CENG-322
Deliverable 5

Team Name: TBD

Project Name: Smart Library Study Room Management and Comfort System

Group: 2

Members:
Mathew Anderson-Saavedra N01436706
Medi Muamba Nzambi N01320883
Safah Virk N01596470

Table Of Contents

Members Info And Participation

Name	Student ID	Github ID	Signature	Effort
Medi Muamba Nzambi	N01320883	MediMuamba0883	Signed by MediMuamba	80
Mathew Anderson-Saavedra	N01436706	MathewAnderson6706	Signed by Mathew A	100
Safah Virk	N01596470	Safahvirk6470	Signed by Safah Virk	80

Project Scope and Goals

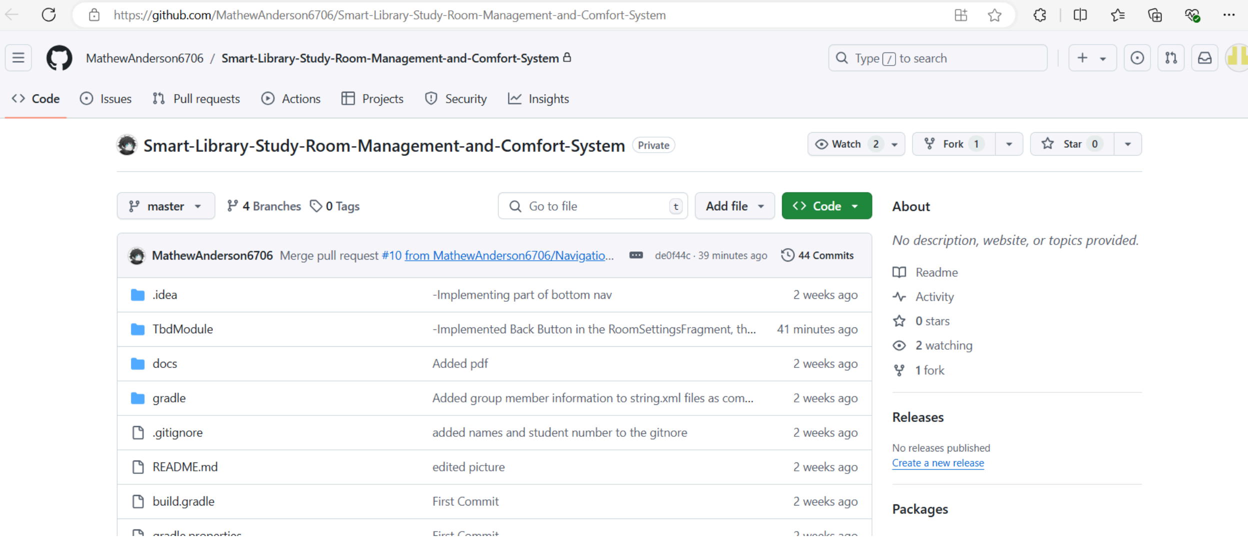
The Scope of this project is to create an easy-to-understand UI, viewing different buildings and accessing different rooms with a variety of different settings to change. The objects in the app will be constantly updated by the database so the user can have the most up-to-date information about the availability of rooms, and different information on the rooms themselves. The goal of this project is to make a well developed and designed app to show off our skills and maybe get this implemented in colleges.

GitHub Repo Link and Strategy

GitHub Repo Link:

https://github.com/MathewAnderson6706/Smart-Library-Study-Room-Management-and-Comfort-System.git

Verify the link is working



https://github.com/MathewAnderson6706/Smart-Library-Study-Room-Management-and-Comfort-System

MathewAnderson6706 / Smart-Library-Study-Room-Management-and-Comfort-System

Smart-Library-Study-Room-Management-and-Comfort-System Private

4 Branches 0 Tags

MathewAnderson6706 Merge pull request #10 from MathewAnderson6706/Navigatio... de0f44c · 39 minutes ago 44 Commits

.idea	-Implementing part of bottom nav	2 weeks ago
TbdModule	-Implemented Back Button in the RoomSettingsFragment, th...	41 minutes ago
docs	Added pdf	2 weeks ago
gradle	Added group member information to string.xml files as com...	2 weeks ago
.gitignore	added names and student number to the gitnore	2 weeks ago
README.md	edited picture	2 weeks ago
build.gradle	First Commit	2 weeks ago
gradle.properties	First Commit	2 weeks ago

About

No description, website, or topics provided.

0 stars

2 watching

1 fork

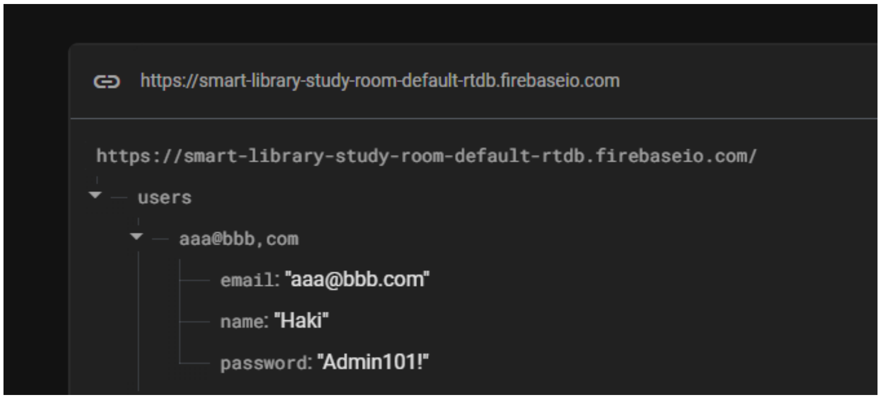
Releases

No releases published
Create a new release

Packages

Login Functionality & Database

Login functionality, I will use the following credentials to test your app: Email: aaa@bbb.com Password: Admin101!
Document any other login credentials I must use i.e. Admin vs. regular user. I should not need to send you an email to login to your app.



```
https://smart-library-study-room-default-rtdb.firebaseio.com

https://smart-library-study-room-default-rtdb.firebaseio.com/
▾ — users
    ▾ — aaa@bbb,com
          email: "aaa@bbb.com"
          name: "Haki"
          password: "Admin101!"
```

Sprint Goals, Description, and Work Completed

17. Sprint 5 goals, list sprint goals and not tasks you worked on.
-Complete the app
-Get all testing done
-Finish everything related with the database
-Polish up the app

18. Describe 3 tasks from the Scrum dashboard that are related
to addressing technical debt.

1. Refactored the Registration Activity to make it look cleaner. That includes making a separate class to contain some of the methods. We made the InputValidator class.
2. Refactored the feedback screen where instead of it being 24hrs per device, it would be 24hrs per user
3. Refactored the feedback screen to use regex, making use of the InputValidator class

19. Describe in detail, the work that has been completed by
each team member in this sprint only.

Mathew:
-Updated Registration Activity to now take phone number input and register it into the database
-Used REGEX in Registration Activity so user can not just put whatever they want
-Added room light object to database, app can read/write to it.
-Implemented google sign in
-Fixed bug with remember me box, as if the user did not press the box, their info will not be saved in user profile fragment
-Refactored classes to make up for technical debt, looks cleaner
-Created 2 Junit test classes
-Added REGEX to feedbackfragment
-Timer is now added to roomsettings, starts when user enters, when it hits 0, will kick user out of room, make room vacant, and display a snackbar letting the user know they have been kicked.
-If a user clicks on a vacant room, it will display a toast letting them know that they have to be in the room to enter in a code.
-Feedback is now restricted per user instead of per device
-Attempted Notifications
-Some refactoring for feedbackFragment

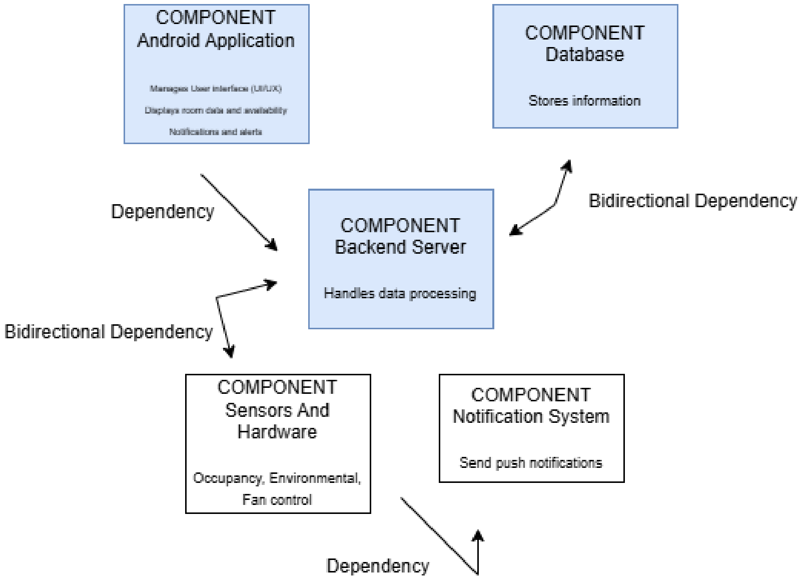
Medi:
-Created Junit test for LoginActivity
-Added a progress bar displaying for the log in page.
-Toast displays in case of database error.

-Added autofillhints in login xml file
-Added FAB
-Added an image to the FAB
-Originally copy and pasted code in each fragment to have FAB, than refactored and made a class to deal with FAB
-Implemented Offline mode
-Refactored classes by removing unused import statements

Safah:
-Name and email shows in naviagtion drawer
-Fixed hardcoding
-Worked on feedback fragment originally by clearing user input once the form is submitted, and restricting the device to once submission every 24hrs
-Once form was submitted, greyed out the submit button, than displayed a timer showing how many hours are remaining
-Added user device from feedback to database
-Showed progress bar when submitting the feedback
-Removed unused imports

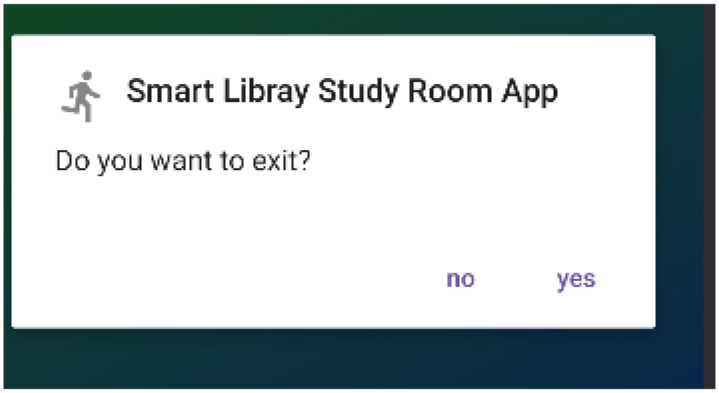
20. Using C4 Model, draw “Component Diagram”. Choose two of the containers in your system. Draw two detailed components Diagram.

21. Use a tool to draw your diagrams, hand drawing will not be accepted. I suggest you use https://app.diagrams.net/



22. Demonstrate the use of all of the following in your app, in the appropriate places: Include screenshots to show.

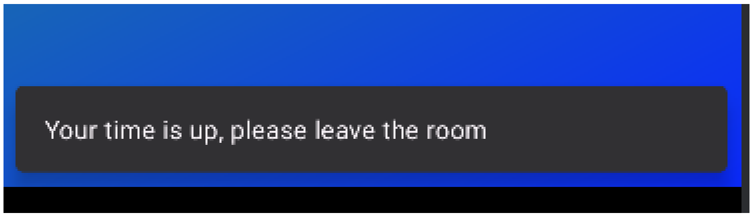
A. AlertDialog



Smart Libray Study Room App

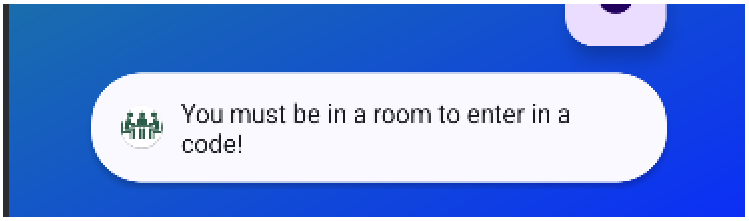
Do you want to exit?

B. Snackbar



Your time is up, please leave the room

C. Toast

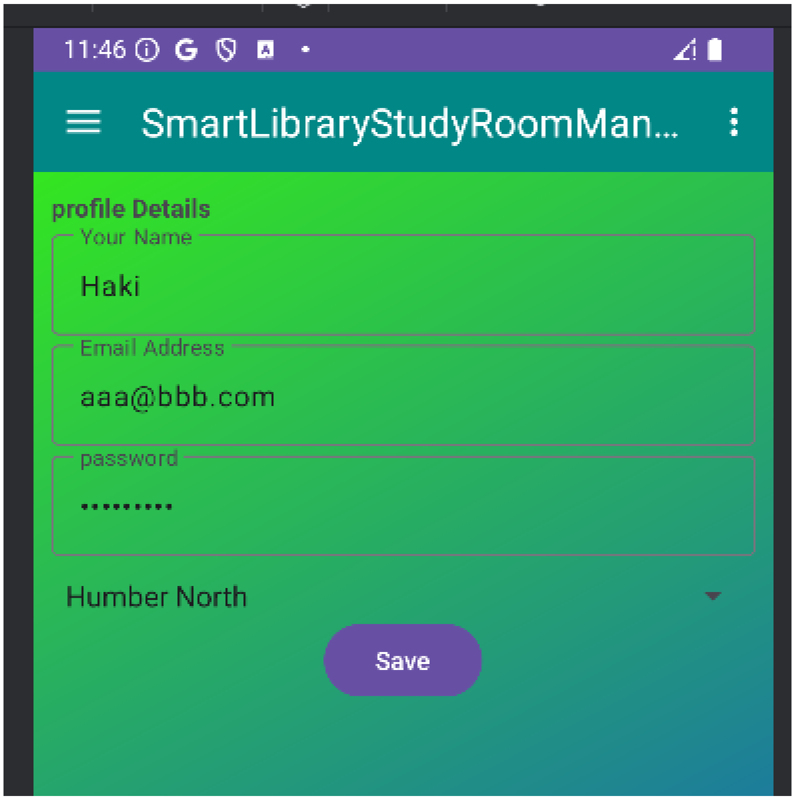


You must be in a room to enter in a code!

D. Notification, handle runtime for API 33 and higher.

23. Screenshot showing the fab button and how the functionality fits into your app.

24. Should implement some functionality for off-line mode. Document what feature will work off-line. Should be meaningful feature. Will be tested, when in Airplane mode. Provide Screenshot.



profile Details

Your Name: Haki

Email Address: aaa@bbb.com

password: •••••••••

Humber North

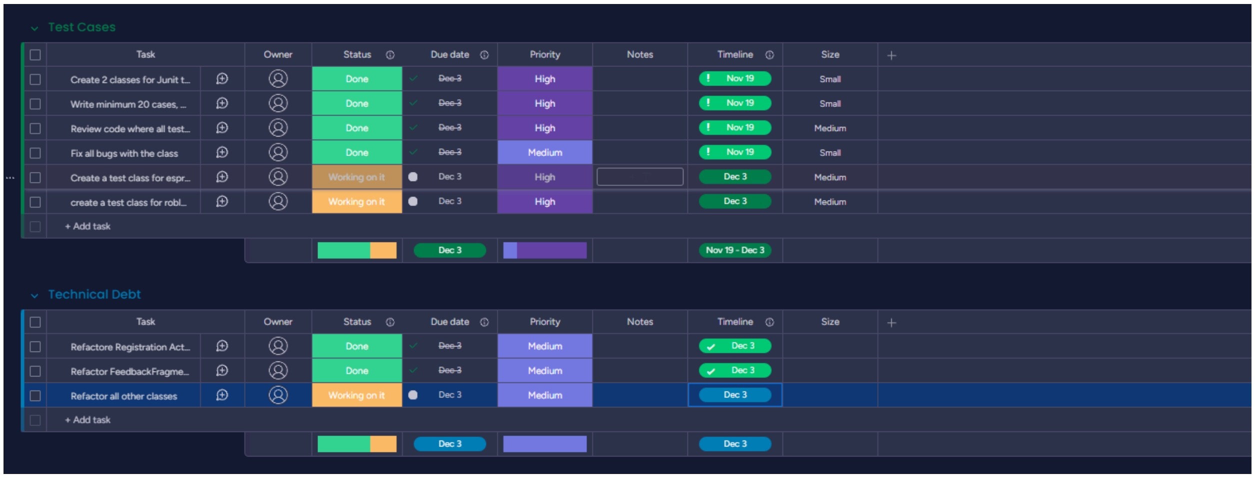
Example on how to document offline feature, don't use the same
Example:

25. Describe how did you handle exceptions, and list the exceptions that you had to handle.

Exceptions were handled using **try-catch** blocks for Google Sign-In errors and Firebase database issues. Input validation errors were managed by displaying error messages. This approach ensures smooth user experience by catching and reporting errors gracefully.

26. Work on the feedback provided in the previous deliverables.

27. Complete Scrum dashboard, with all stories and tasks. Take screenshots of the last sprint only.



Test Cases

□	Task	Owner	Status	Due date	Priority	Notes	Timeline	Size
□	Create 2 classes for Junit t...		Done	✓ ~~Dec 3~~	High		! Nov 19	Small
□	Write minimum 20 cases, ...		Done	✓ ~~Dec 3~~	High		! Nov 19	Small
□	Review code where all test...		Done	✓ ~~Dec 3~~	High		! Nov 19	Medium
□	Fix all bugs with the class		Done	✓ ~~Dec 3~~	Medium		! Nov 19	Small
□	Create a test class for espr...		Working on it	● Dec 3	High		Dec 3	Medium
□	create a test class for robl...		Working on it	● Dec 3	High		Dec 3	Medium
□	+ Add task							
				Dec 3			Nov 19 - Dec 3	

Technical Debt

□	Task	Owner	Status	Due date	Priority	Notes	Timeline	Size
□	Refactore Registration Act...		Done	✓ ~~Dec 3~~	Medium		✓ Dec 3	
□	Refactor FeedbackFragme...		Done	✓ ~~Dec 3~~	Medium		✓ Dec 3	
□	Refactor all other classes		Working on it	● Dec 3	Medium		Dec 3	
□	+ Add task							
				Dec 3			Dec 3	

28. Post-Mortem, Project Review Meeting, document the below:

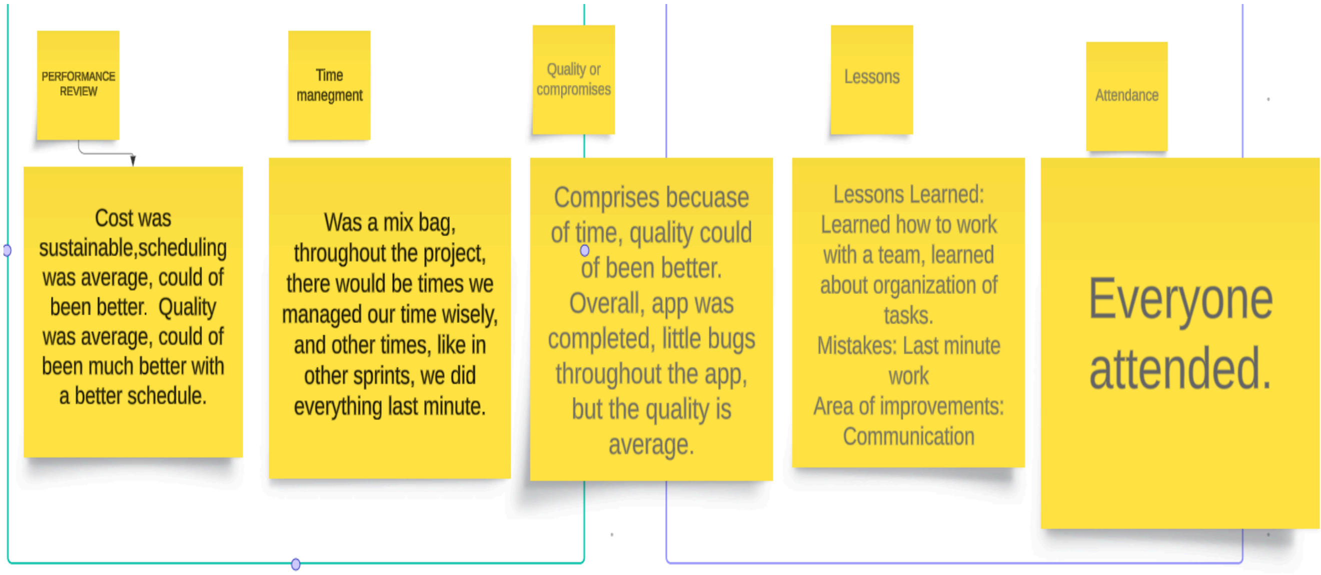
A. Begin your post-mortem, conduct a performance review of the project. In other words, calculate the project's performance in terms of cost, schedule, and quality.

B. Did the team members involved manage their time wisely? Or everything was done last minute.

C. Were there issues with the quality or compromises along the way?

D. Lessons learned, mistakes, and area of improvements.

E. Who attended and who missed the meeting.



PERFORMANCE REVIEW

Cost was sustainable,scheduling was average, could of been better. Quality was average, could of been much better with a better schedule.

Time manegment

Was a mix bag, throughout the project, there would be times we managed our time wisely, and other times, like in other sprints, we did everything last minute.

Quality or compromises

Comprises becuase of time, quality could of been better. Overall, app was completed, little bugs throughout the app, but the quality is average.

Lessons

Lessons Learned: Learned how to work with a team, learned about organization of tasks.
Mistakes: Last minute work
Area of improvements: Communication

Attendance

Everyone attended.

29. Use a tool to record your Project Review Meeting (i.e. https://lucidspark.com/landing/create/online-sticky-notes
https://miro.com/

30. How did you address technical debt in your project?
I minimized technical debt by keeping code clear, reusable, and well-commented, focusing on refactoring redundant parts and avoiding hardcoded values.

31. Document two area of refactoring and why you did it!. Real examples from your code and not just statements. Copy the code from your project and comment on it.

Database Reference Code
Before:
Repeated database reference code.
After:
Created a reusable method to get the reference::

```
private DatabaseReference getUserReference(String username) {
    return
FirebaseDatabase.getInstance().getReference(getString(R.string.users1)).child(username);
}
```

Progress Bar Visibility
Before:
Repeated visibility logic for the ProgressBar.
After:
Moved to a method for reusability:

```
private void setProgressBarVisibility(boolean isVisible) {
    progressBar.setVisibility(isVisible ? View.VISIBLE : View.GONE);
}
```

32. Describe how DevOps would have helped your project.?
Automation: Automated testing and deployment would speed up development and reduce manual errors.
CI/CD: Continuous integration and delivery ensure frequent updates and fast bug fixes.
Version Control: GitHub for version control improves collaboration and code management.
Monitoring: Tools like Prometheus would provide real-time app performance monitoring.

33. What coding standards did you use, and how did you use Them.

Naming Conventions:

- CamelCase for variables and methods (e.g., usernameInput, validateUsername).
- PascalCase for class names (e.g., LoginActivity).

Modular Code:

- Methods handle specific tasks (e.g., validateUsername(), checkUser()).

Error Handling:

- Proper error messages for user input and connection issues.

Comments:

- Used comments to explain key logic.

No Hardcoding:

- Strings and keys are stored in strings.xml and constants.

34. List at least one security vulnerability in your code, copy the code and comment on it. How to address in the future.

```
editor.putString(KEY_PASSWORD, password);
```

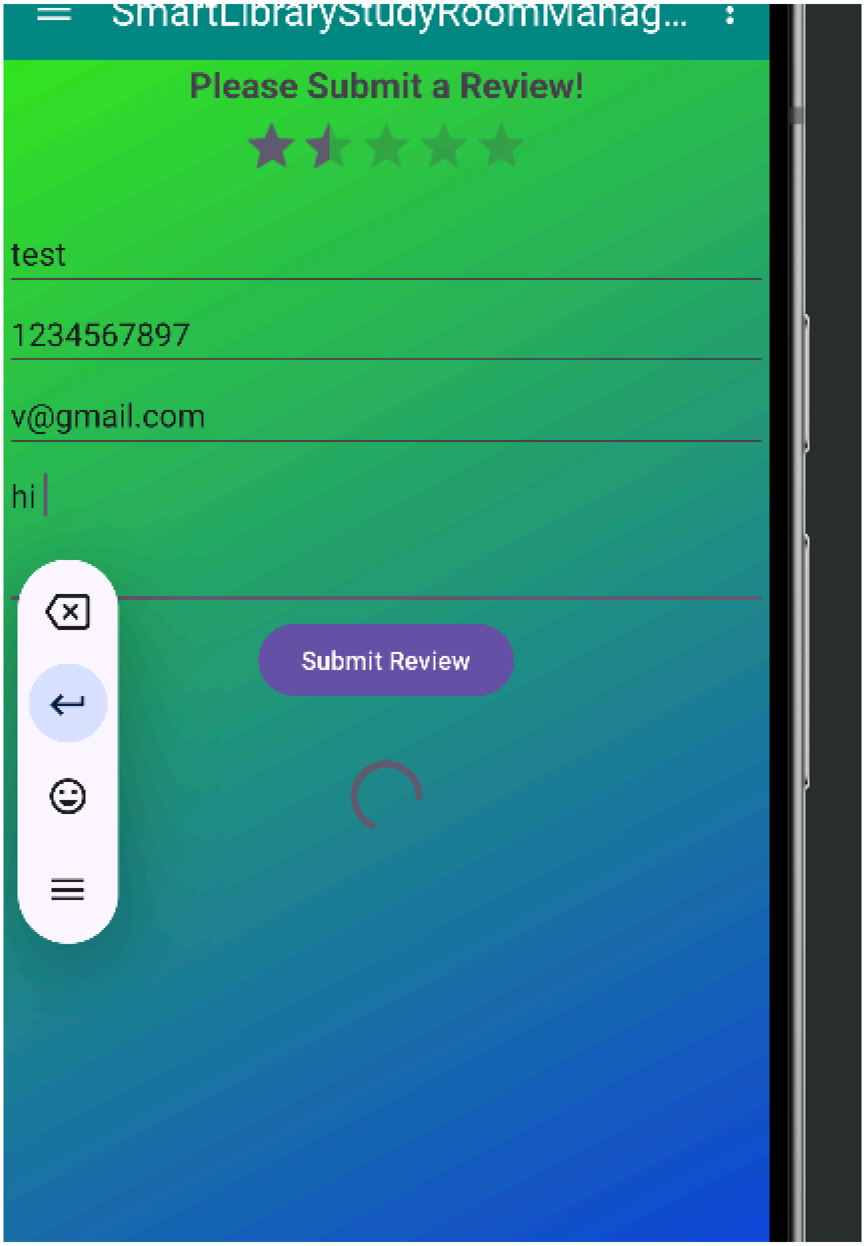
Issue: Storing passwords in **SharedPreferences** as plaintext is insecure.

Solution: Use **EncryptedSharedPreferences** or **Keystore** to store sensitive data securely. This ensures passwords are encrypted before storage.

Example:

```
SharedPreferences sharedPreferences = EncryptedSharedPreferences.create(
   "user_prefs", MasterKey.DEFAULT_MASTER_KEY_ALIAS, this,
   EncryptedSharedPreferences.PREF_KEY_ALIAS,
EncryptedSharedPreferences.PREF_VALUE_ALIAS);
```

35. Add into pdf file screenshot showing the progress bar while the form is getting submitted on the feedback screen.



Please Submit a Review!

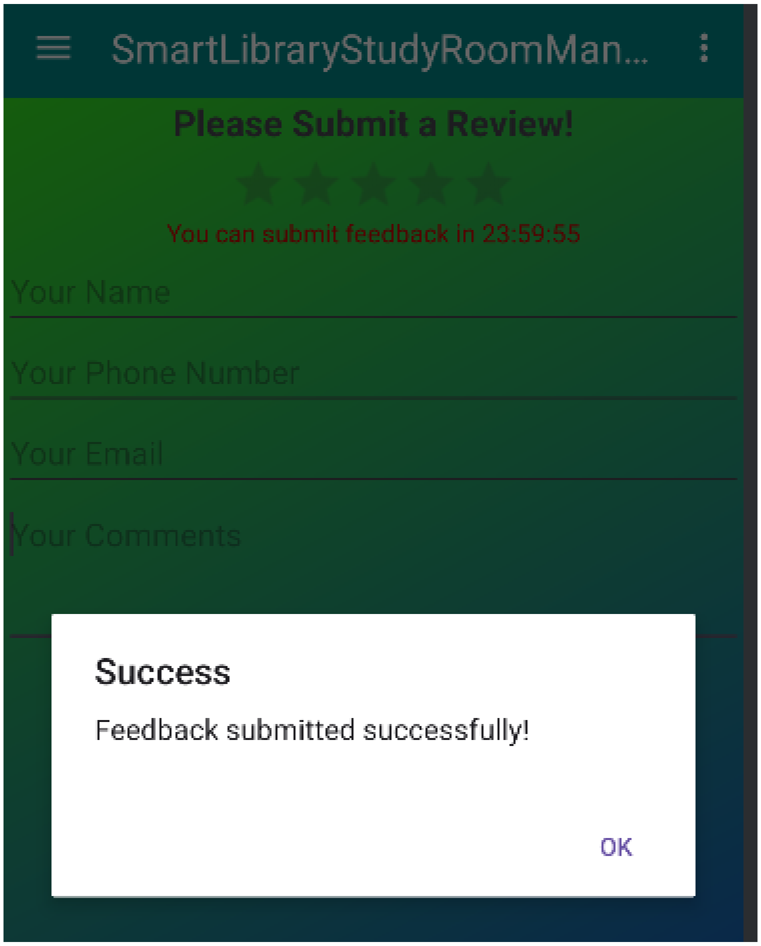
test

1234567897

v@gmail.com

hi

36. Add into pdf file screenshot showing the AlertDialog once the form is submitted successfully on the feedback screen.



Please Submit a Review!

You can submit feedback in 23:59:55

Your Name

Your Phone Number

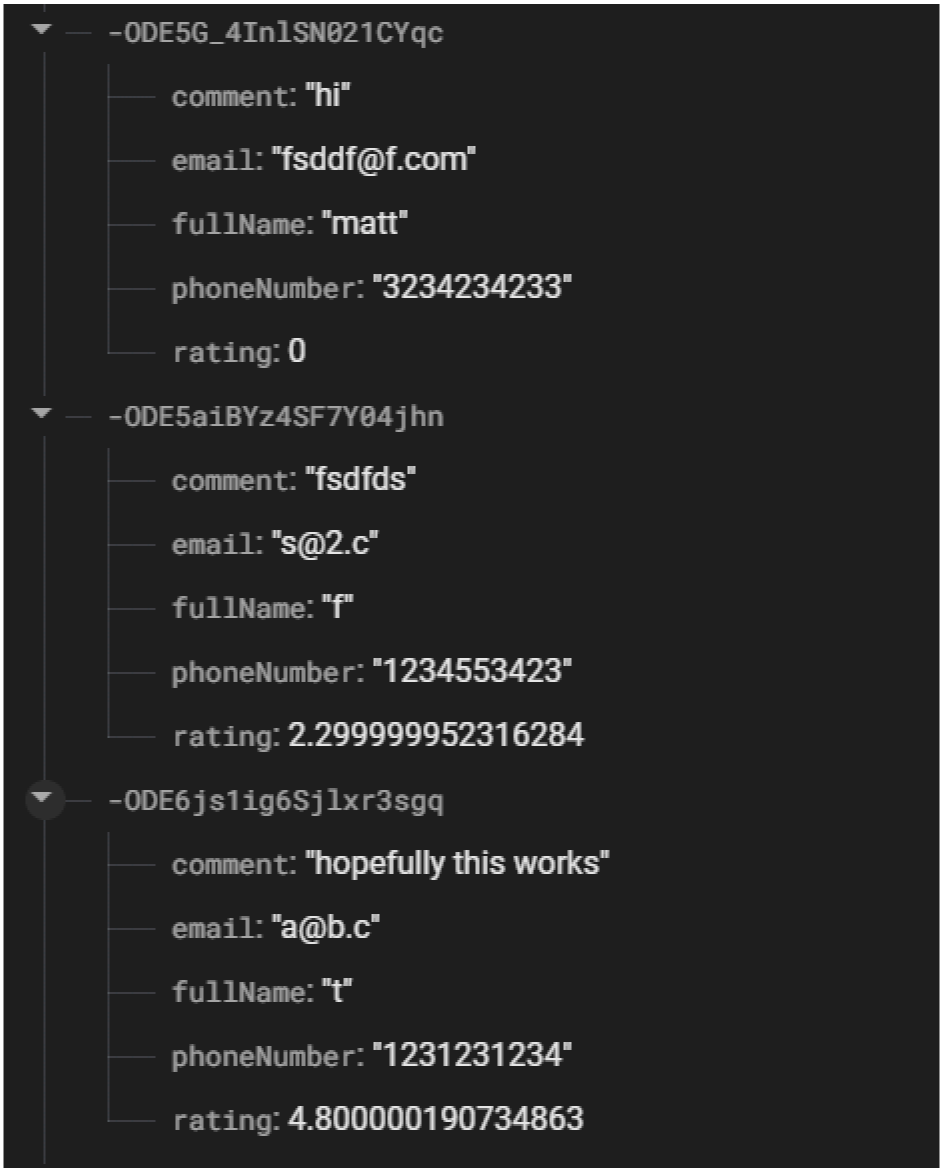
Your Email

Your Comments

Success

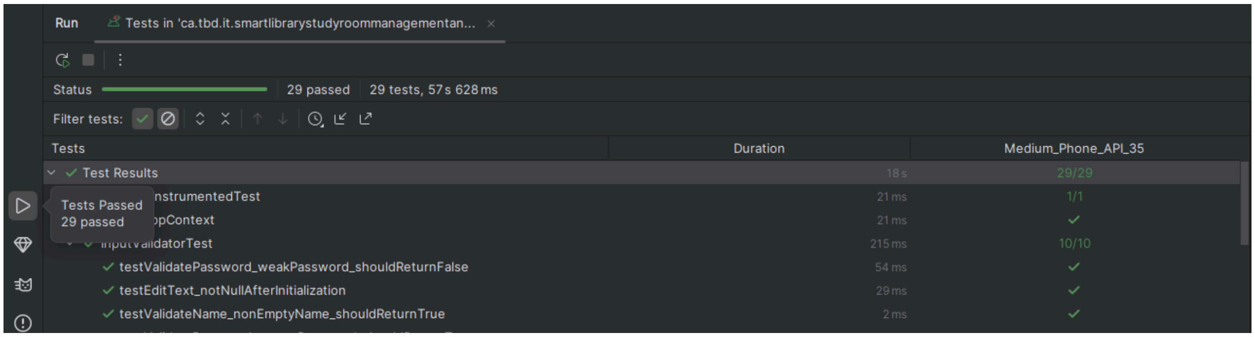
Feedback submitted successfully!

37. Add into pdf file screenshot showing the data stored into the DB from Feedback. Must have at least 3 different entries.



```
-ODE5G_4InlSN021CYqc
    comment: "hi"
    email: "fsddf@f.com"
    fullName: "matt"
    phoneNumber: "3234234233"
    rating: 0
-ODE5aiBYz4SF7Y04jhn
    comment: "fsdfds"
    email: "s@2.c"
    fullName: "f"
    phoneNumber: "1234553423"
    rating: 2.299999952316284
-ODE6js1ig6Sjlxr3sgq
    comment: "hopefully this works"
    email: "a@b.c"
    fullName: "t"
    phoneNumber: "1231231234"
    rating: 4.800000190734863
```

38. Describe your strategy for writing the test cases.
39. Screenshots showing 3 classes junit test cases are passing.



Run   Tests in 'ca.tbd.it.smartlibrarystudyroommanagementan...

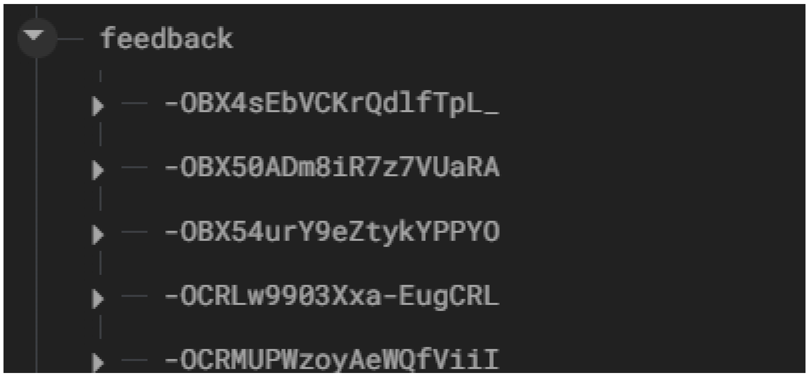
Status   29 passed   29 tests, 57 s 628 ms

Filter tests:

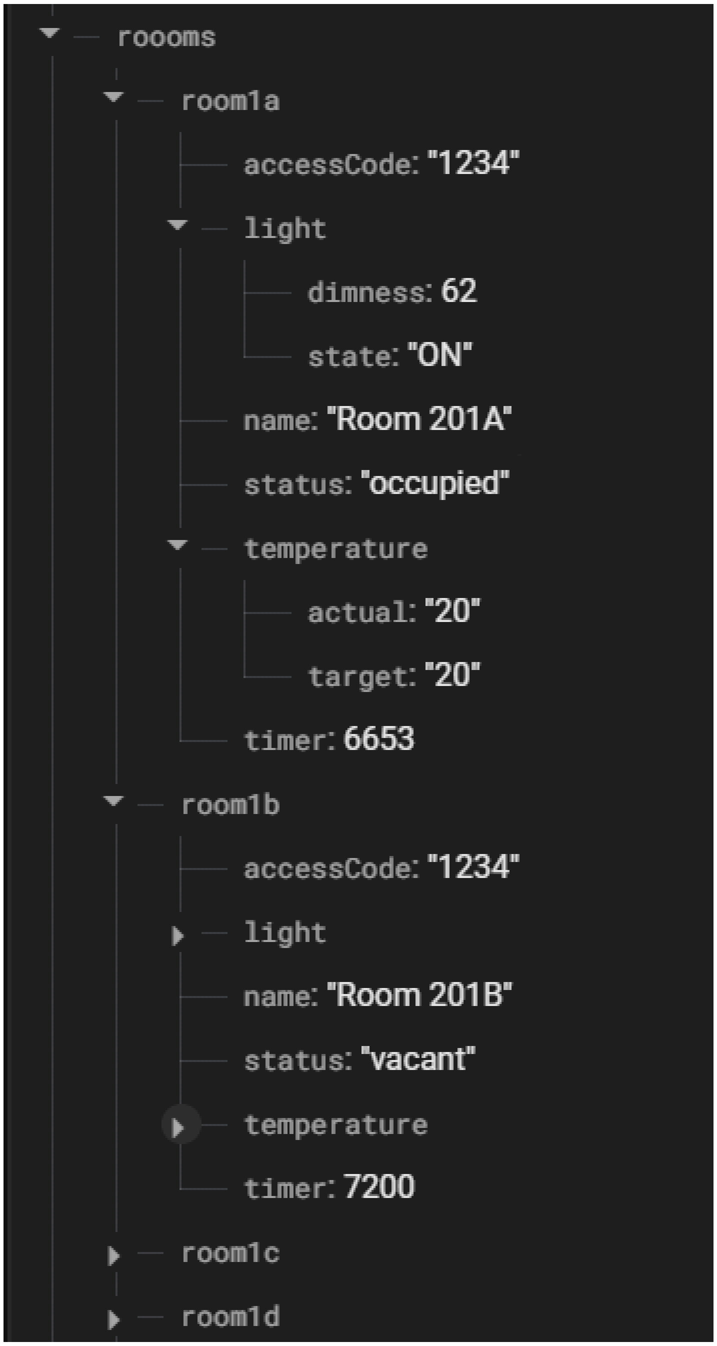
Tests	Duration	Medium_Phone_API_35
✓ Test Results	18 s	29/29
nstrumentedTest	21 ms	1/1
pContext	21 ms	✓
✓ inputValidatorTest	215 ms	10/10
✓ testValidatePassword_weakPassword_shouldReturnFalse	54 ms	✓
✓ testEditText_notNullAfterInitialization	29 ms	✓
✓ testValidateName_nonEmptyName_shouldReturnTrue	2 ms	✓

Tests Passed
29 passed

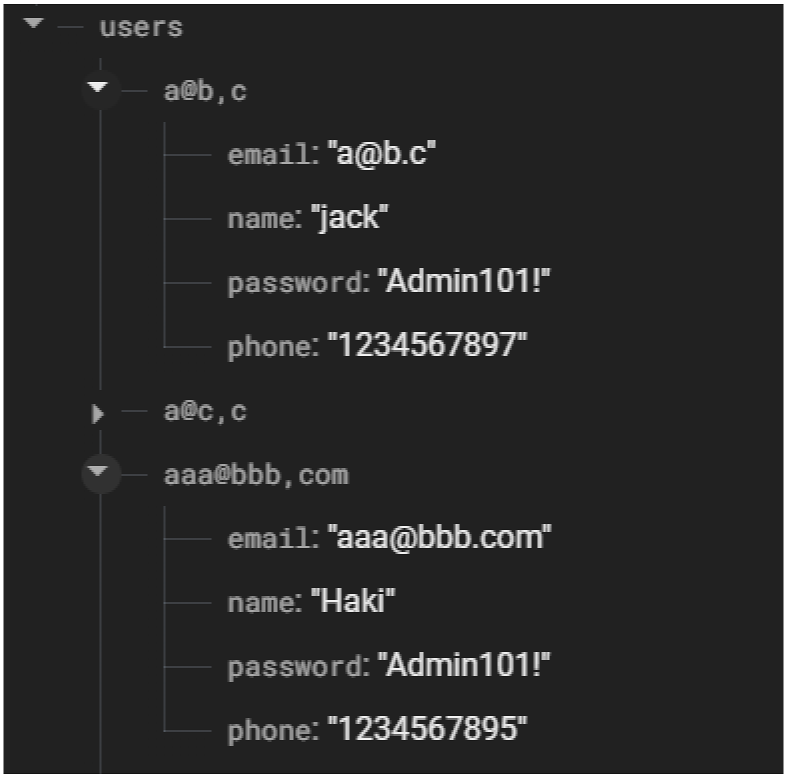
40. Screenshot showing Robleteric test cases are passing
41. Screenshot showing Espresso test cases are passing.
42. Screenshots showing the data in your DB, sensors data, login data (including Google login), and Feedback Screen data.



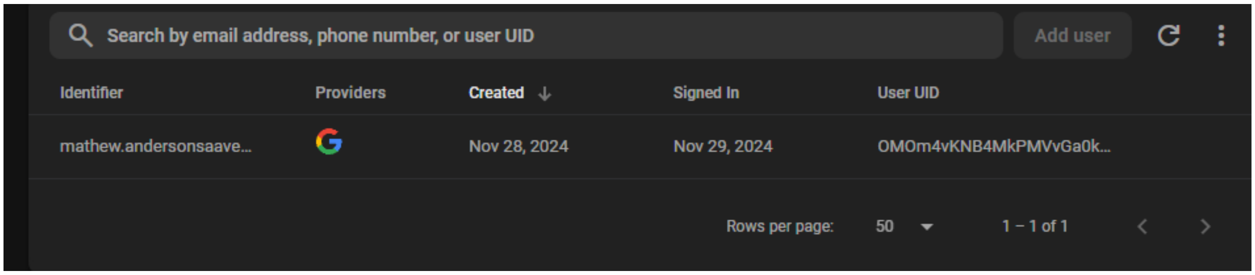
```
feedback
  -OBX4sEbVCKrQdlfTpL_
  -OBX50ADm8iR7z7VUaRA
  -OBX54urY9eZtykYPPYO
  -OCRLw9903Xxa-EugCRL
  -OCRMUPWzoyAeWQfViiI
```



```
rooms
  room1a
    accessCode: "1234"
    light
      dimness: 62
      state: "ON"
    name: "Room 201A"
    status: "occupied"
    temperature
      actual: "20"
      target: "20"
    timer: 6653
  room1b
    accessCode: "1234"
    light
    name: "Room 201B"
    status: "vacant"
    temperature
    timer: 7200
  room1c
  room1d
```



```
users
  a@b,c
    email: "a@b.c"
    name: "jack"
    password: "Admin101!"
    phone: "1234567897"
  a@c,c
  aaa@bbb,com
    email: "aaa@bbb.com"
    name: "Haki"
    password: "Admin101!"
    phone: "1234567895"
```



Identifier	Providers	Created ↓	Signed In	User UID
mathew.andersonsaave...	G	Nov 28, 2024	Nov 29, 2024	OMOm4vKNB4MkPMVvGa0k...

43. Suggestions to the instructor for future projects, things you liked, and things you suggest to be done differently and how.

Suggestions for Future Projects:

1. Provide more tutorials on advanced features like Firebase and Google Sign-In.
2. Offer clearer testing guidelines for unit and UI tests.
3. Consider extending deadlines or breaking tasks into smaller milestones.

Things I Liked:

1. Hands-on experience with real-world technologies.
2. Clear and practical project scope.

Suggestions for Improvement:

1. Emphasize clearer documentation, including setup instructions.
2. Encourage peer reviews or group discussions for better collaboration.

44. Create a subfolder called deliverable5 under docs in the repo and add the pdf file.

45. Commit and push the ppt into deliverable5 folder, this is the

only file that can be committed and pushed pass the due and before your presentation.

46. All members must present. Zero for members who miss the presentation.

47. Pracjce on presenjng, and be ready, each team member will have 3 - 4 minutes to present.

48. If document not in pdf, or not in the proper name (marks will be deducted!)
Please save the PDF document, i.e.
TeamName_ProjectName_Deliverable5.pdf

49. One member to submit the pdf file.