Gyalpozhing College of Information Technology
Royal University of Bhutan

SOFTWARE REQUIREMENT SPECIFICATIONS

Project Topic: Info Saver App

Submitted by:

Name: Karma Choden

Enrollment No: 12190056

Introduction

After the mobile application was introduced in the market, mobile applications have become one of the main ways people communicate, shop, plan their lives, play, and even work. Mobile marketing provides a wonderful opportunity for marketers to engage with customers and candidates anytime, anywhere. This ability for marketers to be "constantly connected" has been made possible largely due to the emergence of mobile applications. The Info saver mobile application will be introduce mainly to manage the reading list of the articles, images and videos form the Internet. The people demand is that they want an easy and reliable ways to perform their tasks with wasting time and their resources. This mobile application exactly does that.

Purpose

The purpose of this project is basically to provide a platform to save the information that they grab from Internet which is useful or referenced in future. This project will also help the users to save their time without having to search for the sources that they had referred before which is in need in their present time. With this app, the people can save the resources in their account and just click on the link of that particular source whenever they are in need of information related to that particular resources. The people using this mobile app can save their resources like mobile data from downloading those files and software which are not often used or referred.

Scope of the Project: The scope of the project is divided into two types and they are:

1. **System scope:** This project will allow the users to add the important links of the websites or the files, images or the videos, and the articles that they come across while browsing. The users have to log in to this mobile app in order to keep or store their list of knowledge. The users can add the knowledge to their list and they can also make that information their favorite to find the information easily. The users can also delete the information from their list if they don't want it anymore.

2. **User scope:** This project aims to provide a proper and simple platform in Gyalpozhing to put their knowledge in a mobile app.

Functional Requirements

Users

- **Registers and login:** In order to save the needed resources, the users have to register in this mobile app and then they have to login to use the functionalities of this mobile app.
- **Adds the information or knowledge:** The users can add or store the information.

- **Mark as favorite:**_The information can be made as the use’s favorite to help them organize their information or data.
- **Delete from their list:**_The users can delete the information form their list if they don’t want it.
- **Manages Users:** The admin manages the users.

Non-functional requirements

- **Security of the System:** From the mobile app or the system, users expect the security of the system. Everyone uses only the secure system rather than insecure ones. So, the security of the system is one of the nonfunctional requirement.
- **User-friendly:** This mobile application will be user-friendly means that every user will be able to uses it without any problems as it will be simple to use.
- **Responsiveness:** This system will be responsive to every types of screens making it more feasible to use it without any collapse and unstructured

Software Requirements:

- Operating System: Windows 10/8 (64-bit) or ubuntu 20
- Android SDK: Android SDK version 16
- Android Studio: Version 3.0 and above
- Java Version: JDK 8 (Java Standard Edition Development Kit)
- Database: Firebase version 19.2.1

Hardware Requirements:

1. **Developers requirements:**
    - Processor: Core i3 and above
    - Disk space: 3GB minimum(8GB recommended)
    - RAM: 4GB and above

2. **User’s requirements:**
        - 2GB RAM

SYSTEM DESIGN

1. **ERD(Entity Relationship Diagram)**

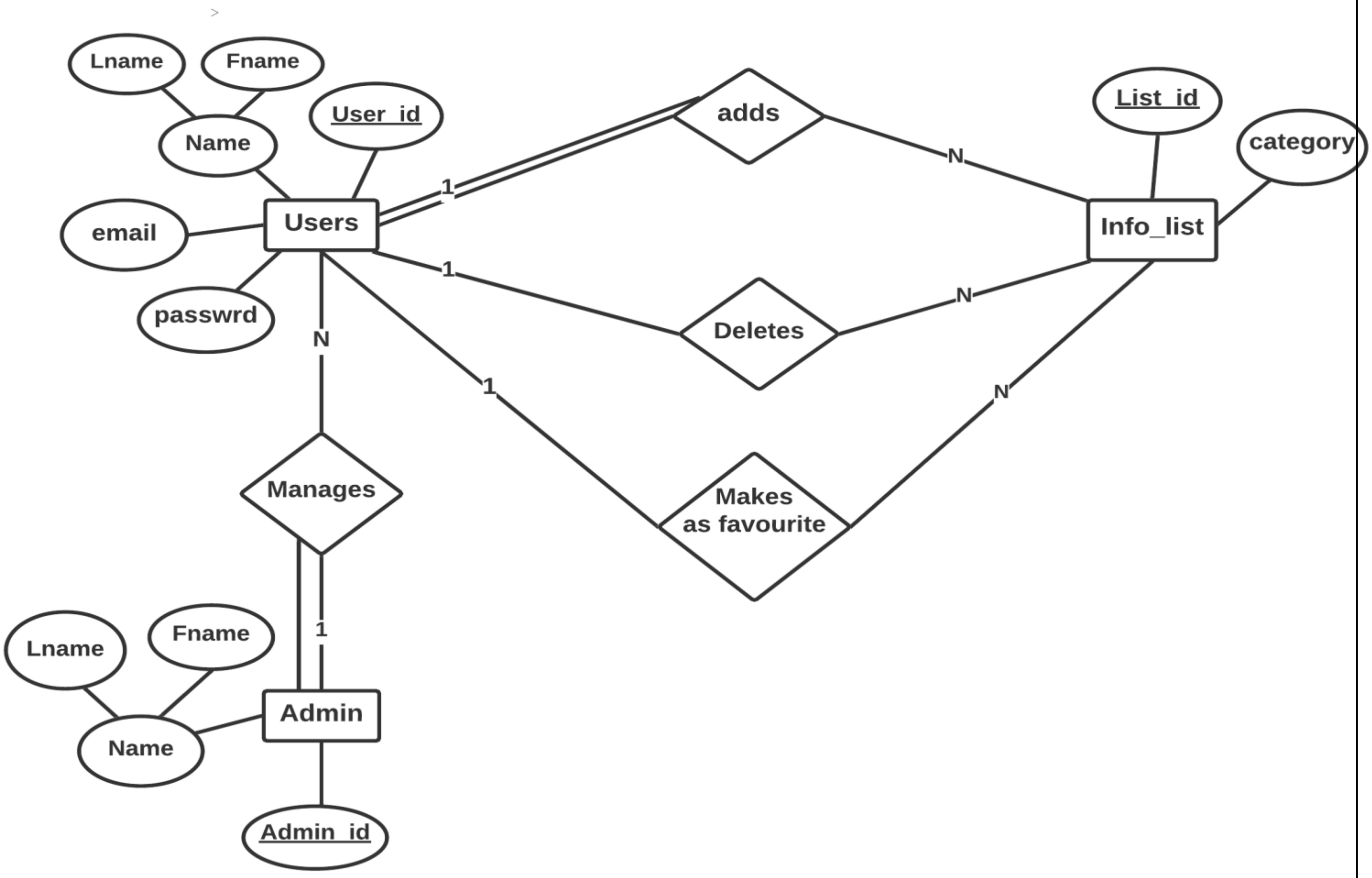


Figure 1

There are three entity with some attributes namely Users entity with name, email, password and User_id(Primary key) attributes, Info_list entity with List_id(Primary key) and category as their attributes and lastly the Admin entity with name and Admin_id(Primary key) as attributes. There are three relation between Users entity and Info_list entity and they are the Users can **add** the information in their list, they can also **delete** the existing information from their list and lastly, the users can make the information as **favourite** so that they can find it easily with one to many cardinality ratio. Since, one user can add, delete and mark as favourite many information in their list. And an information is added or deleted or mark as favourite by one user. In Add relation, the users takes total participation as every users adds at list one information in their list where as not every information in list is added by the users. The relation between the Admin and Users entity is the Admin manages the Users with one to many cardinality ratio as one admin manages many users and a user is managed by one admin. The Admin entity totally participate in this relation as admin manages at least one users and not every user is managed by admin.

2. Relation Schema

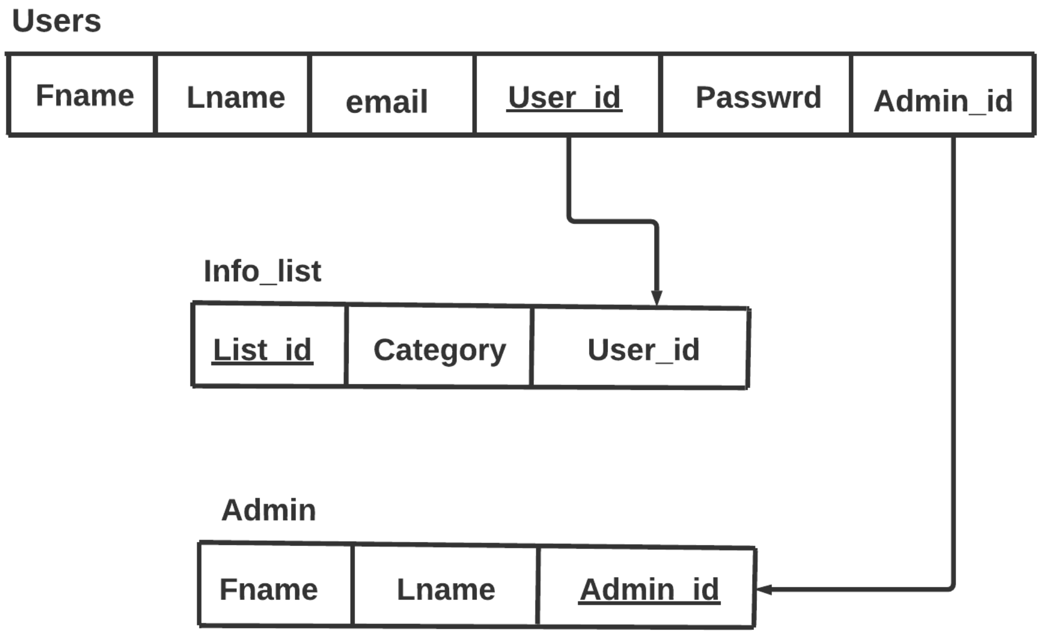


Figure 2

This relation schema is derived from the ERD. The arrows show that the particular is a foreign key in that schema which is brought from the other entity which is pointed by arrow. Like in Users schema, the Admin_id is foreign key brought from Admin schema which is shown by an arrow pointing towards Admin schema. Same like that the User_id is a foreign key which is the primary key of Users schema. The arrows points towards Info_list schema showing that the User_id is the primary key of that schema. The foreign key help to gather the details of which user had added or deleted or marked their info as favourite of the particular information in the list and id admin had managed any users then that details also can be stored in database for future reference.

3. Sequence Diagram

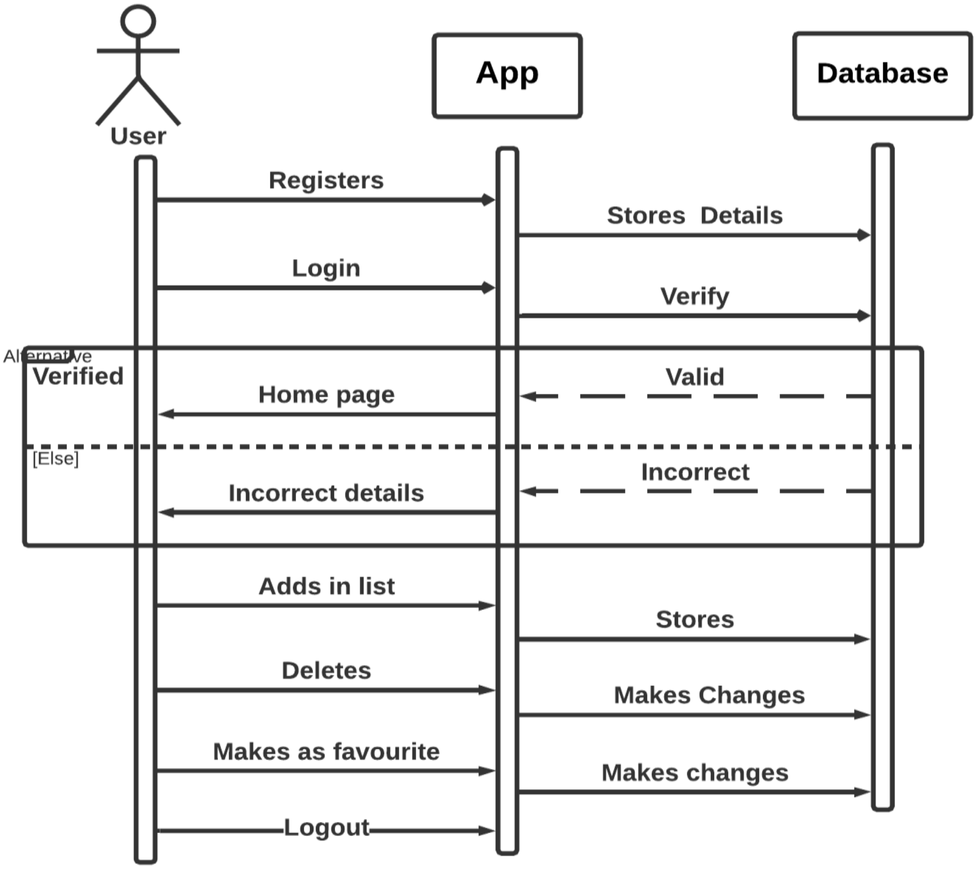
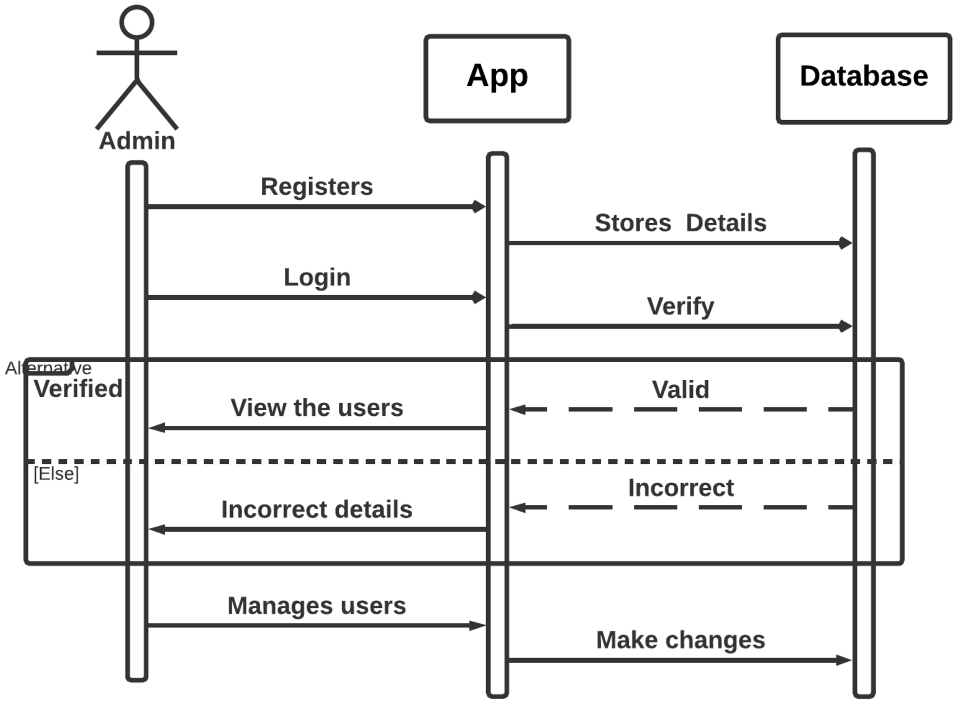


Figure 3

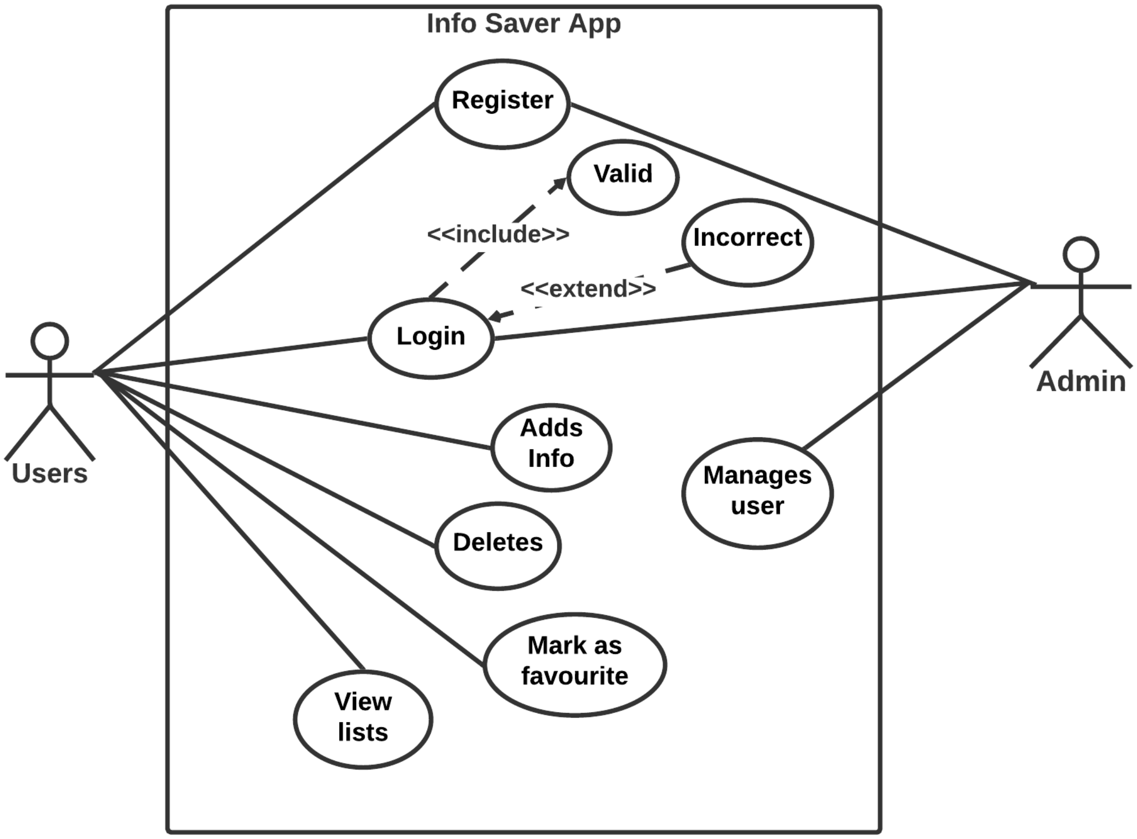
This is the sequence diagram for users. This diagram shows the sequences of the process on how this app will work. First the user register to this mobile app and their details are stored in the database. Then the user login to this mobile app and the database verifies whether the details are correct or not. If the details are correct than the database response valid to the app and the app shows the home page to the users but if the details entered by the user is incorrect than the user will again have to login with correct inputs. After the successful login to this mobile app, the users can add, deleted and mark the existing information as favourite and the changes are made in database too. Finally the user can logout.

Sequence Diagram for Admin



This is the sequence diagram for admin. The admin also registers and their information are stored in database and the admin have to login to this mobile app. When the admin login in this mobile app, the login information are cross checked with the details is database and if the login information is correct, then the admin can view the users using this app but if the login information is not correct then the admin have to try to login with correct details. After the logging into this system, the admin can managed the users.

4. Use case Diagram



The usecase diagram shows some basic function that the system haves it. There are two actor Users and Admin. The Users can register, they can also login but while logging in, the valid function if included as the user logs in with correct details and incorrect is extended as in some cases the users might put incorrect details which happens rarely. The users can add the information to their list and they can also delete the information in their list if it is no longer useful. The users can make the information as their favourite if they want is find to faster from the list and lastly, the user can view the information in their lists. The admin actor also registers and login to the system and they can manage the registered users.