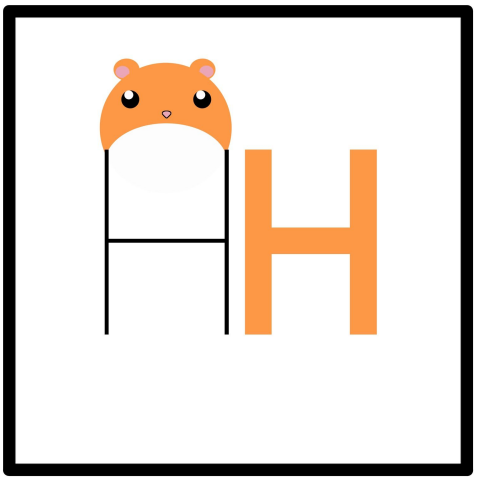
System Design Document
CSCE 361 - Spring 2018

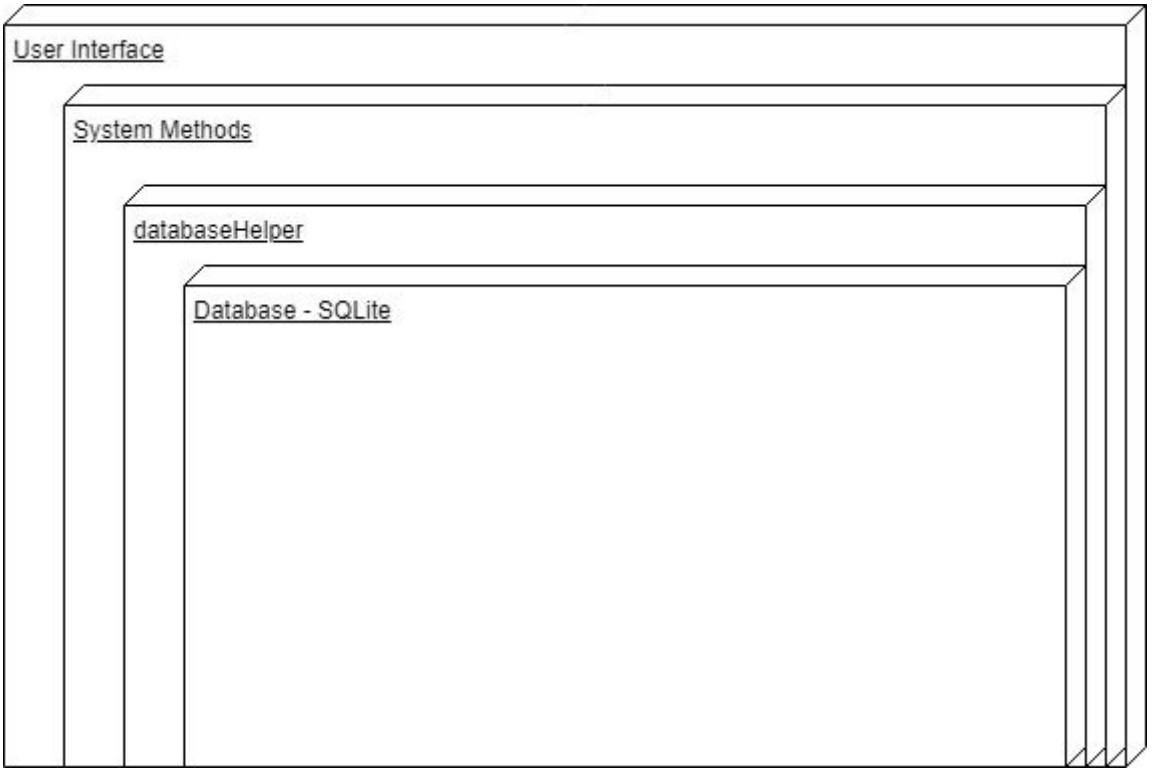


1. Introduction

The purpose of this design document is to describe the system at a architecture level including conditions, data management, and to show the relationship between different modules in the system. This document will also include the relationships between tables in the database as well. The audience of this document will be software engineers, database managers, and system architects who will be implementing this project.

2. Architecture

2.1. Introduction



The high level architecture that we have decided will be a layered model. There are 4 layers in total starting from User Interface as the outermost layer, followed by System Methods layer, databaseHelper layer and finally the Database - SQLite layer. The Database layer will be an SQLite database to store the information of users. The databaseHelper layer provides the connection of the database for the System Methods layer. The System Methods layer comprised of several different methods that interacts with the GUI, the databaseHelper layer and the Database - SQLite layer. The outermost layer, the User Interface layer will be used for interaction between the user and the system.

2.2. Modules

2.2.1. Database Layer

This layer is to hold all the data for the application, to keep track of the flow of data and defining the relationship for each data for different methods. The database that we will be using is SQLite which give us the ease of setting up the connection of the project and the database because Java has libraries for it. The relationship of data will be described in Section 2.2 below.

2.2.2. DatabaseHelper Layer

This layer act as the backbone of the application. This layer is responsible for setting up and closing the connection between the database and the methods from System Methods Layer (2.2.3). This layer will consists of general methods and libraries which help us to maintain code easily and help to reduce the time and effort needed to make the application.

2.2.3. System Methods Layer

This layer contains the methods needed for the application to work. This module will be interacting with the Framework System layer (2.2.2) to access the database (2.2.1). It will also interact with the User Interface (GUI) Layer (2.2.4) to take in different parameters for different methods to search up the data in the database (2.2.1) and display it in the User Interface (GUI) Layer (2.2.4).

2.2.4. User Interface (GUI) Layer

This is the first layer that generates the user interface, and show the interaction between the user and the application as well as shows the image of the application. In this layer, the user will be able to login to the system for the accessibility of the application. This layer will allow users to search up for the helper using the search functions from System Methods Layer (2.2.3) and display the results in the same layer. This layer will be made using Android Studio's prebuilt GUI builder. This layer will also allow new users to register new accounts which the parameters will be directed to a method in System Methods Layer to Framework System Layer and be placed into the Database Layer.

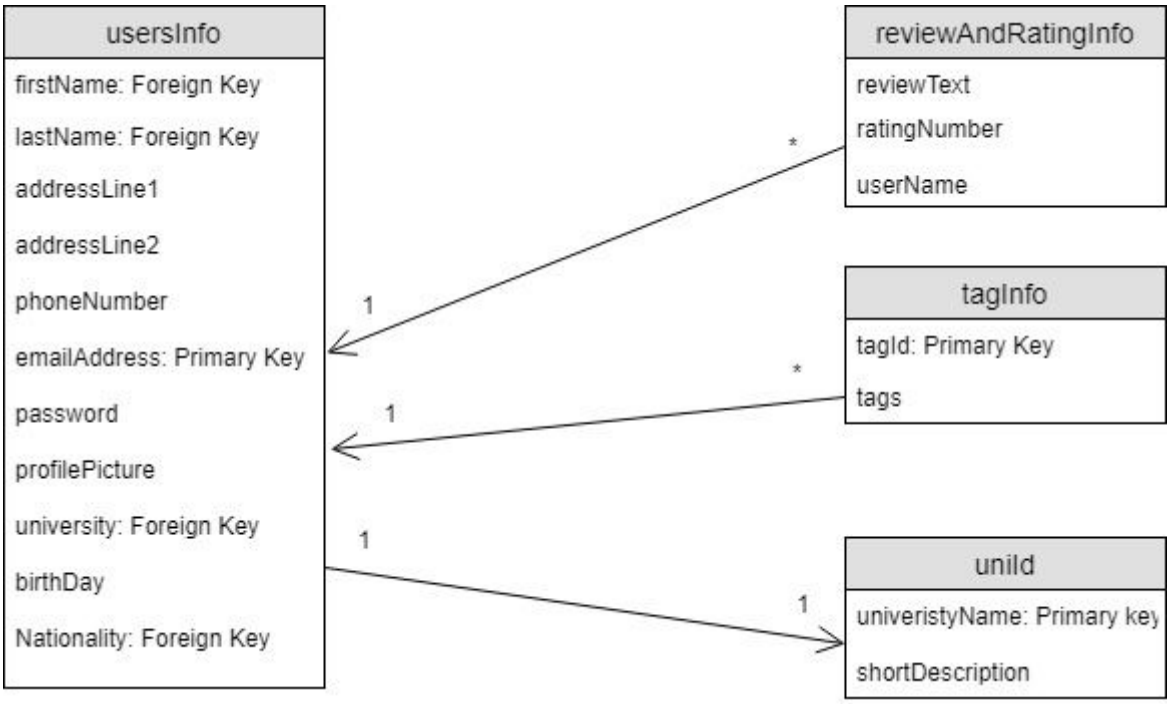
We decided to choose layered system as our choice of architecture because this architecture supports incremental development of sub-systems in different layers. For future development, when we want to improve one of the layer, only the outer layer will be affected. This means that the inner layers will not be affected as all. This architecture also shows the application's structure easily as well for future references.

3. Class Diagrams

3.1. Database Table Classes

The system will be using a MySQL database to store all the necessary data. The tables will be able to describe the information of the users and the classes as well as be able to show the relationship between them. The schema (3.1.1) will describe in detail about the relationships between each tables and what fields that each tables hold.

3.1.1 Schema



3.1.2 Schema Information

The data in the database will follow the organization of the following tables and columns.

usersInfo: -

This table holds the data of different users which parameters such as name, address, university, cell phone number, birthday, email address, password, personID and an image. When the user register for an account, the database will insert a new row into the table where the information of the user will be inserted. The table will be updated as well.

reviewAndRatingInfo:-

This table holds the data of the helper's review and the rating. Users will be able to insert or remove a rating or insert or remove a review from the helper's profile that they have commented on. The database will update the table when an insertion or a removal of data has been made.

uniId:-

This table will hold the information about the university in which the helper is and the year in which the helper is. This table also holds a short description of the university.

tagInfo:

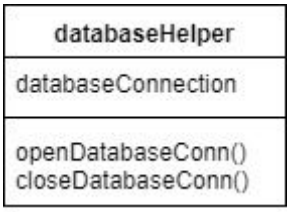
This table holds the data of tags used in association to users and helpers. The database will update the table once there is a new tag inserted or an old tag deleted.

3.2 Class Information

The classes listed below will describe the system module layer and the framework system of the system. The classes are implemented in Java.

3.2.1 databaseHelper

This class acts as an intermediary class for the system modules layer and the database layer. This means that this class is responsible for setting up the connection, closing the connection, and interacting with the database together with the system module layer.

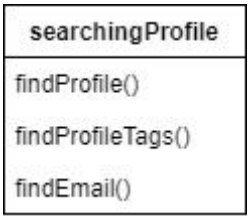


3.2.2 reviewAndRating

This class associates reviews and ratings left by users on the helpers' profile, allowing other users to read the reviews and see the ratings. The users can also sort the searches by rating. The reviewAndRating class will communicate with the database through the databaseHelper class to obtain or edit the review and rating information in the database.

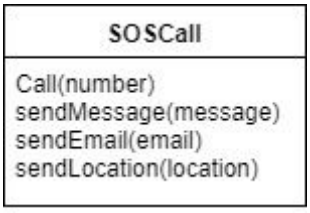
3.2.3 searchingProfile

The Searching Profile class is one where the user will be able to search for the desired university with advanced searches like searching for people who are in the university of their own community as well. The database can be accessed using the databaseHelper class to allow the searching of profiles by tags or email.



3.2.4 SOSCall

The SOS Call is a class where the users can gain emergency help from the helper. The application will have an SOS functionality where the user can use during emergencies by clicking on the SOS icon which will redirect the call to the associated helper. The SOSCall class connects with the databaseHelper class to access the database to retrieve the numbers stored in the database and allows messages, email, and location to be sent.



3.2.5 tagSearch

The Tag Searching class allows users to search up other users based on the tags on the profiles. The tagSearch class retrieves, edits, or removes tags from a profile in a database through the databaseHelper class.

3.3 GUI Layer

The GUI Layer will consists of only three pages, a login screen, the main homepage and the register page. When the user launches the application, the login screen will be shown to the user, prompting them to enter an email address and a password to gain access to the application. If the user doesn't have an account, the user can click on the register button that redirects to the register page. The register page prompts the user to enter several personal details about the user. Once the user is logged in, the user will be redirected to a homepage where there will be a profile tab, home tab, messages, search bar, and a result box.

The homepage is complex compared to the register page and the login page. This is because the homepage will be the place where most of the processes are done. In the homepage, there will be a list of tabs, a search bar and a result box. Searching for a helper in the search bar will require the homepage to generate a list of results. Clicking on the profile tab will show us the profile of the user. Every click action on a tab will lead to a different system module output.