Computer Science and Engineering

©

Project Design-Version 1.0

- Document Number: Project Specification-001

- Team Members

Names	E-mails
Saudamini Shrivastava	ss7679@nyu.edu
Simeng Sun	ss7151@nyu.edu
Shantanu Ranjan	sr3306@students.poly.edu

Revision History

Date	Version	Description of Change
3/23/2014	1.0	Initial Design Document

Table of Contents

1. Problem Overview

The world is technology-savvy. Every person, however hardworking, secretly wishes that there is a technology that does the work for him. If not the entire work, at least some part of it. This does not mean that the mankind is edging towards lethargy. The reason behind the creation of any kind of technology is to lessen the workload of an individual.

This app aims to lessen the work of individuals, be it professor or a student by taking attendance that could otherwise be tedious and laborious if done manually.

1.1 Introduction

TickMark is an attendance collector android mobile application that is useful mainly for academic purposes. It allows collection of attendance in an efficient, quick and easy manner.

Traditionally, attendance is taken either by passing a sheet of paper around the class or calling out names and checking them off the list. All attendees have to either sign their names or wait for their names to be called out. This can be very tedious, even time consuming. The time allotted for learning is spent in marking attendance, thereby decreasing efficiency.

Moreover, managing so many attendances for each day and each lecture is also difficult. It is almost cruel to ask a person to manage so many records and keep track of so many files in this age of technology.

TickMark, reduces effort, both, in terms of time and keeping track of attendances. It generates a one-time code for the attendees to mark attendance. This way all that the attendees have to do is start TickMark on their respective mobile phones, select the class name and mark attendance. The professor too, has to just generate the one-time code each time he wants to take attendance and the result is managed by the app.

2. Current Solutions and Related Work

As expected, the android market is filled with apps that are capable of collecting attendance. But that is all that they are capable of- 'collecting attendance'.

When it comes to managing the attendance collected, the efforts of the user are not reduced. For example, *Attendance Roster* lets you keep track of students' attendance but you are required to import the student list from your computer.

Other apps such as *Attendance Tracker* provide you with an interface wherein you have to manually put a check against names that are present and a cross against names that are absent.

But none of these apps provide the functionality of verifying if the attendee is indeed present in the class. Also, all other apps can just do the attendance work. What if the leader wants to do a quick poll? They would have to turn to another app or do it with paper work. TickMark provides a feature that you can take a small quick poll, and the procedure is similar to the taking attendance. The result is quick and clear. This will save a lot of time for both the attendee and the leader.

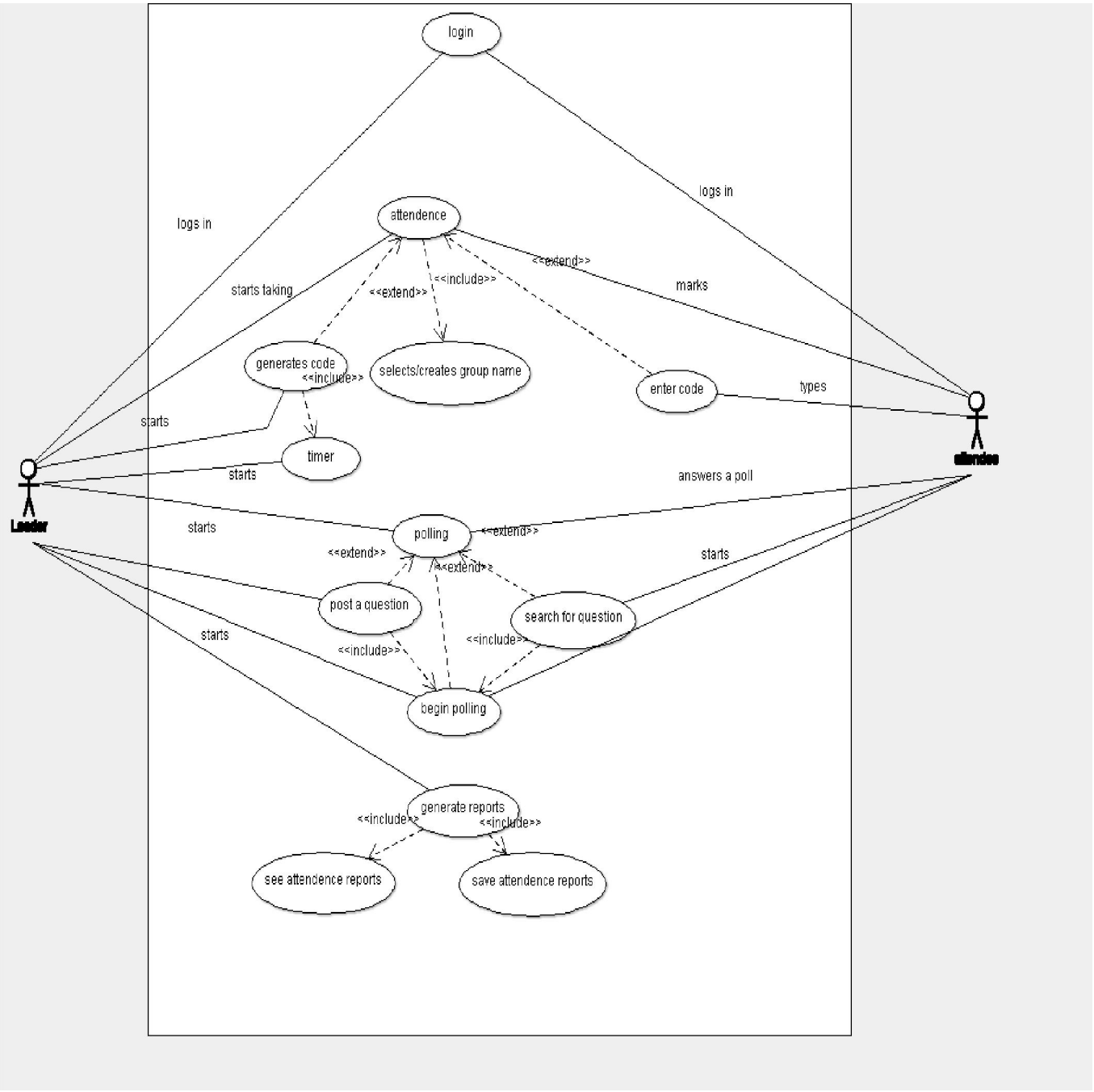
3. Solution

The want for an android application that can take attendance while being easy to manage and also efficient led to the idea of TickMark. This mobile application is different from its contemporaries in terms of successfully identifying if an individual should be allowed to mark attendance based on his location.

The following subsections elaborately describe the application's working with the help of various model diagrams.

3.1 Use Case Design

The following use case diagram describes the various activities the two main users of the application perform.



3.2 MVC Framework

This subsection describes the various components: models, views and controllers of the application.

3.2.1 Models:

Leader, Attendee, Group

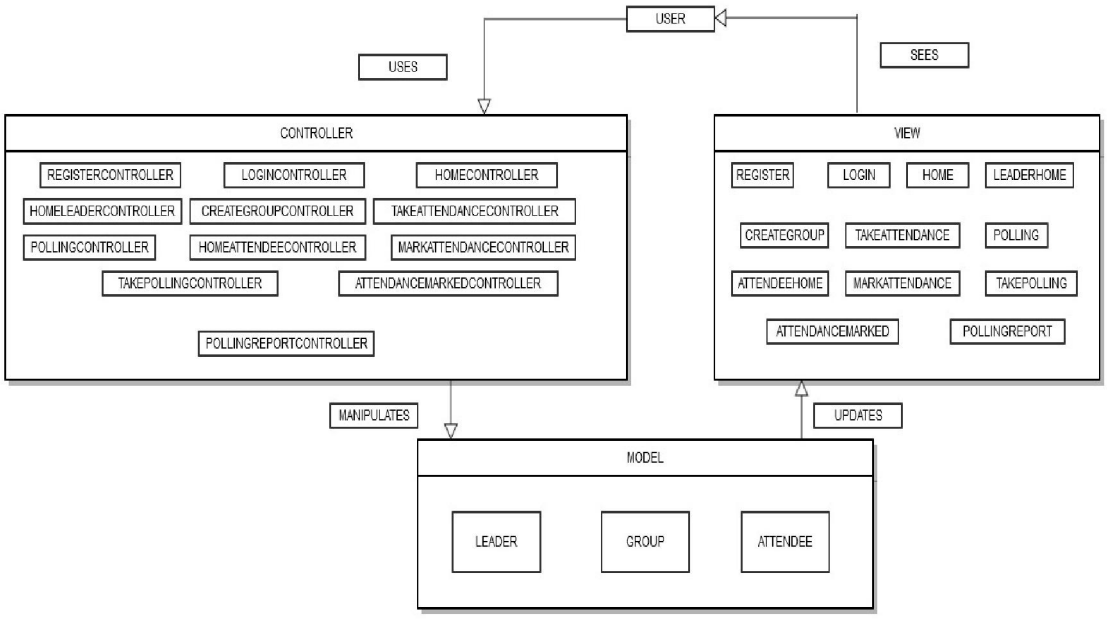
3.2.2 Views:

Register, Login, Home
Leader: Home, Create Group, Take Attendance, Polling
Attendee: Home, MarkAttendance, TakePolling
Report: AttendanceMarked, PollingReport

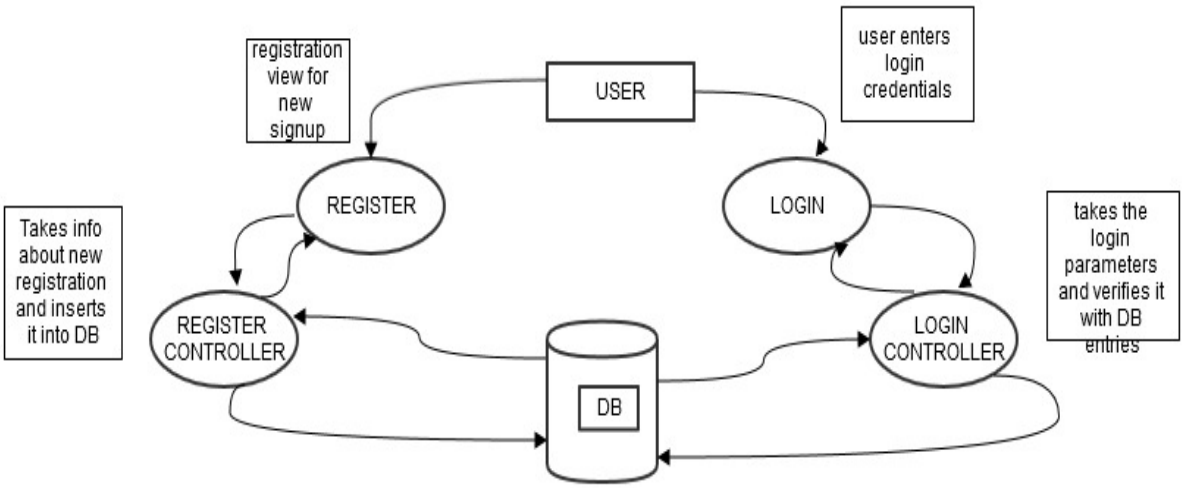
3.2.3 Controllers:

RegisterController,LoginController,HomeController
Leader:HomeLeaderController,CreateGroupController,TakeAttendanceController,Polling Controller.
Attendee: HomeAttendeeController,MarkAttendanceController,TakePollingController
Report:AttendanceMarkedController,PollingReportController.

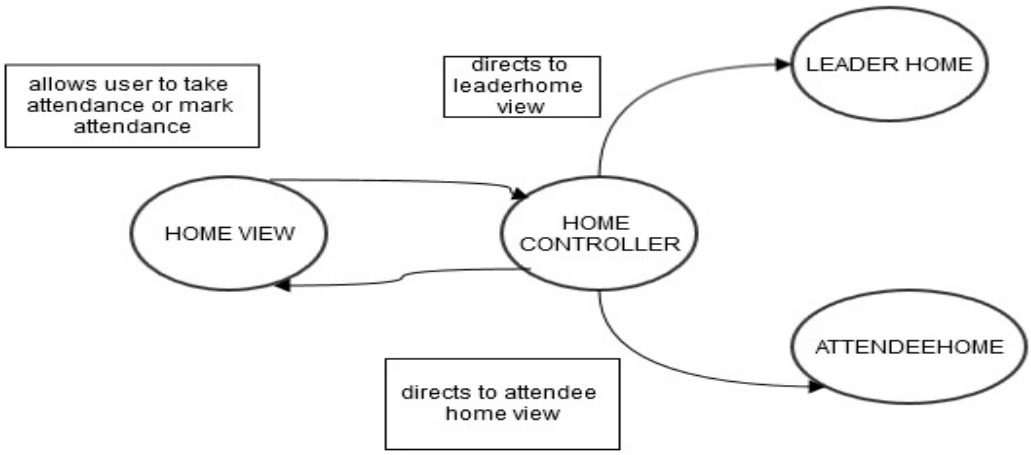
3.2.4 System component Architecture Design



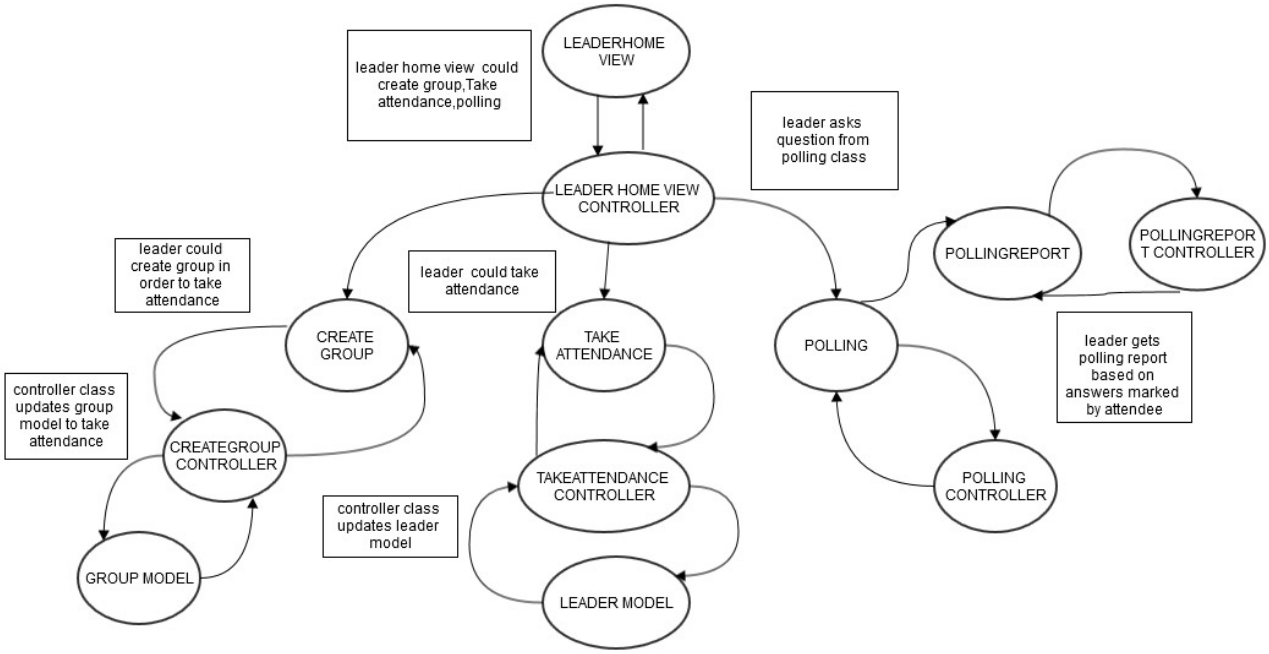
Login and Registration View



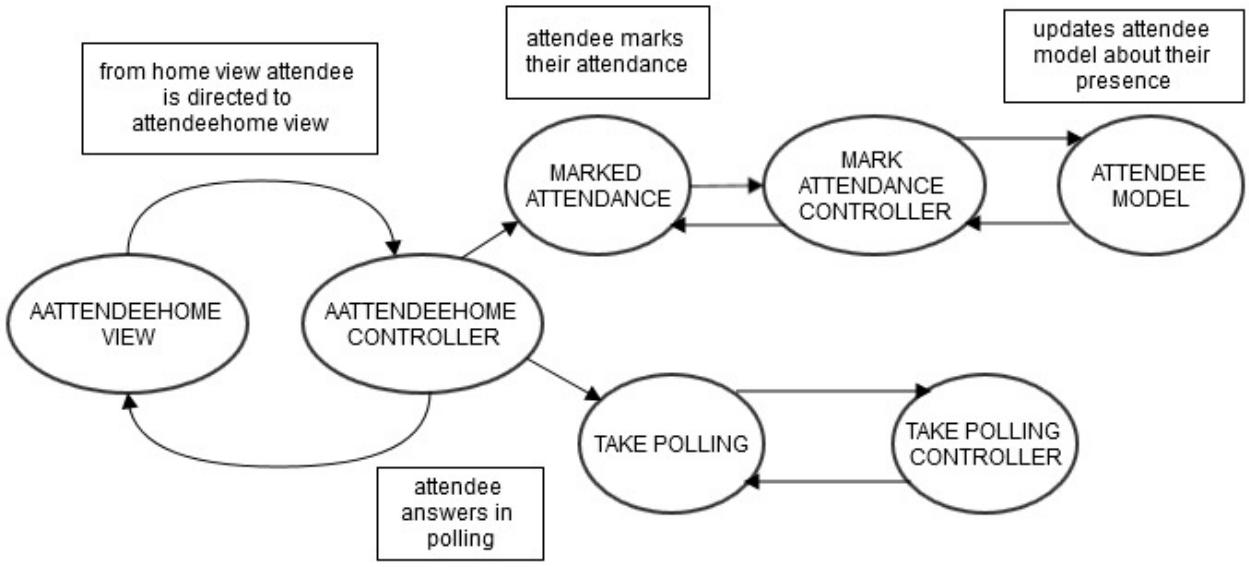
Home View



Leader View

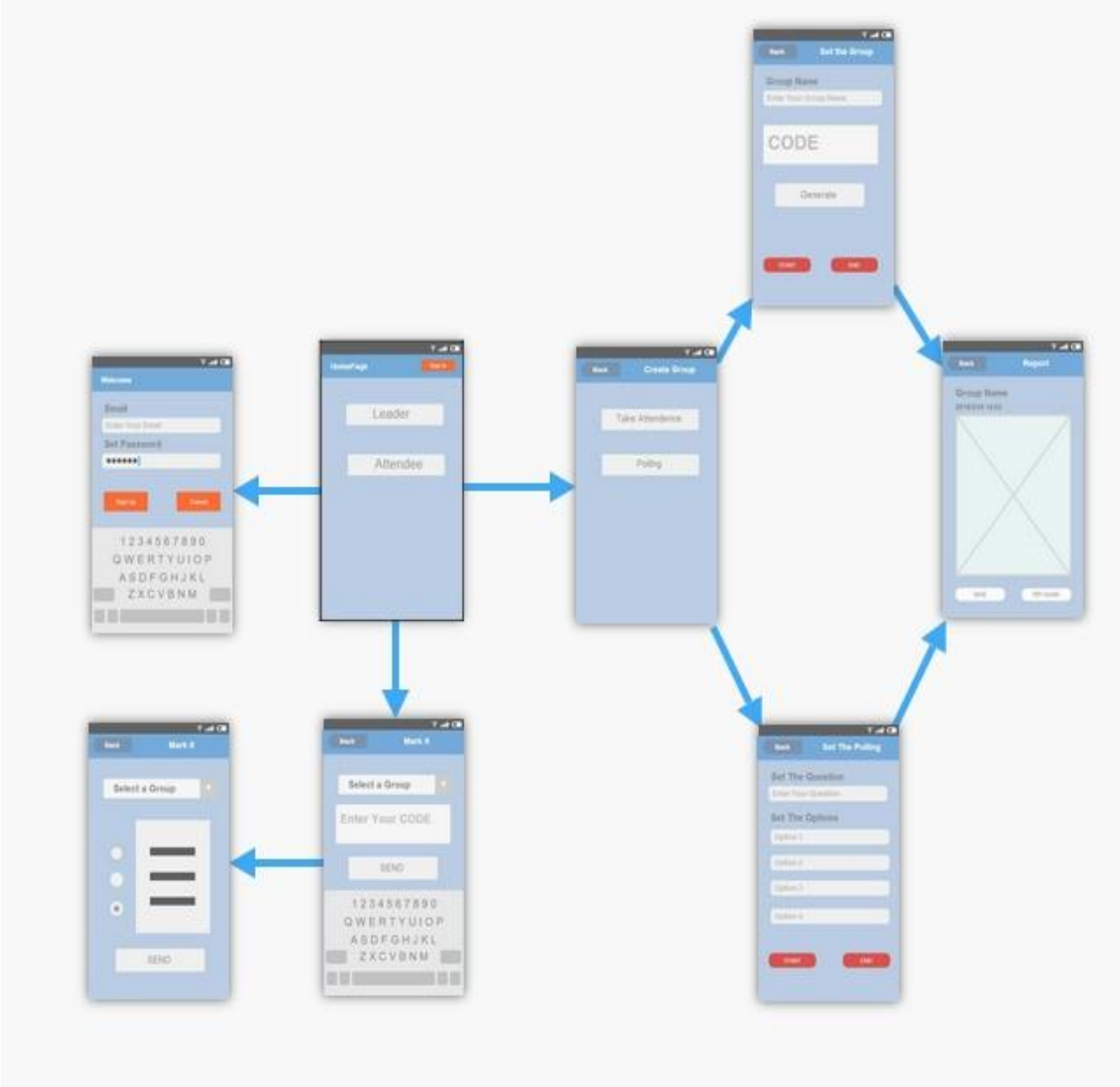


Attendee View

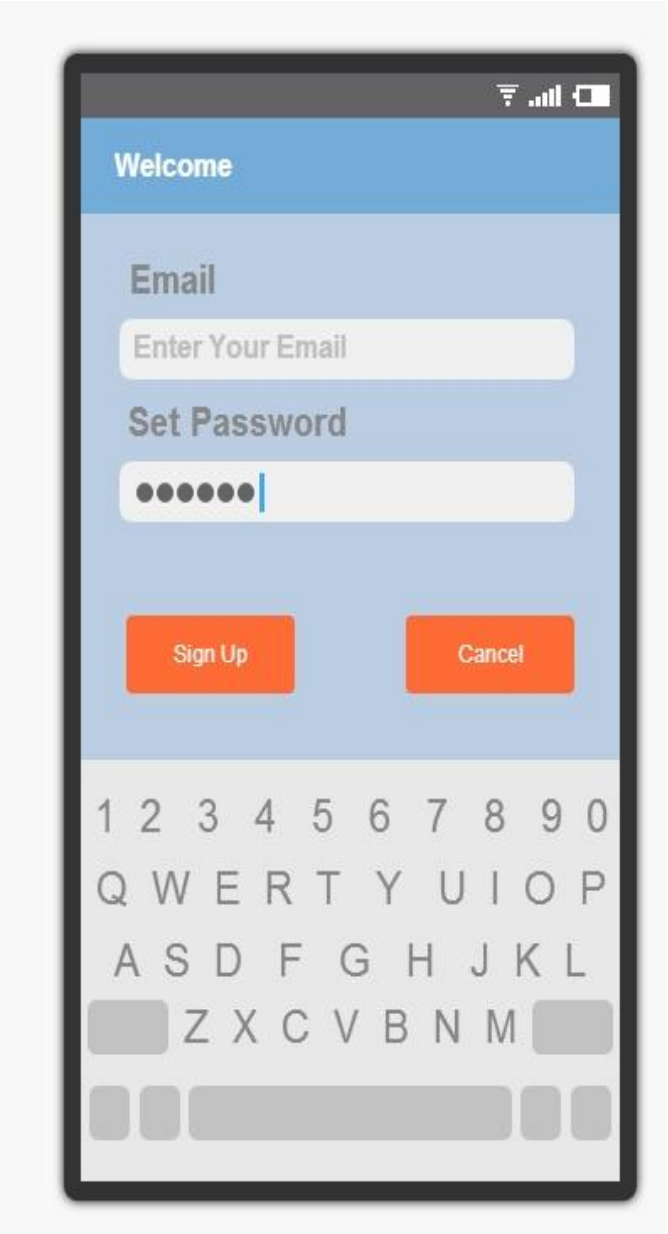


3.3 UI Design

The flow and control of data is detailed in the flowchart below:

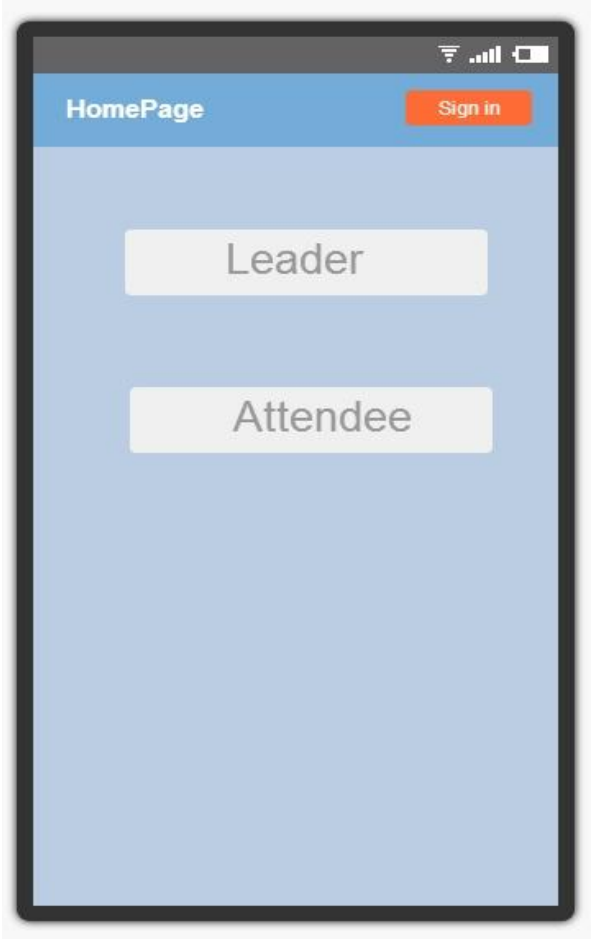


Sign in and Log in Page:



- Sign-Up if new Registration.
- Login with registered credentials.
- Verifies the login credentials by checking it in database.

The Home Page:



- After successful login directed to Home page of user.
- Two GUI buttons marked as Leader or Attendee.
- User could click on Leader to take attendance or could click on Attendee to mark attendance.

From here the user could take either leader path or attendee path.

If the user clicks on Leader button then he is directed to leader page where he could create his group and could take attendance. The flow of leader is described below:

The Leader flow: Create Group Page:

- The user could either click on Create Group to register his own group to take attendance. This is one time creation.The group is uniquely identified by the users registered id.

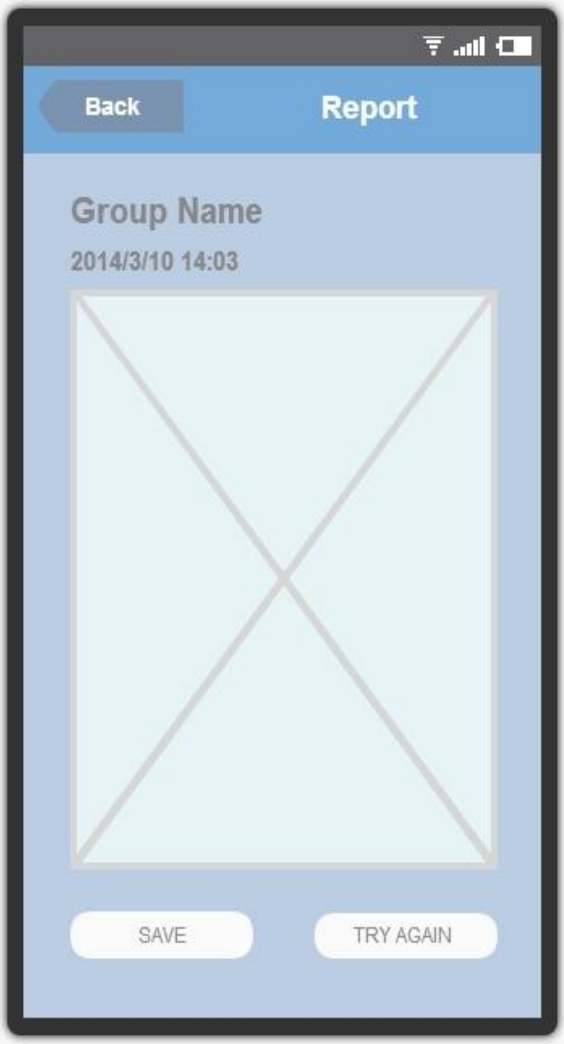
- The leader could click on Take Attendance button to generate unique Id so tha presence could be marked.

- The user could click on Polling button to get feedbacks/questions on the lecture or presentation.

If Leader chooses Take Attendance button then he will be directed to this page.

- Group name that is registered by the leader could be shown in the form of drop down menu or could manually enter the code.
- If the leader clicks on Generate button then a unique alphanumeric code of size 4 would be generated.
- When the leader clicks on start button then a timer would start and attendee must mark their presence for the given time.

The Report Page:



- Once the attendance has been marked by the students/attendee then a report could be generated in a file format and the leader would be allowed to save it on his device or in database.

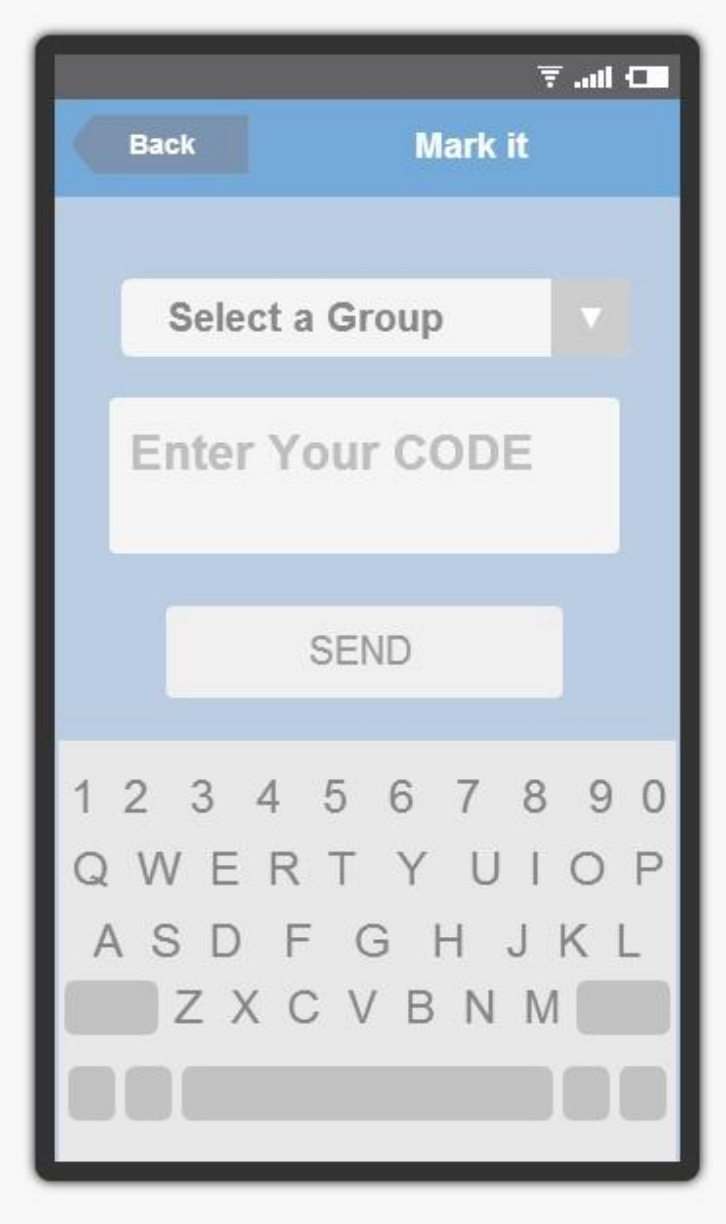
If the leader chooses Polling then he could either take feedback or could ask questions based on the lecture/presentation.

- The leader could set the question and options for it.
- Once it is done then could click on start button so that it will appear on attendee screen to mark their answers or provide feedback.

The Report Page remains the same and the image would be pie chart:

- Once all the students/attendee registered to a particular group has marked their answers then a report would be generated in the form of pie chart/bar graph.

The Attendee flow diagram is shown below:



- When the attendee clicks on attendee button ,he is directed to this page to mark his presence.
- The attendee could select the group for which he wants to mark his presence from the dropdown menu
- He would then be asked to enter the unique code generated by the leader.
- By clicking on send button presence could be marked.

The Take Polling Page:

- This page is used by attendee to answer the questions based on the presentation/lecture.
- The answers marked would confidential and won't reveal the identity of the attendee.

3.4. Android Components

The application will use the following android components:

1. Wifi - Internet: In order to login and take/mark attendance, the app will require connection to Internet or wifi.
2. GPS: The app requires location based services of a smartphone.
3. Amazon web service: The app requires a backend web server and android web services provide a free web server.
4. Amazon relational database web service: The app requires a backend database server that can be acquired from Amazon relation database web service.
5. SQLite: Not all data would be sent to the server. Local data would be stored using android's local database.

6. Project Member Breakdown

There are three project members. Each of them will be contributing equally towards the development of the application. Each member has been assigned few tasks.

1. Saudamini Shrivastava
    - Attendee UI development
    - Activity flow for Attendee
    - Leader code generation development
    - Database Setup
    - Signup page UI development

2. Simeng Sun
    - Leader Polling UI development
    - Activity flow for Leader Polling
    - Leader code generation development
    - Web Server Setup
    - Create group UI development

3. Shantanu Ranjan
    - Leader Report analysis UI development
    - Activity flow for Leader report analysis
    - Leader code generation development
    - Integration Testing
    - Choose role, i.e Leader or Attendee page UI development

5. Milestones

Date	Task completed
3/23/2014	Initial Design Document Due
4/4/2014	• Completion of UI and all activities • Completion of flow of application

Week 1	3/28/2014	Creating activities for all models
Week 2	4/4/2014	Connecting all activities in a basic flow

4/7/2014	Initial prototype presentation
4/28/2014	Presentation at AT&T

Week 1	4/12/2014	Web server connection setup
Week 2	4/19/2014	Database connection setup
Week 3	4/26/2014	Completion of applicaton

5/12/2014	Final Demo

Week 1	5/12/2014	Implementation of new ideas and Testing

6. Bibliography

1. https://play.google.com/store/apps/details?id=com.flapsapps.roster
2. https://play.google.com/store/apps/details?id=peterman.apps.attendance