CENG-322
Deliverable 4

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	70
Mathew Anderson-Saavedra	N01436706	MathewAnderson6706	Signed by Mathew A	80
Safah Virk	N01596470	Safahvirk6470	Signed by Safah Virk	70

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.

Comparing Apps

8. *Compare your application with at least two existing apps in the market. Provide links and description of the two apps you selected.*

Humber's Study rooms
https://humber.libcal.com/
StudyStream
https://play.google.com/store/apps/details?id=live.studystream.app&hl=en_CA&pli=1
9. Highlight the differences between your app and these two apps.

With Humber's Study rooms, you have to book your room in advanced, get to see which rooms are available during a certain time slot. Our app, you see which rooms are available in real time, it is treated as a walk-in, instead of reserving rooms. With our app, you can also see and make changes about the room itself with the sensors. With

StudyStream, its similar to our app in a sense of being able to see available rooms and being able to join it, but it is all virtual, all online. So you can join an online room from around the world and study with people in an online environment.

10. Why you believe your app is better than these two apps.

I believe our app is better because we offer real-time availability for our rooms, allow users to change stuff around the room to better suit them and for their comfortability. That includes temperature, dimness of lighting, etc. It allows the user to see where the rooms are as well. If the user is satisfied or dissatisfied with the app, they can leave a review in which we can address. Also allow for multiple schools/campus to be supported on our app, as we are not just stuck with one school.

11. Create a table pros and cons of the three different apps.

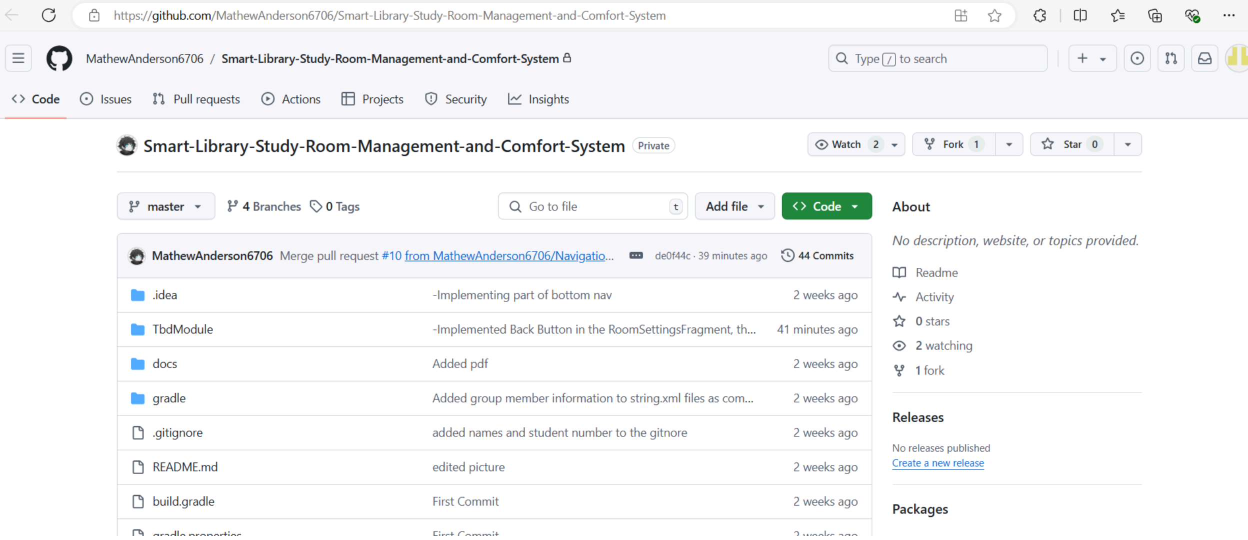
Apps	Pros	Cons
Smart Library Study Room Management and Comfort System	-Real Time availability -Feedback -Supports multiple schools -Change in person room settings(temperature, lighting, etc)	-Only an app -Not a lot of features -Difficult to understand at first of what you can click on
Humber Study Rooms	-Book a room ahead of time -View availability of rooms weeks in advanced	-Cant change lighting or temperature from an app -Supports only 1 school -Only a Website
StudyStream	-Virtual -Creates online rooms for the whole world -textchats and webcams -Allow for creative collaboration -Both an app and website	-Not in person rooms -Can have too many users in a room, become overcrowded

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

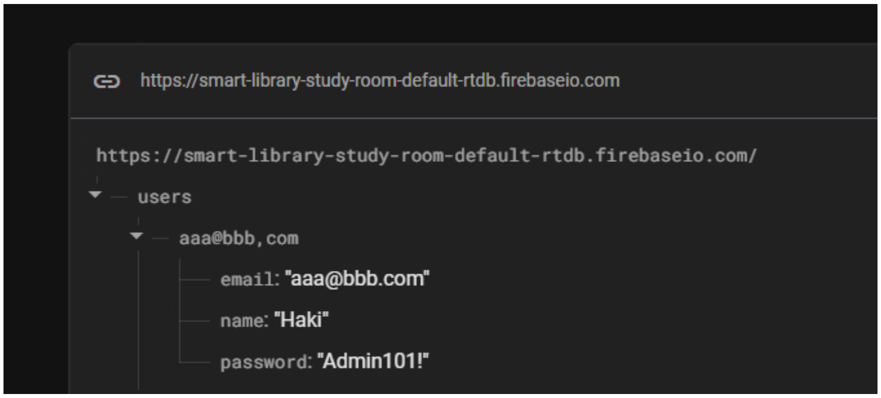

https://github.com/MathewAnderson6706/Smart-Library-Study-Room-Management-and-Comfort-System
MathewAnderson6706 / Smart-Library-Study-Room-Management-and-Comfort-System
Type / to search
Code
Issues
Pull requests
Actions
Projects
Security
Insights
Smart-Library-Study-Room-Management-and-Comfort-System
Private
Watch 2
Fork 1
Star 0
master
4 Branches
0 Tags
Go to file
Add file
Code
About
No description, website, or topics provided.
Readme
Activity
0 stars
2 watching
1 fork
Releases
No releases published
Create a new release
Packages

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

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

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

Mathew has worked on the following:

- Connecting the rooms to the database
- Connected the feedbackFragment to the database
- Removing all text on the image buttons for the rooms
- Change the src of rooms to reflect occupied/vacant rooms
- Create helper classes in the project
- Removed Bottom Navigation as no longer need it
- Connected roomSettingsFragment to the database
  - Basically depending on what room you enter, will show information in the roomSettingsFragment from that room.
- Updated the database with sensor information
- Cleaned up code to follow certain design patterns
- Organized Meetings
- Assigned group members tasks
- Documentation

Medi has worked on the following:

- Redesign of RoomSettings
  -That includes combining all 4 fragments we had for RoomSettings into 1 fragment and making the design user friendly

- Implementations into settings screen
- Drop-down Menu
- Test cases
- System Context Diagram
- Container Diagram
- Documentation

Safah has worked on the following:

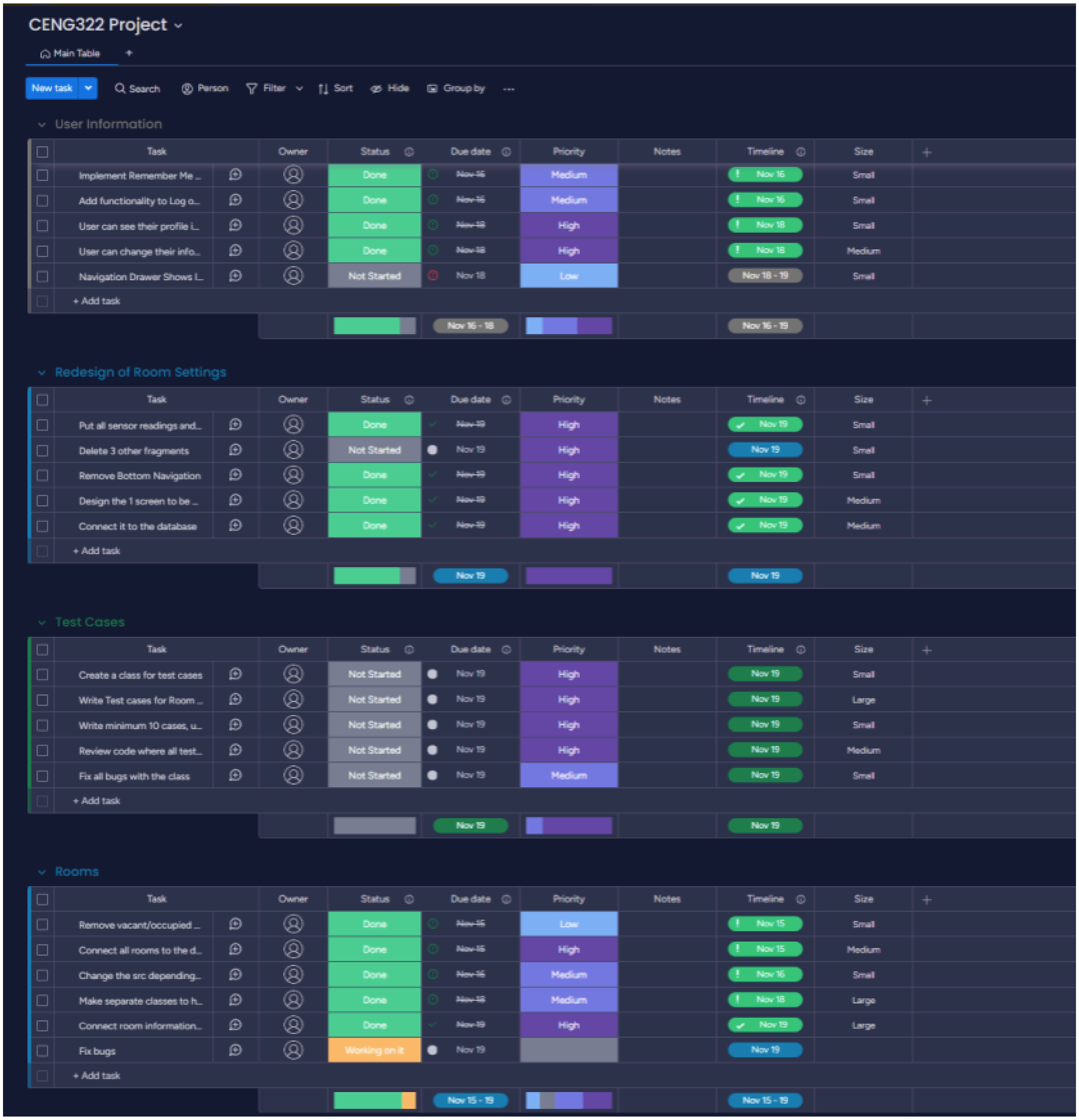
- Check mark box where the user does not have to log in every time
- Saving of user information
  -This includes the user information being shown in the Profile Fragment
- Log out button implementation
- Connected the Profile Fragment to the database
- User can now see and change their information in this fragment
- Add user info to nav drawer

22. Sprint goals, list sprint goals.

- Connect many aspects of our app to the database
- Have test cases up and running
- Complete main aspects of the app

23-28 **Sprint dashboard**

Showing Sprint 4 with closed tasks and tasks you did not complete.
show task, owner, status, startdate, end date, size and priority.



User Information

Task	Owner	Status	Due date	Priority	Notes	Timeline	Size
Implement Remember Me ...		Done	~~Nov 16~~	Medium		! Nov 16	Small
Add functionality to Log o...		Done	~~Nov 16~~	Medium		! Nov 16	Small
User can see their profile i...		Done	~~Nov 18~~	High		! Nov 18	Small
User can change their info...		Done	~~Nov 18~~	High		! Nov 18	Medium
Navigation Drawer Shows I...		Not Started	Nov 18	Low		Nov 18 - 19	Small
+ Add task							
			Nov 16 - 18			Nov 16 - 19	

Redesign of Room Settings

Task	Owner	Status	Due date	Priority	Notes	Timeline	Size
Put all sensor readings and...		Done	✓ ~~Nov 19~~	High		✓ Nov 19	Small
Delete 3 other fragments		Not Started	● Nov 19	High		Nov 19	Small
Remove Bottom Navigation		Done	✓ ~~Nov 19~~	High		✓ Nov 19	Small
Design the 1 screen to be ...		Done	✓ ~~Nov 19~~	High		✓ Nov 19	Medium
Connect it to the database		Done	✓ ~~Nov 19~~	High		✓ Nov 19	Medium
+ Add task							
			Nov 19			Nov 19	

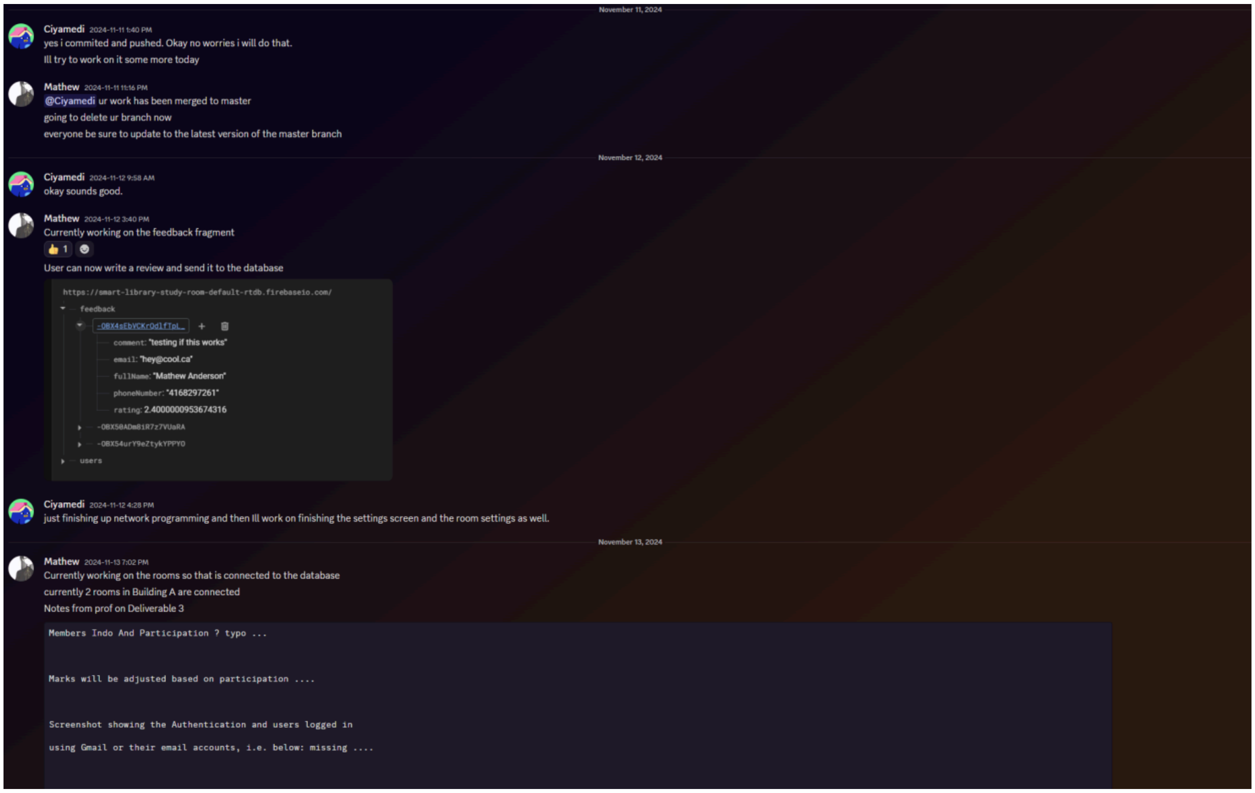
Test Cases

Task	Owner	Status	Due date	Priority	Notes	Timeline	Size
Create a class for test cases		Not Started	● Nov 19	High		Nov 19	Small
Write Test cases for Room ...		Not Started	● Nov 19	High		Nov 19	Large
Write minimum 10 cases, u...		Not Started	● Nov 19	High		Nov 19	Small
Review code where all test...		Not Started	● Nov 19	High		Nov 19	Medium
Fix all bugs with the class		Not Started	● Nov 19	Medium		Nov 19	Small
+ Add task							
			Nov 19			Nov 19	

Rooms

Task	Owner	Status	Due date	Priority	Notes	Timeline	Size
Remove vacant/occupied ...		Done	~~Nov 15~~	Low		! Nov 15	Small
Connect all rooms to the d...		Done	~~Nov 15~~	High		! Nov 15	Medium
Change the src depending...		Done	~~Nov 16~~	Medium		! Nov 16	Small
Make separate classes to h...		Done	~~Nov 18~~	Medium		! Nov 18	Large
Connect room information...		Done	✓ ~~Nov 19~~	High		✓ Nov 19	Large
Fix bugs		Working on it	● Nov 19			Nov 19	
+ Add task							
			Nov 15 - 19			Nov 15 - 19	

29. Take screenshots showing the team discussion, topics discussed, team members. Provide screenshots for three different dates, i.e. WhatsApp, Discord.



November 11, 2024

Ciyamedi 2024-11-11 1:40 PM
yes i commited and pushed. Okay no worries i will do that.
Ill try to work on it some more today

Mathew 2024-11-11 11:16 PM
@Ciyamedi ur work has been merged to master
going to delete ur branch now
everyone be sure to update to the latest version of the master branch

November 12, 2024

Ciyamedi 2024-11-12 9:58 AM
okay sounds good.

Mathew 2024-11-12 3:40 PM
Currently working on the feedback fragment
👍 1
User can now write a review and send it to the database

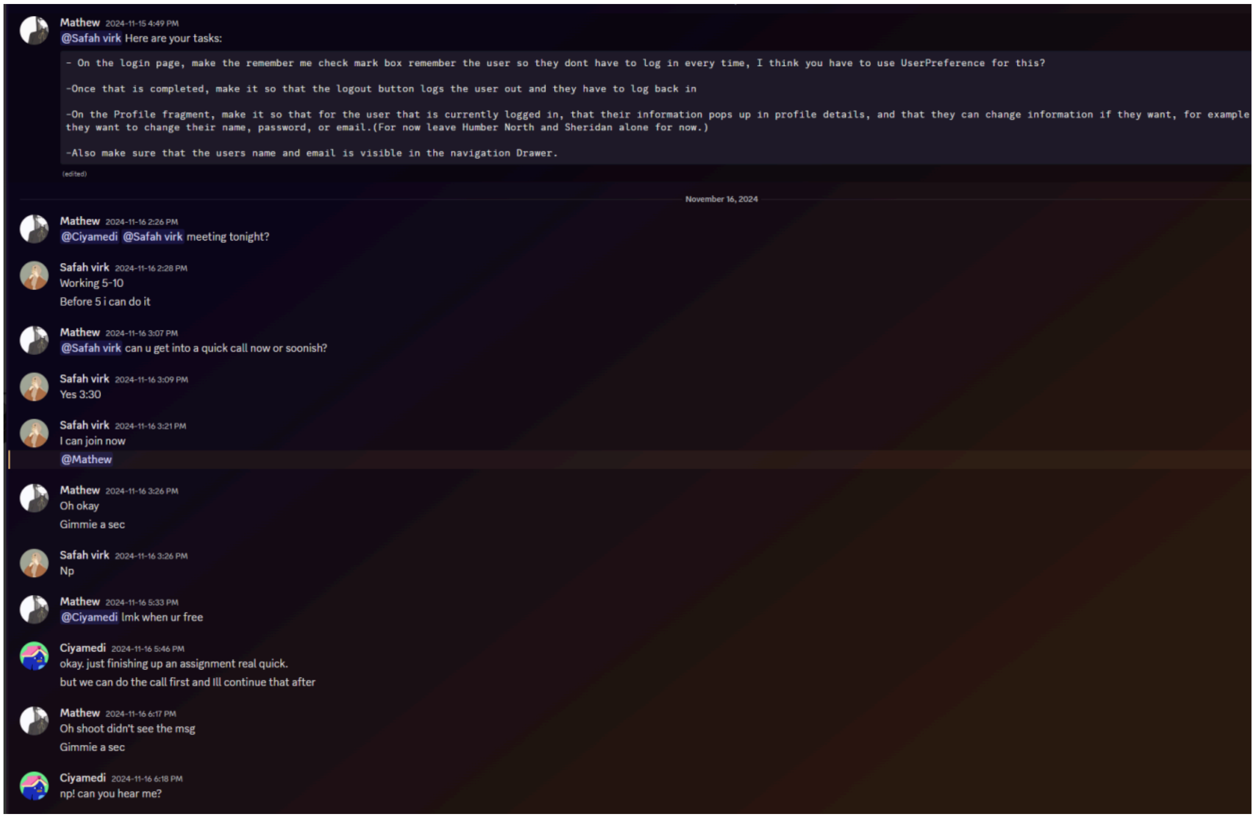
```
https://smart-library-study-room-default-rtdb.firebaseio.com/
▾ feedback
    ▾ -OBX4sEbVCKrOdlfTpL_ + 🗑
        comment: "testing if this works"
        email: "hey@cool.ca"
        fullName: "Mathew Anderson"
        phoneNumber: "4168297261"
        rating: 2.4000000953674316
    ▸ -OBX50ADm81R7z7VUaRA
    ▸ -OBX54urY9eZtykYPPYO
▸ users
```

Ciyamedi 2024-11-12 4:28 PM
just finishing up network programming and then Ill work on finishing the settings screen and the room settings as well.

November 13, 2024

Mathew 2024-11-13 7:02 PM
Currently working on the rooms so that is connected to the database
currently 2 rooms in Building A are connected
Notes from prof on Deliverable 3

```
Members Indo And Participation ? typo ...


Marks will be adjusted based on participation ....


Screenshot showing the Authentication and users logged in

using Gmail or their email accounts, i.e. below: missing ....
```



Mathew 2024-11-15 4:49 PM

@Safah virk Here are your tasks:

```
- On the login page, make the remember me check mark box remember the user so they dont have to log in every time, I think you have to use UserPreference for this?

-Once that is completed, make it so that the logout button logs the user out and they have to log back in

-On the Profile fragment, make it so that for the user that is currently logged in, that their information pops up in profile details, and that they can change information if they want, for example
they want to change their name, password, or email.(For now leave Humber North and Sheridan alone for now.)

-Also make sure that the users name and email is visible in the navigation Drawer.
```

(edited)

November 16, 2024

Mathew 2024-11-16 2:26 PM

@Ciyamedi @Safah virk meeting tonight?

Safah virk 2024-11-16 2:28 PM

Working 5-10

Before 5 i can do it

Mathew 2024-11-16 3:07 PM

@Safah virk can u get into a quick call now or soonish?

Safah virk 2024-11-16 3:09 PM

Yes 3:30

Safah virk 2024-11-16 3:21 PM

I can join now

@Mathew

Mathew 2024-11-16 3:26 PM

Oh okay

Gimmie a sec

Safah virk 2024-11-16 3:26 PM

Np

Mathew 2024-11-16 5:33 PM

@Ciyamedi lmk when ur free

Ciyamedi 2024-11-16 5:46 PM

okay. just finishing up an assignment real quick.

but we can do the call first and Ill continue that after

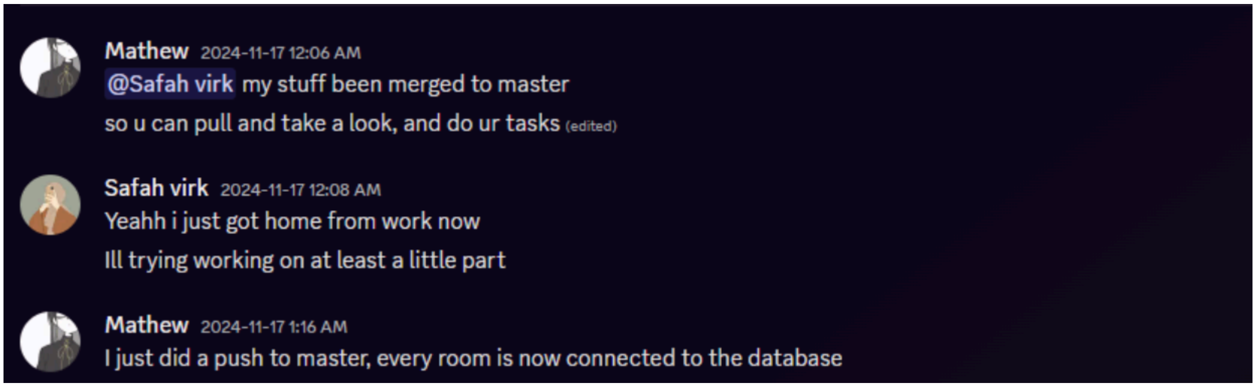
Mathew 2024-11-16 6:17 PM

Oh shoot didn't see the msg

Gimmie a sec

Ciyamedi 2024-11-16 6:18 PM

np! can you hear me?



Mathew 2024-11-17 12:06 AM

@Safah virk my stuff been merged to master

so u can pull and take a look, and do ur tasks (edited)

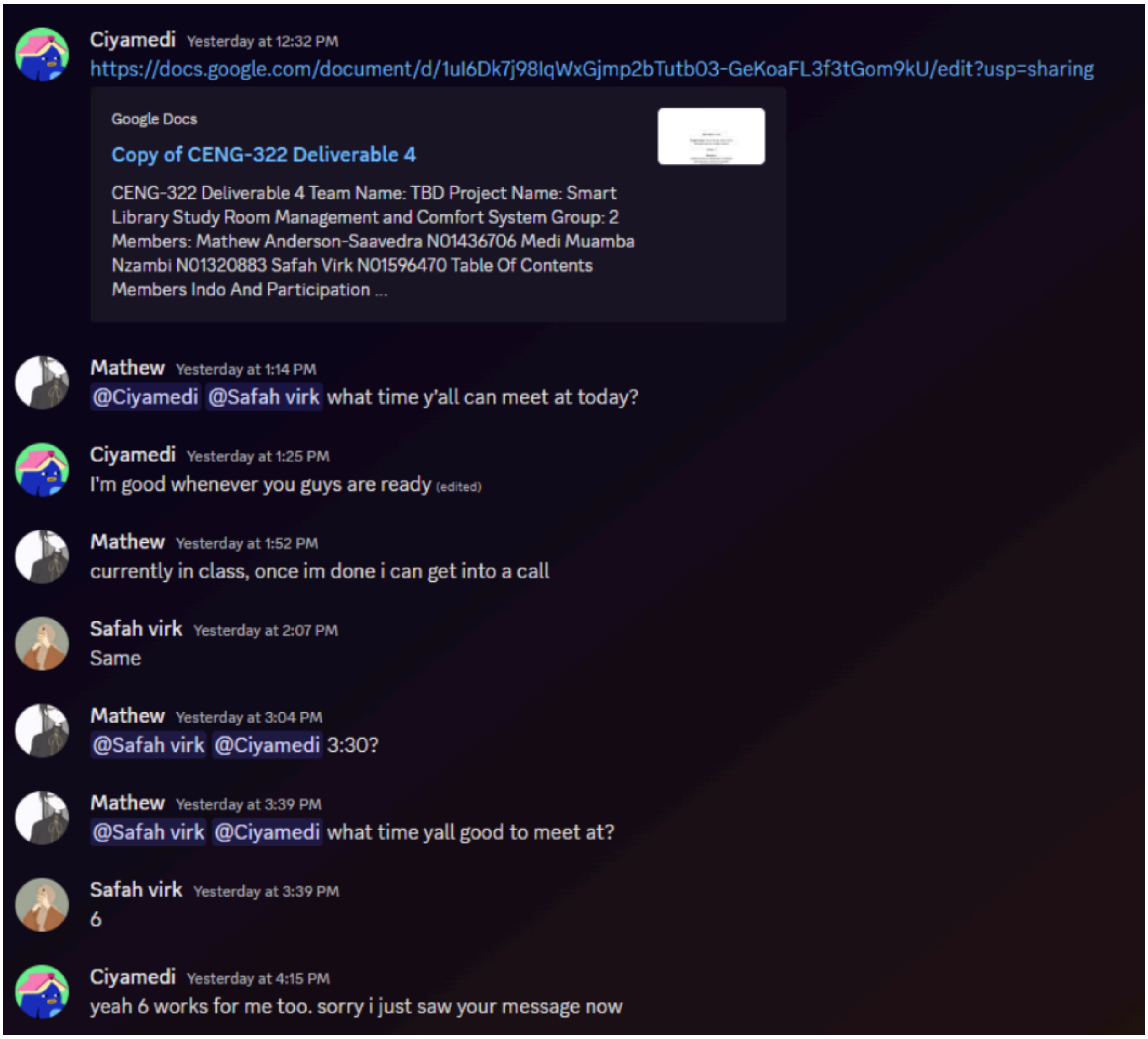
Safah virk 2024-11-17 12:08 AM

Yeahh i just got home from work now

Ill trying working on at least a little part

Mathew 2024-11-17 1:16 AM

I just did a push to master, every room is now connected to the database



Ciyamedi Yesterday at 12:32 PM
https://docs.google.com/document/d/1ul6Dk7j98IqWxGjmp2bTutb03-GeKoaFL3f3tGom9kU/edit?usp=sharing

Google Docs
Copy of CENG-322 Deliverable 4
CENG-322 Deliverable 4 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 Indo And Participation ...

Mathew Yesterday at 1:14 PM
@Ciyamedi @Safah virk what time y'all can meet at today?

Ciyamedi Yesterday at 1:25 PM
I'm good whenever you guys are ready (edited)

Mathew Yesterday at 1:52 PM
currently in class, once im done i can get into a call

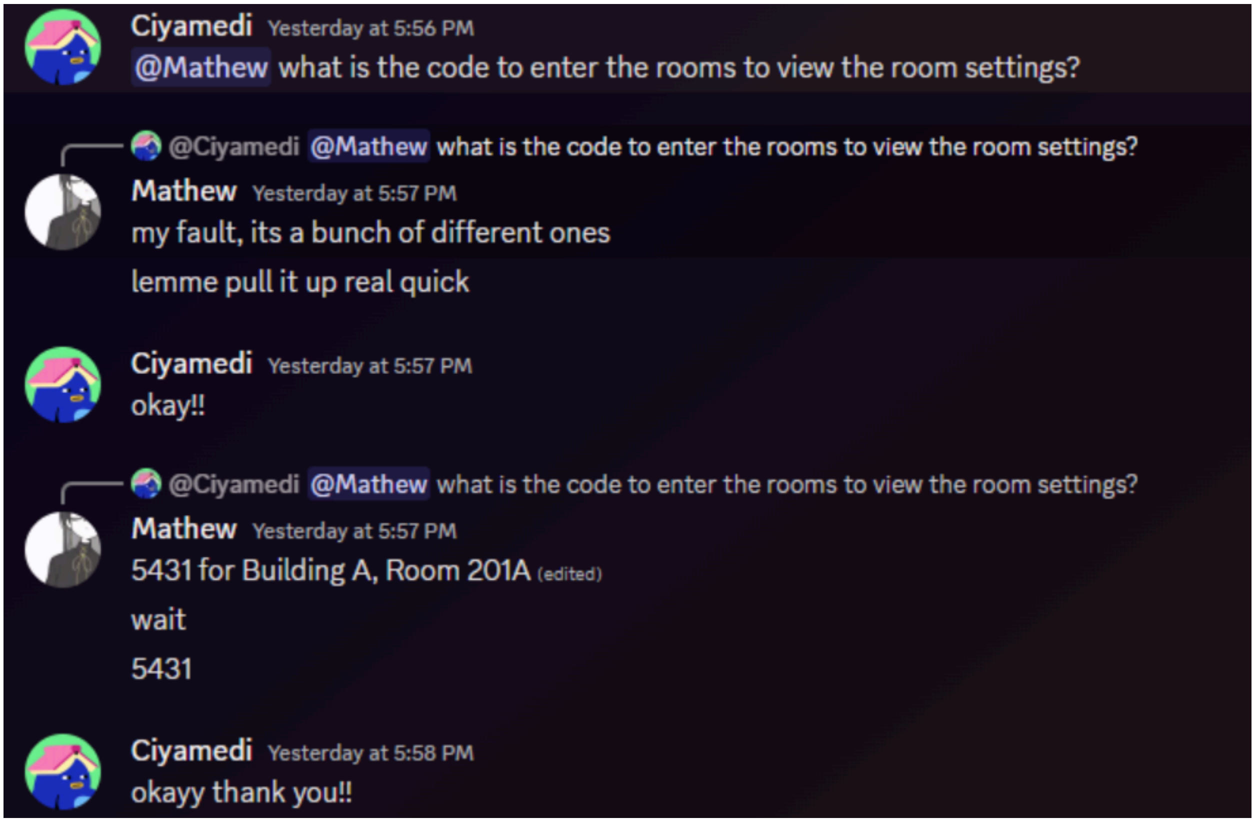
Safah virk Yesterday at 2:07 PM
Same

Mathew Yesterday at 3:04 PM
@Safah virk @Ciyamedi 3:30?

Mathew Yesterday at 3:39 PM
@Safah virk @Ciyamedi what time yall good to meet at?

Safah virk Yesterday at 3:39 PM
6

Ciyamedi Yesterday at 4:15 PM
yeah 6 works for me too. sorry i just saw your message now



Ciyamedi Yesterday at 5:56 PM

@Mathew what is the code to enter the rooms to view the room settings?

@Ciyamedi @Mathew what is the code to enter the rooms to view the room settings?

Mathew Yesterday at 5:57 PM

my fault, its a bunch of different ones

lemme pull it up real quick

Ciyamedi Yesterday at 5:57 PM

okay!!

@Ciyamedi @Mathew what is the code to enter the rooms to view the room settings?

Mathew Yesterday at 5:57 PM

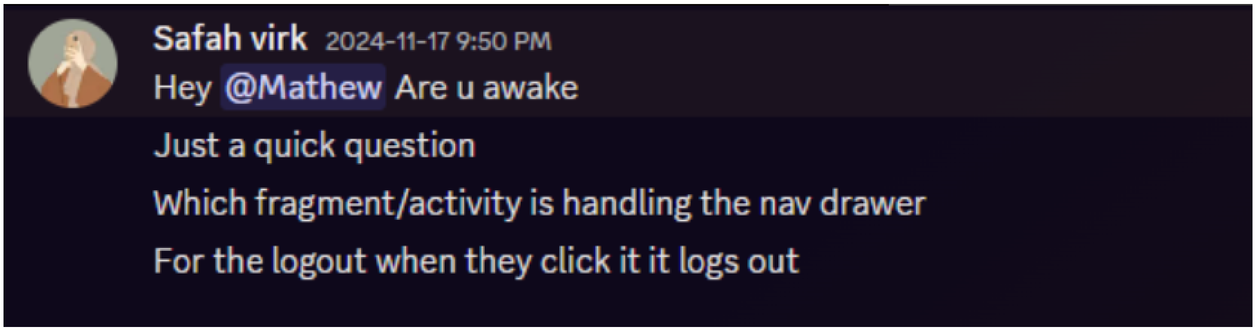
5431 for Building A, Room 201A (edited)

wait

5431

Ciyamedi Yesterday at 5:58 PM

okayy thank you!!



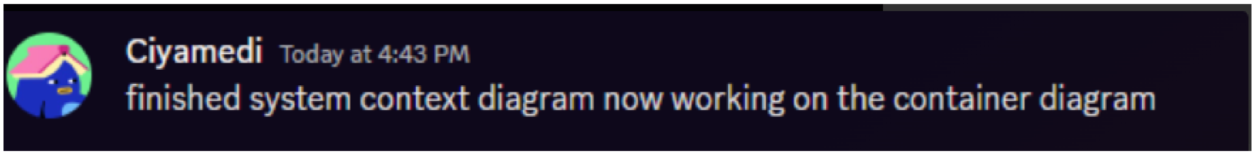
Safah virk 2024-11-17 9:50 PM

Hey @Mathew Are u awake

Just a quick question

Which fragment/activity is handling the nav drawer

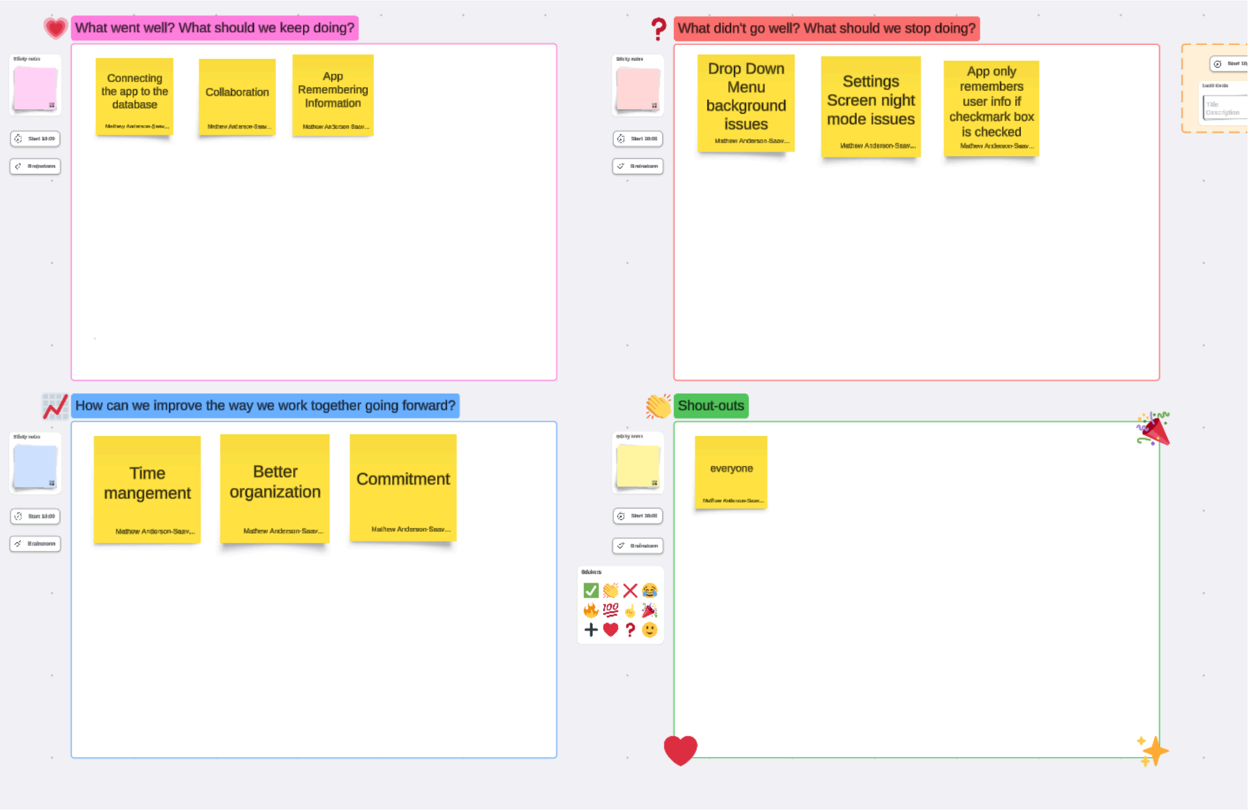
For the logout when they click it it logs out



Ciyamedi Today at 4:43 PM

finished system context diagram now working on the container diagram

30. Use a tool to record your Sprint Retrospective



What went well? What should we keep doing?

- Connecting the app to the database — Mathew Anderson-Saav...
- Collaboration — Mathew Anderson-Saav...
- App Remembering Information — Mathew Anderson-Saav...

What didn't go well? What should we stop doing?

- Drop Down Menu background issues — Mathew Anderson-Saav...
- Settings Screen night mode issues — Mathew Anderson-Saav...
- App only remembers user info if checkmark box is checked — Mathew Anderson-Saav...

How can we improve the way we work together going forward?

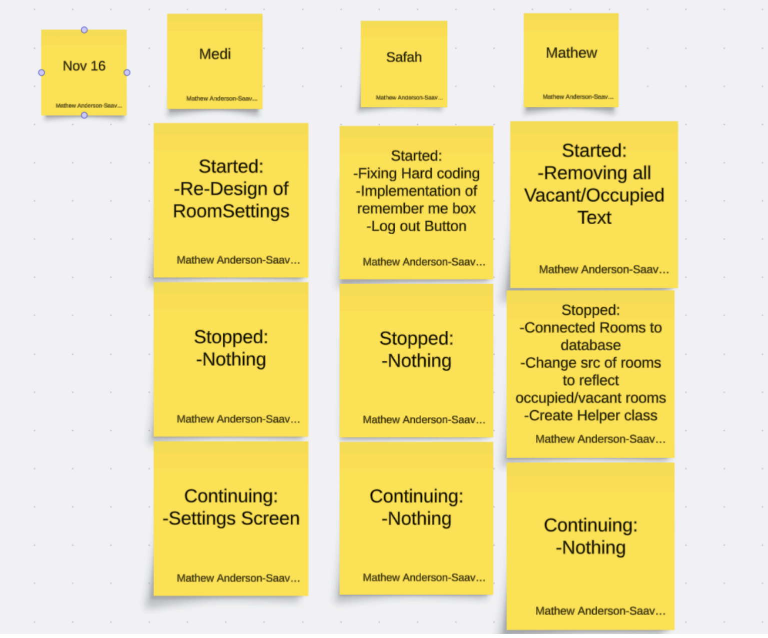
- Time mangement — Mathew Anderson-Saav...
- Better organization — Mathew Anderson-Saav...
- Commitment — Mathew Anderson-Saav...

Shout-outs

- everyone — Mathew Anderson-Saav...

1. Who missed the meeting, marks will be deducted for missing the retro.

1. Start doing.
2. Stop doing.
3. Continue doing.



Nov 16

Mathew Anderson-Saav...

Medi

Mathew Anderson-Saav...

Started:
-Re-Design of RoomSettings

Mathew Anderson-Saav...

Stopped:
-Nothing

Mathew Anderson-Saav...

Continuing:
-Settings Screen

Mathew Anderson-Saav...

Safah

Mathew Anderson-Saav...

Started:
-Fixing Hard coding
-Implementation of remember me box
-Log out Button

Mathew Anderson-Saav...

Stopped:
-Nothing

Mathew Anderson-Saav...

Continuing:
-Nothing

Mathew Anderson-Saav...

Mathew

Mathew Anderson-Saav...

Started:
-Removing all Vacant/Occupied Text

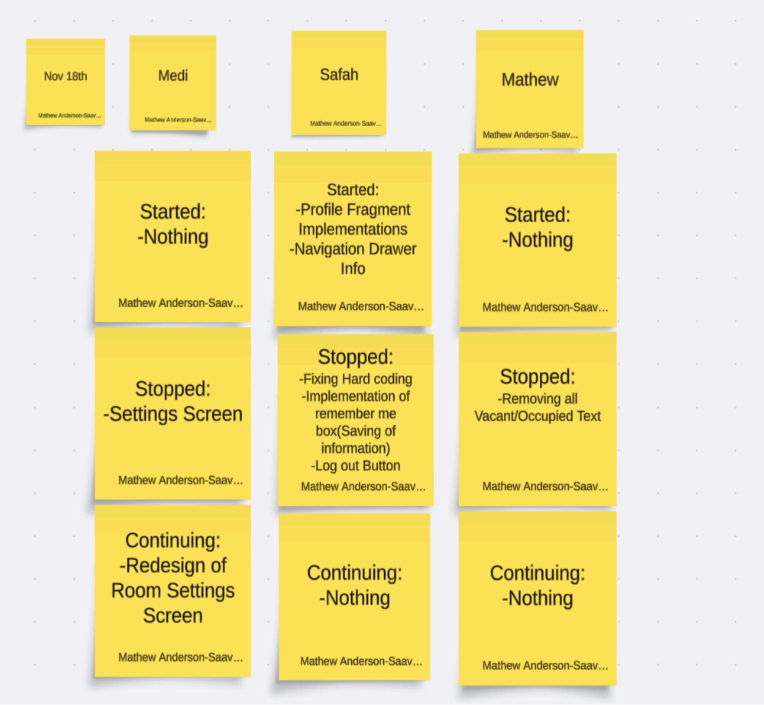
Mathew Anderson-Saav...

Stopped:
-Connected Rooms to database
-Change src of rooms to reflect occupied/vacant rooms
-Create Helper class

Mathew Anderson-Saav...

Continuing:
-Nothing

Mathew Anderson-Saav...



Nov 18th

Mathew Anderson-Saav...

Medi

Mathew Anderson-Saav...

Started:
-Nothing

Mathew Anderson-Saav...

Stopped:
-Settings Screen

Mathew Anderson-Saav...

Continuing:
-Redesign of Room Settings Screen

Mathew Anderson-Saav...

Safah

Mathew Anderson-Saav...

Started:
-Profile Fragment Implementations
-Navigation Drawer Info

Mathew Anderson-Saav...

Stopped:
-Fixing Hard coding
-Implementation of remember me box(Saving of information)
-Log out Button

Mathew Anderson-Saav...

Continuing:
-Nothing

Mathew Anderson-Saav...

Mathew

Mathew Anderson-Saav...

Started:
-Nothing

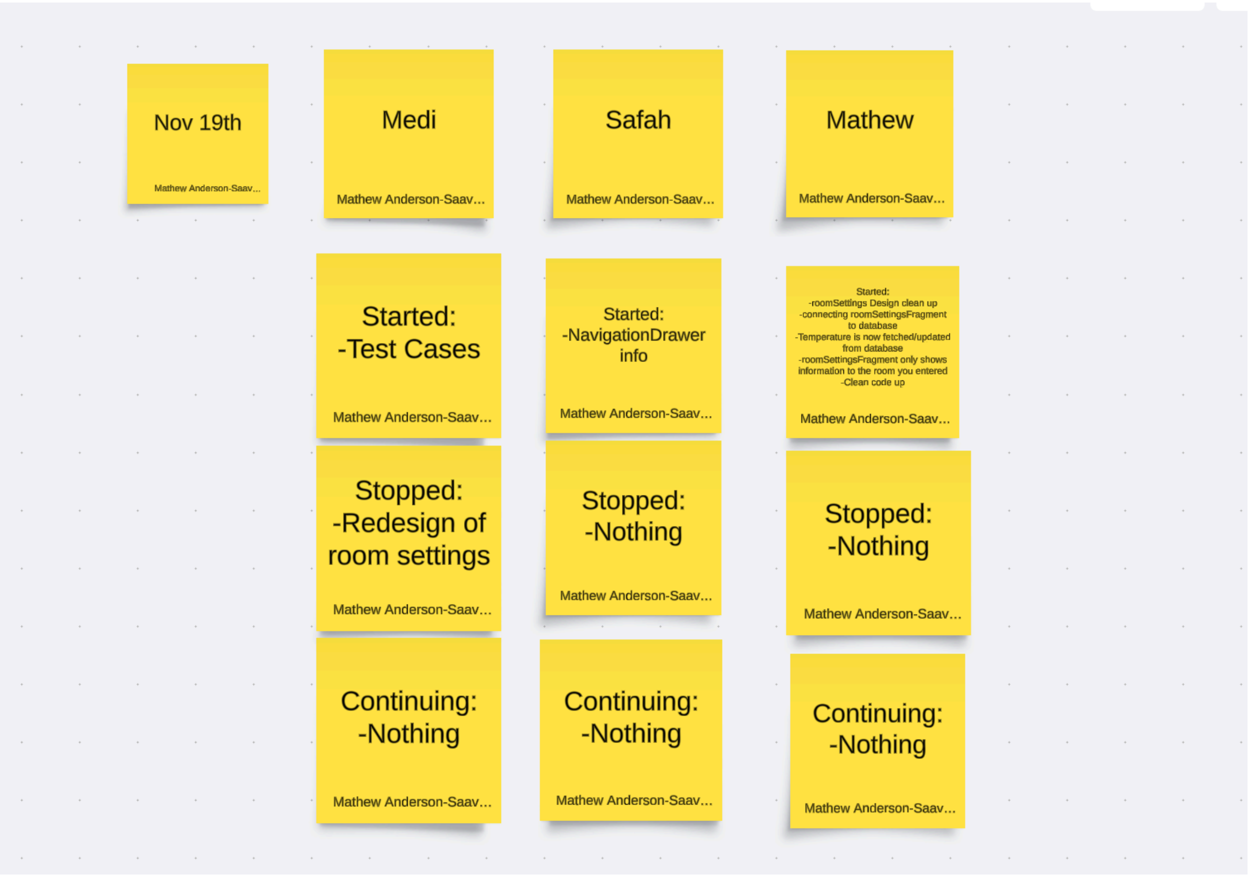
Mathew Anderson-Saav...

Stopped:
-Removing all Vacant/Occupied Text

Mathew Anderson-Saav...

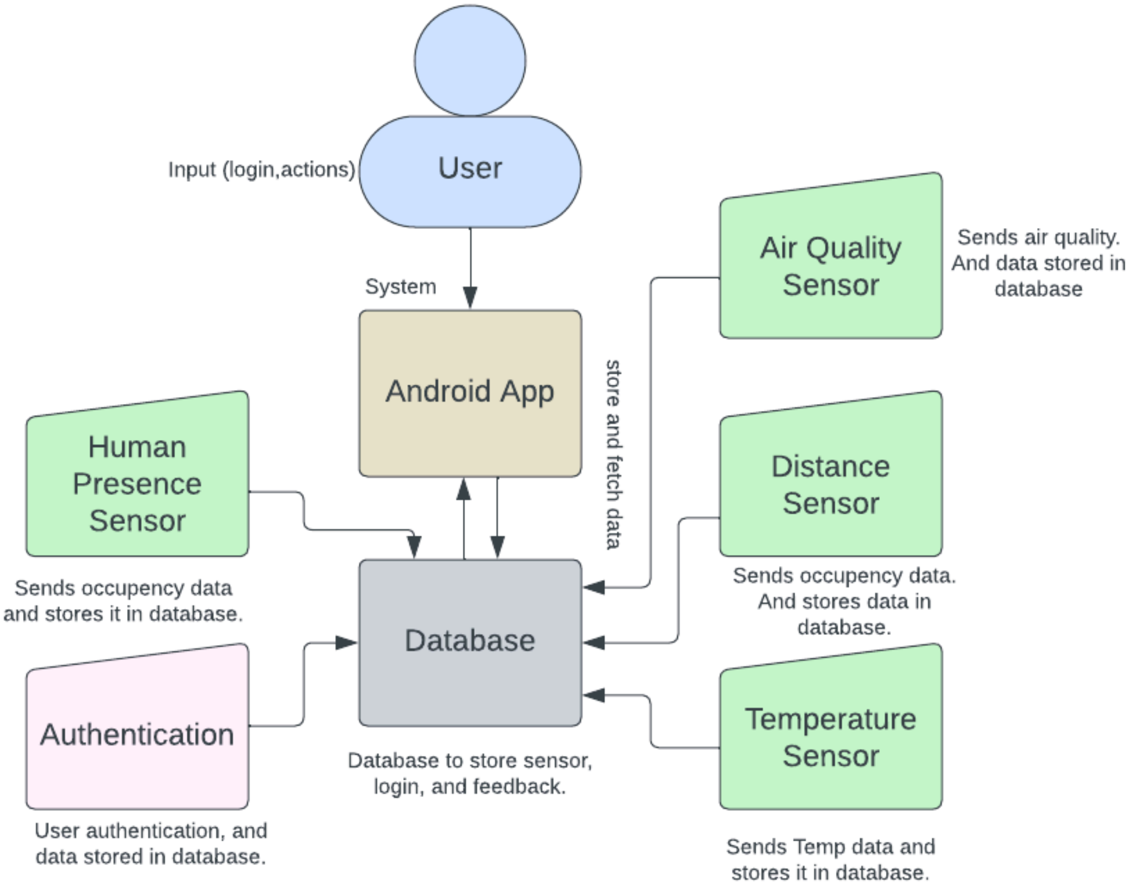
Continuing:
-Nothing

Mathew Anderson-Saav...



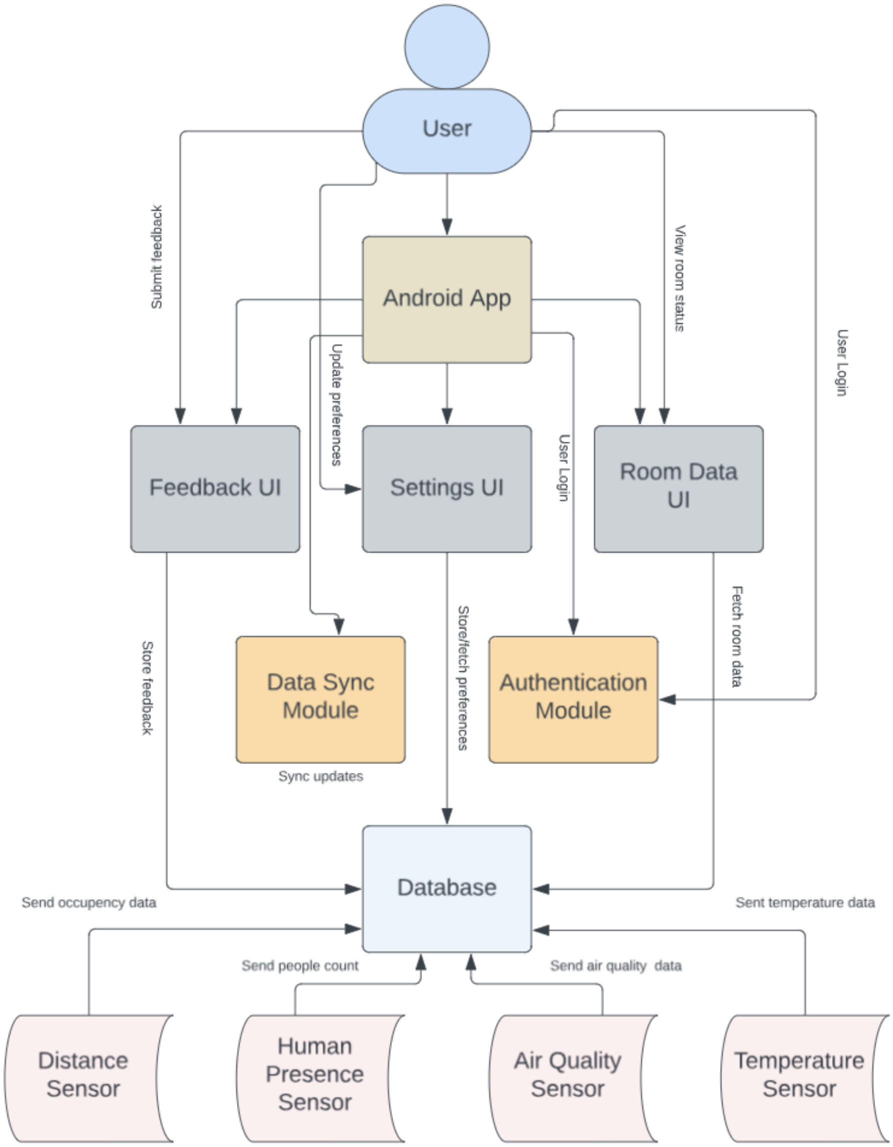
Nov 19th Mathew Anderson-Saav...	Medi Mathew Anderson-Saav...	Safah Mathew Anderson-Saav...	Mathew Mathew Anderson-Saav...
	Started: -Test Cases Mathew Anderson-Saav...	Started: -NavigationDrawer info Mathew Anderson-Saav...	Started: -roomSettings Design clean up -connecting roomSettingsFragment to database -Temperature is now fetched/updated from database -roomSettingsFragment only shows information to the room you entered -Clean code up Mathew Anderson-Saav...
	Stopped: -Redesign of room settings Mathew Anderson-Saav...	Stopped: -Nothing Mathew Anderson-Saav...	Stopped: -Nothing Mathew Anderson-Saav...
	Continuing: -Nothing Mathew Anderson-Saav...	Continuing: -Nothing Mathew Anderson-Saav...	Continuing: -Nothing Mathew Anderson-Saav...

31. Using C4 Model, show “System Context Diagram”.



User
Input (login,actions)
System
Android App
store and fetch data
Air Quality Sensor
Sends air quality. And data stored in database
Human Presence Sensor
Sends occupency data and stores it in database.
Distance Sensor
Sends occupency data. And stores data in database.
Database
Authentication
Database to store sensor, login, and feedback.
User authentication, and data stored in database.
Temperature Sensor
Sends Temp data and stores it in database.

32. Using C4 Model, show “Container Diagram”.

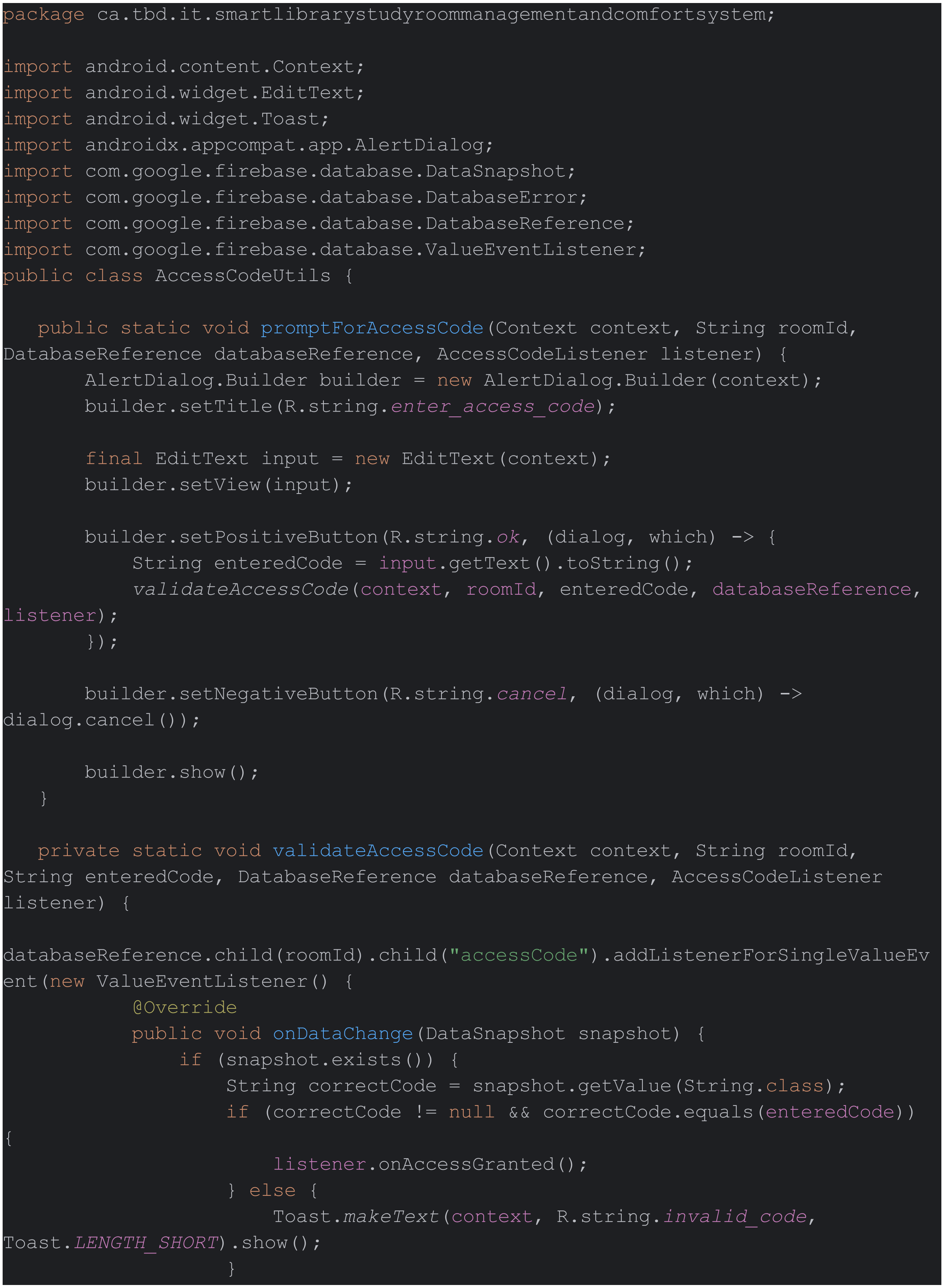


33. Document two different design patterns used in the code. Copy the code you used, and add your explanation. Use the ones covered in the class.

34. Your code should take Design Principles and Design Patterns into consideration, the ones in covered in the class.

Used DRY and KISS

Example code:



```
package ca.tbd.it.smartlibrarystudyroommanagementandcomfortsystem;

import android.content.Context;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AlertDialog;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.ValueEventListener;
public class AccessCodeUtils {

   public static void promptForAccessCode(Context context, String roomId,
DatabaseReference databaseReference, AccessCodeListener listener) {
       AlertDialog.Builder builder = new AlertDialog.Builder(context);
       builder.setTitle(R.string.enter_access_code);

       final EditText input = new EditText(context);
       builder.setView(input);

       builder.setPositiveButton(R.string.ok, (dialog, which) -> {
           String enteredCode = input.getText().toString();
           validateAccessCode(context, roomId, enteredCode, databaseReference,
listener);
       });

       builder.setNegativeButton(R.string.cancel, (dialog, which) ->
dialog.cancel());

       builder.show();
   }

   private static void validateAccessCode(Context context, String roomId,
String enteredCode, DatabaseReference databaseReference, AccessCodeListener
listener) {

databaseReference.child(roomId).child("accessCode").addListenerForSingleValueEv
ent(new ValueEventListener() {
           @Override
           public void onDataChange(DataSnapshot snapshot) {
               if (snapshot.exists()) {
                   String correctCode = snapshot.getValue(String.class);
                   if (correctCode != null && correctCode.equals(enteredCode))
{
                       listener.onAccessGranted();
                   } else {
                       Toast.makeText(context, R.string.invalid_code,
Toast.LENGTH_SHORT).show();
                   }
```

```
                } else {
                    Toast.makeText(context, R.string.access_code_not_set,
Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onCancelled(DatabaseError error) {
                Toast.makeText(context, "Error: " + error.getMessage(),
Toast.LENGTH_SHORT).show();
            }
        });
    }

    public interface AccessCodeListener {
        void onAccessGranted();
    }
}
```

The code above was needed in multiple fragment classes, so instead of repeating this code over and over again, I created a class so I only had to write it once and use it in multiple fragments. It is also simple code that people can understand.

35. Coding work progress since deliverable 3. What additional features/functionality added since deliverable 3.

-Rooms are now updated from the database to be either vacant or occupied
-Temperature can now be read and updated in roomSettingsFragment
-roomSettingsFragment is redesigned
-App now remembers the users information if they click the remember me box
-User can now change their information in userProfileFragment

36. Write unit test cases. Select one of your Java classes and write a minimum of 10 test cases.

37. Demonstrate the use of assertEqual, assertTrue, assertFalse, and asssertNotNull in your testing.

```
@Test
public void validateUsername_shouldReturnTrueForValidInput() {

when(loginActivity.usernameInput.getText().toString()).thenReturn("validUser");

    boolean result = loginActivity.validateUsername();

```

```
   assertTrue("Username validation should return true for valid input",
result);
}

@Test
public void validateUsername_shouldReturnFalseForEmptyInput() {
   when(loginActivity.usernameInput.getText().toString()).thenReturn("");

   boolean result = loginActivity.validateUsername();

   assertFalse("Username validation should return false for empty input",
result);
}

@Test
public void validatePassword_shouldReturnTrueForValidInput() {

when(loginActivity.passwordInput.getText().toString()).thenReturn("validPass");

   boolean result = loginActivity.validatePassword();

   assertTrue("Password validation should return true for valid input",
result);
}

@Test
public void validatePassword_shouldReturnFalseForEmptyInput() {
   when(loginActivity.passwordInput.getText().toString()).thenReturn("");

   boolean result = loginActivity.validatePassword();

   assertFalse("Password validation should return false for empty input",
result);
}

@Test
public void saveUserInfo_shouldStoreUsernameAndPassword() throws Exception {
   // Access private saveUserInfo using reflection
   Method saveUserInfo = LoginActivity.class.getDeclaredMethod("saveUserInfo",
String.class, String.class);
   saveUserInfo.setAccessible(true);
```

38. All test cases must pass.
The test cases did not pass.

39. Take screenshot showing all you test cases are passing.

40. Functionality on the Customer Feedback Screen below.

41. Display an error and don’t submit if invalid input, i.e. invalid phone number, …etc.

42. Display progress bar while the info is getting submitted, implement some delay for 5 seconds.

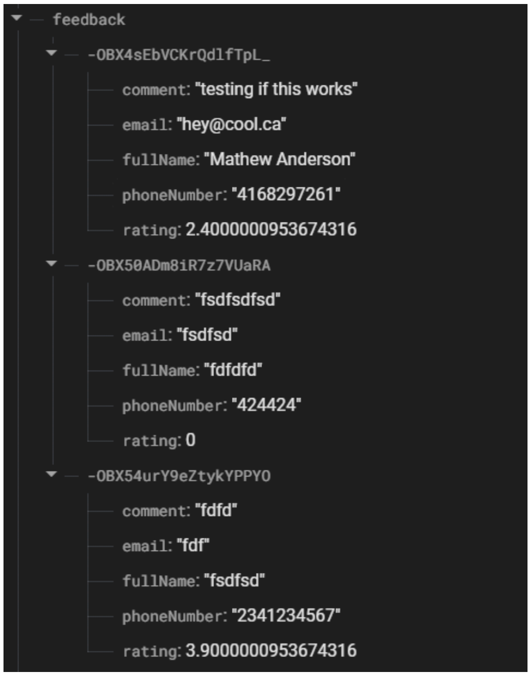
43. Once you receive a confirmation from the DB, display an AlertDialog with OK confirming the form has been submitted.

44. Verify you are submitting device model and stored into DB. You extract device model programmatically (don’t ask for device model in the form).
https://www.tutorialspoint.com/how-to-check-android-phonemodel-programmatically

45. Add into pdf file screenshot showing the progress bar while the form is gekng submiled.

46. Add into pdf file screenshot showing the AlertDialog once the form is submiled successfully.

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



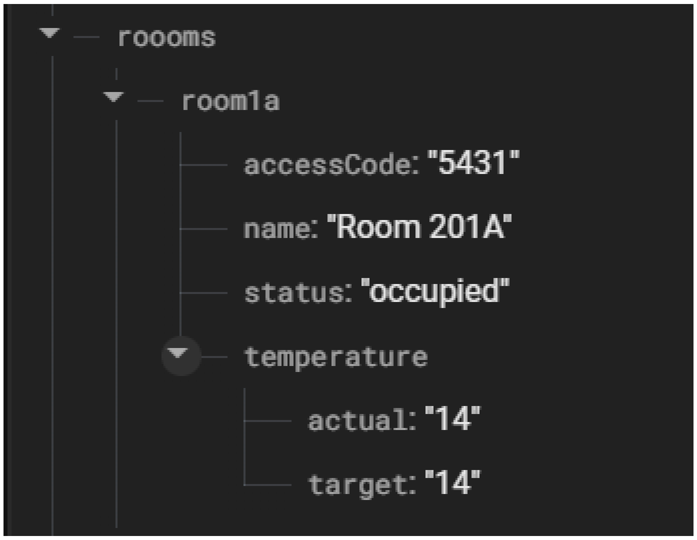
```
feedback
  -OBX4sEbVCKrQdlfTpL_
    comment: "testing if this works"
    email: "hey@cool.ca"
    fullName: "Mathew Anderson"
    phoneNumber: "4168297261"
    rating: 2.4000000953674316
  -OBX50ADm8iR7z7VUaRA
    comment: "fsdfsdfsd"
    email: "fsdfsd"
    fullName: "fdfdfd"
    phoneNumber: "424424"
    rating: 0
  -OBX54urY9eZtykYPPYO
    comment: "fdfd"
    email: "fdf"
    fullName: "fsdfsd"
    phoneNumber: "2341234567"
    rating: 3.9000000953674316
```

48. Clear the user input once the form is submitted. Restrict user submission to once per 24 hrs.

49. Once the user submits the form successfully, gray out the submit button, and display a timer showing how many hours and minutes remaining when the user can submit another feedback.

50. Describe in the pdf file on how you satisfied this Requirement.

51. Take screenshot showing all sensor data fetched and updated from the DB.



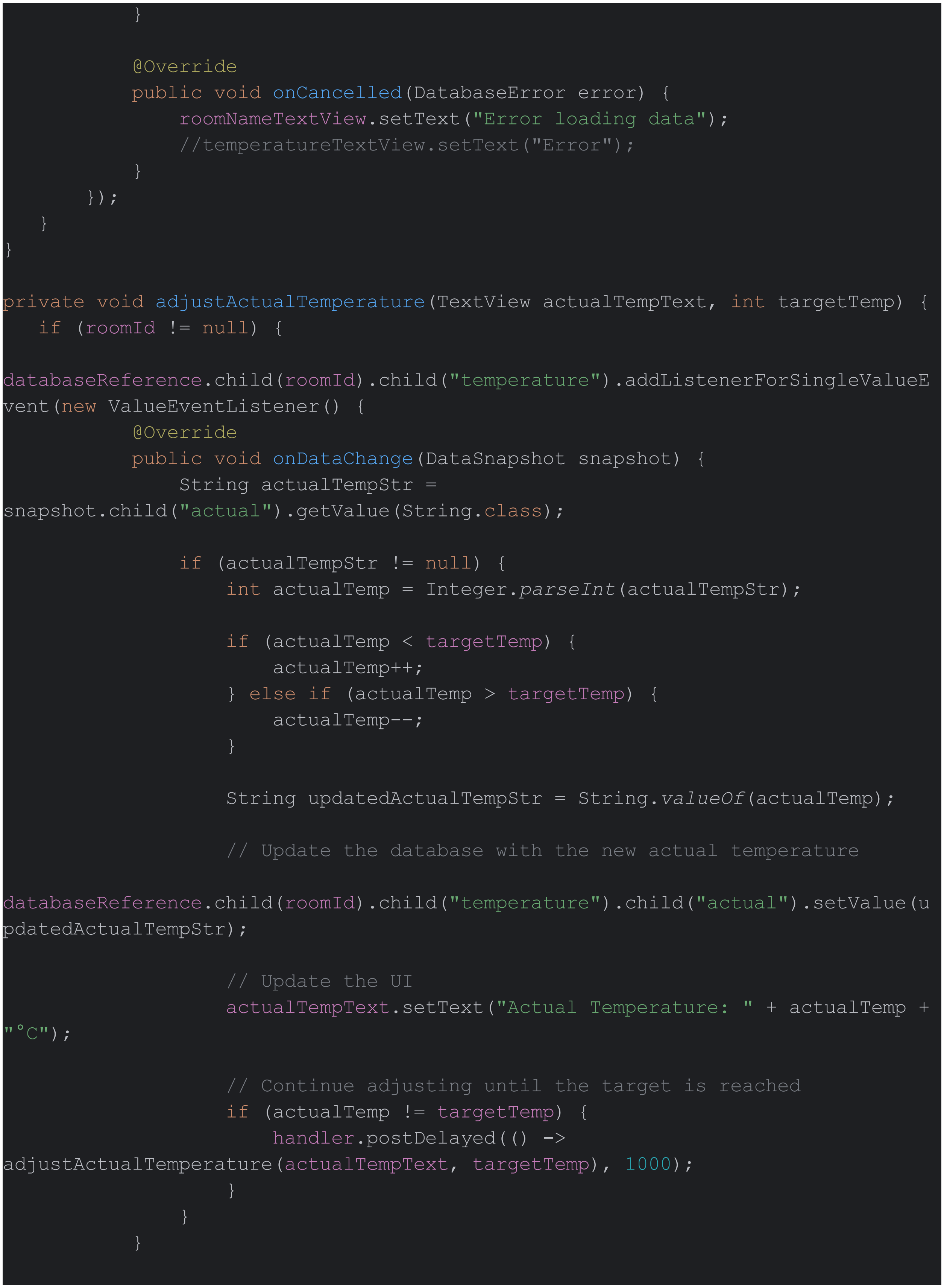
```
rooooms
  room1a
    accessCode: "5431"
    name: "Room 201A"
    status: "occupied"
    temperature
      actual: "14"
      target: "14"
```

Below fetches if the room is occupied or vacant

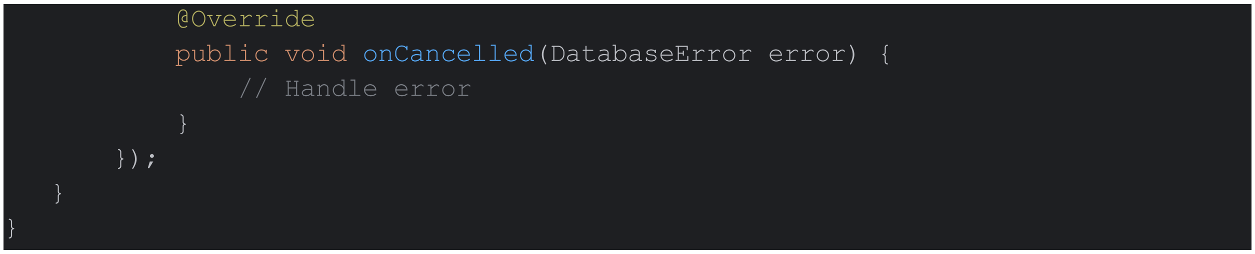
```
public static void setupRoom(Context context, ImageButton roomButton, String
roomId, DatabaseReference databaseReference, RoomActionListener listener) {
   databaseReference.child(roomId).addListenerForSingleValueEvent(new
ValueEventListener() {
       @Override
       public void onDataChange(DataSnapshot snapshot) {
           if (snapshot.exists()) {
               String status = snapshot.child("status").getValue(String.class);
               boolean isOccupied = "occupied".equals(status);

               roomButton.setEnabled(isOccupied);

               if (isOccupied) {
                   roomButton.setImageResource(R.drawable.roombooked);

roomButton.setBackgroundColor(ContextCompat.getColor(context,
R.color.colorPrimary));
                   roomButton.setOnClickListener(v ->
listener.onRoomSelected(roomId));
               }
           } else {
               Toast.makeText(context, "Room data not found",
Toast.LENGTH_SHORT).show();
           }
       }

       @Override
       public void onCancelled(DatabaseError error) {
           Toast.makeText(context, "Error: " + error.getMessage(),
Toast.LENGTH_SHORT).show();
       }
   });
}
```

Below fetches and sets Temperature

```
// Set target temperature
    setTemperatureButton.setOnClickListener(v -> {
        if (roomId != null) {
            int targetTemp = temperatureSeekBar.getProgress();
            String targetTempStr = String.valueOf(targetTemp);
            databaseReference.child(roomId).child("temperature").child("target").setValue(targetTempStr);
            targetTemperatureTextView.setText("Target Temperature: " + targetTemp + "°C");
            adjustActualTemperature(actualTemperatureTextView, targetTemp);
        }
    });


    return view;
}

private void fetchRoomData(TextView roomNameTextView, TextView actualTempText, TextView targetTempText, SeekBar seekBar) {
    if (roomId != null) {
        databaseReference.child(roomId).addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot snapshot) {
                if (snapshot.exists()) {
                    String roomName = snapshot.child("name").getValue(String.class);
                    String actualTemp = snapshot.child("temperature").child("actual").getValue(String.class);
                    String targetTemp = snapshot.child("temperature").child("target").getValue(String.class);

                    roomNameTextView.setText(roomName != null ? roomName : "Room Name Not Found");
                    actualTempText.setText(actualTemp != null ? "Actual Temperature: " + actualTemp + "°C" : "Actual Temperature Not Found");
                    targetTempText.setText(targetTemp != null ? "Target Temperature: " + targetTemp + "°C" : "Target Temperature Not Found");

                    if (targetTemp != null) {
                        seekBar.setProgress(Integer.parseInt(targetTemp));
                    }
                } else {
                    roomNameTextView.setText("Room Not Found");
                    actualTempText.setText("N/A");
                    targetTempText.setText("N/A");
                }
```



```
            }

            @Override
            public void onCancelled(DatabaseError error) {
                roomNameTextView.setText("Error loading data");
                //temperatureTextView.setText("Error");
            }
        });
    }
}

private void adjustActualTemperature(TextView actualTempText, int targetTemp) {
    if (roomId != null) {

databaseReference.child(roomId).child("temperature").addListenerForSingleValueE
vent(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot snapshot) {
                String actualTempStr =
snapshot.child("actual").getValue(String.class);

                if (actualTempStr != null) {
                    int actualTemp = Integer.parseInt(actualTempStr);

                    if (actualTemp < targetTemp) {
                        actualTemp++;
                    } else if (actualTemp > targetTemp) {
                        actualTemp--;
                    }

                    String updatedActualTempStr = String.valueOf(actualTemp);

                    // Update the database with the new actual temperature

databaseReference.child(roomId).child("temperature").child("actual").setValue(u
pdatedActualTempStr);

                    // Update the UI
                    actualTempText.setText("Actual Temperature: " + actualTemp +
"°C");

                    // Continue adjusting until the target is reached
                    if (actualTemp != targetTemp) {
                        handler.postDelayed(() ->
adjustActualTemperature(actualTempText, targetTemp), 1000);
                    }
                }
            }
```



```
            @Override
            public void onCancelled(DatabaseError error) {
                // Handle error
            }
        });
    }
}
```

52. Must implement at least two features of your application, they are related to the core functionality. Describe what main functionality added

-Ability to see what the current temperature is, and what you want the temperature to be at. User has the ability to change the target temperature, and the current temperature will change to what the target is
-rooms will either be vacant or occupied depending on what the database is.
.
53. Describe the main functionality added in this sprint.

-Main functionality is room availability and temperature