add(jMenuItem23);

        jMenuItem3.setText("KVA types");
        jMenuItem3.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenuItem3ActionPerformed(evt);
            }
        });
        jMenu6.add(jMenuItem3);

        jMenuItem9.setText("Zone Details");
        jMenuItem9.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        jMenuItem9ActionPerformed(evt);
    }
});
jMenu6.add(jMenuItem9);

jMenuItem10.setText("Grid Details");
jMenuItem10.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem10ActionPerformed(evt);
    }
});
jMenu6.add(jMenuItem10);

jMenuItem11.setText("Feeder Details");
jMenuItem11.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem11ActionPerformed(evt);
    }
});
jMenu6.add(jMenuItem11);

jMenuItem12.setText("Section Details");
jMenuItem12.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem12ActionPerformed(evt);
    }
});
jMenu6.add(jMenuItem12);

jMenuItem17.setText("Distribution substation");
jMenuItem17.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem17ActionPerformed(evt);
    }
});
jMenu6.add(jMenuItem17);

jMenuItem18.setText("LV Lines Details");
jMenuItem18.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem18ActionPerformed(evt);
    }
});
jMenu6.add(jMenuItem18);

jMenu1.add(jMenu6);

jMenuBar1.add(jMenu1);

jMenu2.setText("Transaction");
jMenu2.addActionListener(new java.awt.event.ActionListener() {
```

```
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenu2ActionPerformed(evt);
            }
        });

        jMenu7.setText("Breakdown entries");

        jMenuItem13.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_N,
java.awt.event.InputEvent.CTRL_MASK));
        jMenuItem13.setText("New Breakdown");
        jMenuItem13.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenuItem13ActionPerformed(evt);
            }
        });
        jMenu7.add(jMenuItem13);

        jMenuItem14.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_U,
java.awt.event.InputEvent.CTRL_MASK));
        jMenuItem14.setText("Update Breakdown");
        jMenuItem14.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenuItem14ActionPerformed(evt);
            }
        });
        jMenu7.add(jMenuItem14);

        jMenu2.add(jMenu7);

        jMenu8.setText("Breakdown Issues");

        jMenuItem15.setText("New issues");
        jMenuItem15.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenuItem15ActionPerformed(evt);
            }
        });
        jMenu8.add(jMenuItem15);

        jMenuItem16.setText("Update issues");
        jMenuItem16.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenuItem16ActionPerformed(evt);
            }
        });
        jMenu8.add(jMenuItem16);

        jMenu2.add(jMenu8);

        jMenuItem21.setText("Load Reading Data");
        jMenuItem21.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
      jMenuItem21ActionPerformed(evt);
    }
  });
  jMenu2.add(jMenuItem21);

  jMenuItem27.setText("Breakdown Restore");
  jMenuItem27.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
      jMenuItem27ActionPerformed(evt);
    }
  });
  jMenu2.add(jMenuItem27);

  jMenuBar1.add(jMenu2);

  jMenu3.setText("Report");
  jMenu3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
      jMenu3ActionPerformed(evt);
    }
  });

  jMenuItem19.setText("Top Most Breakdown Grid");
  jMenuItem19.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
      jMenuItem19ActionPerformed(evt);
    }
  });
  jMenu3.add(jMenuItem19);

  jMenuItem24.setText("Top Most Breakdown Feeder");
  jMenuItem24.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
      jMenuItem24ActionPerformed(evt);
    }
  });
  jMenu3.add(jMenuItem24);

  jMenuItem25.setText("Top Most Breakdown Section");
  jMenuItem25.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
      jMenuItem25ActionPerformed(evt);
    }
  });
  jMenu3.add(jMenuItem25);

  jMenuItem26.setText("Top Most Breakdown Sin");
  jMenuItem26.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
      jMenuItem26ActionPerformed(evt);
```

```
        }
    });
    jMenu3.add(jMenuItem26);

    jMenuItem22.setText("Breakdown Details Report");
    jMenuItem22.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem22ActionPerformed(evt);
      }
    });
    jMenu3.add(jMenuItem22);

    jMenuBar1.add(jMenu3);

    setJMenuBar(jMenuBar1);

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
      layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addComponent(jDesktopPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    );
    layout.setVerticalGroup(
      layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(layout.createSequentialGroup()
        .addComponent(jDesktopPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 672,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, Short.MAX_VALUE))
    );

    pack();
    setLocationRelativeTo(null);
  }// </editor-fold>

  private void jMenu5ActionPerformed(java.awt.event.ActionEvent evt) {

  }

  private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {
    region rs = new region();
    jPanel1.add(rs);
    rs.setLocation(350, 75);
    rs.setVisible(true);
  }

  private void jMenu1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
  }

  private void jMenuItem5ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    provinces ps = new provinces();
    jPanel1.add(ps);
    ps.setLocation(350, 75);
    ps.setVisible(true);
}

private void jMenuItem6ActionPerformed(java.awt.event.ActionEvent evt) {
    areas aria = new areas();
    jPanel1.add(aria);
    aria.setLocation(350, 75);
    aria.setVisible(true);
}

private void jMenuItem7ActionPerformed(java.awt.event.ActionEvent evt) {
    CSC csc = new CSC();
    jPanel1.add(csc);
    csc.setLocation(350, 20);
    csc.setVisible(true);
}

private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
    breakdown_type brtype = new breakdown_type();
    jPanel1.add(brtype);
    brtype.setLocation(350, 60);
    brtype.setVisible(true);
}

private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {
    kva_types kva = new kva_types();
    jPanel1.add(kva);
    kva.setLocation(350, 60);
    kva.setVisible(true);
}

private void jMenuItem9ActionPerformed(java.awt.event.ActionEvent evt) {
    zone_details zone = new zone_details();
    jPanel1.add(zone);
    zone.setLocation(350, 60);
    zone.setVisible(true);
}

private void jMenuItem10ActionPerformed(java.awt.event.ActionEvent evt) {
    grid_details grid = new grid_details();
    jPanel1.add(grid);
    grid.setLocation(350, 60);
    grid.setVisible(true);
}

private void jMenuItem11ActionPerformed(java.awt.event.ActionEvent evt) {
    feeder_details feeder = new feeder_details();
    jPanel1.add(feeder);
```

```
    feeder.setLocation(350, 20);
    feeder.setVisible(true);
}

private void jMenuItem12ActionPerformed(java.awt.event.ActionEvent evt) {
    section_details section = new section_details();
    jPanel1.add(section);
    section.setLocation(350, 20);
    section.setVisible(true);
}

private void jMenuItem13ActionPerformed(java.awt.event.ActionEvent evt) {
    newbreakdown_details nbrk = new newbreakdown_details();
    jPanel1.add(nbrk);
    nbrk.setLocation(150, 20);
    nbrk.setVisible(true);
}

private void jMenuItem8ActionPerformed(java.awt.event.ActionEvent evt) {
    user user = new user();
    jPanel1.add(user);
    user.setLocation(350, 20);
    user.setVisible(true);
}

private void jMenuItem17ActionPerformed(java.awt.event.ActionEvent evt) {
    distribution_substation ds = new distribution_substation();
    jPanel1.add(ds);
    ds.setLocation(350, 20);
    ds.setVisible(true);
}

private void jMenuItem18ActionPerformed(java.awt.event.ActionEvent evt) {
    lvlineDetails lv = new lvlineDetails();
    jPanel1.add(lv);
    lv.setLocation(350, 20);
    lv.setVisible(true);
}

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
    userAccount ua = new userAccount();
    jPanel1.add(ua);
    ua.jLabel7.setText(jLabel7.getText());
    ua.setLocation(150, 40);
    ua.setVisible(true);

    String userId = jLabel7.getText();
    String csc_name = "";
    try {
        Statement s = DB.getcon().createStatement();
        Statement s1 = DB.getcon().createStatement();
```

```
      ResultSet rs = s.executeQuery("SELECT * FROM `user` WHERE `user_id`='" + userId + "'");
      while (rs.next()) {
        ua.jLabel10.setText(rs.getString("province"));
        ua.jLabel12.setText(rs.getString("region"));
        csc_name = rs.getString("csc");

        ua.jLabel16.setText(rs.getString("area"));

      }
      ResultSet rs1 = s1.executeQuery("SELECT * FROM `csc` WHERE `name`='" + csc_name + "'");
      while (rs1.next()) {
        ua.jLabel14.setText(rs1.getString("csc_char_no"));
      }
    } catch (Exception e) {
      e.printStackTrace();
    }
}

private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}

private void jMenu3ActionPerformed(java.awt.event.ActionEvent evt) {

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    newbreakdown_details nbrk = new newbreakdown_details();
    jPanel1.add(nbrk);
    nbrk.setLocation(150, 20);
    nbrk.setVisible(true);
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    breakdown_reason brresone = new breakdown_reason();
    jPanel1.add(brresone);
    brresone.setLocation(150, 60);
    brresone.setVisible(true);
    brresone.jLabel8.setText(jLabel7.getText());
}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}

private void jMenuItem14ActionPerformed(java.awt.event.ActionEvent evt) {
    update_breakdown nbrku = new update_breakdown();
    jPanel1.add(nbrku);
    nbrku.setLocation(150, 20);
    nbrku.setVisible(true);
}
```

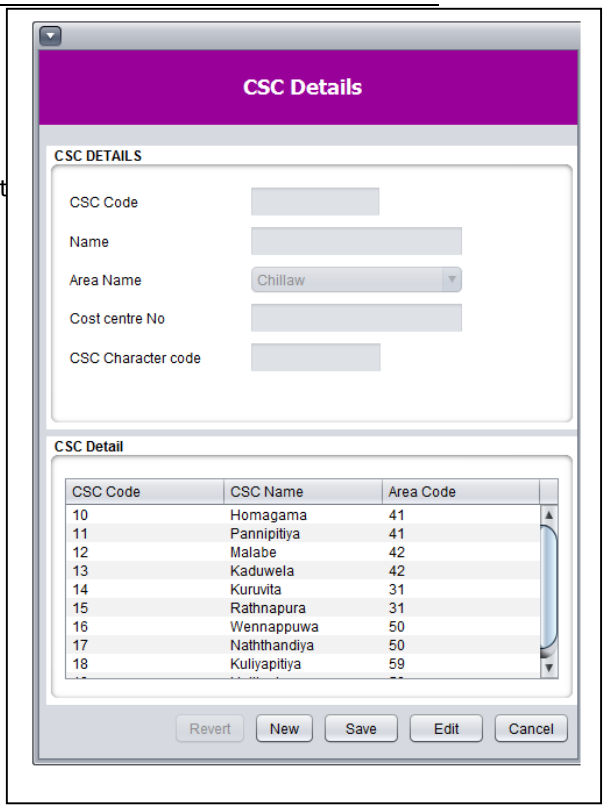
```
private void jMenuItem19ActionPerformed(java.awt.event.ActionEvent evt) {
   top_most_grid dmgreport = new top_most_grid();
   dmgreport.setVisible(true);
}

private void jMenuItem20ActionPerformed(java.awt.event.ActionEvent evt) {
   add_item additem = new add_item();
   jPanel1.add(additem);
   additem.setLocation(150, 20);
   additem.setVisible(true);
}

private void jMenuItem15ActionPerformed(java.awt.event.ActionEvent evt) {
   breakdown_issues brissue = new breakdown_issues();
   jPanel1.add(brissue);
   brissue.setLocation(150, 20);
   brissue.setVisible(true);
}

private void jMenuItem21ActionPerformed(java.awt.event.ActionEvent evt) {
   load_reading_data lrd = new load_reading_data();
   jPanel1.add(lrd);
   lrd.setLocation(150, 20);
   lrd.setVisible(true);
}

private void jMenuItem22ActionPerformed(java.awt.event.ActionEvent evt) {
   time_report timereport = new time_report();
   timereport.setVisible(true);
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
   user asd = new user();
   jPanel1.add(asd);
   asd.setLocation(150, 20);
   asd.setVisible(true);
}

private void jMenuItem16ActionPerformed(java.awt.event.ActionEvent evt) {
   breakdownIsuarUpdate issueup = new breakdownIsuarUpdate();
   jPanel1.add(issueup);
   issueup.setLocation(150, 20);
   issueup.setVisible(true);
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
   breakdown_issues brissue = new breakdown_issues();
   jPanel1.add(brissue);
   brissue.setLocation(150, 20);
   brissue.setVisible(true);
```

```
    }

    private void jMenuItem23ActionPerformed(java.awt.event.ActionEvent evt) {
        create_resons brrtype = new create_resons();
        jPanel1.add(brrtype);
        brrtype.setLocation(350, 60);
        brrtype.setVisible(true);
    }

    private void jMenuItem24ActionPerformed(java.awt.event.ActionEvent evt) {
        top_most_feeder dmgfeeder = new top_most_feeder();
        dmgfeeder.setVisible(true);
    }

    private void jMenuItem25ActionPerformed(java.awt.event.ActionEvent evt) {
        top_most_section dmgsection = new top_most_section();
        dmgsection.setVisible(true);
    }

    private void jMenuItem26ActionPerformed(java.awt.event.ActionEvent evt) {
        top_most_sin dmgsin = new top_most_sin();
        dmgsin.setVisible(true);
    }

    private void jMenuItem27ActionPerformed(java.awt.event.ActionEvent evt) {
        breakdown_reason brresone = new breakdown_reason();
        jPanel1.add(brresone);
        brresone.setLocation(150, 60);
        brresone.setVisible(true);
        brresone.jLabel8.setText(jLabel7.getText());
    }

    private void jMenu2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
```

```
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(home.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(home.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(home.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(home.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new home().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
public static javax.swing.JButton jButton1;
public static javax.swing.JButton jButton2;
public static javax.swing.JButton jButton3;
public static javax.swing.JButton jButton4;
public static javax.swing.JButton jButton5;
public static javax.swing.JButton jButton6;
public static javax.swing.JButton jButton7;
public static javax.swing.JDesktopPane jDesktopPane1;
public static javax.swing.JLabel jLabel1;
public static javax.swing.JLabel jLabel2;
public static javax.swing.JLabel jLabel3;
public static javax.swing.JLabel jLabel4;
public static javax.swing.JLabel jLabel5;
public static javax.swing.JLabel jLabel6;
public static javax.swing.JLabel jLabel7;
public static javax.swing.JLabel jLabel8;
public static javax.swing.JMenu jMenu1;
public static javax.swing.JMenu jMenu2;
public static javax.swing.JMenu jMenu3;
public static javax.swing.JMenu jMenu5;
public static javax.swing.JMenu jMenu6;
public static javax.swing.JMenu jMenu7;
public static javax.swing.JMenu jMenu8;
public static javax.swing.JMenuBar jMenuBar1;
public static javax.swing.JMenuItem jMenuItem1;
```

```
    public static javax.swing.JMenuItem jMenuItem10;
    public static javax.swing.JMenuItem jMenuItem11;
    public static javax.swing.JMenuItem jMenuItem12;
    public static javax.swing.JMenuItem jMenuItem13;
    public static javax.swing.JMenuItem jMenuItem14;
    public static javax.swing.JMenuItem jMenuItem15;
    public static javax.swing.JMenuItem jMenuItem16;
    public static javax.swing.JMenuItem jMenuItem17;
    public static javax.swing.JMenuItem jMenuItem18;
    public static javax.swing.JMenuItem jMenuItem19;
    public static javax.swing.JMenuItem jMenuItem20;
    public static javax.swing.JMenuItem jMenuItem21;
    public static javax.swing.JMenuItem jMenuItem22;
    public static javax.swing.JMenuItem jMenuItem23;
    public static javax.swing.JMenuItem jMenuItem24;
    public static javax.swing.JMenuItem jMenuItem25;
    public static javax.swing.JMenuItem jMenuItem26;
    public static javax.swing.JMenuItem jMenuItem27;
    public static javax.swing.JMenuItem jMenuItem3;
    public static javax.swing.JMenuItem jMenuItem4;
    public static javax.swing.JMenuItem jMenuItem5;
    public static javax.swing.JMenuItem jMenuItem6;
    public static javax.swing.JMenuItem jMenuItem7;
    public static javax.swing.JMenuItem jMenuItem8;
    public static javax.swing.JMenuItem jMenuItem9;
    public static javax.swing.JPanel jPanel1;
    public static javax.swing.JPanel jPanel2;
    // End of variables declaration

}
```

---




```
/*
 * To change this license header, choose License Headers in Project
 Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package pages;

import connection.DB;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.Vector;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/**
```

```
 *
 * @author MyLaptop
 */
public class CSC extends javax.swing.JInternalFrame {

    /**
     * Creates new form CSC
     */
    public CSC() {
        initComponents();
        loadArea();
        textfeald_desable();
        loadcscdata();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        jLabel2 = new javax.swing.JLabel();
        txtcsccode = new javax.swing.JTextField();
        txtname = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jComboBox1 = new javax.swing.JComboBox();
        jLabel5 = new javax.swing.JLabel();
        txtcostcenter = new javax.swing.JTextField();
        jLabel6 = new javax.swing.JLabel();
        txtcsccharacter_no = new javax.swing.JTextField();
        jPanel2 = new javax.swing.JPanel();
        jScrollPane1 = new javax.swing.JScrollPane();
        jTable1 = new javax.swing.JTable();
        jButton3 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jButton1 = new javax.swing.JButton();
        jButton4 = new javax.swing.JButton();
        revert = new javax.swing.JButton();
        jPanel3 = new javax.swing.JPanel();
        jLabel7 = new javax.swing.JLabel();

        setBackground(new java.awt.Color(255, 255, 255));

        jPanel1.setBackground(new java.awt.Color(255, 255, 255));
        jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder("CSC DETAILS"));
```

```
        jLabel2.setText("CSC Code");

        jLabel3.setText("Name");

        jLabel4.setText("Area Name");

        jLabel5.setText("Cost centre No");

        jLabel6.setText("CSC Character code");

        javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jLabel3)
                    .addComponent(jLabel4)
                    .addComponent(jLabel5)
                    .addComponent(jLabel2)
                    .addComponent(jLabel6))
                .addGap(41, 41, 41)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(txtcsccode, javax.swing.GroupLayout.PREFERRED_SIZE, 114,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
                        .addComponent(jComboBox1, 0, 185, Short.MAX_VALUE)
                        .addComponent(txtname)
                        .addComponent(txtcostcenter))
                    .addComponent(txtcsccharacter_no, javax.swing.GroupLayout.PREFERRED_SIZE, 115,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );
        jPanel1Layout.setVerticalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel2)
                    .addComponent(txtcsccode, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel3)
                    .addComponent(txtname, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel4)
```

```
                .addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel5)
                .addComponent(txtcostcenter, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel6)
                .addComponent(txtcsccharacter_no, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addContainerGap(34, Short.MAX_VALUE))
        );

        jPanel2.setBackground(new java.awt.Color(255, 255, 255));
        jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder("CSC Detail"));

        jTable1.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {

            },
            new String [] {
                "CSC Code", "CSC Name", "Area Code"
            }
        ));
        jTable1.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                jTable1MouseClicked(evt);
            }
        });
        jScrollPane1.setViewportView(jTable1);

        javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 0, Short.MAX_VALUE)
        );
        jPanel2Layout.setVerticalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel2Layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 177,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap())
        );

        jButton3.setText("Cancel");
        jButton3.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
      jButton3ActionPerformed(evt);
    }
});

jButton2.setText("Edit");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
      jButton2ActionPerformed(evt);
    }
});

jButton1.setText("Save");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
      jButton1ActionPerformed(evt);
    }
});

jButton4.setText("New");
jButton4.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
      jButton4ActionPerformed(evt);
    }
});

revert.setText("Revert");
revert.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
      revertActionPerformed(evt);
    }
});

jPanel3.setBackground(new java.awt.Color(153, 0, 153));

jLabel7.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
jLabel7.setForeground(new java.awt.Color(255, 255, 255));
jLabel7.setText("CSC Details");

javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
jPanel3.setLayout(jPanel3Layout);
jPanel3Layout.setHorizontalGroup(
    jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel3Layout.createSequentialGroup()
      .addGap(175, 175, 175)
      .addComponent(jLabel7)
      .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);
jPanel3Layout.setVerticalGroup(
    jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel3Layout.createSequentialGroup()
```

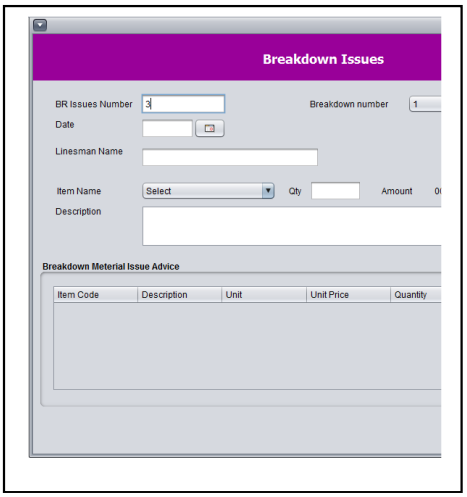
```
            .addGap(20, 20, 20)
            .addComponent(jLabel7)
            .addContainerGap(22, Short.MAX_VALUE))
        );

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
          layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
            .addContainerGap(115, Short.MAX_VALUE)
            .addComponent(revert)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jButton4)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 67,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 67,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jButton3, javax.swing.GroupLayout.PREFERRED_SIZE, 67,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap())
          .addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
          .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
              .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
              .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
        );
        layout.setVerticalGroup(
          layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(layout.createSequentialGroup()
            .addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
              .addComponent(jButton1)
              .addComponent(jButton2)
              .addComponent(jButton3)
              .addComponent(jButton4)
```

```
                .addComponent(revert))
            .addContainerGap(18, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        textfeald_enable();
        revert.setEnabled(true);
        jButton2.setEnabled(false);
        jButton1.setEnabled(false);
        jButton4.setEnabled(false);
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        String csccode = txtcsccode.getText();
        String name = txtname.getText();
        String costcenter = txtcostcenter.getText();
        String charactercode = txtcsccharacter_no.getText();
        String areaname = jComboBox1.getSelectedItem().toString();
        String areacode = "";

        if (csccode.isEmpty() || name.isEmpty() || costcenter.isEmpty() || charactercode.isEmpty()) {
            JOptionPane.showMessageDialog(this, "Some textfeaild is empty");

        } else {
            try {
                Statement s = DB.getcon().createStatement();

                ResultSet rs = s.executeQuery("SELECT * FROM `csc` WHERE `csc_code`='" + csccode + "' or
`name` = '" + name + "'");
                if (rs.next()) {
                    JOptionPane.showMessageDialog(this, "Alreddy Active csc");
                } else {
                    s.executeUpdate("INSERT INTO `csc`(`csc_code`, `name`, `cost_center_no`, `csc_char_no`)
VALUES ('" + csccode + "','" + name + "','" + costcenter + "','" + charactercode + "')");

                    ResultSet rs1 = s.executeQuery("SELECT * FROM `area` WHERE `name`='" + areaname + "'");
                    while (rs1.next()) {
                        areacode = rs1.getString("area_code");
                    }
                    s.executeUpdate("INSERT INTO `area_csc`(`csc_csc_code`, `area_area_code`) VALUES ('" +
csccode + "','" + areacode + "')");
                    JOptionPane.showMessageDialog(this, "Successful entryes");
                    loadcscdata();
                }
```

```
        } catch (Exception e) {
        }
      }
    }

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
      textfeald_enable();
    }

    private void revertActionPerformed(java.awt.event.ActionEvent evt) {
      revert.setEnabled(false);
      jButton2.setEnabled(true);
      jButton1.setEnabled(true);
      jButton4.setEnabled(true);

      try {
        Statement s = DB.getcon().createStatement();
        String csc_code = txtcsccode.getText();
        String name = txtname.getText();
        String cost_center = txtcostcenter.getText();
        String csc_charctr_code = txtcsccharacter_no.getText();
        String area_name = jComboBox1.getSelectedItem().toString();
        int areacode = 0;

        s.executeUpdate("UPDATE `csc` SET `name`='" + name + "',`cost_center_no`='" + cost_center +
"',`csc_char_no`='" + csc_charctr_code + "' WHERE `csc_code`='" + csc_code + "'");

        ResultSet rs2 = s.executeQuery("SELECT * FROM `area` WHERE `name`='" + area_name + "'");
        while (rs2.next()) {
          areacode = rs2.getInt("area_code");
        }

        s.executeUpdate("UPDATE `area_csc` SET `area_area_code`='" + areacode + "' WHERE
`csc_csc_code`='" + csc_code + "'");

        JOptionPane.showMessageDialog(this, "Update Successful");
        loadcscdata();
      } catch (Exception e) {
      }
    }

    private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
      DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();
      int selectedrow = jTable1.getSelectedRow();

      txtcsccode.setText(dtm.getValueAt(selectedrow, 0).toString());
      txtname.setText(dtm.getValueAt(selectedrow, 1).toString());
      String areacode = dtm.getValueAt(selectedrow, 2).toString();
      String csccode = dtm.getValueAt(selectedrow, 0).toString();
      try {
```

```
        Statement s = DB.getcon().createStatement();
        Statement s1 = DB.getcon().createStatement();
        ResultSet rs = s.executeQuery("SELECT * FROM `area` WHERE `area_code`='" + areacode + "'");
        while (rs.next()) {
          jComboBox1.setSelectedItem(rs.getString("name"));
          ResultSet rs1 = s1.executeQuery("SELECT * FROM `csc` WHERE `csc_code`='" + csccode + "'");
          while (rs1.next()) {
            txtcostcenter.setText(rs1.getString("cost_center_no"));
            txtcsccharacter_no.setText(rs1.getString("csc_char_no"));
          }
        }
      } catch (Exception e) {
      }
    }


    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JButton jButton4;
    private javax.swing.JComboBox jComboBox1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTable jTable1;
    private javax.swing.JButton revert;
    private javax.swing.JTextField txtcostcenter;
    private javax.swing.JTextField txtcsccharacter_no;
    private javax.swing.JTextField txtcsccode;
    private javax.swing.JTextField txtname;
    // End of variables declaration
  void textfeald_desable() {
      txtcsccode.setEnabled(false);
      txtname.setEnabled(false);
      txtcostcenter.setEnabled(false);
      txtcsccharacter_no.setEnabled(false);
      revert.setEnabled(false);
      jComboBox1.setEnabled(false);
    }

    void textfeald_enable() {
      txtcsccode.setEnabled(true);
      txtname.setEnabled(true);
```

```
        txtcostcenter.setEnabled(true);
        txtcsccharacter_no.setEnabled(true);
        revert.setEnabled(false);
        jComboBox1.setEnabled(true);
    }

    void loadArea() {
        try {
            Statement s = DB.getcon().createStatement();
            ResultSet rs = s.executeQuery("SELECT * FROM `area`");
            while (rs.next()) {
                jComboBox1.addItem(rs.getString("name"));

            }
        } catch (Exception e) {
        }
    }

    void loadcscdata() {
        try {
            String x;
            String procode;
            Statement s = DB.getcon().createStatement();
            Statement s1 = DB.getcon().createStatement();
            String sql = "SELECT * from `csc`";
            ResultSet rs = s.executeQuery(sql);
            DefaultTableModel td = (DefaultTableModel) jTable1.getModel();
            td.setRowCount(0);

            while (rs.next()) {
                Vector v = new Vector();
                x = rs.getString("csc_code");
                String sql2 = "SELECT * FROM `area_csc` WHERE `csc_csc_code`='" + x + "'";
                ResultSet rsq = s1.executeQuery(sql2);
                v.add(rs.getString("csc_code"));
                v.add(rs.getString("name"));
                while (rsq.next()) {
                    v.add(rsq.getString("area_area_code"));
                }

                td.addRow(v);
            }

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```




```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package pages;

import connection.DB;
import java.awt.event.KeyEvent;
import java.sql.ResultSet;
import java.sql.Statement;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Vector;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author MyLaptop
 */
public class breakdown_issues extends javax.swing.JInternalFrame {

    /**
     * Creates new form breakdown_issues
     */
    public breakdown_issues() {
        initComponents();
        searchBreakdownNo();
        ItemNameLoad();
        setIssurID();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jPanel2 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jTextField1 = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        jDateChooser1 = new com.toedter.calendar.JDateChooser();
        jLabel4 = new javax.swing.JLabel();
        jComboBox1 = new javax.swing.JComboBox();
```

```
        jPanel1 = new javax.swing.JPanel();
        jScrollPane1 = new javax.swing.JScrollPane();
        jTable1 = new javax.swing.JTable();
        jLabel5 = new javax.swing.JLabel();
        jComboBox2 = new javax.swing.JComboBox();
        jLabel6 = new javax.swing.JLabel();
        jTextField2 = new javax.swing.JTextField();
        jLabel7 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();
        jLabel9 = new javax.swing.JLabel();
        jScrollPane2 = new javax.swing.JScrollPane();
        jTextArea1 = new javax.swing.JTextArea();
        jLabel10 = new javax.swing.JLabel();
        jTextField3 = new javax.swing.JTextField();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();

        jPanel2.setBackground(new java.awt.Color(153, 0, 153));

        jLabel1.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
        jLabel1.setForeground(new java.awt.Color(255, 255, 255));
        jLabel1.setText("Breakdown Issues");

        javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel2Layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jLabel1)
                .addGap(272, 272, 272))
        );
        jPanel2Layout.setVerticalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel2Layout.createSequentialGroup()
                .addContainerGap(23, Short.MAX_VALUE)
                .addComponent(jLabel1)
                .addGap(19, 19, 19))
        );

        jLabel2.setText("BR Issues Number");

        jLabel3.setText("Date");

        jLabel4.setText("Breakdown number");

        jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder("Breakdown Meterial Issue
Advice"));

        jTable1.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
```

```
            },
            new String [] {
                "Item Code", "Description", "Unit", "Unit Price", "Quantity", "Amount"
            }
        ));
        jScrollPane1.setViewportView(jTable1);

        javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jScrollPane1)
                .addContainerGap())
        );
        jPanel1Layout.setVerticalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 147,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(13, Short.MAX_VALUE))
        );

        jLabel5.setText("Item Name");

        jComboBox2.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Select" }));
        jComboBox2.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                jComboBox2MouseClicked(evt);
            }
        });
        jComboBox2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jComboBox2ActionPerformed(evt);
            }
        });

        jLabel6.setText("Qty");

        jTextField2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jTextField2ActionPerformed(evt);
            }
        });

        jLabel7.setText("Amount");

        jLabel8.setText("00.00");
```

```
        jLabel9.setText("Description");

        jTextArea1.setColumns(20);
        jTextArea1.setRows(5);
        jTextArea1.addAncestorListener(new javax.swing.event.AncestorListener() {
            public void ancestorMoved(javax.swing.event.AncestorEvent evt) {
            }
            public void ancestorAdded(javax.swing.event.AncestorEvent evt) {
                jTextArea1AncestorAdded(evt);
            }
            public void ancestorRemoved(javax.swing.event.AncestorEvent evt) {
            }
        });
        jTextArea1.addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyPressed(java.awt.event.KeyEvent evt) {
                jTextArea1KeyPressed(evt);
            }
        });
        jScrollPane2.setViewportView(jTextArea1);

        jLabel10.setText("Linesman Name");

        jButton1.setText("Save");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        jButton2.setText("Close");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel2,                                    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addGroup(layout.createSequentialGroup()
                .addGap(31, 31, 31)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jLabel2)
                    .addComponent(jLabel3)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                        .addComponent(jLabel5)
                        .addComponent(jLabel9))
```

```
            .addComponent(jLabel10))
          .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
          .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
              .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                  .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
                    .addComponent(jTextField1)
                    .addComponent(jDateChooser1, javax.swing.GroupLayout.DEFAULT_SIZE, 117,
Short.MAX_VALUE))
                  .addGap(116, 116, 116)
                  .addComponent(jLabel4)
                  .addGap(27, 27, 27)
                  .addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED_SIZE, 81,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(layout.createSequentialGroup()
                  .addComponent(jComboBox2, javax.swing.GroupLayout.PREFERRED_SIZE, 186,
javax.swing.GroupLayout.PREFERRED_SIZE)
                  .addGap(18, 18, 18)
                  .addComponent(jLabel6)
                  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                  .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE, 71,
javax.swing.GroupLayout.PREFERRED_SIZE)
                  .addGap(28, 28, 28)
                  .addComponent(jLabel7)
                  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 31,
Short.MAX_VALUE)
                  .addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED_SIZE, 98,
javax.swing.GroupLayout.PREFERRED_SIZE)))
              .addGap(104, 104, 104))
            .addGroup(layout.createSequentialGroup()
              .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 436,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED_SIZE, 246,
javax.swing.GroupLayout.PREFERRED_SIZE))
              .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
          .addContainerGap()
          .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
          .addContainerGap())
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
          .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
          .addComponent(jButton1)
          .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
          .addComponent(jButton2)
          .addGap(20, 20, 20))
    );
    layout.setVerticalGroup(
```

```
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel2)
                .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel4)
                .addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel3)
                .addComponent(jDateChooser1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(9, 9, 9)
                    .addComponent(jLabel10))
                .addGroup(layout.createSequentialGroup()
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                    .addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addGap(18, 18, 18)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jComboBox2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel6)
                .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel7)
                .addComponent(jLabel8)
                .addComponent(jLabel5))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel9)
                .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 58,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(18, 18, 18)
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jButton1)
                .addComponent(jButton2))
            .addContainerGap(31, Short.MAX_VALUE))
    );

    pack();
```

```
}// </editor-fold>

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
   this.dispose();
}

private void jComboBox2MouseClicked(java.awt.event.MouseEvent evt) {

}

private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {

   String itemname = jComboBox2.getSelectedItem().toString();
   double unitPrice = 0.0;
   String qty = jTextField2.getText();
   double qtx = Double.parseDouble(qty);
   try {
      Statement s = DB.getcon().createStatement();
      ResultSet rs = s.executeQuery("SELECT * FROM `item` WHERE `name`='" + itemname + "'");
      while (rs.next()) {
         unitPrice = rs.getDouble("unit_price");
      }
      double x = unitPrice * qtx;
      jLabel8.setText(String.valueOf(x));
   } catch (Exception e) {
      e.printStackTrace();
   }
}

private void jTextArea1AncestorAdded(javax.swing.event.AncestorEvent evt) {

}

private void jTextArea1KeyPressed(java.awt.event.KeyEvent evt) {
   if (evt.getKeyCode() == KeyEvent.VK_ENTER) {
      String itemname = jComboBox2.getSelectedItem().toString();
      String qty = jTextField2.getText();
      String amount = jLabel8.getText();
      String description = jTextArea1.getText();
      try {
         Statement s = DB.getcon().createStatement();
         ResultSet rs = s.executeQuery("SELECT * FROM `item` WHERE `name`='" + itemname + "'");
         DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();
         while (rs.next()) {
            Vector v = new Vector();
            v.add(rs.getString("item_code"));
            v.add(description);
            v.add(rs.getString("unit"));
            v.add(rs.getString("unit_price"));
            v.add(qty);
            v.add(amount);
```

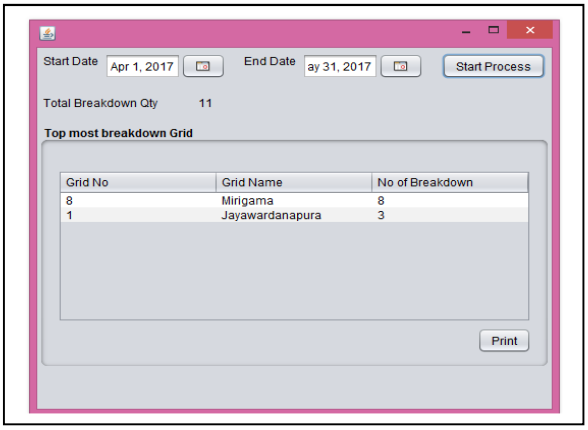
```
                dtm.addRow(v);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    String brIssurID = jTextField1.getText();
    String breakdownNo = jComboBox1.getSelectedItem().toString();
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
    Date date = jDateChooser1.getDate();

    String linsman_name = jTextField3.getText();
    String itemName = jComboBox2.getSelectedItem().toString();
    String qty = jTextField2.getText();
    String amount = jLabel8.getText();

    try {
        Statement s = DB.getcon().createStatement();
        ResultSet rs = s.executeQuery("SELECT * FROM `breakdown_issur` WHERE `br_issue_number`='" +
brIssurID + "'");
        if (rs.next()) {
            JOptionPane.showMessageDialog(this, "Already Active Issur Number");
        }
        if (linsman_name.isEmpty()) {
            JOptionPane.showMessageDialog(this, "Try Again! lines man name empty");
        } else {
s.executeUpdate("INSERT INTO `breakdown_issur`(`br_issue_number`, `date`, `linsman_name`) VALUES
('" + brIssurID + "','" + date + "','" + linsman_name + "')");
s.executeUpdate("INSERT INTO `breakdown_issues`(`breakdown_issur_br_issue_number`,
`breakdown_code`) VALUES ('" + brIssurID + "','" + breakdownNo + "')");

            DefaultTableModel dsa = (DefaultTableModel) jTable1.getModel();
            for (int i = 0; i < jTable1.getRowCount(); i++) {
                String t1 = dsa.getValueAt(i, 0).toString();
                String t2 = dsa.getValueAt(i, 1).toString();
                String t3 = dsa.getValueAt(i, 2).toString();
                String t4 = dsa.getValueAt(i, 3).toString();
                String t5 = dsa.getValueAt(i, 4).toString();
                String t6 = dsa.getValueAt(i, 5).toString();

s.executeUpdate("INSERT INTO `breakdown_item`(`breakdown_issur_br_issue_number`,
`item_item_code`, `qty`, `amount`, `description`) VALUES
('"+brIssurID+"','"+t1+"','"+t5+"','"+t6+"','"+t2+"')");
            }
```

```
      JOptionPane.showMessageDialog(this, "Successful Entry");
       }
   } catch (Exception e) {
      e.printStackTrace();
   }

}

private void jComboBox2ActionPerformed(java.awt.event.ActionEvent evt) {
   // TODO add your handling code here:
}


// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JComboBox jComboBox1;
private javax.swing.JComboBox jComboBox2;
private com.toedter.calendar.JDateChooser jDateChooser1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JTable jTable1;
private javax.swing.JTextArea jTextArea1;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField3;
// End of variables declaration

void searchBreakdownNo() {
   try {
      Statement s = DB.getcon().createStatement();
      ResultSet rs = s.executeQuery("SELECT * FROM `breakdown`");
      while (rs.next()) {
         jComboBox1.addItem(rs.getInt("code"));
      }
   } catch (Exception e) {
   }
}
```

```
    void ItemNameLoad() {
        try {
            Statement s = DB.getcon().createStatement();
            ResultSet rs1 = s.executeQuery("SELECT * FROM `item`");
            while (rs1.next()) {
                jComboBox2.addItem(rs1.getString("name"));
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    void setIssurID() {
        int issurNo = 1;

        try {
            Statement s = DB.getcon().createStatement();
            ResultSet rs = s.executeQuery("SELECT MAX(`br_issue_number`) as `maxid` FROM
`breakdown_issur`");
            while (rs.next()) {
                issurNo = rs.getInt("maxid");
                issurNo++;
            }

            jTextField1.setText(String.valueOf(issurNo));
        } catch (Exception e) {
        }

    }

}
```

---

```
*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package Report;


import static ceb.home.jLabel4;
import connection.DB;
import java.sql.ResultSet;
import java.sql.Statement;
import java.text.SimpleDateFormat;
```



```
import java.util.Date;
import java.util.Vector;
import javax.swing.table.DefaultTableModel;
import static pages.breakdown_reason.jTable1;

/**
 *
 * @author MyLaptop
 */
public class top_most_grid extends javax.swing.JFrame {

    /**
     * Creates new form top_most_grid
     */
    public top_most_grid() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jDateChooser1 = new com.toedter.calendar.JDateChooser();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jDateChooser2 = new com.toedter.calendar.JDateChooser();
        jButton1 = new javax.swing.JButton();
        jPanel3 = new javax.swing.JPanel();
        jScrollPane2 = new javax.swing.JScrollPane();
        jTable2 = new javax.swing.JTable();
        jButton2 = new javax.swing.JButton();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

        jLabel1.setText("Start Date");

        jLabel2.setText("End Date");

        jButton1.setText("Start Process");
```

```
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        jPanel3.setBorder(javax.swing.BorderFactory.createTitledBorder("Top most breakdown
Grid"));

        jTable2.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {

            },
            new String [] {
                "Grid No", "Grid Name", "No of Breakdown"
            }
        ));
        jScrollPane2.setViewportView(jTable2);

        jButton2.setText("Print");

        javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
        jPanel3.setLayout(jPanel3Layout);
        jPanel3Layout.setHorizontalGroup(
            jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel3Layout.createSequentialGroup()
                .addContainerGap()

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 0,
Short.MAX_VALUE)
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel3Layout.createSequentialGroup()
                        .addGap(0, 0, Short.MAX_VALUE)
                        .addComponent(jButton2)))
                .addContainerGap())
        );
        jPanel3Layout.setVerticalGroup(
            jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel3Layout.createSequentialGroup()
                .addGap(21, 21, 21)
                .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 167,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jButton2)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
```

```
      );

      jLabel3.setText("Total Breakdown Qty");

      jLabel4.setText("jLabel4");

      javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
      getContentPane().setLayout(layout);
      layout.setHorizontalGroup(
         layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
         .addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
         .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
               .addGroup(layout.createSequentialGroup()
                  .addComponent(jLabel1)
                  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                  .addComponent(jDateChooser1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                  .addGap(18, 18, 18)
                  .addComponent(jLabel2)
                  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                  .addComponent(jDateChooser2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                  .addGap(18, 18, 18)
                  .addComponent(jButton1))
               .addGroup(layout.createSequentialGroup()
                  .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 134,
javax.swing.GroupLayout.PREFERRED_SIZE)
                  .addGap(18, 18, 18)
                  .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 95,
javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
      );
      layout.setVerticalGroup(
         layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
         .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                  .addComponent(jLabel1)
                  .addComponent(jLabel2))
               .addComponent(jDateChooser1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
                .addComponent(jDateChooser2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jButton1))
            .addGap(18, 18, 18)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel3)
                .addComponent(jLabel4))
            .addGap(17, 17, 17)
            .addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(41, Short.MAX_VALUE))
        );

        pack();
        setLocationRelativeTo(null);
    }// </editor-fold>

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy/MM/dd");
        Date d1 = jDateChooser1.getDate();
        String date1 = sdf.format(d1);
        Date d2 = jDateChooser2.getDate();
        String date2 = sdf.format(d2);

        String gridname = "";


        try {
            Statement s = DB.getcon().createStatement();
            Statement s2 = DB.getcon().createStatement();
            ResultSet rs = s.executeQuery("SELECT COUNT(`code`) as `brcount` from `breakdown`
where `date_time` BETWEEN '" + date1 + "' and '" + date2 + "'");

            while (rs.next()) {
                jLabel4.setText(rs.getString("brcount"));
            }

        } catch (Exception e) {
            e.printStackTrace();
        }

        try {
            DefaultTableModel td = (DefaultTableModel) jTable2.getModel();
            td.setRowCount(0);
            Statement s = DB.getcon().createStatement();
            Statement s1 = DB.getcon().createStatement();
```

```
        Statement s2 = DB.getcon().createStatement();
        ResultSet rsx = s.executeQuery(" SELECT DISTINCT `grid` FROM `breakdown` WHERE
`date_time` between '" + date1 + "' and '" + date2 + "'");
        while (rsx.next()) {
          Vector v = new Vector();
          gridname = rsx.getString("grid");
          ResultSet rex = s1.executeQuery("SELECT * FROM `grid_details` WHERE `name`='" +
gridname + "'");
          while (rex.next()) {
            v.add(rex.getString("grid_code"));
            v.add(rex.getString("name"));
            ResultSet ress = s2.executeQuery("SELECT COUNT(`grid`) as `dmgridcount` from
`breakdown` where `grid` = '" + gridname + "'");
            while(ress.next()){
            v.add(ress.getString("dmgridcount"));
            }
          }
          td.addRow(v);
        }
      } catch (Exception e) {
        e.printStackTrace();
      }

    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
      /* Set the Nimbus look and feel */
      //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
      /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
       * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
       */
      try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
          if ("Nimbus".equals(info.getName())) {
            javax.swing.UIManager.setLookAndFeel(info.getClassName());
            break;
          }
        }
      } catch (ClassNotFoundException ex) {
```

```
java.util.logging.Logger.getLogger(top_most_grid.class.getName()).log(java.util.logging.Level.SE
VERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(top_most_grid.class.getName()).log(java.util.logging.Level.SE
VERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(top_most_grid.class.getName()).log(java.util.logging.Level.SE
VERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(top_most_grid.class.getName()).log(java.util.logging.Level.SE
VERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new top_most_grid().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private com.toedter.calendar.JDateChooser jDateChooser1;
    private com.toedter.calendar.JDateChooser jDateChooser2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JTable jTable2;
    // End of variables declaration
}
```