A Sales Force Automation [SFA] Solution for a complexly diversified Organization

M D P S Chathuranga

169306V

Faculty of Information Technology

University of Moratuwa

2019

A Sales Force Automation [SFA] Solution for a complexly diversified Organization

M D P S Chathuranga

169306V

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

February 2019

Declaration

I declare that this thesis is my 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 Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Name of Student **Signature of Student**

M D P S Chathuranga ……

Date:

Supervised by:

Name of Supervisor **Signature of supervisor**

Mr. Saminda Premarathna ……

Date:

Acknowledgement

First and foremost, I would like to express my honest gratitude to my supervisor, Mr. Saminda Premarathne, Senior Lecturer, Faculty of Information Technology, University of Moratuwa. During this research completion period, he spent much valuable time for superior guidance, supervision, and advice to complete this as the best manner. With his guidance, I understand and identify the correct and efficient way to complete the research and add more valuable modules. Also, I would like to express gratitude to all who supported me to complete this project.

Abstract

Nowadays, there is a massive number of business are competing in the industry to achieve their business goals and earn high profits. Furthermore, many large organizations consist of complexly diversified sales based businesses across the globe. Many businesses are depending on each other, and some are running independently. In the software industry, most of IT software solutions are developing based on requirements analysis and considering the individual behaviors of the business.

For a Sales Force Automation System (SFA) this becomes a large scaled problem when implementing a system from one business to another business. Diversities of the businesses, less adoption and software bugs may occur when implementing one SFA system to another business.

In this thesis, identify the problems of the existing sales force automation system's implementations among multiple businesses and deliver a proper dynamic flow-based software solution (implementation) with rectifying those problems.

Table of Content

Declaration i
Acknowledgement ii
Abstract iii
Table of Content iv
List of Figures vii
Table of Content viii
Chapter 1 1
Introduction 1
1.1 Prolegomena 1
1.2 Problem Statement 1
1.3 Aim and Objectives 2
1.3 Background and Motivation 2
1.4 Problem Definition 3
1.4 Proposed Solution 3
1.5 Summary 4
Chapter 2 5
Literature Review 5
2.1 Introduction 5
2.2 Importance of a Cloud and Mobile based solution 6
2.3 Advantages of cloud computing 7
2.4 Disadvantages of cloud computing 7
2.4 Microsoft Azure Machine Learning 7
2.6 Machine Learning Algorithms 8
2.5 Importance of Azure Machine Learning 8
2.6 Summary 9
Chapter 3 10
Technology Adopted 10
3.1 Introduction 10
3.2 Microsoft Visual Studio (Programming IDE) 10
3.3 Microsoft SQL Server (Database) 11
3.4 Xamarin (Mobile development) 12
3.5 Proposed system's Development Architecture 13

3.6 Summary....14
Chapter 4....15
Methodology....15
4.1 Introduction....15
4.2 Methodology of the Proposed System....15
4.2.1 Dynamic Sales Flow....15
4.2.2 Sales Prediction....16
4.3 Summary....16
Chapter 5....17
Analysis and Design....17
5.1 Introduction....17
5.2 Project flow....17
5.3 Top level design view of the system....18
5.4 Multi-Tenant Architecture....19
5.4 Single Tenant vs Multiple Tenant....20
5.5 Proposed System's Multi-Tenant Architecture....21
5.6 Summary....21
Chapter 6....22
Implementation....22
6.1 Introduction....22
6.2 Requirement Gathering and Analysis....22
6.3 Design....23
6.4 Development....23
6.6 Summary....23
Chapter 7....24
Evaluation....24
7.1 Introduction....24
7.2 Evaluation of Dynamic Sales Force Automation (SFA) solution....24
7.3 Summary....25
Chapter 8....26
Conclusion and Discussion....26
8.1 Introduction....26
8.2 Conclusion....26
8.3 Limitations....26
8.4 Future works....26

8.5 Summary....26
References....27
Appendix....28
Appendix A - Create a tenant....28
Appendix B - Account Creation....28
Appendix C - Multiple Accounts types....29
Appendix D - Account Branch Create....29
Appendix E - Add new Country....30
Appendix F - Company wise document access allocation (Dynamic)....30
Appendix G - Department Creation....31
Appendix H - Employee Creation....31
Appendix I - Assign Employee to an Account (Dynamic)....32
Appendix J - Access Control (Tenant wise)....32
Appendix K - Sales Workflow Indication (Dashboard)....33
Appendix L - Product Category....33
Appendix M - Product Creation.....33
Appendix N - Purchase Requisition....34
Appendix O - Purchase Order (Dynamic - either PR exists or not)....34
Appendix P - Goods Received Note....35
Appendix Q - Sales order....35
Appendix R – Invoice....36
Appendix S - Dispatch out Note....36
Appendix T - Receipt (Cash, Card and Cheque)....36
Appendix U - Cheque realization....37
Appendix V - Vehicle Routes....38
Appendix X - Route Plan....38
Appendix Y - Vehicle Loading....39
Appendix Z - Vehicle Unloading....39
Sales Forecasting Module – Azure Machine Learning....40
Appendix AA - Adding dataset....40
Appendix AB - Add dataset to new project....40
Appendix AC - Training Model (experiment)....41
Appendix AD - Convert to Predictive analysis....41
Appendix AE - Deploy as a Web service.....42
Appendix AF - Generated web service....42

List of Figures

Figure 1: MVC Architecture 13
Figure 2: Project Phases 17
Figure 3: Project flow 17
Figure 4: Top level view 18
Figure 5: Single vs Multi-tenant 20
Figure 6: Proposed system architecture 21

Table of Content

Table 7-1 Evaluation.....25

Chapter 1

Introduction

1.1 Prolegomena

A Sales Force Automation [SFA] system is a computer system that automates and increase efficiency, performance, and reliability of business functionalities such as inventory management, sales, significant information visibility and tracking of customer interactions, as well as analyzing sales forecasts/performance. Businesses may consist of a different kind of versions developed - specifically based on their needs [1].

Existing Sales Force Automation system research would mainly focus on increasing the adoption and use of SFA by sales team but analyzed some important problems such as:

- In the remote areas – Sales officers send their receipts through the courier due to no access to the existing computer system.
- No real-time visibility/transparency of an organization's sales to higher management.
- Salesman does not visit every shop in the area.
- Manual inventory control.
  Merchandising for visibility analysis & audit for availability of the product.

1.2 Problem Statement

When an organization consists of a decrease in Sales, weak customer relationship, improper inventory management, and low sales forecasts, the organization is moving to a Sales Force Automation system. Once a software development company receives this requirement, the system will be developed based on the organization’s requirements.

This is acceptable to the business as mentioned above since the system is developed based on their behavior and requirements analyzing. There are various kind of Sales Force Automation (SFA) systems that exist in the industry. If the organization requires to implement the same SFA system to another business inside the organization, there are so many implementation issues because the initial system is developed based on

their requirements analysis. There are different sales flows among businesses. Thus, when it comes to dynamically process flow changing system, which consists of an industry standard, is not available.

The purpose of this implementation is to analyze multiple business requirements, sales flow with proper sales forecasts and develop a dynamic sales flow based Sales Force Automation (SFA) solution to a complexly diversified organization with rectifying the above problems.

1.3 Aim and Objectives

The aim and objectives will be completed to obtain the solution.

Aim

The Aim of this research is to identify and analyze the existing sales force automation processes, developed systems and resolve the existing implementation problems across multiple businesses through a dynamic sales flow based computerized system which can be accessed anywhere in the globe by providing reliable Artificial Intelligence based sales prediction module for a complexly diversified organization.

Objectives

- Analyze and discuss with internal and external parties on sales processes, existing Sales force automation (SFA) systems in the industry.
- Identify the each and every sales force flows, transport flows, customer management and forecasting modules.
- Design the project, such as suitable development language, database, mobile platform and sales predictive module.
- Develop the proposed solution under software development standards.

1.3 Background and Motivation

Today's businesses, vastly growing and trying to achieve their business goals by getting more profits. They use so many software solutions such as Inventory control systems, Point of Sales systems, financial based systems, Enterprise Resource Planning (ERP) systems, and Sales Force Automation (SFA) systems to keep the efficiency and effectiveness of the businesses. Many businesses are using Sales Force Automation (SFA) systems because of high efficiency, mobility, and less cost than an Enterprises

Resource Planning (ERP) systems. A Sales Force Automation system is a similarity to an Enterprises Resource Planning (ERP) system but consists of some differences.

Typically Sales Force Automation (SFA) systems are developed based on individual behaviors and flow of the business. This approach gets more complicated when implementing one business to another business because of the difference in business workflow. Nowadays technologies are updating day by day. With this condition, an Artificial Intelligence (AI) forecast is more valuable. Also, many of systems are migrating to Cloud solutions with this transformation since they can handle their multiple businesses across the world by a single hand.

For a complexly diversified business, a dynamic sales flow based Sales Force Automation (SFA) system a must because of the variety of their businesses. The proposed system becomes more valued when the system consists of Artificial Intelligence (AI) to forecast the sales figures. Currently, there is no cloud and dynamic flow-based Sales Force Automation (SFA) web system which consists of Artificial Intelligence (AI) forecasting module.

1.4 Problem Definition

In the current industry, most of the existing Sales Force Automation (SFA) systems developed based by analyzing individual business behaviors and flows only. When implementing this kind of systems from one business to another business, there are so many limitations and problems. To avoid such kind of issues needs a dynamic sales flow and cloud-based Sales Force Automation (SFA) system with proper sales forecasting module.

1.4 Proposed Solution

The proposed solution will be developed by considering “Tenant” based modules, and those tenets represent each business. In a tenant has multiple “Accounts” which represent business, suppliers, and customers. Every tenant will be able to define their own business flows according to the standards of the system (based on industry standards).

The proposed solution will be developed based on Analysis, Designing, and phases. In the Analysis phase, there will be having meetings and discussions with multiple

businesses. Thus, expecting to gather much more information and clarifications about the proposed system. Further will be analyzed mobile development, load balancing of the proposed system and more technical terms; with the support of software architects.

1.5 Summary

In this chapter, evaluated the introduction, aim and objectives, background, problem definition, and proposed solution. The next chapter will discuss the Literature review.

Chapter 2

Literature Review

2.1 Introduction

This Chapter covers the Literature reviews of the project by considering Cloud, Mobile technologies and their advantages and disadvantages.

A Sales Force Automation (SFA) system delivers various potential benefits such as increases in business sales effectiveness, efficiency, productivity, and enhanced customer relationship management enormously. Most of the existing Sales Force Automation systems developed on web and mobile-based, and those are highly independent systems. Many research papers expose that most of the existing Sales Force Automation (SFA) systems failed due to low user acceptance, different kind of sales, managerial expectations and lack of managerial support for the system.

Businesses have their system flows which adapted to day to day transactions. Existing Sales Force Automation (SFA) systems are developed based on analyzing individual businesses requirements and flows only. Consequently, a complexly diversified business may have to use customized systems to manage their businesses [1] [2].

Numerous researches focused on overall sales and sales performance increasing instead of implementing one SFA solution to another business. The overall sales can be categorized as follows: [3]

- Sales Volume
- Sales Objectives
- Sales Presentation
- Customer Relations
- Market Knowledge
- Time Management

Also the performance increasing (efficiency) can be categorized as follows:

- Communication
- Data management
- Order accuracy
- Ordering time

2.2 Importance of a Cloud and Mobile based solution

A Sales Force Automation system requires mobile-based modules such as GPS tracking, online invoicing, payment collection, and goods return management. For natural and adoption with the new technology is much required on mobile based solutions [2].

For a complexly diversified organization, a cloud solution is a must. High accessibility, less maintainability, Strong user-based security (High security for hacking/attacks), Load balancing with proper alerts, low cost are expose the benefits of using a cloud solution.

A flexible infrastructure is required for mobile applications with a dynamic workload demand to meet performance with reducing resource costs. Further, cloud computing directs to pay per use technology. Means, cloud computing by using resource allocation algorithms will allocate resource to the user as per user request, and the user needs to pay only for what they use (e.g., electricity) [3].

There are many existing cloud spaces available, such as;

1. Microsoft Azure.
2. Amazon EC2 - virtual IT, IBM Cloud.
3. Google App Engine - application hosting.
4. Google Apps and Microsoft Office Online - software as a service.
5. Apple iCloud - network storage. etc.

When the terms of cloud computing, the multitenant architecture is used for software development. The Multi-Tenant architecture is a single instance which supports for multiple customers. Each customer represents a tenant. There are two tenant architectures called single-tenant and multi-tenant. In cloud computing, the multi-tenancy architecture has extended because of new service models that take advantage

of virtualization and remote access. The commonly used technique is Software as a Service (SaaS). For example, it can provide a system/instance to multiple customers. This leads to reduce cost, share system services and applications and hardware resources because the software development and maintenance costs shared. Additionally, this provides a centralized hosting service, proper client-server management [11].

Below listed the more advantages and disadvantages of cloud solution [4] [5] [6].

2.3 Advantages of cloud computing

1. Improved Disaster Recovery
2. Increased Collaboration and Flexibility
3. Pay-Per-Use
4. Manageability

2.4 Disadvantages of cloud computing

1. Accessibility problems (disconnection of internet)
2. Data security (privacy of important data can be viewed by 3rd party)

2.4 Microsoft Azure Machine Learning

Microsoft Azure Machine Learning incorporates cloud services that allow the creation, deployment, and management of applications by developers through a global network located in large data centers of Microsoft Corporation. Azure Machine Learning provisions multiple ML algorithms associated with regression, classification, and clustering [7].

The users can create their models once trained and converted into a predictive experiment their experiment. It offers a different kind of algorithms with a single clustering algorithm. The Cortana Intelligence Gallery characterizes azure ML. This is a collection of ML solutions created by the community for reuse researched by data scientists [7].

There are two categories:

1. Azure Bot Service
2. Azure Machine

2.6 Machine Learning Algorithms

The different machine language algorithms can be categorized such as;

- Support Vector Machine (SVM)
- Network Regression (NNR)
- Logistic Regression
- Decision Forest

The Support Vector Machine (SVM) algorithm is a managed learning approach used to resolve classification problems. The SVM accepts labeled training data and produces hyperplane which is used to increase the boundary between high-dimensional spaces. The Logistic regression is a statistical linear algorithm used in task classification, and it is usually used to resolve simple problems. This can be used as a prediction model also, and it predicts values by applying "statistical" analysis.

The Decision Forest algorithm is a learning technique that consists of multiple classification methods. It can build and manage decision trees with a different classification. This can perform aggregation and sum histograms to obtain each label's probabilities. The Neutral Network Regression (NNR) algorithm creates a classification model by merging two algorithms;

- Neural Network
- Logistic Regression.

It utilizes a logistic function. As such, its output is similar to that of Logistic Regression. This requires the use of a dataset to test an algorithm[7].

2.5 Importance of Azure Machine Learning

There are many advantages of using Azure ML such as:

1. This can be used as a service - Cloud-based machine learning.
2. A user needs browser only to work with the system (Web-based solution). No need a setup, installation, and maintenance concerns or complications.
3. Several ready to use built-in regression modules.
4. Easy to use. Provides drag and drop canvas user interface for automatically remove aggregating computing modules into an experiment.

5. This is a password-protected integrated development environment which leads ease of access,
6. Ability to publish results of experiments to the web.
7. Ability to re-use the published experiments/components.
8. It is a Low fee (pay per use only) or even free service.

2.6 Summary

In this chapter, discussed the Sales Force Automation (SFA) system, Importance of a Cloud and Mobile-based solution, advantages disadvantages, and Microsoft Azure Machine Learning. The next chapter will discuss the technology adopted.

Chapter 3

Technology Adopted

3.1 Introduction

This chapter will discuss the technologies that can be used to develop the proposed system.

3.2 Microsoft Visual Studio (Programming IDE)

Microsoft Visual Studio is an Integrated Development Environment (IDE) which is produced by Microsoft Corporation. This IDE is used to develop computer-related programs, web sites, and web applications, web services and mobile apps. The Visual Studio produces both native codes and managed code and uses Microsoft software development platforms such as;

- Windows API
- Windows Forms
- Windows Presentation Foundation
- Windows Store
- Microsoft Silverlight.

Visual Studio doesn't support any other programming language, solution or tool basically, but allows use as a plugging of functionality coded as a VS Package. The functionality will be available as a "Service" once the installation process is completed.

Visual studio products

1. Visual Studio 97 (February 1997)
2. Visual Studio 6.0 (June 1998)
3. Visual Studio .NET (February 13, 2002)
4. Visual Studio .NET 2003 (April 24, 2003)
5. Visual Studio 2005 (November 7, 2005)
6. Visual Studio 2008 (November 19, 2007)
7. Visual Studio 2010 (April 12, 2010)
8. Visual Studio 2012 (September 12, 2012)
9. Visual Studio 2013 (October 17, 2013)
10. Visual Studio 2015 (July 20, 2015)
11. Visual Studio 2017(March 7, 2017)

12. Visual Studio 2019 (To be announced)

3.3 Microsoft SQL Server (Database)

Microsoft SQL Server is a relational database management system which is produced by Microsoft Corporation. As a database server, this is a software product which includes the primary function of storing and retrieving data as requested by other software applications. This can be run either on the same computer or on another one through a network (including the Internet).

This supports to create Database tables, views, functions, stored procedures, backup schedules, logins, linked servers, database diagrams, server roles, triggers, policies management and server logs.

SQL Server Architecture

In the SQL Server Architecture, the protocol layer implements the external interface to SQL Server. All operations that can invoke in SQL Server linked through a Microsoft well-defined format. It is called tabular data flow (TDS), and the TDS is an application layer protocol which can be used to transfer data between a database server and a client. Originally this was designed and developed by "Sybase Incorporation." The TDS packages can be incorporated another protocol dependent on physical transport, including TCP / IP, named pipes and shared memory. As an outcome, access to SQL Server is available through these kinds of protocols. Furthermore, the SQL Server API also exposed through web services.

Supportive data types

1. Bigint
2. Int
3. Smallint
4. Bit
5. Decimal
6. Money
7. Numeric
8. Smallmoney
9. Float
10. Datetime
11. Char/ Varchar/ Nchar
12. Nvarchar/Varchar
13. Text

14. Image
15. Table
16. Spatial Geometry Types
17. Spatial Geography Types
18. Sqlvariant
19. Hierarchyid
20. Cursor
21. Xml etc.

3.4 Xamarin (Mobile development)

Xamarin is a Microsoft owned software product which is San Francisco, California-based software company founded in May 2011. The engineers that created Mono, Mono for Android and Mono Touch, which use cross-platform implementations of the Common Language Infrastructure (CLI) and Common Language Specifications (often called Microsoft .NET).

With a C#-shared Cade set, the developers might able use the Xamarin tools to create their native Android, iOS, and Windows applications with native user interfaces and also share their software codes through the multiple platforms including Windows and Mac OS. According to the Xamarin, over 1.4 million developers were using Xamarin's products in 120 countries around the world as of April 2017.

Xamarin platform

The Xamarin programming language introduced in February, 2003 as a version called 2.0. Both Xamarin Android and Xamarin iOS that enables the possibility of developing software tools such as;

- native Android
- iOS
- Windows development in C#.

The developers can be re-used their existing C# codes and shared important code sets through the device platforms. Also, Xamarin combines Visual Studio, Microsoft's IDE for the .NET Framework and extending VS for both Android and iOS development.

3.5 Proposed system's Development Architecture

Model-View-Controller (MVC)

Model-view-controller (MVC) is an architectural pattern frequently used to develop user interfaces that divides an application into three interconnected parts. This is completed to decompose the internal illustrations of the information from the ways of information is presented to and accepted from the user. The MVC design pattern deallocates these major components allowing for efficient code reuse and similar development.

Usually, this is used for desktop graphical user interfaces (GUIs). This architecture has become more popular such as

- Designing web applications
- Designing mobile applications
- Desktop and other clients.

The most popular programming languages like Java, C#, Ruby, PHP have MVC frameworks that are used in web application development straight out of the box. Figure 1.

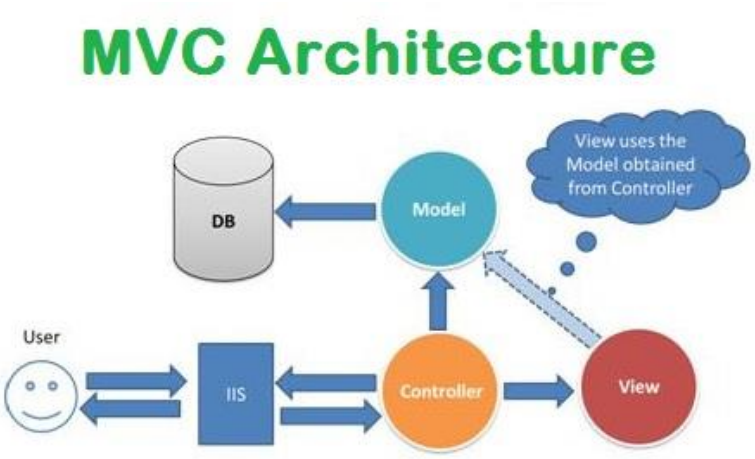


Figure 1: MVC Architecture

3.6 Summary

In this chapter discussed about the technology used for the proposed system, Microsoft Visual Studio, Microsoft SQL Server, Xamarin and MVC Architecture. The next chapter will discuss the Methodology of the proposed project.

Chapter 4

Methodology

4.1 Introduction

In this chapter, describes the Methodology of the proposed system.

4.2 Methodology of the Proposed System

The proposed system will be developed by **C#** as the programming language, The **SQL Server** for database management system where located in **Microsoft Azure Cloud** basis and "**Xamarin**" will be Mobile development (cross-platform support). The system will be developed based on the "**MVC**" design pattern. The sales prediction module will be developed by using **Microsoft Azure Machine Learning**.

4.2.1 Dynamic Sales Flow

Based on the above declaration, the proposed system mainly focused on dynamic sales flow based module. In this module identify each sales processes and defined as the initial module is "Sales Order" and the final is "Un-loading" module. Every module should exist in that boundary. Identified sales flows are below:

Sales order → Invoice → Dispatch → Loading → Unloading

Sales order → Dispatch → Invoice → Loading → Unloading

Sales order → Dispatch → Loading → Invoice → Unloading

A user needs to add tenant wise sales flow in the system. According to the mapping, the system identifies the sales flow and map the relevant documents and the modules. As an example, company "A" sales flow defined as a Sales order, Invoice Dispatch, Loading, and Unloading. Initially, the user needs to create a sales order. The invoice will be mapped with the sales order because company A's sales flow is defined Sales order and Invoice. The particular module references are also mapped based on the defined master sales workflow.

4.2.2 Sales Prediction

In this module, initially, need to upload a sales history data sheet (CSV) to Microsoft Azure machine learning portal's workspace. After that, the uploaded data model needs to be trained with azure boosted decision tree model and convert to prediction module. After the prediction conversion, the model is ready to forecast. The prediction module needs to deploy as a web service in the same portal, and finally, it can be used once this completed as a web reference. The experiment can be re-trained at any time and use as usual.

4.3 Summary

In this Chapter, evaluated the Methodology of proposed Sales Force Automation system. The next chapter will discuss the Analysis and Design of the proposed system.

Chapter 5

Analysis and Design

5.1 Introduction

In this Chapter will discuss the proposed system's analysis and design methods in detail before the development.

5.2 Project flow

The proposed solution will be developed based on Analysis, Designing, and Development (Implement) phases. Figure 2. For the Analysis phase, meetings and discussions with multiple businesses should require to analyze the existing businesses flows. Hence, gathering further information and clarifications about the proposed system. Further will analyze the mobile development, load balancing of the proposed system and more technical terms; with the assistance of software architects and internet.

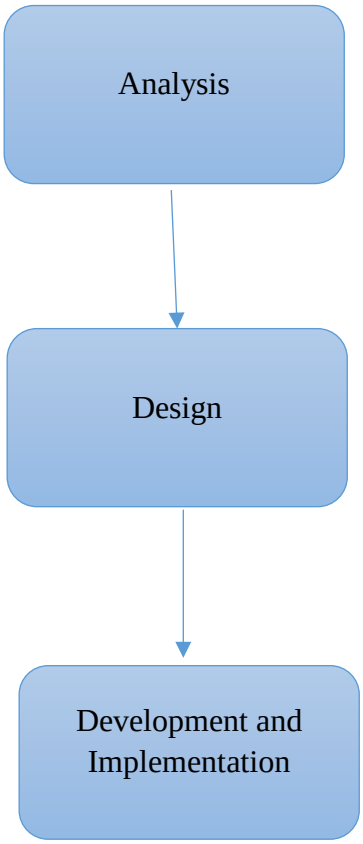


Figure 2: Project Phases

5.3 Top level design view of the system.

Based on the below design view, the System Administrator, Business (Tenant), Accounts (Supplier, Distributor, Dealer and Customer) are connected to the system. Figure 4

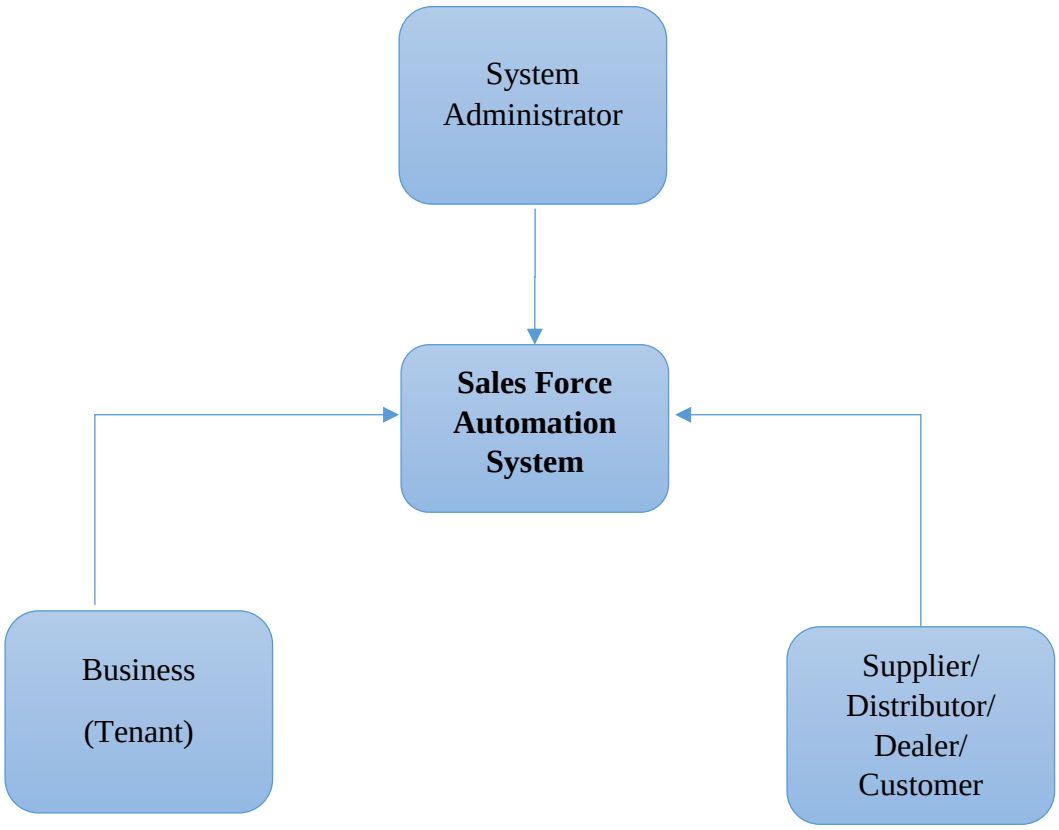


Figure 4: Top level view

5.4 Multi-Tenant Architecture

Tenant is a single group of users/customer sharing the same kind of view in the system. Multi-tenancy is the best approach to share an application instance among multiple tenants. These tenants are providing each tenant as an allocated “share” of the instance, which is isolated from other shares concerning performance and data privacy [14].

Multi-tenancy directs to enhance efficiency, improved scalability and reduced costs. With the recent growth of Cloud Computing and Software-as-a-Service (SaaS) in specific, a flexible and scalable multi-tenant architecture is becoming highly essential.

In multi-tenant applications, each tenant represents its customers, users, and administrators and even lean towards to be divided into multiple subtenants (Accounts). As the number of tenants, users and amount of data growths, consequently a scalable architecture for the access control system is needed [12].

5.4.1 Benefits of using multitenant Architecture

1. A portion of multiple clients or tenants by a single application instance with an isolation of each tenant’s data.
2. Directs to Increased utilization of available hardware resources.
3. Provide improved ease of maintenance and deployment flexibility.
4. Overall application low costs.
5. Can pass the development and deployment costs to the customer.
6. Increasing competitive advantages.
7. Optimized Efficiency in Performance
8. Easy Customizations

5.4 Single Tenant vs Multiple Tenant

Single Tenant

The Single tenet is a single instance of the software and providing infrastructure service as a single customer. Through the single tenancy, each tenant consists of their independent database and instance of the software. Essentially, no distribution/sharing is happening with this option.

Limitations of Single Tenancy

1. usually more costly
2. Requires more maintenance
3. Resource utilization isn't always maximized

Multi-Tenant

The Multi-tenancy is a single instance of a software and it's providing infrastructure service for multiple customers. Each customer shares/distributes the software application and also shares a single database. Each tenant's data is isolated and remains invisible to other tenants.

The Figure 5 expresses that the Single-tenant vs Multi-Tenant more clearly.

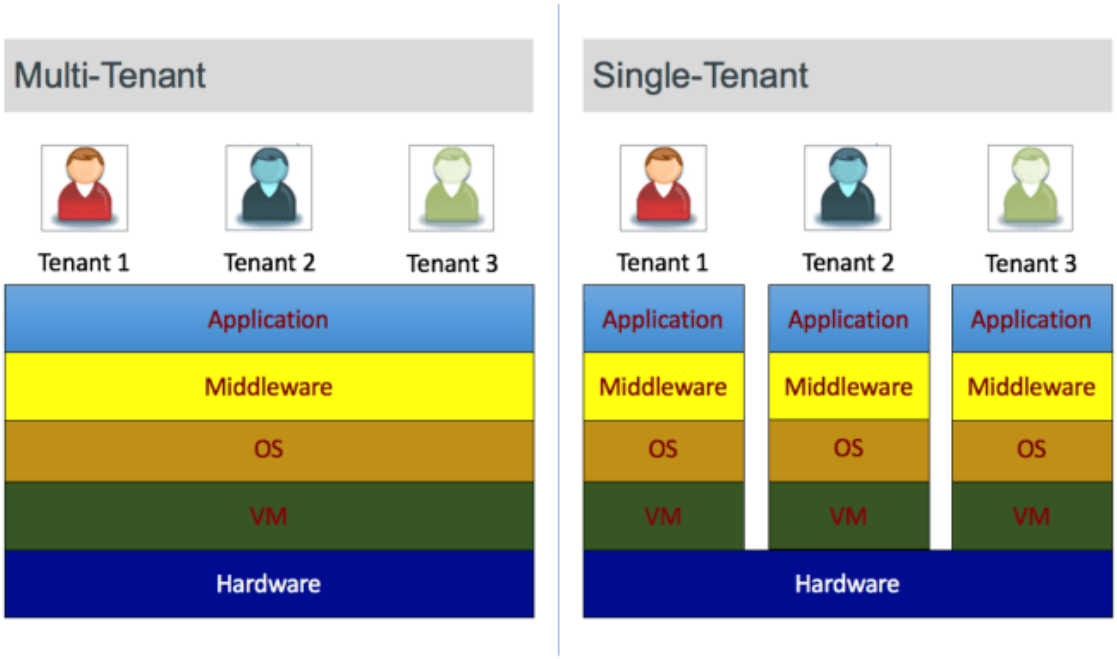


Figure 5: Single vs Multi-tenant

5.5 Proposed System's Multi-Tenant Architecture

In the proposed system, a System administrator creates a Tenant and afterward the tenant administrator can create multiple accounts in the proposed system. Figure 6. Each accounts represent business entities such as Distributor, Dealer, Supplier, and Customer.

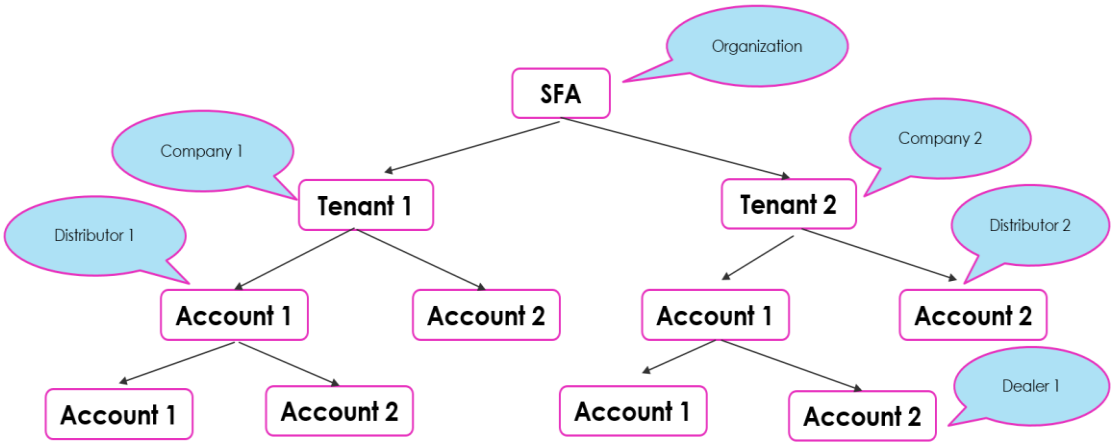


Figure 6: Proposed system architecture

5.6 Summary

In this chapter, discussed the proposed project flow, overview of top-level design, single and multi-tenant architectures and proposed system's architecture. The next chapter will discuss the Implementation of the project.

Chapter 6

Implementation

6.1 Introduction

In this chapter, describes the Implementation of the proposed system.

6.2 Requirement Gathering and Analysis.

In this analysis, gathering requirements from different types of businesses such as GAS, Lubricant, and Sales, etc. and identified their actual sales flow and behaviors of the businesses.

Requirements gathered using:

1. Interviews
2. Discussions.
3. Internet

Focused modules:

1. Purchase requisition, Purchase order and Goods receive
2. Sales order
3. Invoice
4. Vehicle load/ unload
5. Dispatch
6. Sales Return
7. Supplier Return
8. Payment Collection
9. Cheque realization
10. Debtor management
11. Mobile Accessibility (GPS)
12. Sales Route plan
13. Sales Targets
14. Good Issue

6.3 Design

Compared and analyzed the single vs. multi-tenant architectures and identified as the Multi-Tenant architecture is best for a dynamic sales flow-based system. Designed the development of the project as follows:

Development Language	: **C#**
Database	: **SQL Sever**
Development Architecture	: **Multi-Tenant**
Design Pattern	: **MVC**
Host	: **Microsoft Azure clouds**
Forecasting Module	: **Microsoft Azure Machine Learning**

6.4 Development

The proposed Sales Force web application system will be developed based on the design including Sales Ordering, Invoicing, Sales Returns collecting and Payments collection and Sales forecast module.

6.6 Summary

In this chapter, discussed about Implementation including briefing of project phases and images of developed Dynamic Sales Force Automation (SFA) system and Sales forecasting module (Microsoft Azure Machine Learning). The next chapter will discuss the Evaluation of the project.

Chapter 7

Evaluation

7.1 Introduction

In this chapter, evaluate the entire project phase and development.

7.2 Evaluation of Dynamic Sales Force Automation (SFA) solution

The research problem is identified as existing Sales Force Automation (SFA) system's implementation problems among diversified organizations. To rectify these issues, the proposed system will be developed as explained in above chapters. Further, in this project - analyzed the existing Sales Force Automation systems, Sales and Inventory systems and identified the problem where exists in the industry. Later, analyzed many research papers, journals and internet sources to improve the knowledge regarding each and every area.

In the analysis phase, considered several requirements and business behaviors with many discussions and findings. In the discussions, had many interviews with both internal and external business parties to identify sales related business flows. Also identified that the traditional sales prediction methods are less effective and leads to lower performances.

In the design phase, the project was designed to develop by using C# language, a suitable database is Microsoft SQL Server (Azure-Cloud), and sales prediction module is Microsoft Azure Machine Learning. For future developments, identified the Xamarin is the proper mobile development (Cross-platform supports).

In the development phase, developed the project as a cloud-based web application. Sales forecasting module is created in the Microsoft Azure portal and finally integrated with the system.

The importance of this project is the dynamical sales flow change and Artificial Intelligence based effective sales prediction. With this solution, sort out enormous implementing issues of SFA system in a complexly diversified organization.

Refer the Table 7-1 evaluation table.

Existing Problem	Proposed Solution	Outcome
Implementation problems among businesses	Developed a dynamic business wise flows	Dynamic flow for business wise
Contains traditional sales forecasting modules	Introduced Artificial Intelligence module	More reliable than traditional modules
Mobile applications are not running on every device	Introduced Xamarin	Provisions to cross-platform support (both Android and iOS)
Difficulty of manage the IT infrastructure	Introduced Azure Cloud service	Not required to consider infrastructure and able to use pay as service

Table 7-1 Evaluation

7.3 Summary

In this chapter, discussed and evaluated the proposed system and next chapter will be the Conclusion and Discussion.

Chapter 8

Conclusion and Discussion

8.1 Introduction

In this chapter, evaluate the conclusion and discussion of the proposed Sales Force Automation System.

8.2 Conclusion

Overall achievement of the A Sales Force Automation [SFA] Solution for a complexly diversified Organization is completed and successful. With this solution, can improve extra knowledge about the individual business behaviors, standardization and deliver a proper solution. Also can focus on the latest technologies, adoptions, trends, and services. Additionally, can get a good experience of huge implementation and challenge of this kind of project.

As a conclusion, after spending many months, the dynamic Sales Force Automation (SFA) system is completed as planned in a proper manner.

8.3 Limitations

When the data model is training, required to train and predictive the model by manually if there is a newly added product to the system.

8.4 Future works

In this project, not developed the mobile application because currently there are so many applications exists. But in this study, directed to Xamarin, Offline facility (when no-Internet) and GPS tracking. Any researcher can go through this research project and improve their knowledge with this dynamic flow and focus on Mobile development.

8.5 Summary

In this chapter, discussed and evaluated the overall research project. This is the final chapter.

References

[1] Z. Nasir, “Sales Force Automation Using Mobile Technology,” pp. 389–394.

[2] E. Scornavacca and A. Sutherland, “The perceived impact of mobile sales force automation systems on salespeople’s performance,” *Proc. - 7th Int. Conf. Mob. Business, ICMB 2008, Creat. Converg.*, pp. 270–279, 2008.

[3] M. Drozdova, S. Rusnak, P. Segec, J. Uramova, and M. Moravcik, “Contribution to cloud computing security architecture,” *2017 15th Int. Conf. Emerg. eLearning Technol. Appl.*, pp. 1–6, 2017.

[4] N. F. Efozia, E. Ariwa, D. C. Asogwa, O. Awonusi, and S. O. Anigbogu, “A review of threats and vulnerabilities to cloud computing existence,” *7th Int. Conf. Innov. Comput. Technol. INTECH 2017*, pp. 197–204, 2017.

[5] Y. Lu, G. Zhang, and P. Peng, “The research on efficiency mode of scientific and technological innovation system of three provinces in northeast China,” *2009 Int. Conf. Inf. Manag. Innov. Manag. Ind. Eng. ICIII 2009*, vol. 2, pp. 215–219, 2009.

[6] www.smallbusinesscomputing.com/biztools/the-pros-and-cons-of-cloud.html

[7] O. Harfoushi, D. Hasan, and R. Obiedat, “Sentiment Analysis Algorithms through Azure Machine Learning : Analysis and Comparison,” vol. 12, no. 7, pp. 49–58, 2018.

[8] A. Botchkarev, “Evaluating Hospital Case Cost Prediction Models Using Azure Machine Learning Studio,” pp. 1–28, 2018.

[9] S. Chen, Y. Shen, and Y. Zhu, “Modeling Conceptual Characteristics of Virtual Machines for CPU Utilization Prediction.”

[10] M. Brundage, J. Clark, G. C. Allen, C. Flynn, S. Farquhar, R. Crootof, and J. Bryson, “The Malicious Use of Artificial Intelligence : Forecasting , Prevention , and Mitigation,” no. February 2018.

[11] J. Kabbedijk, M. Pors, S. Jansen, and S. Brinkkemper, “Multi-tenant Architecture Comparison,” pp. 202–203, 2014.

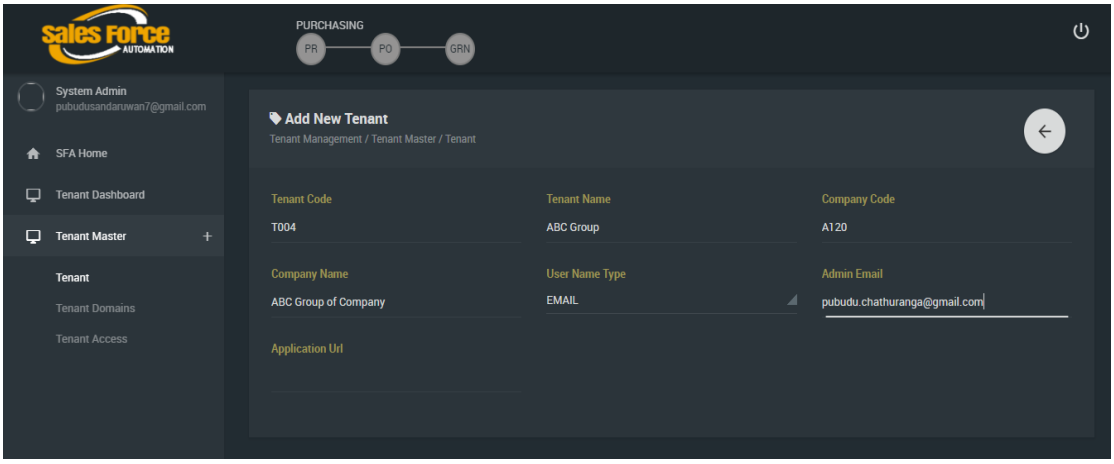
[12] P. Maenhaut, H. Moens, M. Decat, and J. Bogaerts, “Characterizing the Performance of Tenant Data Management in Multi-Tenant Cloud Authorization Systems.”

[13] J. Schroeter, S. Cech, S. Götz, C. Wilke, and U. Aßmann, “Towards Modeling a Variable Architecture for.”

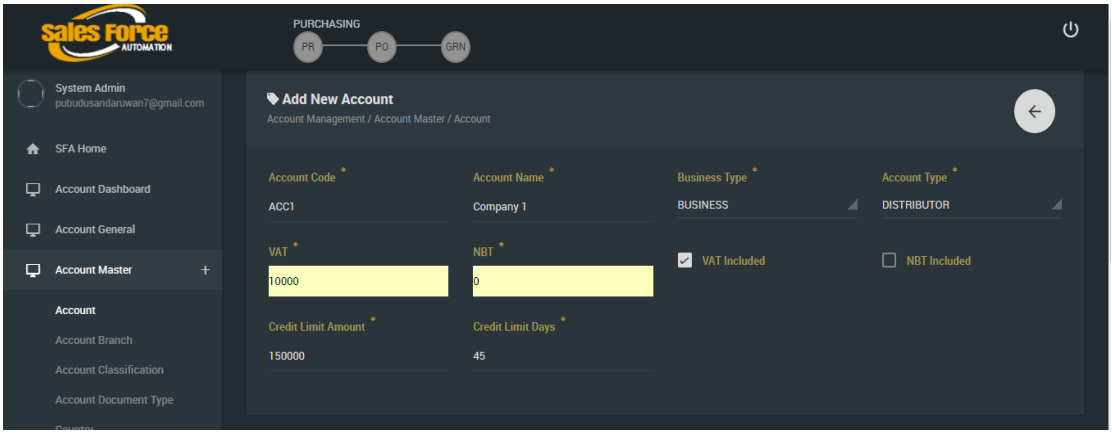
[14] R. Krebs, C. Momm, and S. Kounev, “Architectural Concerns in Multi-Tenant SaaS Applications,” 2009.

Appendix

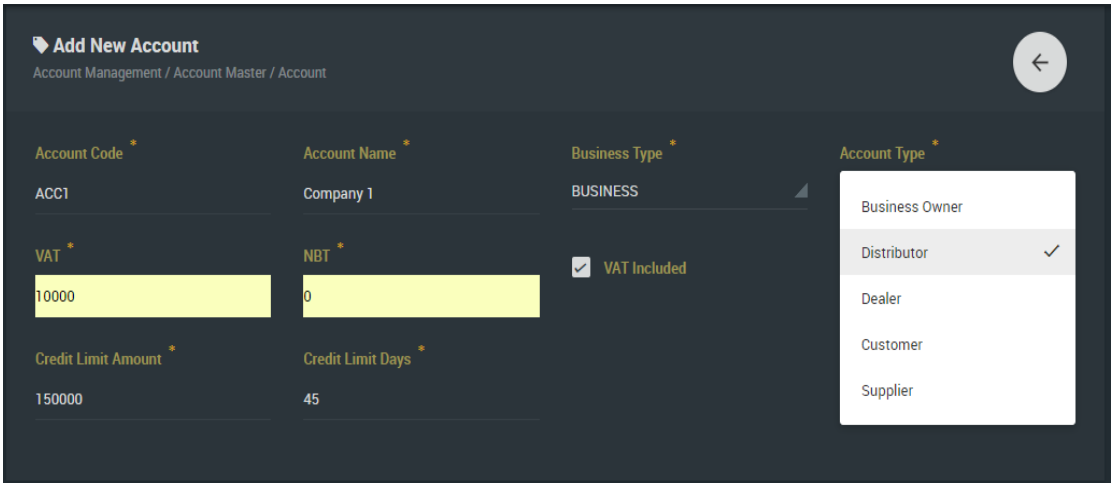
Appendix A - Create a tenant



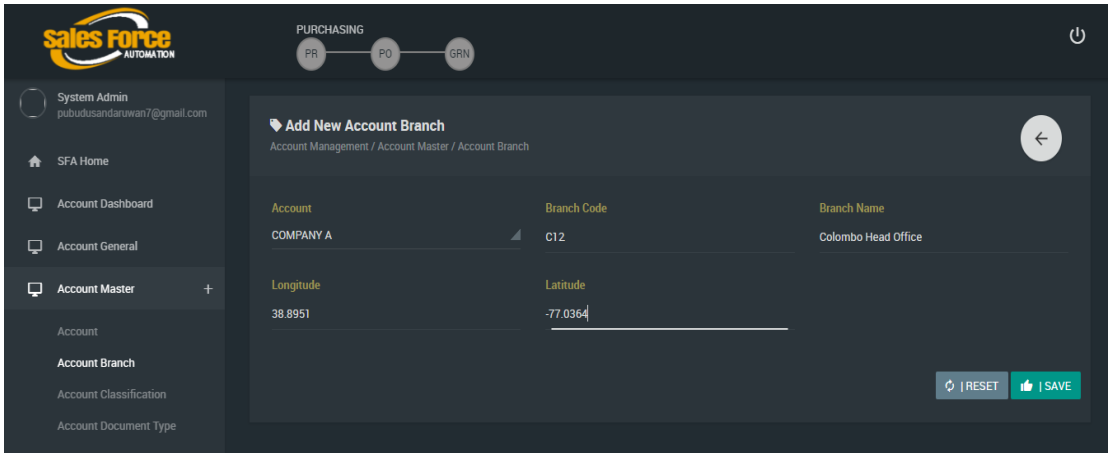
Appendix B - Account Creation



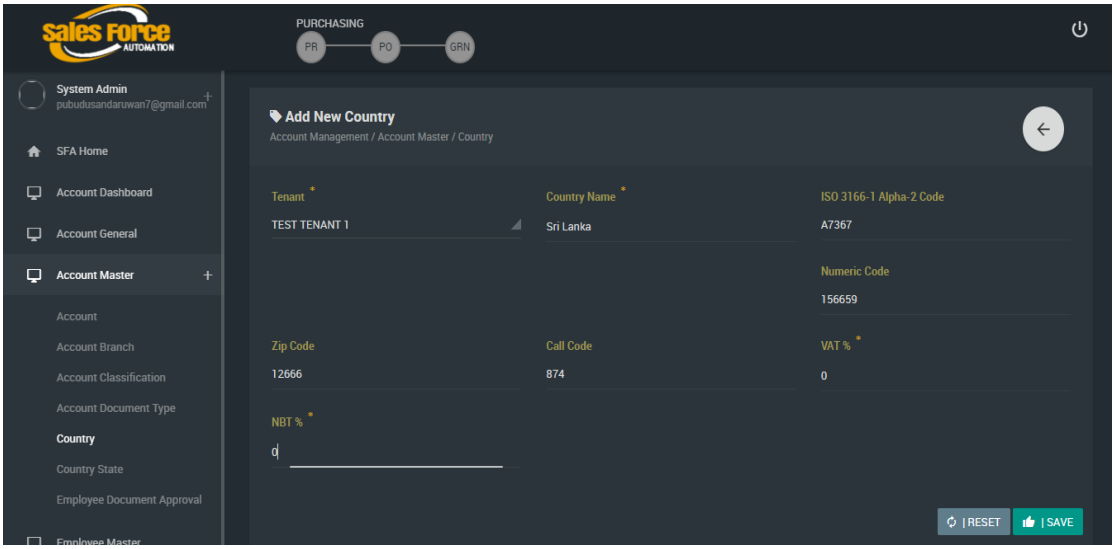
Appendix C - Multiple Accounts types



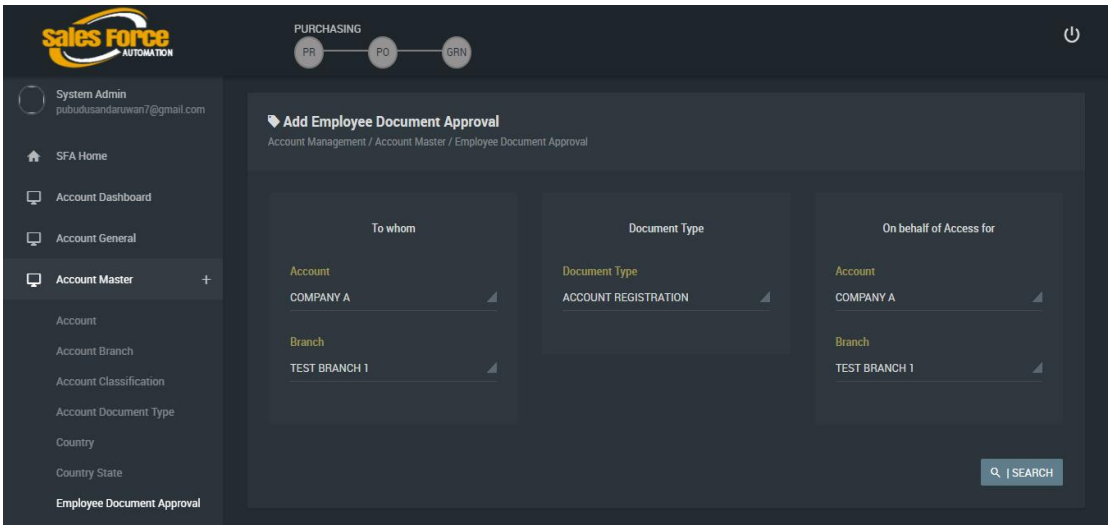
Appendix D - Account Branch Create



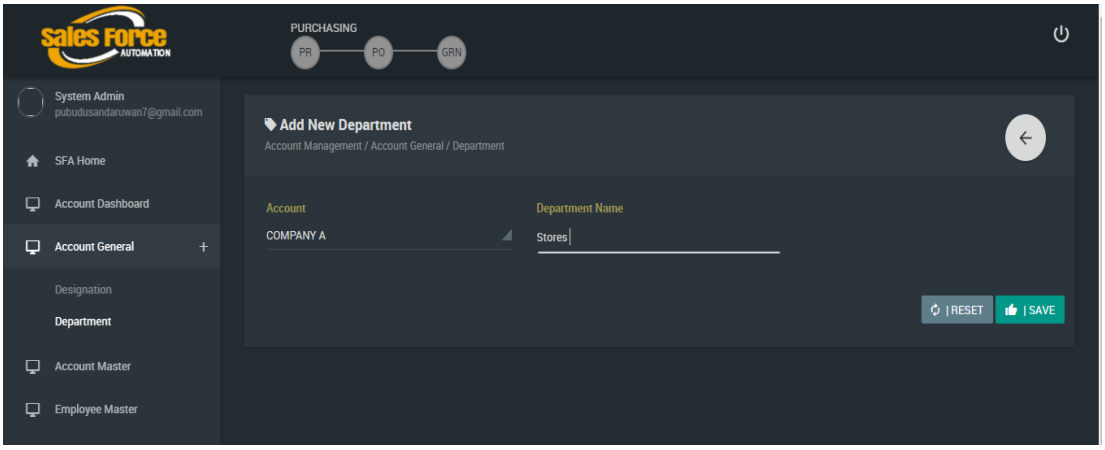
Appendix E - Add new Country



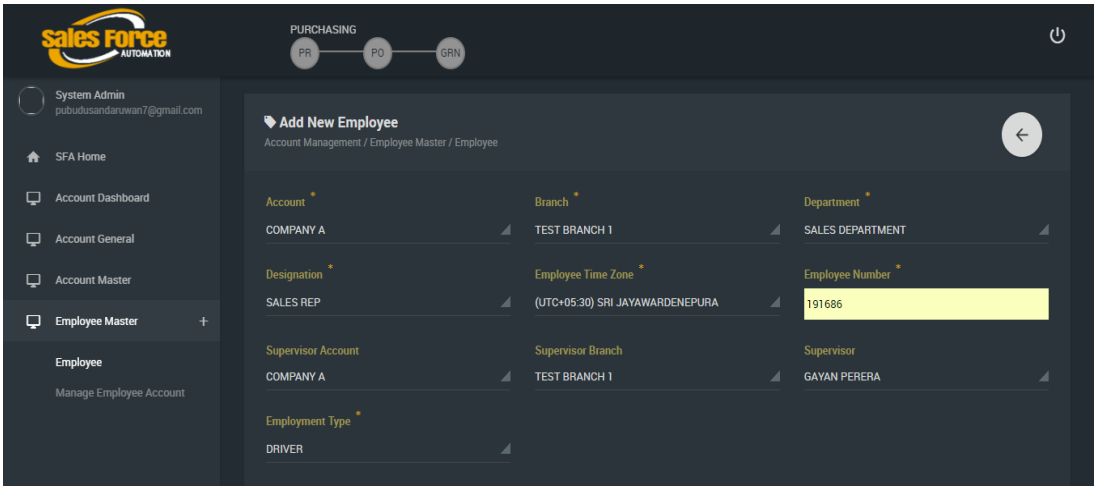
Appendix F - Company wise document access allocation (Dynamic)



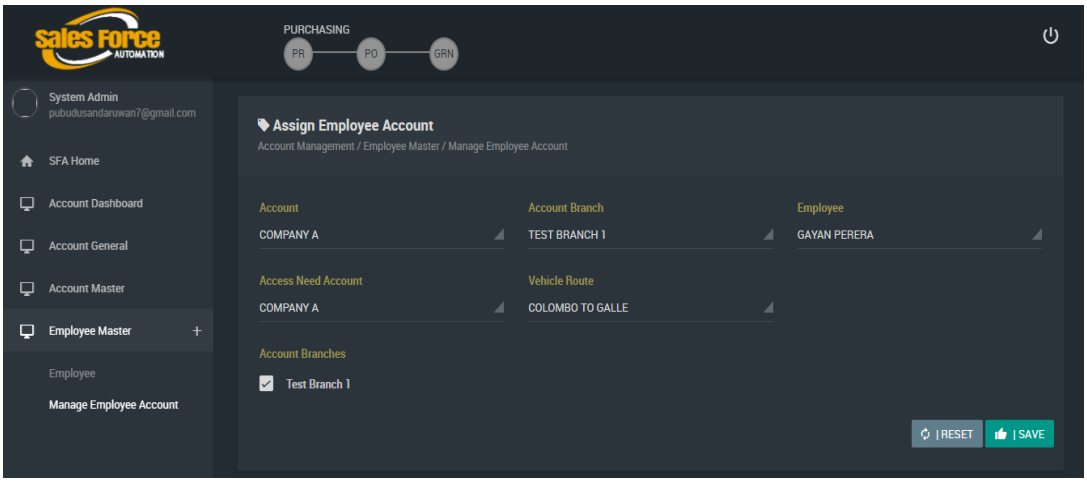
Appendix G - Department Creation



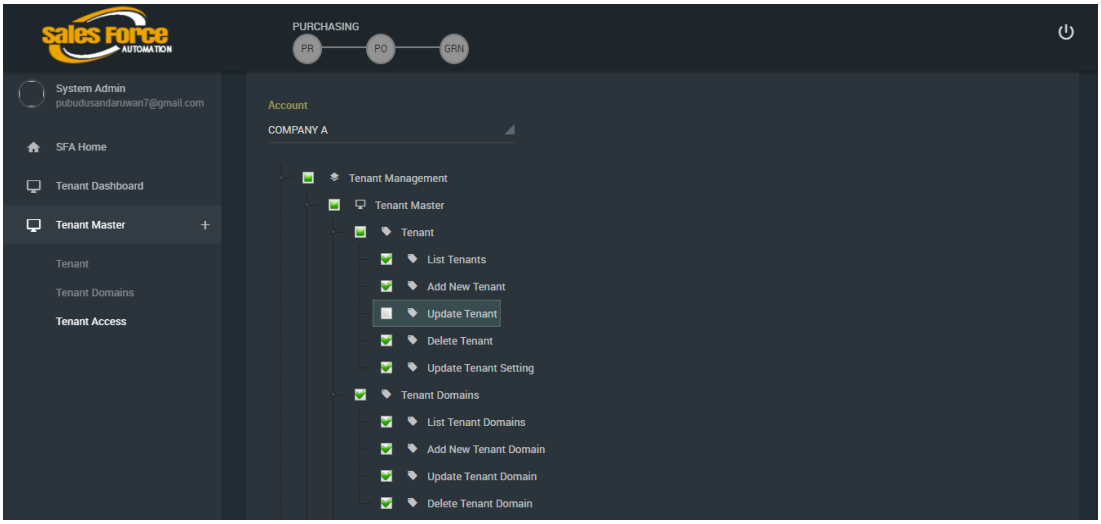
Appendix H - Employee Creation



Appendix I - Assign Employee to an Account (Dynamic)

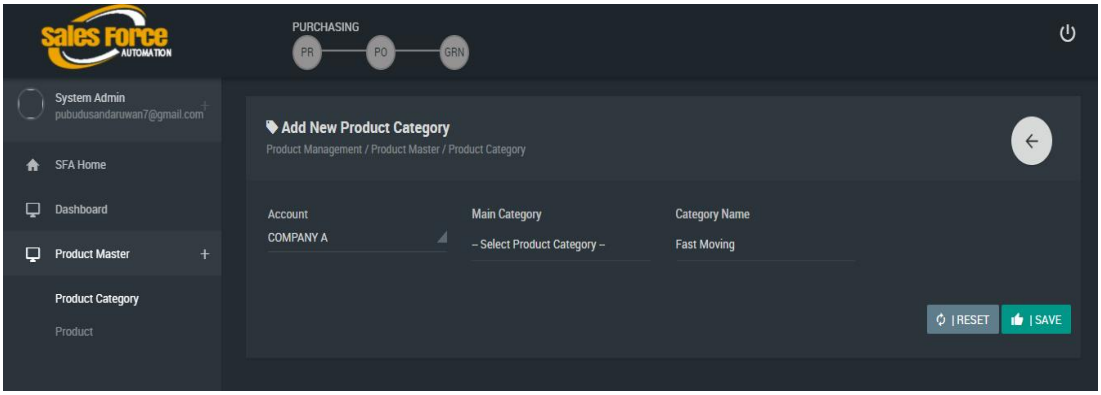


Appendix J - Access Control (Tenant wise)

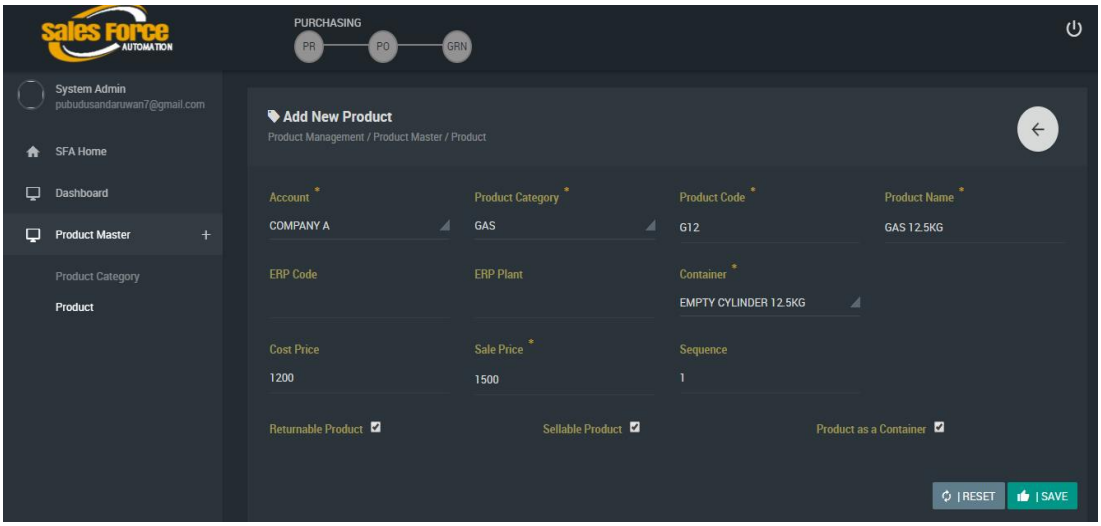


Appendix K - Sales Workflow Indication (Dashboard)

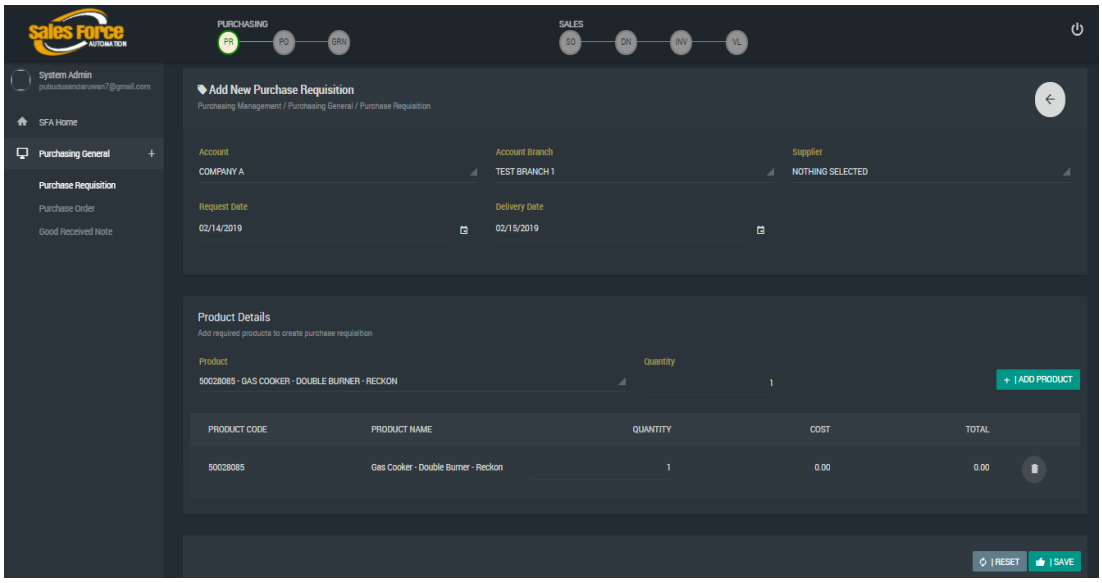
Appendix L - Product Category



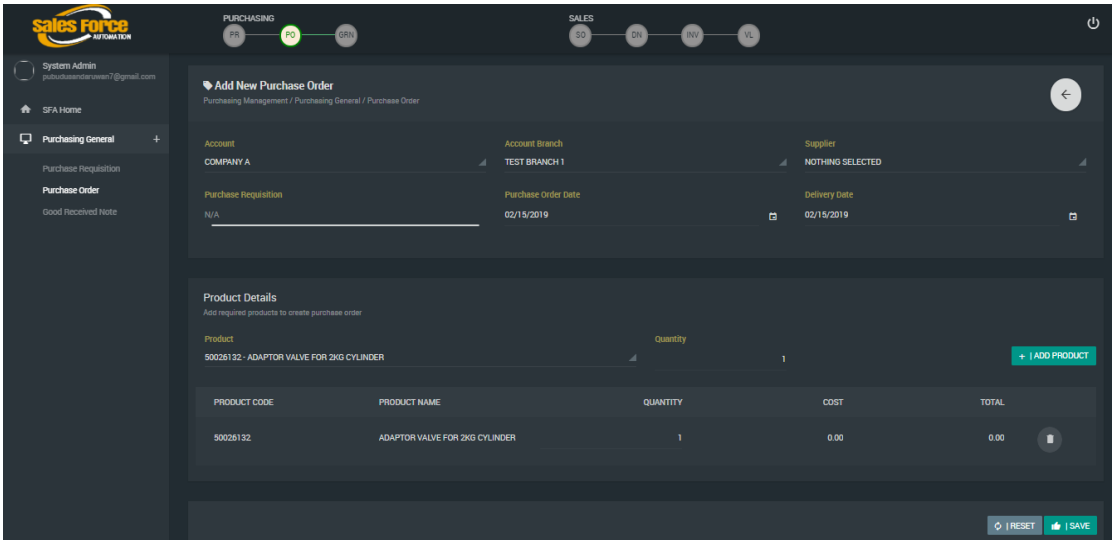
Appendix M - Product Creation.



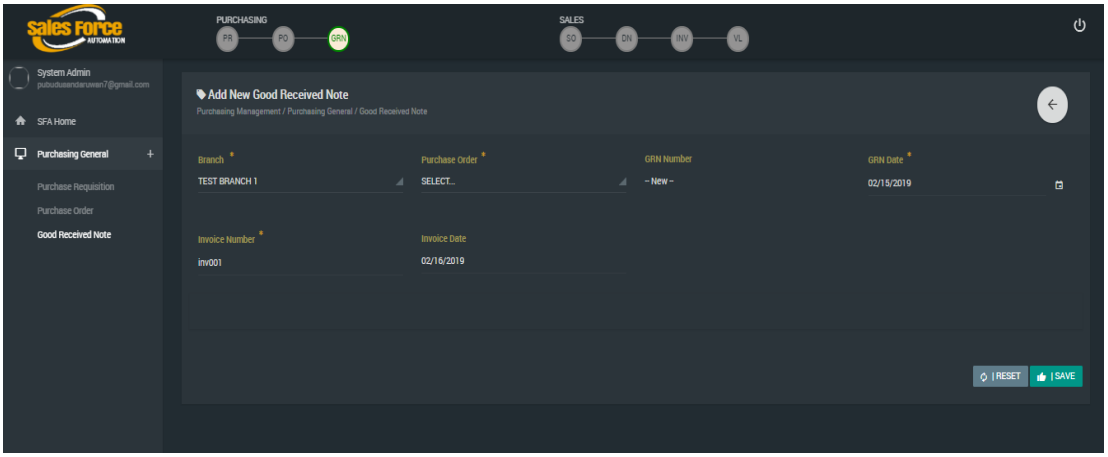
Appendix N - Purchase Requisition



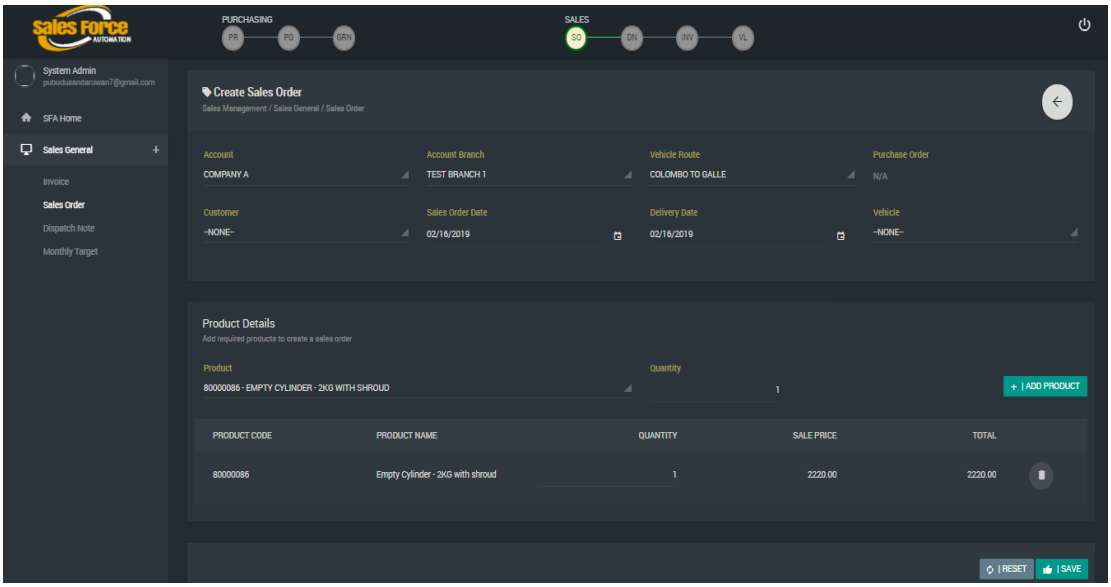
Appendix O - Purchase Order (Dynamic - either PR exists or not)



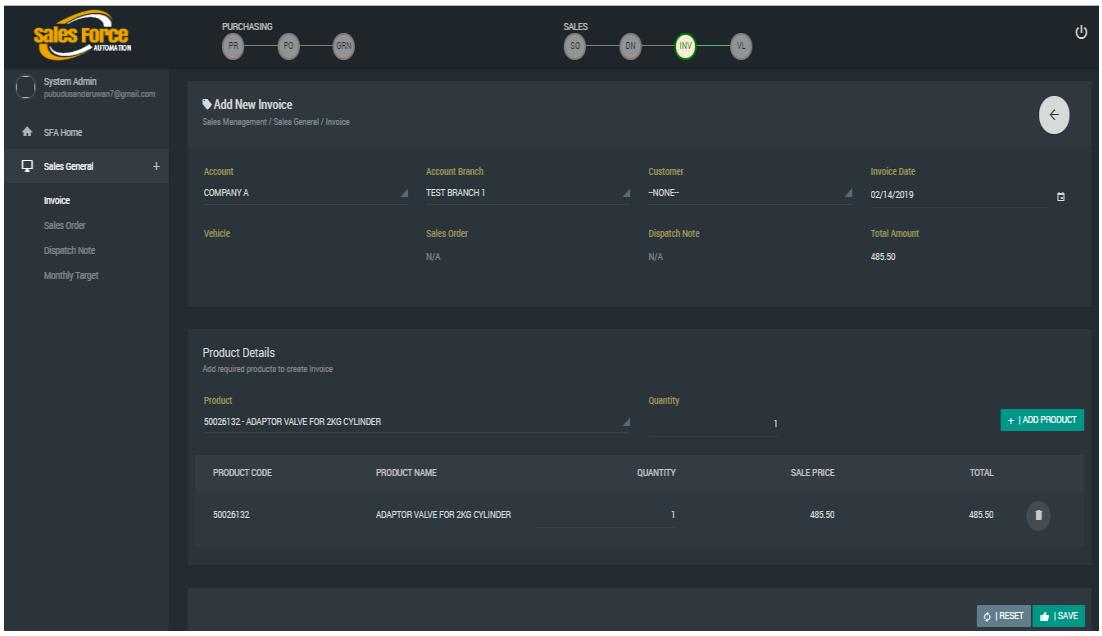
Appendix P - Goods Received Note



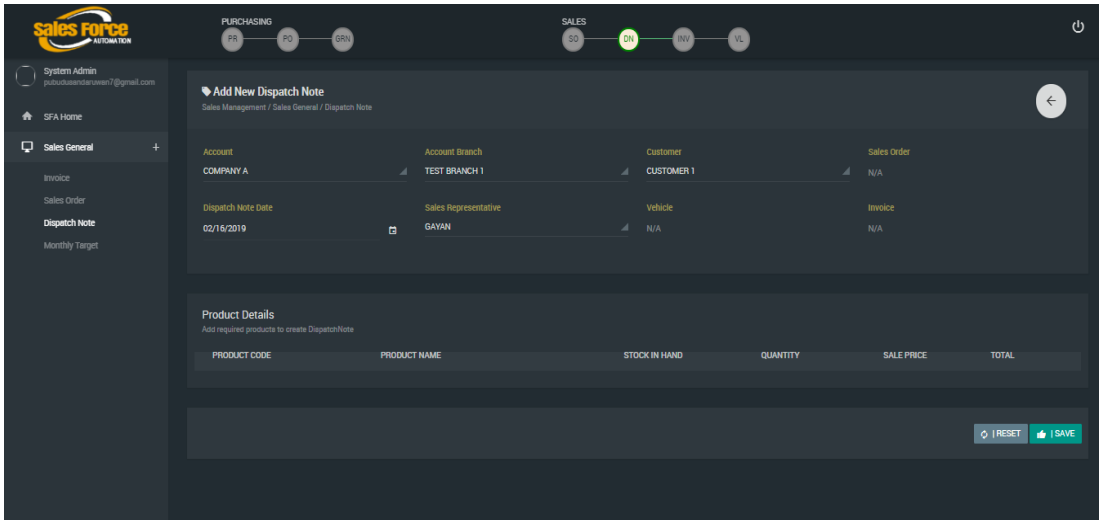
Appendix Q - Sales order



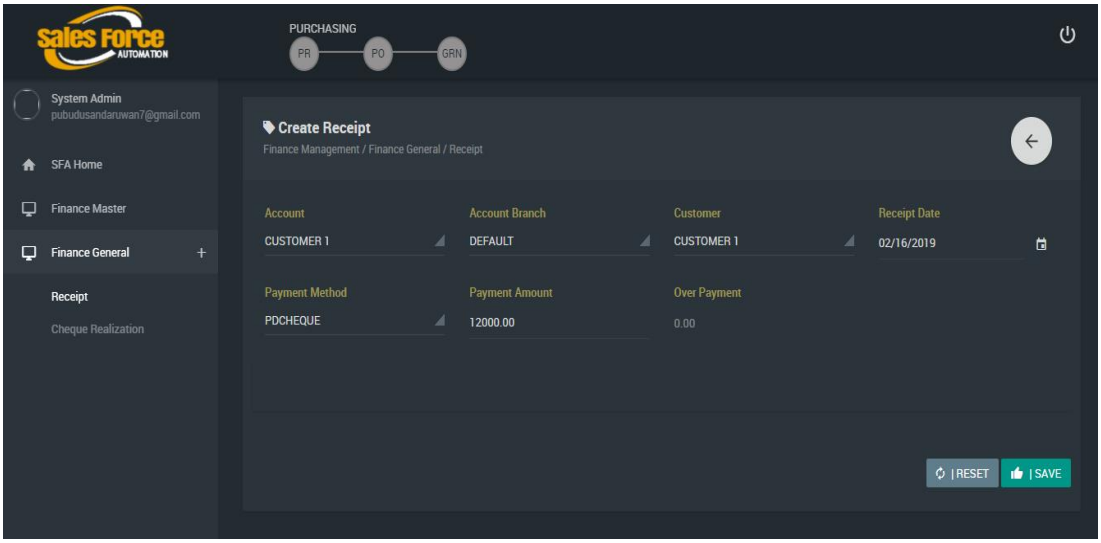
Appendix R – Invoice



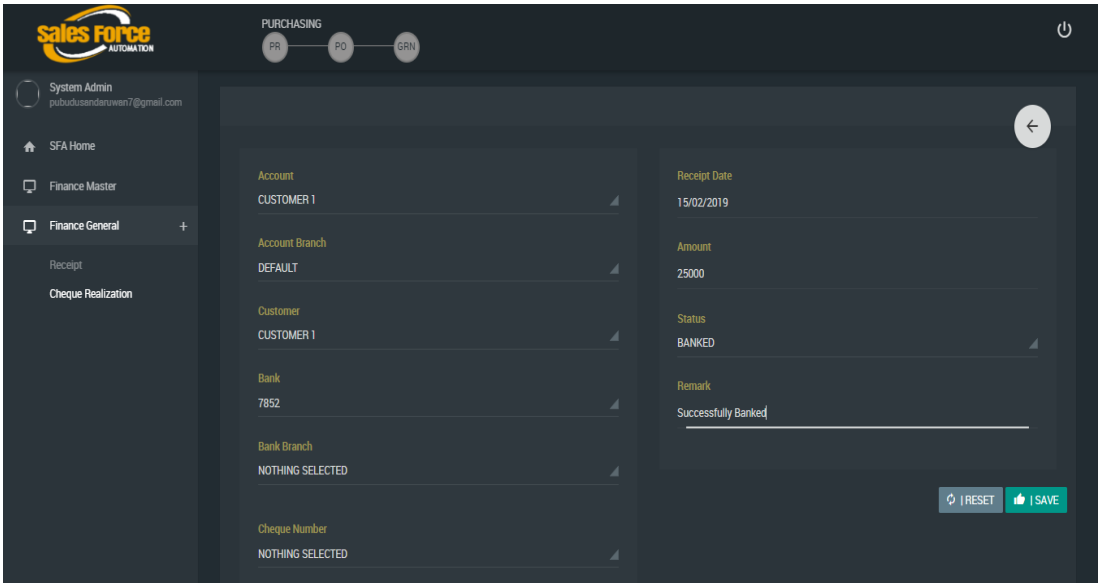
Appendix S - Dispatch out Note



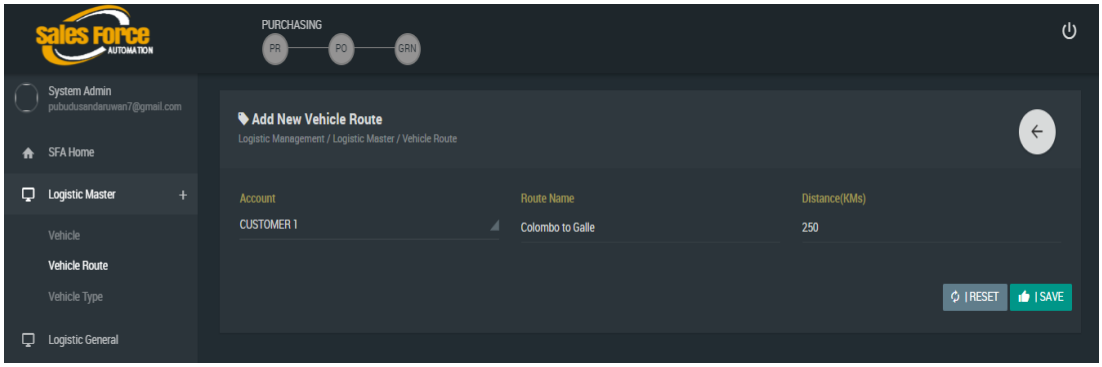
Appendix T - Receipt (Cash, Card and Cheque)



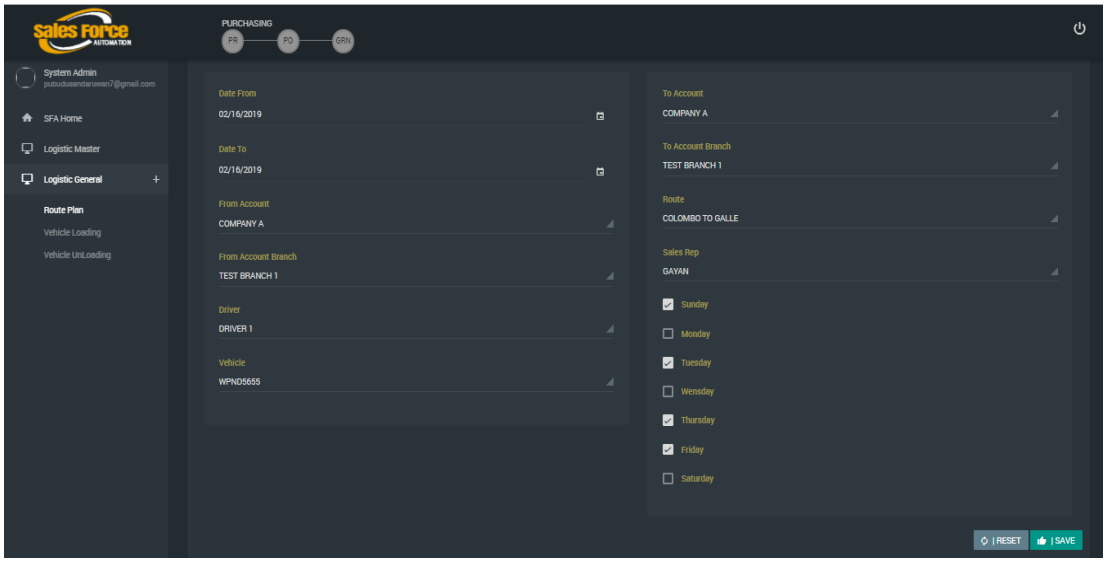
Appendix U - Cheque realization



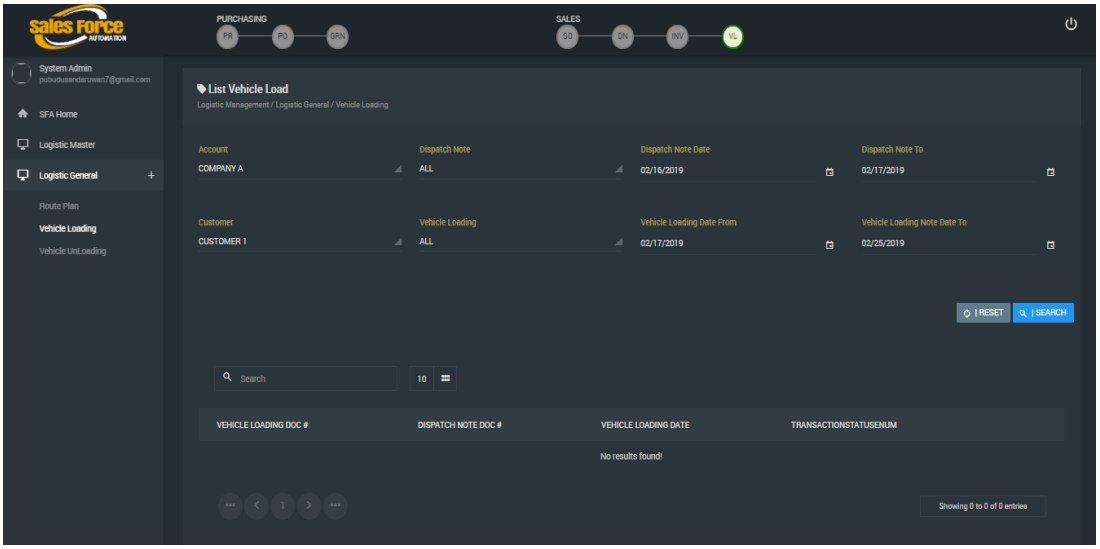
Appendix V - Vehicle Routes



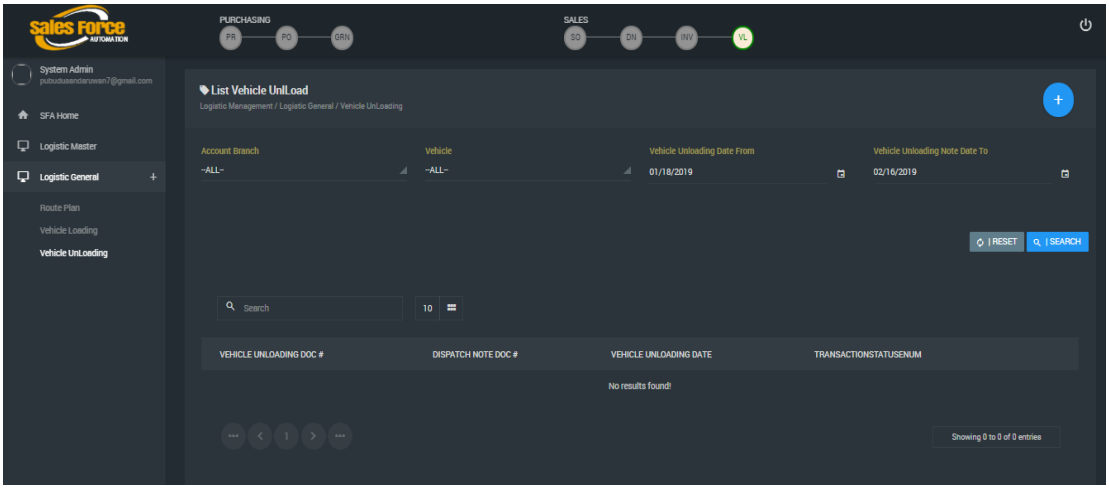
Appendix X - Route Plan



Appendix Y - Vehicle Loading

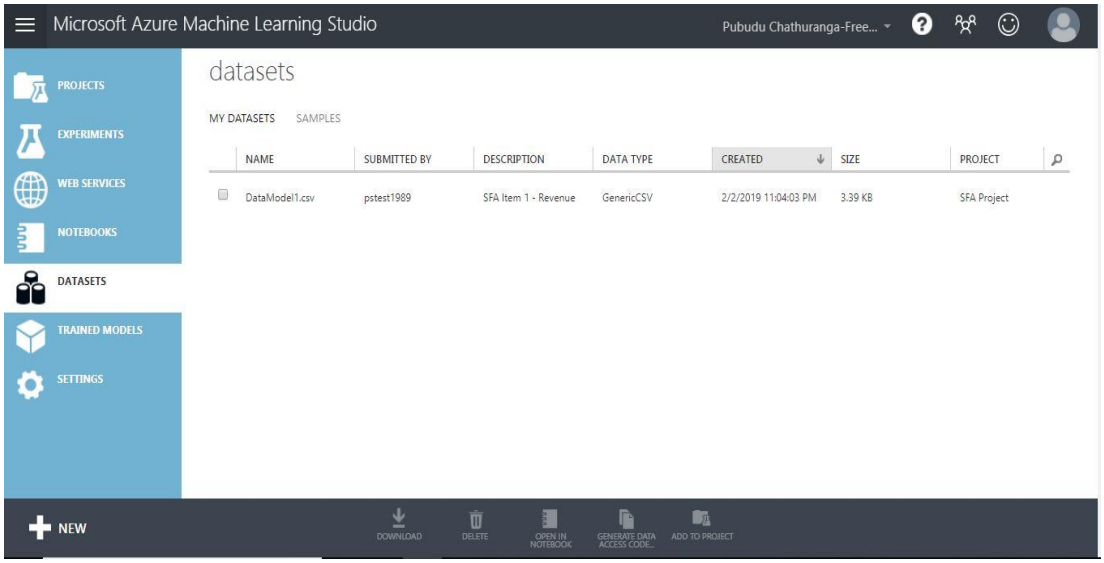


Appendix Z - Vehicle Unloading

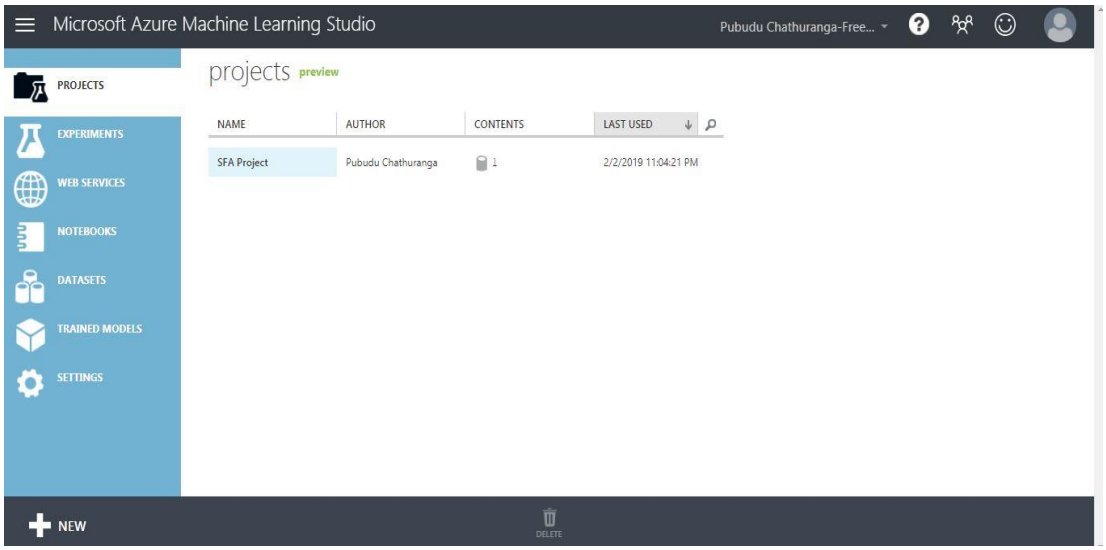


Sales Forecasting Module – Azure Machine Learning

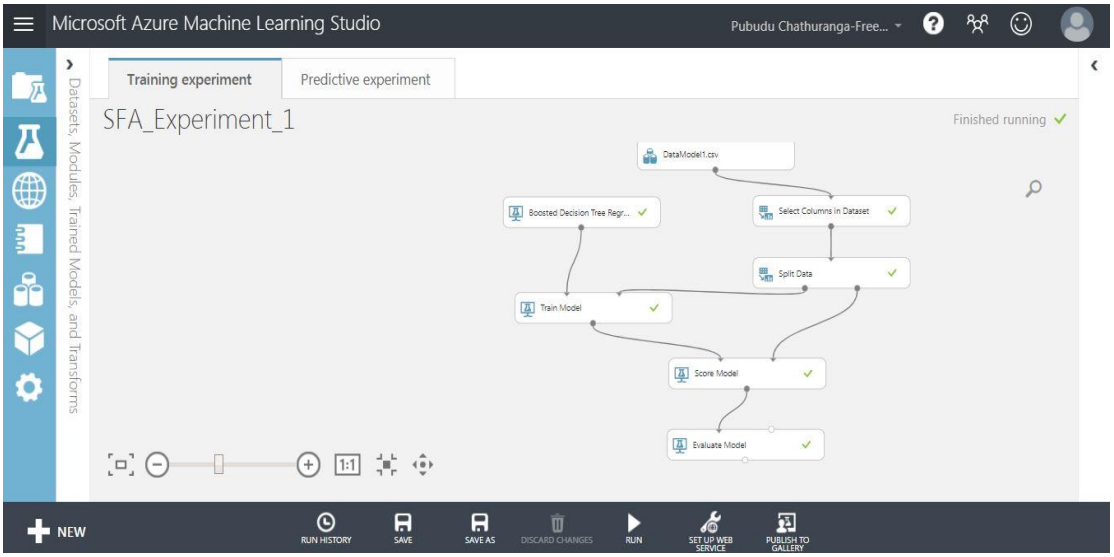
Appendix AA - Adding dataset



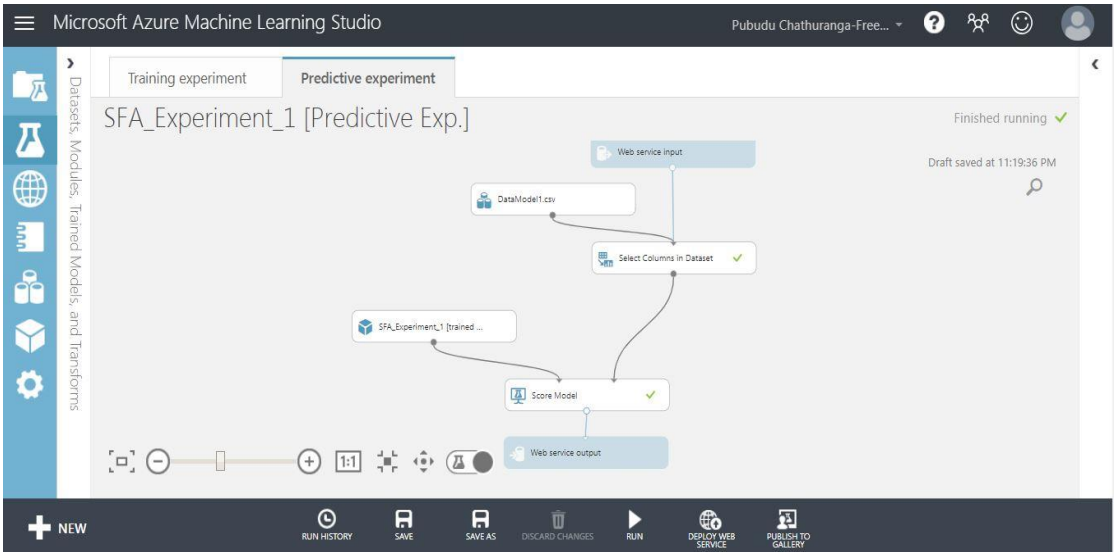
Appendix AB - Add dataset to new project



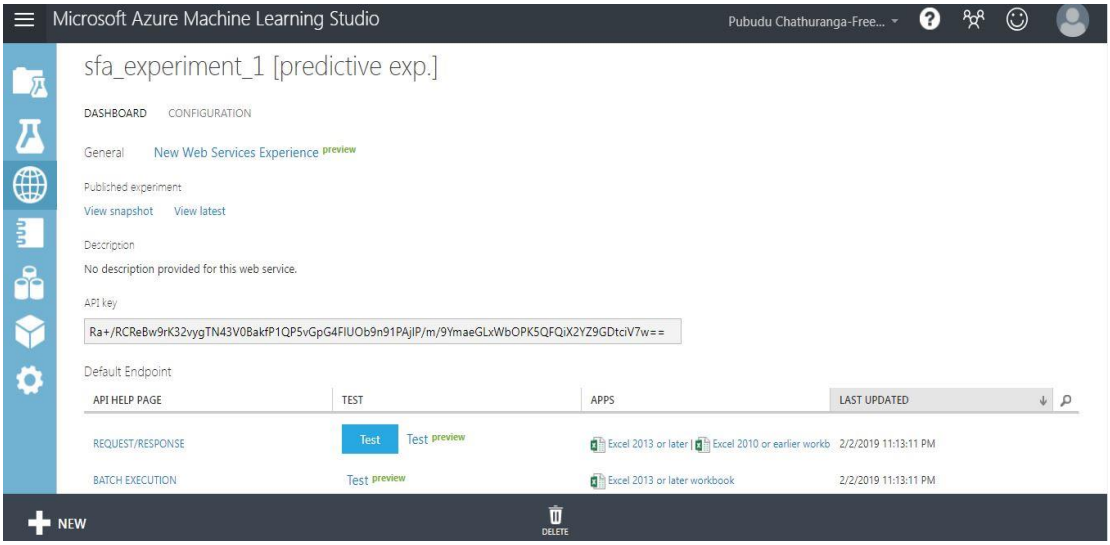
Appendix AC - Training Model (experiment)



Appendix AD - Convert to Predictive analysis



Appendix AE - Deploy as a Web service.



Appendix AF - Generated web service

Sample Code

C# | Python | R

Select sample code

```
// This code requires the Nuget package Microsoft.AspNet.WebApi.Client to be installed.
// Instructions for doing this in Visual Studio:
// Tools -> Nuget Package Manager -> Package Manager Console
// Install-Package Microsoft.AspNet.WebApi.Client

using System;
using System.Collections.Generic;
using System.IO;
using System.Net.Http;
using System.Net.Http.Formatting;
using System.Net.Http.Headers;
using System.Text;
using System.Threading.Tasks;

namespace CallRequestResponseService
{

    public class StringTable
    {
        public string[] ColumnNames { get; set; }
        public string[,] Values { get; set; }
```