Technical Design Document

Pecunia Banking System

Capgemini Technology Services

Airoli, Navi Mumbai,

India

Date: 10-11-2019

**Table of Contents**

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

Explanation of Use Case Contents………………………………………………………………………4

Admin

Employee

ER Diagram…………………………………………………………………………………………………………………………..5

DB Diagram……………………………………………………………………………………………………………………………………….6

Modules ….…………………………………………………………………………………………………………………………..7

CUSTOMER

EMPLOYEE

ACCOUNT

TRANSACTIONS

UTILITIES

LOANS

**INTRODUCTION**

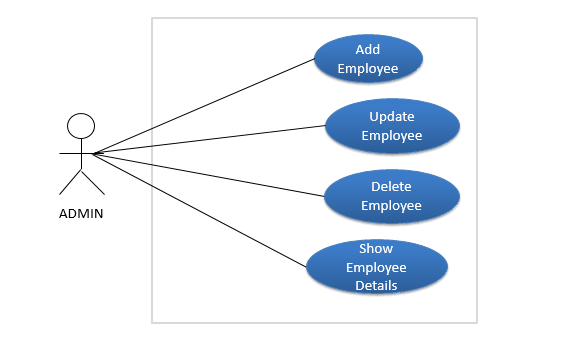
Pecunia₹ Finance Limited, established in the year 2015, is an Indian private sector [bank](https://en.wikipedia.org/wiki/Bank) headquartered in Bangalore and is functioning only in Bangalore, [India](https://en.wikipedia.org/wiki/India). In January 2019, [Reserve Bank of India](https://en.wikipedia.org/wiki/Reserve_Bank_of_India) (RBI) issued the license to Pecunia₹ Finance Limited, to carry on banking business.

They have one of the largest and most respected Wealth Management teams in India providing the widest range of solutions to individuals and employed professionals. As of December 30, 2018 Pecunia₹ has a net-worth of approximately ₹ 730 crore and micro-finance loans of ₹ 300 crore in its books.

It offers a wide range of banking products and financial services for retail customers through a variety of delivery channels in the areas of [personal finance](https://en.wikipedia.org/wiki/Personal_finance),  [life insurance](https://en.wikipedia.org/wiki/Life_insurance) etc.

**1. Use Case Diagrams**

1. Admin



Description: Pecunia admin logs in and can perform various functions.

Actor: Pecunia Administrator.

Preconditions : Admin has logged in and is now ready to perform the functions available to him.

Post Conditions: Admin can view, edit, add and delete employees. Admin has the full control of the operations being performed.

Flow:

1. Admin can add employees.
2. Admin can edit employee details.
3. Admin can remove employees.
4. Employee



Description: Pecunia employee logs in and can perform various functions like adding a customer, creating an account etc.

Actor: Pecunia Employee.

Preconditions: Employee has logged in and is now ready to perform the functions available to him.

Post Conditions: Employee can view and edit customer details. He can also create a new account, update account details, and close an existing account. He can even apply for loans and utilities services.

Flow:

1. Employee can create a new customer and view or edit customer details.
2. Employee can create a new account, update details and close an account.
3. Employee can perform credit and debit transactions from one account to another.
4. Employee can apply for loans and check eligibility criteria and reject applied loans.
5. Employee can apply for utility services viz. credit cards, debit cards and cheque books.
6. He can even block cards.
7. **ER Diagram**

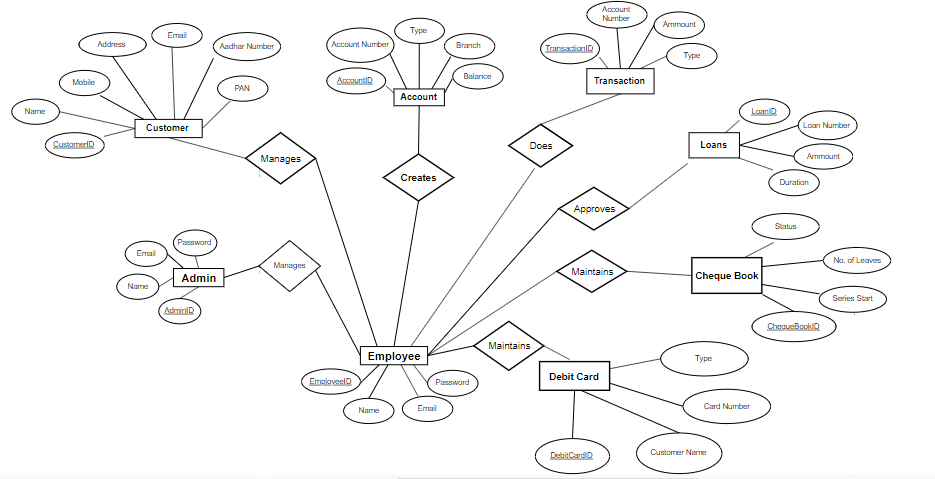
****

Fig: Entity Relationship diagram of Pecunia Banking System

Points to note:

1. 2 type of users: employee and admin.
2. Each user has a name, email ID, password, unique system ID.
3. Customer has fields like customer name, customer no, email id, PAN, aadhar, address with CustomerID as primary key.
4. Account has fields like account no, balance, branch, account type with AccountID as primary key.
5. Transaction has fields like account no, type and amount with TransactionID as primary key.
6. Debit Card has fields like Customer name, card number and type with DebitCardID as primay key.
7. Cheque Book has fields like series start, number of leaves and status as status with ChequeBookID as primary key.
8. Loans has fields like loan number, amount, duration with LoanID as primary key.

**3. DB Diagram**

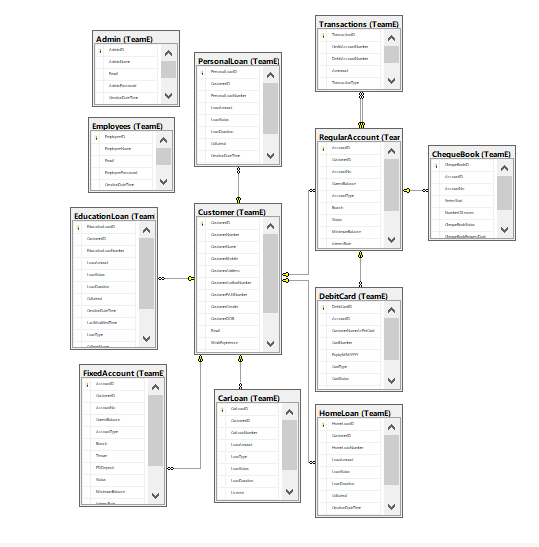


Fig: DB diagram of Pecunia Banking System

**3. Modules**

**1. Customer Module**

EPIC: Employee should add and update customers

User Stories:

* Employee should be able to view existing customers.
* Generate unique customer number on adding a new customer
* Employee should be able to update customer details.

Flow:

* Employee logs in. If the credentials are valid he is directed to the about page.
* Employee goes to customer menu.
* Here he can see the already added customers.
* He can view, edit or add customers.
* After validations on adding, customer is added.

**Class Diagram: Customer**

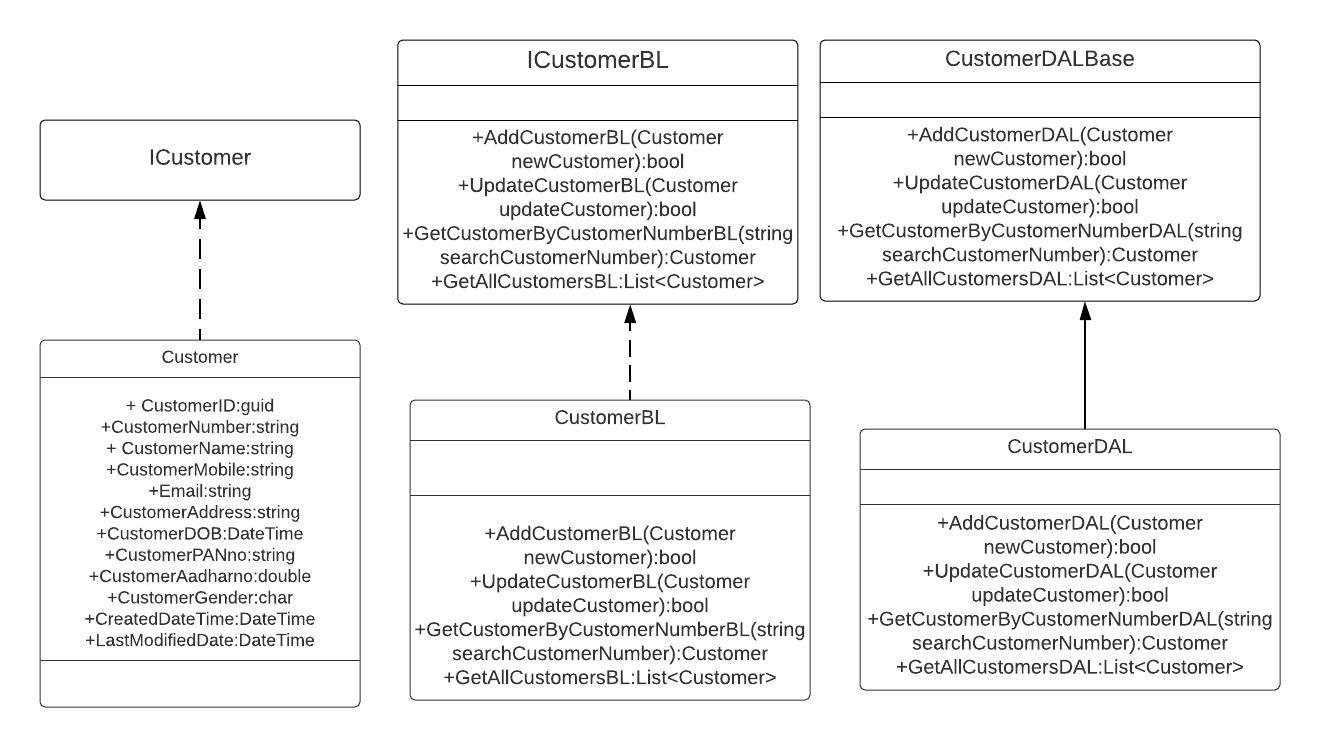


Fig: Class Diagram of Customer

**Sequence Diagram to add new customer**

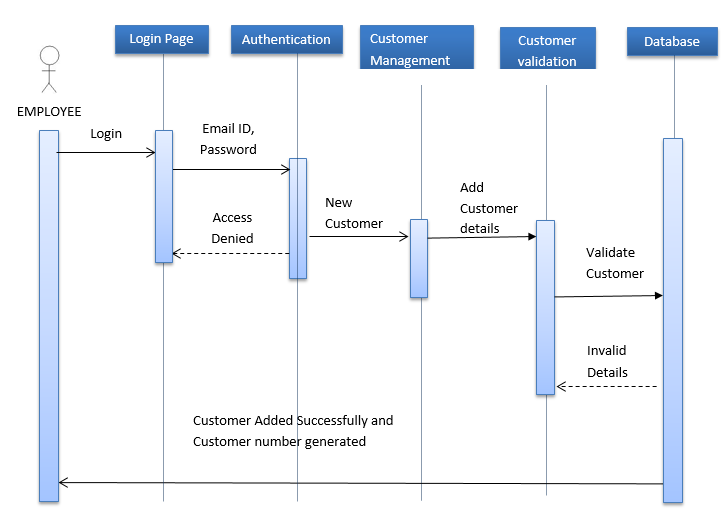
****

Fig: Sequence Diagram to add customer

**Activity Diagram to add new customer**

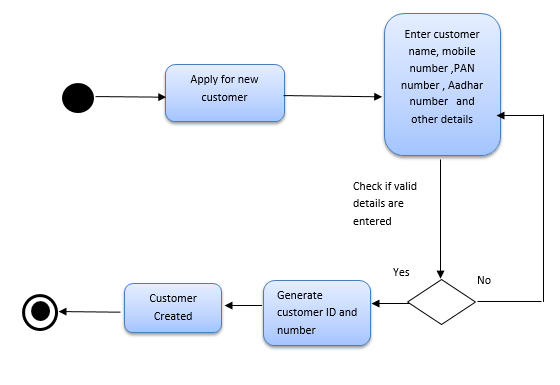


Fig: Activity Diagram to add customer

**Sequence Diagram to update customer details**

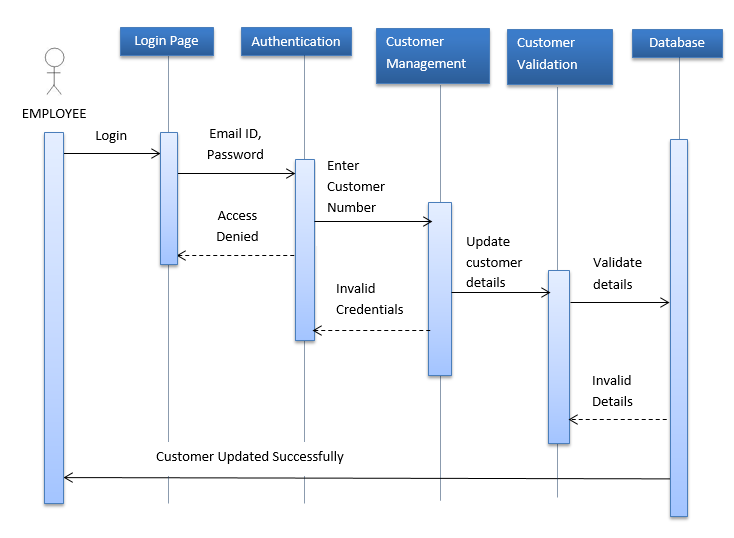


Fig: Sequence Diagram to update customer details

**Activity Diagram to update customer details**

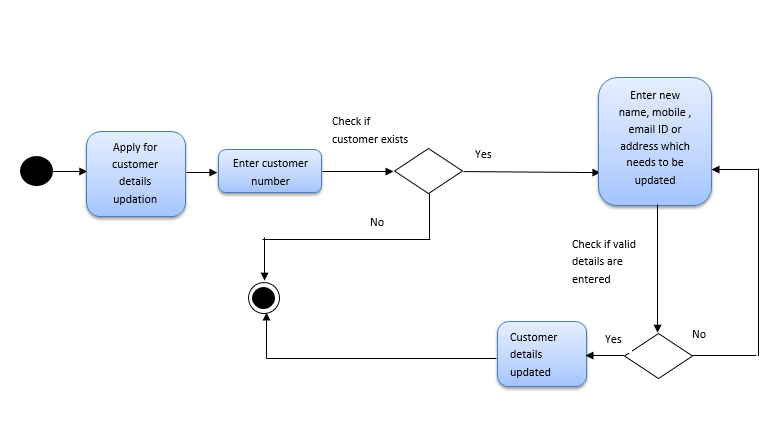
****

Fig: Activity Diagram to update customer details

**2. Employee Module**

**Class Diagram : Employee**

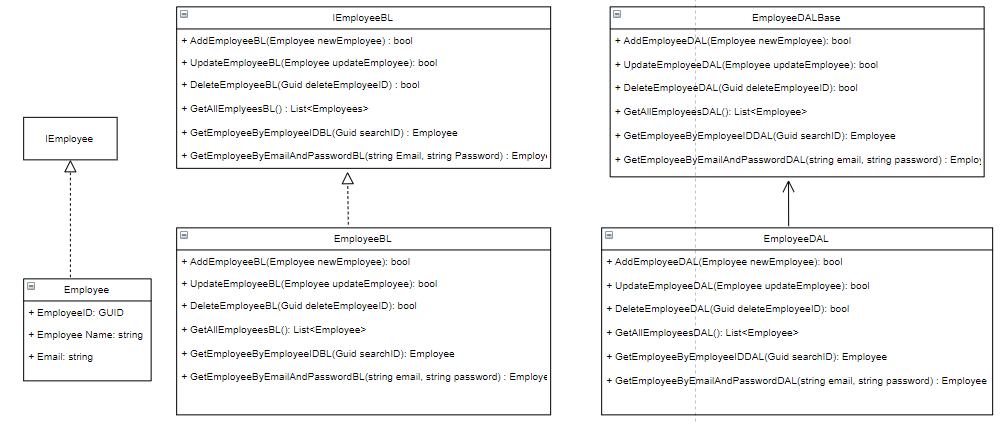


Fig: Class Diagram of Employee

**Sequence Diagram to add employee**

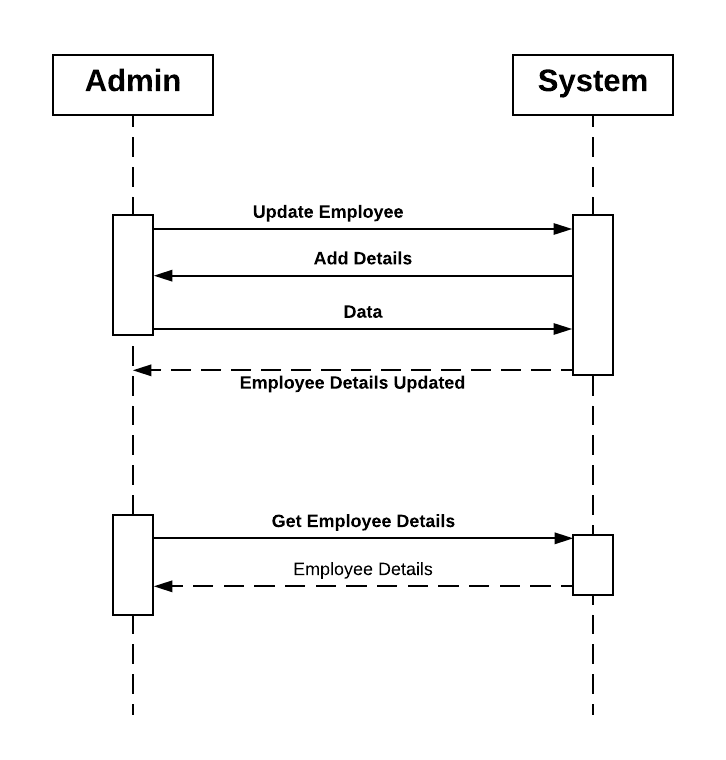


Fig: Sequence Diagram to add employee

**Activity Diagram for managing employee**

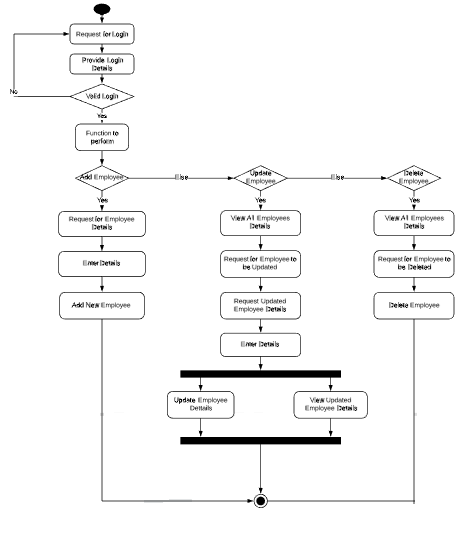


Fig: Activity Diagram for managing employee

**3. Accounts Module**

EPIC: Employee should be able to create a new account and perform related CRUD operations.

User Stories:

* Employee should be able to create account for a customer
* Generate account number
* Employee should be able to change the account type
* Employee should be able to change the home branch
* Employee should be able to close an existing account

**Class Diagram: RegularAccount**



Fig: Class Diagram of RegularAccount

**Class Diagram: FixedAccount**

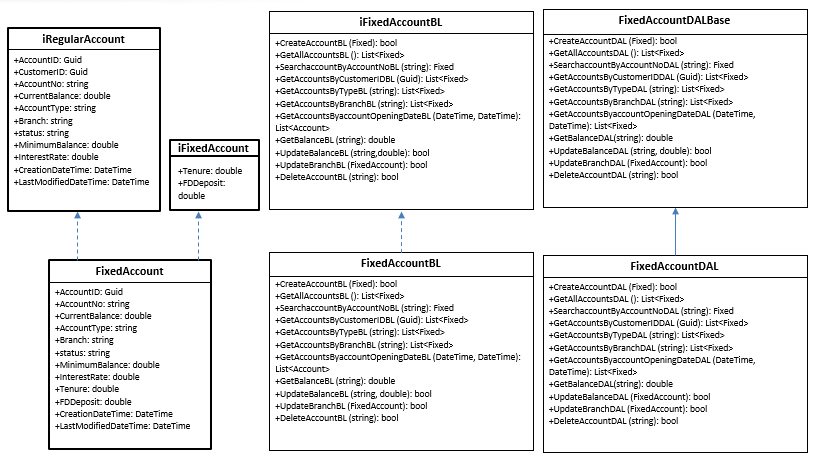


Fig: Class Diagram of FixedAccount

**Sequence Diagram for Account Creation**

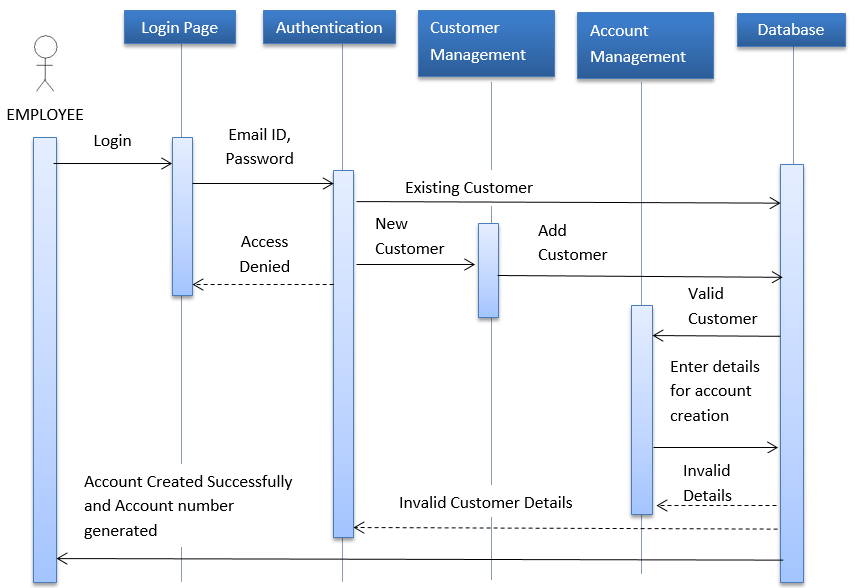
****

Fig: Sequence Diagram for Account Creation

**Activity Diagram for Account Creation**

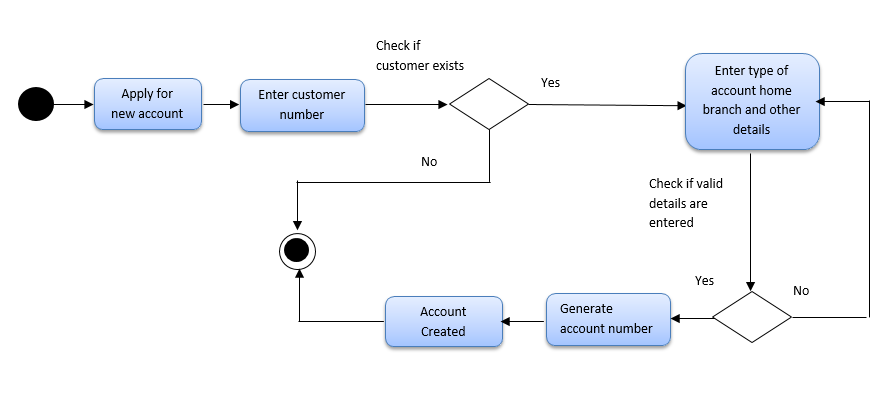


Fig: Activity Diagram for Account Creation

**Sequence Diagram for Account Details Updation**

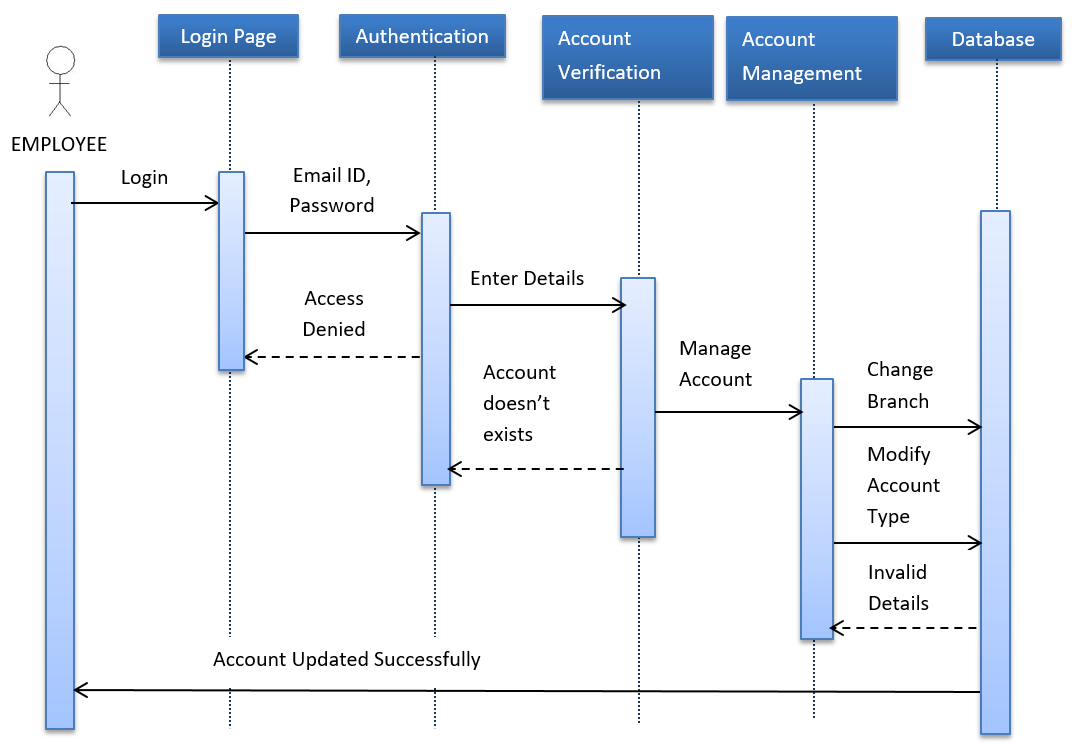
****

Fig: Sequence Diagram for Account Details Updation

**Activity Diagram for Account Details Updation**

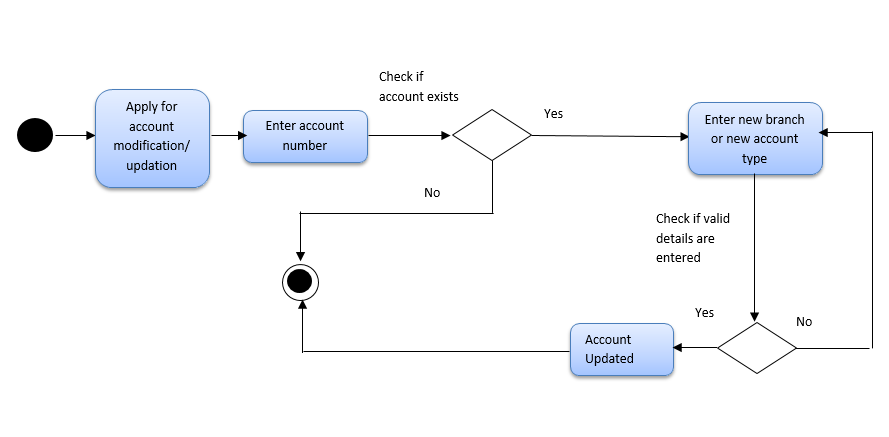


Fig: Activity Diagram for Account Details Updation

**6. Utilities Module**

EPIC: Employee should be able to apply for debit cards and perform related CRUD operations.

User Stories:

* Employee should be able to apply for debit card
* Generate Card number
* Employee should be able to update the card type
* Employee should be able to block a card

**Class Diagram: DebitCard**

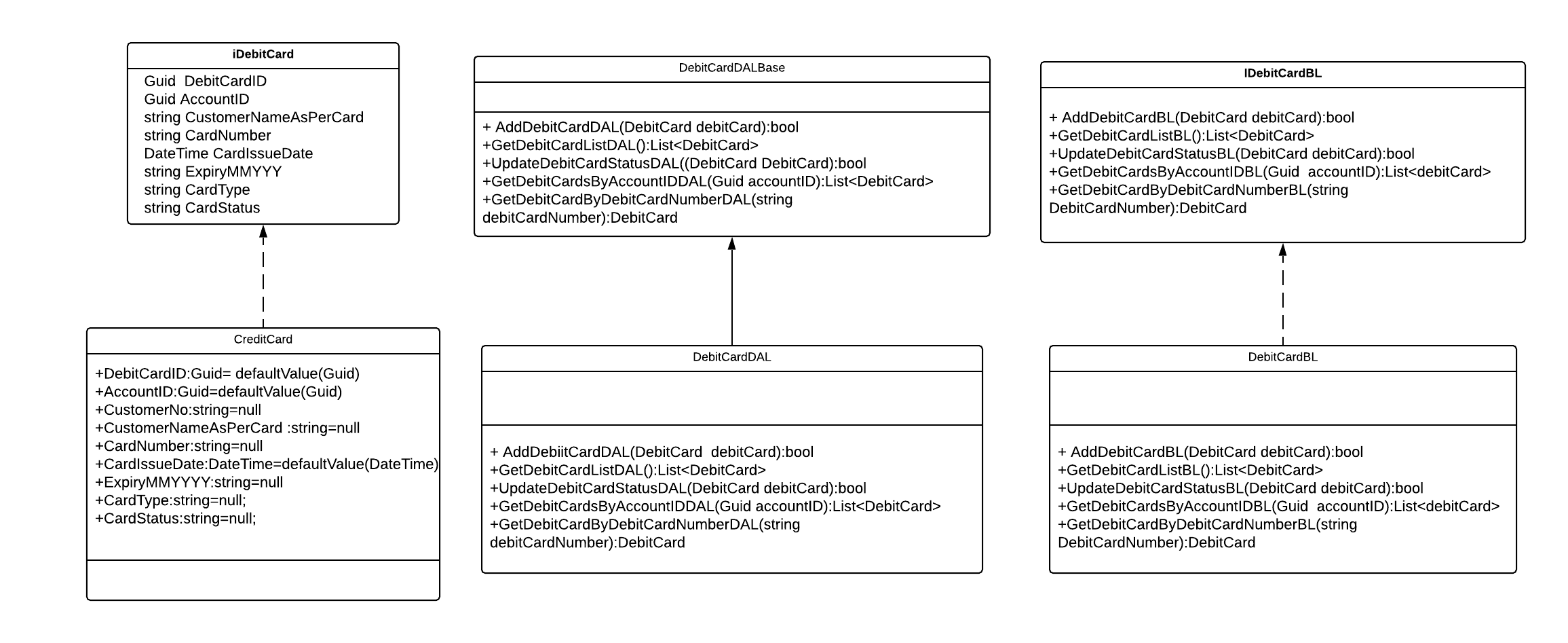


Fig: Class Diagram for DebitCard

**Sequence Diagram for DebitCard Issue**

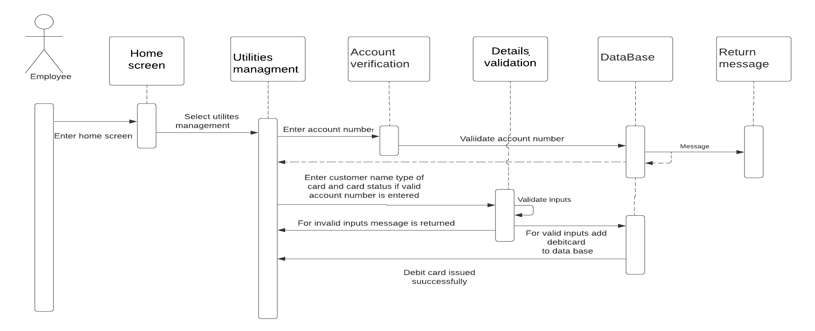


Fig: Sequence Diagram for DebitCard Issue

**Activity Diagram for DebitCard Issue**

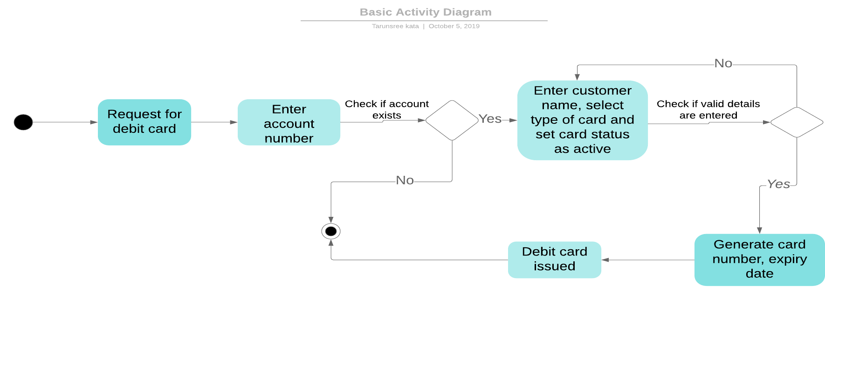


Fig: Activity Diagram for DebitCard Issue

**4. Transaction Module**

EPIC: Employee should be able to perform credit and debit transactions between 2 accounts.

User Stories:

* Employee should be able to perform credit transactions
* Employee should be able to perform debit transactions
* The current balance should be reflected in both the accounts

**Class Diagram: Transaction**



Fig: Class Diagram for Transaction

**Sequence Diagram of Transaction**

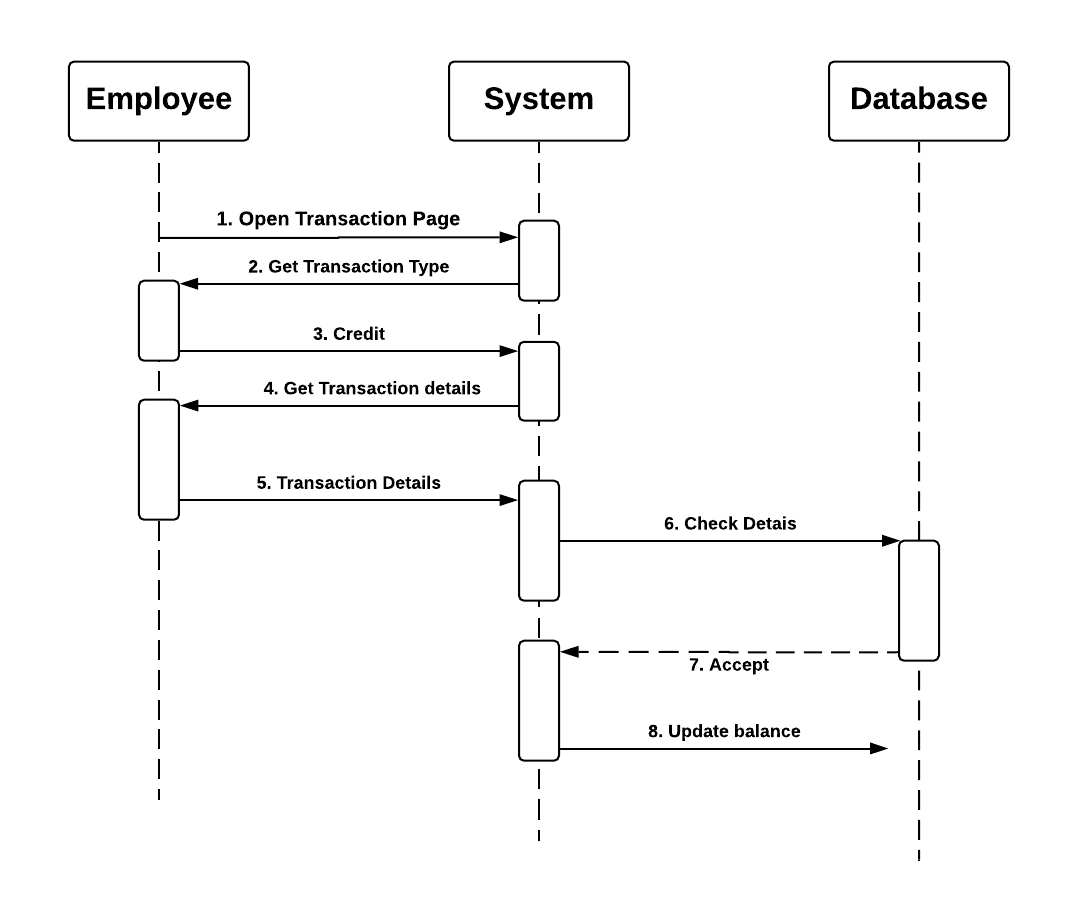


Fig: Sequence Diagram of Transaction

**Activity Diagram of Transaction**

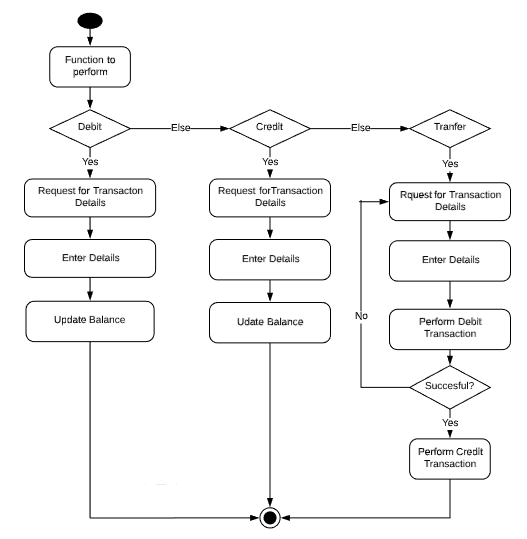


Fig: Activity Diagram of Transaction

**7. Loans Module**

EPIC: Employee should be able to apply for loans and perform related CRUD operations.

User Stories:

* Employee should be able to apply for home loan and car loan.
* Generate loan number.
* Employee should be able to check the eligibility of applied loan.
* Employee should be able to reject ineligible loan applications.

**Activity Diagram for applying new loan**

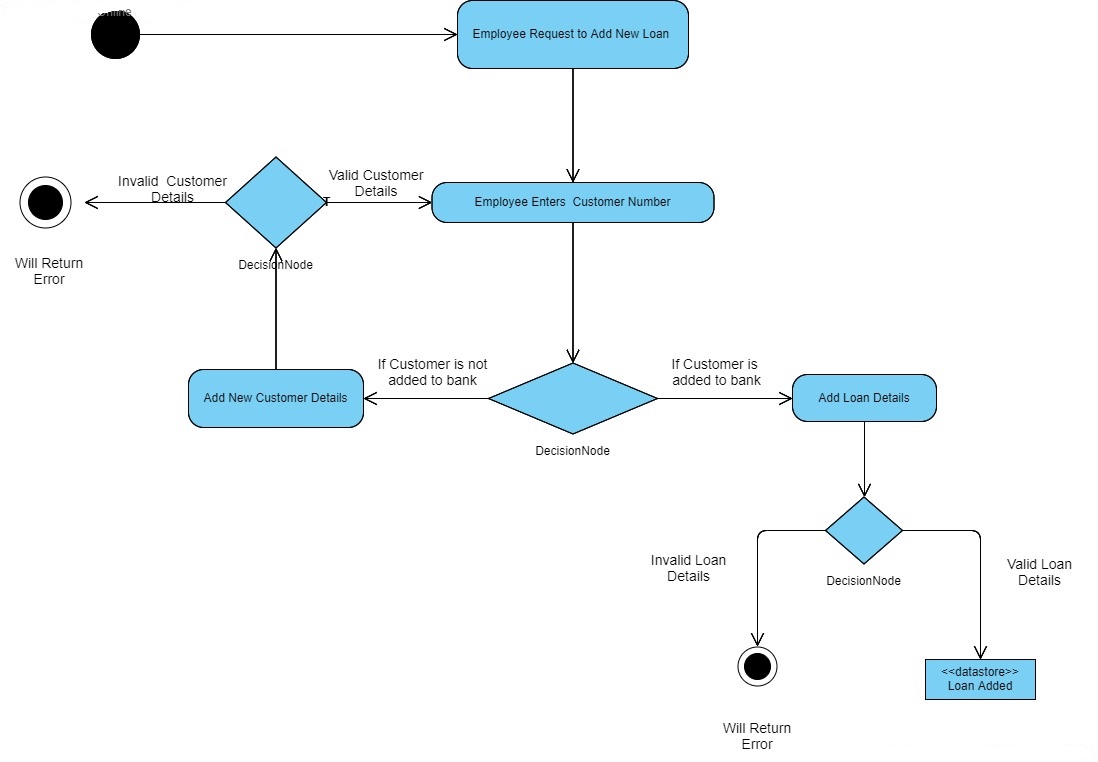
****

Fig: Activity Diagram for applying new loan

**Activity Diagram for updating loan details**

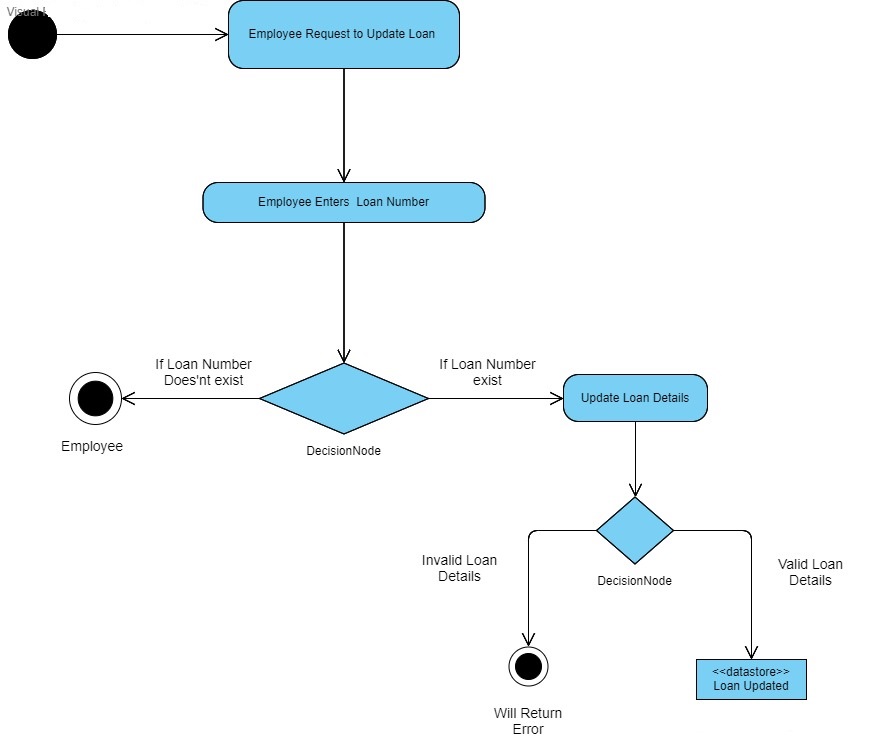
****

Fig: Activity Diagram for updating loan details