Software Engineering Lab

Implementation Report II - Build II

Course Assistant for Educators

Streamlining the teaching process

Dibyadarshan Hota (16CO154)

Omkar Prabhu (16CO233)

March 30, 2018

1. Basic Information

All the existing functional requirements have been implemented without any changes. Two additional functional requirements have been added:

- Contact us: This allows the user to contact the management team directly.
- View deadlines: This allows the user to view all the project deadlines.

The non-functional requirements remain unchanged.

2. Pending / New Functional Requirements (Implemented)

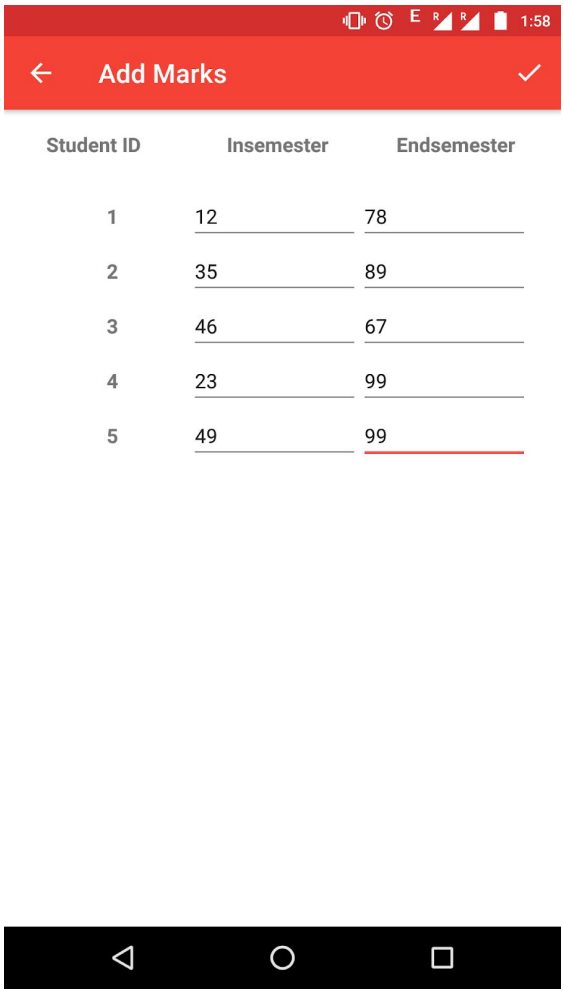
FR.ID	Name	Type (Pending or New)	Description
FR-1.5	Add Marks	Pending	Add in-semester and end-semester marks
FR-1.6	Add Project Deadlines	Pending	Users can set project deadlines
FR-1.7	Display Marks and Attendance Percentage	Pending	User can view marks and attendance percentage
FR-1.8	Search by Student	Pending	Attendance and marks will be displayed student-wise.
FR-1.9	Sort in Student List	Pending	Sorted list of students will be visible
FR-1.10	View Attendance for a Day	Pending	Attendance for a day will be visible
FR-1.14	View Project Deadlines	New	View all project deadlines set by the user
FR-1.15	Contact Us	New	Users facing difficulty in accessing their data or use of the application can contact management team directly

3. Functional Requirements (Not Implemented)

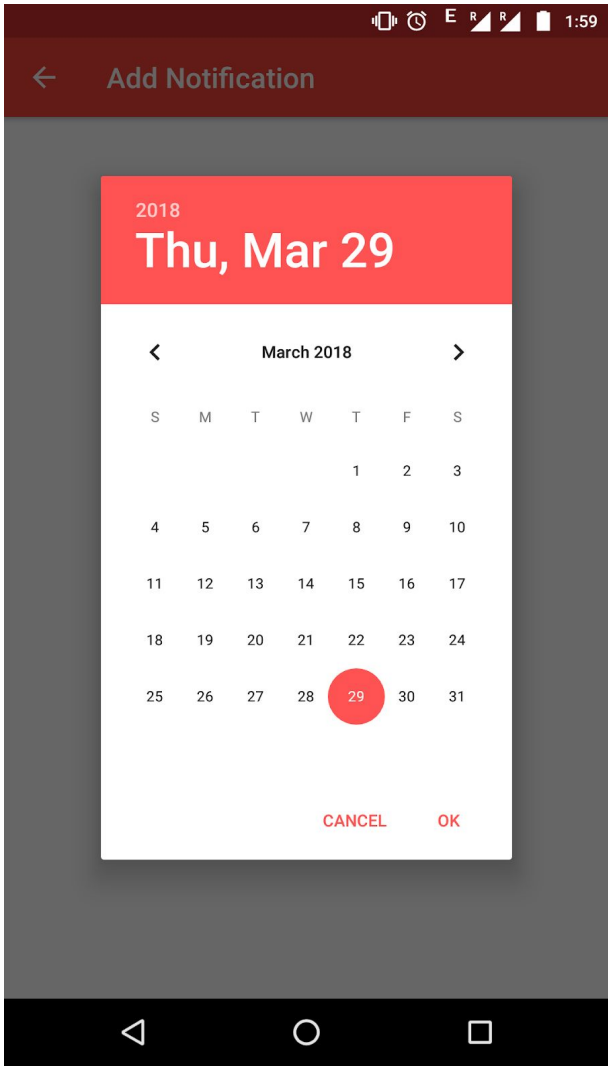
All functional requirements have been implemented in this build.

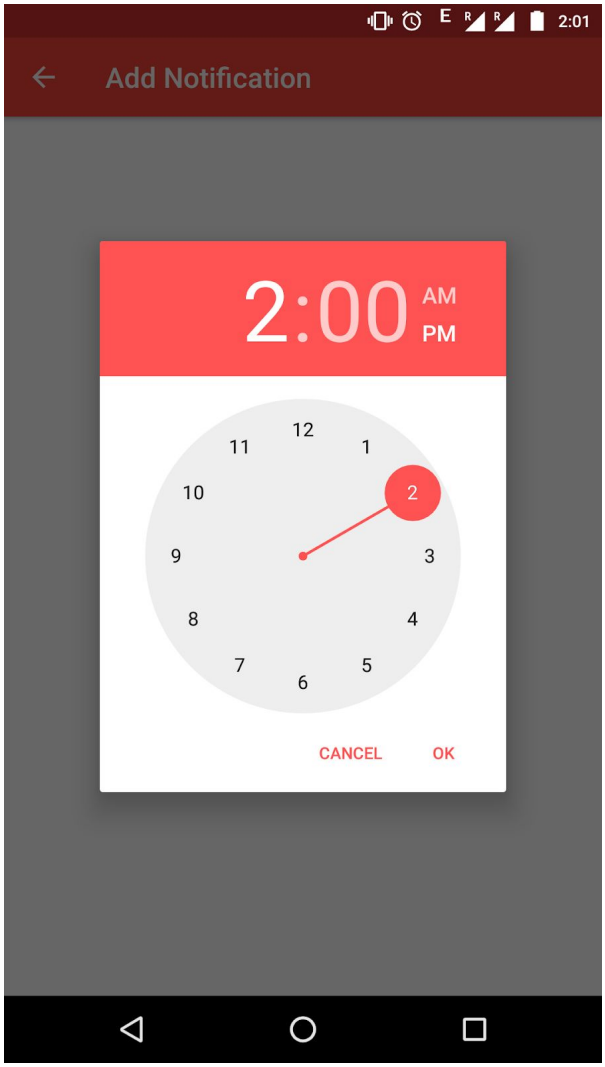
4. Screenshots of Functional Requirements Implemented in this Build

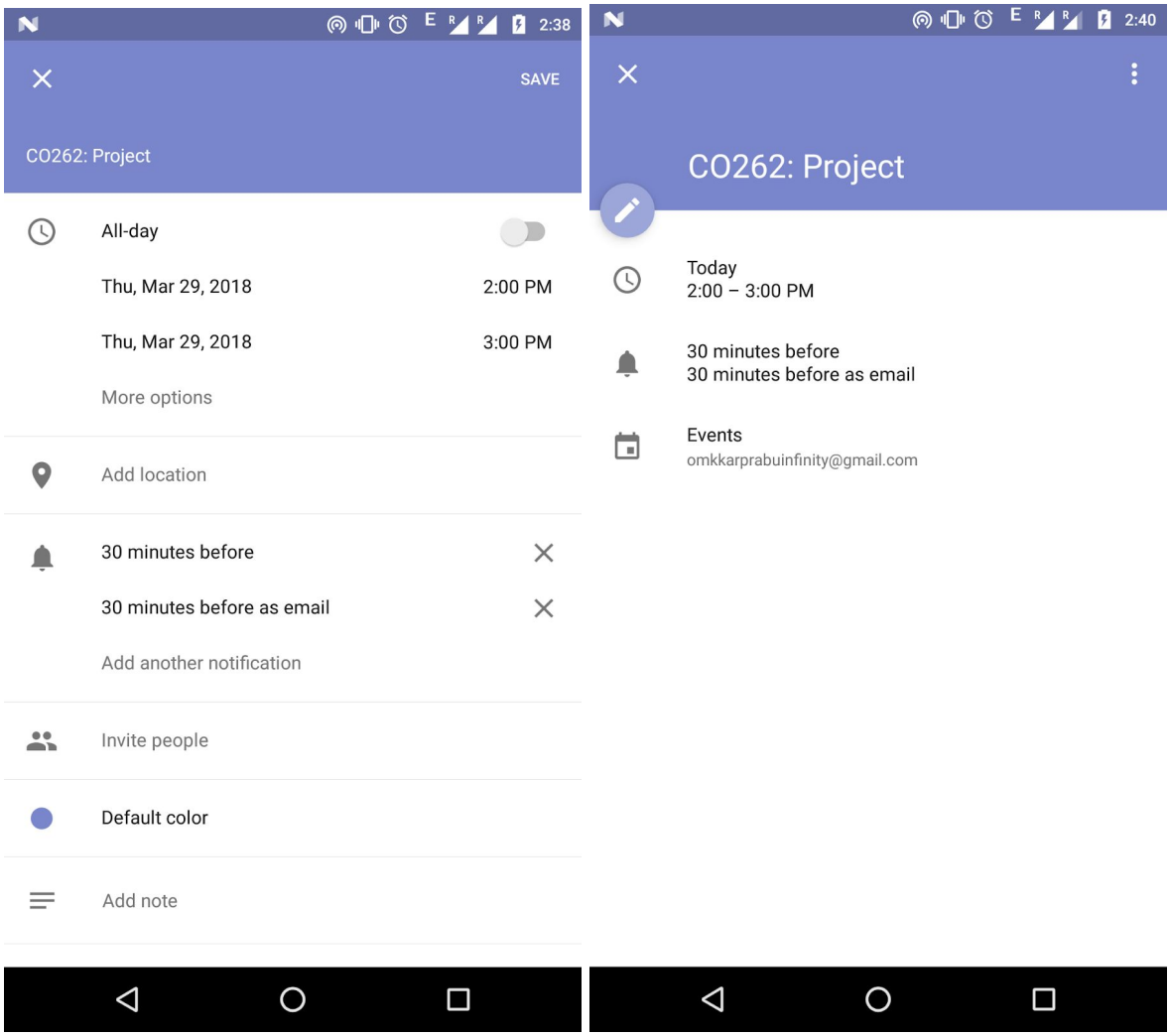
FR-1.5 Add Marks



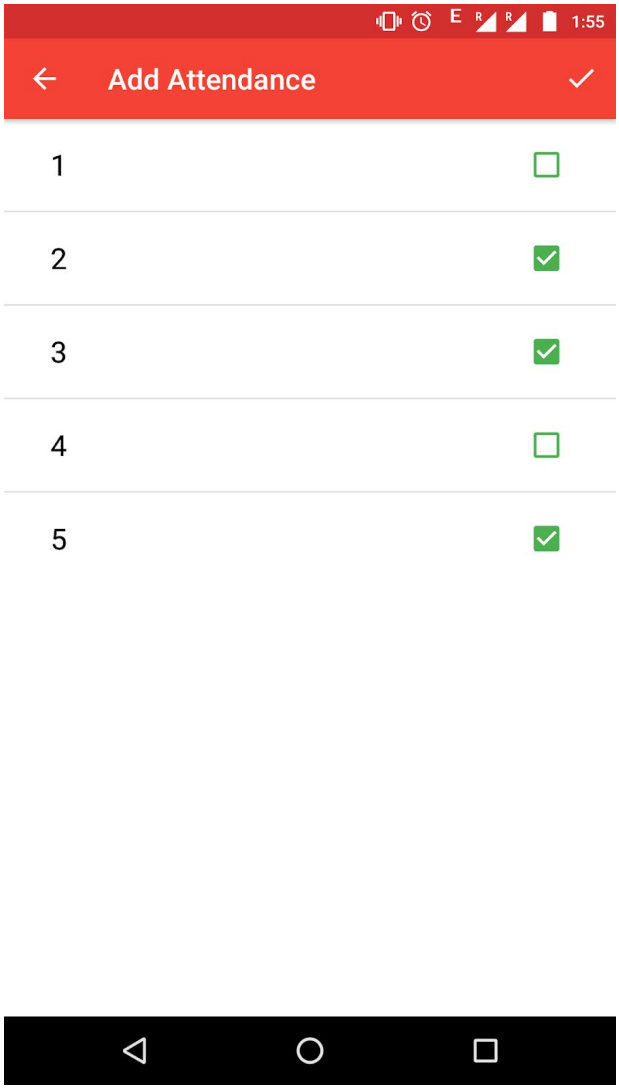
FR-1.6 Add Project Deadlines

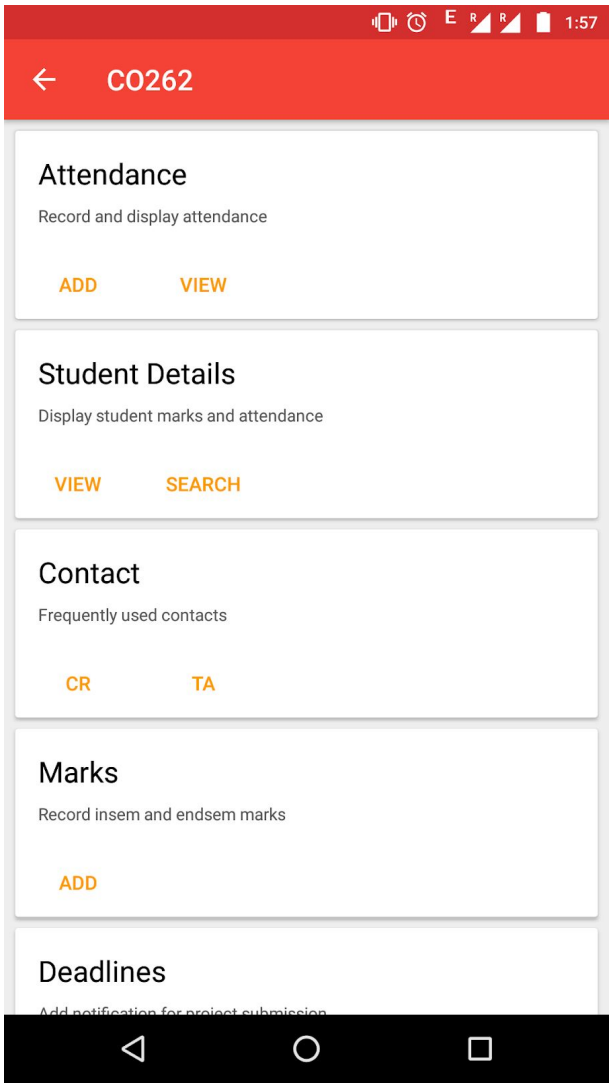


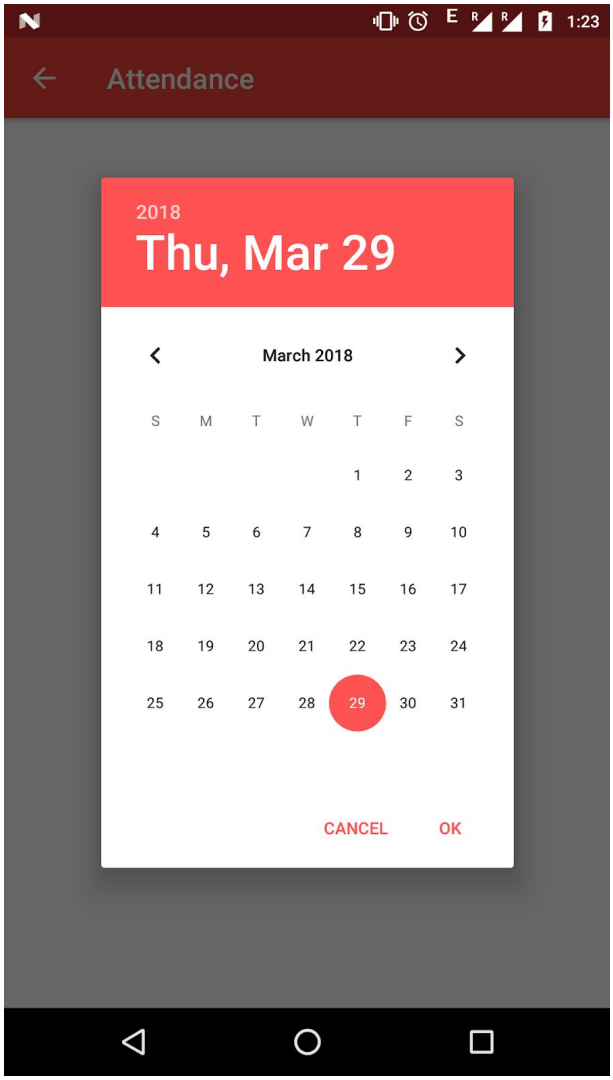


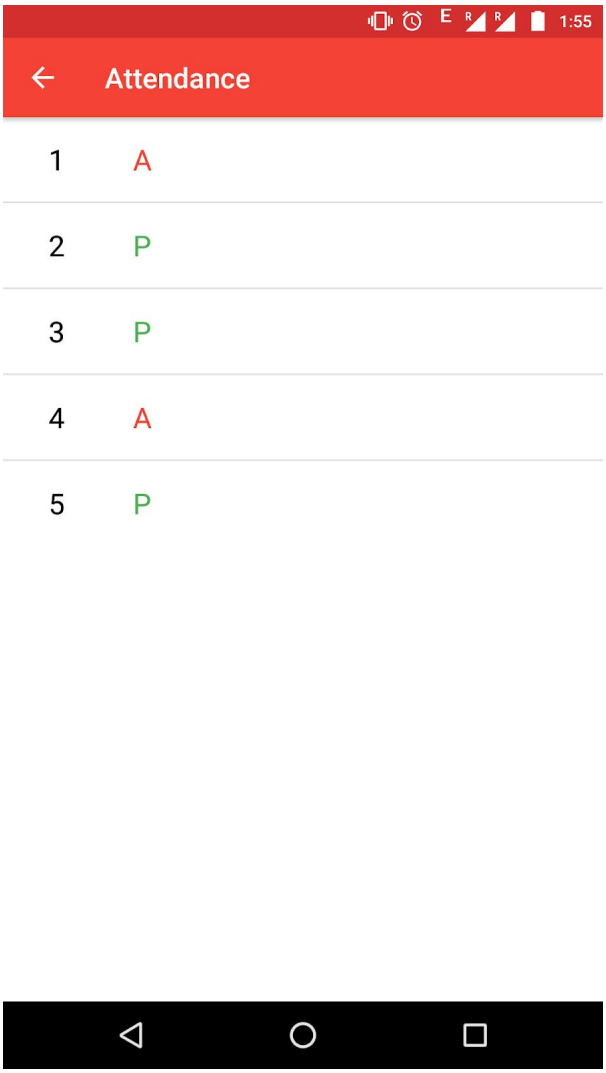

2:38
SAVE
CO262: Project
All-day
Thu, Mar 29, 2018
2:00 PM
Thu, Mar 29, 2018
3:00 PM
More options
Add location
30 minutes before
30 minutes before as email
Add another notification
Invite people
Default color
Add note
2:40
CO262: Project
Today
2:00 – 3:00 PM
30 minutes before
30 minutes before as email
Events
omkkarprabuinfinity@gmail.com

FR-1.10 View Attendance for a Day

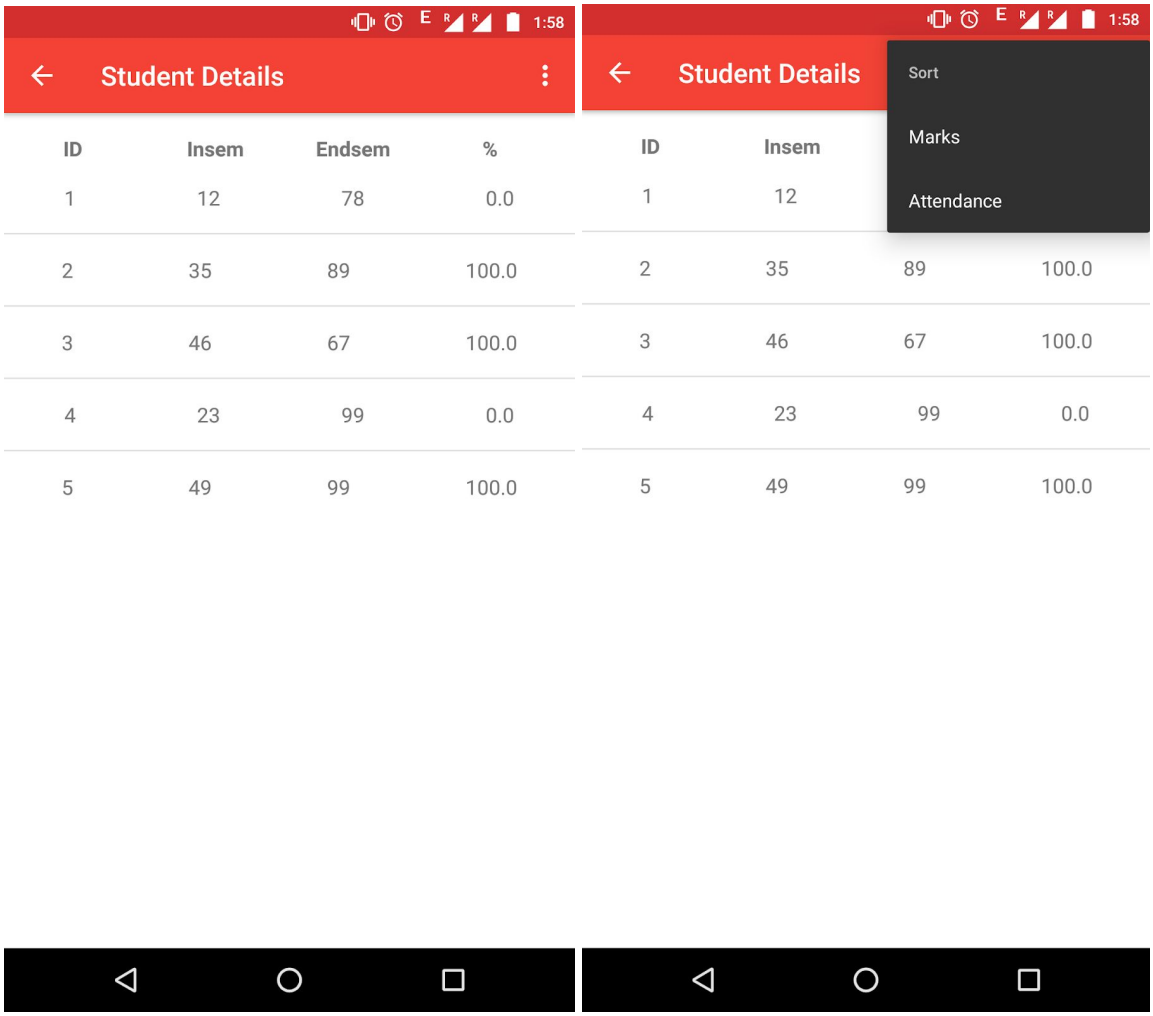





1:23
Attendance
2018
Thu, Mar 29
March 2018
S M T W T F S
1 2 3
4 5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
CANCEL
OK

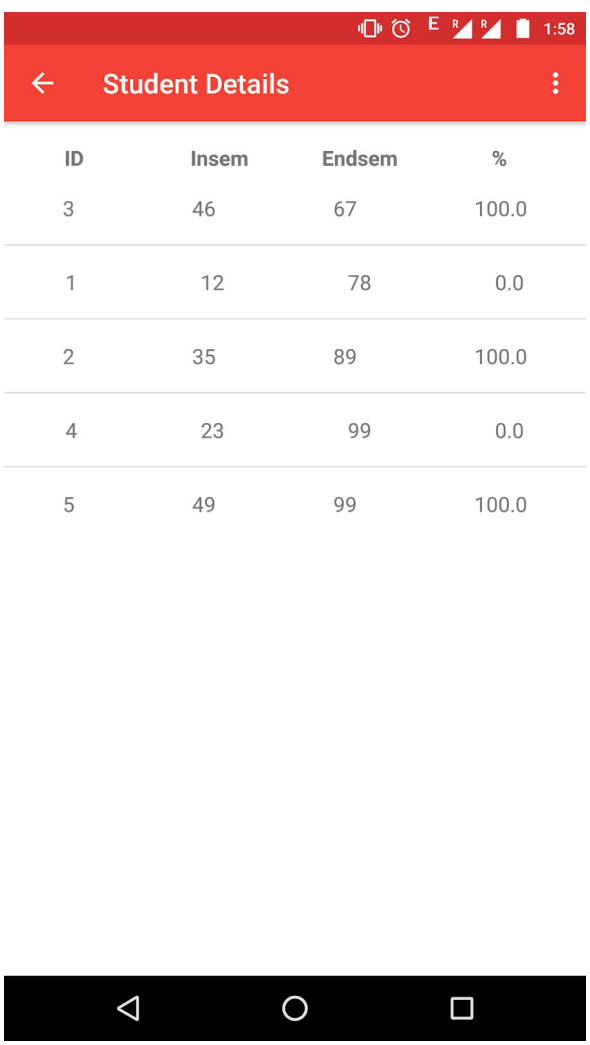

1:55
Attendance
1 A
2 P
3 P
4 A
5 P

FR-1.7 Display Marks and Attendance Percentage

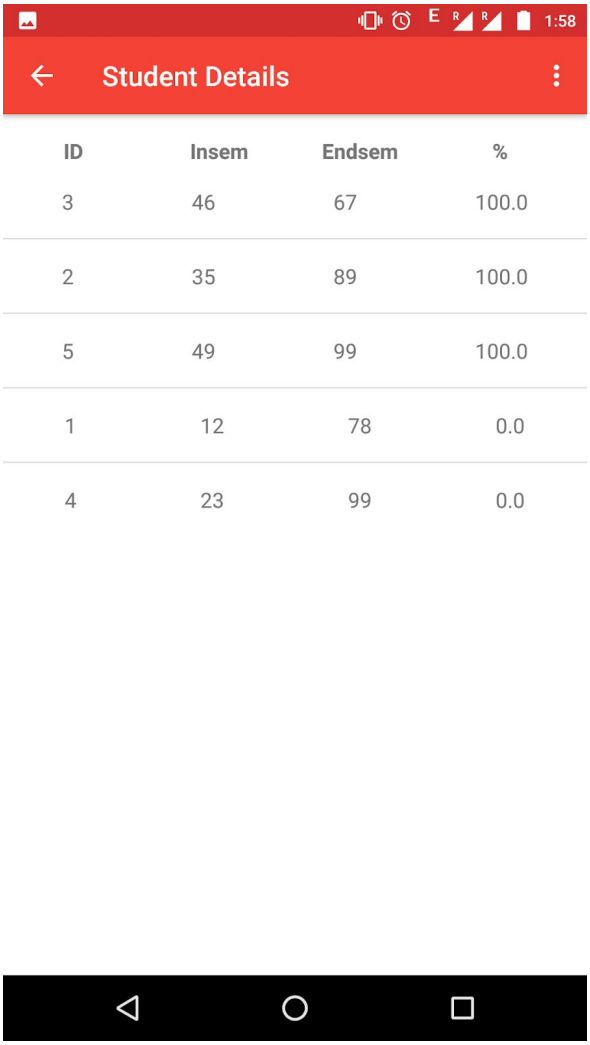


FR-1.9 Sort in Student List

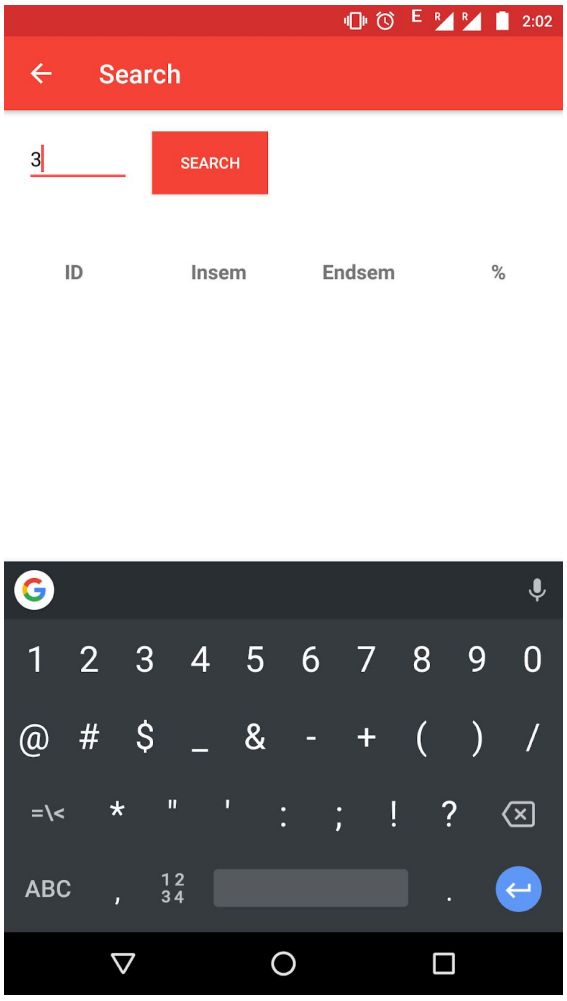
1) BY ENDSEM MARKS

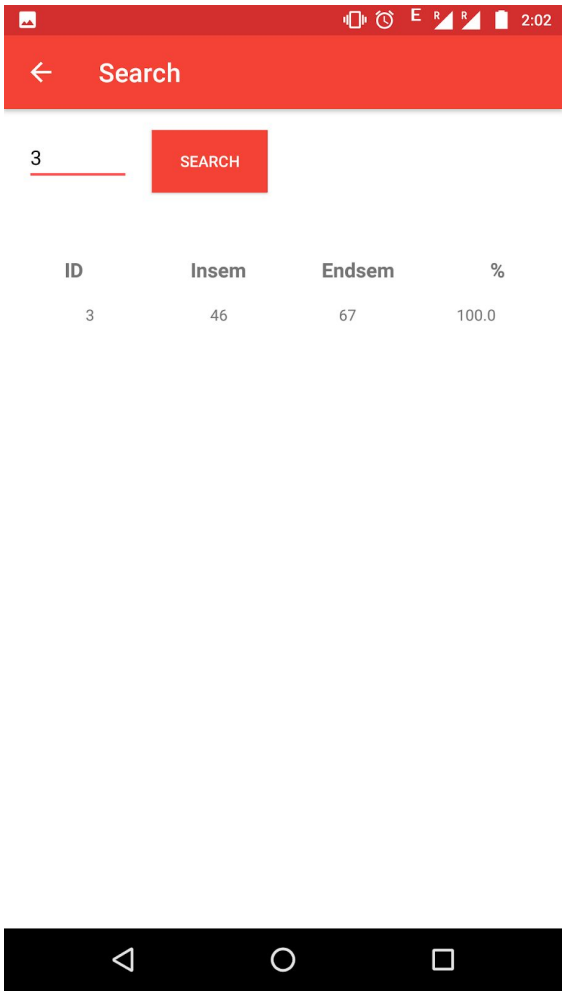


2) BY ATTENDANCE %

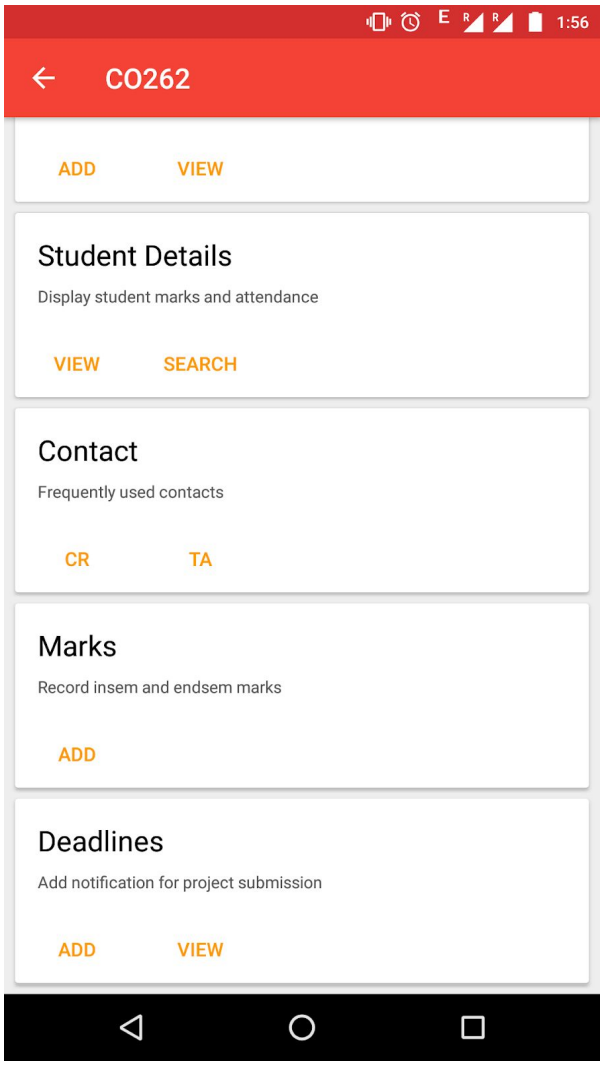


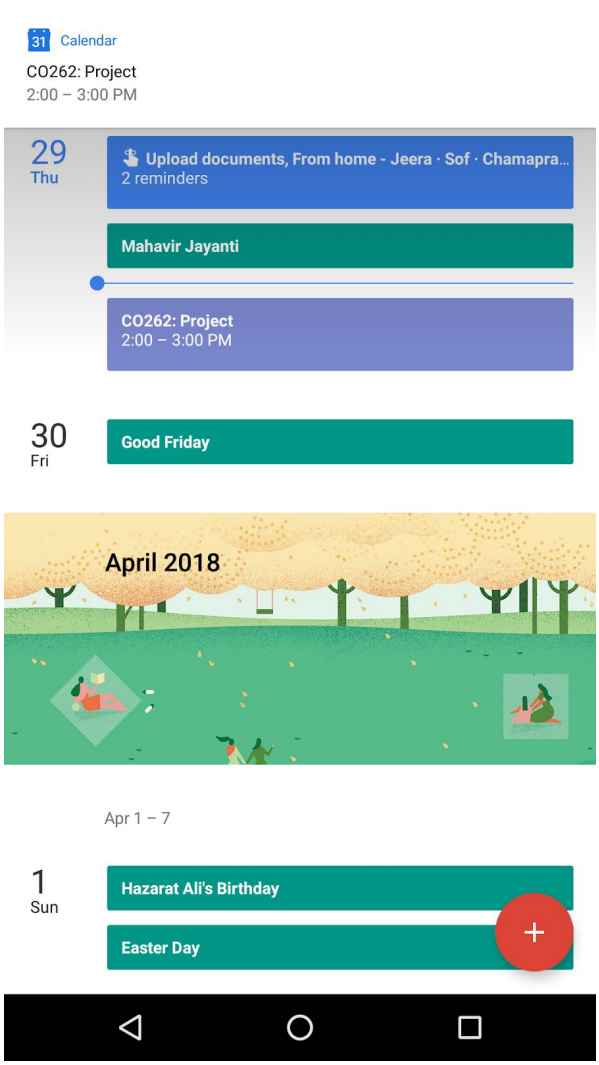
FR-1.8 Search By Student



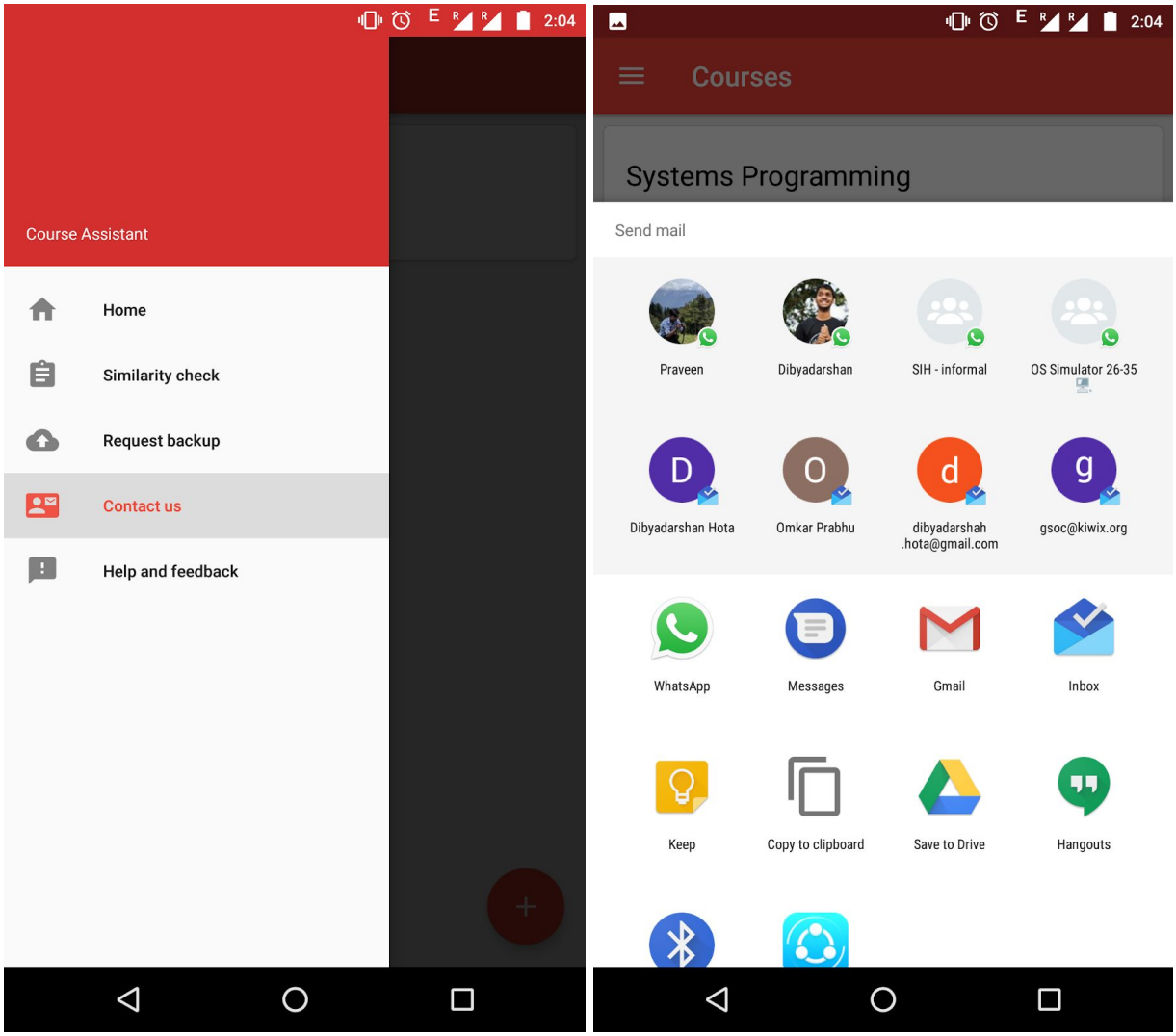


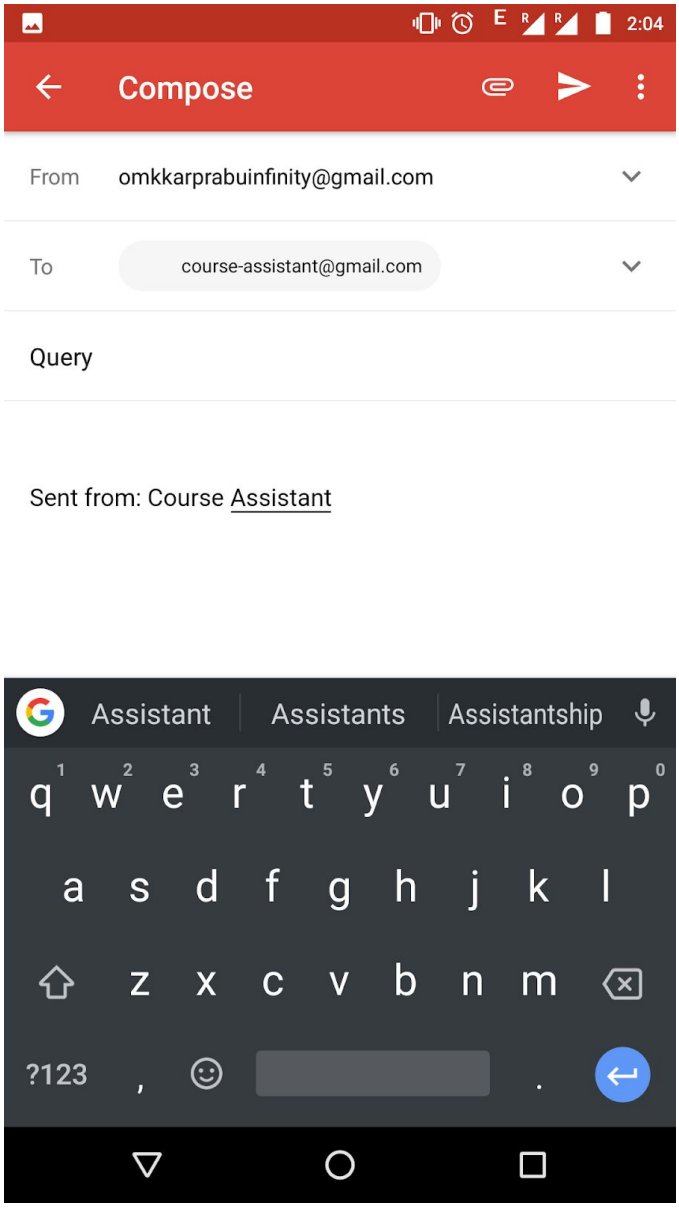
FR-1.14 View Project Deadlines





FR-1.15 Contact Us




2:04
Compose
From
omkkarprabuinfinity@gmail.com
To
course-assistant@gmail.com
Query
Sent from: Course Assistant
Assistant
Assistants
Assistantship
?123

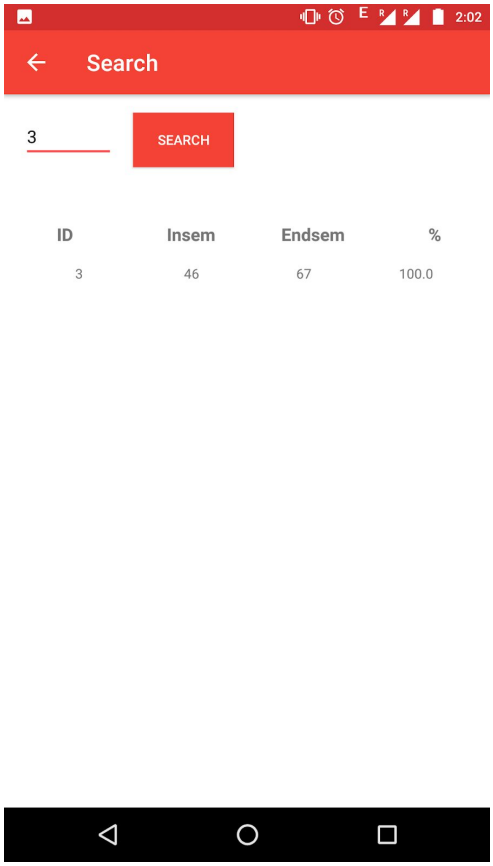
5. Non Functional Requirements Considered

NFR	Name	Description
1. Performance Requirements	Prominent search feature	The search feature should be prominent and easy to find for the user.
	System dependability	The fault tolerance of the system. If the system loses the connection to the Internet or the system gets some strange input, the user should be informed
	Document Similarity Check	Document similarity check can help teachers compare two submissions and calculate the degree of similarity between two documents. This feature uses a highly efficient algorithm which is a variation of cosine edit distance.
2. Safety Requirements	Database Access	The submitted attendance records would be locked to avoid accidental modification by someone else. There would be a backup created which can be used in case of accidental deletion of the database.
3. Security Requirements	Application and Database Access	There would be email and password verification to ensure that not everyone can open the application and make modifications to the database. Cloud backup service should also provide verification while updation to be done in the cloud database for the user.
4. Software System Attributes	System Availability	The application and backup features should be available 24/7.
	Internet Connection	The application should be connected to the Internet for initial login and feedback delivery.
5. Maintainability	Application extensibility	The application should be easy to extend. The code should be written in a way that it favors implementation of new functions.
	Application testability	Test environments should be built for the application to allow testing of the applications different functions.

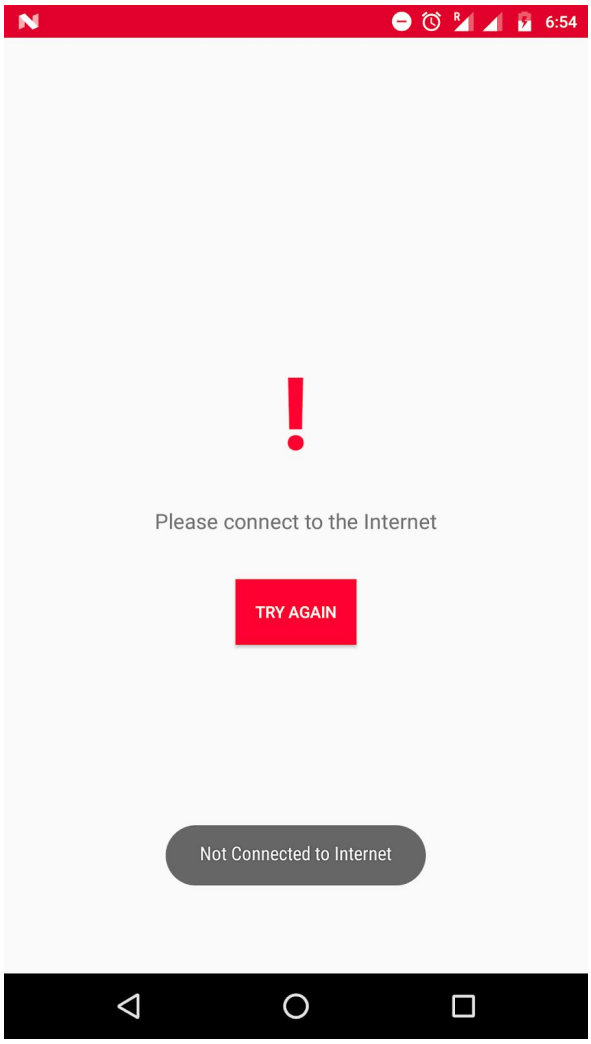
6. Portability		The application should be portable with Android. The adaptable platform for the application to run on

6. Screenshots of Non Functional Requirements

NFR-1.1 Prominent search feature

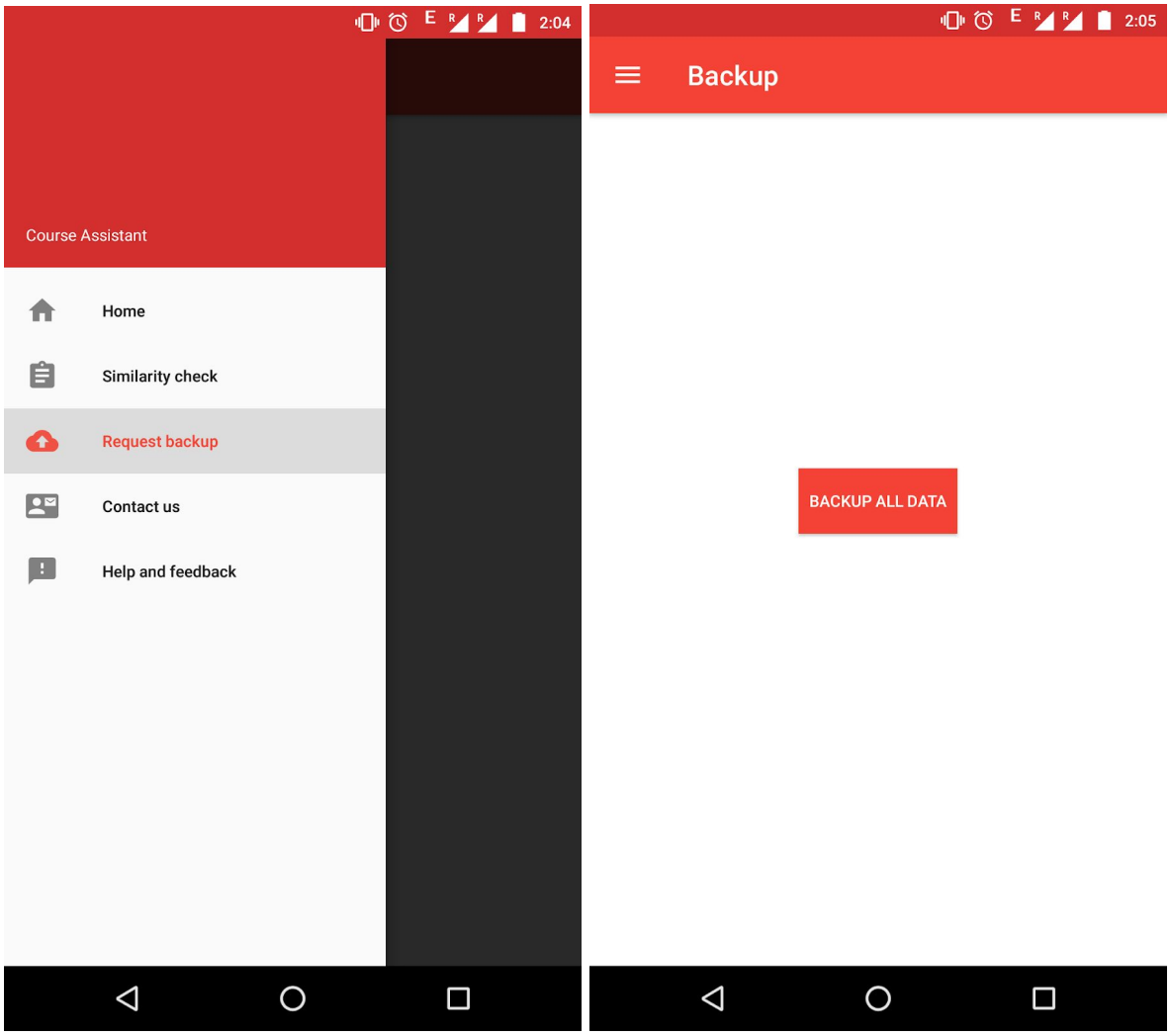


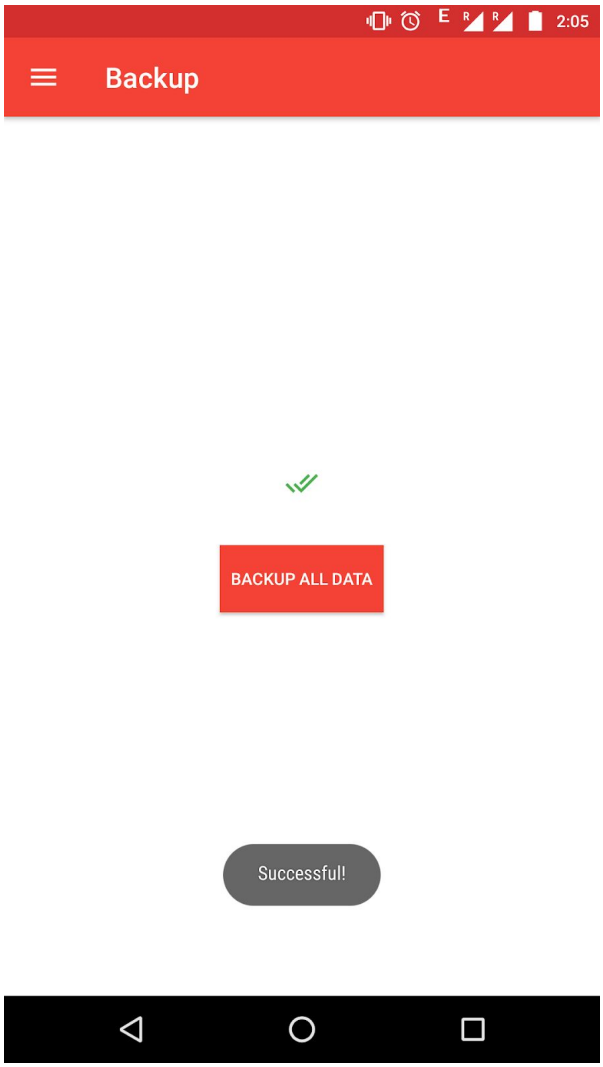
NFR-1.2 System Dependability

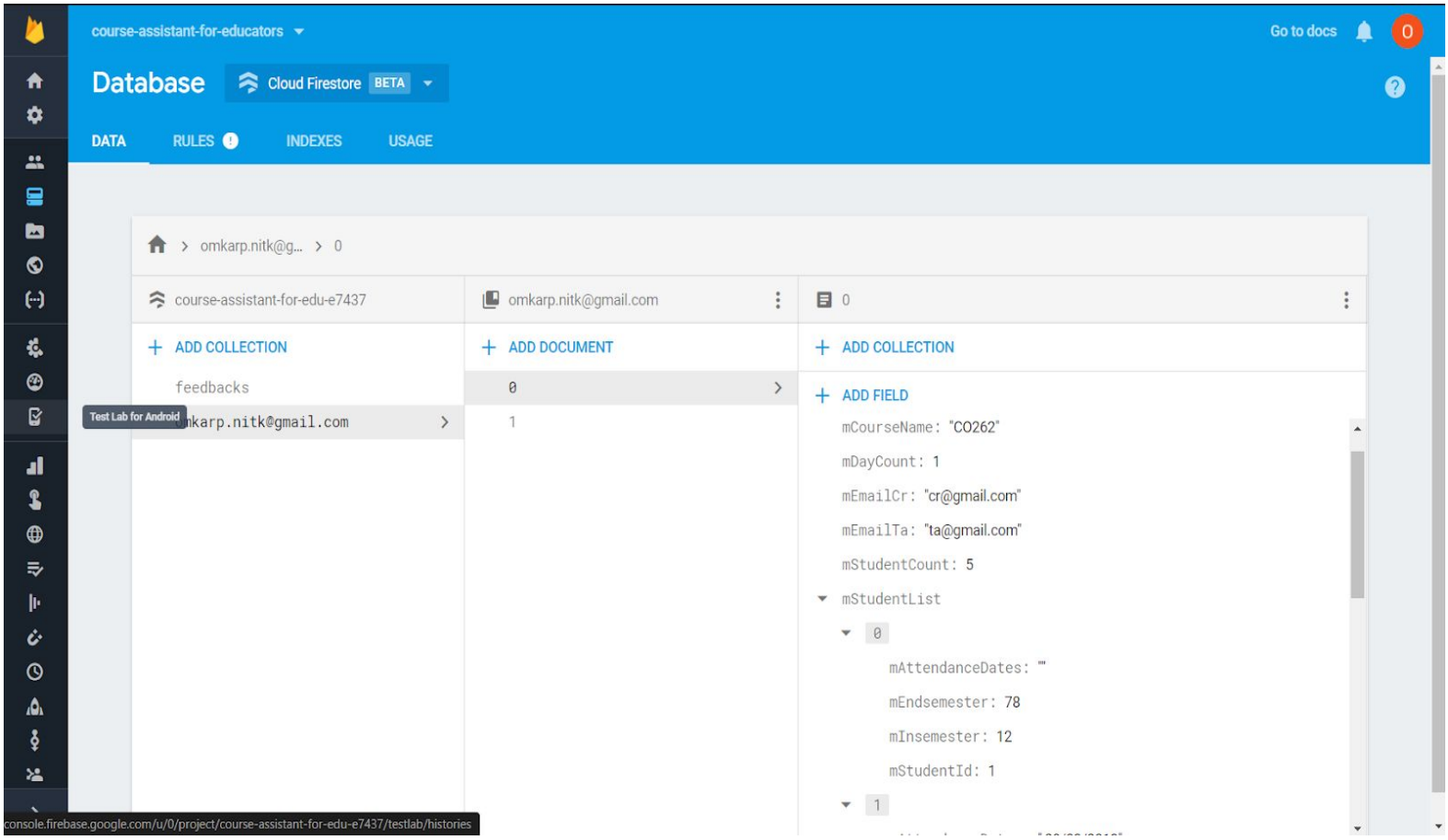


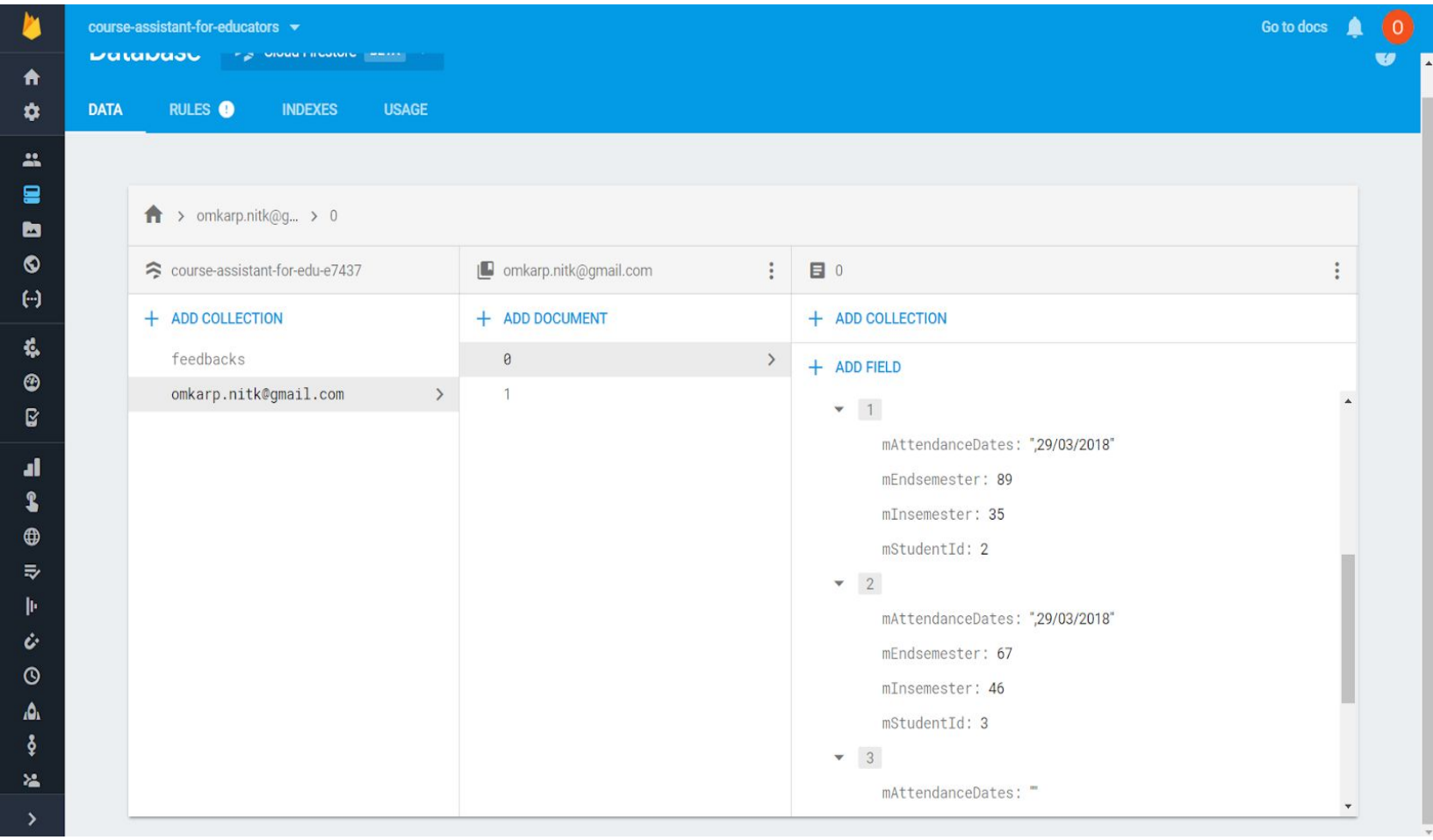
NFR-2.1 Database Access

Backup all local data to Firestore



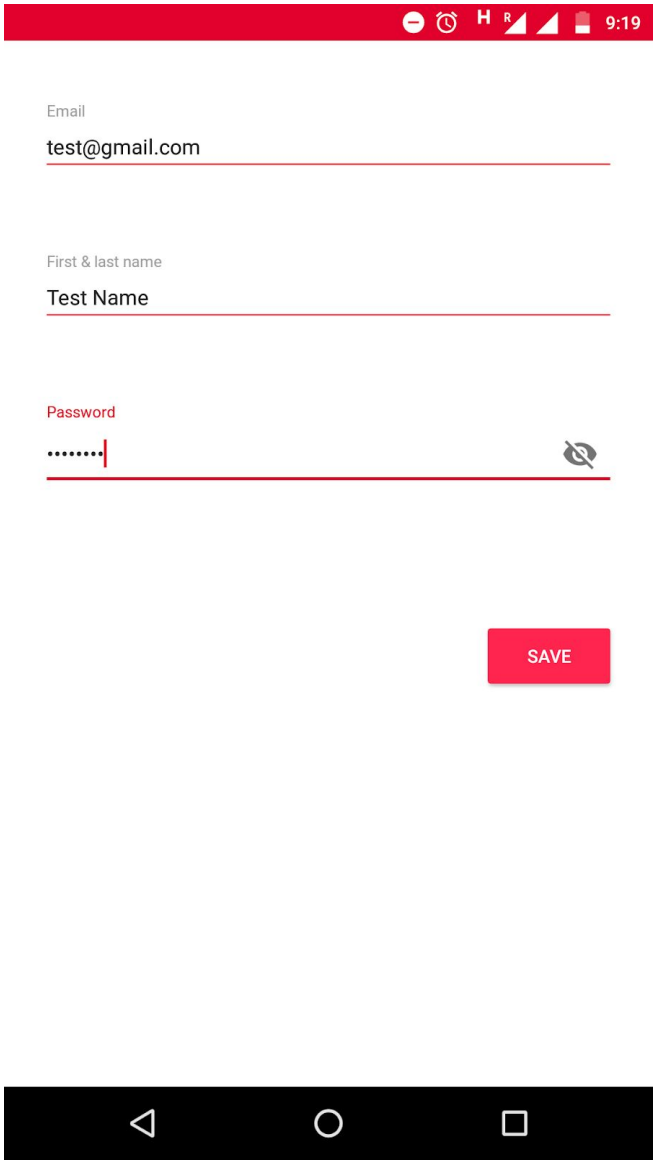

2:05
Backup
BACKUP ALL DATA
Successful!


course-assistant-for-educators
Go to docs
Database
Cloud Firestore BETA
DATA
RULES
INDEXES
USAGE
omkarp.nitk@g... > 0
course-assistant-for-edu-e7437
omkarp.nitk@gmail.com
0
ADD COLLECTION
ADD DOCUMENT
ADD COLLECTION
feedbacks
0
1
ADD FIELD
Test Lab for Android
mCourseName: "CO262"
mDayCount: 1
mEmailCr: "cr@gmail.com"
mEmailTa: "ta@gmail.com"
mStudentCount: 5
mStudentList
0
mAttendanceDates: ""
mEndsemester: 78
mInsemester: 12
mStudentId: 1
1
console.firebase.google.com/u/0/project/course-assistant-for-edu-e7437/testlab/histories


course-assistant-for-educators
Go to docs
DATA
RULES
INDEXES
USAGE
omkarp.nitk@g...
0
course-assistant-for-edu-e7437
omkarp.nitk@gmail.com
0
ADD COLLECTION
ADD DOCUMENT
ADD COLLECTION
feedbacks
omkarp.nitk@gmail.com
0
1
ADD FIELD
1
mAttendanceDates: ",29/03/2018"
mEndsemester: 89
mInsemester: 35
mStudentId: 2
2
mAttendanceDates: ",29/03/2018"
mEndsemester: 67
mInsemester: 46
mStudentId: 3
3
mAttendanceDates: ""

NFR-3.1 Application and Database Access

Login used at the first phase will be used to ensure that user will login every time before starting the application.



NFR-5.1 Application extendability

Codes which can be viewed on GitHub are modular with proper naming that is easily expandable that helps to expand it even if anyone is unfamiliar with the code base

7. Summary of Test Plan

1. Tests were run by Travis CI whenever new code was pushed into the repository.
2. Type of testing to be performed:
    a. Validate all the input fields.
    b. Check if there is any loss of information while creating a backup of the local database.
    c. Check for constraints on database fields.
    d. Look for minor UI corrections like addition of ScrollView etc.
3. Majority of the functional requirements were validated during the development phase itself. Functional requirements yet to be checked are listed below, the ones which are not listed have been validated already:
    a. FR1: Check for compatibility across various devices
    b. FR3: Validate the inputs
    c. FR5: Apply constraints
    d. FR8: Check for presence of student ID
    e. FR12: Improve the efficiency of the algorithm
4. Few non-functional requirements are yet to be checked:
    a. Minor changes like making member functions and variables private or protected depending on their use, adding fragments wherever necessary etc must done to ensure that the code is readable and the data is not modified accidentally.
    b. Check if there is any loss of information while creating a backup of the local database.
    c. Minor UI fixes shall make the software easy to use. The customers can use the application seamlessly after these tests.
5. The outcome of this would be a bug-free software.

8. Summary

All of the functional and non-functional requirements have been implemented successfully. The final task left is to check the code once to ensure that there aren't any bugs. The code has been written in a highly modular fashion.

Continuous Integration using Travis CI has been followed throughout the project development with additional services from GitHub for project management and collaboration like milestones, issue tracker, project board, etc.

.