EATIFY

Most delicious Android app, it's worth to try :)

150114002 - Ayşenur Yılmaz

150109009 - Ceren Birinci

524118023 - Mustafa Zortul

1. Introduction

Purpose of this project is to provide a mobile platform to people who loves to cook and wants to interact with others to share the secret recipes. System will allow to users to teach and learn each others.

This design document explains detailed specifications of Eatify before implementation phase.

1.1 Purpose

The objective of this document is introducing a detailed explanation of the designs of Eatify. First of all, this document is conformed for implementation phase which helps programmers to understand design clearly. We can see this document as a guideline for them. Also, this document can be used by designer to upgrade or change existing design.

1.2 Scope

This document includes description of the software architecture of Eatify. Additionally, document contains a class diagram that shows how program should be implemented.

1.3 Definitions, Acronyms and Abbreviations

- IDE: Integrated Development Environment
- OS: Operating System

2. Design Considerations

2.1 Assumptions

The user group of our system is assumed to have knowledge of using mobile devices such as smartphones and tablets. Additionally, application is compatible with devices which have Android OS. That's why, users shall be aware of Android OS.

2.2 Constraints

The system is implemented using Java, and shall need Android OS 7.0 or a higher version of it.

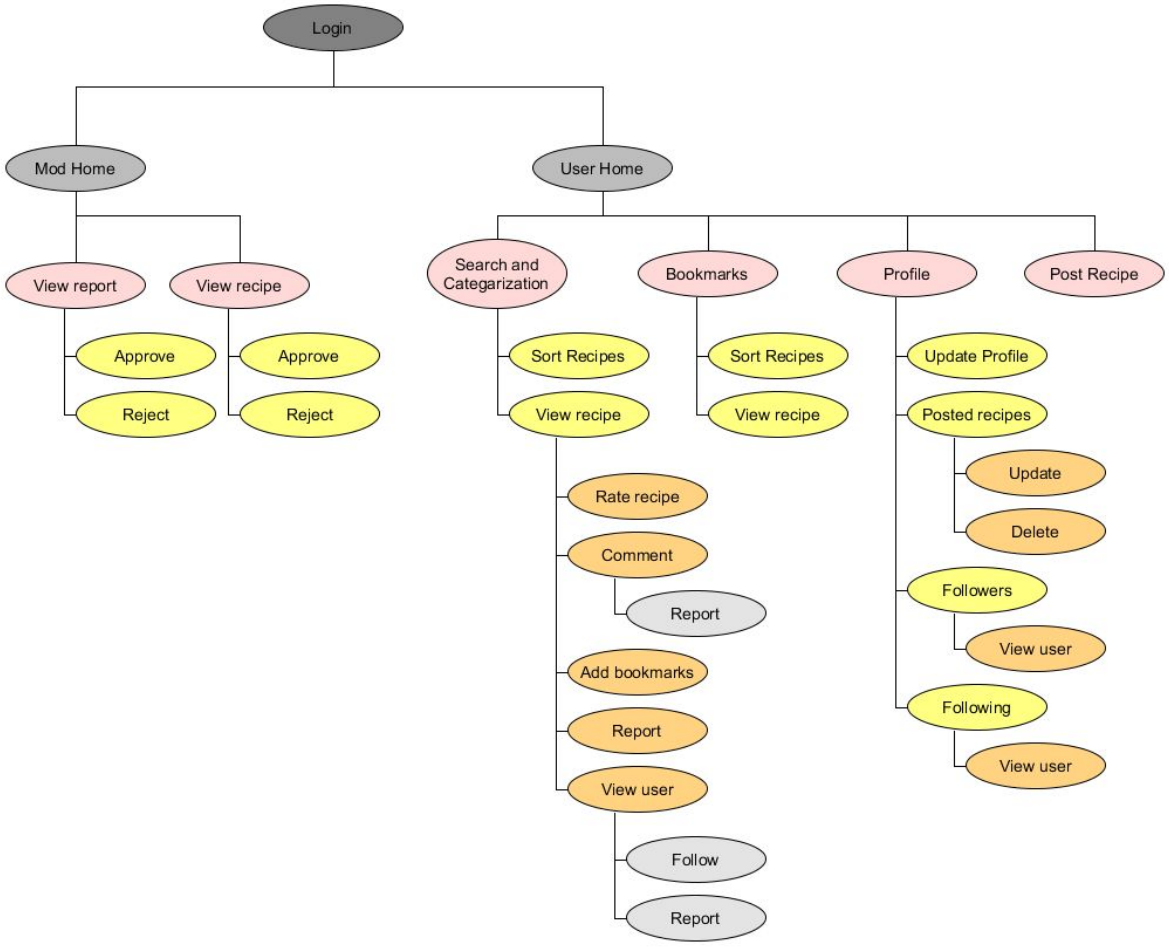
2.3 System Environment

Application will be developed using Android Studio IDE. Each user shall need an Android compatible mobile device, smartphones or tablets.

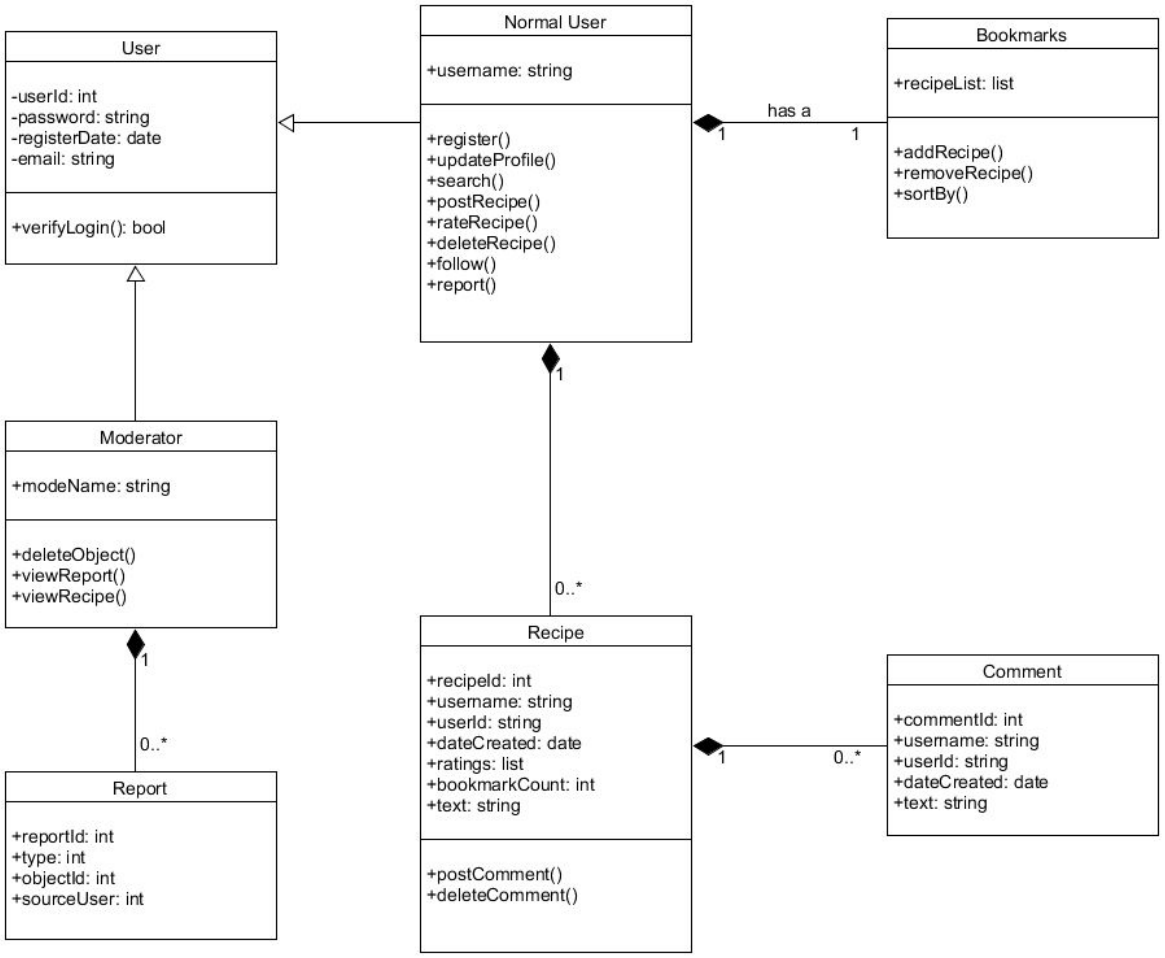
3. System Design

3.1 System Decomposition

Decomposition of main functionalities of the system are shown below. “View recipe” functionality below “Bookmarks” has the same sub-functionalities of “View recipe” below “Search and Categorization”.



3.2 Class Diagram



3.2.1 Explanation of Classes

Class attributes and methods are described below.

- User: Parent class of *Normal User* and *Moderator* classes. Basic attributes like a unique id, password, date of registration and email informations are stored in this class. Class also has a method for checking login credentials.

- Moderator: Only attribute is moderator name which is public name of moderator. Methods are delete object(recipes, comments or users), view report(view a report and the related content), view recipe(view a freshly uploaded recipe for reviewing). Moderator accounts don't need registration, they are created by admins.

- Report: Reports are issued by users. Class has a unique id of report, type(0,1,2 for user, recipe and comment, respectively) and id of the reported object and id of source user as attributes. This class has multiplicity relationship with *Moderator* class. Each *Moderator* can have desired number of *Report* objects and *Report* object can't stay alive without *Moderator* object.

- Normal User: Only attribute is username which is user's visible name in the platform. Methods are register(invoked initially), update profile, search(for both of categorization and search features), post recipe(owned by user), delete recipe(owned by user), rate recipe(owned by other users), follow and report(users, recipes and comments can be reported). It's worth mentioning that update profile method gets called once within register method.

- Bookmarks: Only attribute is recipe list which is an array that basically holds recipe ids that are bookmarked by user. This class has composition relationship with *Normal User* class. Each *Normal User* object has a *Bookmarks* object and a *Bookmarks* object can't stay alive without a *Normal User* object. It has methods for adding, removing and sorting recipes by desired parameters.

- Recipe: Attributes are a unique recipe id, username(username of poster of recipe), user id(user id of poster of recipe), date of creation, ratings(list of value, user id pairs), bookmark count(number of users added that recipe to their bookmarks) and text(a prestructured string that holds text information of recipe). There are two methods, one for posting and one for deleting comments. This class has multiplicity relationship with *Normal User* class each *Normal User* can have desired number of recipes and *Recipe* object can't stay alive without *Normal User* object.

- Comment: Attributes are a unique id, username(username of poster of recipe), user id(user id of poster of recipe), date of creation and text(a string that holds the actual comment data). This class has multiplicity relationship with *Recipe* class. Each recipe can have desired number of comments and *Comment* object can't stay alive without *Recipe* object.

4. Workload Table

Below table shows the workload on each member of the group. In general, we have to say that all documents those submitted by us prepared all together, so it is kind of hard to say who did what actually. For every submission we had two or three times meetings and that's why we are all wrote these documents.

Name	No	Sections
Ayşenur Yılmaz	150114002	Cover, 2.1, 2.2, 2.3, 3.2.1
Ceren Birinci	150109009	1.1, 1.2, 1.3, 3.1
Mustafa Zortul	524118023	3.1, 3.2, 3.2.1