Online Banking Application

**TEAM MEMBERS**

Duvuru Babureddy, Deepika

Sivakumar, Sujatha

Agrawal, Shruti

Category

Introduction………………………………………………………………………………………………………………………………………….3

Part 1: Planning.............................................................................................................................................4

1.1 Introduction about the problem domain......................................................................................4

1.2 System Request.............................................................................................................................5

1.3 Feasibility analysis.........................................................................................................................6

1.4 Requirements Definition...............................................................................................................8

Part2: Analysis............................................................................................................................................10

2.1 Activity diagram ............................................................................................................................10

2.2 Use Case Diagram .........................................................................................................................11

2.3 Use Case Description ....................................................................................................................12

2.4 Initial Class Diagram....................................................................................................................18

2.5 Sequence Diagram......................................................................................................................19

Part 3: Design..............................................................................................................................................25

3.1 Final Class Diagram .....................................................................................................................25

3.2 Package Diagram.........................................................................................................................26

3.3 Database design..........................................................................................................................27

3.4 Data access and manipulation design.........................................................................................28

**Introduction**

This document is the final report of the team project. The project will be shown in four parts: planning, analysis, design and implementation.

The Planning part includes the introduction about the problem domain, System Request, Feasibility analysis and Requirements Definition.

The analysis part includes the activity diagram, use case descriptions, use case diagrams, initial class diagrams and sequence diagrams.

The design part includes the final class diagram, package diagram, database design and data access and manipulation design.

The implementation part includes the screen shot of the software.

**PART 1: PLANNING**

**1.1 Introduction about the problem domain**

Online banking, also known as internet banking, e-banking or virtual banking, is an electronic payment system that enables customers of a bank or other financial institution to conduct a range of financial transactions through the financial institution's website. The online banking system will typically connect to or be part of the core banking system operated by a bank and is in contrast to branch banking which was the traditional way customers accessed banking services.

Nowadays, people transfer money online rather than going to bank. In this project we are mainly focused about tracking the transfer amount in different stages till the time it reaches the beneficiary. We can check it online instead of asking the beneficiary to check their account. Also, this project deals with developing other functions of an online banking application system.

* 1. **SYSTEM REQUEST**

**Project Sponsor:** Project sponsorship is an active senior management role, responsible for identifying the business need, problem or opportunity. The sponsor ensures the project remains a viable proposition and that benefits are realized, resolving any issues outside the control of the project manager.

**Business Need**: The main purpose of this Online Banking application is to enable bank customers to access their accounts and make payments easily using this application.

**Business Requirements:**

Using this Online Banking application, customers should be able to perform all the banking related operations online without the need of visiting the bank in person. They should be able to check the balances in their checking and savings account. They can also perform transfers to the accounts added. Following are the key features of Online Banking Application:

* View current balance in Checking and Savings Account.
* Generate the statements for the accounts.
* Perform account to account transfers.
* Credit card bill payment.
* Avail Loan feature.
* Special offers and deal.
* Update personal information.

**Business Value:**

We expect by creating this Online Banking application there will be a significant increase in the number of customers. By creating this global application we wish to attract customers from the cities where there are no physical branches for the bank. We aim to eventually reduce the number of physical branches and promote the use of Online Banking using our application. We aim to achieve 100% customer satisfaction by ease of use of this application and reducing the number of actual visits to the physical branch. We are targeting IOS, Android and Windows devices like phones and tablets.

**Special issues or Constraints:**

* The top Management views this is a strategic plan. Since the application is internet based it will add value to the overall application.
* Professionals with expertise in multiple platforms will be needed to develop this application.
* Thorough testing of this application should be done to avoid unexpected crashes and remove critical bugs.
* Continuous customer interaction and feedback should be maintained throughout the development phase to build a user friendly application.

**1.3 FEASIBITY ANALYSIS**

**Technical Feasibility**

The online banking system is a technically feasible project with few risks associated.

Transfer of funds:

* The risk associated with transferring of funds is medium to high.
* The process of fund transfer is clear.
* The risk associated with duration taken to do the transfer to the relative accounts is high.

Programming Language:

* There are three team members in the project team. All three of them have profound experience in the software sector.
* Two of the team members are good at Java coding, while the other is well versed in object oriented concepts.
* The tools for the development of app are available in the internet.

Project Size:

* The project size is considerably big as there are a lot of functionalities involved. Hence the risk is high.
* The time for formulating the deliverable is about 45 days which is considerably enough for developing the app, provided there are less delays due to personal issues.

System Compatibility:

* This will be a new system which will not be integrated with any existing technologies.
* The system will be using highly confidential and high speed databases to store data.
* The downtime must be minimal.

**Economic Feasibility:**

* No financial cost is involved.
* The performance of the associated bank is expected to increase as everything becomes spontaneous and computerized with the help of the app and less man power is associated in doing the actual banking work.
* The bank is expected to gain more customers as the app enhances their comfort of doing money transfers from wherever they want at any time.

**Organizational Feasibility:**

From the organizational point of view the project has medium risk.

* The project’s size and speed suit the needs of the fast paced world.
* Every team member is bound to know the concepts of the system and its workflow.
* Appreciation is expected from the customers as it makes their work easy and saves time.
* The risk of maintaining the confidentiality of the data involved is high.
  1. **REQUIREMENTS DEFINITION**

**Nonfunctional Requirements**

1. Operational Requirements

• The system will automatically calculate the amount in the account wherever some amount is credited or debited.

• The system will be able to show the lists of customers’ accounts to the administrator.

2. Performance Requirements

* The system must be consistent.
* System with high speed is a demanding requirement.

3. Security Requirements

• There is a unique account for the administrator.

• Each customer’s customer id is decided by administrator.

• The system must be highly confidential and reliable.

4. Cultural and Political Requirements

None

**Functional Requirements**

1. Account inquiry

• By logging on to the online banking system the customers can check the balance in their checkin and savings accounts.

• Customers can get a bank statement for every monthly transaction with description.

2. Amount transfer

• Transfer can be done in two ways. One using the sure pay and the other through standard transfer.

* Transfer is done using the beneficiary name and email id.
* Transfer status can be tracked using the confirmation number.
* Transfer within the savings and checking account.

3. Cards

* Debit card and credit card transactions are available for the customer.
* Credit card payments are done online by linking the debit card details to it.
* Reward points are given for credit card payments and can be used to get gift vouchers.
* Credit card and the debit card information can be linked with other online services to make payments online.

4. Management

• The administrator has ability to manage (modify, create) the customer id for each account.

• The administrator can also access to the list of accounts.

5. Log in

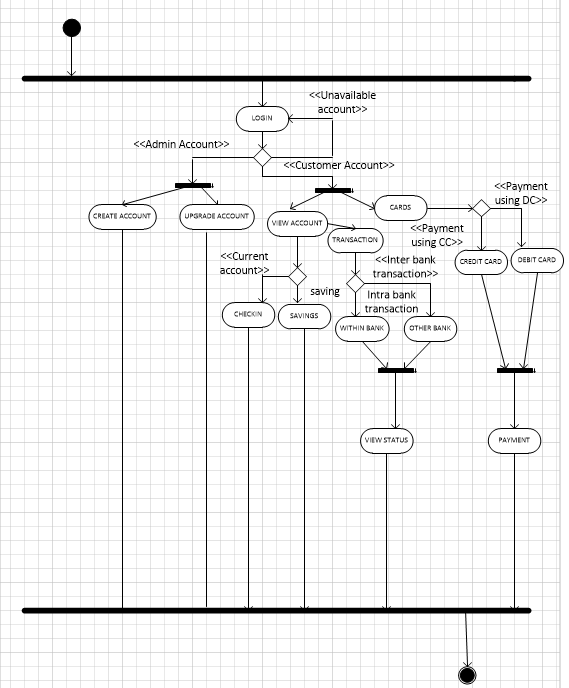
• Customers and administrator should input their customer id and password before they operate the online banking system.

**PART 2: ANALYSIS**

* 1. **Activity Diagram:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows) Activity diagrams show the overall flow of control.

In our activity diagram we have one loop. When the customer logs in he/she can view account, transaction and cards option. When Admin logs in he/she can create account for new customers or upgrade account for the existing customers.

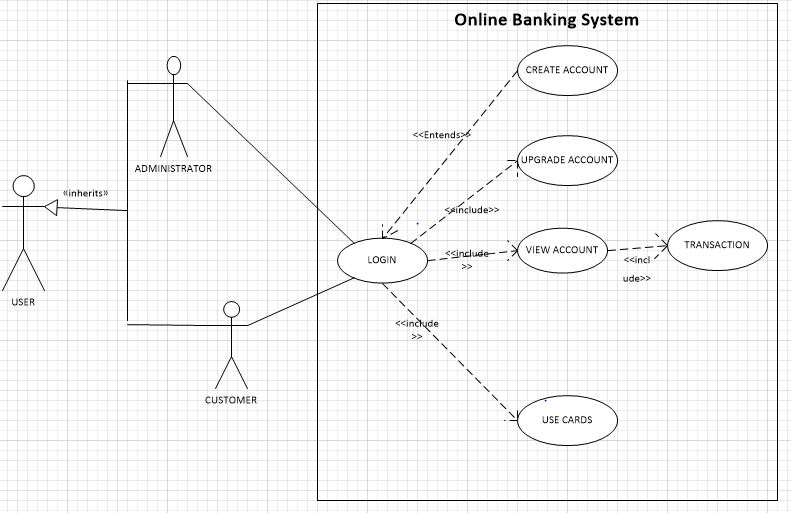


* 1. **Use Case Diagram**

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

We have 6 Use Cases:

* Login
* Create account
* Upgrade account
* View account
* Transaction
* Use Cards



* 1. **Use Case Description**

|  |  |  |
| --- | --- | --- |
| **Use-Case Name:** Login | **ID:** 1 | **Importance level:** high |
| **Primary Actor:** Customer | **User Case Type:** Detail, essential | |
| **Stakeholders and Interests:** **Customer:** wants to login and access the home page for accessing the accounts, initiating transfers and payments | | |
| **Brief Description:**  This use case describes how the customer logins into the system and access to different module | | |
| **Trigger:** The customer clicks on the "login" button **Type:** External | | |
| **Relationships:**  **Association:  Include:** View Account, Upgrade Account and Use Cards **Extend:** Create Account **Generalization:** | | |
| **Normal Flow of Events:** 1. The customer clicks on the “login” button present on the login page. 2. The customer enters their login username and password 3. The username and password are valid and customer is navigated to the home page.  4.The customer can see “View account”, “Upgrade account” and “Use cards” option. | | |
| **Alternate/Exceptional Flows:** 1. The username or password is invalid 2. The system displays an error 3. The customer is navigated back to login page  4. The “Create account” option should be available to customer. | | |

|  |  |  |
| --- | --- | --- |
| **Use-Case Name:** New Account | **ID:** 2 | **Importance level:** high |
| **Primary Actor:** Customer | **User Case Type:** Detail, essential | |
| **Stakeholders and Interests:** **Customer:** wants his credentials to be created so that customer can access his accounts initiate transfer and payments. **Admin:** wants to access the create customer, update customer info and show all customers information. | | |
| **Brief Description:**  This use case describes how the customer logins into the system and access to different module | | |
| **Trigger:** The customer initiates the create customer request. **Type:** External | | |
| **Relationships:**  **Association:** Admin, Customer **Include:**  **Extend: Generalization:** | | |
| **Normal Flow of Events:** 1. The new customer initiates the process for his credentials. 2. The administrator creates a new customer. 3. The new customer details are added to the database. 4. Existing customer wants his details to be updated. 5. Administrator updated the details for the existing customer. 6. Changed details are recorded in the database. 7. Administrator can retrieve the list of all the customers. | | |
| **Alternate/Exceptional Flows:** | | |

|  |  |  |
| --- | --- | --- |
| **Use-Case Name:** Upgrade Account | **ID: 3** | **Importance Level:** Medium |
| **Primary Actor:** Administrator | | **Use Case Type:** Detail, essential |
| **Stakeholders and Interests:**  Administrator – wants to upgrade customer’s account | | |
| **Brief Description:**  This use case describes how the administrator upgrades the account information | | |
| **Trigger:**  The Administrator clicks on the “upgrade account” button  **Type:**  External | | |
| **Relationships:**  **Association:**  **Include:**  **Extend:**  **Generalization:** | | |
| **Normal Flow of Events:**  1. The administrator accesses to the home page and press the “upgrade account info” button  2. The system shows all accounts as a list, there exist buttons beside each account item called “upgrade”  3. The administrator choose the account record he/she wants to update and clicks on the “upgrade” button next to the record  4. The system will provide a table for administrator to change the information of the account  5. The administrator type the new information into the table  6. The administrator clicks on the “submit” button  7. The administrator can repeat 2-6, or quit the system | | |
| **Sub flows:** | | |
| **Alternate/Exceptional Flows:**  3a1: Not all tables have been filled  3a2: Administrator will see a page showing error  3a3: Administrator is returned and continue to fill in the blanks | | |

|  |  |  |
| --- | --- | --- |
| **Use-Case Name:** View Account | **ID: 4** | **Importance Level:** Medium |
| **Primary Actor:** Customer | | **Use Case Type:** Detail, essential |
| **Stakeholders and Interests:**  Customer – to view account details using View Account Details module | | |
| **Brief Description:**  This use case describes how the customer views account information | | |
| **Trigger:**  The Customer clicks on the “View account Details” button  **Type:**  External | | |
| **Relationships:**  **Association:**  **Include:** Transaction  **Extend:**  **Generalization:** | | |
| **Normal Flow of Events:**  1. The Customer accesses the account details using View Account Details button.  2. The system shows checking and savings accounts as a list, there exist buttons beside to select the accounts.  3. The Customer chooses the account he/she wants to view.  4. The system will provide a table for customer showing the balance of the account.  5. The customer accesses the monthly account statement.  6.The system shows both checking and savings account statement as requested.  7. The customer can download the statements if required.  8.The customer can see “Transaction” option. | | |
| **Sub flows:** | | |
| **Alternate/Exceptional Flows:** | | |

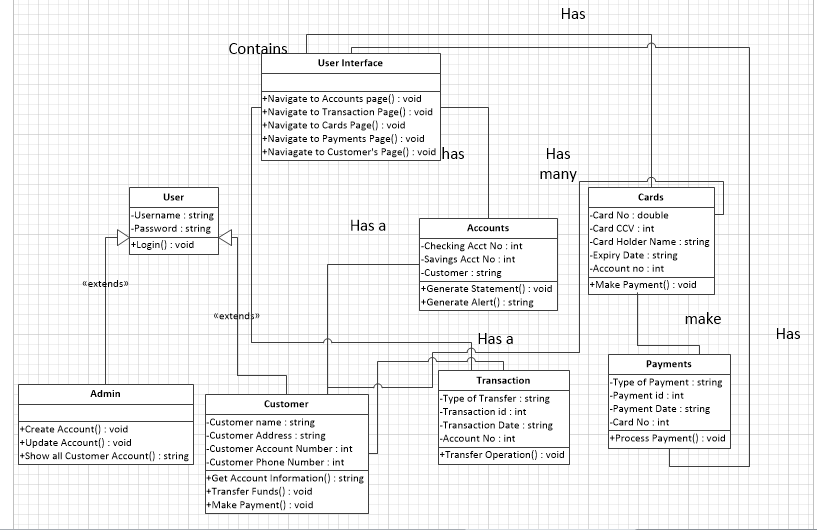
|  |  |  |
| --- | --- | --- |
| **Use Case Name** **:** Transaction | **ID** **:** 5 | **Importance level** **:** High |
| **Primary Actor** **:** Customer | **Use Case Type :** Detail , Essential | |
| **Stakeholder and Interests:**  Customer: Wants to make a transaction.  Administrator : Updates account balance | | |
| **Brief Description:**  This use case describes how the customer makes a transaction using the system. | | |
| **Trigger:**  Customer clicks the “transaction” option.  **Type:**  External | | |
| **Relationships:**  Association:  Generalization: | | |
| **Normal Flow of Events:**   1. The customer access the system. 2. The customer clicks “Transaction” option. 3. The system shows the account types belonging to the customer. 4. The customer chooses account. 5. The customer enters transaction details. 6. The customer clicks “Proceed” option. 7. The transaction success/failure message pops up. 8. The details are saved in database and account balance is updated. 9. The customer can repeat steps 2-6 or logout. | | |
| **Sub flows:** | | |
| **Alternate /Exceptional flows:**  5a1. The input account details are wrong.  5a2. An error message is triggered.  5a3. The system returns to input details page. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name :**Use Cards | | **ID :** 6 | **Importance Level :** High |
| **Primary Actor :** Customer | **Use case type :** Detail, Essential | | |
| **Stakeholders and Interest:**  Customer: Wants to make a payment.  Administrator: Update credit/debit balance. | | | |
| **Brief Description:**  This use case describes how customer makes a payment using debit/credit card. | | | |
| **Trigger:**  The customer clicks the “payment” option.  **Type:**  External | | | |
| **Relationships:**  Association:  Generalization: | | | |
| **Normal flow of events:**   1. The customer access the system. 2. The customer chooses “payment” option. 3. The system displays card options. 4. The customer chooses credit/debit card. 5. The customer enters payment details. 6. The customer enters card details. 7. The customer clicks “Proceed” option. 8. The transaction success/failure message pops up. 9. The credit/debit history is saved in database. 10. The credit/debit balance is updated. 11. The customer can repeat steps 2-7 or logout off the system. | | | |
| **Sub flows:** | | | |
| **Alternate/Exceptional Flows:**  6a1. The card details are wrong.  6a2. The error message is triggered.  6a3. The system returns to input page. | | | |

* 1. **Initial Class Diagram:**

This is our initial Class Diagram. We’ve produced eight classes for our project: User Class, Admin Class, Customer Class, Accounts Class, Transaction Class, Cards Class, Payments Class and User Interface Class. There is a generalization relationship between User class, Admin Class and Customer Class. In our Online Banking system we have Accounts class, Transaction class, Cards Class, Payments Class and User Interface class all linked to Customer Class.

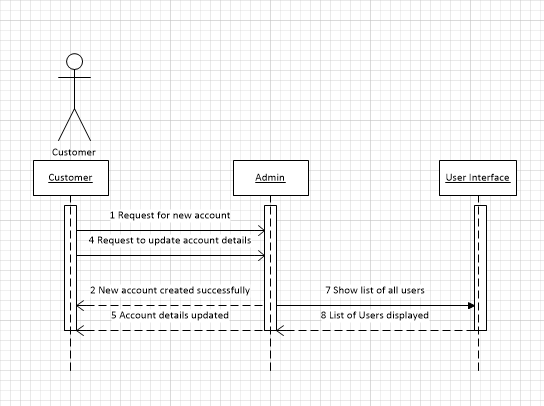
User Interface class is used to display the details page for Accounts class, Transaction class, Cards Class and Payments Class.



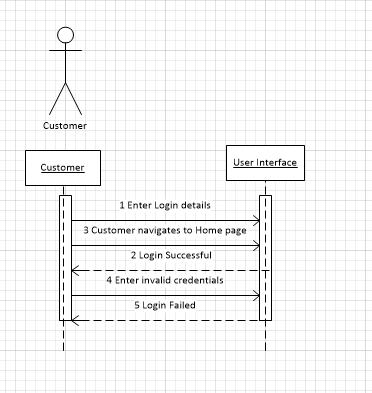
* 1. **Sequence Diagrams:**

A Sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the

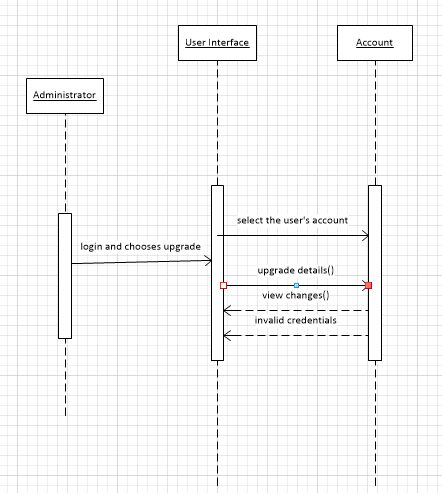
**2.5.1 Create Account Sequence Diagram:**



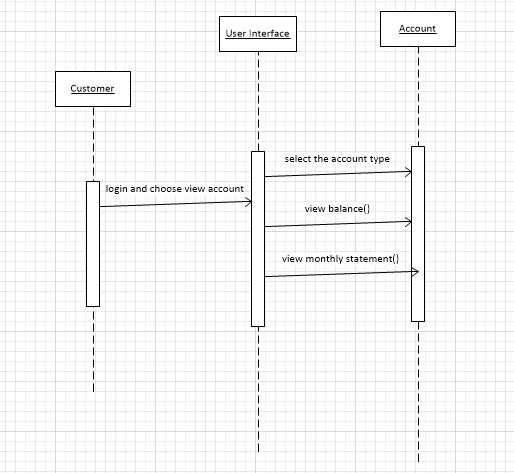
**2.5.2 Login Sequence Diagram:**



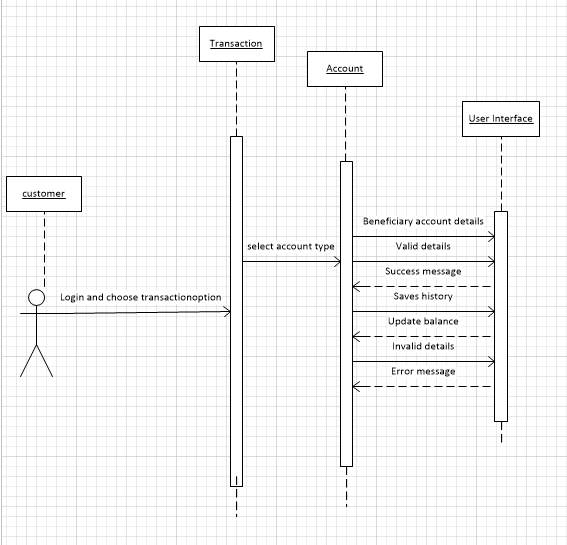
**2.5.3 Upgrade Account Sequence Diagram**



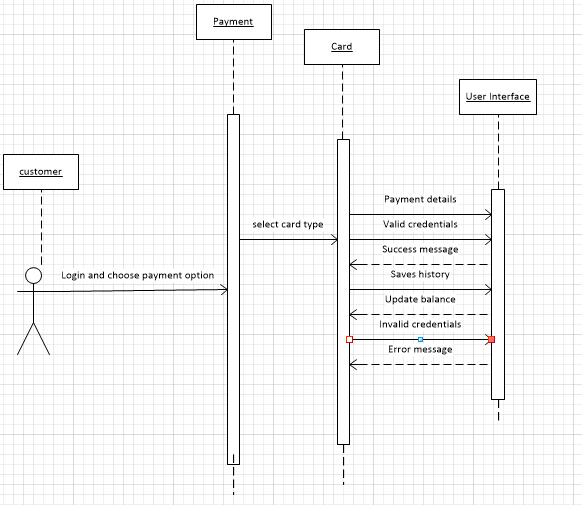
**2.5.4 View Account Sequence Diagram**



**2.5.5 Transaction use case sequence diagram:**



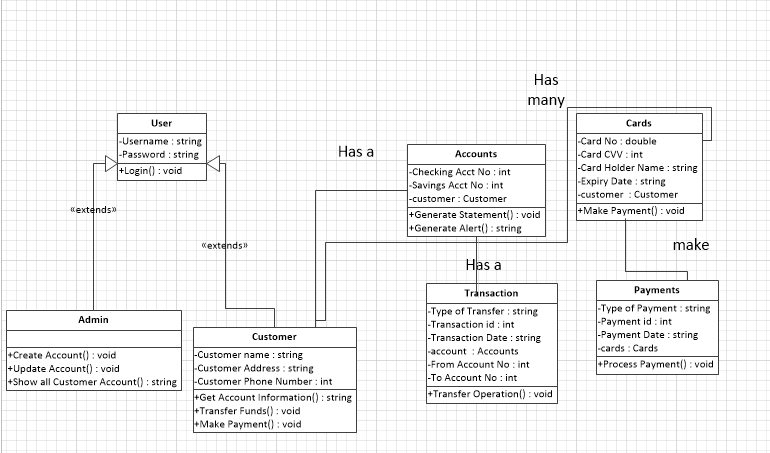
**2.5.6Card Use case Sequence diagram:**



**PART 3: DESIGN**

* 1. **Final Class Diagram**

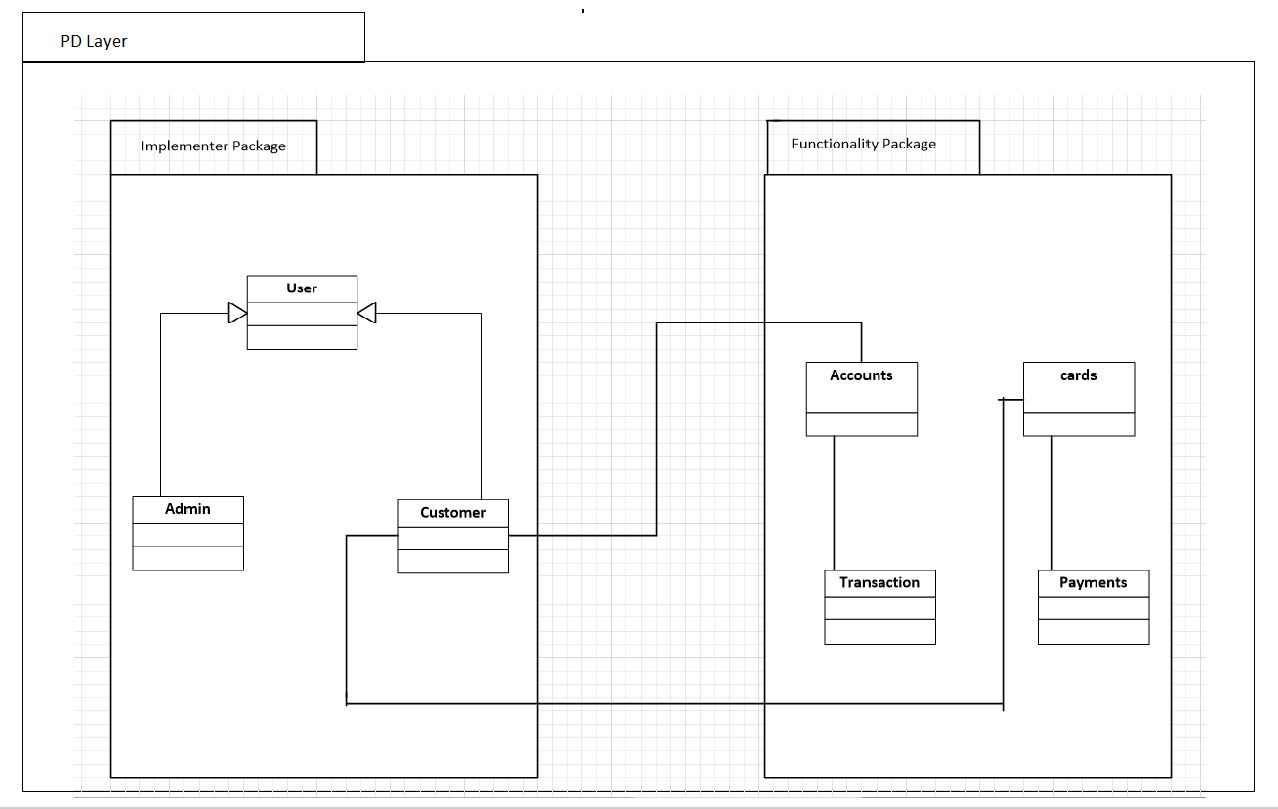
The final class diagram is slightly different with the initial one. Firstly, the User Interface class has been deleted in the diagram because we found that there is no need to put the User Interface class in the class diagram. Also in the Transaction class we added the attributes for From Account Number and To Account Number.



* 1. **Package Diagram**

A package diagram in the Unified Modeling Language depicts the dependencies between the packages that make up a mode

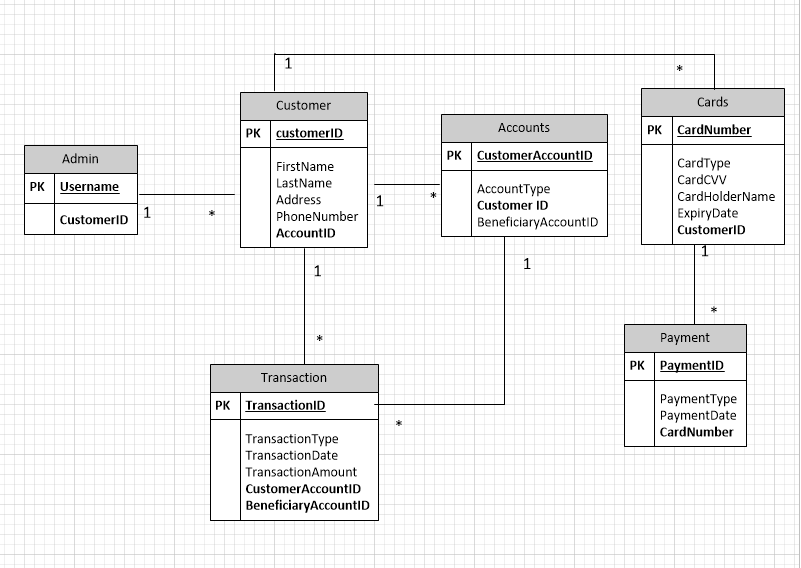
In our case, we have two packages – Implementer and Functionality. Implementer package consists of user, admin, customer classes. Functionality Package consists of accounts, cards, transaction and payments classes.



* 1. **Database Design:**

The database design is shown below. It consists of six tables:

* Admin
* Customer
* Accounts
* Cards
* Transaction
* Payment



* 1. **Data Access and Manipulation Diagram:**

The Data access and manipulation design shows the relationship between Problem domain classes and databases (Object persistance). The Problem domain classes use DAM class to connect with the storage database. The line pointed from database to class shows the corresponding relationship between classes and databases.

