Android Application Prototype for Attendance, Grades and Health Care Services

GRADUATE PROJECT REPORT

Submitted to the Faculty of
the Department of Computing Sciences
Texas A&M University-Corpus Christi
Corpus Christi, Texas

In Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science

By

Lakshmi Gottumukkala
Spring 2017

Committee Members

Dr. Dulal C. Kar
Committee Chairperson

Dr. David Thomas
Committee Member
In any university, a mobile based application that can serve the needs of students, faculty members and university administration for tasks such as grades management, attendance management and booking appointments in healthcare services is helpful. It can greatly reduce the management overhead which is involved in manual processing of aforementioned tasks. When we talk about constant access and regular updates, the mobile phone is one such thing that comes to our mind. It is one such device which is being widely used now-a-days and most students have at least one. Currently, web based solutions are not efficient to run on handheld devices such as android based mobile phones and tablets. An android based application is required to provide more efficient access to these services. The objective of this project is to develop a mobile based application that serves as a platform for students, faculty members and university service providers to get easy access to the attendance management system, grades management system and university healthcare services. Using this application, the services can be availed at anytime from anywhere avoiding the need to always have a computer/laptop. This application would help the faculty members and students have better access to the services currently provided.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>ii</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>iii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>vi</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>x</td>
</tr>
<tr>
<td>1. BACKGROUND AND RATIONALE</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Existing Applications</td>
<td>2</td>
</tr>
<tr>
<td>2. NARRATIVE</td>
<td>4</td>
</tr>
<tr>
<td>2.1 Problem Statement</td>
<td>4</td>
</tr>
<tr>
<td>2.2 Motivation</td>
<td>4</td>
</tr>
<tr>
<td>2.3 Product description</td>
<td>4</td>
</tr>
<tr>
<td>2.4 Product Scope</td>
<td>5</td>
</tr>
<tr>
<td>2.4.1 Faculty Members</td>
<td>5</td>
</tr>
<tr>
<td>2.4.2 Students</td>
<td>5</td>
</tr>
<tr>
<td>2.4.3 Service Providers</td>
<td>5</td>
</tr>
<tr>
<td>2.5 System Requirements</td>
<td>6</td>
</tr>
<tr>
<td>3. DEVELOPED SYSTEM DESIGN</td>
<td>9</td>
</tr>
<tr>
<td>3.1 Android Studio</td>
<td>10</td>
</tr>
</tbody>
</table>
3.2 Amazon Web Services ................................................................. 11
3.3 MySQL .................................................................................. 11
3.4 Apache Tomcat ........................................................................ 11
3.6 ZBar bar code reader ............................................................... 11
3.7 Navigator API .......................................................................... 11
3.8 Eclipse .................................................................................... 11
3.9 Design Flow ............................................................................ 13
3.10 Use Case Diagram .................................................................. 13
3.11 Class Diagram ......................................................................... 14
3.12 User Interface .......................................................................... 17

4. IMPLEMENTATION OF APPLICATION MODULES ................... 37
4.1 XML Code for Android permissions ......................................... 37
4.2 Code for REST Service Client Point ....................................... 38
4.3 Code for Server Side Controller ............................................. 39
4.4 Code for Database Connections in Hibernate ......................... 40
4.5 Code for MD5 hashing ............................................................. 40
4.6 JavaScript Code for Getting User Client Location .................. 41
4.7 Code for Generating QR Codes .............................................. 41
5. TESTING AND EVALUATION .................................................................................42

5.1 Registration .................................................................................................42

5.2 Login ..............................................................................................................43

5.3 Reset Password ............................................................................................44

5.4 Attendance Module .......................................................................................45

5.5 Other Miscellaneous testing .........................................................................55

5.6 User Evaluations ...........................................................................................56

5.7 Limitations of this application ......................................................................56

5.8 Integrating with university software ............................................................56

5.9 Publish application in Google Play Store ....................................................56

6. CONCLUSION AND FUTURE WORK ............................................................56

BIBLIOGRAPHY and REFERENCES ..................................................................57
LIST OF FIGURES

Figure 3.1 Architecture of the application .................................................................9
Figure 3.2 Design Flow Diagram ..................................................................................12
Figure 3.3 Use Case Diagram for Faculty .....................................................................13
Figure 3.4 Use Case Diagram for Students .................................................................14
Figure 3.5 Class Diagram ...............................................................................................16
Figure 3.6 Faculty Login Screen ..................................................................................17
Figure 3.7 Faculty Registration Screen .........................................................................18
Figure 3.8 Forgot Password Screen (1) .......................................................................19
Figure 3.9 Forgot Password Screen (2) .......................................................................20
Figure 3.10 Faculty Home Screen ................................................................................21
Figure 3.11 Select Courses Screen ...............................................................................22
Figure 3.12 Course Joined Screen ...............................................................................23
Figure 3.13 Course Removed Screen ............................................................................24
Figure 3.14 Courses Screen .........................................................................................25
Figure 3.15 Login Page of Attendance Web Page .......................................................26
Figure 3.16 Share Location ...........................................................................................27
Figure 3.17 Select Course Screen ................................................................. 29
Figure 3.18 Displaying QR Code ................................................................. 30
Figure 3.19 Select Course Screen ................................................................. 31
Figure 3.20 Select Student Screen ................................................................. 33
Figure 3.21 Attendance Report (1) ................................................................. 34
Figure 3.22 Attendance Report (2) ................................................................. 35
Figure 3.23 Attendance Report (3) ................................................................. 36
Figure 3.24 Attendance Report (4) ................................................................. 37
Figure 3.25 Attendance Export File ............................................................... 38
Figure 3.26 Grades Screen (1) ....................................................................... 39
Figure 3.27 Grades Screen (2) ....................................................................... 40
Figure 3.28 Appointments Screen (1) ............................................................ 40
Figure 3.29 Appointments Screen (2) ............................................................ 41
Figure 3.30 Student Home Screen ............................................................... 42
Figure 3.31 Student Select Course Screen .................................................... 43
Figure 3.32 Scan QR Code Screen (1) .......................................................... 43
Figure 3.33 Scan QR Code Screen (2) .......................................................... 44
Figure 3.34 Student Grades (1) ................................................................. 45
Figure 3.35 Student Grades (2) ................................................................. 45

Figure 3.36 Manage Appointments .......................................................... 46

Figure 3.37 Announcements ..................................................................... 47

Figure 3.38 Security for Passwords ............................................................ 47

Figure 3.39 Security for Grades ................................................................. 47

Figure 3.40 Security for Attendance Records ............................................. 47

Figure 3.41 Code for avoiding SQL Injection ............................................. 47

Figure 4.1 Code for Android permission ................................................... 48

Figure 4.2 Code for REST service ............................................................. 48

Figure 4.3 Code for Server Side Controller .............................................. 49

Figure 4.4 Code for Database Connections in Hibernate ......................... 49

Figure 4.5 Code for MD5 Hashing ............................................................. 49

Figure 4.6 JavaScript Code for User Client Location ................................. 49

Figure 4.7 Code for generating QR Code ............................................... 49

Figure 5.1(a) Positive Case ...................................................................... 51

Figure 5.1(b) Positive Case ...................................................................... 51

Figure 5.1(c) Positive Case ...................................................................... 51

Figure 5.1(d) Negative Case .................................................................... 51
Figure 5.1(e) Negative Case ................................................................. 52
Figure 5.2(a) Negative Case ................................................................. 52
Figure 5.2(b) Negative Case ................................................................. 54
Figure 5.2(c) Negative Case ................................................................. 54
Figure 5.3(a) Positive Case ................................................................. 54
Figure 5.3(b) Positive Case ................................................................. 54
Figure 5.3(c) Positive Case ................................................................. 54
Figure 5.3(d) Negative Case ................................................................. 55
Figure 5.4(a) Positive Case ................................................................. 55
Figure 5.4(b) Negative Case ................................................................. 55
Figure 5.4(c) Negative Case ................................................................. 55
LIST OF TABLES

Table 1.1 Disadvantages of Existing Applications .................................................9

Table 5.1 Registration Testing ..............................................................................54

Table 5.2 Login Testing .........................................................................................54

Table 5.3 Attendance Module Testing .................................................................54

Table 5.4 Grades Module Testing .........................................................................55

Table 5.5 University Health Care Services Testing ...........................................55

Table 5.6 Application Security Testing ...............................................................55

Table 5.7 Forgot Password Testing ....................................................................55

Table 5.8 Users Evaluation ..................................................................................55
1. BACKGROUND AND RATIONALE

1.1 Introduction

As a part of Student-Faculty services both students and faculty members should have access to university facilities. Generally, from students’ perspective their focus will be on issues such as grades, attendance and how to claim their services from the university like university health center services. To fulfill their needs, Universities should maintain a portal with all the required services. Traditional methods of maintaining registries for attendance records or grades increase work load on the management. This can be handled effectively with the help of latest technologies such as web or mobile applications.

Classroom teaching always helps students in gaining knowledge through thorough discussions. Even if we rely only on the course material, students tend to gain more information in their classes. The instructor may go over examples and applications that students haven't experienced yet. It will present them in a unique way as compared in the text, and gives flexibility to students for discussion and raising questions. According to Woody Allen- “80% of success is showing up”. So, it is very important for students to attend their classes regularly.

The University health center provides the services regarding any health issues that a student or faculty encounters. It is difficult and time consuming for both students and faculty members to visit and take an appointment in the University health center. This explains the need for a better appointment scheduling system for the university.

To handle such issues, a model application is developed for the benefit of students, faculty and the University health care service management. This application will allow the faculty members to keep track of the students’ attendance and grades. Students can also view their
attendance and grades in more convenient way. This application also proves beneficial for the University health care management in managing their appointment scheduling process.

1.2 Existing Applications

There are few existing android applications discussed below that deal with handling above mentioned issues.

1.2.1 My Grades [3]

In application [3], users can view their grades and grades are automatically saved during their course period. The application also instructs a user as to what grade do they need to attain for future courses to maintain their grade point average (GPA). It helps reducing the time doing manual calculations to know what grade is required. This application still requires the data to be entered manually by the user.

1.2.2 Student Health 101 [4]

Islander Mobile is a university specific application [4], which is developed to stay connected with Texas A&M University-Corpus Christi. It has many services for users but one of the features in this application is “student health 101”, which shows images and posts events about various health issues.

1.2.3. Attendance [5]

Attendance [5] is an application for instructors to take attendance in class using mobile phone. With Attendance, we can create courses, terms and course occurrences, create groups for students with different schedules, add students to groups and finally take attendance. Attendance status can be configured for each occurrence and are “Present”, ...
“Absent” and “Late” by default [5]. The main disadvantage with this application lies in the fact that the faculty members needs to mark the attendance for each individual student manually one after the other.

Disadvantages of existing applications:

Table 1.1 Disadvantages of Existing Applications.

<table>
<thead>
<tr>
<th>Application name</th>
<th>Manual Effort</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>My Grades</td>
<td>Moderate</td>
<td>None</td>
</tr>
<tr>
<td>Student Health-101</td>
<td>High</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Attendance</td>
<td>High</td>
<td>None</td>
</tr>
</tbody>
</table>
2. NARRATIVE

2.1 Problem Statement

University services play an important role in providing effective interaction between students, faculty members and university administration. It is important for each student and faculty member to have a constant access to it. There are several web applications available to facilitate this interaction. However, that makes it a bit difficult to access all times unless they have a computer/laptop with data connection. Mobile phone browsers can also be used to open the website, but it is not convenient due to relatively small screen size.

For recording attendance, there is no smart system at many universities. Faculty members will pass a sign in sheet and every student must write their name and sign it which is highly inconvenient for the faculty members to manage attendance records. Moreover, students may also misuse it. For booking appointments in the healthcare services, they need to make a call and take an appointment. To reduce the manual efforts, an autonomous application such as a mobile based application will be useful that can deal with those challenges.

2.2 Motivation

Currently, there are several mobile based applications to address various university services as mentioned in Section 1.2. However, these applications are limited to specific tasks such as attendance, grades or healthcare appointments etc. So far, there is no single mobile based application that integrates the aforementioned tasks together to have a better exposure to the university services. This was the main motivation for developing an
android application, which provides a better user experience and will help in reaching multiple university services for students and faculty members.

2.3 Product Description

An android based application is developed to handle university services to facilitate students and faculty members. The application provides different level of abstraction to the user based on their status. The application is mainly entitled for three classes of users namely, Students, Faculty and Service providers. This android application helps faculty members to easily look their students list, record attendance, add grades, manage appointments at healthcare services. For students, this application will help to report and view attendance, view grades, manage appointments at healthcare services. Also, announcements will be sent to the students and faculty members from healthcare services about their appointments, and other related messages.

2.4 Product Scope

This application is developed for students and faculty members on the android operating system. Students can login through credentials provided and email. Moreover, faculty members will have a special token generated for their initial registration and later they can login through their credentials or email. University service providers can also use email and password to access this application. This application is compatible with different versions [1] of android starting from Honeycomb (Version 3.0 – 3.2) to Nougat (Version 7.0) [1].
2.4.1 Faculty Members

Faculty members can login to the application and access various features. Initially, while registering they can use a token for registration. Faculty members can view the registered students for their course, and enter grades for students in their list and can take attendance in class through a website. Each faculty member is given access to a website where they can login through credentials during the class time.

Taking the attendance is implemented using QR code scanning feature. Instructor need not scan each student’s QR code. To minimize the time lag, they are provided with the website login credentials using which they can login to the webpage that will display the QR code. This webpage has an internal mechanism of refreshing the QR code for every five seconds. There might be cases where a student might try to send a picture of the QR code to another student to mark the attendance. This frequent refreshing of the QR code reduces such false attendances. In addition, geo location constraints, are also be added to restrict the students from accessing the application outside the classroom. As soon as student captures the QR code his/her attendance is updated in the database and gets reflected in the application at both faculty and student end.

It is difficult to stand or walk-in to the health center and wait for the appointment for a longer period. In this application, a user can click on health center tab and schedule an appointment, reschedule the appointment, announcements will be sent to the users about their appointments.
2.4.2 Students

Students can register to the application using their email-id and password and get access to numerous services. They can view courses offered by a faculty member and choose according to their requirement. Students can view grades entered by the instructor. Students can also report attendance as faculty member will project the QR code and students can use the camera option in the application to capture the code and their attendance will be recorded and updated. For University health center services, we can click on health center tab and schedule appointment or cancel appointment and reschedule the appointment, and announcements will be sent to the students.

2.4.3 Service Providers

Services providers can login using their email-id and login to the application and provide services. Service providers only deal with health care services. They can manage booking appointments, post details about the health care programs and reminders will be sent to the both faculty and students about their appointments.

2.5 System Requirements

The following are required to develop this android application.

- Operating System: Windows
- Database: Amazon web services [6]

Amazon Web Services (AWS) is a web service and cloud computing resource provider by Amazon. It is an on-demand computing platform. AWS provides huge computing capacity faster and cheaper to many client companies.

- IDE: Android Studio
Android Studio provides Gradle based support, Lint tools for performance analysis, rich layout editor for drag-and-drop UI components and an Android Virtual Device emulator.

- **Software: Android SDK (Software Development Kit), Java SE 8**
  
  Java is a general-purpose computer programming language that is an object oriented. It is platform independent that works on Java Virtual Machine (JVM). Java SE 8 is the latest version released on 18 March 2014.

- **Android Device: Android phone**
  1. A Personal Computer
  2. Android Mobile Device
3. DEVELOPED APPLICATION DESIGN

The architecture of the application is shown in Figure 1. The application is comprised of Android User interface, Rest services, GCM server, JSON, Tomcat server, MySQL database and Amazon Web Service provider and Barcode API.

![Figure 3.1. Architecture of the Application.](image)

In this application, the Android client will call a Servlet designed inside the Amazon Web Services (AWS). The call is used with the help of the REST services. The Servlet then triggers the call to Hibernate to obtain the data from the databases. The Hibernate can forward the object containing information about the data to be collected. The Database
actions can be performed using MySQL and the required data can be collected. The Hibernate receives the data and forwards it to the Servlet which converts it to readable format using the JSON service. Finally, the data from the JSON is displayed in the Android. The Servlet and the Hibernate framework is hosted by the Tomcat server and by using the services provided by the Amazon Web Services.

3.1 Android Studio [12]

Android studio is the official integrated development platform for Android platform. It was designed based on IntelliJ IDEA. The first stable build of android studio was released in December 2014 and from then it replaced Eclipse Android Development Tools (ADT) as Google’s primary IDE for native android application development. Features like Graphical Layout editor, Multiple APK generation, Gradle-based build support and Code Refactoring makes it easy to work on projects and manage resources. We can also test the applications developed on a virtual device provided by Android Studio [12].

3.2 Amazon Web Services[6]

Amazon Web Services (AWS) is a subsidiary of the leading online shopping company, Amazon.com. This offers a high-level Cloud Computing platform. Their services can be used for numerous services such as computing, analytics, deployment and mobile developer tools [6].

3.3 MySQL

MySQL is an open source Relational Database Management System. It was established in May 1995 to provide the various database operations. It acts as a support for various applications where data retrieval plays a key role.
3.4 Apache Tomcat [11]

Apache Tomcat [11], also known as Tomcat Server, is an open source Java Servlet Container developed by the Apache Software Foundation. This is used to implement several Java Enterprise Edition specifications including Java Servlets, Java Server Pages and Web Sockets. It works on HTTP web server technology.

3.5 ZBar bar code reader [7]

ZBar is an open source software suite for reading bar codes from various sources, such as video streams, image files and raw intensity sensors. It supports many popular symbologies including EAN-13/UPC-A, UPC-E, EAN-8, Code 128, Code 39, Interleaved 2 of 5 and QR Code. ZBar is licensed under the GNU LGPL 2.1 to enable development of both open source and commercial projects [7].

3.6 Navigator API [8]

The Navigator interface [8] represents the state and the identity of the user agent. It allows scripts to query it and to register themselves to carry on some activities. A Navigator object can be retrieved using the read only “window.navigator” property. Navigator does not inherit any properties, but implements those defined Navigator Id, Navigator Language, Navigator Online, Navigator Cookies etc.

3.7 Eclipse [9]

Eclipse [9] is famous for Java Integrated Development Environment(IDE). It can easily combine language support and other features into any of the default packages in it. Eclipse market place allows for virtually unlimited customization and extension. It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse
is written mostly in Java and its primary use is for developing Java applications, but it may also be used to develop applications in other programming languages via plug-ins, including Ada, ABAP, C, C++, COBOL, D, Fortran, Haskell, JavaScript, Julia, Lasso, Lua, NATURAL, Perl, PHP, Prolog, Python, R, Ruby.

3.8 MP Android Charts

MP Android Chart is a powerful and easy to use chart library for Android, supporting line, bar, scatter, candlestick, bubble, pie and radar charts (spider web), as well as scaling, dragging (panning), selecting and animations. It works on Android 2.2 (API level 8) and upwards.

3.9 Design Flow

Figure 3.2 shows the design flow for our application. The “student” or “faculty member” or “other members” after installing the application, logs into the application using their email id and password. If the user is new to the application, they will need to register first. If the registration is successful they will be redirected to the home page. If any user forgets password, they can choose “forgot password?” option and input their email id, if the email id is valid, then a mail is sent to them with a link to reset their password. Once the login credentials are validated, the users are taken to the home page and user the modules present in the application. At the end, the student can logout of the application using the logout option.
3.10 Use Case Diagram

Figure 3.3 shows the Use case diagram for the Faculty. The users will register himself/herself if he/she is using the services for the first time or logs in using their email id and password or uses forgot password if he/she forgets his/her password. The students and faculty members can look for the courses, record and view attendance records, update and views grades, book appointments with health care services, reschedule appointments and receive notifications from health care services.
Figure 3.3 Use Case Diagram for Faculty.

Figure 3.4 shows the Use case diagram for Students. The users will register himself/herself if he/she is using the services for the first time or logs in using their email id and password or uses forgot password if he/she forgets his/her password. The other members (Health care service members) can go through the requested appointments. They can accept any appointment if they have vacancy, otherwise they cancel the request. They will also send announcements to the users regarding healthcare issues.
3.11 Class Diagram

Figure 3.5 represents class diagram of students’, faculty members and health care services. If a Faculty member wants to create attendance reports then the function “attendanceActivity()” is called. Similarly, all the functions mentioned below will be called when Faculty member wants to create appointment, fetch appointments and assign grades. All the functions which are represented in the class diagram are finally connected to the database. This database uses REST services and pops out the data on to the application.
3.12 User Interface

There are mainly three different classes of users in this application namely “Faculty”, “Students” and “Others”.

3.12.1 Login Screens

Figure 3.5 shows the first screen that appears in the application. Basically, this allows accommodates three various kinds of login privileges. So, Figure 3.6 shows the screen which appears when user tries to use this application with faculty member privilege.
There are similar kind of screens for both students type of users and others type of users. If the user is old member they can directly login through their credentials or if they are new to the application they can get registered and then use the application with their credentials.

### 3.12.2 Registration Screen

Figure 3.7 shows the registration screen. When the user is new to the application, they can just click on the register button. Then, as we can see in Figure 3.7, they must enter their username, unique faculty id, Valid email address and password. If they meet all requirements mentioned in Figure 3.7, they will be registered successfully and can login to this application.

The other two types of login privileges also has the same kind of the registration page for the first time users.
3.12.3 Reset Password

Figure 3.8 and Figure 3.9 will show the screens to reset the password if any user forgets it. The user needs to give their email id as input.

Once the user clicks on “forgot password”, the screen will appear as shown in Figure 3.8. Here, they need to enter valid email id then they need to click on “continue” and the screen will appear as shown in Figure 3.9. Shortly, users will receive an email to their registered email id and they will be provided with an OTP. In order to reset their password, users need
to enter that OTP and they can reset their password. After this, users can login into the application with the new password.

3.12.4 Faculty Home

Figure 3.10 shows the faculty home screen which shows immediately after entering the login credentials. After seeing this screen faculty members must select course that they are offering using select course tab.

Figure 3.11 shows the screen that appears after clicking select course tab. The faculty can enter the department in which he/she is teaching to filter the courses available for that department. Here they can select the courses that they are offering.

![Faculty home screen](image1.png)  ![Select course screen](image2.png)

**Figure 3.10 Faculty home screen.**  **Figure 3.11 Select course screen.**
After selecting a course, they can see a pop-up which says course has been added and after teaching that course for a semester if the faculty member wants to remove that course details we next semester starts he/she can leave the course and automatically the data related to the removed courses will no longer appear in the application.

The following screens in Figure 3.12 and Figure 3.13 appear after clicking “join a course” and similarly after clicking “remove course”. If the faculty member is teaching more than one course, then she/he can select multiple courses.

![Figure 3.12 Course joined screen.](image1)  ![Figure 3.13 Course removed screen.](image2)

After a course selection is done by a faculty member, the courses will appear to the students and they can join the courses which they have registered. If the students add the course of
a faculty member then the faculty can see number of students added to his/her course and their details.

Figure 3.14 we can see a sample screen which is given, it has three courses being displayed. Figure 3.14 appears if the students add the course which is offered by a faculty member. This screen typically displays the course name and if any students adds the course then it will show the number of students, if nobody adds it then it shows a zero. If for any reason students remove a course from his/her account then that will be immediately reflects here.

![Courses screen](image.png)

Figure 3.14 Courses screen.
After a faculty member has his/her student details they can use this application for various purposes like recording attendance, view attendance analysis monthly, weekly etc., to export attendance file.

To record attendance the faculty member needs to open the web page which is linked with the application. The login details for this webpage is same as the login details of the application. By logging in to that page it will prompt for accessing location, on clicking “accept”, the location will be captured by the webpage and then faculty member must select the course for which they want to record attendance. Figure 3.15, Figure 3.16 and Figure 3.17 demonstrate the process more clearly.

Figure 3.15 shows the login page of attendance webpage. Here, faculty member should enter login credentials.

Figure 3.15 Login page of Attendance web page.
Figure 3.16 shows the attendance webpage, where it will prompt for location access permission. The faculty member should allow webpage to access the location so that it starts generating QR code for that location and students can use that to note their presence in the class.

![Image of the attendance webpage](image)

**Figure 3.16 Share location screen.**

After the permissions are given they must select the course for which he/she wants to generate the QR codes for recording attendance. Figure 3.17 shows the course selection screen.
Figure 3.17 Select course screen.

Figure 3.18. shows how QR codes are generated by using location based services. Students can record attendance only in that location. By this we can avoid students giving attendance for other student who are not present in the class.

Figure 3.18 Displaying QR Code screen.
The information of all the students who successfully marked their attendance gets reflected in the faculty end. The following screens as shown in Figure 3.19 and Figure 3.20 represent the attendance analysis screens.

If the faculty members click on the attendance reports tab, they will be able to see all the courses that are offered by them. Faculty members should click on the course to look for students enrolled in that courses, later they need to click on each student to view their attendance details.

![Select course screen](image1)
![Select student screen](image2)

**Figure 3.19 Select course screen.**  **Figure 3.20 Select student screen.**

Figure 3.19 and Figure 3.20 shows the different classes offered by a faculty member and number of students he/she has.
Faculty members can see attendance of their students in various levels like the current day’s attendance by clicking on “today”, entire week attendance by clicking on “this week” tab and so on. Figure 3.21 represents today’s attendance and Figure 3.22 represents a drop-down box which helps faculty member to select according to his/her choice. In Figure 3.23 the screen that appears is the one which appears after clicking details tab in the attendance module. Figure 3.24 is the screen which appears after clicking export button.

Figure 3.21 Attendance report (1).  Figure 3.22 Attendance report (2).
By clicking on the export button the faculty member will be able to export all the details of the student to his/her email. The following Figure 3.25 will represent the exported file.
Faculty member also has an opportunity to assign grades to their students using this application. Figure 3.26 describes about diverse types in the grade assigning. In the grades assigning module, faculty members are given the opportunity to enter the assignment name, maximum possible points for the assignment. After entering the maximum possible points, Faculty members can select any one option from the three options provided shown in the Figure 3.27. If the students have completed the assignment on time, or is it a late submission or not submitted can be selected and grades can be assigned accordingly.

Figure 3.26 Grades screen (1).
Figure 3.27 Grades screen (2).
By using this application, faculty members can also book appointments with the health care services and they can also receive announcements from them. A faculty member can request for an appointment using this application. As soon as an appointment is requested the screen as shown in Figure 3.28 appears. If user is looking for a particular time slot and that time slot is already filled, then that time slot will not appear in the time picker. So, after requesting for the appointment, the request will be sent to health care services administration and the application prompts us in “pending status” which is shown in Figure 3.29. As we know that we may experience delays, so if that time slot is empty, then the healthcare service people accept it or they will reject it. If they accept the request then the appointment is successfully booked.

Figure 3.28 Appointments screen (1).

Figure 3.29 Appointments screen (2).
3.12.5 Student Home

Students should add the courses that they have already registered. They can add and delete the course at any point of time. They can report their attendance using this application. Moreover, there are additional features provided by the application such as students can view their attendance analysis, look at their grades, students also can book appointments and get the announcements from the health care services.

Figure 3.30 shows student home screen. Figure 3.31 shows the students that how they can select their required department and later add or drop classes according to their requirement.
Students must capture the QR code which is projected by faculty member to record their presence in the class. Students can also look for various levels of attendance analysis as discussed earlier in the faculty home section. They can also export their attendance details to their email if required.

Figure 3.32 shows how to capture attendance and how it works. After capturing the QR code, students can immediately open the application and check whether the attendance is recorded or not. They can also check their percentage of attendance till date. Figure 3.33 shows the screen which appears after reporting attendance.
Students can also check the grades which are assigned by the faculty members. As soon as students click grades tab, they are able to see all the courses they are registered in. They need to click on a specific course to look for more details about their grades. As mentioned in Section 3.12.4 that, faculty members have access to bookings in healthcare services, similarly students also have the access to the healthcare services and their functionality is same in both student end and faculty end.

Figure 3.34 shows how students can look at their grades. Figure 3.35 shows the grade of the student in a specific course.
**3.12.6 Others Home**

Others home feature is strictly designed for health care services. University health care services basically has two modules in it. Figure 3.36 shows the first module is health care services management in which all the requests sent from the faculty members and students will be considered and ordered according to their requests. Some requests are accepted and few of them are not accepted as per the health care services policy. Figure 3.37 shows the second module, which is announcements module. In this module, the health care center can send any important announcement to all the users.

**Figure 3.36 Manage Appointments.**

**Figure 3.37 Announcements.**
3.12.7 Application Security

Security is very important for any application which deals with sensitive information like passwords, grades and attendance. For this application security is provided by using MD5 algorithm for passwords.

MD5 hashing algorithm is a one-way cryptographic function that accepts a message of any length as input and returns as output a fixed-length digest value to be used for authenticating the original message.

In Figure 3.38 in the column named as “token”, a random code is shown. These random codes generated are passwords of different users which are shown in encrypted form. By using this technique all the passwords and user names are encrypted to provide security for the application. In situations where database gets crashed or if there are any “Man -in -the- Middle attacks”, intruder cannot see the passwords by using this technique.

![Figure 3.38 Security for Passwords.](image)

Figure 3.39 shows security provided for the grades and the backend database view. In the database at the backend we can see assignment number, maximum possible points and id. All the grades declared by using “Volatile” datatypes. Due to this, grades will not appear to anybody except the student.
Figure 3.39 Security for Grades.

Figure 3.40 shows the security provided for the attendance records. Figure 6.3 shows the database view. Here we have column “token” which represents unique random number. This random is generated by taking “date + longitude + latitude + system time + random token”. Intruders cannot identify the attendance values as it is encrypted using different parameters.

Moreover, to provide additional security to this application all the server side code which connects “My SQL” to database is written with “JAVA Prepared Statements”. By using prepared statements “SQL Injections” can be avoided.

Java PreparedStatement [13] represents a precompiled SQL statement that can be executed multiple times without having to recompile for every execution. Figure 3.41
shows code which is not vulnerable to SQL Injection because it correctly uses parameterized queries.

```java
PreparedStatement stmt = connection.prepareStatement("SELECT * FROM users WHERE useri
stmt.setString(1, userid);
stmt.setString(2, password);
ResultSet rs = stmt.executeQuery();
```

**Figure 3.41 [13] Code for avoiding SQL Injections.**

By utilizing Java's PreparedStatement class, bind variables and the corresponding setString methods, SQL Injection can be easily prevented.

Additionally, this application uses Amazon web services as backend web server. Amazon web server provides secure cloud services using DOD Cloud Security [14]. The new AWS services added to the authorization include advanced database, low-cost storage, data warehouse, security, and configuration automation solutions that will help users with IL4 workloads increase the productivity and security of their data in the AWS Cloud. For example, with AWS Cloud Formation users can deploy AWS resources by automating configuration processes. AWS Key Management Service enables users to create and control the encryption keys to encrypt data.
4. IMPLEMENTATION OF APPLICATION MODULES

4.1 XML Code for Android Permissions

Figure 4.1 shows the code for Android permissions, which is used to run this application successfully, user is asked for permissions namely, internet, camera, location and to write and read to external storage.

```xml
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission INTERNET" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

Figure 4.1 Code for Android permission.

4.2 Code for Rest Service Client Point(Android)

Figure 4.2 shows the code for Rest Service Client Point(Android). REST stands for Representational State Transfer. It relies on a stateless, client-server, cacheable communications protocol -- and in virtually all cases, the HTTP protocol is used.

```java
@AutoWired
UserService userService;

@RequestMapping(value = "login", method = RequestMethod.POST)
public @ResponseBody User isUserValid(@RequestBody String params) {
    Map<String, String> paramsmap = Util.splitQuery(params);
    User user = new User();
    user.setEmail(paramsmap.get("email"));
    user.setPassword(paramsmap.get("password"));
    user.setRole(paramsmap.get("role"));
    return userService.isUserValid(user.getEmail(), user.getPassword(), user.getRole());
}
```

Figure 4.2 Code for REST service implementation.
4.3 Code for Server Side Controller

Figure 4.3 shows the code for server side controller. The server side code helps in accessing the location and saving it in the database. Later, this saved location is used to match if the user it trying to report attendance from outside of the class.

```javascript
var qrCode = new QRCode(document.getElementById("qrCode"), {
    width: 350,
    height: 350
});

getLocation();
var interval = null;
$("#courses").on('change', function(e) {
    console.log(globalPosition);
    $("#qrCode").addClass("hide");
    clearInterval(interval);
    var optionSelected = $("option:selected", this);
    var course = this.value;
    var today = new Date();
    today.setHours(0, 0, 0, 0);
    var date = Math.round(today.getTime() / 1000);
    var userid = $("#userid").val();
    $.ajax(
        url: 'startattendance',
        type: 'POST',
        dataType: 'html',
        data: 'professorid=' + userid + '&courseid=' + course + '&timestamp=' + date,  
        success: function(data) {}
    
    
    

    });

    var token = Math.floor(Math.random() * 90000) + 10000;
    makeCode(date, course, token);
    interval = setInterval(function() {
        token = Math.floor(Math.random() * 90000) + 10000;
        makeCode(date, course, token);
        }, 7000);

});
```

**Figure 4.3 Code for QR Code generation.**
4.4 Code for Database Connections in Hibernate

Figure 4.4 shows the code for Database Connections in Hibernate. Hibernate code is used to pull data from the database and validate it according to the requirements.

```java
URL url;
url = new URL(URL + controller);
HttpURLConnection connection =
    (HttpURLConnection) url.openConnection();

connection.setRequestMethod(HTTP_METHOD_POST);
connection.setRequestProperty(
    HTTP_REQUEST_PROPERTY_CONTENT_TYPE,
    HTTP_REQUEST_PROPERTY_CONTENT_TYPE_VALUE_URL_ENCODED);

connection.setDoOutput(true);
connection.setUseCaches(false);
connection.setAllowUserInteraction(false);
connection.setInstanceFollowRedirects(false);

DataOutputStream requestData = new DataOutputStream(
    connection.getOutputStream());
requestData.writeBytes(keyvaluepair);

requestData.flush();
requestData.close();

BufferedReader reader = new BufferedReader(          
    new InputStreamReader(connection.getInputStream()));
```

**Figure 4.4 Code for Database connection.**

4.5 Code for Encrypting Passwords Using MD5

The most important security aspect of this application lies with securing the account passwords of all the users. To secure these passwords, we have implemented an MD5 Hashing algorithm to store the passwords in the database.
Figure 4.5. shows the MD5 hashing algorithm is a one-way cryptographic function that accepts a message of any length as input and returns as output a fixed-length digest value to be used for authenticating the original message.

```java
String plaintext = "your text here";
MessageDigest m = MessageDigest.getInstance("MD5");
m.reset();
m.update(plaintext.getBytes());
byte[] digest = m.digest();
BigInteger bigInt = new BigInteger(1, digest);
String hashtext = bigInt.toString(16);
// Now we need to zero pad it if you actually want the full 32 chars.
while(hashtext.length() < 32 ){
    hashtext = "0"+hashtext;
}
```

**Figure 4.5 MD5 Encryption code.**

### 4.6 JavaScript Code for Getting User Client Location

Figure 4.6. shows the code for getting client location. The “Navigator.geolocation” read-only property returns a Geolocation object that gives Web content access to the location of the device. This allows a website or app to offer customized results based on the user's location.

```javascript
var globalposition;

function getLocation() {
    if(navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(showPosition);
    } else {
        var output = "Geolocation is not supported by this browser.";
    }
}

function showPosition(position) {
    var output = "Latitude: " + position.coords.latitude + "<br>Longitude: " + position.coords.longitude;
    globalposition = position;
}
```

**Figure 4.6 Code for getting Client location.**
4.7 Code for Generating QR Codes

The following code in Figure 4.7 generates random QR codes after every 5 seconds. These generations are done by taking some important constraints into account such as date, course id, token, latitude, longitude.

```xml
<bean id="entityManagerFactoryBean"
     class="org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean">
     <property name="dataSource" ref="dataSource"/>
     <property name="packagesToScan" value="com.attendance.management.model"/>
     <property name="jpaVendorAdapter">
     <bean class="org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter"/>
     </property>
     <property name="jpaProperties">
     <props>
     <prop key="hibernate.hbm2ddl.auto">validate</prop>
     <prop key="hibernate.dialect">org.hibernate.dialect.MySQL5Dialect</prop>
     </props>
     </property>
</bean>

<bean id="dataSource"
     class="org.springframework.jdbc.datasource.DriverManagerDataSource">
     <property name="driverClassName" value="com.mysql.jdbc.Driver"/>
     <property name="url" value="jdbc:mysql://localhost:3306/attendancemanagement"/>
     <property name="username" value="root"/>
     <property name="password" value=""/>
</bean>

<bean id="transactionManager" class="org.springframework.orm.jpa.JpaTransactionManager">
     <property name="entityManagerFactory" ref="entityManagerFactoryBean"/>
</bean>

<mvc:annotation-driven />
<tx:annotation-driven />
```

Figure 4.7 Code for generating QR Code.
5. TESTING AND EVALUATION

This application is tested on Nexus 5X, with android version 7.1.1, Nexus Tab (Android version-5), MotoG 4 (Android version-6), BLU LIFE PURE (Android version-4.2), and MicroMax (Android version-4.4) Android Devices. The following test cases are used to test the application. The evaluation was done by almost 20 other users with their own Android Devices. Such wide compatibility of the application is made possible with the help of Android version 4 implementation that is being used by almost 91.9% (as of 04-26-2017 [15]) of android mobile devices.

5.1 Registration

The user needs to fill all the fields in the registration page to login to the application.

Positive Test Case

If all the fields in the registration form are filled, based on the three different login privileges available for the application, the following screens will appear according to the type of user. Figure 5.1(a), Figure 5.1(b) and Figure 5.1(c) show positive test cases.

![Figure 5.1(a) Positive case.](image1)

![Figure 5.1(b) Positive case.](image2)
Figure 5.1(c) Positive case.

Negative Test Case

When user doesn’t enter all the required fields in the registration page, a message will appear saying that “This field is required”. If a user tries to register second time with the same email id. A message will be prompted stating that “the email id already exists”. If user enters in the email id field an invalid email, then a message is prompted saying “This is not a valid email”. Figure 5.1(d) and Figure 5.1(e) show negative test cases.
5.1(e) Negative test case. 5.1(f) Negative test case.

Figure 5.1 Registration.

5.2 Login
User needs to enter email id and password to login to the application. Users can login using two ways. When a user gets registered for the first time, it will automatically redirect to the main home page. Secondly, if the user is using the application in the same device, they need not to enter the login credential again since the application has an option to save the credentials. Moreover, if they try to login using some other device, then they need to provide login credentials.

Positive Test Case
User needs to enter correct password and email id or should register for the first time giving all details correctly.
Negative Test Case

When user enters only one field and tries to login. A message will be prompted stating that “This field is required”. When user enters wrong credentials, a message will be prompted stating that “invalid credentials”. When a user with faculty privilege tries to login with student account, then a message will be shown stating “invalid credentials”. When faculty member enters wrong details in the web page, then a message will be shown stating “invalid credentials”. Figure 5.2(a), Figure 5.2(b) and Figure 5.2(c) show negative test cases.

Figure 5.2(a) Negative test case.

Figure 5.2(b) Negative test case.
Figure 5.2(c) Negative test case.

5.3 Reset Password

When a user wants to reset the password, then the user must enter a valid email id. Later, an OTP will be sent to the email id provided by the user, which in turn helps resetting the password. Figure 5.3(a), Figure 5.3(b) and Figure 5.3(c) show positive test cases.

Positive Test Case

Figure 5.3(a) Positive test case.  
Figure 5.3(b) Positive test case.
Negative Test Case

When user enters an invalid email id, then a message will be prompted stating the “invalid email”. Figure 5.3(d) shows negative test case.
5.4 Attendance Module

Positive Test Case

Figure 5.4(a) shows positive test case of reporting attendance.

![Success](image)

**Figure 5.4(a) Positive test case.**

Negative Test Case

When the user tries to capture image repeatedly, then a message will be prompted stating that “attendance for this date is already recorded”. When a user tries to send QR code through social networking services and other user tries to capture it, then a message will be prompted stating that “Invalid token”. Figure 5.4(b) and 5.4(c) shows negative test cases.
5.5 Other Miscellaneous Testing

Table 5.1 Registration testing.

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Bug</th>
<th>Checked</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email accepts only valid email type values</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Usernames cannot be reused</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Blank fields in registrations are not allowed</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Test Case</td>
<td>Bug</td>
<td>Checked</td>
</tr>
<tr>
<td>--------------------------------------------------------------------------</td>
<td>------</td>
<td>---------</td>
</tr>
<tr>
<td>Trying to register second time</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td><strong>Table 5.2 Login testing.</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blank Fields not allowed</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Login not accessed without internet access</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Email accepts only valid email type values</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Students cannot login through Faculty and Others account</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Faculty cannot login through Student and Others account</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Others cannot login through Student and Faculty account</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td><strong>Table 5.3 Attendance module testing.</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Works only when camera access is given</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>User cannot record attendance twice</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Works only when location access permission is given</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Location captured accurately</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Attendance saved only once for a date</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Test Case</td>
<td>Bug</td>
<td>Checked</td>
</tr>
<tr>
<td>---------------------------------------------------------------</td>
<td>-----</td>
<td>---------</td>
</tr>
<tr>
<td>Grades are visible only to that particular student</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Grades are assigned to only one course at a time</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Grades at Faculty-end are integrated properly with Student-end</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Faculty can edit the grades</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Grades are not higher than the maximum possible points</td>
<td></td>
<td>YES</td>
</tr>
</tbody>
</table>

Table 5.4 Grades module testing.
Table 5.5 University Health Care Services module testing.

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Bug</th>
<th>Checked</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appointments are successfully scheduled</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Timeslot spinning done</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>One appointment at a specified time</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Announcements are visible to faculty and students</td>
<td></td>
<td>YES</td>
</tr>
</tbody>
</table>

Table 5.6 Application Security testing.

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Bug</th>
<th>Checked</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passwords are encrypted</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Grades are encrypted</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Attendance records are encrypted</td>
<td></td>
<td>YES</td>
</tr>
</tbody>
</table>

Table 5.7 Forgot Password testing.

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Bug</th>
<th>Checked</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generates OTP with in minimal time</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Notifies if email id not found</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Notifies after updating new password</td>
<td></td>
<td>YES</td>
</tr>
</tbody>
</table>
5.6 Users Evaluations Report

Several students were contacted to receive feedback, on the usability of the application. Most students liked the Attendance module, also they found it user friendly and very helpful. Students were asked to scale the application between 1 and 10. Table 5.8 shows the evaluations.

**Table 5.8 Users Evaluations.**

<table>
<thead>
<tr>
<th>Questions</th>
<th>Average Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application User Interface</td>
<td>8.5</td>
</tr>
<tr>
<td>User friendliness</td>
<td>8.67</td>
</tr>
<tr>
<td>Attendance Module</td>
<td>9.67</td>
</tr>
<tr>
<td>Grades Module</td>
<td>8.5</td>
</tr>
<tr>
<td>University Health Care Services Module</td>
<td>8.5</td>
</tr>
<tr>
<td>Application Security</td>
<td>8.5</td>
</tr>
<tr>
<td>Password Reset Feature</td>
<td>9</td>
</tr>
<tr>
<td>Would you recommend this application</td>
<td>8.6</td>
</tr>
</tbody>
</table>

5.7 Limitations of this application

- This application is a prototype for few university services, which is strictly designed for Android users.
- In this application, Amazon Web Services, Tomcat, and all other services used are trial versions. The application may crash when number of requests increases.
• Amazon Web Services (trail version) can withhold up to 15,000 users, which is the biggest limitation of the application. This application is related to University services, Universities may have more than 15,000 students. This application may crash when the number of users exceeds a certain limit.

5.8 Integrating with the university software

The main advantage with this application is that, it uses Hibernate technology to connect with the back-end server. Hibernate has its own query language, “Hibernate query language” which is database independent [16]. This makes application to easily interact with any kind of database. Currently, this application is linked to Amazon Web Services. In future, to make this application work with real data back-end server link should be changed to the respective University server which can withheld vast number of users.

Currently, Texas A & M University - Corpus Christi has 12,175 [17] enrollments. As discussed earlier, this application is utilizing the services from Amazon Web Services (free version). This can withhold up to 15,000 users as discussed in Section 5.7. This application can be successfully used by the present enrollments. If the number of enrollments increases, then application may work abnormally or sometimes may crash.

5.9 Publish application in Google Play Store

This application can be published in google play store. The gradle version build for this application is “Android 4”, it works on all the Android versions above Android 4.0 (Jelly Bean) -Android 7.0 (Nougat).
As discussed in Sections 5.7 and 5.8, application can withhold up to 15,000 users. If the number of enrollments increases, then application may work abnormally or sometimes may crash.
6 CONCLUSION AND FUTURE WORK

In this project, a mobile based application is successfully developed that can serve as a platform for students, faculty members and university service providers to get an easy access to the attendance management system, grades management system and university healthcare services. The main motivation behind this work was to come up with an efficient access to the aforementioned services, provided the current web-based solutions are not efficient to run on handheld devices such as android based mobile phones and tablets.

This application provides three different functionalities namely, attendance management system, grades management system and university healthcare services. For attendance management system, user can record attendance, view attendance at various levels, view complete details about attendance, export the attendance file. Both students and faculty members have different type of privileges over the data. For grades management system, faculty members can assign, update and view grades and students can view their grades assigned. For university healthcare services, both faculty members and students can book appointments at healthcare service and view announcement posted by healthcare services.

Some of the future implementations that can be done are:

- Adding dining services, rooms booking services in the Others login.
- Developing this application for iOS users.
- Developing help section which helps users to navigate properly.
- Developing a module which helps user to upload their profile pictures etc.
BIBLIOGRAPHY AND REFERENCES


