

MART MANAGEMENT SYSTEM

Requirement Analysis

Prepared by

Muhammad Danish

Hakeemullah

Rafaqat Ali

Table of Contents

1. Introduction
 - 1.1.Purpose of the system
 - 1.2.Scope of the system
 - 1.3.Overview
2. Current System
3. Software Requirement Specification
 - 3.1.Functional Requirement
 - 3.2.Non-Functional Requirement
 - 3.3.System Models
4. References

1. Introduction

1.1. Purpose of the system

Mart Management System (MMS) is a standalone system that helps the owners of mini mart to manage the data and information of their mini mart. It provides the solution for replacing the manual work done to manage the daily records of mini marts

1.2. Scope of the system

A full-featured management system is needed for mini marts to manage all the stored data and information systematically and efficiently. MMS application is developed for mini marts to manage inventory, staff, supplier and finance records.

1.3. Overview

A fully functional MMS application that featured with user friendly is provided at the end of the development process. With the developed application, users can manage the business of mini mart systematically.

2. Current System

Current system can be divided into two component on the bases on working. The initial component which is responsible for creating account, change password and authenticating an account with privilege access as on the bases on type of user. The final component which provide access on features on the bases of authentication type, now this component further more can be divided into two subcomponent. One subcomponent is responsible for searching, adding, deleting and finalizing customer data with slip and the other subcomponent for creating list or updating list for items present in store.

In other words:

User maybe a worker or admin and both have different privileges, when they are authenticated, they have less or full access. Worker can only insert or search customer data and admin can access all these stuff plus can change item list and other stuff.

As it has some problems and need some updates like admin access should be increase and need to be managed well which not present in current system, it will be updated or evaluated as per customer demand as customer required.

3. Software Requirement Specification

3.1. Functional Requirement

- Sign Up:
The user will create account with authentication of admin key
- Sign In:
Authentication start with two different login availability user and admin
- Manage all Customer Record:
Show total price and real date time and searching option for customer record.
- Customer data inserted by code:
We have to put only specific code number and quantity in order to insert item in database with add button
- Data Item list creation:
List page contain list of the item with functions of add values, delete, update and search
- Delete Record (effective):
Delete the current mistake of insertion and delete record of the particular customer

3.2. Non-Functional Requirement

- Manageability:
Differentiate/ Separate feature for User and Super User
- Privacy and Security
- Usability:
Easy to Search, Easy Update feature (Special feature by own when default function was not working)
- Availability:
PDF creation
- User Friendly:
Mostly background process and less features on screen

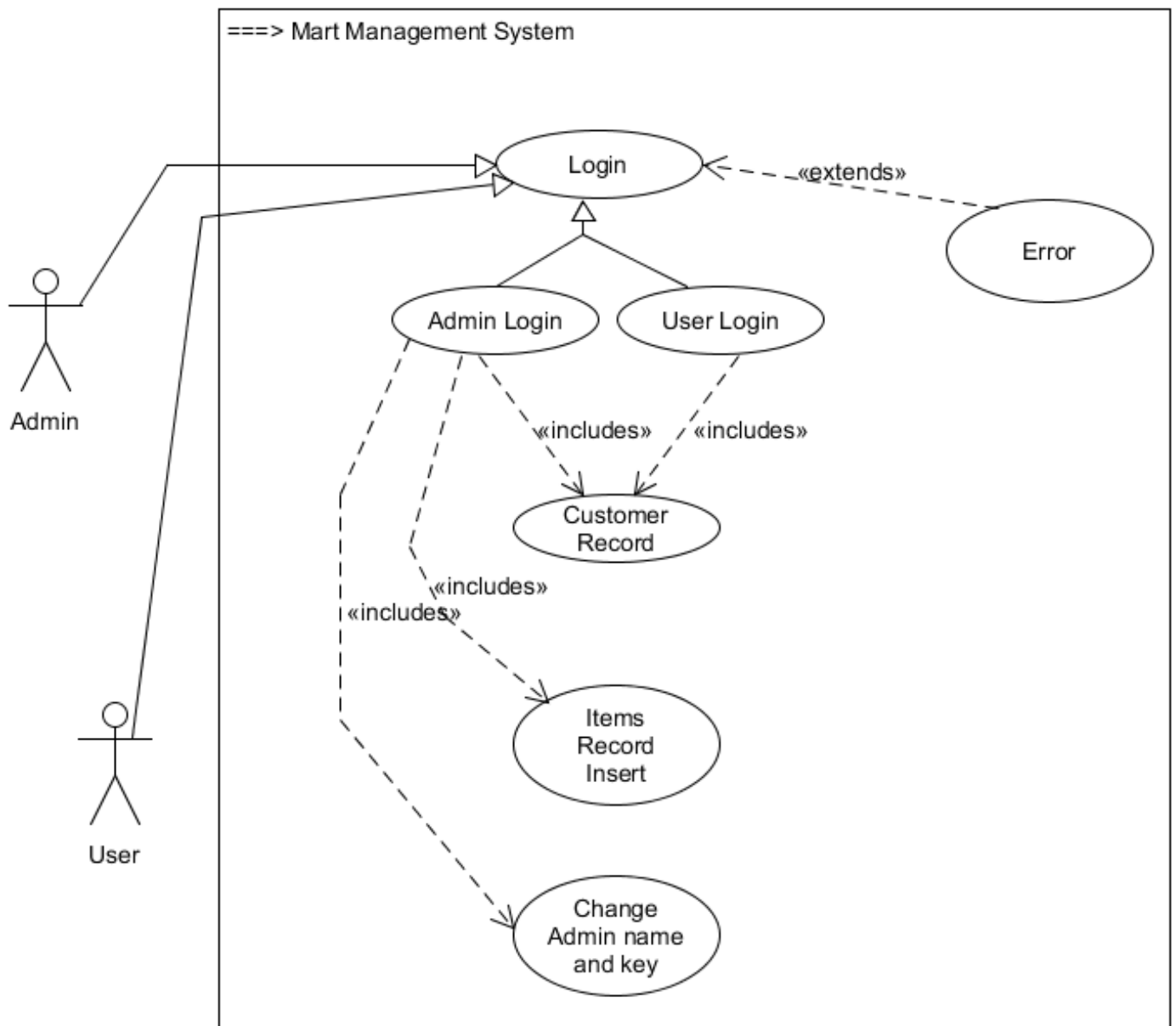
3.3. System Models

3.3.1. Scenarios

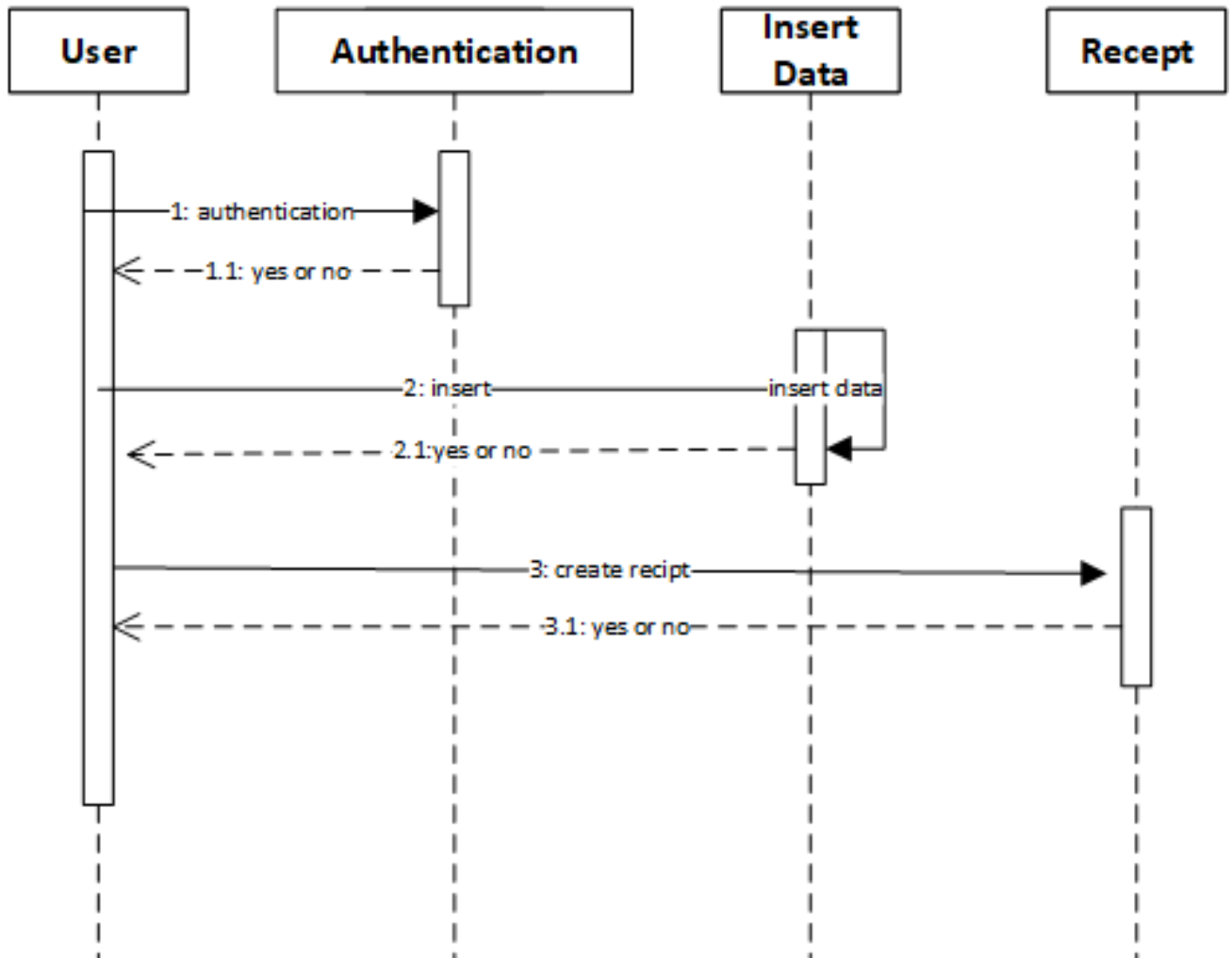
As in agile process software is updated iteratively whenever a new functionality is needed to include or need to change functionality.

Evaluation is a part of software because it need to be evolve because often business requirement changes as the time passes. Software started to use when it is seemed to cover a basic or most required or necessary part of software functionality and evolved as the time passes. In business time change so quickly and here we need rapid application development so as hurry as possible they want solution is to be in use and they can take advantage of it.

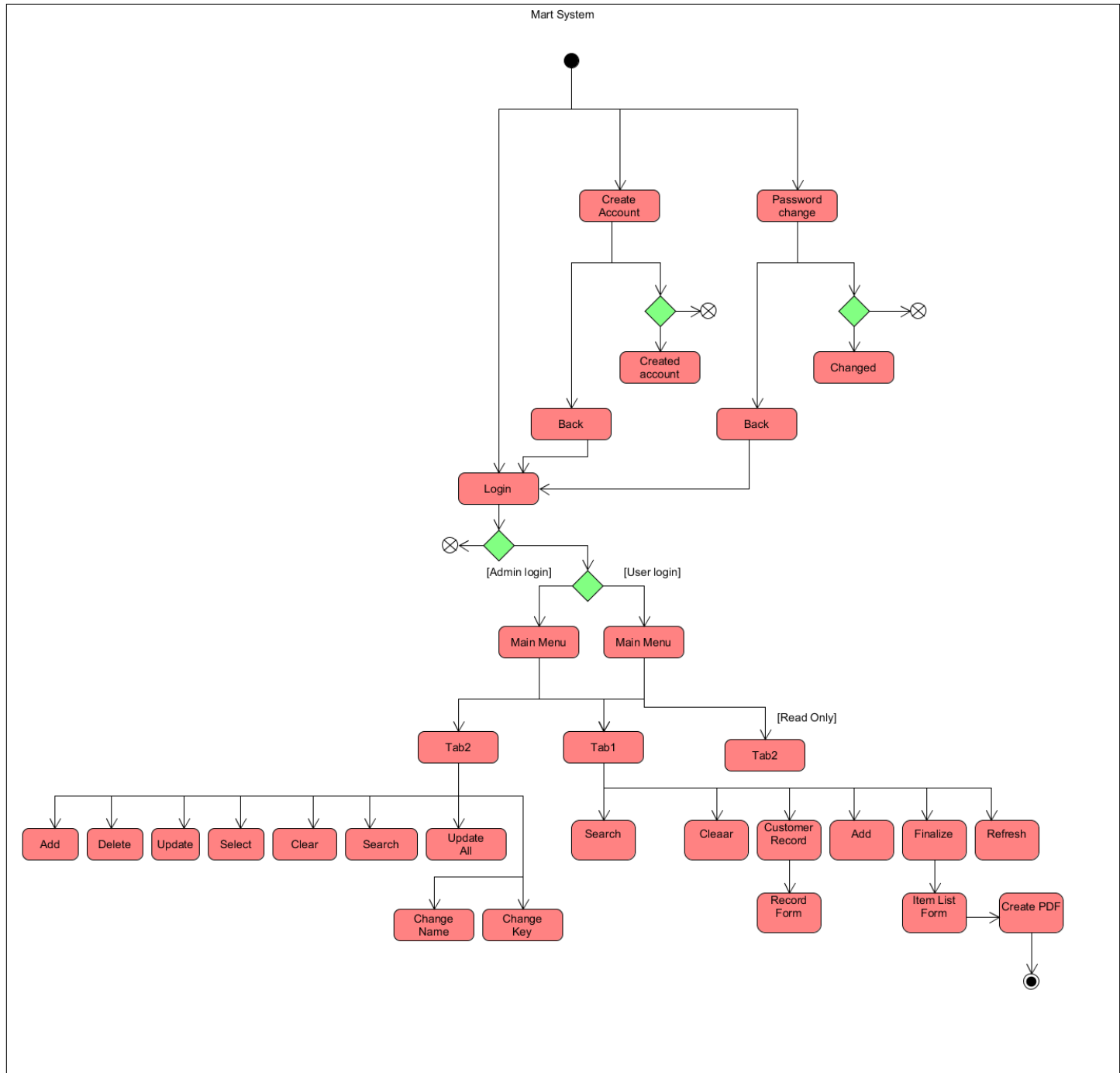
3.3.2. Use case Diagram:



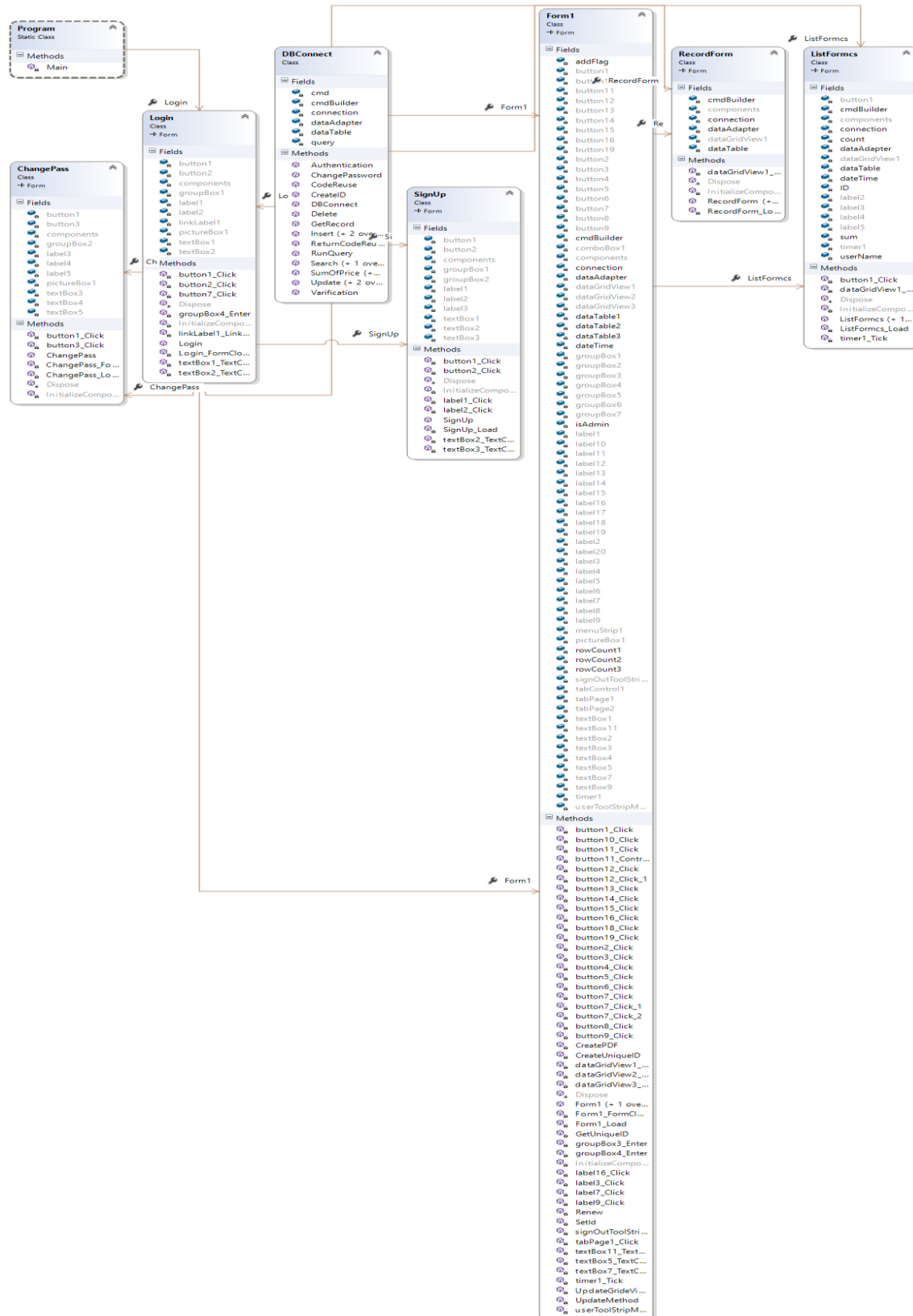
3.3.3. Sequence Diagram:



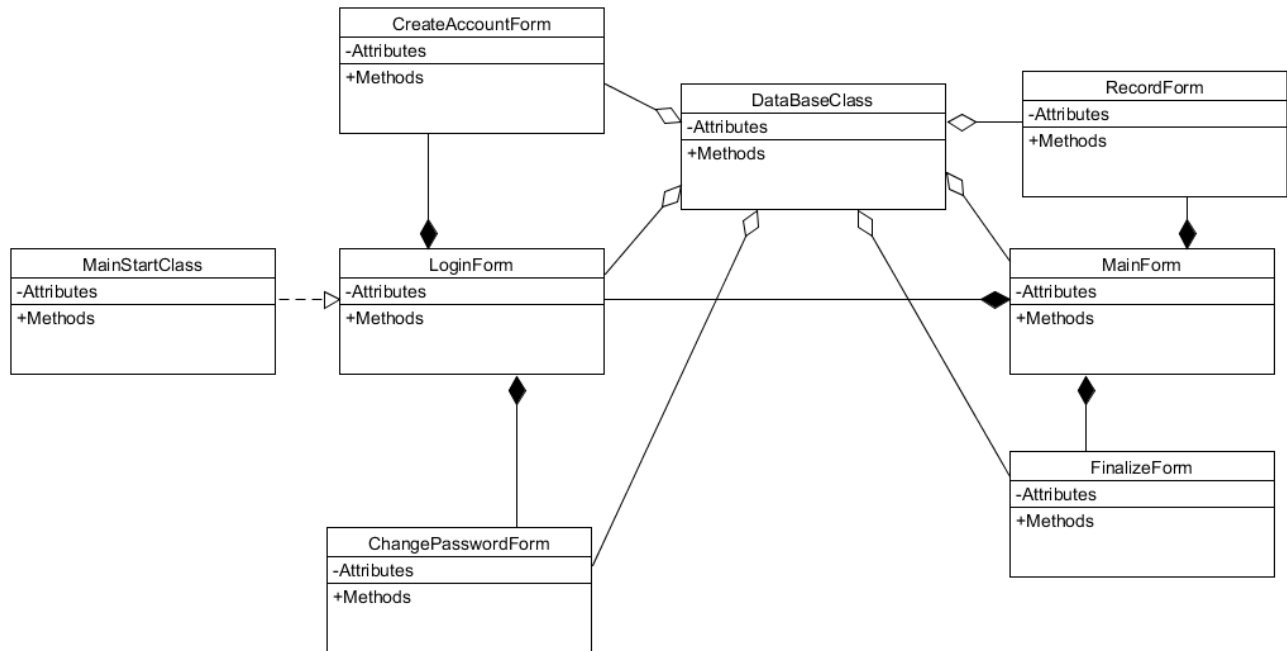
3.3.4. Activity Diagram:



3.3.5. UML Class Diagram



3.3.6. UML Class Relation Diagram:



4. References

- [1] <https://afterlogic.com/mailbee-net/docs-itextsharp/>
- [2] <https://dev.mysql.com/doc/connectors/en/>
- [3] <http://library.utem.edu.my:8000/elmui/index.jsp?mod...>