CS 487 Course Project

Hands Plus - Lending a Helping Hand

Prepared by: Arjun Rajagopalan ,FNU
Liu,Xin
Singh,Himanshu
Yum, Sang-Il

EXECUTIVE SUMMARY

Objective

The purpose of this document is to present a detailed description of the Hands+ student assist software. It will explain the purpose and features of the software, the interfaces of the software, what the software will do, the constraints under which it must operate and how the system will react to external stimuli. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system for the development team.

Project Scope

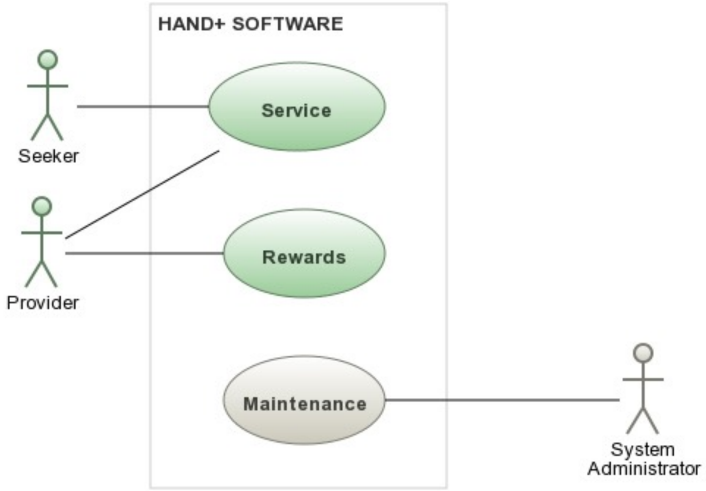
H+ is a student assistant application that lets you help a fellow student in times of need. For example, I need help, moving a table from one place to another, instead of me waiting for a friend, I could just post a request in the Need Help section, anyone who is free and willing to help me could just select me and help me out. His reward could vary from a variety of things such as candy bars, a cup of coffee to even money, depending upon the person requesting the help. The Need Help request could vary from a simple request to move things from one place to another to tutoring someone in learning an instrument. This application would be used only by the students of IIT and would require an ***iit.edu*** email ID to login.

The Development for this project would be done using the ionic framework, which is a cross-platform mobile development framework. Ionic lets, us develop applications for both iOS and Android. The Database that we would be using is mySQL and the server side of the program would be in PHP.

PROJECT DESCRIPTION

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It will also describe what type of stakeholders that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.

System Environment



The Hand+ System is mobile application software. The mobile application will be used to post the Seekers service and and view information if anybody is willing to offer it, Similarly the Provider can use the application to respond to he service.The Hands+ System has 3 Actors and 1 System. The Seeker will ask for the service, the provider will be the provider of the service and in the lieu of the service he will get Rewards(It will depends on the mutual agreement of Seeker and Provider), and System Administrator will be responsible for the Maintenance of the System.

User characteristics

There are two types of users that interact with the system: users of the mobile application, and administrators. Each of these two types of users has different use of the system so each of them has their own requirements.

Constraints

The Internet connection is also a constraint for the application. Since the application fetches data from the database over the Internet, it is crucial that there is an Internet connection for the application to function.

The mobile application will be constrained by the capacity of the database. Since the database is shared between both application it may be forced to queue incoming requests and therefor increase the time it takes to fetch data.

Assumptions and dependencies

One assumption about the product is that it will always be used on mobile phones that have enough performance. If the phone does not have enough hardware resources available for the application, for example the users might have allocated them with other applications, there may be scenarios where the application does not work as intended or even at all.

Another assumption is that the GPS components in all phones work in the same way. If the phones have different interfaces to the GPS, the application need to be specifically adjusted to each interface and that 6 would mean the integration with the GPS would have different requirements than what is stated in this specification.

SPECIFIC REQUIREMENTS

This section contains all of the functional and quality requirements of the system.

Functional Requirements

Functional requirement 1

Title: Download mobile application

DescrIption: A user should be able to download the mobile application through either an application store or similar service on the mobile phone. The application should be free to download.

Functional requirement 2

Title: Download and notify users of new releases

DescrIption: When a new/updated version or release of the software is released, the user should check for these manually. The download of the new release should be done through the mobile phone in the same way as downloading the mobile application

Functional requirement 3

Title: User registration-Valid Hawk ID

DescrIption: Given that a user has downloaded the mobile application, then the user should be able to register through the mobile application. The user must provide user-name, password and e-mail address. The user can choose to provide a regularly used phone number for contact info.

Functional requirement 4

Title:: User log-in - Mobile application

DescrIption: Given that a user has registered, then the user should be able to log in to the mobile application. The log-in information will be stored on the phone and in the future the user should be logged in automatically.

Functional requirement 5

Title: Mobile application – Post

DescrIption: Given that a user is logged in to the mobile application, then the page that is shown should be the Post Service. The user should be able to Post the service. Post/View options include Post Service, Post service in Specific category.

Functional requirement 6

Title: Mobile application - View

DescrIption: The user should be able to View the list of profiles willing to serve the service. View option will include View the profile, contact info and description section.

Non-Functional Requirements

The Application will be on a mobile with high speed Internet capability. The software developed here assumes the user have valid HAWK id.

Performance Requirements

Performance of the application depends hugely on resource available on the user's device.

Performance should not be an issue because all of our server queries involve small pieces of data. Changing screens will require very little computation and thus will occur very quickly. Server updates should only take a few seconds as long as the phone can maintain a steady signal.

Safety Requirements

Hand+ will not affect data stored outside of its servers nor will it affect any other applications installed on the user's phone. It cannot cause any damage to the phone or its internal components. The only potential safety concern associated with this application applies to virtually all handset apps: Hands+ should not be used while operating a vehicle or in any other situation where the user's attention must be focused elsewhere.

Security Requirements

This application assumes that only the user will have access to his/her Android handset. With that being said, only a Hawk email address is required to verify the identity of the user upon opening the app. Since it is authenticate password from IIT's web server, there is no potential threat for illegitimate users.

SYSTEM MODEL

Context Model:

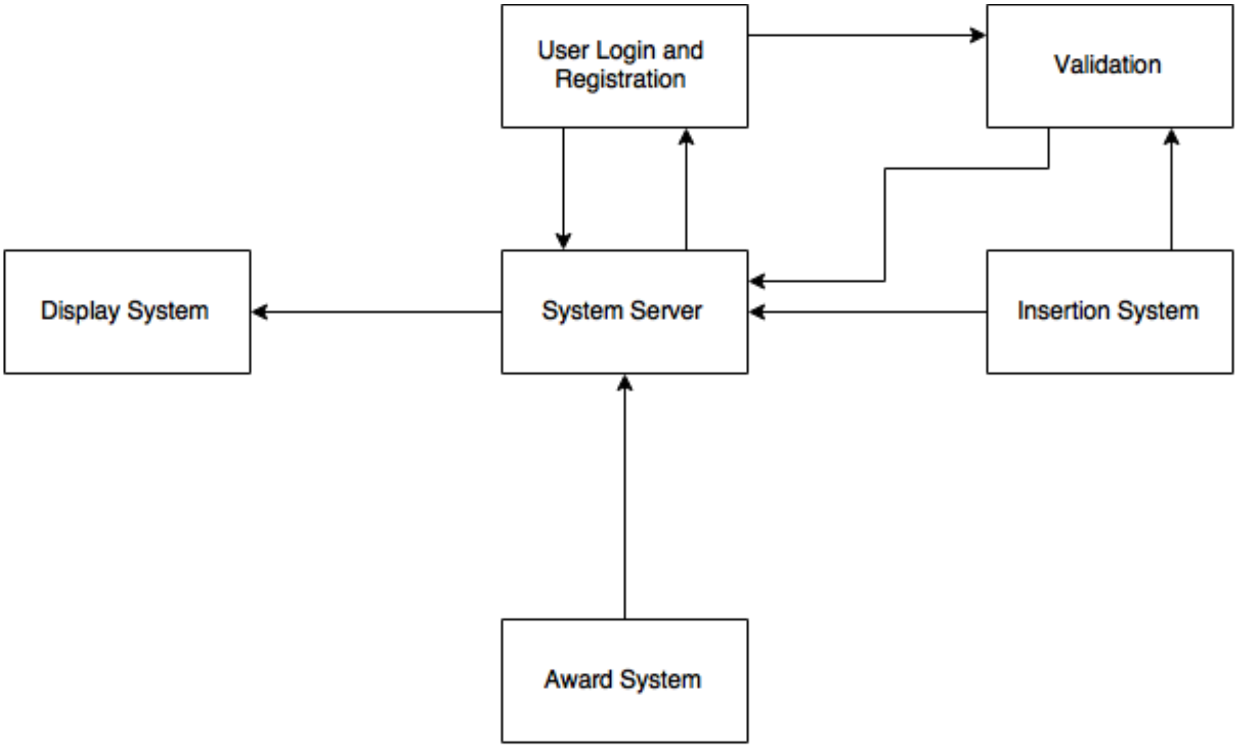


Figure: Context Model

The context model gives a detailed overview of all the systems and how they interact with each other.

The various systems involved include the User login and Registration, Insertion Display, Validation and Award System.

the User login and Registration system will let the user use the application. the Insertion and Display system, will insert and retrieve data from the database to display the various help requests. The Validation system will validate the data the user would input ,so that the system Server does not get affected. The Award System, keeps a track of the award points the user gets or uses while using the application.

State Model for Hands +:

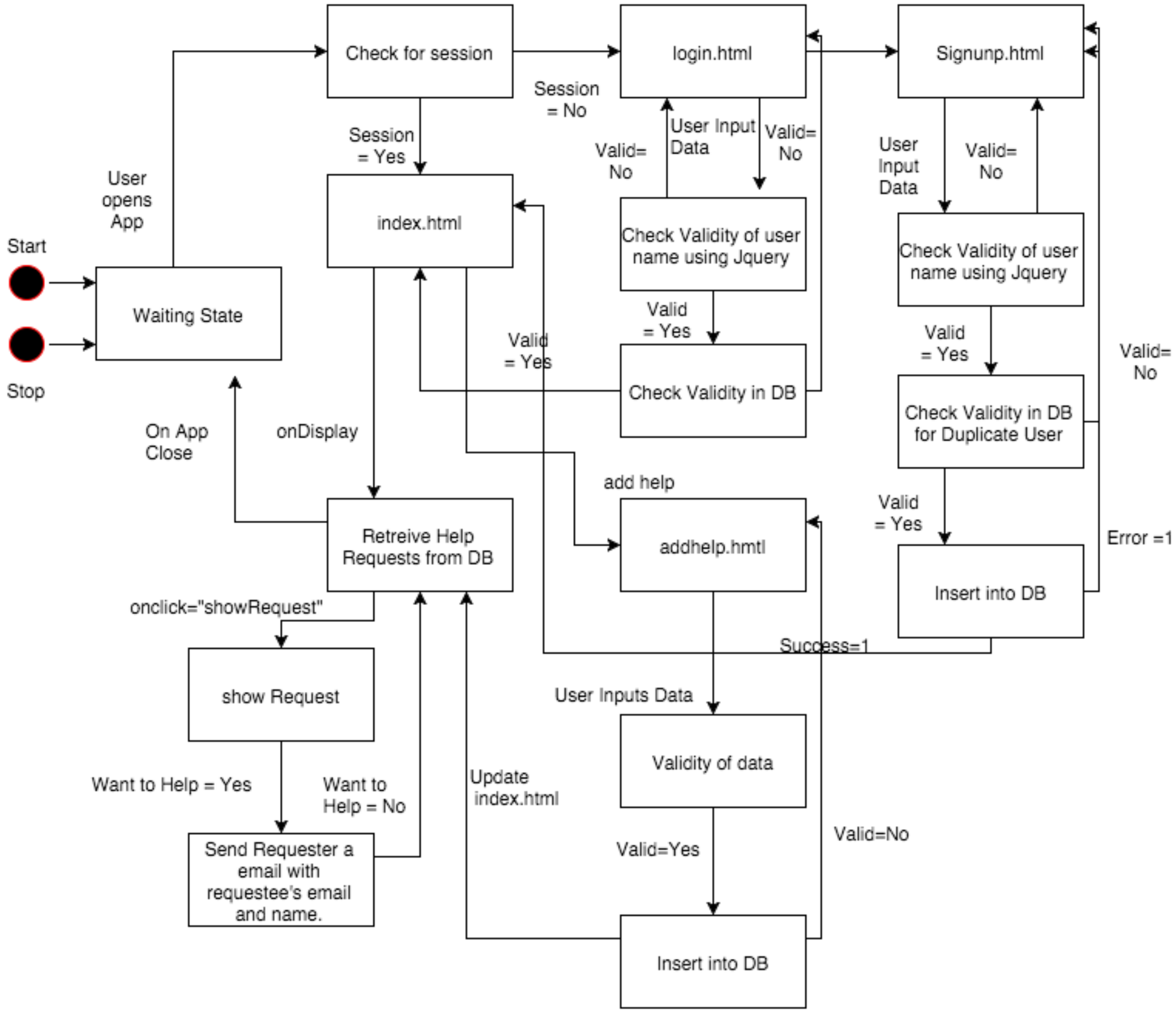


Figure:State Machine Diagram of Detailed operation.

Once the user downloads and installs the application for his device, if he there is no history of his previous login into the system, he will be asked to login. If the user is a first time user, he can click on the Sign Up page, or he could enter his credentials and login. If a previous history of login in detected, he will be directed directly to the index page, where the user can access the various features of the software.

For the sign up process, the user will be asked to enter the name, email address, password and re-enter password. Once the user enters these values, they are checked for validation. The Re-enter password textfield will provide us with more validation with respect to the password field.All these values will be entered into the systems Database for the user to be registered successfully. Once he is successfully registered he can access his account hence forth.

For the Sign in process, the user will have to enter his hawk email ID and password that eh set up during the signup process. the system will now validate to check if there is no invalid characters which could possible break the system. If the user's credentials are found in the system, he will be logged in and a serialized object of his credentials will be saved in the application storage. In case the user forgets his password, he will be asked to enter his email address, the system will hence send him a link a temporary password, which he can reset it to the password he desires.

Once the user login process is successful, the index page is displayed. The index page retrieves data from the database, using Angular JS. We have also added an additional refresh tab in the menu, so that our users can stay up to track with the requests.

When the users selects a help request, the help tab expands and lets the user see the help request in detail. In this details tab, We have given an options as to whether, the user would like to accept or deny the offer. If he accepts it an email is sent to the user who has put up the request. Once the help request is complete, the user who set up the request can add an award point to that person.

The award point system is such that, in order to request an help, the user will require an award point. The only way to win a award point is by helping other people.

The add help feature, is accessed by the add button on homepage. Once pressed, the add help page opens and the user will have to add the details and post it on the system.

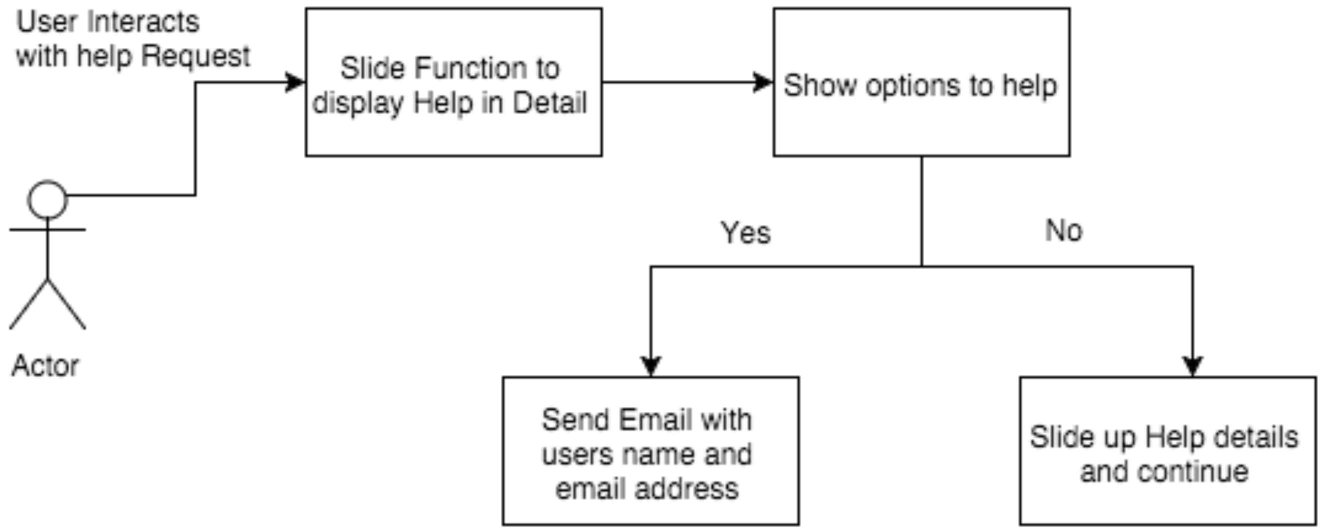

User Interacts with help Request
Actor
Slide Function to display Help in Detail
Show options to help
Yes
No
Send Email with users name and email address
Slide up Help details and continue

Figure: Help Function

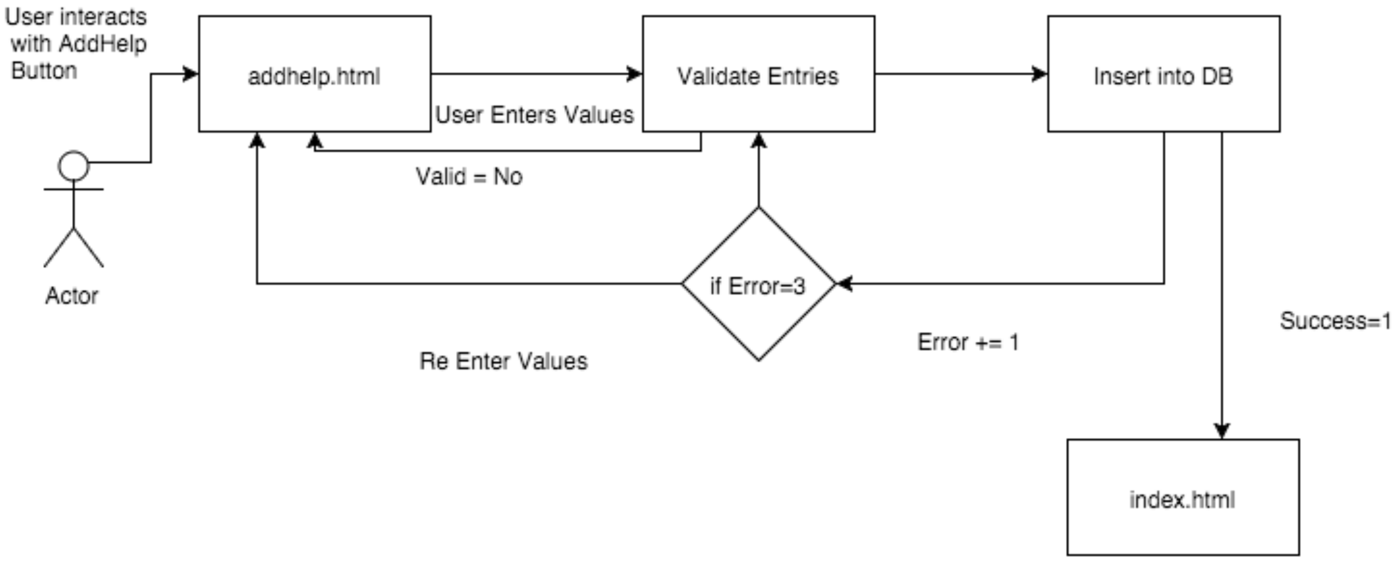


Figure: Adding Help to the System

USER INTERFACE:

The User Interface (UI) of our application is displayed below.

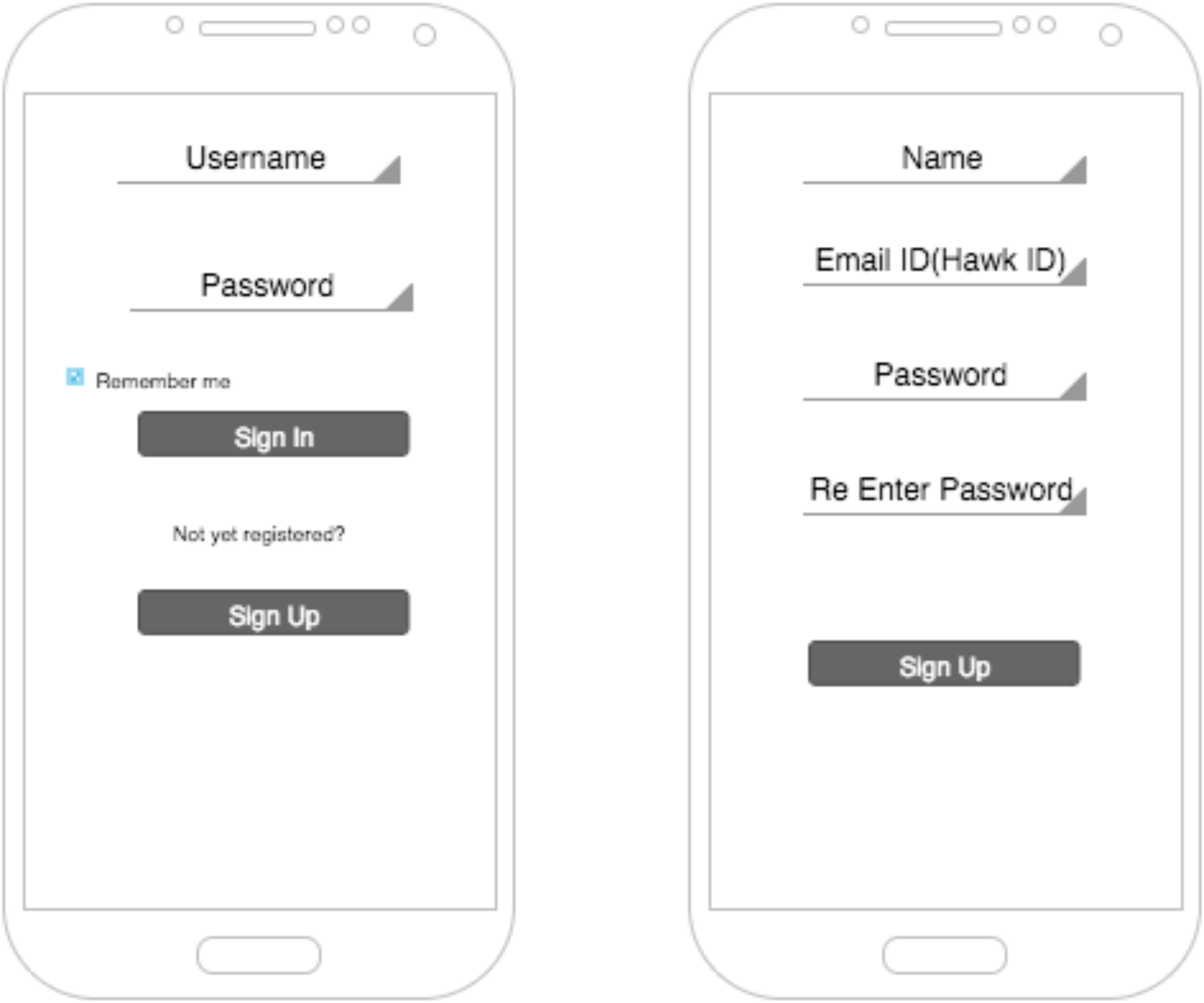


Figure: UI of Sign In and Sign up

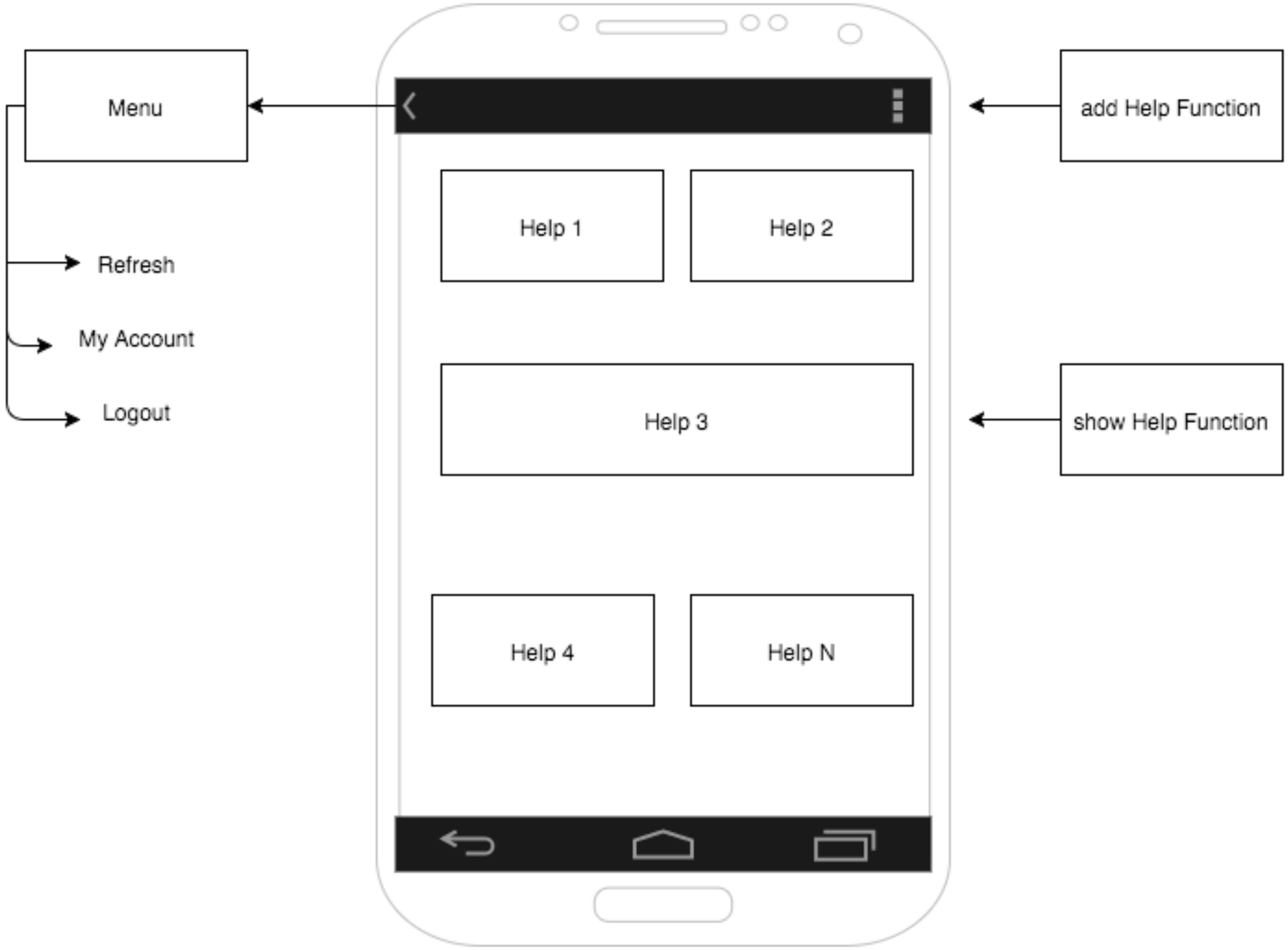


Figure: Index Page

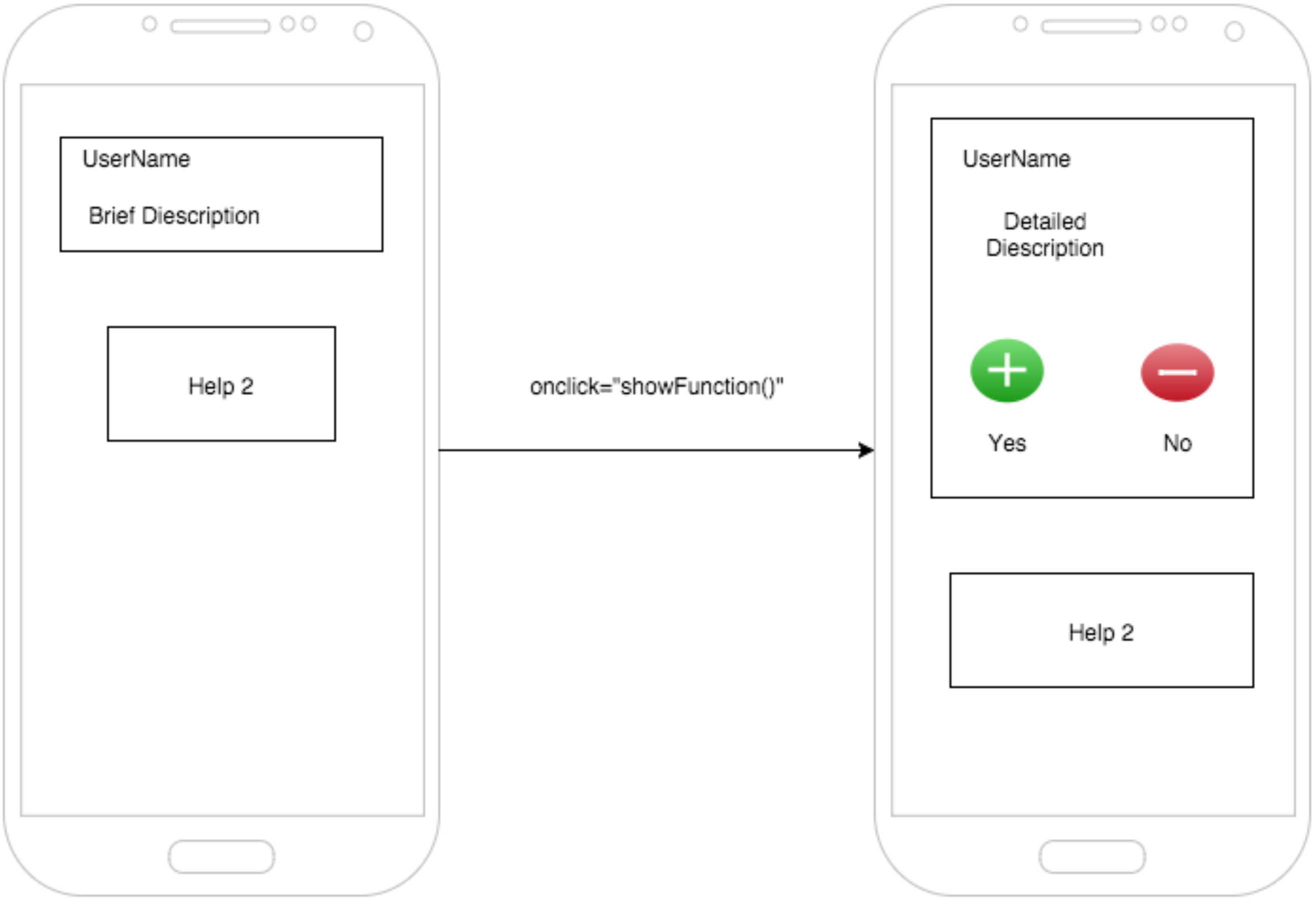


Figure: Slide Function of Help

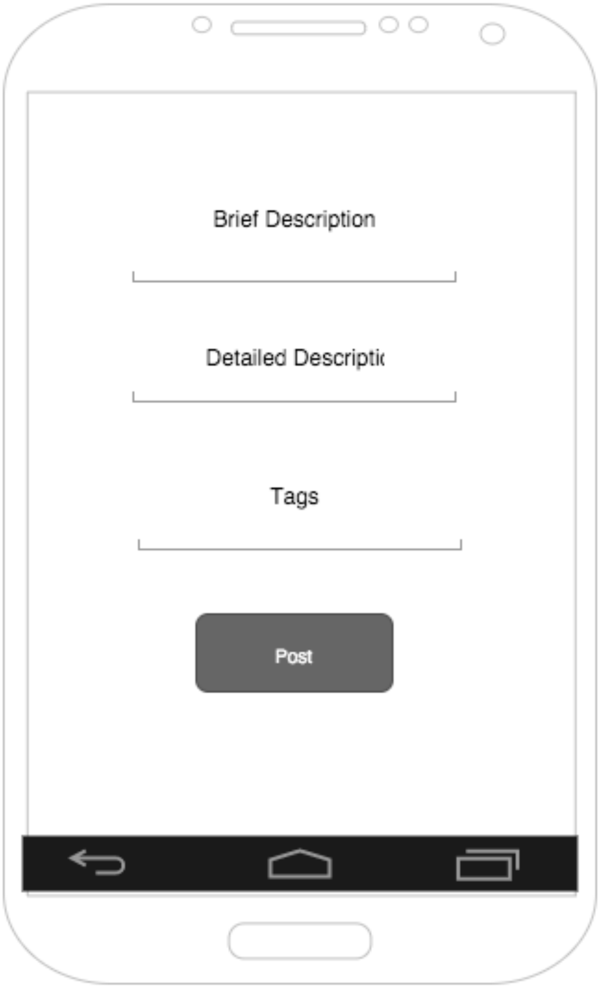


Figure: Add Help Page

DATA MODEL

AGGREGATION

Help class

- Each Help instance can have multi-Comment instance. It is added when user make comment on the help.
- Each Help instance can have multi-Multimedia Content. It is added multimedia data such as photos and video when use upload help request or needs.
- Each Help instance have exactly two date instance. One is UploadDate, another is AvailableTime.

User

- Each User have exactly one Multimedia Content instance to save profile photos

Comment

- Each Comment have exactly one Date instance to save submit time.

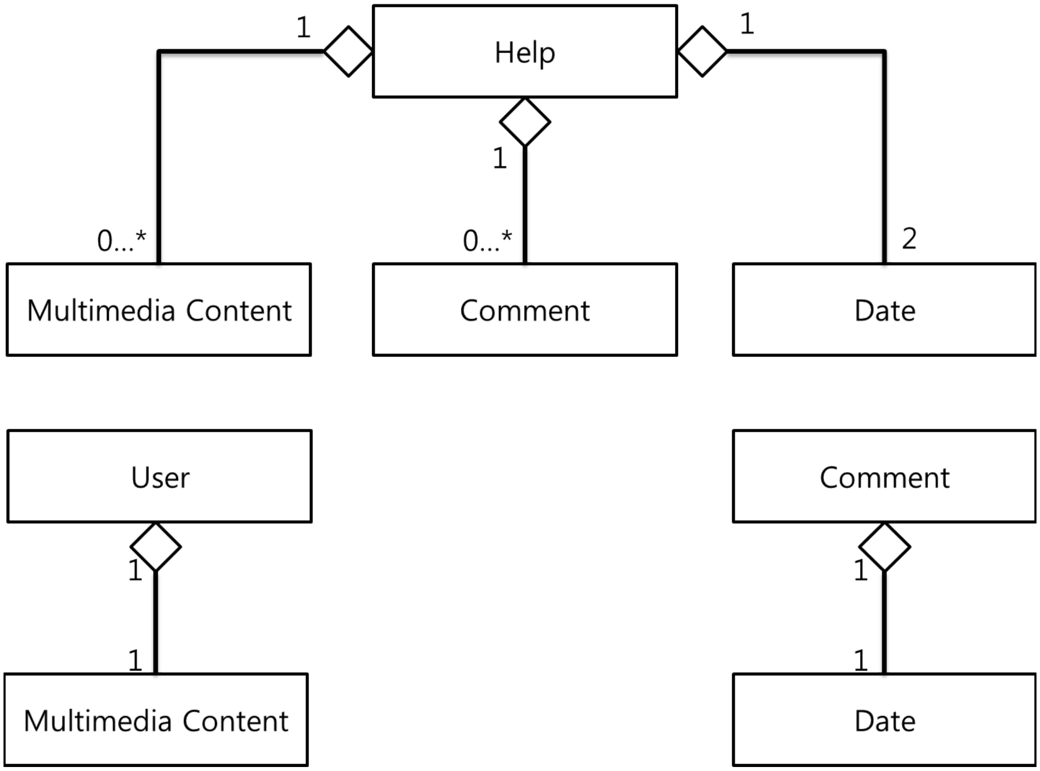


Figure: Aggression Model

CLASS AND ASSOCIATION

User instance

- Refer multi-Help instance to point help history
- Refer multi-Comment instance to point comment history

Help instance

- Refer exactly one User instance. The User instance point UploadUser

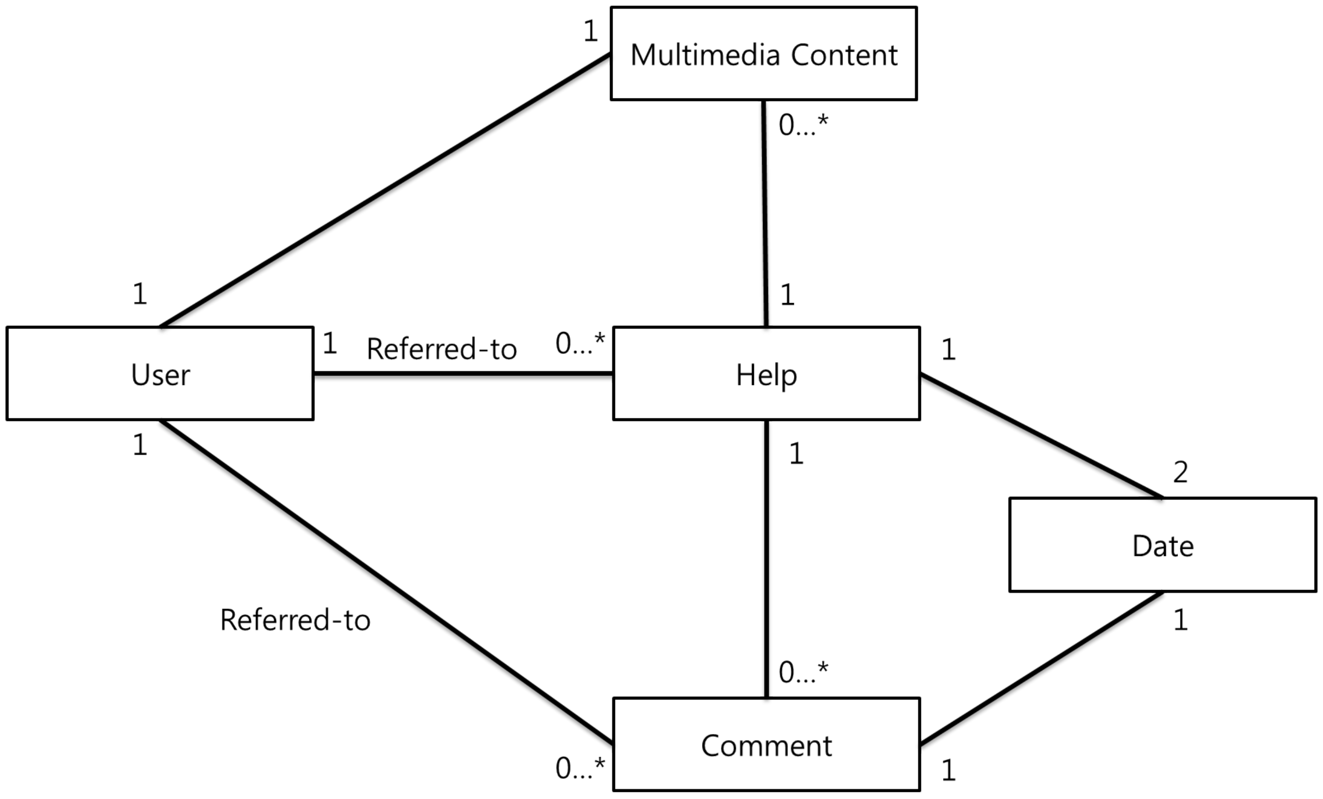
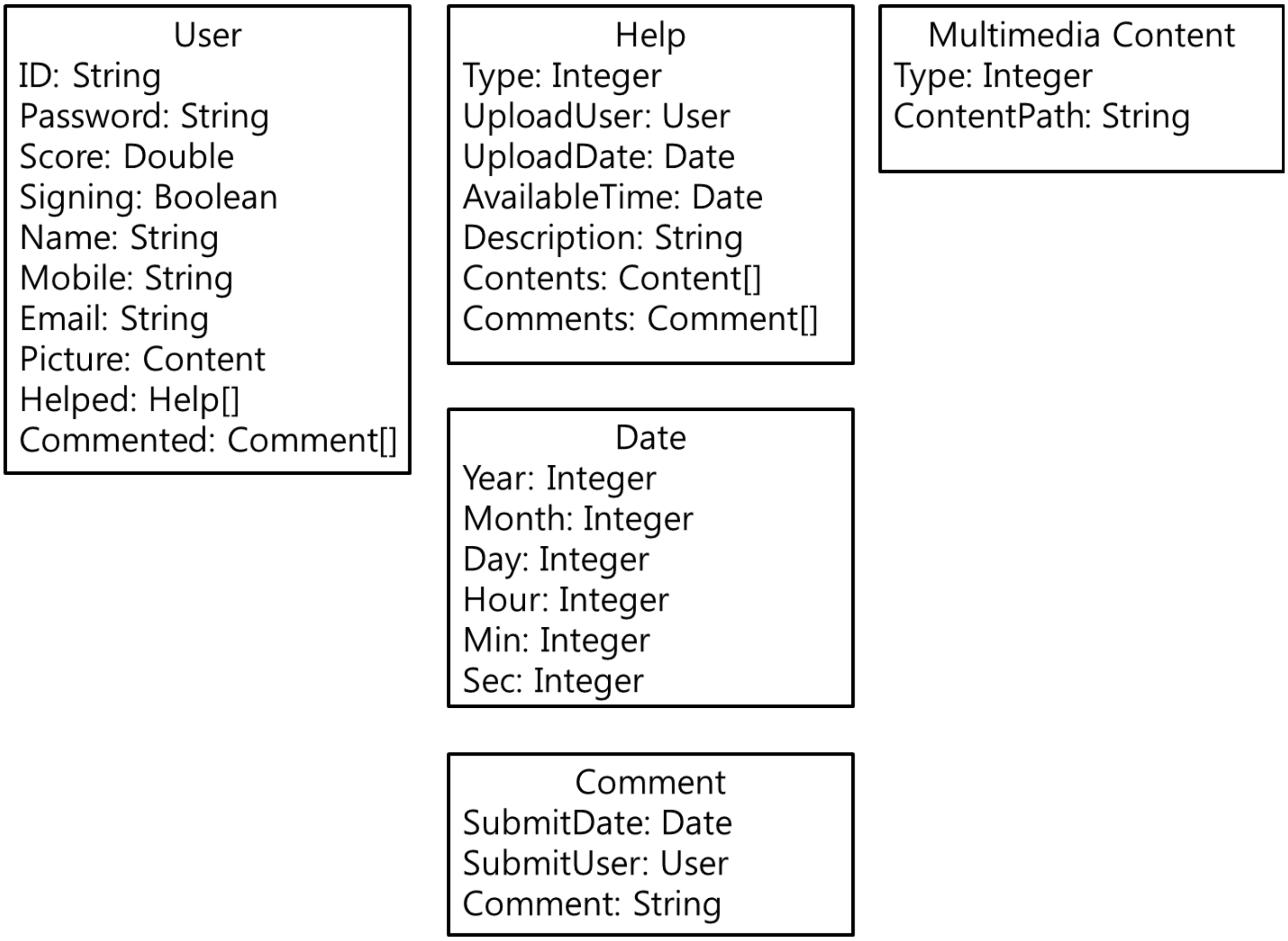


Figure: Class and Association

DATA

- Date class is separated because more than one class refers it, and if date saves as string, sorting by date is not convenience.

- Multimedia Content is separated because more than one class refers it, and type of multimedia content can be multiple such as photos, video, etc. Therefore, to cover all of type, Multimedia class indicates type and the content path in the file system or databases.
- User class includes ID/Password to login, Score to show user reliability, signing to indicate user current login status. Name/Mobile/Email/Picture to save user personal information and Helped/Commented to save history of user activities.
- Help class includes type to indicate help request/needs, upload user to save who upload the help, UploadDate/AvailableTime/Description/Contents to save help detail information and comments to save comments on the help.



User
ID: String
Password: String
Score: Double
Signing: Boolean
Name: String
Mobile: String
Email: String
Picture: Content
Helped: Help[]
Commented: Comment[]

Help
Type: Integer
UploadUser: User
UploadDate: Date
AvailableTime: Date
Description: String
Contents: Content[]
Comments: Comment[]

Multimedia Content
Type: Integer
ContentPath: String

Date
Year: Integer
Month: Integer
Day: Integer
Hour: Integer
Min: Integer
Sec: Integer

Comment
SubmitDate: Date
SubmitUser: User
Comment: String

Figure: Data Model

SOFTWARE TEST PLAN

Introduction：

Purpose

The purpose of the Software Test Plan (STP) is to test the functionality and make sure that the requirements stated in the Software Requirement Specification are fulfilled.

Definitions and Acronyms

SRS	Software Requirements Specification
STP	Software Test Plan.

Scope

The Hands+ is an application which allow its user to provide or get helps.
This application is developed just for students. Personal information is highly protected.
Besides safety, efficiency is also important.

Testing objective

1.Define the activities required to prepare for and conduct System, Beta and User Acceptance testing.
2. Communicate to all responsible parties the System Test strategy. § Define deliverables and responsible parties.
3. Communicate to all responsible parties the various Dependencies and Risks

Test items

1. Functions to be tested
   1. identify the identity of the user when he is registering.
   2. After a new user registered, add his information to the database
   3. Show the information of a registered user

4. One user want to offer a help, store this in the database and show this information to users who need this kind of help
5. One user can see the information that may help himself out
6. A user want to help another user, he can send a message to the him.
7. A user can review the users who have helped him and can give some commons.

Non-Functional Items to be Tested:

1. User's information security
2. Software safety.
3. Software efficiency

Test strategy

Software testing shall be performed throughout the development phases starting with unit testing, integration, incremental testing, and finishing with system qualification testing. The software shall be tested on the system requirements.

Program Unit Testing is performed by the developers on the Programming Team who implemented that particular program unit to verify that it performs according to its intended design.

A suite of automated tests will be developed to test the basic functionality of the system and perform regression testing on areas of the systems that previously had critical defects. The tool will also assist us b y executing user scenarios thereby emulating several users.

A list of the various planned tests and a brief explanation follows below.

Function Test

The function test focus on the behavior of thesystem. Its purpose is to verify that the outputs meets the functional requirements defined in the requirements document.

For a same group of input data, compare the expected output to the actual output. According to the SRS, develop some test cases, do not need to consider the internel components of the software.

Performance Test

Performance test will be conducted to ensure that the payroll system's time-sensitive requirements meet the user expectations.

This test will conduct in two situation: input normal amount of data, and input overload data. We can use the test cases in the function testing, just need some little changes.

Security Test

Security tests verify whether the system is security considering its data and access control etc.
In this system, only registered users can access to the information, only students can be a user of this application. According to the SRS, create different type of users, one of them are not student, verify the data is correctly available of denied. In other cases, we should test whether this system can survive from external attacks.

Stress and Volume Test

There maybe some errors aroused when the server, client or database is overload. To discovery and test this part of data, we can use the stress and volume test.
The System will be stress tested using twice the number of expected users . Meanwhile each user may have different action or they request the same thing in the same time.

Recovery Test

Recovery Testing ensures that the target-of-test can successfully recover from a variety of hardware or network malfunctions with undue loss of data or data integrity.
Recovery tests will force the system to fail in a various ways and verify the recovery is properly performed. Such as power interruption of the server or client.

Documentation Test

Tests will be conducted to check the accuracy of the user documentation. These tests will ensure that no features are missing, and the contents can be easily understood.

User Acceptance Test

The purpose of these tests is to confirm that the system is developed according to the specified user requirements and is ready for operational use

Testing process

Test plan schedule

Test process	Description
Test requeriment review	Based on the SRS, developing what should be tested.
Test cases design	Design some cases to test the software
Test cases implementation	Implementing the test cases
Test result analysis	Analysis the test result. output project problem report
Retesting when the developers fix the problems	Output the final test summary report

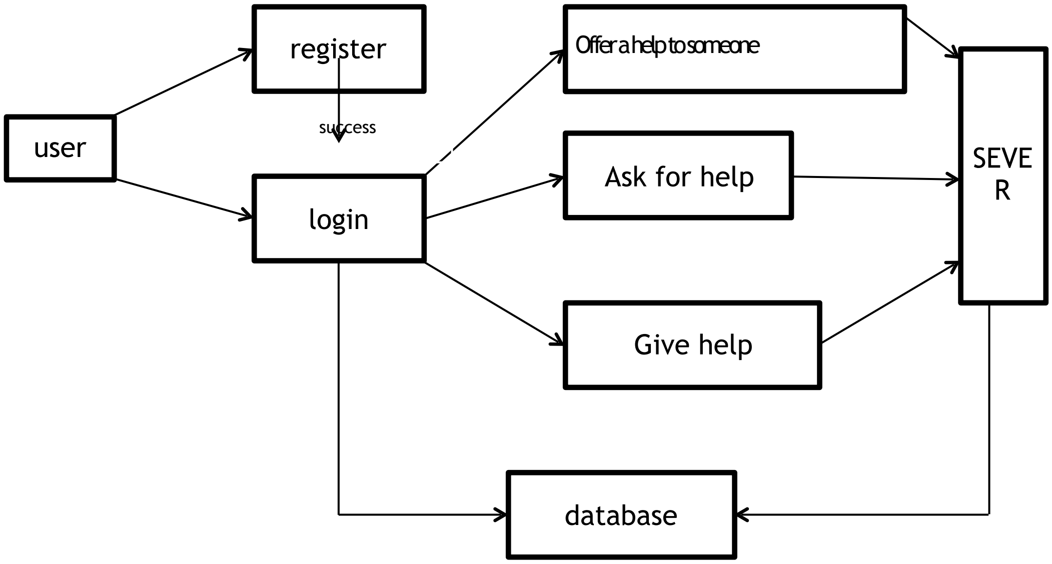
Implementation/suspension/completion criteria

Implementation criteria	hardware environment is appropriate , software is ready for operational use.
Suspension criteria	Source code contains one or more critical defects, which seriously prevents o r limits testing progress. exception and unexpected result appears in the testing progress
Completion criteria	fulfill the requirements in the requirement documents, the actual output equal to the expected output.

Test Case Design

Scenarios testing cases:

BASIC ACTION OF A USER:



Scenarios design:

Scenario one	register successfully
Scenario two	a user who want to register is not a student
Scenario three	login successfully
Scenario four	password mistake,still have chance
Scenario five	password mistake, do not have chance
Scenario six	give help successfully
Scenario seven	when providing help, give a wrong description
Scenario eight	when ask for help, give a wrong description
Scenario nine	Database is filled, but there are new data need to be stored

Scenario test cases design:

(statement: √ : the input satify the requirement ; --- : no matter what data; × : input wrong ; n/a : don't need such input)

Test cases	username	password	Email address	Real name	Expected result
Scenario one	√	√	√	√	Register successfully, store the user's information in database
Scenario two	---	-----	×	--------	Hint: email address is not reight. Request denied.
Scenario three	√	√	n/a	n/a	A registerd user login successfully
Scenario four	√	×	n/a	n/a	Warning message. Go bak to login step to input the passwore again.
Scenario five	√	×	n/a	n/a	Send Warning message and hints to user's email address.

(statement:in the 'give help section' and 'ask for help section', user should provide titile, keywords, description. If the information provided is wrong, then the system won't process it)

Test cases	Title	Keywords	description	Expected result
Scenario six	√	√	√	This information will stored in the database. And someone who need help will see this information
Scenario seven	√	√	×	Waring. Ask user to try again.
Scenario eight	√	√	×	Warning. Ask user to try again.

Test cases	design	Expected result
Scenario nine	Fill out the database. use Scenario one to test	Warning the user, do not store the data. Send message to administrator.