References


MD5, 2017. . Wikipedia.


SHA-1, 2017. . Wikipedia.


## Appendix A – ISO8583 Message Sample

### Sample of Online Payment

#### Sample Request Message

```
0200FABC44C12880C0000000000000040000001641234567891472582000000000100000
00000000000121401262761000000660544012746121419096011051000406914442324
691760100091433 =1909221164593210050534862736EPAYSWITCH ONLN
LOCATION COLOMBO LKA144144164123456789147258
```

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTI</td>
<td>0200</td>
</tr>
<tr>
<td>[001] Bitmap (hex 62)</td>
<td>FABC44C12820C000000000000004000000164123456789147258200000000100000</td>
</tr>
<tr>
<td>[002] Primary Account Number (n..16)</td>
<td>16 4123456789147258</td>
</tr>
<tr>
<td>[003] Processing Code (n6)</td>
<td>200000</td>
</tr>
<tr>
<td>[004] Amount, Transaction (n12)</td>
<td>000000100000</td>
</tr>
<tr>
<td>[005] Amount, Settlement (n12)</td>
<td>000000000000</td>
</tr>
<tr>
<td>[007] Trans. Date/Time (n10)</td>
<td>1214012627</td>
</tr>
<tr>
<td>[009] Conversion rate (n8)</td>
<td>10000006</td>
</tr>
<tr>
<td>[011] System trace audit number (n6)</td>
<td>605440</td>
</tr>
<tr>
<td>[012] Time (n6)</td>
<td>012746 (HHmmss)</td>
</tr>
<tr>
<td>[013] Date (n4)</td>
<td>1214 (MMdd)</td>
</tr>
<tr>
<td>[014] Date, expiration (n4)</td>
<td>1909</td>
</tr>
<tr>
<td>[018] Merchant type (n4)</td>
<td>6011</td>
</tr>
<tr>
<td>[022] Point of service entry mode (n3)</td>
<td>051</td>
</tr>
<tr>
<td>[025] Point of service condition code (n2)</td>
<td>00</td>
</tr>
<tr>
<td>[026] Point of service capture code (n2)</td>
<td>04</td>
</tr>
<tr>
<td>[032] Acquiring institution identification code (n..6)</td>
<td>06 914442</td>
</tr>
<tr>
<td>[035] Track 2 data (n..32)</td>
<td>32 4691760100091433 =19092211645932</td>
</tr>
<tr>
<td>[037] Retrieval reference number (an12)</td>
<td>100505348627</td>
</tr>
<tr>
<td>[043] Card acceptor terminal identification (ans..32)</td>
<td>36 EPAYSWITCH ONLN LOCATION COLOMBO LKA</td>
</tr>
<tr>
<td>-----------------------------------------------------</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>[049] Currency code, transaction (n3)</td>
<td>144</td>
</tr>
<tr>
<td>[050] Currency code, settlement (n3)</td>
<td>144</td>
</tr>
<tr>
<td>[103] Account identification 1 (n..16)</td>
<td>16 4123456789147258</td>
</tr>
</tbody>
</table>
Sample Response Message

0200FABC44C12880C00000000000004000000016412345678914725820000000000100000
000000000000121401262761000000660544012746121419096011051000406914442324
691760100091433=190922116459321005053486270036EPAYSWITCH ONLN
LOCATION COLOMBO LKA144144

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTI</td>
<td>0210</td>
</tr>
<tr>
<td>[001] Bitmap (hex 62)</td>
<td>FABC44C12A20C000000000000000000000000000000</td>
</tr>
<tr>
<td>[002] Primary Account Number (n..16)</td>
<td>16 4123456789147258</td>
</tr>
<tr>
<td>[003] Processing Code (n6)</td>
<td>200000</td>
</tr>
<tr>
<td>[004] Amount, Transaction (n12)</td>
<td>00000001000000</td>
</tr>
<tr>
<td>[005] Amount, Settlement (n12)</td>
<td>00000000000000</td>
</tr>
<tr>
<td>[007] Trans. Date/Time (n10)</td>
<td>1214012627</td>
</tr>
<tr>
<td>[009] Conversion rate (n8)</td>
<td>10000006</td>
</tr>
<tr>
<td>[011] System trace audit number (n6)</td>
<td>605440</td>
</tr>
<tr>
<td>[012] Time (n6)</td>
<td>012746 (HHmmss)</td>
</tr>
<tr>
<td>[013] Date (n4)</td>
<td>1214 (MMdd)</td>
</tr>
<tr>
<td>[014] Date, expiration (n4)</td>
<td>1909</td>
</tr>
<tr>
<td>[018] Merchant type (n4)</td>
<td>6011</td>
</tr>
<tr>
<td>[022] Point of service entry mode (n3)</td>
<td>051</td>
</tr>
<tr>
<td>[025] Point of service condition code (n2)</td>
<td>00</td>
</tr>
<tr>
<td>[026] Point of service capture code (n2)</td>
<td>04</td>
</tr>
<tr>
<td>[032] Acquiring institution identification code (n..6)</td>
<td>06 914442</td>
</tr>
<tr>
<td>[035] Track 2 data (n..32)</td>
<td>32 4691760100091433 =19092211645932</td>
</tr>
<tr>
<td>[037] Retrieval reference number (an12)</td>
<td>100505348627</td>
</tr>
<tr>
<td>[039] Response Code (an2)</td>
<td>Success: 00 Declined: 07 Not enough funds: 51 Suspicious: 59</td>
</tr>
<tr>
<td>[043] Card acceptor terminal identification</td>
<td>36 EPAYSWITCH ONLN LOCATION</td>
</tr>
<tr>
<td>(n..32)</td>
<td>COLOMBO LKA</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>[049] Currency code, transaction</td>
<td>144</td>
</tr>
<tr>
<td>(n3)</td>
<td></td>
</tr>
<tr>
<td>[050] Currency code, settlement</td>
<td>144</td>
</tr>
<tr>
<td>(n3)</td>
<td></td>
</tr>
</tbody>
</table>
Appendix B – Important Code Segments

API Service Implementation

Customers.java

```java
public class Customers implements Serializable {

    private static final long serialVersionUID = 1L;
    private Long customerId;
    private String fullName;
    private String nid;
    private String mobile;
    private short isActive;
    private String username;
    private String password;
    private String respo;

    public Customers() {
    }
    public Long getCustomerId() {
        return customerId;
    }
    public void setCustomerId(Long customerId) {
        this.customerId = customerId;
    }
    public String getFullName() {
        return fullName;
    }
    public void setFullName(String fullName) {
        this.fullName = fullName;
    }
    public String getNid() {
        return nid;
    }
    public void setNid(String nid) {
        this.nid = nid;
    }
    public String getMobile() {
        return mobile;
    }
    public void setMobile(String mobile) {
        this.mobile = mobile;
    }
    public short getIsActive() {
        return isActive;
    }
    public void setIsActive(short isActive) {
        this.isActive = isActive;
    }
    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
```
this.password = password;
}
public String getRespo() {
    return respo;
}
public void setRespo(String respo) {
    this.respo = respo;
}

public Customers[] SignIn(String username, String password) {
    Customers[] ccArray = null;
    dbAccess db = new dbAccess();
    Connection con = db.GetConnection();
    try {
        Statement stmt = con.createStatement();
        ResultSet result;
        result = stmt.executeQuery("CALL sp_login('" + username + "," +
password + ")");
        ccArray = new Customers[1];
        while (result.next()) {
            Customers cc = new Customers();
            if (result.getString("resp") equals("0x9300")) {
                cc.setRespo(result.getString("resp"));
            } else {
                cc.setRespo(result.getString("resp"));
            }
            ccArray[0] = cc;
        }
    } catch (SQLException ex) {
        Logger.getLogger(card_service_1.class.getName()).log(Level.SEVERE,
null, ex);
    }
    return ccArray;
}

public String Register(Customers customers) {
    dbAccess db = new dbAccess();
    Connection con = db.GetConnection();
    String response = "";
    try {
        Statement stmt = con.createStatement();
        ResultSet result;
        result = stmt.executeQuery("CALL sp_register(
" + customers.fullName + "," +
customers.nid + "," +
customers.mobile + "," +
customers.username + "," +
customers.password + ")");
        while (result.next()) {
            response = result.getString("resp");
        }
    } catch (SQLException ex) {
        Logger.getLogger(card_service_1.class.getName()).log(Level.SEVERE,
null, ex);
    }
    return response;
}
String ChangePass(String CustomerID, String OldPassword, String NewPassword)
{
    dbAccess db = new dbAccess();
    Connection con = db.GetConnection();
    String response = "";
    try {
        Statement stmnt = con.createStatement();
        ResultSet result;
        result = stmnt.executeQuery("CALL sp_change_password(" + "'" + CustomerID + ", " + "'" + OldPassword + ", " + "'" + NewPassword + ")");
        while (result.next()) {
            response = result.getString("resp");
        }
    } catch (SQLException ex) {
        Logger.getLogger(card_service_1.class.getName()).log(Level.SEVERE, null, ex);
    }
    return response;
}

CustomerChannels.java

public class CustomerChannels implements Serializable {
    private static final long serialVersionUID = 1L;
    private long customerId;
    private int channelId;
    private String accountName;
    private String accountSha256;
    private String accountMask;
    private String expiry;
    private String nameIn;
    private short isActive;
    private short isOn;
    private Float maxOnlineValue;
    private short isOnlineOn;
    private Float maxOfflineValue;
    private short isOfflineOn;
    private Float maxWithdrawValue;
    private short isWithdrawOn;

    public CustomerChannels() {
    }
    public long getCustomerId() {
        return customerId;
    }
    public void setCustomerId(long customerId) {
        this.customerId = customerId;
    }
    public int getChannelId() {
        return channelId;
    }
    public void setChannelId(int channelId) {
        this.channelId = channelId;
    }
    public String getAccountName() {
        return accountName;
    }
    public void setAccountName(String accountName) {
        this.accountName = accountName;
    }
    public String getAccountSha256() {
return accountSha256;
}
public void setAccountSha256(String accountSha256) {
    this.accountSha256 = accountSha256;
}
public String getAccountMask() {
    return accountMask;
}
public void setAccountMask(String accountMask) {
    this.accountMask = accountMask;
}
public String getExpiry() {
    return expiry;
}
public void setExpiry(String expiry) {
    this.expiry = expiry;
}
public String getNameIn() {
    return nameIn;
}
public void setNameIn(String nameIn) {
    this.nameIn = nameIn;
}
public short getIsActive() {
    return isActive;
}
public void setIsActive(short isActive) {
    this.isActive = isActive;
}
public short getIsOn() {
    return isOn;
}
public void setIsOn(short isOn) {
    this.isOn = isOn;
}
public Float getMaxOnlineValue() {
    return maxOnlineValue;
}
public void setMaxOnlineValue(Float maxOnlineValue) {
    this.maxOnlineValue = maxOnlineValue;
}
public short getIsOnlineOn() {
    return isOnlineOn;
}
public void setIsOnlineOn(short isOnlineOn) {
    this.isOnlineOn = isOnlineOn;
}
public Float getMaxOfflineValue() {
    return maxOfflineValue;
}
public void setMaxOfflineValue(Float maxOfflineValue) {
    this.maxOfflineValue = maxOfflineValue;
}
public short getIsOfflineOn() {
    return isOfflineOn;
}
public void setIsOfflineOn(short isOfflineOn) {
    this.isOfflineOn = isOfflineOn;
}
public Float getMaxWithdrawValue() {
    return maxWithdrawValue;
}
public void setMaxWithdrawValue(Float maxWithdrawValue) {
    this.maxWithdrawValue = maxWithdrawValue;
}

/**
public short getIsWithdrawOn() {
    return isWithdrawOn;
}

/**
 * @param isWithdrawOn
 */
public void setIsWithdrawOn(short isWithdrawOn) {
    this.isWithdrawOn = isWithdrawOn;
}

/**
 * @param userID
 * @param IsActive
 * @return
 */
public CustomerChannels[] GetCards(Long userID, String IsActive) {
    CustomerChannels[] ccArray = null;
    dbAccess db = new dbAccess();
    Connection con = db.GetConnection();
    try {
        Statement stmnt = con.createStatement();
        ResultSet result, backResult;
        result = stmnt.executeQuery("CALL sp_get_cards(" +
                + "'" + userID + "," +
                + "'" + IsActive + ")");
        int j = 0;
        while (result.next()) {
            j++;
            ccArray = new CustomerChannels[j];
        }
        int i = 0;
        while (backResult.previous()) {
            CustomerChannels cc = new CustomerChannels();
            cc.setAccountSha256(backResult.getString("account_sha256"));
            cc.setAccountMask(backResult.getString("account_mask"));
            cc.setAccountName(backResult.getString("account_name"));
            cc.setIsActive(Short.parseShort(backResult.getString("is_active")));
            cc.setIsOn(Short.parseShort(backResult.getString("is_on")));
            ccArray[i] = cc;
            i++;
        }
    } catch (SQLException ex) {
        Logger.getLogger(card_service_1.class.getName()).log(Level.SEVERE,
                null, ex);
    }
    return ccArray;
}

/**
 * @param cardData
 * @return
 */
public String AddCard(CustomerChannels cardData) {
    dbAccess db = new dbAccess();
    Connection con = db.GetConnection();
    String response = "";
try {
    Statement stmnt = con.createStatement();
    ResultSet result = stmnt.executeQuery("CALL sp_addcard(" + "+ '' + cardData.getCustomerId + ''");
    while (result.next()) {
        response = result.getString("resp");
    }
} catch (SQLException ex) {
    Logger.getLogger(card_service_1.class.getName()).log(Level.SEVERE, null, ex);
    return response;
}

/**
 * @param cardSha256
 * @return
 */
public CustomerChannels[] GetCardDetails(String cardSha256) {
    CustomerChannels[] ccArray = new CustomerChannels[1];
    dbAccess db = new dbAccess();
    Connection con = db.GetConnection();
    try {
        Statement stmnt = con.createStatement();
        ResultSet result = stmnt.executeQuery("CALL sp_cardview(' + cardSha256 + ");
        while (result.next()) {
            CustomerChannels cc = new CustomerChannels();
            cc.setAccountSha256(result.getString("account_sha256"));
            cc.setAccountMask(result.getString("account_mask"));
            cc.setAccountName(result.getString("account_name"));
            cc.setExpiry(result.getString("expiry"));
            cc.setNameIn(result.getString("name_in"));
            cc.setIsOn(Short.parseShort(result.getString("is_on")));
            cc.setMaxOnlineValue(Float.parseFloat(result.getString("max_online_value")));
            cc.setIsOnlineOn(Short.parseShort(result.getString("is_online_on")));
            cc.setMaxOfflineValue(Float.parseFloat(result.getString("max_offline_value")));
            cc.setIsOfflineOn(Short.parseShort(result.getString("is_offline_on")));
            cc.setMaxWithdrawValue(Float.parseFloat(result.getString("max_withdraw_value")));
            cc.setIsWithdrawOn(Short.parseShort(result.getString("is_withdraw_on")));
            ccArray[0] = cc;
        }
    } catch (SQLException ex) {
        Logger.getLogger(card_service_1.class.getName()).log(Level.SEVERE, null, ex);
        return ccArray;
    }

    public String UpdateCard(CustomerChannels ccUpdateCard) {

}
dbAccess db = new dbAccess();
Connection con = db.GetConnection();
String response = ""
try {
    Statement stmt = con.createStatement();
    ResultSet result;
    result = stmt.executeQuery(
        "CALL sp_updatecard(" + 
            "'" + ccUpdateCard.getAccountSha256() + ", ", 
            "'" + ccUpdateCard.getCustomerId() + ", ", 
            "'" + String.valueOf(ccUpdateCard.getMaxOnlineValue()) + ", "
            "'" + ccUpdateCard.getIsOnlineOn() + ", "
            "'" + String.valueOf(ccUpdateCard.getIsOnlineOn()) + ", "
            "'" + ccUpdateCard.getMaxOfflineValue() + ", "
            "'" + String.valueOf(ccUpdateCard.getIsOfflineOn()) + ", "
            "'" + ccUpdateCard.getMaxWithdrawValue() + ", "
            "'" + String.valueOf(ccUpdateCard.getIsWithdrawOn()) + ", ", 
            ")");
    while (result.next()) {
        response = result.getString("resp");
    }
} catch (SQLException ex) {
    Logger.getLogger(card_service_1.class.getName()).log(Level.SEVERE, null, ex);
}
return response;
}
String DeleteCard(String CardSha256, String CustomerID) {
    dbAccess db = new dbAccess();
    Connection con = db.GetConnection();
    String response = ""
    try {
        Statement stmt = con.createStatement();
        ResultSet result;
        result = stmt.executeQuery(
            "CALL sp_deletecard(" + 
                "'" + CardSha256 + ", "
                "'" + CustomerID + ", 
            ")");
        while (result.next()) {
            response = result.getString("resp");
        }
    } catch (SQLException ex) {
        Logger.getLogger(card_service_1.class.getName()).log(Level.SEVERE, null, ex);
    }
    return response;
}

class user_service_1 {
    /**
     * This is a sample web service operation
     * @param username
     * @param password
     * @return
     */
    @WebMethod(operationName = "SignIn")
    public Customers[] SignIn(
        @WebParam(name = "username") String username,
@WebParam(name = "password") String password) {
    Customers ccSignIn = new Customers();
    return ccSignIn.signIn(username, password);
}

@WebMethod(operationName = "Register")
public String Register(
    @WebParam(name = "fullname") String fullname,
    @WebParam(name = "nic") String nic,
    @WebParam(name = "mobile") String mobile,
    @WebParam(name = "username") String username,
    @WebParam(name = "password") String password) {
    Customers ccRegister = new Customers();
    ccRegister.setFullName(fullname);
    ccRegister.setNid(nic);
    ccRegister.setMobile(mobile);
    ccRegister.setUserName(username);
    ccRegister.setPassword(password);
    return ccRegister.register(ccRegister);
}

@WebMethod(operationName = "ChangePass")
public String ChangePass(
    @WebParam(name = "CustomerID") String CustomerID,
    @WebParam(name = "OldPassword") String OldPassword,
    @WebParam(name = "NewPassword") String NewPassword) {
    Customers ccSignIn = new Customers();
    return ccSignIn.changePass(CustomerID, OldPassword, NewPassword);
}

public class card_service_1 {

    /**
     * This is a sample web service operation
     * @param UserID
     * @param IsActive
     * @return CustomerChannels_
     */
    @WebMethod(operationName = "GetCards")
    public CustomerChannels[] GetCards(
        @WebParam(name = "userID") Long UserID,
        @WebParam(name = "IsActive") String IsActive) {
        CustomerChannels ccInactive = new CustomerChannels();
        return ccInactive.getCards(UserID, IsActive);
    }

    /**
     * @param CardSha256
     * @return
     */
    @WebMethod(operationName = "GetCardDetails")
    public CustomerChannels[] GetCardDetails(@WebParam(name = "CardSha256") String CardSha256) {
        CustomerChannels ccInactive = new CustomerChannels();
        return ccInactive.getCardDetails(CardSha256);
    }

    /**
     * @param CustomerID
     * @param CardName
     */

}
@WebMethod(operationName = "AddCard")
public String AddCard(
    @WebParam(name = "CustomerID") Long CustomerID,
    @WebParam(name = "CardName") String CardName,
    @WebParam(name = "CardSha256") String CardSha256,
    @WebParam(name = "CardMask") String CardMask,
    @WebParam(name = "Expiry") String Expiry,
    @WebParam(name = "NameInCard") String NameInCard) {
    CustomerChannels ccAddCard = new CustomerChannels();
    ccAddCard.setCustomerId(CustomerID);
    ccAddCard.setAccountName(CardName);
    ccAddCard.setAccountSha256(CardSha256);
    ccAddCard.setAccountMask(CardMask);
    ccAddCard.setExpiry(Expiry);
    ccAddCard.setNameInNameInCard;
    return ccAddCard.AddCard(ccAddCard);
}

/**
* @param CustomerID
* @param CardName
* @param CardSha256
* @param IsOn
* @param MaxOnlineValue
* @param IsOnlineOn
* @param MaxOfflineValue
* @param IsOfflineOn
* @param MaxWithdrawValue
* @param IsWithdrawOn
* @return
*/
@WebMethod(operationName = "UpdateCard")
public String UpdateCard(
    @WebParam(name = "CustomerID") Long CustomerID,
    @WebParam(name = "CardName") String CardName,
    @WebParam(name = "CardSha256") String CardSha256,
    @WebParam(name = "IsOn") String IsOn,
    @WebParam(name = "MaxOnlineValue") String MaxOnlineValue,
    @WebParam(name = "IsOnlineOn") String IsOnlineOn,
    @WebParam(name = "MaxOfflineValue") String MaxOfflineValue,
    @WebParam(name = "IsOfflineOn") String IsOfflineOn,
    @WebParam(name = "MaxWithdrawValue") String MaxWithdrawValue,
    @WebParam(name = "IsWithdrawOn") String IsWithdrawOn) {
    CustomerChannels ccUpdateCard = new CustomerChannels();
    ccUpdateCard.setCustomerId(CustomerID);
    ccUpdateCard.setAccountName(CardName);
    ccUpdateCard.setAccountSha256(CardSha256);
    ccUpdateCard.setIsOn(Short.parseShort(IsOn));
    ccUpdateCard.setMaxOnlineValue(Float.parseFloat(MaxOnlineValue));
    ccUpdateCard.setIsOnlineOn(Short.parseShort(IsOnlineOn));
    ccUpdateCard.setMaxOfflineValue(Float.parseFloat(MaxOfflineValue));
    ccUpdateCard.setIsOfflineOn(Short.parseShort(IsOfflineOn));
    ccUpdateCard.setMaxWithdrawValue(Float.parseFloat(MaxWithdrawValue));
    ccUpdateCard.setIsWithdrawOn(Short.parseShort(IsWithdrawOn));
    return ccUpdateCard.UpdateCard(ccUpdateCard);
}

/**
* @param CardSha256
*/
* @param CustomerID
* @return
*/
@WebMethod(operationName = "DeleteCard")
public String DeleteCard(
    @WebParam(name = "CardSha256") String CardSha256,
    @WebParam(name = "CustomerID") String CustomerID
) {
    CustomerChannels ccDeleteCard = new CustomerChannels();
    return ccDeleteCard.DeleteCard(CardSha256, CustomerID);
}

SOAP Service Requests

“Register” Request

```xml
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/
<soap:Header/>
<soap:Body>
<soa:Register>
<full Name>ABC Silver</full Name>
<mobile>01180596197</mobile>
<username>silver</username>
<password>6e15b6e33302f8784e6129675f9e6f6b682e27</password>
</soa:Register>
</soap:Body>
</soap:Envelope>
```

“Register” Response

```xml
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/
<soap:Body>
<soa:RegisterResponse xmlns:ns2="http://services.epay.co/
<return>result</return>
</soap:Body>
</soap:Envelope>
```

“SignIn” Request

```xml
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/
<soap:Header/>
<soap:Body>
<soa:SignIn>
<username>silver</username>
<password>6e15b6e33302f8784e6129675f9e6f6b682e27</password>
</soa:SignIn>
</soap:Body>
</soap:Envelope>
```

“SignIn” Response

```xml
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/
<soap:Body>
<soa:SignInResponse xmlns:ns1="http://services.epay.co/
<return>
</soap:Body>
</soap:Envelope>
```

“AddCard” Request

```xml
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/
<soap:Header/>
<soap:Body>
<soa:AddCard>
<Customer ID>11</Customer ID>
<Bank>A B C Bank</Bank>
<CardSha256>1ef5f15b916a97a1e7b2f6b93467e014773515abcdc3</CardSha256>
</soap:Body>
</soap:Envelope>
```

“AddCard” Response

```xml
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/
<soap:Body>
<soa:AddCardResponse xmlns:ns1="http://services.epay.co/
<return>result</return>
</soap:Body>
</soap:Envelope>
```

“UpdateCard” Request

```xml
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/
<soap:Header/>
<soap:Body>
<soa:UpdateCard>
<Customer ID>11</Customer ID>
<CardSha256>1ef5f15b916a97a1e7b2f6b93467e014773515abcdc3</CardSha256>
</soap:Body>
</soap:Envelope>
```

“UpdateCard” Response

```xml
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/
<soap:Body>
<soa:UpdateCardResponse xmlns:ns1="http://services.epay.co/
<return>result</return>
</soap:Body>
</soap:Envelope>
```
“GetCards” List Request

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/
  <soap:Header/>
  <soap:Body>
    <ns3:GetCards/>
  </soap:Body>
</soap:Envelope>
```

“GetCards” List Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/
  <soap:Body>
    <ns2:GetCardsResponse xmlns:ns2="http://services.epay.com">
      <return>
        <IsInactive2>false</IsInactive2>
      </return>
    </ns2:GetCardsResponse>
  </soap:Body>
</soap:Envelope>
```

“GetCardDetails” Request

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/
  <soap:Header/>
  <soap:Body>
    <ns3:GetCardDetails/>
  </soap:Body>
</soap:Envelope>
```

“GetCardDetails” Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/
  <soap:Body>
    <ns2:GetCardDetailsResponse xmlns:ns2="http://services.epay.com">
      <return>
        <IsInactive2>false</IsInactive2>
      </return>
    </ns2:GetCardDetailsResponse>
  </soap:Body>
</soap:Envelope>
```

“DeleteCard” Request

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/
  <soap:Header/>
  <soap:Body>
    <ns3:DeleteCard/>
  </soap:Body>
</soap:Envelope>
```

“DeleteCard” Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/
  <soap:Body>
    <ns2:DeleteCardResponse xmlns:ns2="http://services.epay.com">
      <return>
      </return>
    </ns2:DeleteCardResponse>
  </soap:Body>
</soap:Envelope>
```
“GetTransactions” Request

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <soapenv:GetDeleteCard>
      <CardHash>1b4fde115204d2a5742b3d09f927061510d</CardHash>
      <CustomerID>11</CustomerID>
    </soapenv:GetDeleteCard>
  </soapenv:Body>
</soapenv:Envelope>
```

“GetTransactions” Response

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <soapenv:GetTransactions>
      <CustomerID>11</CustomerID>
      <Range>150</Range>
    </soapenv:GetTransactions>
  </soapenv:Body>
</soapenv:Envelope>
```
Appendix C – User Interfaces

Mobile Application

Register All Fields Required

Register Username Exists

Sign In All Fields Required

Add New Card All Fields Required
Customer and Cards Manager

User Login

Inactive Customers (Activate Customers)

<table>
<thead>
<tr>
<th>Customer ID</th>
<th>Full Name</th>
<th>NIC Number</th>
<th>Mobile</th>
<th>A/D</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>M.R.N.K Malawatigoda</td>
<td>875235852V</td>
<td>0718305526</td>
<td>Activate</td>
</tr>
<tr>
<td>11</td>
<td>A R C Sha</td>
<td>8402456542V</td>
<td>0713096373</td>
<td>Activate</td>
</tr>
</tbody>
</table>

Active Customers (Deactivate Customers)

<table>
<thead>
<tr>
<th>Customer ID</th>
<th>Full Name</th>
<th>NIC Number</th>
<th>Mobile</th>
<th>A/D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>G T C Liyanage</td>
<td>8724353162V</td>
<td>0713096373</td>
<td>Deactivate</td>
</tr>
<tr>
<td>7</td>
<td>K Kinnerara</td>
<td>872435168v</td>
<td>0771576855</td>
<td>Deactivate</td>
</tr>
</tbody>
</table>

Inactive Cards (Activate Cards)

<table>
<thead>
<tr>
<th>Added By</th>
<th>Card Number</th>
<th>Expiry</th>
<th>Name In Card</th>
<th>A/D</th>
</tr>
</thead>
<tbody>
<tr>
<td>M R N K Malawatigoda</td>
<td>47694500000001472</td>
<td>11/19</td>
<td>Malawatigoda</td>
<td>Activate</td>
</tr>
<tr>
<td>G T C Liyanage</td>
<td>41234500000001234</td>
<td>03/17</td>
<td>G T C Liyanage</td>
<td>Activate</td>
</tr>
</tbody>
</table>
## Active Cards (Deactivate Cards)

<table>
<thead>
<tr>
<th>Added By</th>
<th>Card Number</th>
<th>Expiry</th>
<th>Name in Card</th>
<th>A/D</th>
</tr>
</thead>
<tbody>
<tr>
<td>K Kamsara</td>
<td>4147250000004725</td>
<td>12/20</td>
<td>A B C Kumara</td>
<td>[Deactivate]</td>
</tr>
<tr>
<td>K Kamsara</td>
<td>412345000001230</td>
<td>12/18</td>
<td>Kumara</td>
<td>[Deactivate]</td>
</tr>
</tbody>
</table>
Appendix D – Questionnaire and Responses

ePaySwitch

This questionnaire was prepared as evaluation technique to the research "Implementing a Software Switch and a Mobile Application to Prevent Fraud and Control the Usage of Electronic Transactions" done by G. T. C. Liyanage as a partial fulfillment of the M. Sc, in IT at University of Moratuwa, Faculty of Information Technology.

Please be kind enough to spend a little from your valuable time to answer this as it will help me in completing the evaluation process of the above mentioned task.

Thank You,

Sincerely,
#Thilanga (G T C Liyanage)

*Required

ePaySwitch
50 responses

1. Please Specify Your Age Group *

Mark only one oval.

☐ <20 years
☐ 21-25 years
☐ 26-30 years
☐ 31-35 years
☐ 36-40 years
☐ 41-45 years
☐ 45-50 years
☐ >51 years
2. Do you have a Credit or Debit Card? *
   Mark only one oval.
   - Yes      Skip to question 3.
   - No       Skip to question 12.

3. How frequently you use Credit/Debit card for transactions? *
   Mark only one oval.
   - Often    Skip to question 4.
   - Occasionally  Skip to question 12.
   - Never    Skip to question 12.
4. Have you ever lost your card? *
Mark only one oval.

☐ Yes
☐ No

Have you ever lost your card? (27 responses)

70.4% Yes
29.6% No

5. Have you ever thought someone will steal your card & perform transaction before you know?
Mark only one oval.

☐ No
☐ Yes, But I keep using
Have you ever thought someone will steal your card & perform transaction before you know?
(27 responses)

- 88.9%
- 11.1%

6. Do you aware of transaction amounts you perform via cards? *
Mark only one oval.
- Yes, I keep track of them
- No, I have no idea
- No, Anyway it's my money

Do you aware of transaction amounts you perform via cards?
(27 responses)

- Yes, I keep track of them: 33.3%
- No, I have no idea: 48.1%
- No, Anyway it's my money: 18.5%

7. If someone steal your card and try to perform transactions, how will YOU be able to block it? *
Mark only one oval.
- No way
- By calling card center
- Other: ____________________________
If some one steal your card and try to perform transactions, how will YOU be able to block it?
(27 responses)

8. What if you have an application to switch on/off your card in order to perform a transaction? Will you try it?
Mark only one oval.
- Yes. That would be grate
- No. I am OK with what I have
- May be, Need to try

Skip to question 10.

What if you have an application to switch on/off your card in order to perform a transaction? Will you try it?
(27 responses)
9. What if you have an application to switch on/off your card in order to perform transactions? Will you use a card? *  
*Mark only one oval.*

☐ Yes. I will  
☐ No. I don't think so  
☐ May be. Need to try  

**What if you have an application to switch on/off your card in order to perform transactions? Will you use a card?**  
(18 responses)

![Pie chart showing responses]

- 72.2% Yes, I will
- 16.7% No, I don't think so
- 11.1% May be. Need to try

In the below shown figure, we suggest you an application which can control your card transactions. You can switch on/off your card completely or for specific types on transactions.

**Card Control App**
### Search Transactions

<table>
<thead>
<tr>
<th>Search Last</th>
<th>Days</th>
<th>Date/Time</th>
<th>Amount</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>412345XXXXX7258</td>
<td>7</td>
<td>2017-08-29 14:51:39.0</td>
<td>11000.0</td>
<td>ONLINE LIMIT</td>
</tr>
<tr>
<td>412345XXXXX7258</td>
<td>7</td>
<td>2017-08-29 14:47:19.0</td>
<td>9000.0</td>
<td>ONLINE SUCCESS</td>
</tr>
<tr>
<td>412345XXXXX7258</td>
<td>7</td>
<td>2017-08-29 14:38:19.0</td>
<td>9000.0</td>
<td>CARD OFF</td>
</tr>
<tr>
<td>412345XXXXX7258</td>
<td>7</td>
<td>2017-08-29 14:36:58.0</td>
<td>9000.0</td>
<td>CARD OFF</td>
</tr>
</tbody>
</table>

10. If you have given a card control application as in image, do you think you will have a more control when your card lost/stolen and its' transactions? *

Mark only one oval.

- [ ] Yes
- [ ] No
- [ ] Maybe

If you have given a card control application as in image, do you think you will have a more control when your card lost/stolen and its' transactions?

(42 responses)

![Pie Chart](image)

- Yes: 64.3%
- No: 31%
- Maybe: 4.7%
11. What is your most suitable comment on the proposed ePaySwitch? *
Mark only one oval.
- Worth it, May Try
- I already have a control mechanism like this
- This solution does not fit me

Stop filling out this form.

What is your most suitable comment on the proposed ePaySwitch?
(42 responses)

85.7%

12. Most relevant reason not to have, use or occasionally use card? *
Mark only one oval.
- I use cash and cards both  Skip to question 9.
- I will loose my money (frauds)  Skip to question 9.
- I will not have a control over my transactions  Skip to question 9.
- I do not know how to use it  Stop filling out this form.
- I am not interested  Stop filling out this form.
Most relevant reason not to have, use or occasionally use card?
(23 responses)

- 52.2%: I use cash and cards both
- 13%: I will lose my money (frauds)
- 8.7%: I will not have control over my transactions
- 8.7%: I do not know how to use it
- 17.4%: I am not interested