Student Companion

Third Iteration Report

COMP-SCI 5551 Advanced Software Engineering

Group 7

Sri Harsha Chennavajjala (7)

Teja Garidepally (12)

Raj Kiran Reddy Munnangi (32)

Suhas Sai Raparthi (51)

Introduction

This document is intended to provide an overall description of the project named "Student Companion" in detail. The project schedule and the plan of action is also discussed. The proposal document will give an insight on the project. The outcome of the second increment is the login page validation, database creation, home page creation and side menu creation.

Project Goal and Objectives

The goal of this project is to provide various functionalities that a student uses regularly such as updating the profile, checking for computer lab availability, library study room reservation etc. The student details will already present in the database. The student has to login before he uses these functionalities. Main objectives of this application are:

- To reduce the student's stress and to save the student's time by providing the latest availability of the computer labs.
- To develop an application that helps the students in taking the decision on to which laboratory the students have to go.
- To secure the information by providing a login form to the end user.
- To provide a tool with which the students will be able to reserve the library study rooms.
- To ensure that the student will never miss his schedules by setting reminders.
- To enable the Student Assistants to view their shifts, post and take substitutions.
- To provide the students with the option to update their address or mobile phone number etc.

Project Background and Related Work

Some functionalities of this application are already exist. We are creating a new android application which integrates (mash up) all the available and new functionalities under one hood, thus making the application a viable one. Some functionalities will be developed by importing the existing APIs into our application like Google Calendar API, Google Maps API etc. We are inspired by the problems that the Students are currently facing in reserving the study rooms, problems related to their working shifts and we came up with a solution which can resolve the existing challenges.

Significance:

The major significance of the application lies in mashing up of all the useful services under one system. This will save the student's time and increases the productivity. The application will prevent the fraudulent usage by restricting the resources access to only the students who
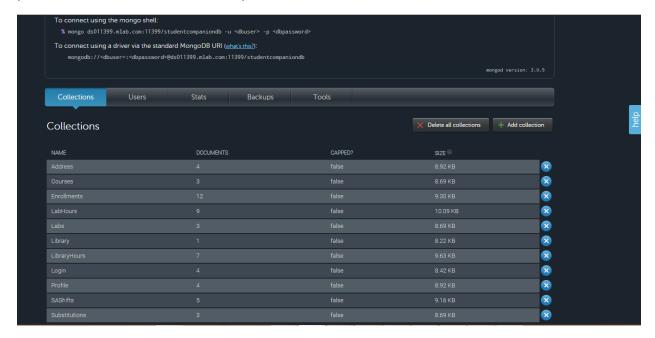
successfully logged in to the system. As of now, the student assistants has to go through a lengthy process in order to post or take shifts. The proposed application will make it easier for the student to perform such tasks by providing on-the-go support.

Work flow

The first workflow of the application is that of the designing the User Interface for the system. Our application goal is to provide instant access to the user regarding lab availability and library room availability. So, for this we are developing an mobile application. We are using ionic framework so that we can develop hybrid application.

Second, we have created the login page and database using MongoLab and we have successfully authenticated the user based on his details which are stored in database in mongodb.

The third workflow deals with the creating a mongodb database to store and retrieve the user profile and information about IS labs, library study rooms etc. The work flow of the each and every feature of the application is explained through sequence diagrams and state chart diagram presented in the first increment report.

Technological and architecture requirements

In the application system need to interact with the database for retrieving the data to do this interaction between the system and database is done through REST technology. We use CRUD API calls of the REST for accessing the database.

Ionic framework, it is a powerful SDK used to develop hybrid applications using web technologies like HTML, CSS and JavaScript. It is also a core in providing better UI to the user.

The architectural requirement aims at the development of flexible architecture for the better interaction between the system and the database.

Existing Services/API

We have so far used the MongoDB in our application. MongoDB is a cross-platform document-oriented database and it is a NoSQL database, MongoDB avoids the traditional table-based relational database structure in favor of JSON-like documents with dynamic schemas making the integration of data in certain types of applications easier and faster.

We have used mongoDB to store the information of the users and data about the IS labs, library study rooms, student assistant information etc. Using REST API calls data is retrieved from the database and displayed to the user.

mLab is a fully managed cloud database service that hosts and provide featuring automated provisioning and scaling of MongoDB databases. Data can be accessed from mongoDB using two ways. Fist, by using the DATA APT URL in which it contains database name, collection name and API key.
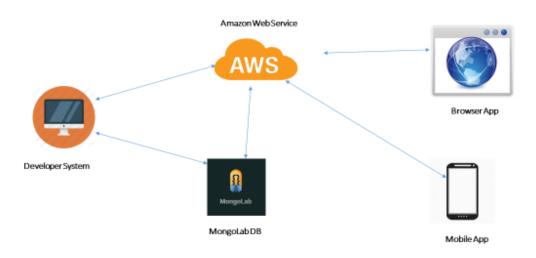
Example URL:

https://api.mongolab.com/api/1/databases/studentcorner/collections//Ase_project/' + id + **'?apiKey=Q_u73BV4oOdMGpnu3WFGmJ8YH_lxHDHO**

Second, accessing the mongoDB using a driver to connect the database. An example driver:

```
mongodb://<dbuser>:<dbpassword>@ds011399.mlab.com:11399/studentcompaniondb
```

REST Services:

We have developed an API through which we perform the REST services for our application. As in the second increment we have developed database in the mongoLab and the data is fetched from the mongoLab through REST services.

We have used Amazon Web Services as a server to host our REST API. Our REST API will run in background and listen to port 9000.Our application send data request to the REST API hosted on Amazon EC2 cloud server. REST API will fetch the request data from mongoDB collections and perform the filtering of data according to the criteria and sends responds to the client application.

This enables us complex queries on the server instead on the client thereby increasing the performance of client application.
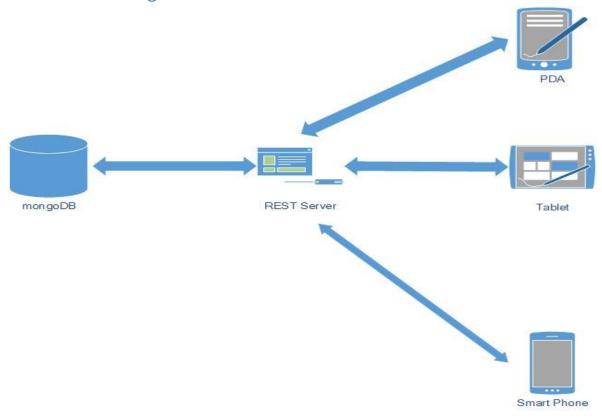
Architecture Diagram

Fig 1: Architecture Diagram for StudentCompanion

The architecture diagram for our application shows that it is 3-tier architecture. Application which deployed in smart phones, PDA's and Tablet uses the REST services for interacting with the database for storing and retrieval of required information from the mongoDB.

Project plan

The proposed project plan is outlined the by the screenshot from the ZenHub tool. The project is divided into four milestones. Each iteration has several states namely future tasks, new issues, to do, development in progress, testing in progress, done and closed.
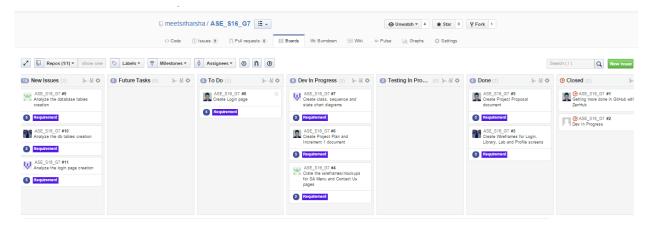
Fig. 2 ZenHub Board showing the project plan and current tasks.

Milestone 3:

This milestone mainly deals with the designing the end to end functionality of the system. The tasks of this phase mainly focuses six tasks.

First, the development of REST API which performs the communication between Client application and database.

Second, development of Profile page and display the profile of the user which is stored in the mongoDB.

Third, development of Lab Information page and display the Lab name, available work stations and systems which are in use in that lab.

Fourth, development of Reserve Study Room page and display the Library information, study rooms which are reserved by the user and available study rooms.

Fifth, development SA Menu page and display the SA Shifts of the user on a or after a particular date.

The results of this milestone contains the screenshots of the application and the related class, sequence and state chart diagrams.

Third Increment Report

This document contains the report of third increment of work done on the Student Companion application. This document emphasizes the end to end implementation of the application using different technologies such as REST API, Amazon Web Services, Ionic framework and mongoDB. We're using the HTML5, CSS ,Node.js, , ExpressJS and Angular JS for the end to end implementation.

We have created REST API which host on the Amazon Web Services through which communication between client application and database.

We have created the Profile, Lab Information, Reserve Study Room, SA Menu, Contact Us pages. We have also checked the work so far by deploying the application in the mobile and manual testing is done.

The detail development in third increment include the creation of REST API, Profile, Lab Information, Reserve Study Room, SA Menu, Contact Us pages. We are going to work enhancement of application, improving use interface, bug fixes and the remaining functionalities of the application.

Class diagram for the high level design of the application:

ComputerLibrary

- ssoId : string
- name : string
- building : string
- room : int
- address1 : string
- address2 : string
- city : string
- zipcode : int
- state : string
- country : string
- capacity : int
- printerCount : int
- colorPrintCount : int
- inUse : count
- hours : count

+ getHours( ) : int
+ getCapacity( ) : int
+ getPrinterCount( ) : int

Student

- ssoId : string
- firstName : string
- lastName : string
- email : string
- password : string
- mobile : int
-address1 : string
- address : string
- city : string
- zipcode : int
- state : string
- country : string
- studentType : string

+ authenticateStudent( ):boolean
+ updateAddress( ):boolean
+ updateMobileNum( ):boolean

LabReserves

- ssoId : string
- labId : string
- startDateTime : Date
- endDateTime : Date

+ getStartDateTime( ): Date
+ endDateTime( ): Date

UIController

-ssoId : string
-screenId : string

+ detDetails() : string
+ displayDetails( ) : string

LibraryReserve

- ssoId : string
- mail : string
- mobile : int
- start : time
- End : time

+ reserve( ) : boolean

Library

- ssoId : string
- roomId : int
- capacity : int
- projectorCount : int
- hoursWorking : int

+ getCapacity ( ) ; int
+ getHoursWorking ( ): int

SA Tasks

- id : string
- startShift : time
- endShift : time
- shiftId : int

+ getStart ( ) : time
+ getEnd ( ) : time

SA Menu

- ssoId : string
- id : string

+ postSubstitute( ) : boolean
+ addSubstitute ( ) : boolean
+ getShift ( ) : boolean

SA Shifts

- shiftId : string
- saId : string
- locationId : string
- start : time
- end : time

+ getShift ( ) : int
+ getLocation : string

Fig. 2 demonstrates a class diagram of high level design of application

Sequence diagrams for the high level design of the application:
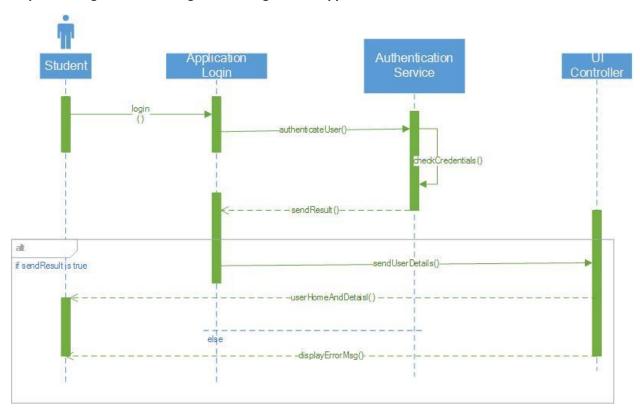
Fig. 3 Sequence diagram for student login activity
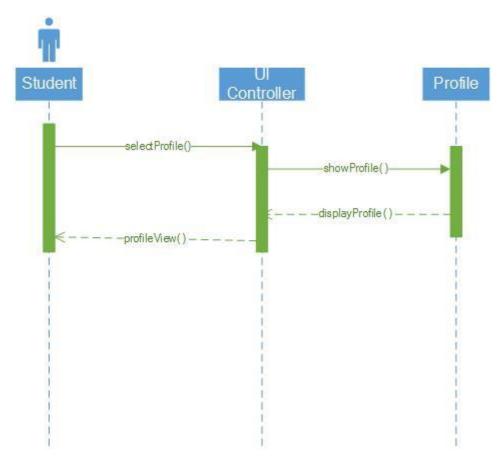
Fig. 4 Sequence diagram shows the control flow for "view user profile" task
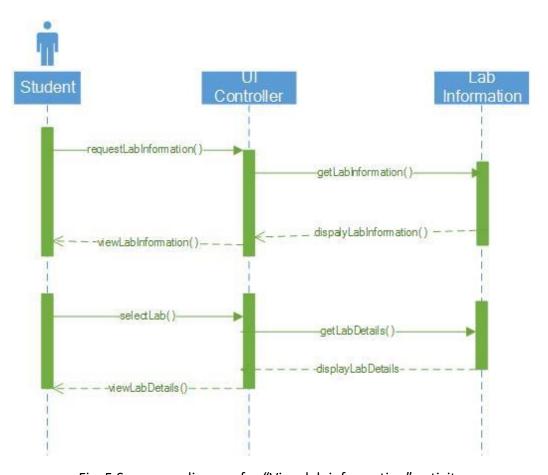
Fig. 5 Sequence diagram for "View lab information" activity
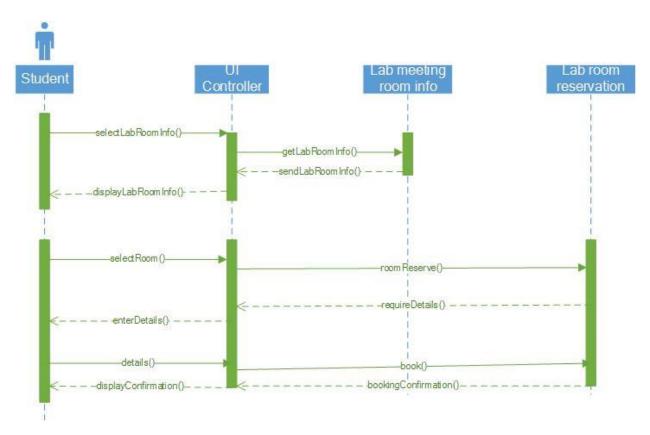
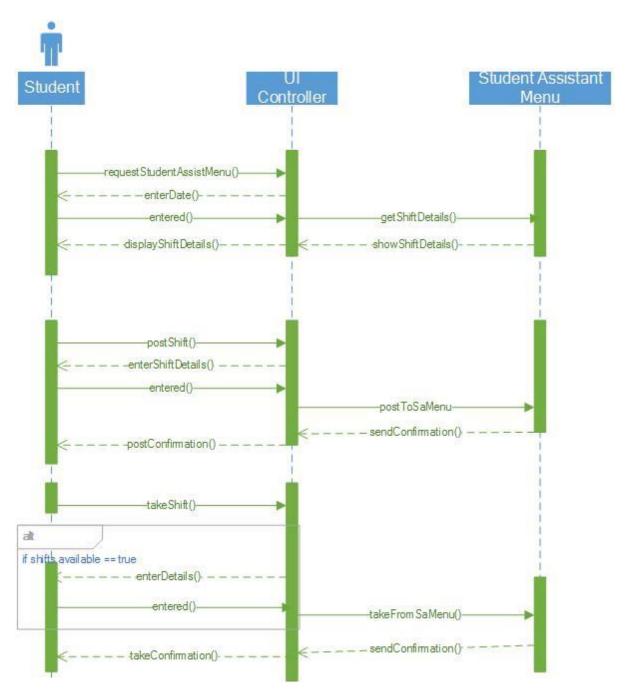Fig. 6 Sequence diagram for reserving library study room activity

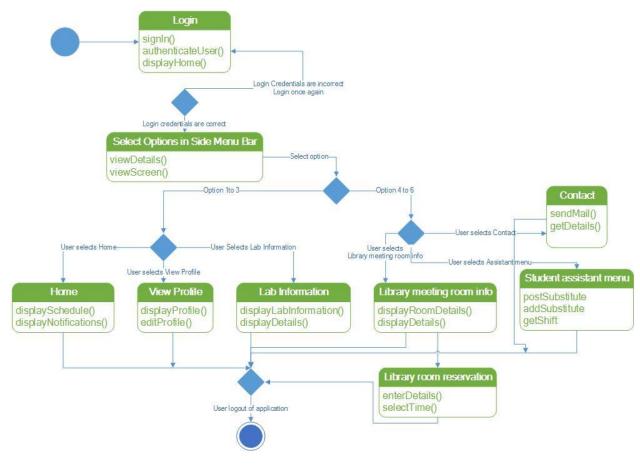Fig. 7 Sequence diagram for student assistant activities like post or take shifts.
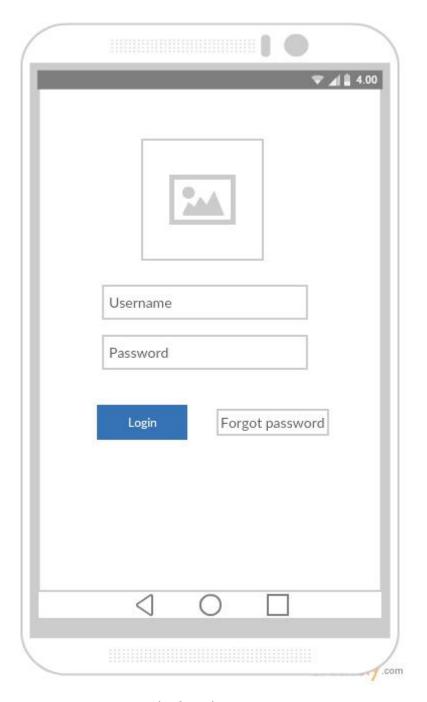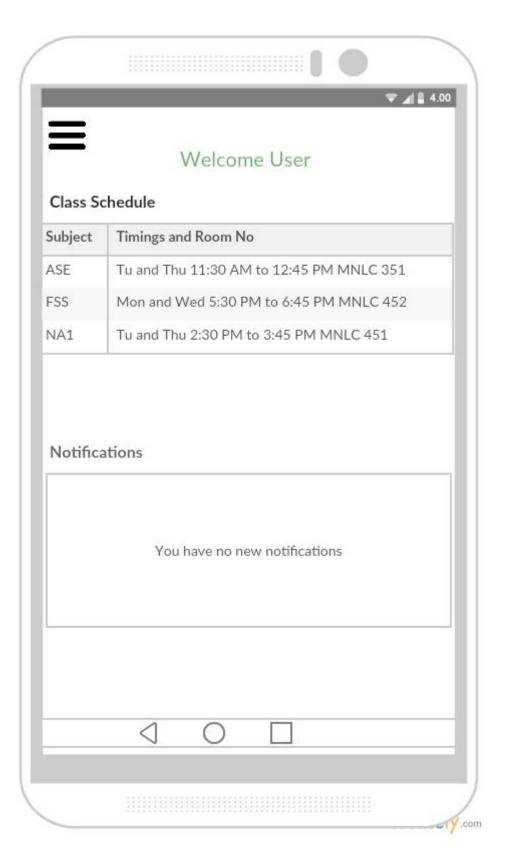
Fig. 8 State chart diagram for the application modules.

Wireframes of the application

Fig. 9 Login screen

Fig. 10 Main Home page

Fig. 11 Side menu bar

Lab Information

Lab	Capacity	In-Use
Bloch School 110	50	30
Royal Hall 203	40	30
MNLC 303	50	25
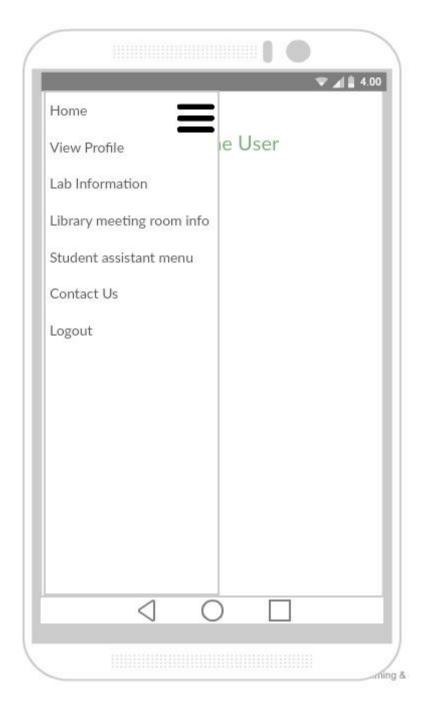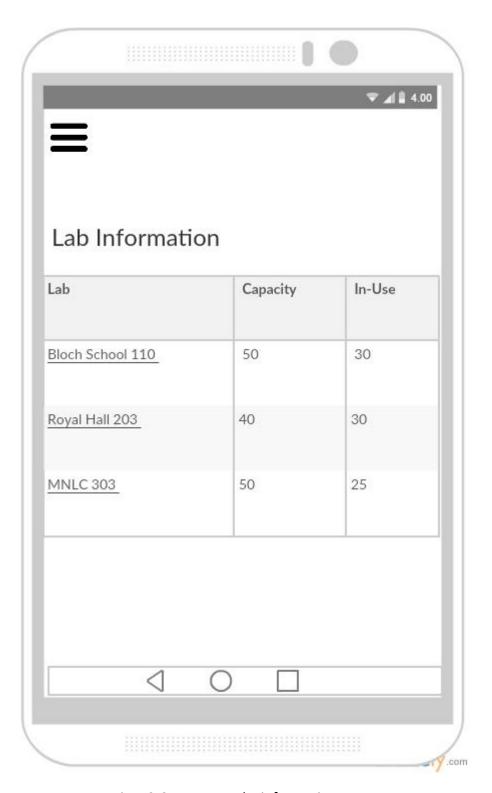
Fig. 12 Computer Labs information page
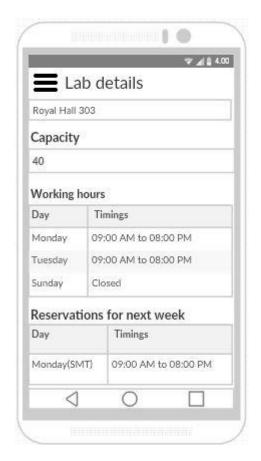
Fig. 13 Detailed information of a computer lab
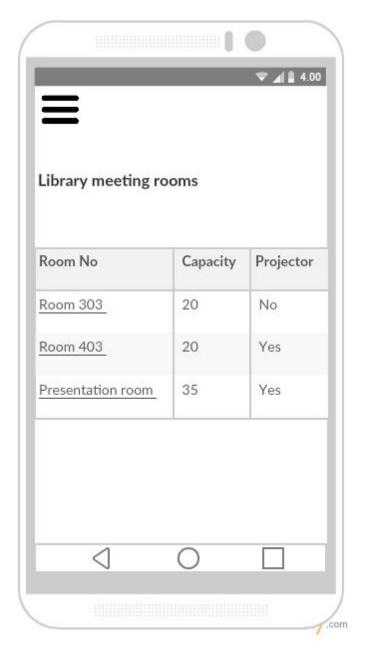
Fig. 14 Library study room information
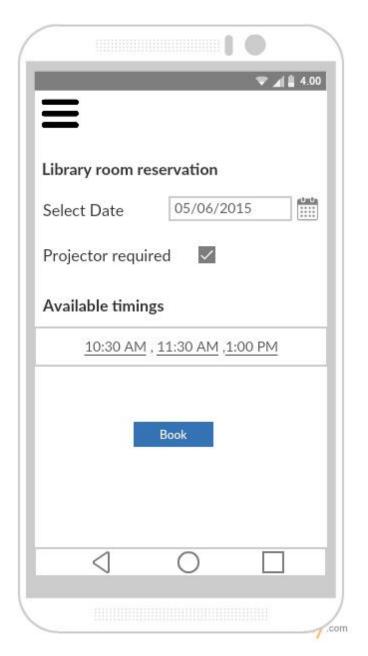
Fig. 15 Library room reservation page
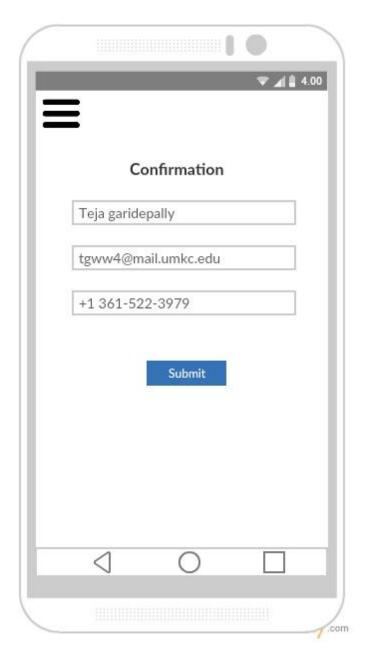
Fig. 16 Library room reservation form
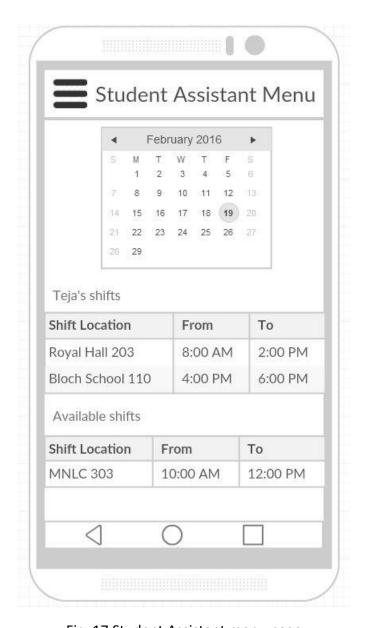
Fig. 17 Student Assistant menu page
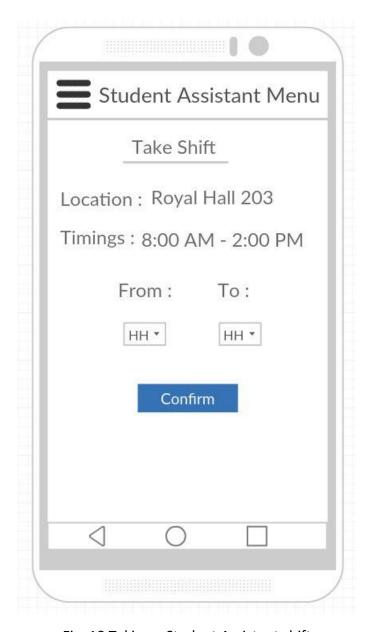
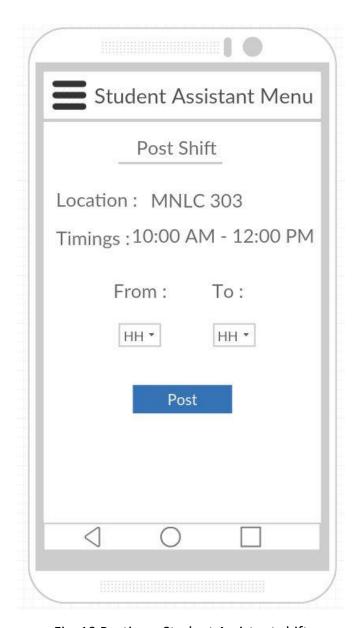Fig. 18 Taking a Student Assistant shift

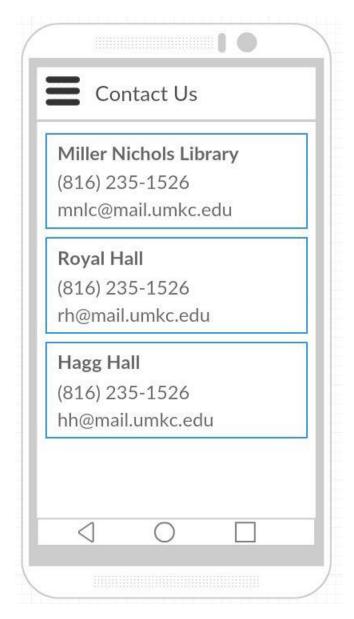Fig. 19 Posting a Student Assistant shift

Fig. 20 Student profile page

Fig. 21 Contact Us information page

User Stories:

- As a User, I can see the UI of StudentCompanion App.
- As a user, I can see the Login Screen.
- As a user, I can enter my Details and Login to the App.
- As a user, I can see the Side menu bar on Home Page with side menu options like Profile, Lab Information, Reserve Study Room, and SA Menu.
- As a User, I can see my Profile.
- As a User, I can see the Lab Information of specified lab.

- As a User, I can Reserve Study Room.
- As a User, I can choose the SA Menu option
- As a User, I can utilize the options in SA Menu.

As a user, I can Sign-out from the Application.

Deployment

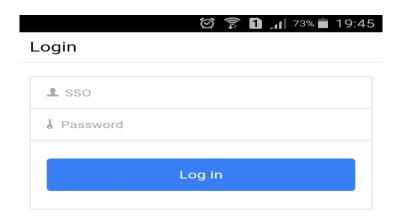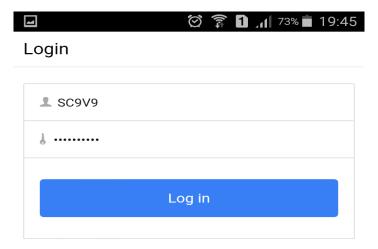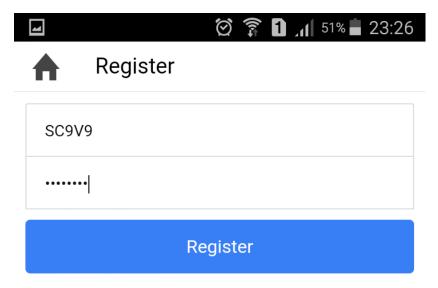Deployment of application to Mobile phone.

Screen Shots:

Fig: Login Screen

👤 SC9V9

🌡 ....

Log in

Invalid Credentials! Please try again....

Fig : Login Failed

Fig: Register
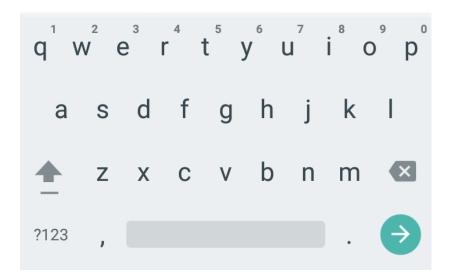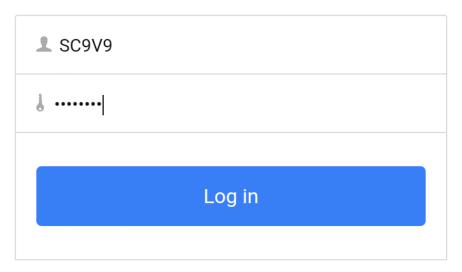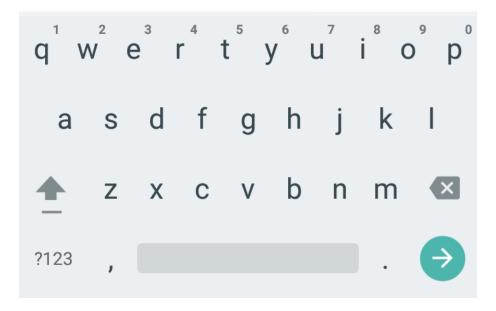
Fig: Login Success

Fig: Home Screen

Home

Profile

Lab Information

Reserve Study Room

SA Menu

Contact Us

Miller Nichols

800 E 51st Stree

Kansas City, MO

8162351526

Fig: Side Menu

☰ **User Profile** ⇥

First Name :

Sri Harsha

Last Name:

Chennavajjala

DOB:

1990-01-09

Email Id:

sc9v9@mail.umkc.edu

Fig Profile

Labs Information

Lab	Capacity	In-Use
Bloch School (Room 110)	40	31
Health Sciences Building (Room 3304)	50	38
Miller Nichols Library (Room 303)	40	15
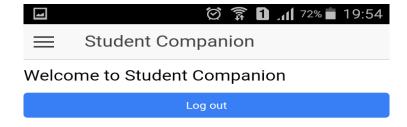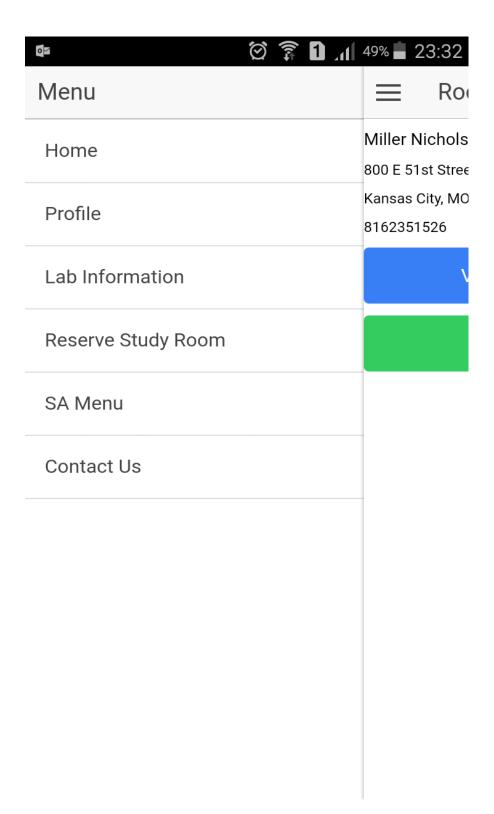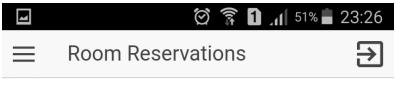
Fig:Labs

Miller Nichols Library
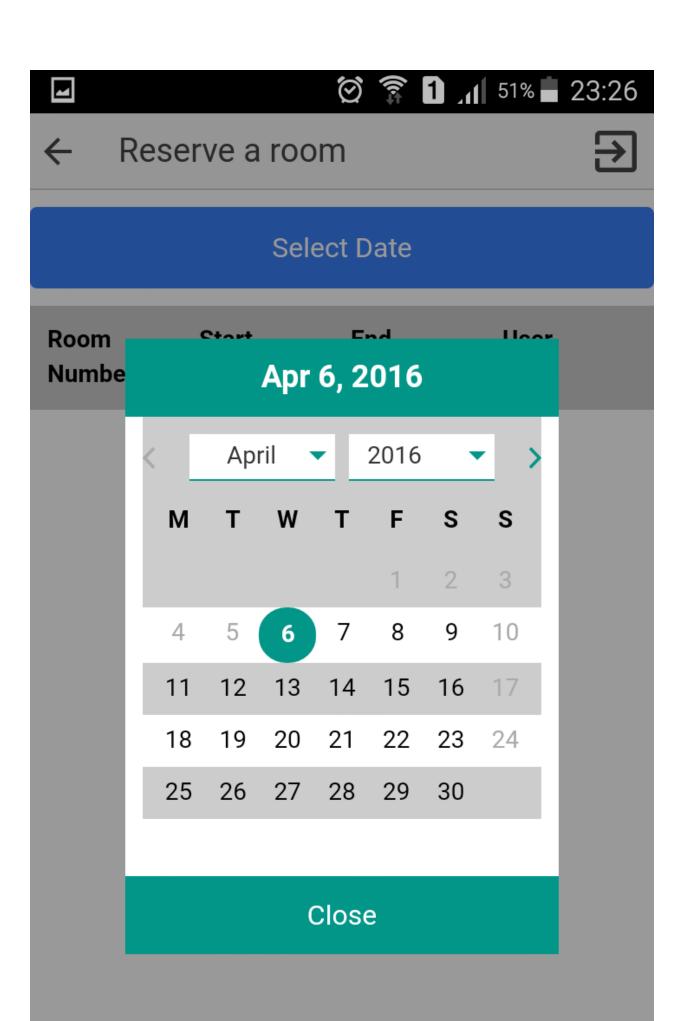
800 E 51st Street

Kansas City, MO, 64110

8162351526

View My Room Reservations
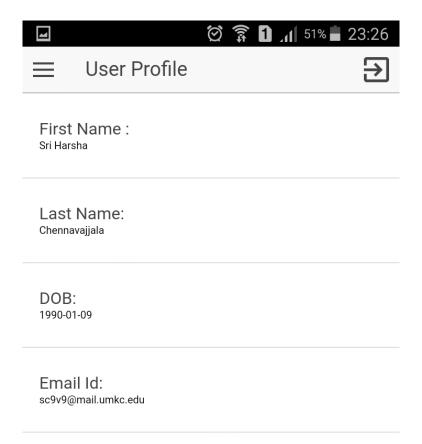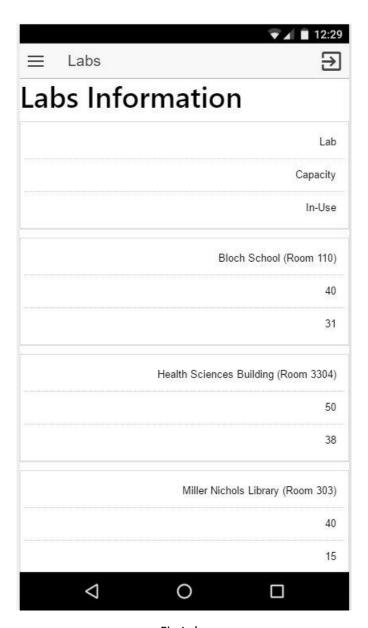
Reserve a room

← Reserve a room
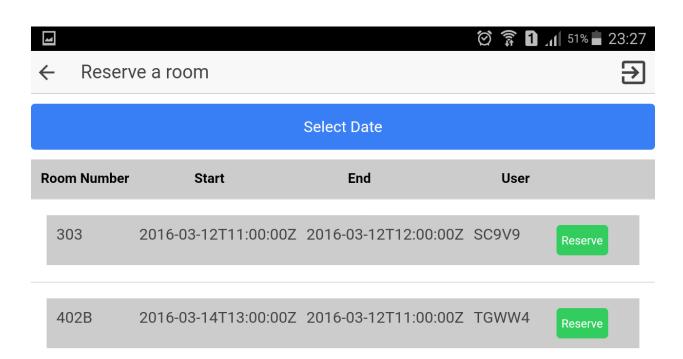
Select Date

Room
Numbe

Start   End   User

Apr 6, 2016

| ‹ | April ▾ | 2016 ▾ | › |

M	T	W	T	F	S	S
				1	2	3
4	5	**6**	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Close

← Reserve a room →
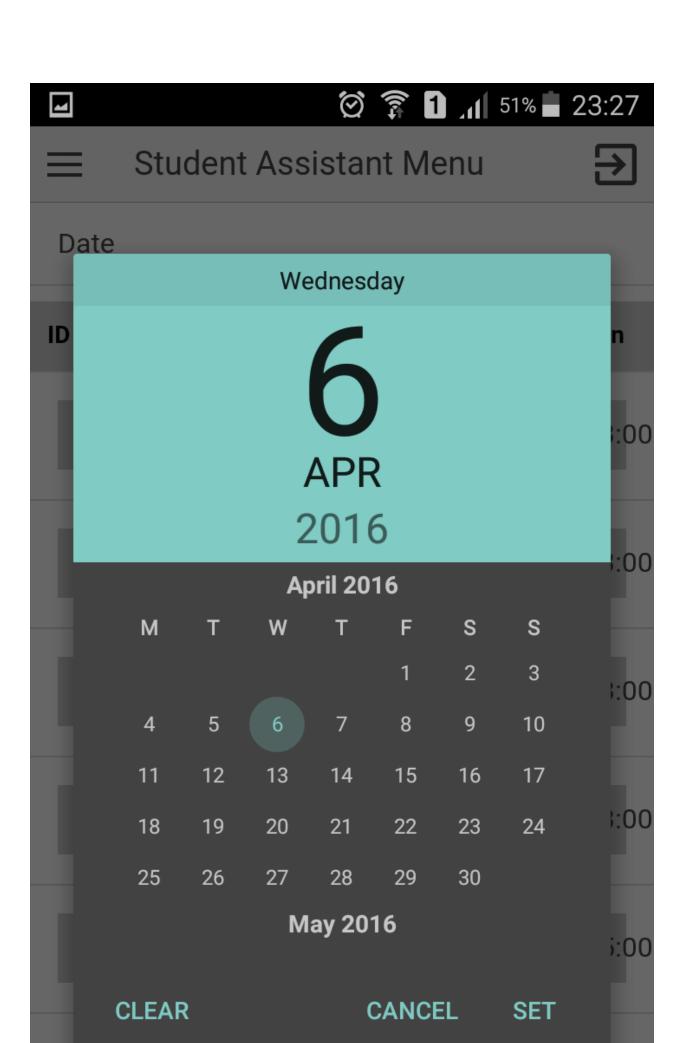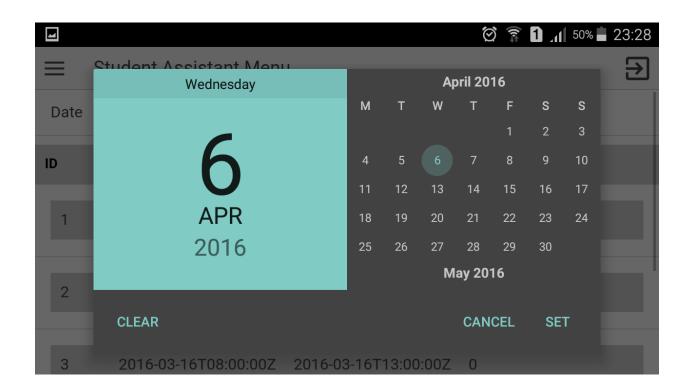
Room Number	Start	End	User
303	2016-03-12T11:00:00Z	2016-03-12T12:0(	
402B	2016-03-14T13:00:00Z	2016-03-12T11:(	
303	2016-03-12T10:00:00Z	2016-03-12T11:0(	
209	2016-03-16T09:00:00Z	2016-03-12T12:0(	
402B	2016-03-18T11:00:00Z	2016-03-12T13:(	

Room Number	Start	End	User	
303	2016-03-12T11:00:00Z	2016-03-12T12:00:00Z	SC9V9	Cancel Reserva
402B	2016-03-14T13:00:00Z	2016-03-12T11:00:00Z	TGWW4	Cancel Reserva
303	2016-03-12T10:00:00Z	2016-03-12T11:00:00Z	SC9V9	Cancel Reserva

Select Date

Room Number	Start	End	User	
303	2016-03-12T11:00:00Z	2016-03-12T12:00:00Z	SC9V9	Reserve
402B	2016-03-14T13:00:00Z	2016-03-12T11:00:00Z	TGWW4	Reserve

☰ Student Assistant Menu →

Date

ID   n

Wednesday

6

APR

2016

April 2016

M	T	W	T	F	S	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

May 2016

CLEAR   CANCEL   SET

☰ Student Assistant Menu   ➡

Date

ID

| Wednesday | April 2016 |

Wednesday

6
APR
2016

M	T	W	T	F	S	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

May 2016

CLEAR   CANCEL   SET

1

2

3   2016-03-16T08:00:00Z   2016-03-16T13:00:00Z   0

☰ Contact Us   ➡

Call center

The Call Center can be reached via email at callcenter@umkc.edu, by phone 816.235.2000

Mon-Thur	7:00 AM - 7:00 PM
Fri	7:00 AM - 5:00 PM
Saturday and Sunday	Closed

Call center

The Call Center can be reached via email at callcenter@umkc.edu, by phone 816.235.2000

Mon-Thur

7:00 AM - 7:00 PM

Fri

7:00 AM - 5:00 PM

Saturday and Sunday

Closed
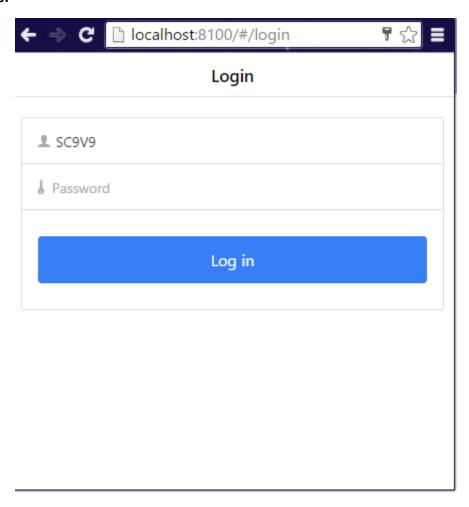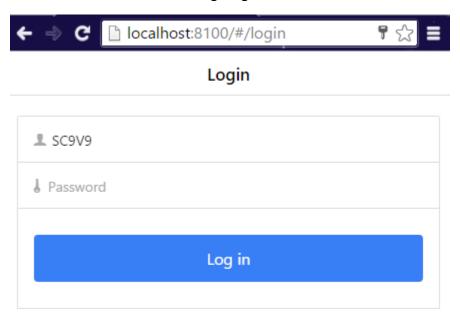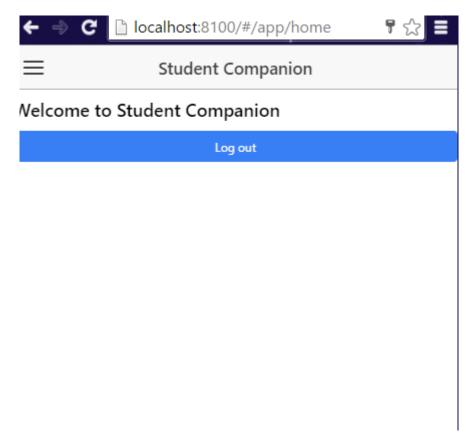
Web Application:

Screenshots:

Fig: Login

Fig: Login fail

Fig : Home
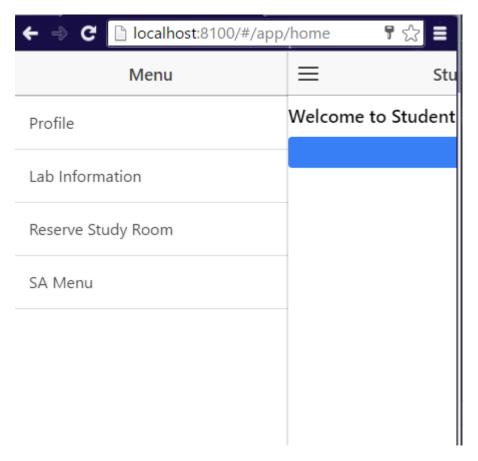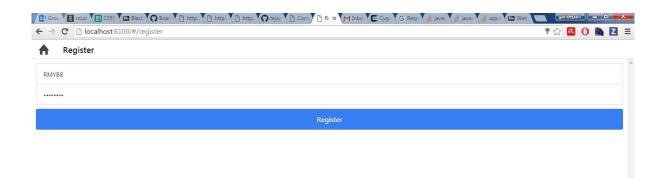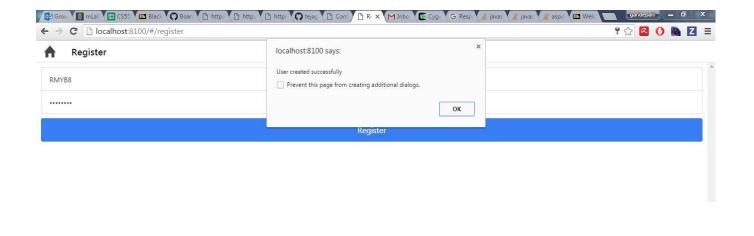
Fig: Side Menu
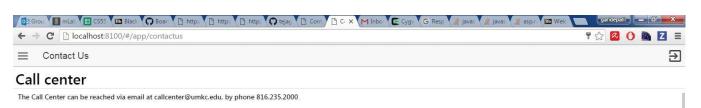
Register

RMYB8

········

Register

localhost:8100 says:

User created successfully

☐ Prevent this page from creating additional dialogs.

OK

localhost:8100/#/login

Login

RMYB8

········

Log in

Register Here

localhost:8100/#/app/contactus

☰ Contact Us

Call center

The Call Center can be reached via email at callcenter@umkc.edu, by phone 816.235.2000
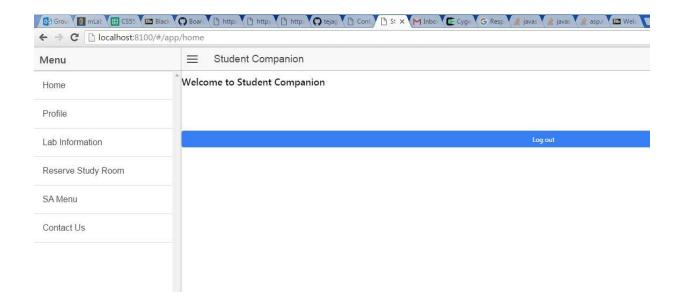
Mon-Thur	7:00 AM-7:00 PM
Friday	7:00 AM-5:00 PM
Sat and Sun	Closed

Project URL:

https://github.com/meetsriharsha/ASE_S16_G7/

Second Increment Document URL:

https://github.com/meetsriharsha/ASE_S16_G7/Documentation

Project Management
Work Completed

In detailed analysis of the system, environment and technical requirements for the application development. Project proposal documentation, Project tasks created in ZenHub and assigned the tasks to team members. Created the Project increment 1 document. All team members are involved in this task. Total time taken is 30 hours per person. Contributions: Harsha 30%, Teja 25%, Raj 25%, Suhas 20%.

Work To Be Completed

We need to work on login page creation (both UI and logic) and the backend database tables creation. We would like to use mongo db as our database server. Raj Kiran and Harsha will work on login page creation. Teja and Suhas will work on creation of sample database tables. Projected person participation: Harsha 25%, Teja 25%, Raj 25%, Suhas 25%.

Bibliography

- University of Missouri Library - http://library.umkc.edu/
- UMKC IS Labs - http://www.umkc.edu/is/labs
- Top Library management android applications -
http://appcrawlr.com/androidapps/best-apps-library-manager
- SFU: Student Companion -
https://play.google.com/store/apps/details?id=com.teamzeta.sfu&hl=en
- ZenHub - Agile project management tool for GitHub - https://www.zenhub.io