

Development of Mobile Learning System Based On Android App



**Department of Computer Science and Engineering
School of Engineering and Computer Science
BRAC University**

Supervisor

Dr. Amitabha Chakrabarty

Sama Samrin

13101125

CSE

Lazina Aftab

17141028

CS

Declaration

This is the thesis Submission to the Department of Computer Science and Engineering, BRAC University, Dhaka, submitted by the authors for the purpose of obtaining the degree of Bachelor of Science in Computer Science, and the degree of Bachelor of Engineering in Computer Science and Engineering. We hereby announce that the results of this thesis are entirely based on our research and surveys. Resources taken from any research conducted by other researchers are mentioned through references. This thesis either in whole or in part, has not been previously submitted for any degree.

Supervisor

Dr. Amitabha Chakrabarty
Assistant Professor
Department of Computer Science and Engineering
BRAC University
amitabha@bracu.ac.bd

Authors

Lazina Aftab
ID: 17141028
lazina.aftab@gmail.com

Sama Samrin
ID: 13101125
samasamrin@gmail.com

ABSTRACT

In the modern world of mobile technology and their increasing availability and affordability, mobile devices particularly Android platforms play an important role in the field of communication, entertainment and learning. It has greatly changed the functioning of the world, even the learning process. Learning through mobile phones, abbreviated as Mobile Learning has become part of the educative system. Mobile Learning or M-Learning is a specific type of e-learning that provides educational materials through mobile phones. The purpose of this study is to access the learning materials, interact with faculties and view other related documentations along with other facilities, by using android based mobile phone inside and outside the classroom. This will also determine the users' prior knowledge on the personal and educational use and application of mobile phones. Our main aim behind this whole project is to exercise a proper way to bring forth all the necessary elements a student or a teacher requires to carry out his job efficiently. Our app named EdEasy will help the users gain knowledge about their past records, with which they can improve their way of studying and thus make the whole competition a lot healthier and much more inspiring. Through this app, they can also access all the publicly available academic resources like books from any department. As a result, EdEasy will ensure the reach of education to every part of the institution regardless of any barriers.

Acknowledgement

The idea of this thesis project was suggested by our Supervisor - Dr. Amitabha Chakrabarty, Assistant Professor of the School of Computer Science and Engineering of BRAC University. He considered the two of us – Lazina Aftab and Sama Samrin, being worthy of carrying out this thesis successfully. Both of the group members have contributed their level best for this thesis which has resulted into this work we are presenting to you today.

We would like to express our heartiest thanks and utmost gratitude to our supervisor, Dr. Amitabha Chakrabarty, Assistant Professor of the School of Computer Science and Engineering of BRAC University for providing his valuable perception, suggestion and guidance at our thesis progress. A continuous supervision and the spirit to push us more towards developing a good thesis project was received from him, without which this thesis could not be successful. We are honored that he allowed us to work under his supervision.

We would also be very grateful to Annajiat Alim Rasel Sir. He helped us to get allocated Lab room to do our research work.

Furthermore, we would like to thank Muhammad Abdur Rahman Adnan sir. He helped us to get over the baffling loop of code logistics and therefore solve the critical problems we were facing at the time.

We would like to remember the help and support of our friend Argha Kamal Roy in the initial phases of the app development. Though he passed away at a tragic incident, we have remembered him throughout our thesis journey. Besides, a gracious thanks to one of our senior brothers S. M.

Mohi-Uz Sunnat for his utmost help and continuous inspiration to complete this thesis. We would like to say thanks for your honest support.

We convey gratitude from the depth of our hearts towards the faculty members of Computer Science and Engineering department for providing us with all the knowledge throughout our undergraduate life so that we could do this thesis. Finally, we would like to thank our parents who have supported us throughout, no matter what. Without their support we could not have come this far. They helped us in more ways than we can count. They encouraged us to continue our research no matter how indecisive we became of our own capabilities. This work would not see the light of the day without their collective help.

Table of Contents

| | |
|------------------------------|-----|
| Abstract | III |
| Acknowledgement | IV |

Chapter I : Introduction

| | | |
|-----|---------------------|---|
| 1.1 | Introduction | 1 |
| 1.2 | Objective | 1 |
| 1.3 | Motivation..... | 4 |
| 1.4 | Thesis Outline..... | 5 |

Chapter II : Literature Review

| | | |
|-------|-----------------------------|----|
| 2.1 | Background Researches | 7 |
| 2.1.1 | Online Survey | 7 |
| 2.1.2 | Why Android..... | 13 |

Chapter III : Contribution

| | | |
|-------|--------------------------------|----|
| 3.1 | Quick View of the Project..... | 15 |
| 3.1.1 | Languages | 15 |
| 3.1.2 | Tools and Server..... | 15 |
| 3.1.3 | Libraries..... | 16 |

| | | |
|-------|-------------------------------|----|
| 3.1.4 | Permissions..... | 18 |
| 3.2 | Algorithm..... | 20 |
| 3.3 | Features..... | 23 |
| 3.3.1 | Logic and Implementation..... | 25 |

Chapter IV : Conclusion

| | | |
|-----|--------------------|----|
| 4.1 | Limitations | 50 |
| 4.2 | Future Works | 51 |
| 4.3 | References | 51 |

List of Figures

| | | |
|---------------------------|--|----|
| Figure2.1.1(a) | Pie Chart of Students Ratio | 8 |
| Figure2.1.1(b) | Pie Chart of App Usage for Institutional Courses | 8 |
| Figure2.1.1(c) | Pie Chart of M-Learning App Users | 9 |
| Figure2.1.1(d) | Pie Chart of Smartphone Users | 9 |
| Figure2.1.1(e) | Pie Chart of Mobile OS Users | 10 |
| Figure2.1.1(f) | Pie Chart of Users Interested in M-learning Apps | 10 |
| Figure2.1.1(g) | Pie Chart of Institutions Interested in M-learning Apps | 11 |
| Figure2.1.1(h) | Pie Chart of Nationality of Users | 11 |
| Figure2.1.1(i) | Bar Chart of Most Requested Features | 12 |
| Figure2.1.2(a) | Usage of cellphones based on Android OS in Dhaka City | 13 |
| Figure2.1.2(b) | Android phone users in Bangladesh from January ‘14 to June ‘17 | 14 |
| Figure3.1.3 | Imported Libraries in Gradle File | 16 |
| Figure3.1.4 | Permissions of this app in Manifest File | 18 |
| Figure3.3.1.a(i) | Users Database Overview | 26 |
| Figure3.3.1.a(ii) | Selection of Database Branch Based on User Role | 26 |
| Figure3.3.1.a(iii) | Dashboard Overview | 27 |
| Figure3.3.1.b(i) | Database Structure of Assigned Courses | 28 |
| Figure3.3.1.b(ii) | Extraction of Names and Sections of Assigned Courses from Database | 29 |
| Figure3.3.1.b(iii) | Drawer Menu | 29 |
| Figure3.3.1.c(i) | Sorting Process in Database | 30 |

| | | |
|---------------------------|--|----|
| Figure3.3.1.c(ii) | Extraction of Previous Semester Result from Database | 31 |
| Figure3.3.1.c(iii) | Result view | 32 |
| Figure3.3.1.d(i) | Database Structure of Calendar | 33 |
| Figure3.3.1.d(ii) | Extraction of Events from Database | 34 |
| Figure3.3.1.d(iii) | Calendar Displaying Course Events | 35 |
| Figure3.3.1.e(i) | Database Structure of Chatroom | 36 |
| Figure3.3.1.e(ii) | Chatroom View | 37 |
| Figure3.3.1.e(iii) | Extraction of Chatroom Messages from Database | 38 |
| Figure3.3.1.f | Marksheet View | 39 |
| Figure3.3.1.g(i) | Day Based Routine View | 40 |
| Figure3.3.1.g (ii) | Full Week's Routine View | 41 |
| Figure3.3.1.h(i) | Database Structure of Course Materials | 42 |
| Figure3.3.1.h(ii) | View of Course Materials in Storage | 43 |
| Figure3.3.1.h(iii) | Course Materials View and Download | 44 |
| Figure3.3.1.i(i) | Account Profile | 45 |
| Figure3.3.1.i(ii) | Algorithm of Milestone Ratio and Progress | 46 |
| Figure3.3.1.j | Display of Push Notifications | 47 |
| Figure3.3.1.k | Online Library of CSE Department | 48 |
| Figure3.3.1.l | Results Graph | 49 |

Chapter 1

INTRODUCTION

In introduction we will discuss about our objectives of building this app in details and the motivation behind doing so.

1.1 Introduction

To keep up with the increasing demand of having everything necessary by our side at all times, we are developing an app that ensures availability of essential academic materials and also proper time management for both teachers and students of an institution. This android-based app will include all the necessary information of the user at one place and notify them through a push notification whenever a new file is added to their course folder. Combining these, the authorities of a department can ensure proper management of its members and materials with the help of this app.

1.2 Objectives

1. Centralized learning

To ensure the proper management of all course materials, updates and other necessary aspects of the overall learning process, this app will offer a solution to supervise over all these facets from one centralized system. This system will hold a centralized database, files storage, authentication process and more to make sure that all the entities involved get top-notch services.

2. Simplified learning process

Often we cannot process all the information and references introduced by the faculties in a classroom. Also, many times while studying, we stumble upon some information either from the vast world of internet or from the assigned textbooks, which we need to discuss with peers and the assigned faculty of that course. In case we are not acquainted with any of the members of the course section, we find ourselves in the face of some easily avoidable problems. So, we have introduced the chatroom feature in this app to let all the members of a course section, including the faculty member to be a part of the discussion. These will be chatrooms for strictly formal conversations and exchange of informative links. Besides, the course materials will always be just a touch away as explained in the next point.

3. Amplified Availability

No matter where you are at the moment, you will never find yourself deprived of the official course materials as provided by the faculty. They will go with you anywhere and everywhere. Even if you need to access a file after midnight, you can do that without asking a peer for help. Also, these materials get updated automatically. So, you do not even need to press any Refresh button repeatedly to access the latest files.

4. Interactive environment

One of the best features of today's smartphones and apps is the exceptionally interactive mobile environment and user interface. We took advantage of this by utilizing widget features from the Material theme of android for the design implementation part. Because of this theme, the

elements of this app will go through smooth transitions when interact with it by pressing a button or swiping through activities.

5. Easily accessible

For using this app, you do not have to go through any complicated process of official registration. If your institution decides to adopt this system into their established structure, their database of department courses, students and faculties will automatically be imposed on this app's database. From thereon, all you need to do is collect your pre-set email and password from the respective authorities and sign in to the app. An authentication entry will be created for you automatically through Firebase, along with a unique user id. Then based on your branch in the database tree you can access your previous results, current class routine, current mark sheets, along with all course materials of the courses you are assigned to.

6. Reduced cost

Many of us do not reside at a place in the close proximity of the institution we are studying at. Thus, when we need certain information from the official files and folders, we have to find ourselves a means of transportation just to access that file or folder. Thus, finding all those information at the palm of your hands through this app will greatly reduce the total cost of your daily trips to the institution. A mere connection to your local wifi can guarantee the availability of all those data on your personal device in a matter of milliseconds.

7. Saving time

This app brings all the relevant information of your courses, records and more to your mobile devices saving a lot of precious time. You do not have to wait to reach the establishment

or wait for someone to send you the files to access all those data. You can do more with your time because of this app.

8. Realtime updates

The app will be synchronized with the database and the file storage at all times. Hence, whenever you open or operate the app, you will be getting all the updated data and materials in real time.

1.3 Motivation

In the modern education system, we students do not always get the necessary study materials from one place, nor can we always communicate with the faculty members to discuss and solve our problems. This is where EdEasy comes in. Through this app, the faculties and the students will have no problem sharing the study contents at any time and from anywhere. Also, if the faculty wants to update a certain file, he can do so without going through the hassle of waiting for the next class and letting the students know personally about this update. The moment he changes anything among the contents, that same change will happen to all students' contents once they connect their phone to the internet. Moreover, in a class a student cannot always find familiar faces. EdEasy can also help them in this regard. By using the section-wise chatrooms of our application, any student can communicate with his classmates and start a discussion about any problem he is facing. The admin will be monitoring these discussions to avoid exchange of cheat sheets and use of obscene language. Furthermore, students always do not get the contact information of a teacher. Through

our app, any student enrolled in a particular section of a particular course, can directly start a conversation with the said faculty for any query. Finally, students often forget about their submission deadlines and exam dates and as a result fail to get their desired grade. To solve this problem we include alarms in our app to remind them of these events, which they will have the option to turn on or off depending on their requirement. Thus, our app will help to solve the usual daily problems faced by a student.

1.4 Thesis Outline

Chapter 1 is the introduction of our thesis covering the motivation behind it and also the objectives we have aimed for through this project.

Chapter 2 deals with literature review include background research and reasons we choose android.

Chapter 3 is comprised of our Contribution. Here all the features is given including the logic and the implementation. We explained about the libraries which we have used and some algorithms.

Chapter 4 is Conclusion where we added the limitations of our work, the future plan and reference.

Chapter 2

LITERATURE REVIEW

Over the past few years, several organizations have taken the initiative to implement m-learning. Some of these apps are Mobile Learn™, OppiaMobile Learning, Google Classroom, etc. Mobile Learn™ by Blackboard Inc. ^[2] makes it easier for students to keep up with their courses by letting them access the materials whenever and wherever they want. But it can only be used for free if the institution of the student licenses it, otherwise it has to be purchased at \$1.99^[1]. OppiaMobile is also a similar mobile learning application developed by Digital Campus ^[3], which offers to run content exported from Moodle even when the student is offline. But these apps cannot guarantee that the content created by instructors or other users will always be compatible. For Blackboard's app, if there is no access from institution, features and functionality may at times be limited if the institution has not updated the appropriate software or experiences a server outage. This app received good reviews initially but recently its users have been experiencing problems like, not getting notifications properly, inconvenient user experiences, no option to remove profile picture etc. Users of OppiaMobile have also expressed dissatisfaction for several problems like: app not running at all, failure to upload materials by faculties etc. So, they all offer similar features like ours but have limitations like:

- Inability to run on all mobile phones
- Failure to upload materials

- Frequent freezes and crashes

Also, they don't display the student's record on his/her homepage like in our app. Our app is expected to solve all these problems and deliver a smooth user experience as well as proper connection and communication between user groups.

2.1 Background Researches

Before Started our thesis work first of all, we needed a statistics of the current situation regarding educational system depends on different applications.

2.1.1 Online Survey

In our country or outside our country many people want mobile based learning app that will be easily accessible. Keeping this in mind, we made a google doc form regarding this M-learning application with some important questions and share this survey through Facebook/google. More than 100 people filled up this survey.

Here are the Questions of the Survey and the Pie Charts of Results:

1. Are you a student?

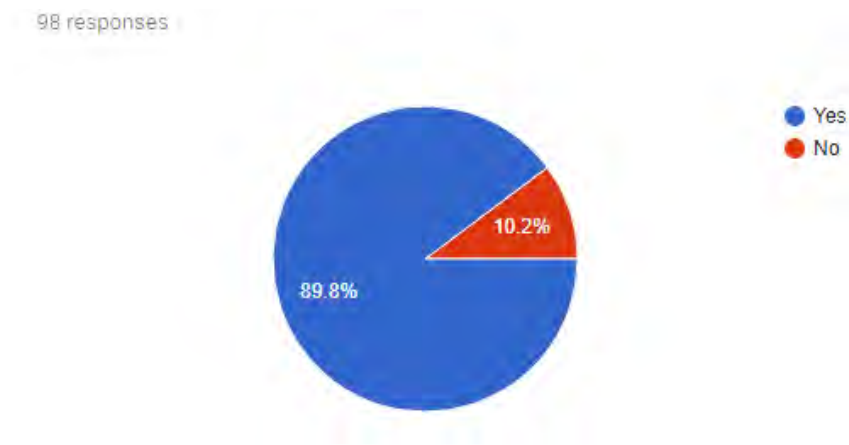


Fig 2.1.1 (a): Pie Chart of Students Ratio

2. Do you use any app to manage your institutional courses (of your university or school)?

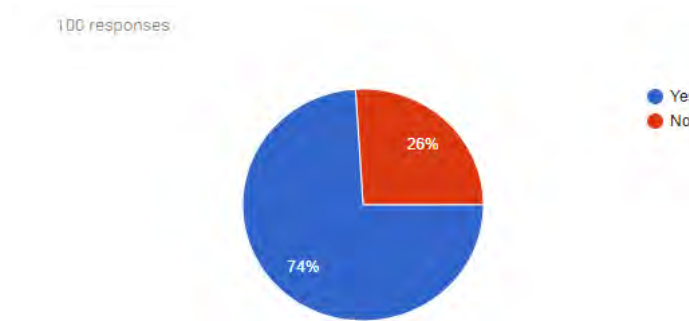


Fig 2.1.1 (b): Pie Chart of App Usage for Institutional Courses

3. Which of these apps do you use currently?

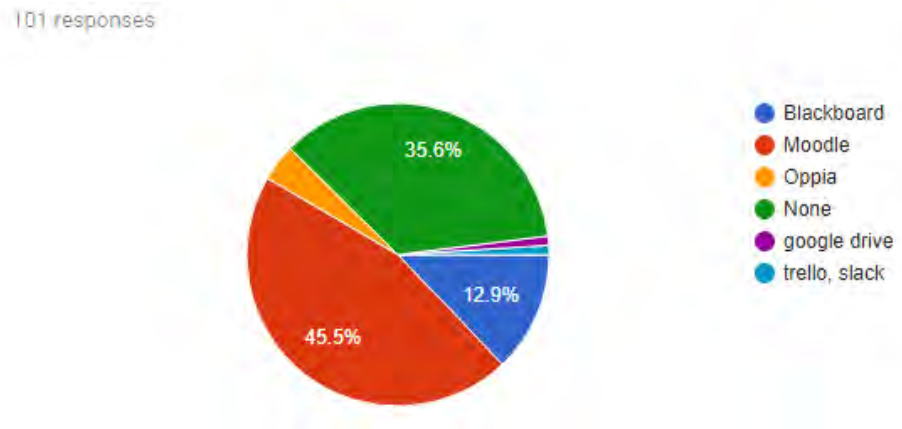


Fig 2.1.1 (c): Pie Chart of M-Learning App Users

4. Do you use smartphone?

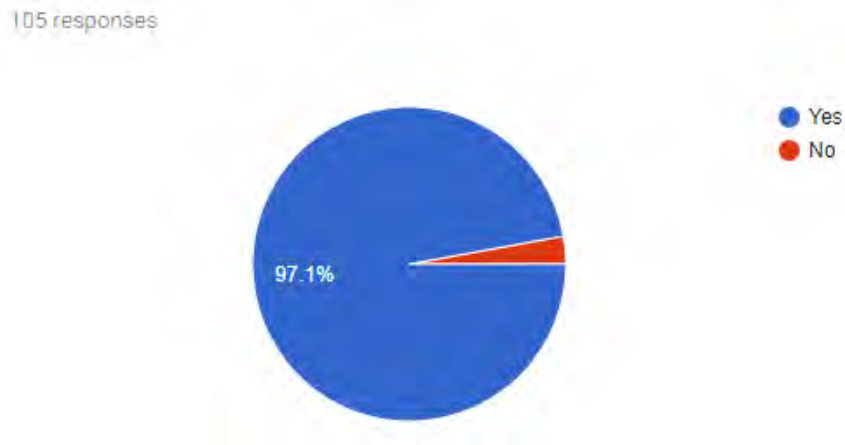


Fig 2.1.1 (d): Pie Chart of Smartphone Users

5. Which of these platforms do you prefer while using this app?

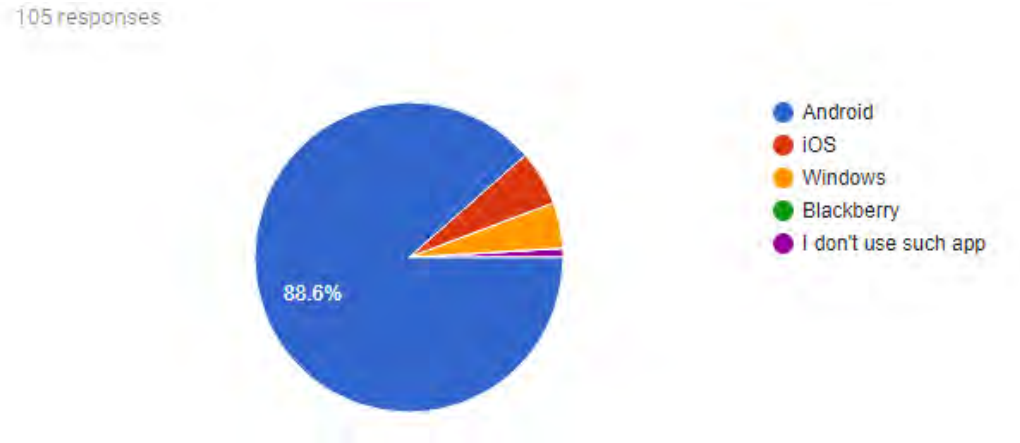


Fig 2.1.1 (e): Pie Chart of Mobile OS Users

6. Do you feel the need of a mobile learning Application?

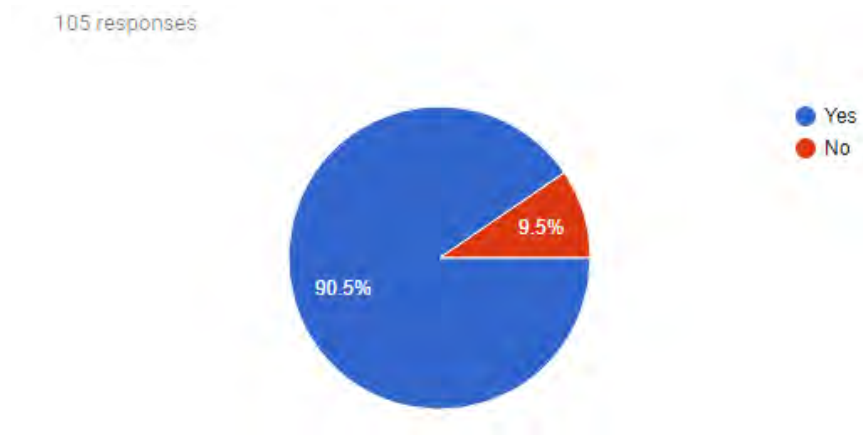


Fig 2.1.1 (f): Pie Chart of Users Interested in M-learning Apps

7. Does your institution encourage the availability of course materials on smartphones and other portable devices?

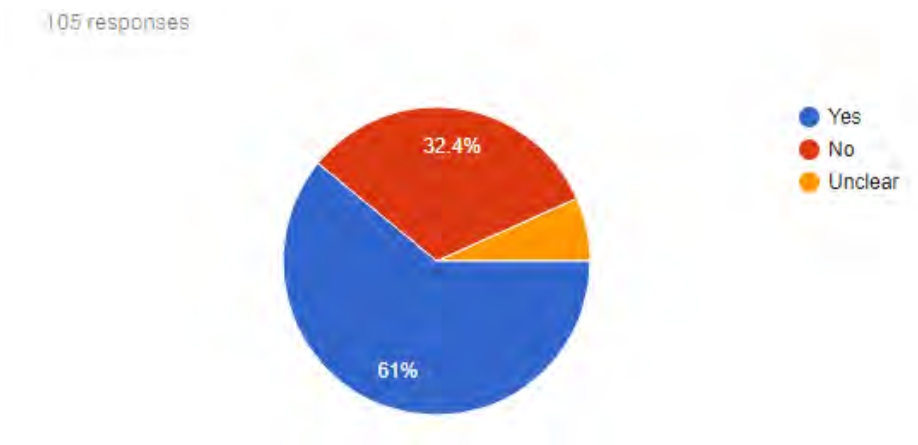


Fig 2.1.1 (g): Pie Chart of Institutions Interested in M-learning Apps

8. Which country are you from?



Fig 2.1.1 (h): Pie Chart of Nationality of Users

9. Select the features you would like to see in the perfect m-learning app

- Results of previous semesters
- Calendar showing important Events
- Course section-based chatrooms
- Class Routine
- Push notifications
- Marksheets of current courses
- Course Materials
- Your profile
- Other:

102 responses

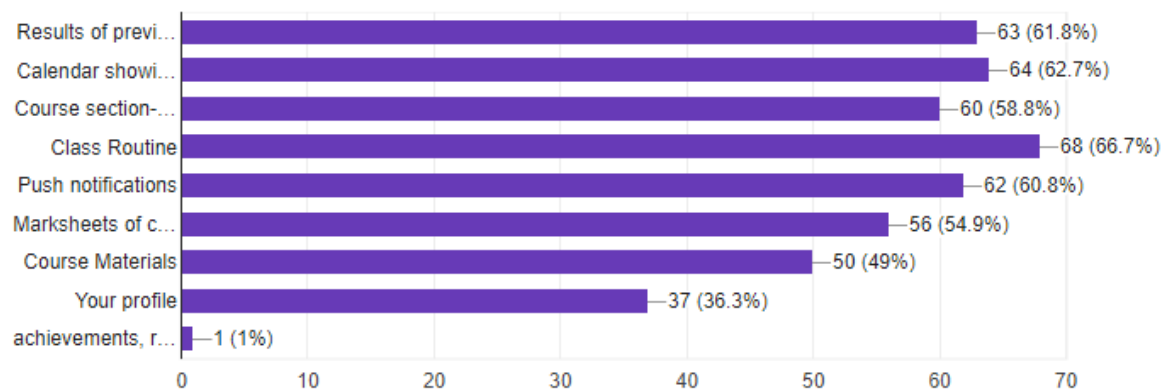


Fig 2.1.1 (i): Bar Chart of Most Requested Features

After this survey we became really pleased to see that people are really curious about the M-learning application and our project will play a vital role in this country as well as outside the country for these people.

2.1.2 Why Android

In Bangladesh, Android, Apple and Windows Phone are the main Operating System for mobile phones. But Android is the main OS^[1] in our country. More than 50% mobile users use Android. Fig 1 shows the pie chart of the Android users in Dhaka city. So, this platform is the best way to reach the users. Besides, these platforms devices are so available and start from very cheap budget. As user experiences, android is very user friendly.

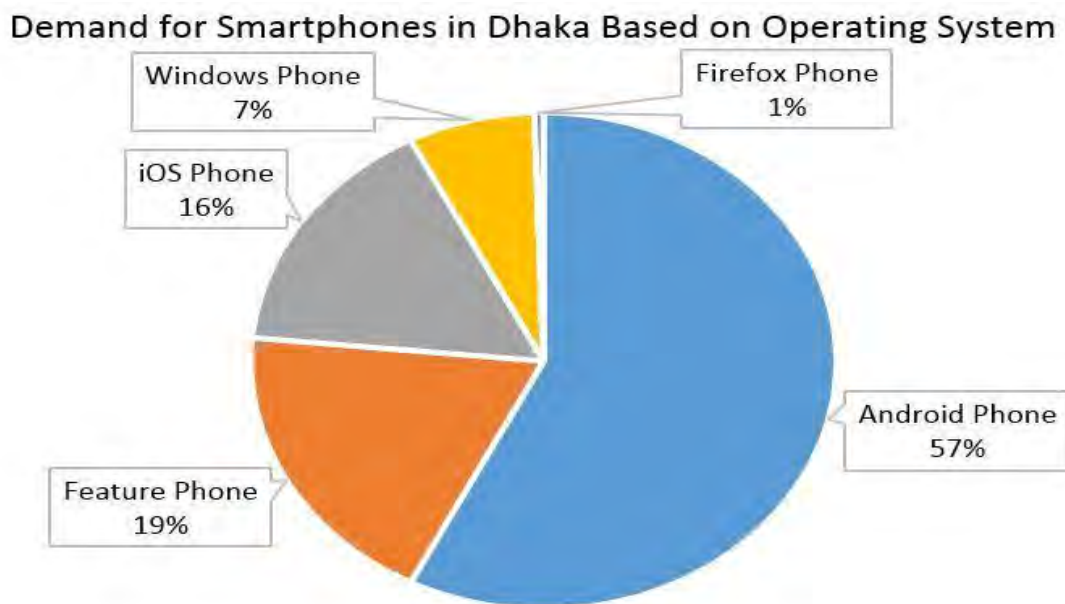


Fig 2.1.2 (a): Usage of cellphones based on Android OS in Dhaka City

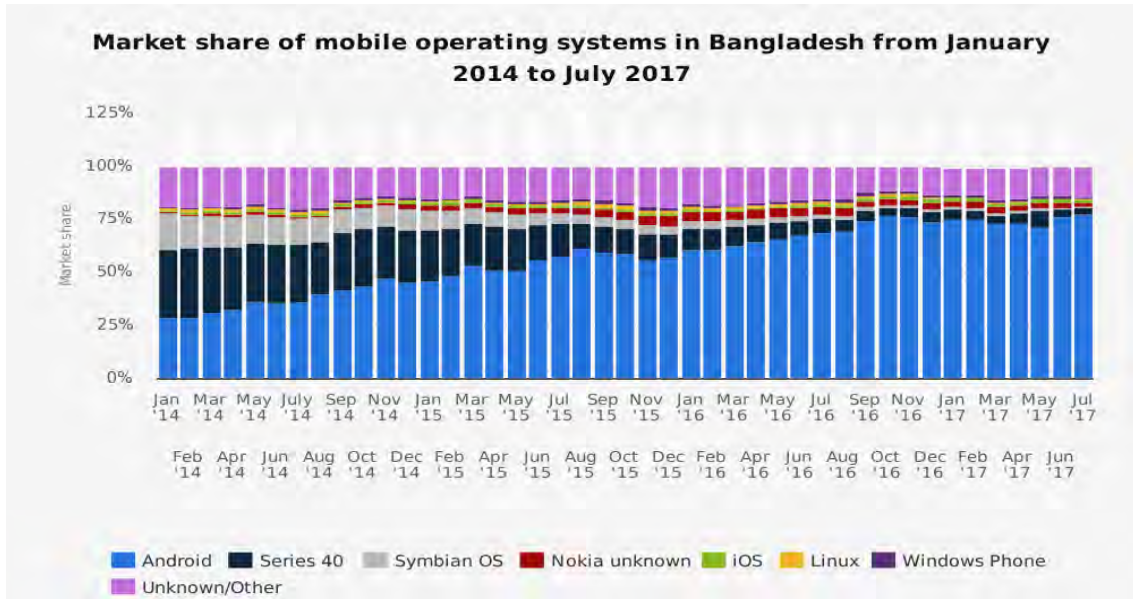


Fig 2.1.2 (b): Android phone users in Bangladesh from January '14 to June '17

According to Fig 2.1.2 (b), in Bangladesh, most of the mobile users use the Android and the statistic shows us that the number of android user is increasing. The android UI, UX is really attractive and easy to use. Android attracts innovation from both hardware and software developers. In android you can run any app you want. It's easily upgradable. Even we can buy this with a very cheap price which is start from 2500 BDT and it's really very available even we can buy this from the nearest Phone shop. Lastly, in our survey 88.6% people voted for android as a platform while using our app. So in every aspect android is better than any other platform in Bangladesh and that's why we choose this platform.

Chapter 3

CONTRIBUTION

In the development of this app, we have used the basic language of Android - Java, as well as XML for establishing the UI properly. This app is built on Android Studio and has taken the help of Firebase as the back-end service. We also needed a few libraries and permissions as well to make the best possible outcome of the built-in widgets. Lastly, the algorithm is based entirely on our research and the flow of activities in the main app. For further explanation of the logical implementation, snaps of code snippets have been attached with each feature.

3.1 Quick View of the Project

3.1.1 Languages

- Java (Android version)
- XML

3.1.2 Tools & Servers

- Android Studio
- Firebase
 - Authentication
 - Realtime Database
 - Storage

- Crash Reporting
- Analytics
- Cloud Messaging
- Google Play Services

3.1.3 Libraries

We have used the following libraries to provide necessary functionalities to our m-learning app:

```
compile 'com.android.support:appcompat-v7:25.3.1'  
compile 'com.android.support:design:25.3.1'  
compile 'com.android.support:cardview-v7:25.3.1'  
compile 'com.android.support:palette-v7:25.3.1'  
compile 'com.android.support:customtabs:25.3.1'  
compile 'com.android.support.constraint:constraint-layout:1.0.2'  
compile 'com.android.support:support-v4:25.3.1'  
compile 'com.android.support:support-vector-drawable:25.3.1'  
compile 'com.google.firebase:firebase-core:11.0.2'  
compile 'com.google.firebase:firebase-database:11.0.2'  
compile 'com.google.firebase:firebase-auth:11.0.2'  
compile 'com.google.firebase:firebase-storage:11.0.2'  
compile 'com.firebaseui:firebase-ui:2.0.1'  
compile 'com.google.firebase:firebase-crash:11.0.2'  
compile 'com.google.firebase:firebase-messaging:11.0.2'  
compile 'com.prolificinteractive:material-calendarview:1.4.3'  
compile 'com.jjoe64:graphview:4.2.1'  
testCompile 'junit:junit:4.12'
```

Fig 3.1.3: Imported Libraries in Gradle File

The main libraries are briefly discussed below:

- **com.android.support: design: 25.3.1:** This is the Android Design Support Library. Above SDK version 23 there are some new built-in design library for android app development. This is called Material Design.
- **com.android.support: cardview-v7:25.3.1:** This is a special feature of the Material Design which allows the developer to make GridViews and ListViews with smooth transitions and attractive edges.
- **com.android.support: palette-v7:25.3.1:** This is a library from Android that can be used to extract prominent colors from an image. It has helped us to get the right colors for the vibrant look of our app.
- **com.android.support: customtabs-v7:25.3.1:** This support library manages the initialization and use of custom tabs in the UI, especially for TabHosts and TabLayout.
- **com.android.support: design: 25.3.1:** This library allows the use of various necessary UI materials of Material Design for this app, like Navigation Drawer and nested tabs.
- **com.google.firebase: firebase-core: 11.0.2:** This is the most essential library for this app because this and its branches provide us a number of the Firebase services. Firebase is a mobile backend service which takes care of the database, storage, analytics, crash reports, authentication and more for an app, built on Android, iOS or other platforms. We used its services to keep track of our authenticated users, maintaining user access privilege and keeping an updated database and files storage in real time.

- **com.firebaseui:firebase-ui: 2.0.1:** This is the Firebase UI library specially developed for Android applications. It is in fact a set of open-source libraries from Google with Apache 2.0 license, which lets the data on your app be updated in real time based on the Firebase Database. It also lets the developer use other sites' information for authentication such as Google, Facebook and Twitter.
- **com.prolificinteractive:material-calendarview: 1.4.3:** We needed this library to use the customized calendar view, so that we can mark the necessary events for students and faculties with different color codes.
- **com.jjoe64:graphview: 4.2.1 :** To implement the results graph, we needed this library which provided us with line graph, bar graph and pie charts.

3.1.4 Permissions

Here are the permissions our app requires from the user's device to run this app. The user will be prompted with a list of these permissions while installing and s/he can choose to grant them.

```

es-permission android:name="android.permission.GET_ACCOUNTS" />
es-permission android:name="android.permission.READ_PROFILE" />
es-permission android:name="android.permission.INTERNET" />
es-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
es-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
es-permission android:name="android.permission.READ_CALENDAR" />
es-permission android:name="android.permission.WRITE_CALENDAR" />
es-permission android:name="com.android.alarm.permission.SET_ALARM"/>
es-permission android:name="android.permission.WAKE_LOCK" />

```

Fig 3.1.4: Permissions of this app in Manifest File

- **android.permission.GET_ACCOUNTS:** Since the minimum SDK our app allows is API 19, we need this permission to get any detail about the user accounts in the Accounts Service. From API 23, this is not required anymore if the authenticator shares the signature with the one managing the account.
- **android.permission.READ_PROFILE:** This permission is to perform queries on the information of the user who owns the device.
- **android.permission.INTERNET:** This app cannot extract the database information and storage materials from the cloud if no internet is present. So this permission to use the internet is essential.
- **android.permission.READ_EXTERNAL_STORAGE:** To upload files like documents and images from the external storage, we need this permission. This will be specifically helpful to the faculty members.
- **android.permission.WRITE_EXTERNAL_STORAGE:** To download the storage files into the external storage like microSD card, we need this permission. This will especially be helpful for the students.
- **android.permission.READ_CALENDAR:** We need this permission to get the current date, day and time in order to display the right events in the Event Display section of our app.
- **android.permission.WRITE_CALENDAR:** This permission helps us to add new information to the default calendar of the device and thus help us to add important events to the calendar.

- **com.android.alarm.permission.SET_ALARM:** This permission is necessary to set alarms automatically according to the event dates shown in the calendar. As soon as there is a new event added in any of the assigned courses, this will be used to set an alarm at 8am of the due date.

3.2 Algorithm

This is an overall step-by-step view of how the different aspects of the app work:

1. **Extracting data from institution database and replacing the dummy data with it**
2. **For all users : Authenticate by**
 - email and password
 - google sign in
 - specific user role
3. **If user_role=student:**
 - Student's database initializes**Else if user_role=teacher:**
 - Teacher's database initializes
4. **Navigation drawer clicked:**
 - Header image, title and subtitle displaying user name and email.
 - Drawer section displaying current course titles and logout option.

5. Files → storageReference.department.course.section

- accessed and displayed in Navigation Drawer Fragments

If user_role=student

Download all button's visibility = ON

Upload button's visibility = OFF

Else

Download all button's visibility = OFF

Upload button's visibility = ON

6. Dashboard clicked:

-Main features visible in one screen

a) Data from completed courses:

Students Reference → user email → courses completed

→ semester

→ course name

→ grade

→ GPA

b) Current routine clicked:

if user_role=teacher

Courses = teachersDatabase.teachersEmail.assignedCourses

elseif user_role=students

`Courses = studentsDatabase.studentsEmail.assignedCourses`

For each course in courses:

- i) find course database reference from
`departmentsDatabase.departmentName.courseName`
- ii) save it to `courseRef`
- iii) `courseNameTab.content = courseRef.routine`

c) Current marksheet clicked:

`StorageRoot→department→course→section→marksheet.xls`

- Non-editable by students
- Regularly updated by faculties

d) Chatroom clicked:

`DatabaseRoot→department→course→section→chatroom`

→ Contains all messages of that particular section

e) Calendar clicked:

Display events from the following branches:

-database root for any global announcement from the institution

-department→course→section for specification

announcements

f) Notification clicked:

-display any change to following branches:

- i) `DbRef.courseSection.Chat`
- ii) `DbRef.courseSection.Calendar`
- iii) `DbRef.department.Calendar`
- iv) `storageRef.courseSection.materials`
- v) `DbRef.user`

7. Log out.

3.3 Features

1. Full Manipulation of course materials by teachers

Faculties of the respective institution can upload, download and edit the corresponding course materials through this app.

2. Downloading and selective uploading of files by students

Students of the institution can only have the full access to download course materials and occasionally upload submissions if allowed by the respective course teacher.

3. Specified user privileges

Not all users are allowed all functionalities of this app. Their roles and privileges are specified acutely through Firebase rules.

4. Events calendar marking important events and deadlines

A customized material `CalendarView` has been used to make sure the users can see all important dates at a glance

5. Activity dashboard

All frequently needed features are kept at one place through dashboard in order to make the user experience better.

6. Push notifications

Whenever a new document is uploaded in any assigned course or whenever an important event is added, the users get push notifications through Firebase Cloud Messaging.

7. Peer chatrooms

The members of a particular course section can discuss among themselves about any relevant problems they are following.

8. Online library of academic books and other documents

From Firebase Storage, all available books and other academic documents of a department are made accessible through this app.

9. Alarm Reminder for Events

From the implemented event calendar, alarms can be triggered for corresponding events via this app.

10. Marksheet view

Updated marksheets of the current courses will be available for viewing to students, while the teachers can edit them in real time through this app.

11. Milestones bar

A milestone bar has been implemented here as the indicator of how much of the current semester has passed and how much is left to look forward to.

12. Results Graph

This is available in the Account Profile section where all previous results of a student are displayed in the form of graph, so that the pupil can detect if he is going uphill or downhill.

13. Student's previous results display

There is also the feature to display the results of previous semesters in details to a student through Expandable Listview.

3.3.1 The Logic and the Implementations

Here we will discuss the various logistics we followed to get the desired values from our database and storage files. Reading this section, it will be clear to the audience how the above algorithm gets executed in the app.

a. Dashboard features according to user role:

Whenever a user signs into the app, the welcome screen prompts them to input their role - of either a student or a faculty. This value then gets drifted to all the activities that follow it. If the user is already signed in during the launch of the app, the Navigation Drawer retrieves his role by calling *Firebase.auth.getCurrentUser()*. This method gives us the current user and by applying *getEmail()* method on the returned value we can get his email. The users are stored in database with the key value of the first part of their email.

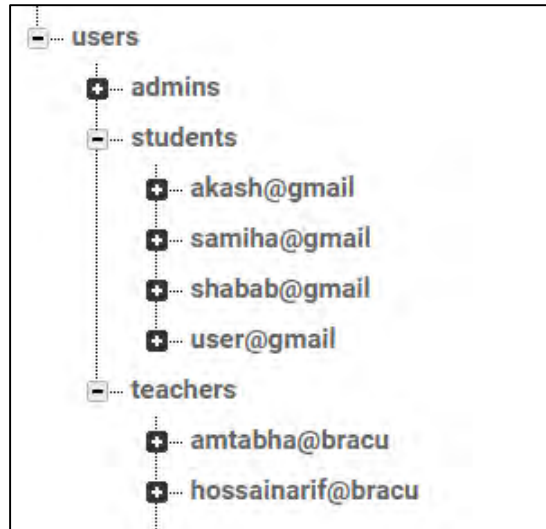


Fig 3.3.1.a (i): Users Database Overview

So with the received email ID, we can identify what role they currently hold by implementing the following logic:

```

intent.putExtra("email", user_email);
int croppedEmailIdLimit = user_email.length() - 4;
String emailID = user_email.substring(0, croppedEmailIdLimit);
if (studentsDatabaseReference.child(emailID) != null)
    selectedRole = "student";
else if (teachersDatabaseReference.child(emailID) != null)
    selectedRole = "teacher";
  
```

Fig 3.3.1.a (ii): Selection of Database Branch Based on User Role

Depending on the role, the features will be displayed in the Dashboard.

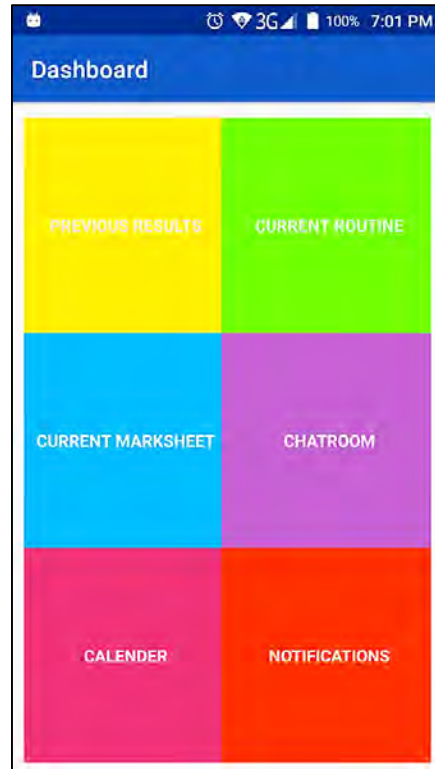


Fig 3.3.1.a (iii): Dashboard Overview

b. Navigation Drawer showing assigned courses:

From the Navigation Drawer's activity, we access the user's database information, including the assigned courses. Each course branch has the course name, the section number as well as the department name attached to it.

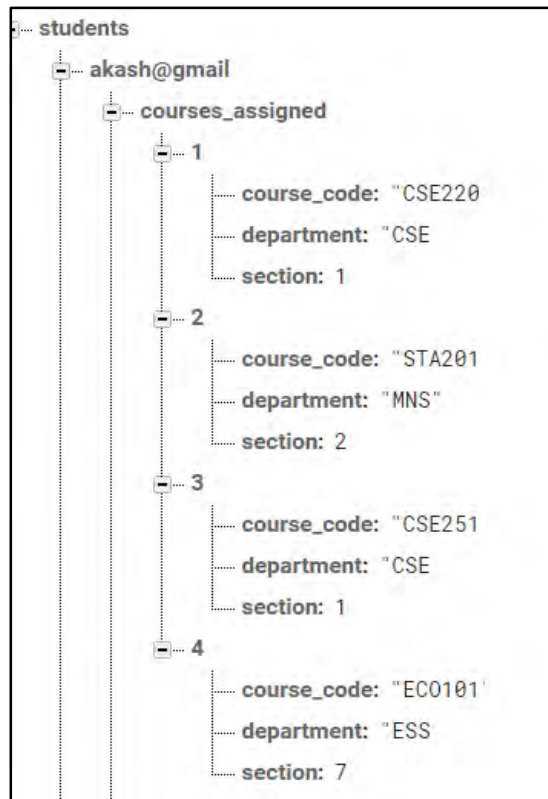


Fig 3.3.1.b (i): Database Structure of Assigned Courses

So we take all these variable values and then run a query on the storage files based on these values. Then through *onDataChange()* we get the desired values and update the drawer menu based on them.

```

NavDrawer | handleCurrentUserInfo() | new ValueEventListener | onDataChange()
val coursesRef = currentUserRef.child("courses_assigned");
coursesRef.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        Log.e(TAG, "onDataChange : COURSES");
        val numberOfChildren = dataSnapshot.childrenCount;
        numberOfCourses = numberOfChildren;
        Log.e(TAG, "#412 : number of courses = $numberOfCourses");
        assignedCourses = new String[numberOfCourses][2];
        departments = new String[numberOfCourses];
        var i = 1;
        for (val postSnapshot : dataSnapshot.children){
            val key = postSnapshot.key;
            val course = postSnapshot.child("course_code").getValue(String.class);
            val section = postSnapshot.child("section").getValue(Long.class);
            assignedCourses[i-1][0] = course;//courseID
            assignedCourses[i-1][1] = section;//section
            val department = postSnapshot.child("department").getValue(String.class);
            departments[i-1] = department;
            i++;
        }
        //filling up the drawer options
        updateOptionsMenu(myMenu);
    }
}

```

Fig 3.3.1.b (ii): Extraction of Assigned Courses Information from Database

The resulting values replace the default values in the drawer menu as you can see in the screenshot below:

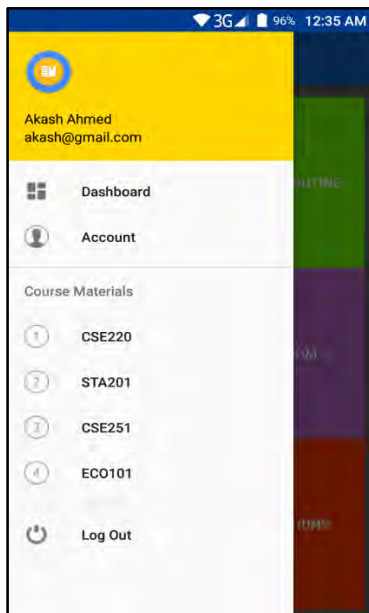


Fig 3.3.1.b (iii): Drawer Menu

c. **Previous Results:**

The results of all previous semesters of each student, in terms of GPA and grades, are stored in the database according to the semesters under the *completed_courses* key. The year is mentioned as the prefix instead of the semester name because that way the results can be sorted in the right order by default in the database.

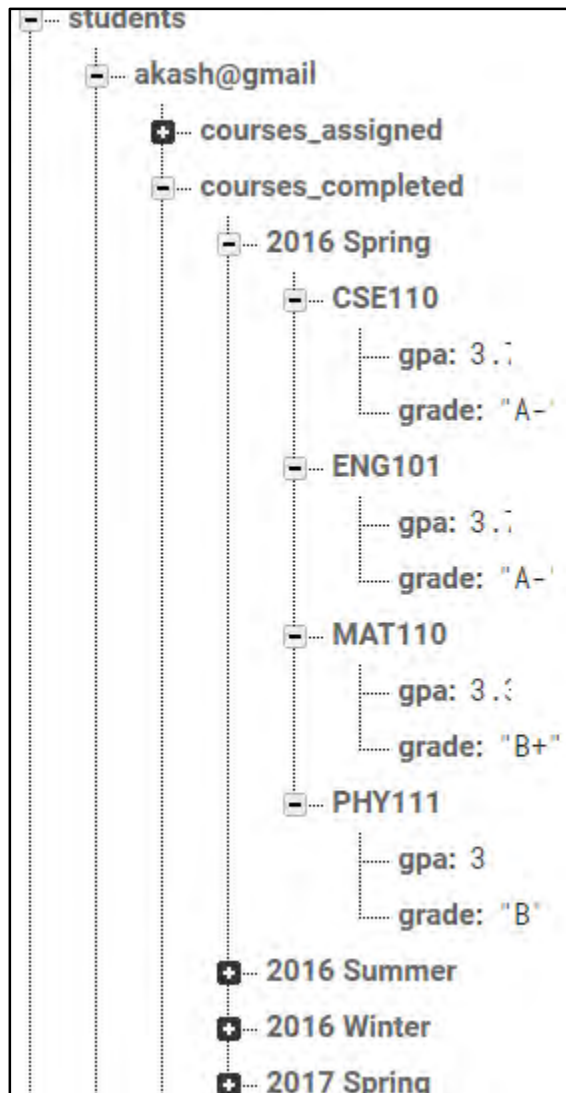


Fig 3.3.1.c (i): Sorting Process in Database

To retrieve these values we implement the following logic, by using HashMaps and generic Lists to store the results onto the semester. We have used arrays for the intermediary data exchange since otherwise it was impossible to get the values sorted into types, namely - course name, course GPA and course grade. We did this sorting because we already made the adapter beforehand with default static values which were programmed to take inputs in this particular way.

```

PreviousResults populateLists() new ValueEventListener onDataChange()
long count = dataSnapshot.getChildrenCount();
numberOfPreviousSemester = (int) count;
numberOfCoursesDone = new int[numberOfPreviousSemester];
resultList = new ArrayList<String>();
int semesterCount = 0;
for (DataSnapshot snap : dataSnapshot.getChildren()) {
    String semester = snap.getKey();
    semester = reverseName(semester);
    semester_headers.add(semester); //semester's name
    long c = snap.getChildrenCount();
    numberOfCompletedCourses = (int) c;
    numberOfCoursesDone[semesterCount] = numberOfCompletedCourses;
    int index = 0;
    Double[] gpas = new Double[numberOfCompletedCourses];
    for (DataSnapshot snapChild : snap.getChildren()){
        resultList.add(snapChild.getKey()); //course name
        if(snapChild.child("gpa").getValue(Double.class)!=null) {
            gpas[index] = snapChild.child("gpa").getValue(Double.class); //gpa
        }
        resultList.add(String.valueOf(gpas[index]));
        resultList.add(snapChild.child("grade").getValue(String.class)); //grade
        index++;
        semester_result_details.put(semester, resultList);
    }
    adapter.notifyDataSetChanged();
}

```

Fig 3.3.1.c (ii): Extraction of Previous Semester Result from Database

While implementing the values through the Adapter we faced a critical problem and took StackOverflow’s help. Then we displayed the values through an Expandable ListView as following.

The screenshot shows a mobile application interface with a status bar at the top displaying 3G signal, 100% battery, and 5:24 PM. The main title is 'Previous Results'. The data is organized into four seasonal sections, each with a yellow header and a table of results.

| SPRING 2016 | | |
|-------------|----|-----|
| CSE110 | A- | 3.7 |
| ENG101 | A- | 3.7 |
| MAT110 | B+ | 3.3 |
| PHY111 | B | 3.0 |
| SUMMER 2016 | | |
| WINTER 2016 | | |
| BUS101 | B+ | 3.3 |
| CSE230 | A- | 3.7 |
| CSE260 | A | 3.7 |
| MAT215 | B | 3.0 |
| SPRING 2017 | | |

Fig 3.3.1.c (iii): Result view

d. Calendar:

In the calendar feature we have included a CalendarView which automatically shows the

current date and a ListView on the bottom to display the important events for the current user. This applies to both the user groups of teacher and students, because this lets them know all the important events of submission deadlines, exam dates and more under one screen.

We retrieve all these events from the database branch of the corresponding section under the department key. The events for individual sections are stored in the database in the following format:

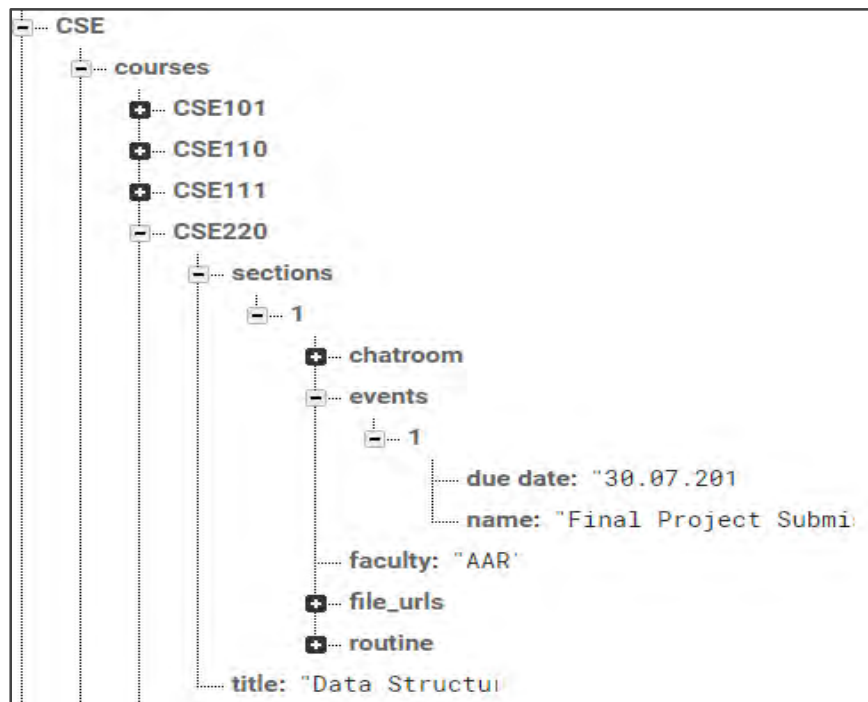


Fig 3.3.1.d (i): Database Structure of Calendar

We use a query with the similar combination of department name, course name and section number to get all these data from the database, by using the following logic:

```

CalendarDisplay | getEventsFromDatabase() | new ValueEventListener | onDataChange()
7         allDepartmentsRef.addListenerForSingleValueEvent(new ValueEventListener() {
8             @Override
9             public void onDataChange(DataSnapshot dataSnapshot) {
10                //retrieving events from dataSnapshot
11                for (int i=0; i<numberOfCourses; i++) {
12                    DataSnapshot ds = dataSnapshot.child(departments[i])
13                        .child("courses").child(assignedCourses[i][0])
14                        .child("sections").child(assignedCourses[i][1])
15                        .child("events");
16                    long eventsCount = ds.getChildrenCount();
17                    int numberOfEvents = (int) eventsCount;
18                    Log.e(TAG, "#104 : " + String.valueOf(numberOfEvents));
19                    for (DataSnapshot snap : ds.getChildren()) {
20                        events[i][0] = snap.child("name").getValue(String.class);
21                        Log.e(TAG, "#106 : event name = " + events[i][0]);
22                        events[i][1] = snap.child("due date").getValue(String.class);
23                        Log.e(TAG, "#110 : due date = " + events[i][1]);
24                    }
25                    //applying it on the CalendarView Adapter
26                    calendarAdapter = new CalendarEventsAdapter(CalendarDisplay.this,
27                        events, calendarView);
28                    eventsView.setAdapter(calendarAdapter);
29                }
30            }
31        }
    
```

Fig 3.3.1.d (ii): Extraction of Events from Database

For displaying these events through a convenient UI, we have implemented the customized CalendarView of the Prolific Interactive package on Git. Because of this package, we can change the visualization of each event date, current date as well as any new date selected by the user. Each event date is marked magenta, the current date displays in green and any other selection by the user is highlighted in yellow. This way the user can view the important dates in the calendar and at the same time they can see the events themselves in the events List below. These events are displayed in the ListView of the calendar feature called *eventsView* and they combinedly look like the following:

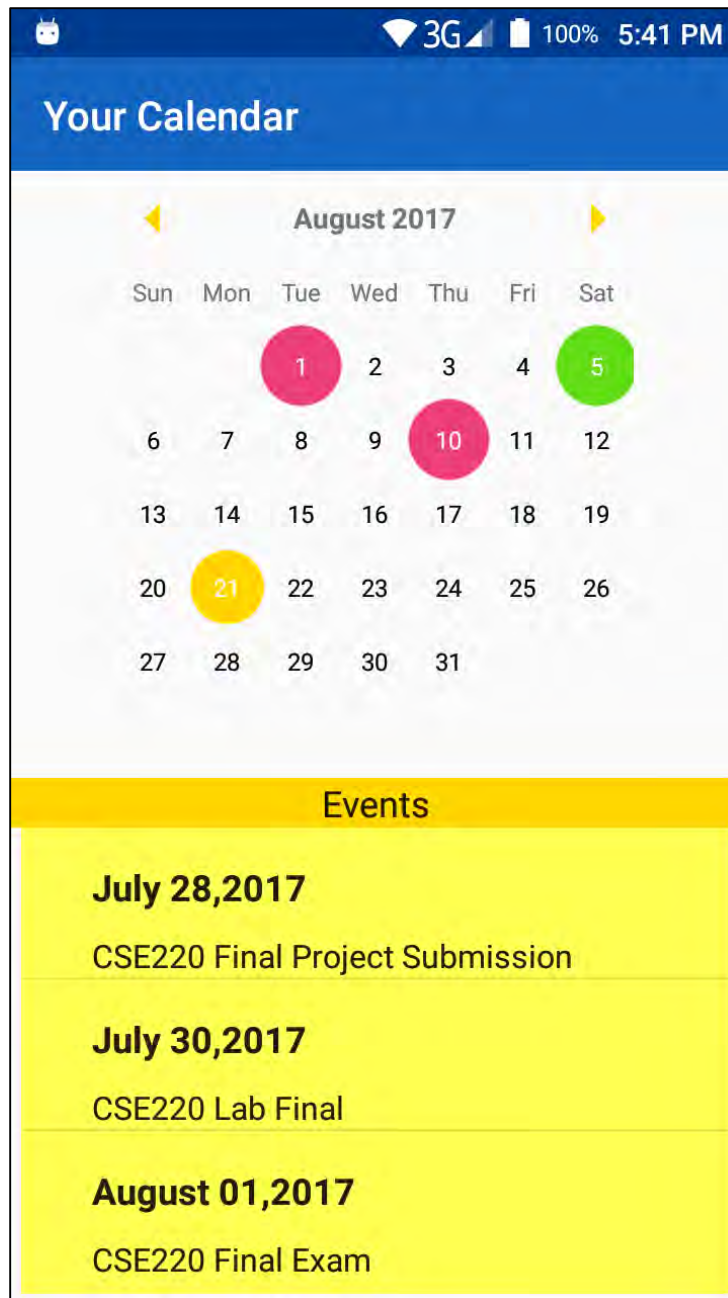


Fig 3.3.1.d (iii): Calendar displaying Course Events

e. **Peer Chat Room:**

In our chatrooms feature all the members of a particular course section can discuss amongst each other about necessary relevant topics. On the corresponding section's branch in the database, the messages are added and displayed in real time. Every time the user presses "Send", the message along with the username gets added here under a unique auto-generated key.

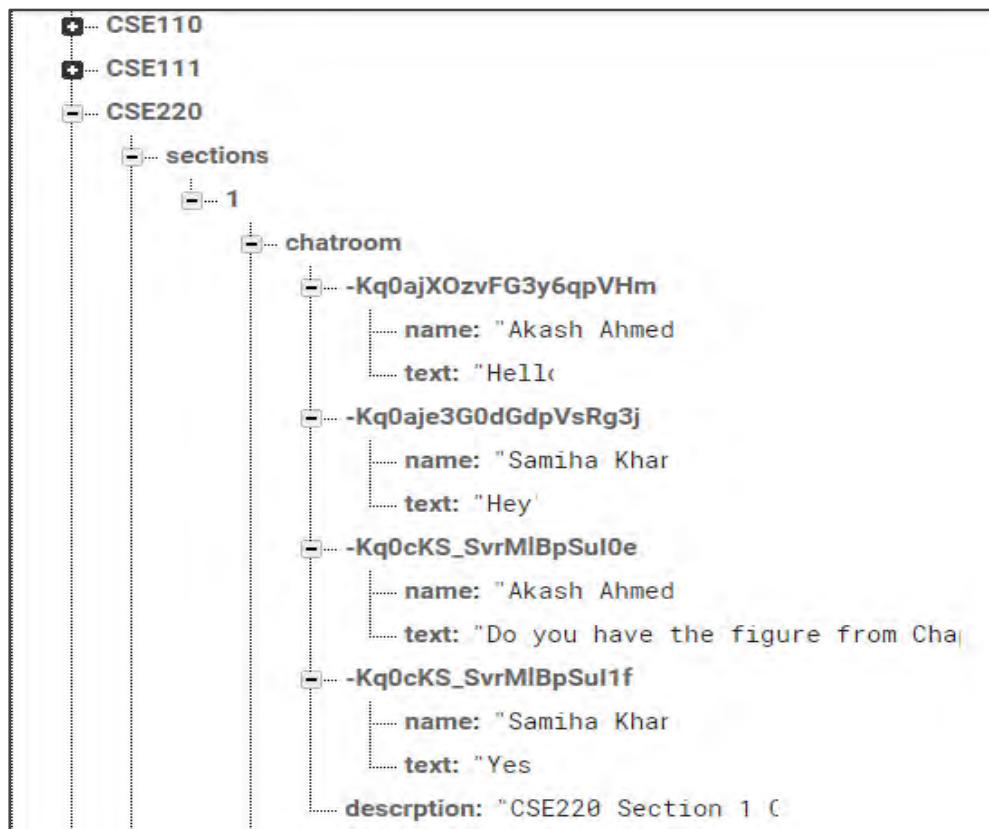


Fig 3.3.1.e (i): Database Structure of Chatroom

This shows up in real time on our chatroom feature in the following way:

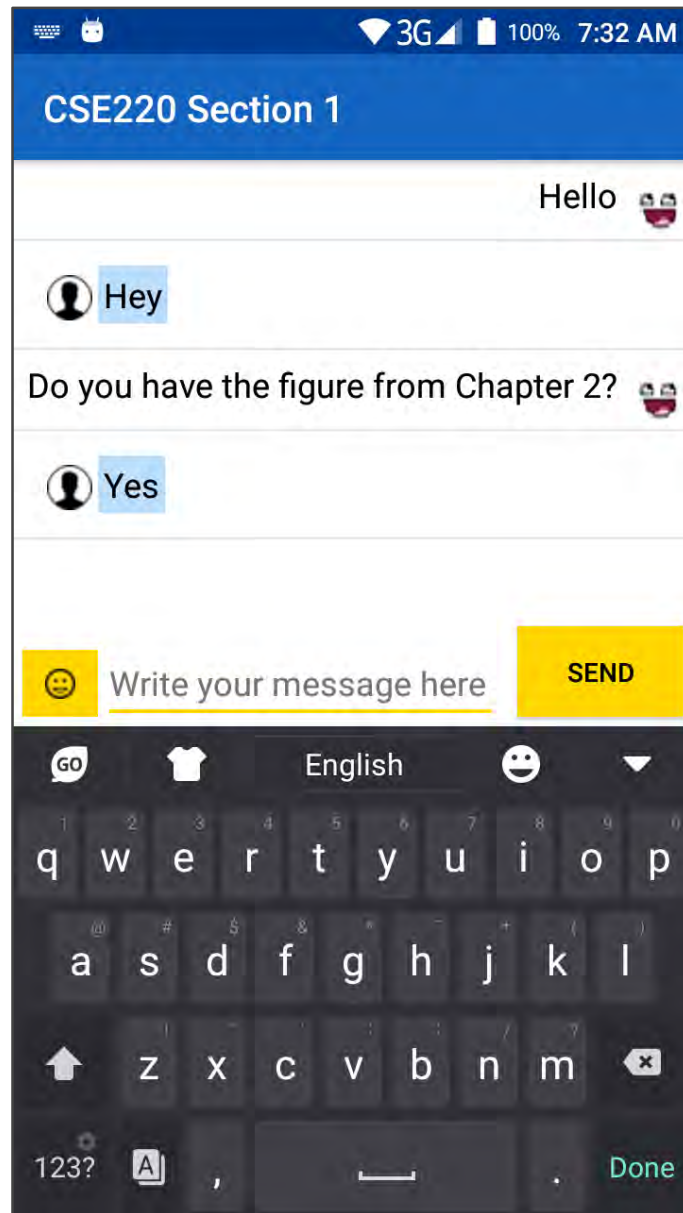


Fig 3.3.1.e (ii): Chatroom View

The logistics behind this implementation is shown below:

```
ChatMessageAdapter | getAllMessage() | new ValueEventListener | onDataChange()
chatsRef.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        Log.e(TAG, "Adapter: onDataChange");
        long messageNumber = dataSnapshot.getChildrenCount() - 1;
        previousMessagesNumber = (int) messageNumber;
        messages = new String[previousMessagesNumber];
        senders = new String[previousMessagesNumber];
        Log.e(TAG, "#105: number of previous messages = " + previousMessagesNumber);
        if (previousMessagesNumber > 0) {
            int messageCounter = 0;
            for (DataSnapshot snap : dataSnapshot.getChildren()) {
                if (!snap.getKey().equals("description") && messageCounter < previousMessagesNumber) {
                    String key = snap.getKey();
                    String message = snap.child("text").getValue(String.class);
                    String senderName = snap.child("name").getValue(String.class);
                    if (senderName != null)
                        senders[messageCounter] = senderName;
                    messages[messageCounter] = message;
                }
                messageCounter++;
            }
        }
    }
})
```

Fig 3.3.1.e (iii): Extraction of Chatroom Messages from Database

f. Current Mark sheet:

The current mark sheet displays the updated marks of the current courses for the student, as provided by the faculty. If the current user is a faculty, then he gets to view the whole mark sheet for each section he is assigned to from the storage files.

Here is a view of how it looks for the student:

The screenshot shows a mobile application interface with a status bar at the top displaying 3G signal, 100% battery, and 7:50 AM. The main title is 'Current Marksheet'. Below the title, there are four tabs: CSE220, STA201, CSE251 (which is selected and underlined), and ECO101. The main content area is a table with a light green background, listing assessment items and their scores. The 'TOTAL' row is highlighted in a darker green.

| CSE220 | STA201 | CSE251 | ECO101 |
|--------|--------|-----------------|---------------|
| | | Days of Absence | 2 |
| | | Quiz 1 | 8/10 |
| | | Quiz 2 | 14/20 |
| | | Quiz 3 | 6/10 |
| | | Quiz 4 | 10/10 |
| | | Mid | 16/20 |
| | | Assignment | 5/5 |
| | | Lab | 18/20 |
| | | Finals | 30/35 |
| | | TOTAL | 77/100 |

Fig 3.3.1.f: Marksheet View

g. Current Routine:

The current routine feature displays the routine of the assigned courses, no matter what the user role is. Both user groups of teacher and student can benefit from this because they can see the assigned slots of the whole week or the whole day in one screen. By default, when clicked on it shows the classes of that particular day only. When pressed on “View Full Routine”, it displays the full routine

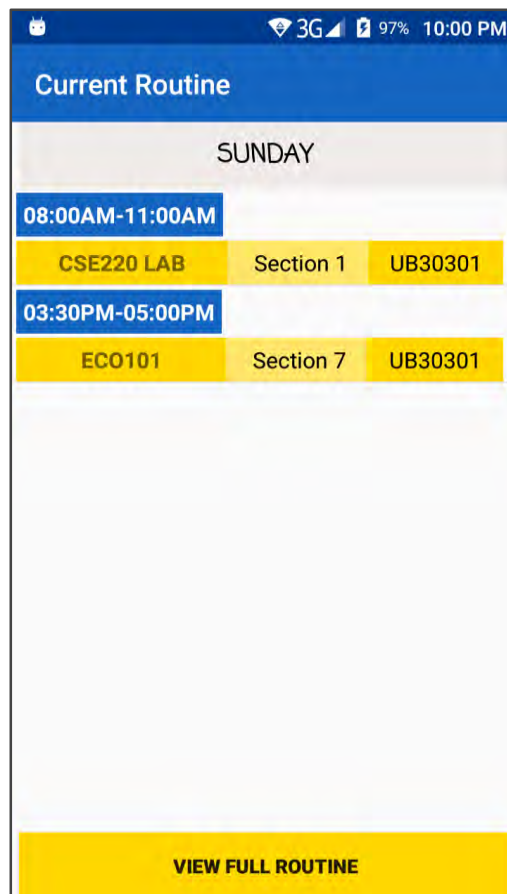


Fig 3.3.1.g (i): Day Based Routine View

| Time Slots | SUNDAY | MONDAY | TUESDAY | WEDNESDAY | THURSDAY |
|------------------|----------------------|------------------|------------------|------------------|------------------|
| 8:00AM - 9:30AM | CSE220 LAB - 1 - FHR | - | - | - | - |
| 9:30AM - 11AM | - | CSE251 - 1 - SUP | CSE251 - 1 - SUP | CSE251 - 1 - SUP | CSE251 - 1 - SUP |
| 11AM - 12:30PM | - | CSE220 - 1 - NUS | CSE220 - 1 - NUS | CSE220 - 1 - NUS | CSE220 - 1 - NUS |
| 12:30PM - 1:00PM | - | - | - | - | - |

Fig 3.3.1.g (ii): Full Week's Routine View

h. Course materials:

The course materials show up as soon as you click on the desired course name from the navigation drawer. There is not default method of getting the list of all the files in a particular directory from the storage. So we made a method such that, every time a file is uploaded to that particular directory by a user, a database entry with auto generated key will be established under the branch of that particular section with the filename and file type as values.

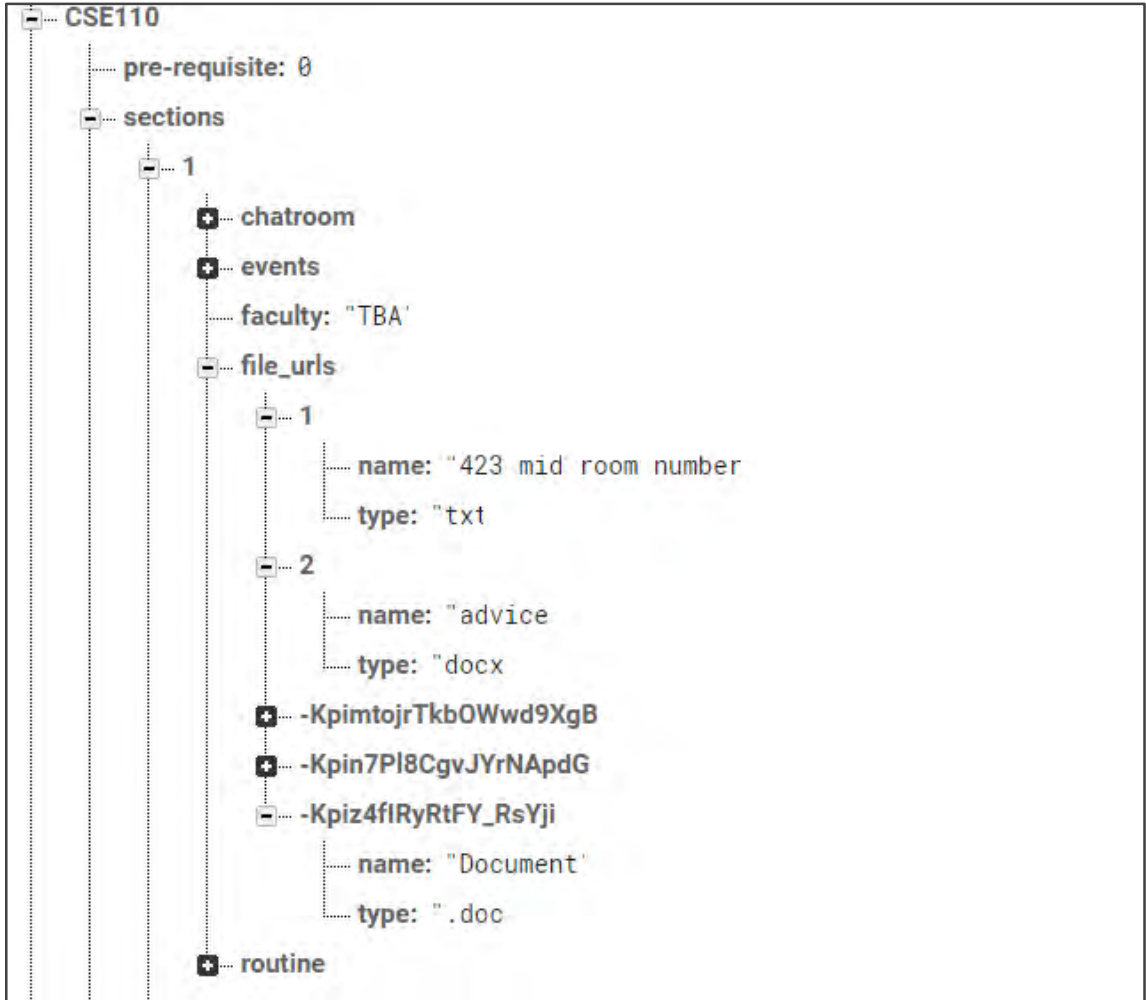


Fig 3.3.1.h (i): Database Structure of Course Materials

All these keys under the *file_urls* branch represent the documents in the storage:

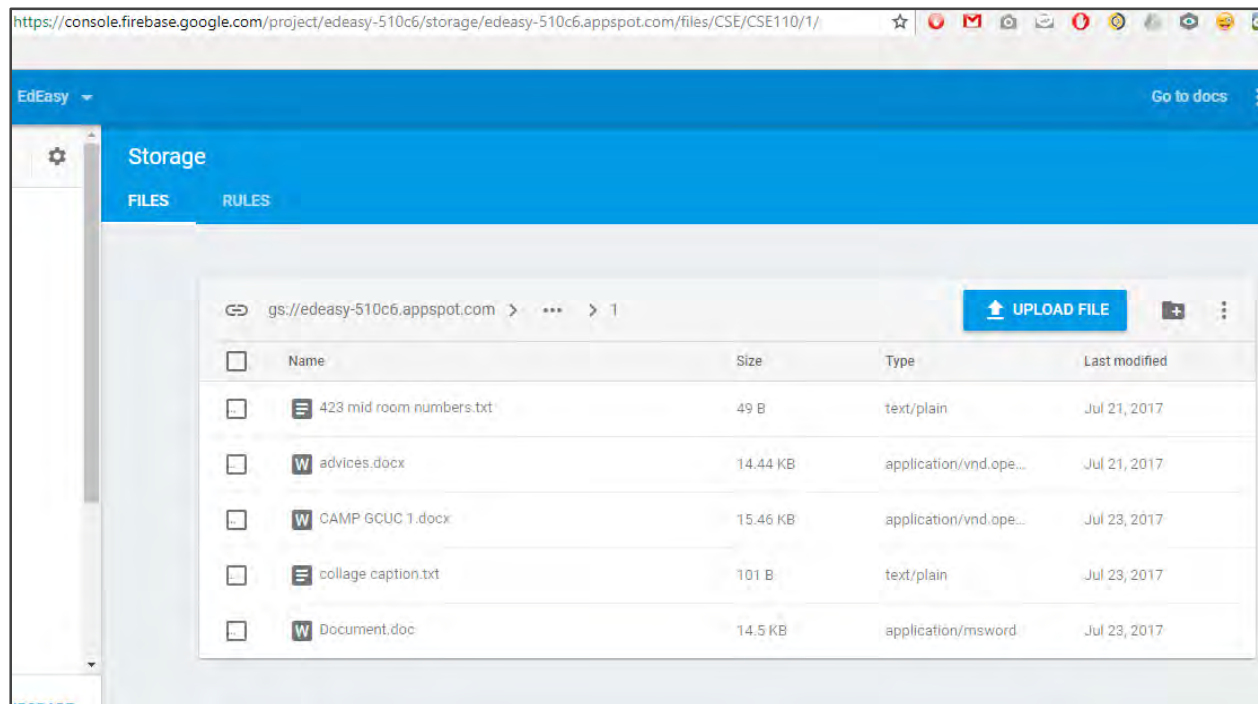


Fig 3.3.1.h (ii): View of Course Materials in Storage

By utilizing these two directories of database and storage we display the course materials in a ListView. Clicking on any of them will show the download option, as follows:

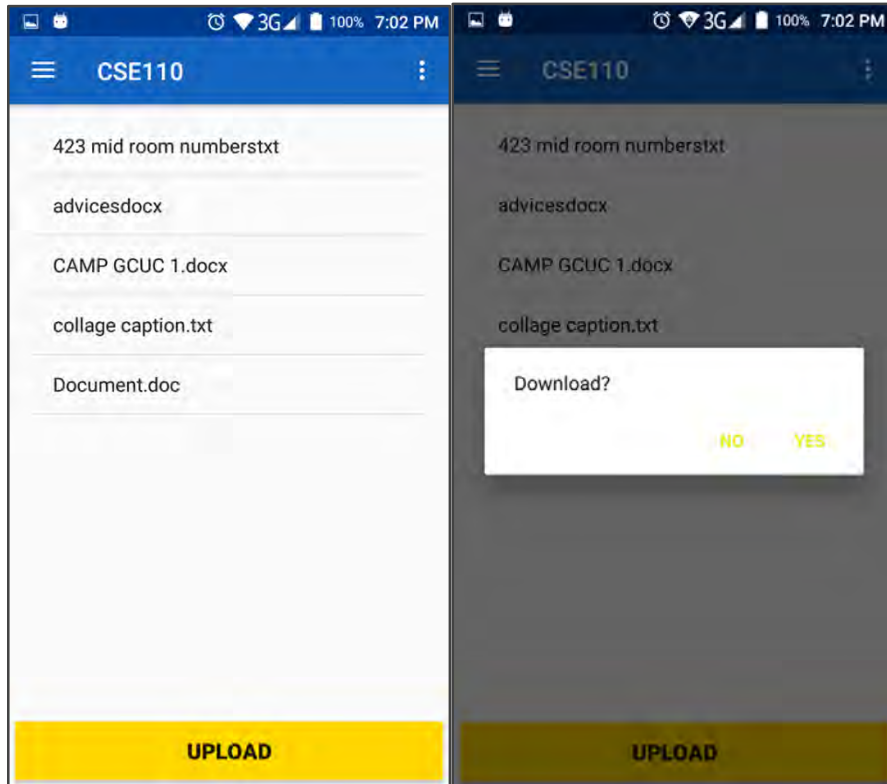


Fig 3.3.1.h (iii): Course Materials View and Download

i. Account/Profile:

You can view a summary of all the relevant information of your account in this section, which is included in the navigation drawer options. You can have a customized photo representing you here, along with all the assigned courses and sections. If you are a student you can view the total number of completed credits and student ID in here.

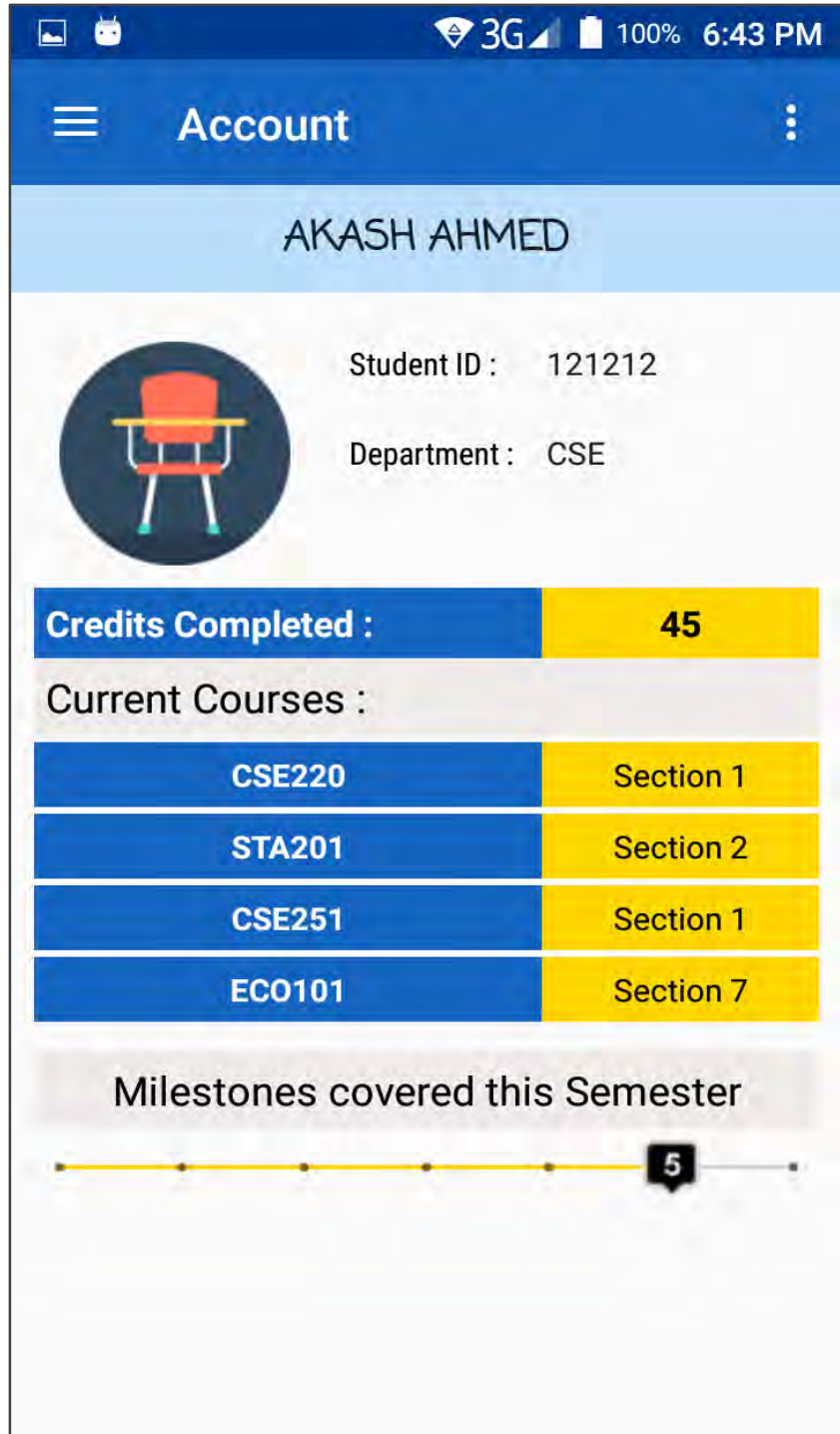


Fig 3.3.1.i (i): Account Profile

This section also includes a Milestone bar which signifies how much of the semester has passed to give you an idea of how much time is left for it to end. There are six milestones in total and they are divided based on a ratio calculation. For that, we have implemented the following algorithm:

```

AccountProfile | setBeginAndEndDates() | new ValueEventListener | onDataChange()
public void onDataChange(DataSnapshot dataSnapshot) {
    beginDate = dataSnapshot.child("begin date").getValue(String.class);
    endDate = dataSnapshot.child("end date").getValue(String.class);
    //calculation
    int totalSemesterDays = getTotalSemesterDays(beginDate, endDate) ;
    todaysMonth = todaysMonth+1;
    int numberOfDaysPassed = 0;
    if (todaysMonth==beginMonth && todaysDay>beginDay)
        numberOfDaysPassed = todaysDay-beginDay;
    else if (todaysMonth==endMonth){
        if (todaysDay<endDay)
            numberOfDaysPassed = totalSemesterDays-(endDay-todaysDay) ;
        else
            numberOfDaysPassed = totalSemesterDays;
    }else if (todaysMonth>beginMonth && todaysMonth<endMonth){
        numberOfDaysPassed = 30-beginDay;
        int fullMonthsDifference = todaysMonth-beginMonth-1;
        numberOfDaysPassed = numberOfDaysPassed + (fullMonthsDifference*30);
        numberOfDaysPassed = numberOfDaysPassed + todaysDay;
    }
    int milestonesRatio = totalSemesterDays/5;
    m_progress = (numberOfDaysPassed/milestonesRatio) + 1;
    setProgressStuff(m_progress);
}

```

Fig 3.3.1.i (ii): Algorithm of Milestone Ratio and Progress

j. Push Notifications:

Whenever there is a change to the respective database of the section or the department assigned to the user, they get a push notification with the help of Firebase Cloud Messaging. Also, if the admins want, they can send notifications about any recent significant change through the Firebase console.

For this we had to create a helper class and also two children classes which extended from *FirebaseInstanceIdService* and *FirebaseMessagingService* respectively. These are

the classes which make it possible to communicate with the Firebase back-end and therefore transfer the messages in between.

As soon as the app receives a notification while in background, the notification message shows up like this:

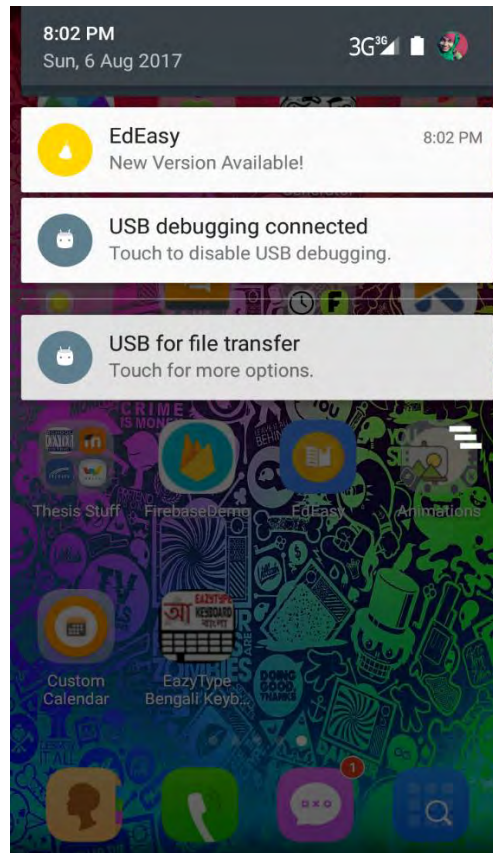


Fig 3.3.1.j: Display of Push Notification

k. Online Library

There are some books and other academic documents which students and faculties both need to access frequently. These can all be accumulated in the respective departments using Firebase Storage and then they will be displayed in

this library. The users can also search for a specific book and download it form here.

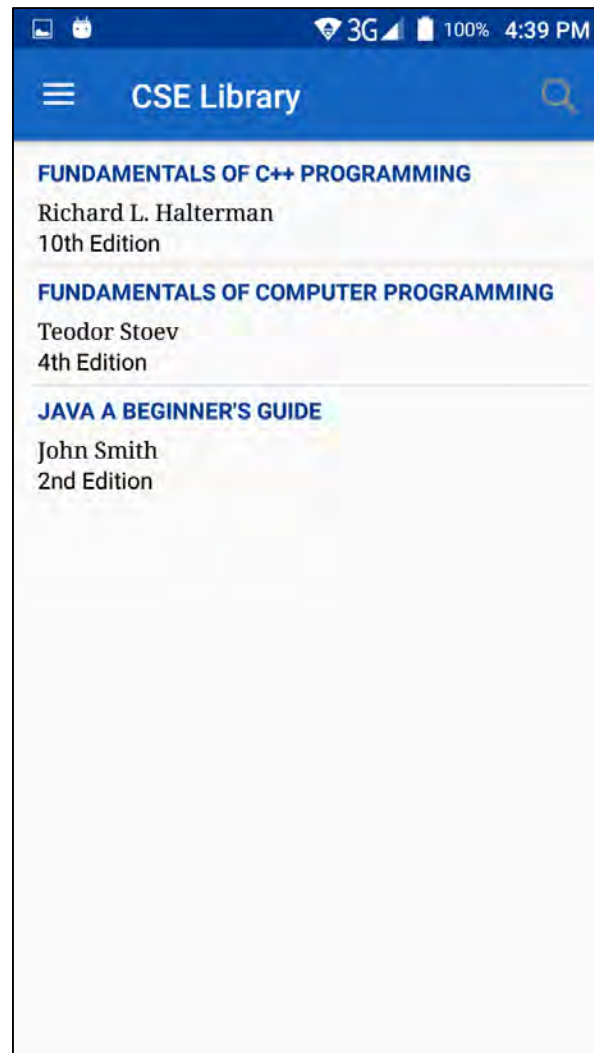


Fig 3.3.1.k: Online Library of CSE Department

I. Results Graph

In the Account Profile feature, there will be a graph representing the ascent or descent of a student's academic progression. For this we have implemented the *LineGraph* element from the

imported library, which begins the graph from the lowest threshold of obtained CGPAs and goes on up to 4.0 vertically. The horizontal measure displays the corresponding semester.

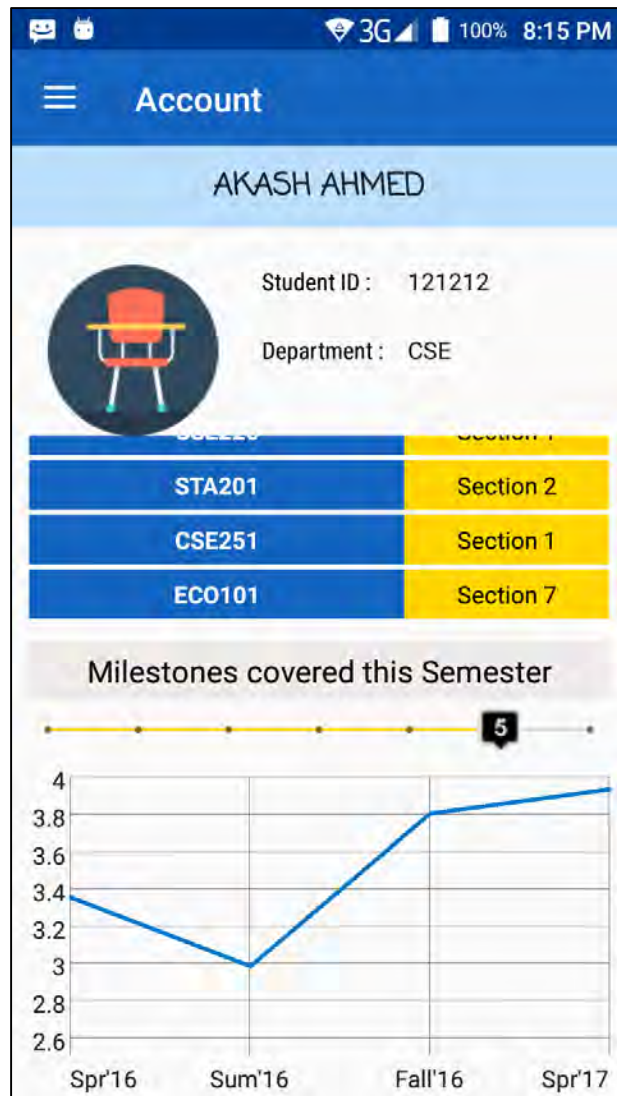


Fig. 3.3.1.1: Results Graph

Chapter 4

CONCLUSION

Here we conclude the documentation by acknowledging the limitations and difficulties we faced during the development of this project. Also we hint at the future aspirations we have for this app and a rough estimation of ways to achieve them. Even though there are many apps with similar motives as us, none of them have successfully been able to provide the users what they claimed. So our main aim in future will be to overcome all those boundaries and difficulties, and create a proper educational app which can solve all the problems of students and coordinators simultaneously.

4.1 Limitations

We could not implement all these features based on the dataset of a real institution. We used dummy data from our university course structures to make sure each feature was functioning the way they are supposed to.

We also had plans of publishing this on Google Play but due to the limited time and our limitations of development, we failed to do so.

4.2 Future Works

- Ensuring proper interaction and access control between user groups
- Enabling additional features (e.g.: online quiz)
- Adding built-in calendar book to let students make their own study plans
- Make the app cross-platform

4.3 References

- [1] Patterson, T. (2012, September 25). *A Review of the Blackboard Learn Mobile App*. Retrieved from <http://www.blackboardguru.com/2012/09/a-review-of-the-blackboard-learn-mobile-app> [Accessed 20 June 2016]
- [2] *Cloud-based Collaboration and Customization Will Drive the M-Education Market in the US Through 2020, Says Technavio*. (2016, May 12). Retrieved from <http://www.businesswire.com/news/home/20160512005038/en/Cloud-based-Collaboration-Customization-Drive-M-Education-Market-2020> [Accessed 17 October 2016]
- [3] Clothier, P. (2014, May 12). *Right Time and Place: mLearning Use Cases*. Retrieved from <https://www.learningsolutionsmag.com/articles/1420/right-time-and-place-mlearning-use-cases> [Accessed 17 October 2016]
- [4] Blackboard Inc. (2015). Mobile Learn™ (4.1.4) [Mobile application software]. Retrieved from <https://play.google.com/store/apps/details?id=com.blackboard.android&hl=en> [Accessed 20 January 2017]
- [5] Digital Campus. (2016). OppiaMobile Learning (6.0.2) [Mobile application software]. Retrieved from <https://play.google.com/store/apps/details?id=org.digitalcampus.mobile.learning&hl=en> [Accessed 28 June 2017]

- [6] Henderson, M. (2015, August 19). *Announcing Lower, Simpler Pricing*. Retrieved from https://firebase.googleblog.com/2015/08/announcing-lower-simpler-pricing_26.html[Accessed 2 August 2017]
- [7] Reese, O. (2016, November 28). *Technology & Virtual Learning: The Secret To Breaking Down Education Barriers & Making Knowledge Accessible*. Retrieved from <http://www.parentherald.com/articles/89910/20161128/technology-virtual-learning-secret-breaking-down-education-barriers-making-knowledge.htm> [Accessed 2 August 2017]
- [8] Stern, S. (2017, June 15). *FirebaseUI for Android v2.0.1*. GitHub Repository, <https://github.com/firebase/FirebaseUI-Android/releases/tag/2.0.1>[Accessed 13 June 2017]
- [9] Colle, Q. (2017, Feb 27). *Prolific Interactive Material Calendar View 1.4.3*. GitHub Repository, <https://prolificinteractive.github.io/material-calendarview/>[Accessed 18 June 2017]