Deliverable 3

Smart Blind Add-on
Blind Dev

Vyacheslav Perepelytsya n01133953
Amit Punit n01203930
Chris Janelle Mutuc n01314607
Andrew Fraser n01309442

Table of Contents

Project Description

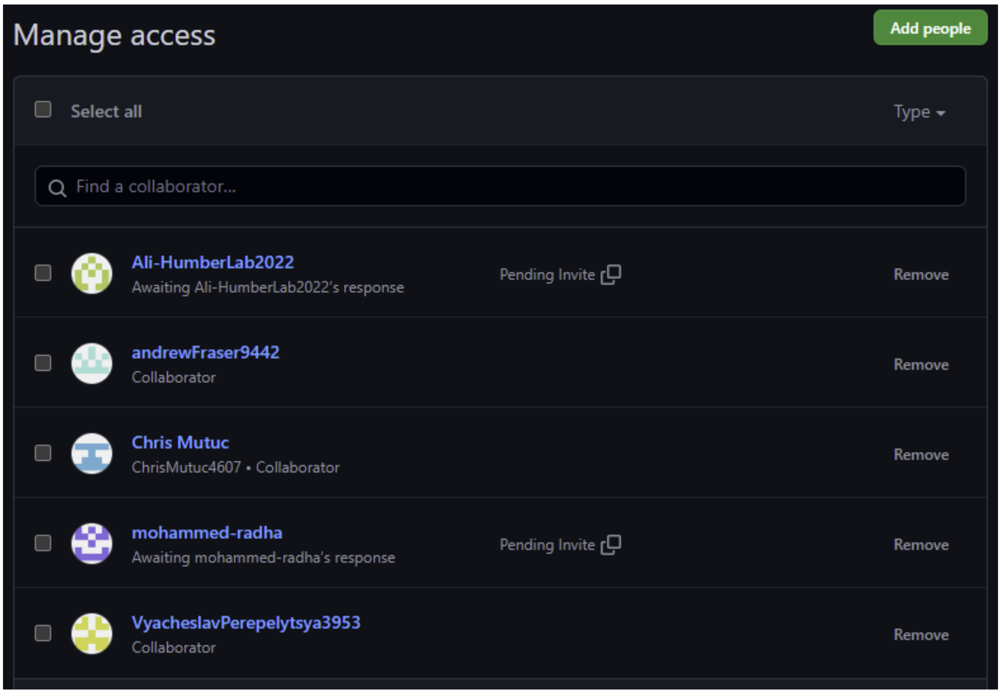
The smart blind addon is a wirelessly controlled device that is to be mounted to a blind existing operation controls in order to control its operation wirelessly. The device needs two things to operate, the app that will display device data and control the device and a connection to a firebase database, in order to exchange data from the app to the device.

Member info and Participation

Name	ID	Signature	Effort
Amit Punit	n01203930	Amit Punit	100%
Andrew Fraser	n01309442	Andrew.f	100%
Chris Janelle Mutuc	n01314607		100%
Vyacheslav Perepelytsya	n01133953		100%

GitHub Project Link

https://github.com/AmitPunit3930/SmartBlindAddon.git



Manage access

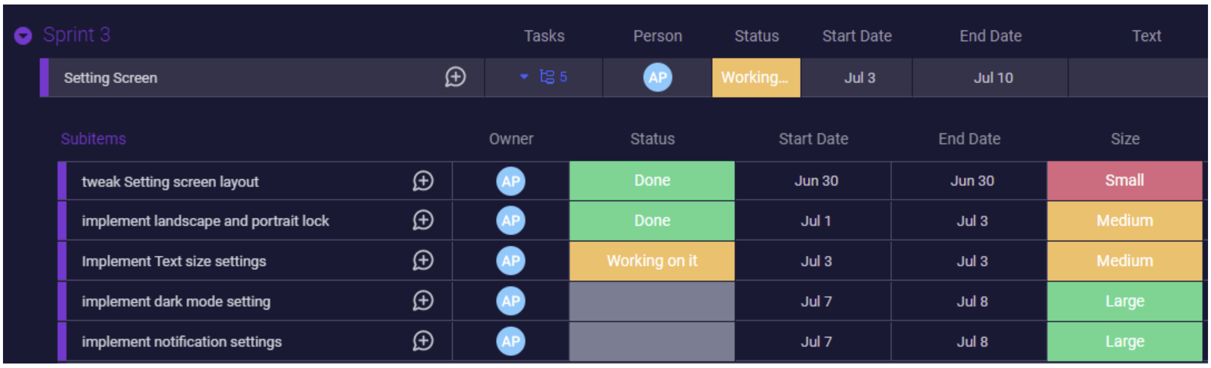
□ Select all

□	Ali-HumberLab2022 Awaiting Ali-HumberLab2022's response	Pending Invite	Remove
□	andrewFraser9442 Collaborator		Remove
□	Chris Mutuc ChrisMutuc4607 • Collaborator		Remove
□	mohammed-radha Awaiting mohammed-radha's response	Pending Invite	Remove
□	VyacheslavPerepelytsya3953 Collaborator		Remove

Sprint goals

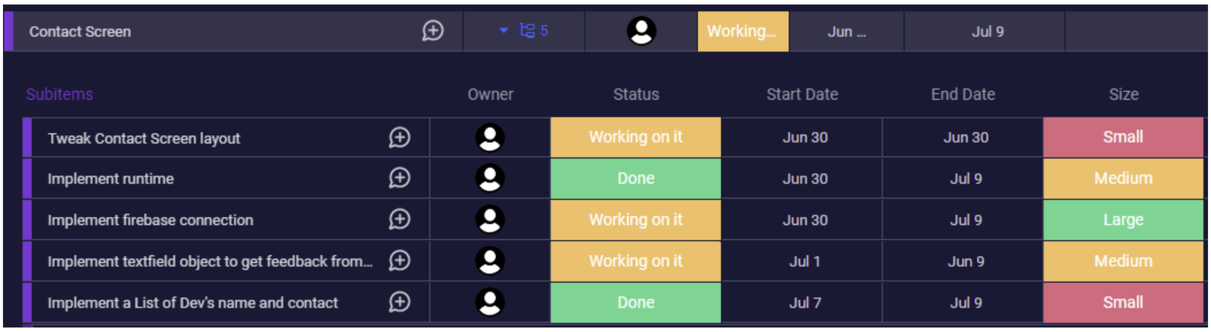
Link to Sprint goals (All sprint tasks completed for Sprint 3.)
https://view.monday.com/2789650579-eb326b7976f0c1eeeba35b004f81cfa4?r=use1

Sprint 3



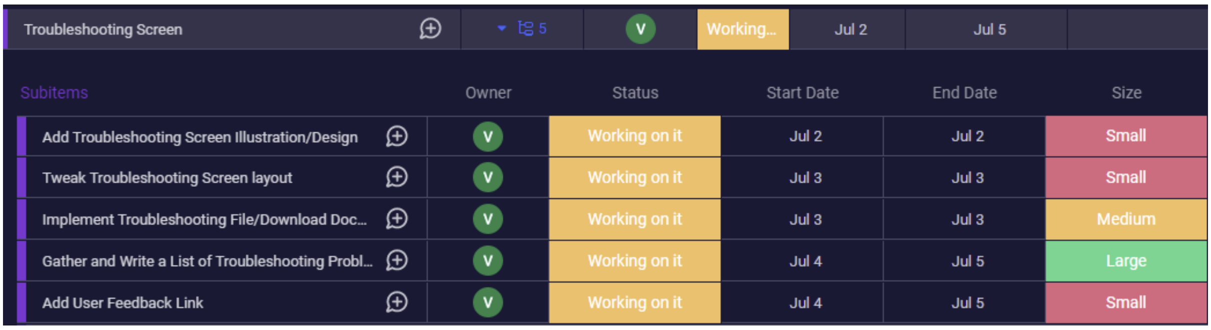
Sprint 3	Tasks	Person	Status	Start Date	End Date	Text
Setting Screen	5	AP	Working...	Jul 3	Jul 10	

Subitems	Owner	Status	Start Date	End Date	Size
tweak Setting screen layout	AP	Done	Jun 30	Jun 30	Small
implement landscape and portrait lock	AP	Done	Jul 1	Jul 3	Medium
Implement Text size settings	AP	Working on it	Jul 3	Jul 3	Medium
implement dark mode setting	AP		Jul 7	Jul 8	Large
implement notification settings	AP		Jul 7	Jul 8	Large



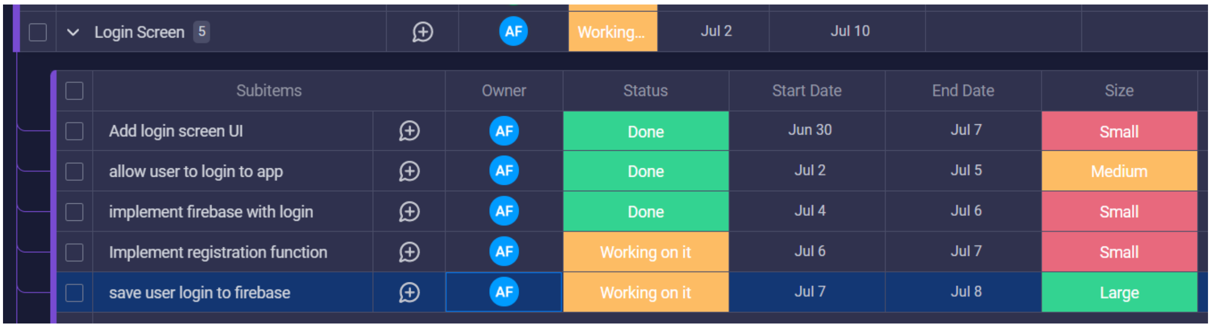
Contact Screen	5	👤	Working...	Jun ...	Jul 9	

Subitems	Owner	Status	Start Date	End Date	Size
Tweak Contact Screen layout	👤	Working on it	Jun 30	Jun 30	Small
Implement runtime	👤	Done	Jun 30	Jul 9	Medium
Implement firebase connection	👤	Working on it	Jun 30	Jul 9	Large
Implement textfield object to get feedback from...	👤	Working on it	Jul 1	Jun 9	Medium
Implement a List of Dev's name and contact	👤	Done	Jul 7	Jul 9	Small



Troubleshooting Screen	5	V	Working...	Jul 2	Jul 5	

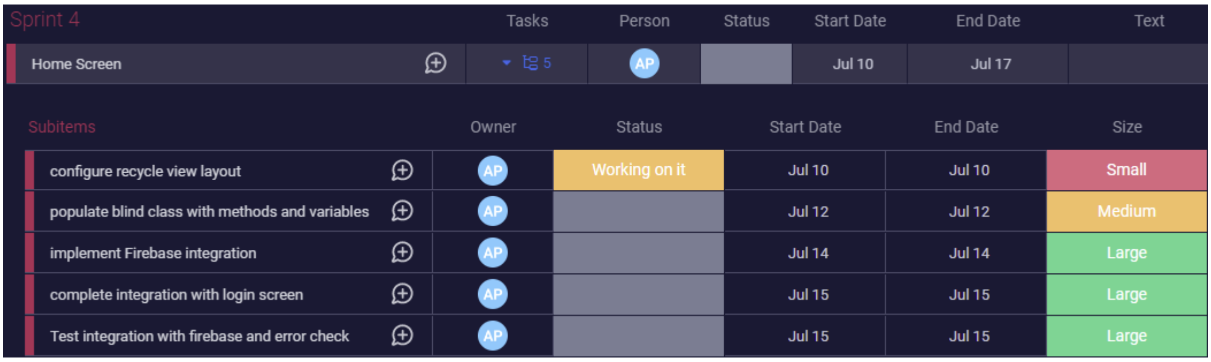
Subitems	Owner	Status	Start Date	End Date	Size
Add Troubleshooting Screen Illustration/Design	V	Working on it	Jul 2	Jul 2	Small
Tweak Troubleshooting Screen layout	V	Working on it	Jul 3	Jul 3	Small
Implement Troubleshooting File/Download Doc...	V	Working on it	Jul 3	Jul 3	Medium
Gather and Write a List of Troubleshooting Probl...	V	Working on it	Jul 4	Jul 5	Large
Add User Feedback Link	V	Working on it	Jul 4	Jul 5	Small



□	Login Screen 5	AF	Working...	Jul 2	Jul 10	

□	Subitems	Owner	Status	Start Date	End Date	Size
□	Add login screen UI	AF	Done	Jun 30	Jul 7	Small
□	allow user to login to app	AF	Done	Jul 2	Jul 5	Medium
□	implement firebase with login	AF	Done	Jul 4	Jul 6	Small
□	Implement registration function	AF	Working on it	Jul 6	Jul 7	Small
□	save user login to firebase	AF	Working on it	Jul 7	Jul 8	Large

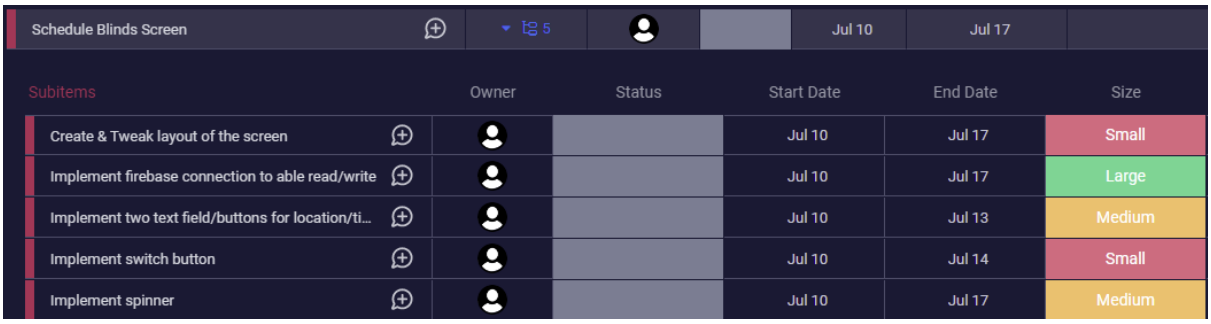
Sprint 4



Sprint 4

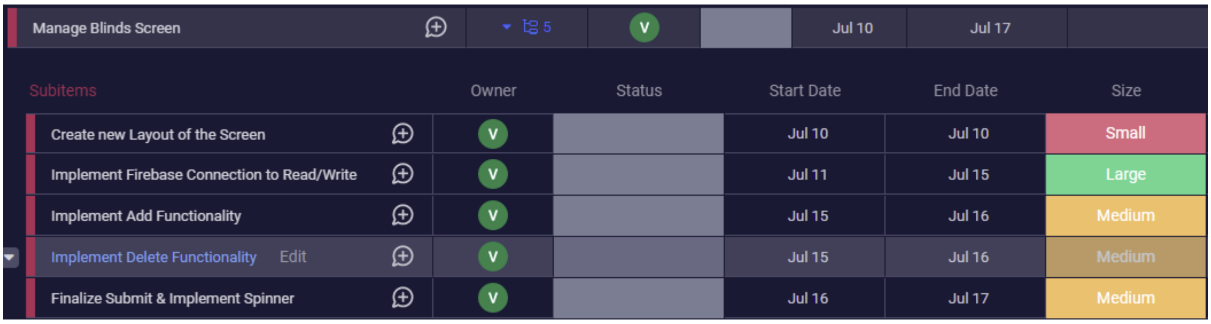
	Tasks	Person	Status	Start Date	End Date	Text
Home Screen	5	AP		Jul 10	Jul 17	

Subitems	Owner	Status	Start Date	End Date	Size
configure recycle view layout	AP	Working on it	Jul 10	Jul 10	Small
populate blind class with methods and variables	AP		Jul 12	Jul 12	Medium
implement Firebase integration	AP		Jul 14	Jul 14	Large
complete integration with login screen	AP		Jul 15	Jul 15	Large
Test integration with firebase and error check	AP		Jul 15	Jul 15	Large



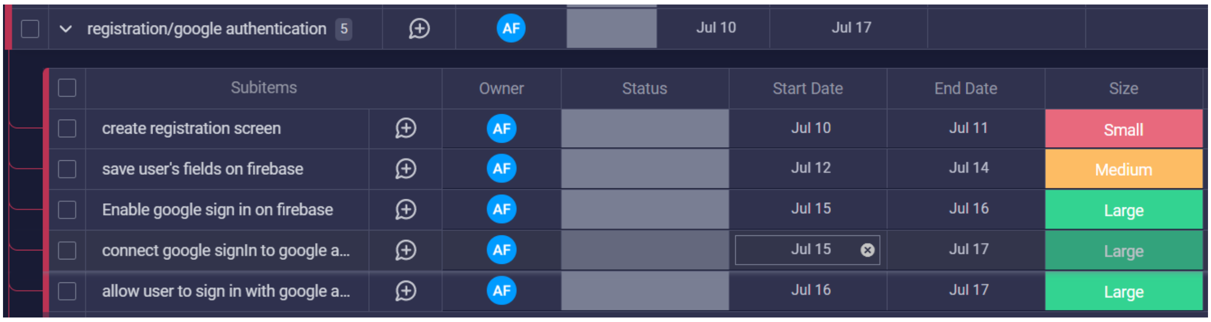
	Tasks	Person	Status	Start Date	End Date	Text
Schedule Blinds Screen	5			Jul 10	Jul 17	

Subitems	Owner	Status	Start Date	End Date	Size
Create & Tweak layout of the screen			Jul 10	Jul 17	Small
Implement firebase connection to able read/write			Jul 10	Jul 17	Large
Implement two text field/buttons for location/ti...			Jul 10	Jul 13	Medium
Implement switch button			Jul 10	Jul 14	Small
Implement spinner			Jul 10	Jul 17	Medium



	Tasks	Person	Status	Start Date	End Date	Text
Manage Blinds Screen	5	V		Jul 10	Jul 17	

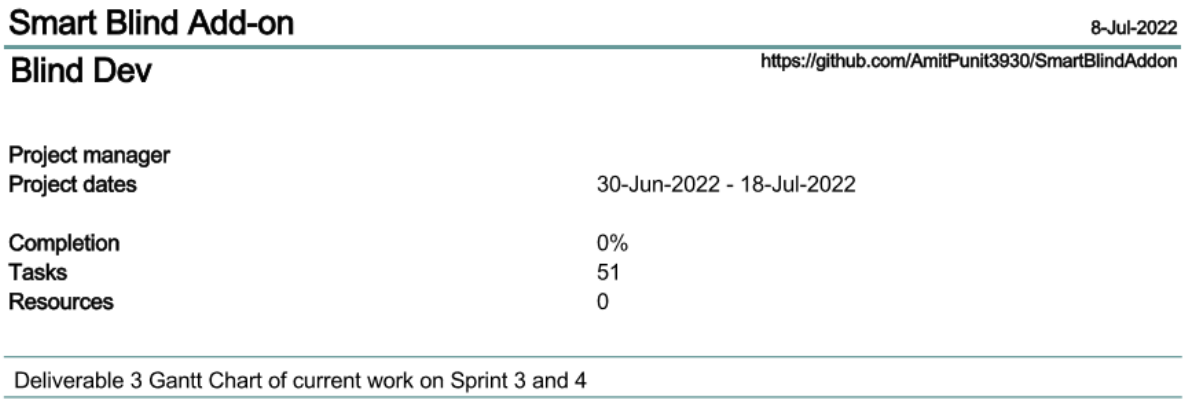
Subitems	Owner	Status	Start Date	End Date	Size
Create new Layout of the Screen	V		Jul 10	Jul 10	Small
Implement Firebase Connection to Read/Write	V		Jul 11	Jul 15	Large
Implement Add Functionality	V		Jul 15	Jul 16	Medium
Implement Delete Functionality Edit	V		Jul 15	Jul 16	Medium
Finalize Submit & Implement Spinner	V		Jul 16	Jul 17	Medium



□ registration/google authentication 5	AF		Jul 10	Jul 17	

□ Subitems	Owner	Status	Start Date	End Date	Size
□ create registration screen	AF		Jul 10	Jul 11	Small
□ save user's fields on firebase	AF		Jul 12	Jul 14	Medium
□ Enable google sign in on firebase	AF		Jul 15	Jul 16	Large
□ connect google signIn to google a...	AF		Jul 15	Jul 17	Large
□ allow user to sign in with google a...	AF		Jul 16	Jul 17	Large

Gantt Chart (Made with GanttProject)



Smart Blind Add-on

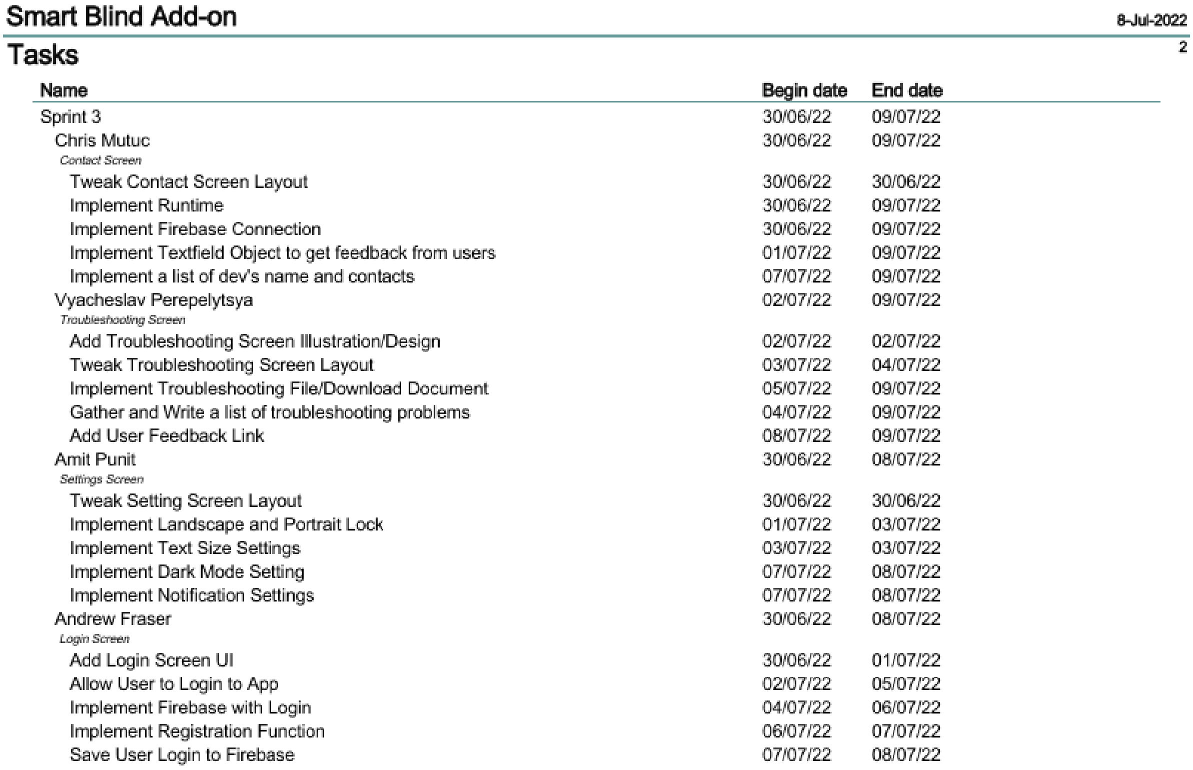
8-Jul-2022

Blind Dev

https://github.com/AmitPunit3930/SmartBlindAddon

Project manager	
Project dates	30-Jun-2022 - 18-Jul-2022
Completion	0%
Tasks	51
Resources	0

Deliverable 3 Gantt Chart of current work on Sprint 3 and 4



Smart Blind Add-on

8-Jul-2022

Tasks

2

Name	Begin date	End date
Sprint 3	30/06/22	09/07/22
Chris Mutuc	30/06/22	09/07/22
Contact Screen		
Tweak Contact Screen Layout	30/06/22	30/06/22
Implement Runtime	30/06/22	09/07/22
Implement Firebase Connection	30/06/22	09/07/22
Implement Textfield Object to get feedback from users	01/07/22	09/07/22
Implement a list of dev's name and contacts	07/07/22	09/07/22
Vyacheslav Perepelytsya	02/07/22	09/07/22
Troubleshooting Screen		
Add Troubleshooting Screen Illustration/Design	02/07/22	02/07/22
Tweak Troubleshooting Screen Layout	03/07/22	04/07/22
Implement Troubleshooting File/Download Document	05/07/22	09/07/22
Gather and Write a list of troubleshooting problems	04/07/22	09/07/22
Add User Feedback Link	08/07/22	09/07/22
Amit Punit	30/06/22	08/07/22
Settings Screen		
Tweak Setting Screen Layout	30/06/22	30/06/22
Implement Landscape and Portrait Lock	01/07/22	03/07/22
Implement Text Size Settings	03/07/22	03/07/22
Implement Dark Mode Setting	07/07/22	08/07/22
Implement Notification Settings	07/07/22	08/07/22
Andrew Fraser	30/06/22	08/07/22
Login Screen		
Add Login Screen UI	30/06/22	01/07/22
Allow User to Login to App	02/07/22	05/07/22
Implement Firebase with Login	04/07/22	06/07/22
Implement Registration Function	06/07/22	07/07/22
Save User Login to Firebase	07/07/22	08/07/22

Smart Blind Add-on

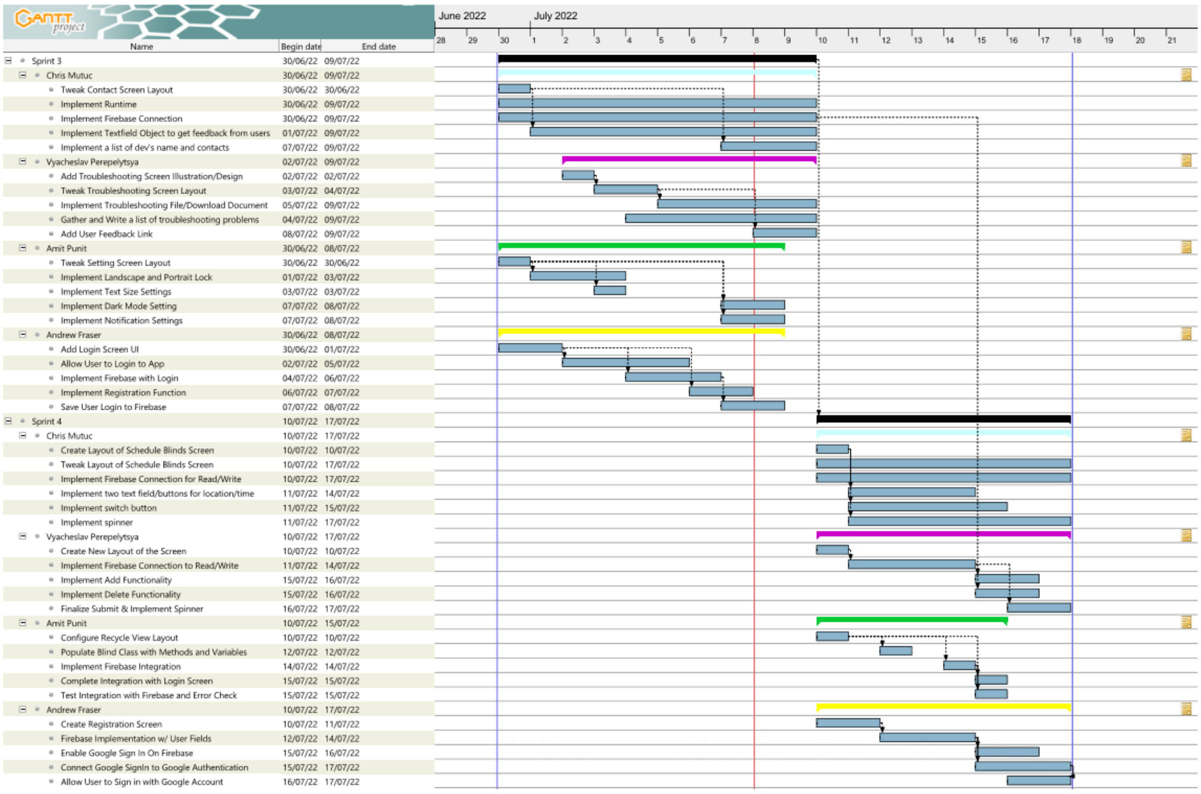
8-Jul-2022

Tasks

Name	Begin date	End date
Sprint 4	10/07/22	17/07/22
Chris Mutuc	10/07/22	17/07/22
Schedule Blinds Screen		
Create Layout of Schedule Blinds Screen	10/07/22	10/07/22
Tweak Layout of Schedule Blinds Screen	10/07/22	17/07/22
Implement Firebase Connection for Read/Write	10/07/22	17/07/22
Implement two text field/buttons for location/time	11/07/22	14/07/22
Implement switch button	11/07/22	15/07/22
Implement spinner	11/07/22	17/07/22
Vyacheslav Perepelytsya	10/07/22	17/07/22
Manage Blinds		
Create New Layout of the Screen	10/07/22	10/07/22
Implement Firebase Connection to Read/Write	11/07/22	14/07/22
Implement Add Functionality	15/07/22	16/07/22
Implement Delete Functionality	15/07/22	16/07/22
Finalize Submit & Implement Spinner	16/07/22	17/07/22
Amit Punit	10/07/22	15/07/22
Home Screen		
Configure Recycle View Layout	10/07/22	10/07/22
Populate Blind Class with Methods and Variables	12/07/22	12/07/22
Implement Firebase Integration	14/07/22	14/07/22
Complete Integration with Login Screen	15/07/22	15/07/22
Test Integration with Firebase and Error Check	15/07/22	15/07/22
Andrew Fraser	10/07/22	17/07/22
Registration/Google Authentication		
Create Registration Screen	10/07/22	11/07/22
Firebase Implementation w/ User Fields	12/07/22	14/07/22
Enable Google Sign In On Firebase	15/07/22	16/07/22
Connect Google SignIn to Google Authentication	15/07/22	17/07/22
Allow User to Sign in with Google Account	16/07/22	17/07/22

Smart Blind Add-on

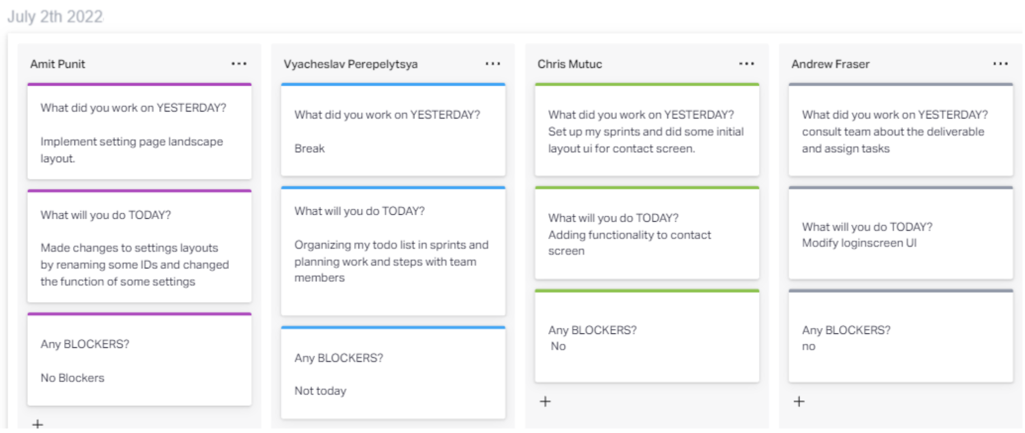
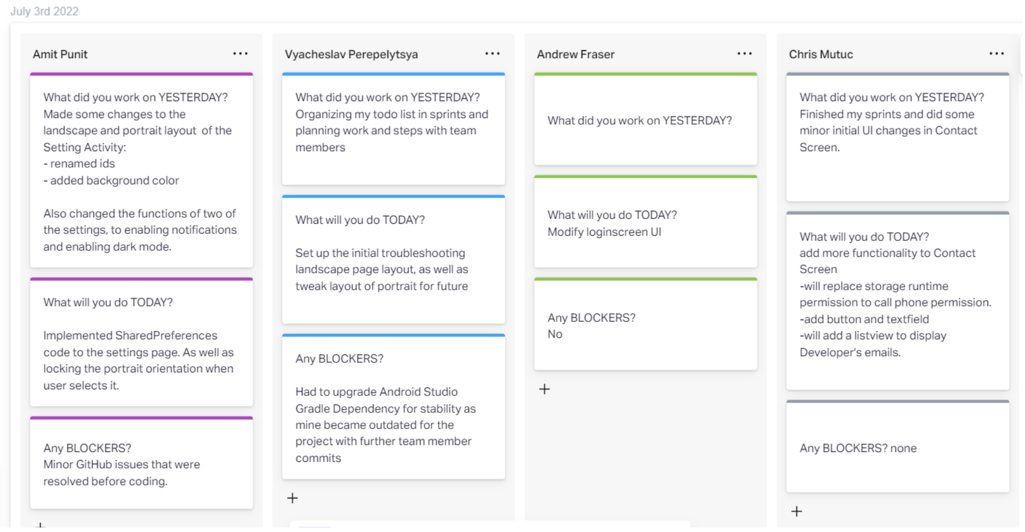
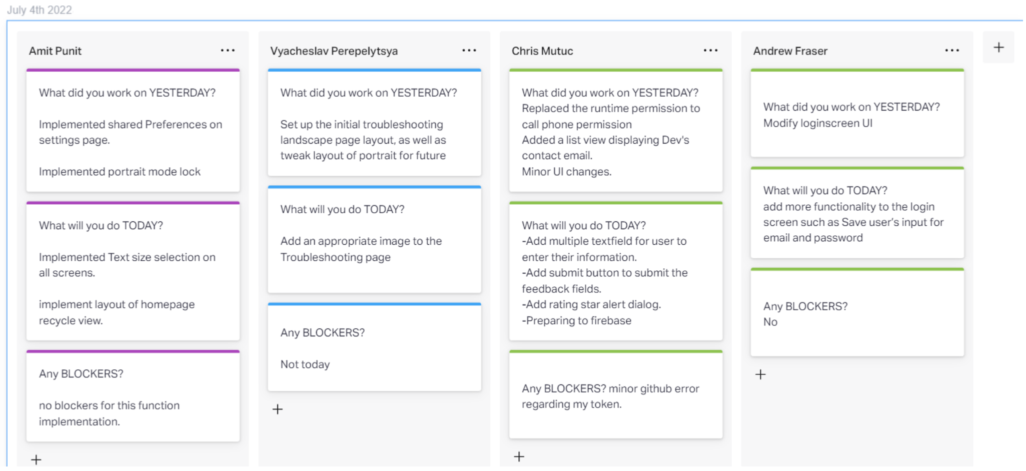
Gantt Chart

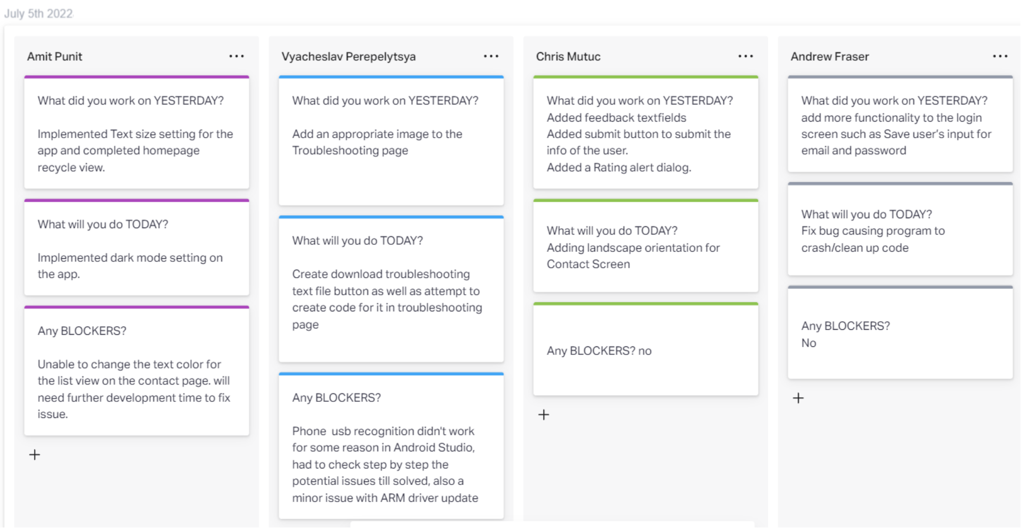
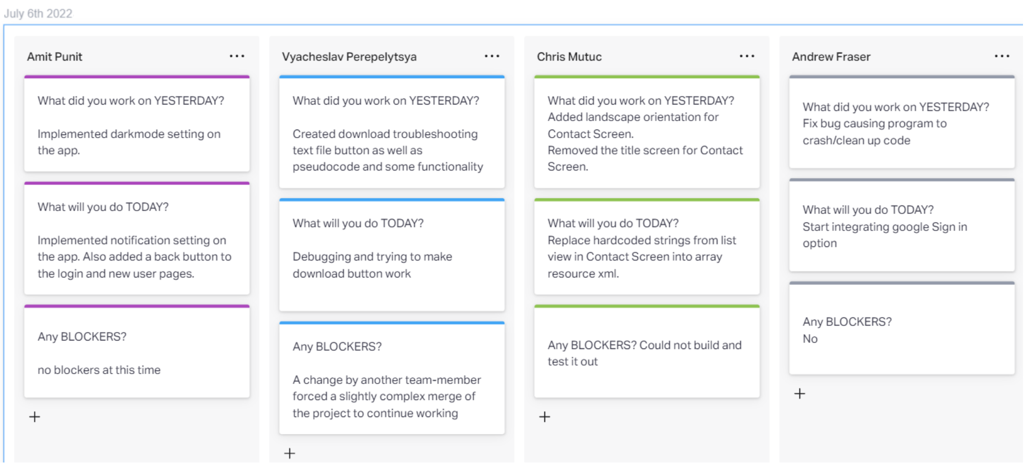
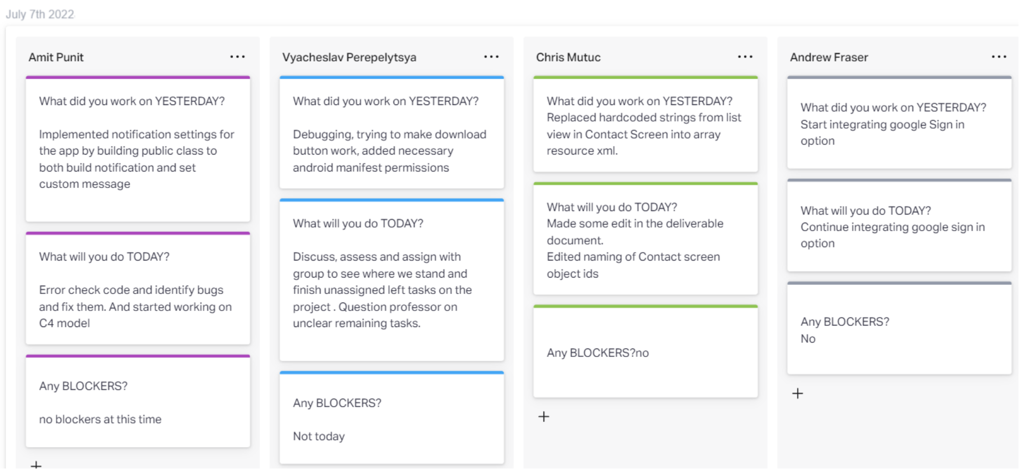


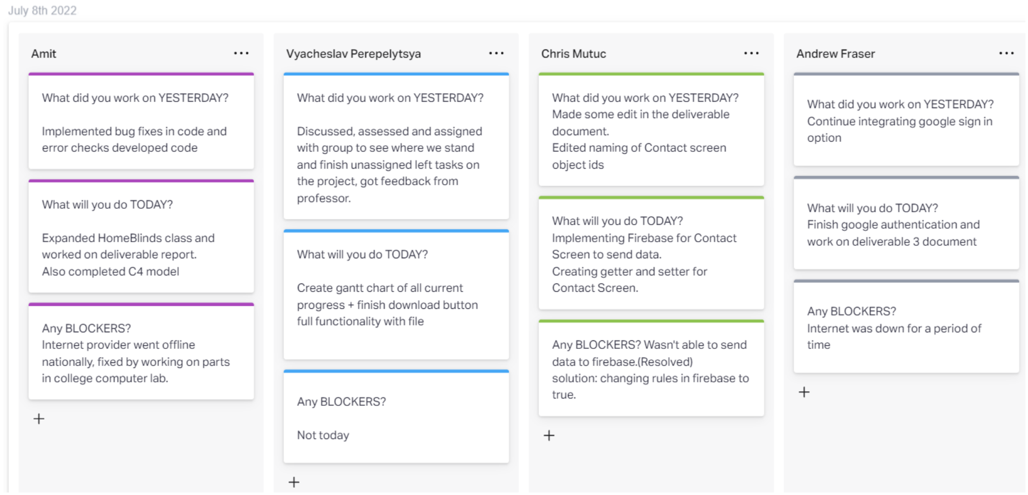
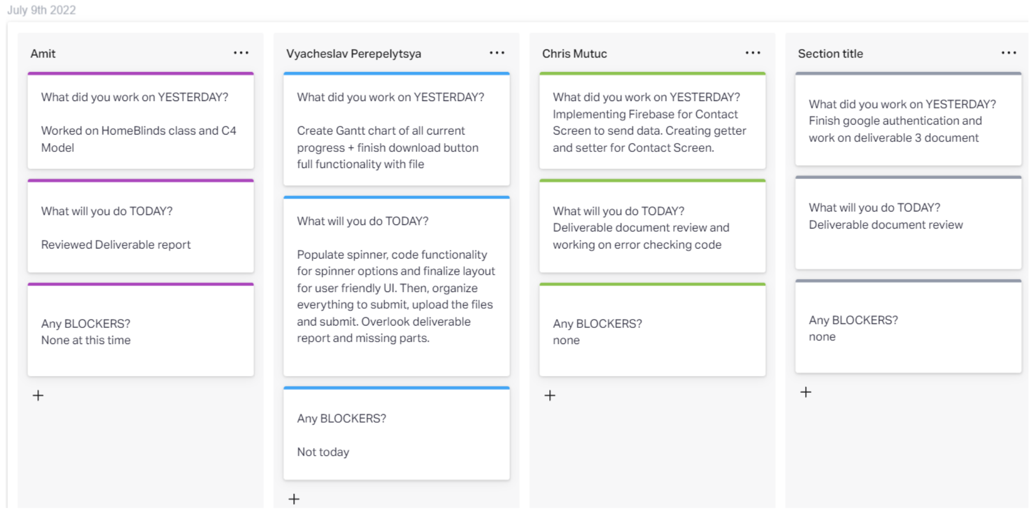
Name	Begin date	End date
Sprint 3	30/06/22	09/07/22
Chris Mutuc	30/06/22	09/07/22
Tweak Contact Screen Layout	30/06/22	30/06/22
Implement Runtime	30/06/22	09/07/22
Implement Firebase Connection	30/06/22	09/07/22
Implement Textfield Object to get feedback from users	01/07/22	09/07/22
Implement a list of dev's name and contacts	07/07/22	09/07/22
Vyacheslav Perepelytsya	02/07/22	09/07/22
Add Troubleshooting Screen Illustration/Design	02/07/22	02/07/22
Tweak Troubleshooting Screen Layout	03/07/22	04/07/22
Implement Troubleshooting File/Download Document	05/07/22	09/07/22
Gather and Write a list of troubleshooting problems	04/07/22	09/07/22
Add User Feedback Link	08/07/22	09/07/22
Amit Punit	30/06/22	08/07/22
Tweak Setting Screen Layout	30/06/22	30/06/22
Implement Landscape and Portrait Lock	01/07/22	03/07/22
Implement Text Size Settings	03/07/22	03/07/22
Implement Dark Mode Setting	07/07/22	08/07/22
Implement Notification Settings	07/07/22	08/07/22
Andrew Fraser	30/06/22	08/07/22
Add Login Screen UI	30/06/22	01/07/22
Allow User to Login to App	02/07/22	05/07/22
Implement Firebase with Login	04/07/22	06/07/22
Implement Registration Function	06/07/22	07/07/22
Save User Login to Firebase	07/07/22	08/07/22
Sprint 4	10/07/22	17/07/22
Chris Mutuc	10/07/22	17/07/22
Create Layout of Schedule Blinds Screen	10/07/22	10/07/22
Tweak Layout of Schedule Blinds Screen	10/07/22	17/07/22
Implement Firebase Connection for Read/Write	10/07/22	17/07/22
Implement two text field/buttons for location/time	11/07/22	14/07/22
Implement switch button	11/07/22	15/07/22
Implement spinner	11/07/22	17/07/22
Vyacheslav Perepelytsya	10/07/22	17/07/22
Create New Layout of the Screen	10/07/22	10/07/22
Implement Firebase Connection to Read/Write	11/07/22	14/07/22
Implement Add Functionality	15/07/22	16/07/22
Implement Delete Functionality	15/07/22	16/07/22
Finalize Submit & Implement Spinner	16/07/22	17/07/22
Amit Punit	10/07/22	15/07/22
Configure Recycle View Layout	10/07/22	10/07/22
Populate Blind Class with Methods and Variables	12/07/22	12/07/22
Implement Firebase Integration	14/07/22	14/07/22
Complete Integration with Login Screen	15/07/22	15/07/22
Test Integration with Firebase and Error Check	15/07/22	15/07/22
Andrew Fraser	10/07/22	17/07/22
Create Registration Screen	10/07/22	11/07/22
Firebase Implementation w/ User Fields	12/07/22	14/07/22
Enable Google Sign In On Firebase	15/07/22	16/07/22
Connect Google SignIn to Google Authentication	15/07/22	17/07/22
Allow User to Sign in with Google Account	16/07/22	17/07/22

Daily Standup

https://amitpunit779464.invisionapp.com/freehand/Daily-Standups-ZWeshMsvH?dsid_h=c298c563c72daf545bbe4504b805def04bc411fced80797da57e79a372e4407d&uid_h=b0e866ad1232fc7f2b396aea082db59a3d7e400a8593065a4e79a4b15763954e

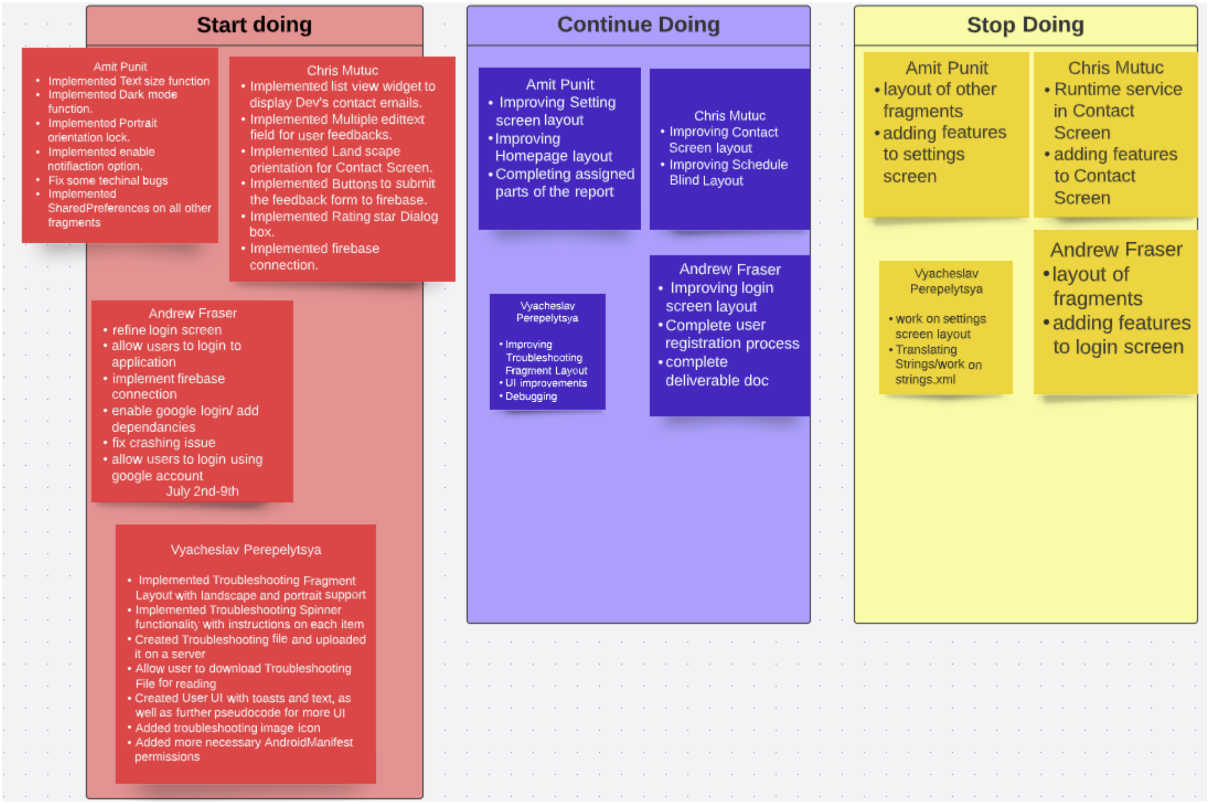
Date	Screenshot
July 2nd 2022	 July 2th 2022 **Amit Punit** What did you work on YESTERDAY? Implement setting page landscape layout. What will you do TODAY? Made changes to settings layouts by renaming some IDs and changed the function of some settings Any BLOCKERS? No Blockers **Vyacheslav Perepelytsya** What did you work on YESTERDAY? Break What will you do TODAY? Organizing my todo list in sprints and planning work and steps with team members Any BLOCKERS? Not today **Chris Mutuc** What did you work on YESTERDAY? Set up my sprints and did some initial layout ui for contact screen. What will you do TODAY? Adding functionality to contact screen Any BLOCKERS? No **Andrew Fraser** What did you work on YESTERDAY? consult team about the deliverable and assign tasks What will you do TODAY? Modify loginscreen UI Any BLOCKERS? no
July 3rd, 2022	 July 3rd 2022 **Amit Punit** What did you work on YESTERDAY? Made some changes to the landscape and portrait layout of the Setting Activity: - renamed ids - added background color Also changed the functions of two of the settings, to enabling notifications and enabling dark mode. What will you do TODAY? Implemented SharedPreferences code to the settings page. As well as locking the portrait orientation when user selects it. Any BLOCKERS? Minor GitHub issues that were resolved before coding. **Vyacheslav Perepelytsya** What did you work on YESTERDAY? Organizing my todo list in sprints and planning work and steps with team members What will you do TODAY? Set up the initial troubleshooting landscape page layout, as well as tweak layout of portrait for future Any BLOCKERS? Had to upgrade Android Studio Gradle Dependency for stability as mine became outdated for the project with further team member commits **Andrew Fraser** What did you work on YESTERDAY? What will you do TODAY? Modify loginscreen UI Any BLOCKERS? No **Chris Mutuc** What did you work on YESTERDAY? Finished my sprints and did some minor initial UI changes in Contact Screen. What will you do TODAY? add more functionality to Contact Screen -will replace storage runtime permission to call phone permission. -add button and textfield -will add a listview to display Developer's emails. Any BLOCKERS? none
July 4th, 2022	 July 4th 2022 **Amit Punit** What did you work on YESTERDAY? Implemented shared Preferences on settings page. Implemented portrait mode lock What will you do TODAY? Implemented Text size selection on all screens. implement layout of homepage recycle view. Any BLOCKERS? no blockers for this function implementation. **Vyacheslav Perepelytsya** What did you work on YESTERDAY? Set up the initial troubleshooting landscape page layout, as well as tweak layout of portrait for future What will you do TODAY? Add an appropriate image to the Troubleshooting page Any BLOCKERS? Not today **Chris Mutuc** What did you work on YESTERDAY? Replaced the runtime permission to call phone permission Added a list view displaying Dev's contact email. Minor UI changes. What will you do TODAY? -Add multiple textfield for user to enter their information. -Add submit button to submit the feedback fields. -Add rating star alert dialog. -Preparing to firebase Any BLOCKERS? minor github error regarding my token. **Andrew Fraser** What did you work on YESTERDAY? Modify loginscreen UI What will you do TODAY? add more functionality to the login screen such as Save user's input for email and password Any BLOCKERS? No

July 5th, 2022	 July 5th 2022 Amit Punit What did you work on YESTERDAY? Implemented Text size setting for the app and completed homepage recycle view. What will you do TODAY? Implemented dark mode setting on the app. Any BLOCKERS? Unable to change the text color for the list view on the contact page. will need further development time to fix issue. Vyacheslav Perepelytsya What did you work on YESTERDAY? Add an appropriate image to the Troubleshooting page What will you do TODAY? Create download troubleshooting text file button as well as attempt to create code for it in troubleshooting page Any BLOCKERS? Phone usb recognition didn't work for some reason in Android Studio, had to check step by step the potential issues till solved, also a minor issue with ARM driver update Chris Mutuc What did you work on YESTERDAY? Added feedback textfields Added submit button to submit the info of the user. Added a Rating alert dialog. What will you do TODAY? Adding landscape orientation for Contact Screen Any BLOCKERS? no Andrew Fraser What did you work on YESTERDAY? add more functionality to the login screen such as Save user's input for email and password What will you do TODAY? Fix bug causing program to crash/clean up code Any BLOCKERS? No
July 6th, 2022	 July 6th 2022 Amit Punit What did you work on YESTERDAY? Implemented darkmode setting on the app. What will you do TODAY? Implemented notification setting on the app. Also added a back button to the login and new user pages. Any BLOCKERS? no blockers at this time Vyacheslav Perepelytsya What did you work on YESTERDAY? Created download troubleshooting text file button as well as pseudocode and some functionality What will you do TODAY? Debugging and trying to make download button work Any BLOCKERS? A change by another team-member forced a slightly complex merge of the project to continue working Chris Mutuc What did you work on YESTERDAY? Added landscape orientation for Contact Screen. Removed the title screen for Contact Screen. What will you do TODAY? Replace hardcoded strings from list view in Contact Screen into array resource xml. Any BLOCKERS? Could not build and test it out Andrew Fraser What did you work on YESTERDAY? Fix bug causing program to crash/clean up code What will you do TODAY? Start integrating google Sign in option Any BLOCKERS? No
July 7th, 2022	 July 7th 2022 Amit Punit What did you work on YESTERDAY? Implemented notification settings for the app by building public class to both build notification and set custom message What will you do TODAY? Error check code and identify bugs and fix them. And started working on C4 model Any BLOCKERS? no blockers at this time Vyacheslav Perepelytsya What did you work on YESTERDAY? Debugging, trying to make download button work, added necessary android manifest permissions What will you do TODAY? Discuss, assess and assign with group to see where we stand and finish unassigned left tasks on the project . Question professor on unclear remaining tasks. Any BLOCKERS? Not today Chris Mutuc What did you work on YESTERDAY? Replaced hardcoded strings from list view in Contact Screen into array resource xml. What will you do TODAY? Made some edit in the deliverable document. Edited naming of Contact screen object ids Any BLOCKERS?no Andrew Fraser What did you work on YESTERDAY? Start integrating google Sign in option What will you do TODAY? Continue integrating google sign in option Any BLOCKERS? No

July 8th, 2022	 July 8th 2022 Amit What did you work on YESTERDAY? Implemented bug fixes in code and error checks developed code What will you do TODAY? Expanded HomeBlinds class and worked on deliverable report. Also completed C4 model Any BLOCKERS? Internet provider went offline nationally, fixed by working on parts in college computer lab. Vyacheslav Perepelytsya What did you work on YESTERDAY? Discussed, assessed and assigned with group to see where we stand and finish unassigned left tasks on the project, got feedback from professor. What will you do TODAY? Create gantt chart of all current progress + finish download button full functionality with file Any BLOCKERS? Not today Chris Mutuc What did you work on YESTERDAY? Made some edit in the deliverable document. Edited naming of Contact screen object ids What will you do TODAY? Implementing Firebase for Contact Screen to send data. Creating getter and setter for Contact Screen. Any BLOCKERS? Wasn't able to send data to firebase.(Resolved) solution: changing rules in firebase to true. Andrew Fraser What did you work on YESTERDAY? Continue integrating google sign in option What will you do TODAY? Finish google authentication and work on deliverable 3 document Any BLOCKERS? Internet was down for a period of time
July 9th, 2022	 July 9th 2022 Amit What did you work on YESTERDAY? Worked on HomeBlinds class and C4 Model What will you do TODAY? Reviewed Deliverable report Any BLOCKERS? None at this time Vyacheslav Perepelytsya What did you work on YESTERDAY? Create Gantt chart of all current progress + finish download button full functionality with file What will you do TODAY? Populate spinner, code functionality for spinner options and finalize layout for user friendly UI. Then, organize everything to submit, upload the files and submit. Overlook deliverable report and missing parts. Any BLOCKERS? Not today Chris Mutuc What did you work on YESTERDAY? Implementing Firebase for Contact Screen to send data. Creating getter and setter for Contact Screen. What will you do TODAY? Deliverable document review and working on error checking code Any BLOCKERS? none Section title What did you work on YESTERDAY? Finish google authentication and work on deliverable 3 document What will you do TODAY? Deliverable document review Any BLOCKERS? none

Sprint Retrospectives

https://lucid.app/lucidspark/2ad1ab5c-fc61-4559-957d-83cf9e6570da/edit?viewport_loc=-681%2C-412%2C2786%2C1359%2C0_0&invitationId=inv_8568d1cf-90e9-47f4-bdee-2e224bc26185#



Start doing

Amit Punit

- Implemented Text size function
- Implemented Dark mode function.
- Implemented Portrait orientation lock.
- Implemented enable notifiaction option.
- Fix some techinal bugs
- Implemented SharedPreferences on all other fragments

Chris Mutuc

- Implemented list view widget to display Dev's contact emails.
- Implemented Multiple edittext field for user feedbacks.
- Implemented Land scape orientation for Contact Screen.
- Implemented Buttons to submit the feedback form to firebase.
- Implemented Rating star Dialog box.
- Implemented firebase connection.

Andrew Fraser

- refine login screen
- allow users to login to application
- implement firebase connection
- enable google login/ add dependancies
- fix crashing issue
- allow users to login using google account

July 2nd-9th

Vyacheslav Perepelytsya

- Implemented Troubleshooting Fragment Layout with landscape and portrait support
- Implemented Troubleshooting Spinner functionality with instructions on each item
- Created Troubleshooting file and uploaded it on a server
- Allow user to download Troubleshooting File for reading
- Created User UI with toasts and text, as well as further pseudocode for more UI
- Added troubleshooting image icon
- Added more necessary AndroidManifest permissions

Continue Doing

Amit Punit

- Improving Setting screen layout
- Improving Homepage layout
- Completing assigned parts of the report

Chris Mutuc

- Improving Contact Screen layout
- Improving Schedule Blind Layout

Vyacheslav Perepelytsya

- Improving Troubleshooting Fragment Layout
- UI improvements
- Debugging

Andrew Fraser

- Improving login screen layout
- Complete user registration process
- complete deliverable doc

Stop Doing

Amit Punit

- layout of other fragments
- adding features to settings screen

Chris Mutuc

- Runtime service in Contact Screen
- adding features to Contact Screen

Vyacheslav Perepelytsya

- work on settings screen layout
- Translating Strings/work on strings.xml

Andrew Fraser

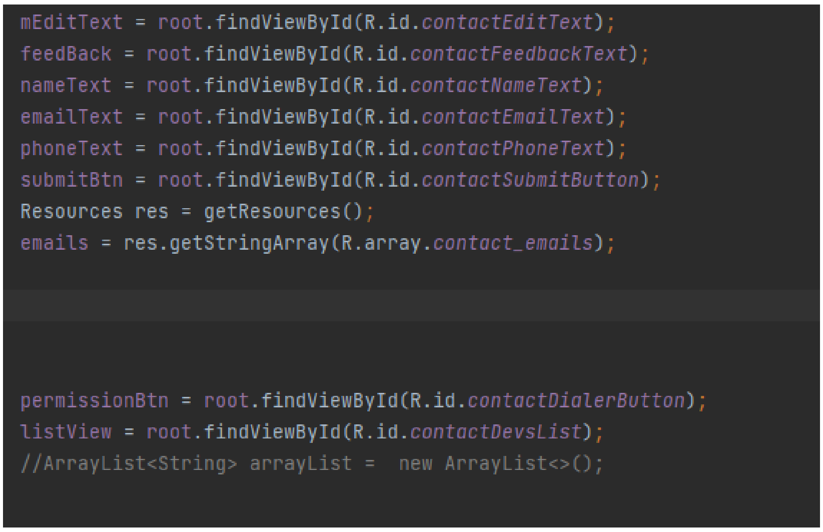
- layout of fragments
- adding features to login screen

Design Principles

Naming Object IDs

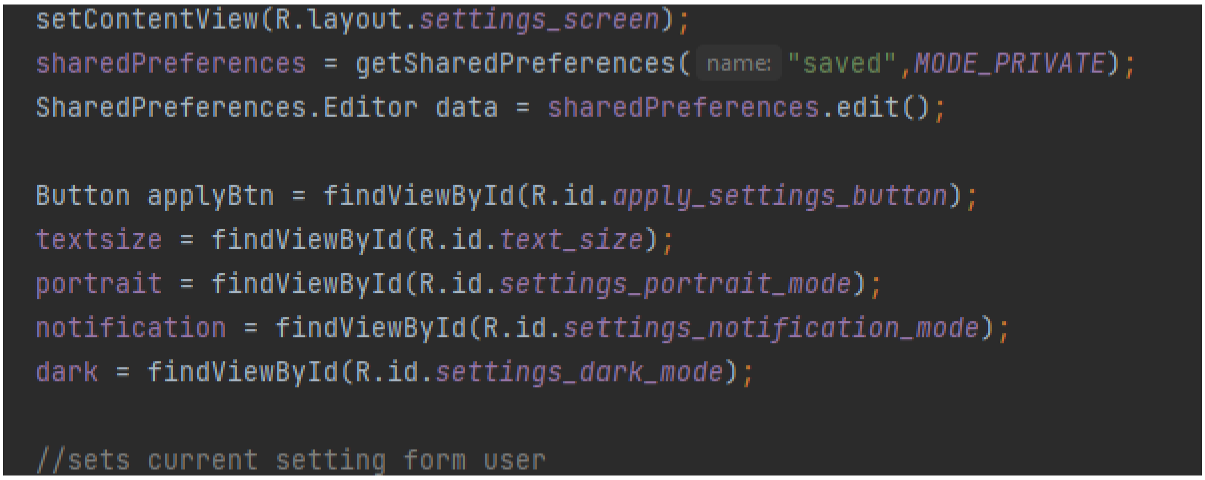
In our team we develop a standard principle of naming our object id's in respect of their screen. Like for example notice the images below the naming of the object id will always have the screen/fragment/activity name. We did this to avoid misusing the object to a different screen than its intended.

Contact Screen Objects



```
mEditText = root.findViewById(R.id.contactEditText);
feedBack = root.findViewById(R.id.contactFeedbackText);
nameText = root.findViewById(R.id.contactNameText);
emailText = root.findViewById(R.id.contactEmailText);
phoneText = root.findViewById(R.id.contactPhoneText);
submitBtn = root.findViewById(R.id.contactSubmitButton);
Resources res = getResources();
emails = res.getStringArray(R.array.contact_emails);



permissionBtn = root.findViewById(R.id.contactDialerButton);
listView = root.findViewById(R.id.contactDevsList);
//ArrayList<String> arrayList =  new ArrayList<>();
```

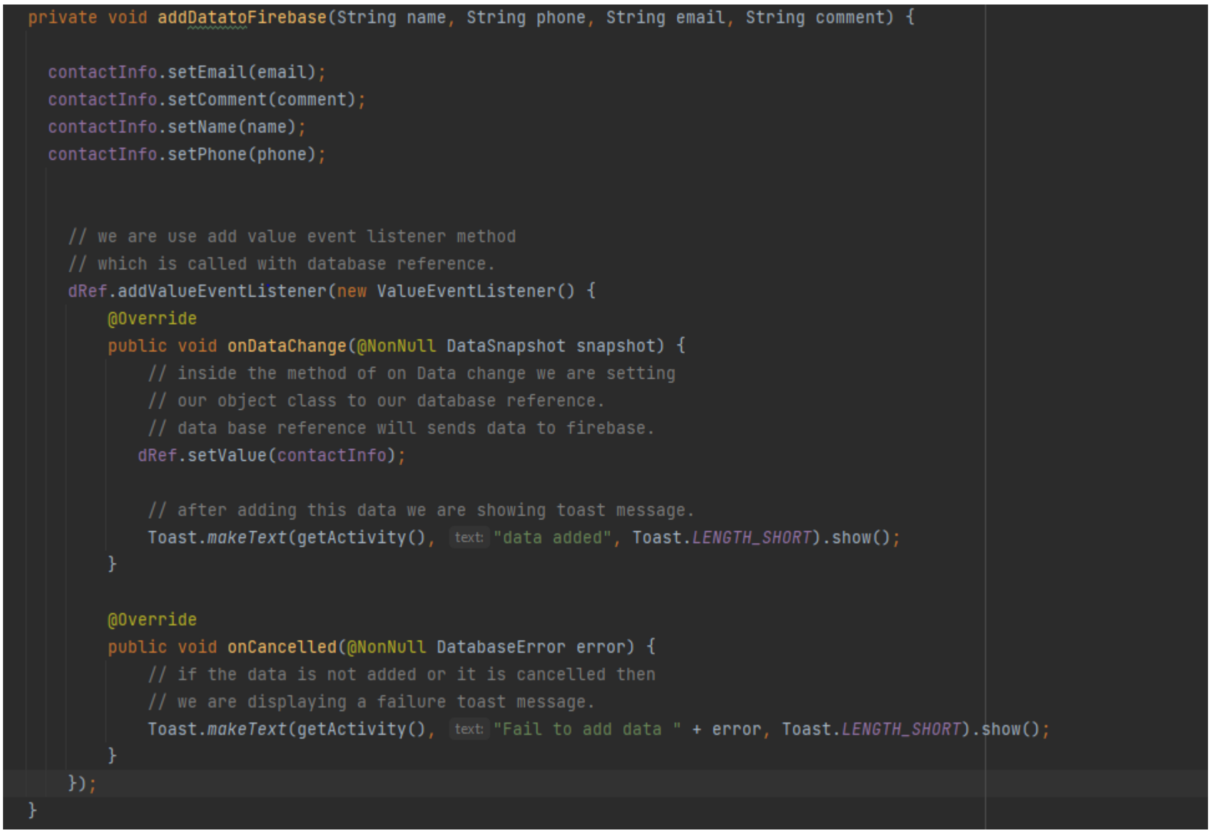
Setting Screen Objects



```
setContentView(R.layout.settings_screen);
sharedPreferences = getSharedPreferences( name: "saved",MODE_PRIVATE);
SharedPreferences.Editor data = sharedPreferences.edit();

Button applyBtn = findViewById(R.id.apply_settings_button);
textsize = findViewById(R.id.text_size);
portrait = findViewById(R.id.settings_portrait_mode);
notification = findViewById(R.id.settings_notification_mode);
dark = findViewById(R.id.settings_dark_mode);

//sets current setting form user
```

Comments

We usually add comments in every complex method just like the figure below. It is usually a brief explanation of how the method will work and how it's implemented in respect of the screen's functionality. We try not to write a lot of comments because it could cause a distraction to whom is reading the code.



```
private void addDatatoFirebase(String name, String phone, String email, String comment) {

  contactInfo.setEmail(email);
  contactInfo.setComment(comment);
  contactInfo.setName(name);
  contactInfo.setPhone(phone);


    // we are use add value event listener method
    // which is called with database reference.
    dRef.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            // inside the method of on Data change we are setting
            // our object class to our database reference.
            // data base reference will sends data to firebase.
           dRef.setValue(contactInfo);

            // after adding this data we are showing toast message.
            Toast.makeText(getActivity(),  text: "data added", Toast.LENGTH_SHORT).show();
        }

        @Override
        public void onCancelled(@NonNull DatabaseError error) {
            // if the data is not added or it is cancelled then
            // we are displaying a failure toast message.
            Toast.makeText(getActivity(),  text: "Fail to add data " + error, Toast.LENGTH_SHORT).show();
        }
    });
}
```

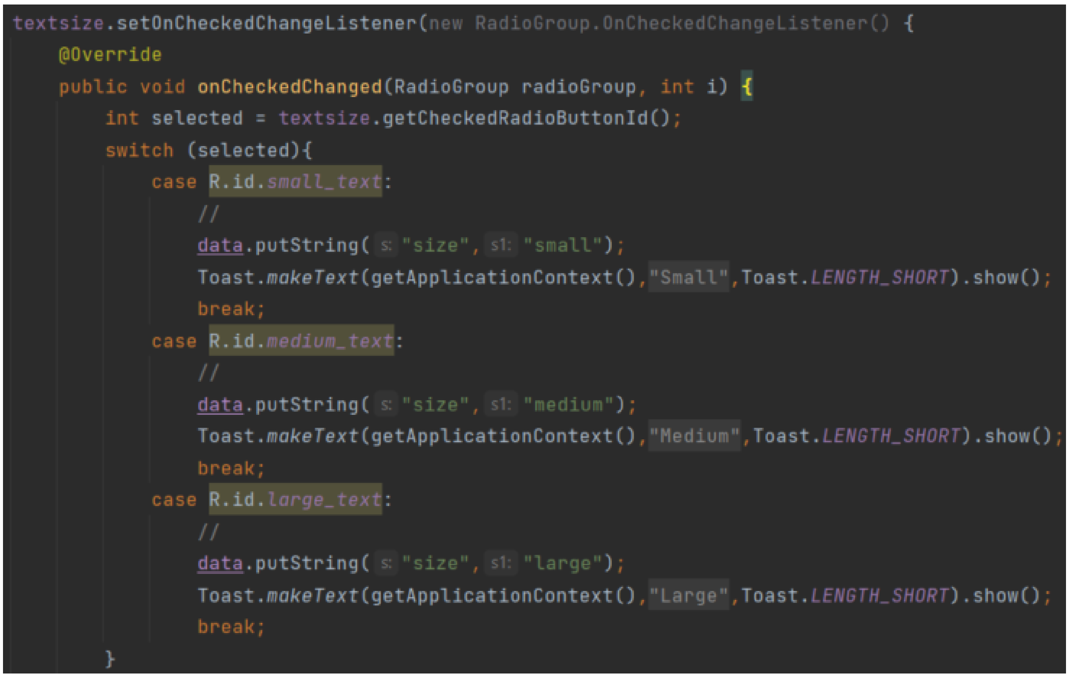
Design Patterns

Dependency Injection

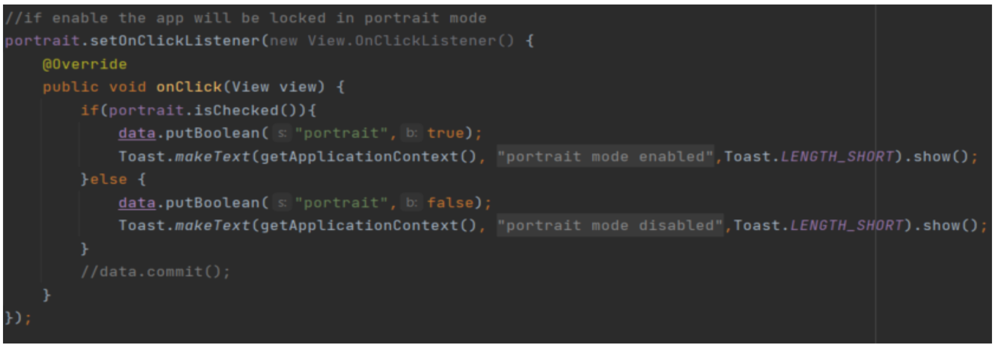
The dependency Injections creation pattern is used by the shared preferences object in the Smart Blinds App. This is used to implement user settings that the user selects in the settings activity. The data saved in the shared preferences are then accessed by all other fragments to apply the settings selected. The data saved in the shared preferences are some boolean values for locking the app in portrait mode, enabling notifications, and enabling a dark mode appearance in the app. The other value saved is strings that will dictate the text size on the app. The Figures below are screenshots of the code used to implement the dependency injection patterns.

Code Example

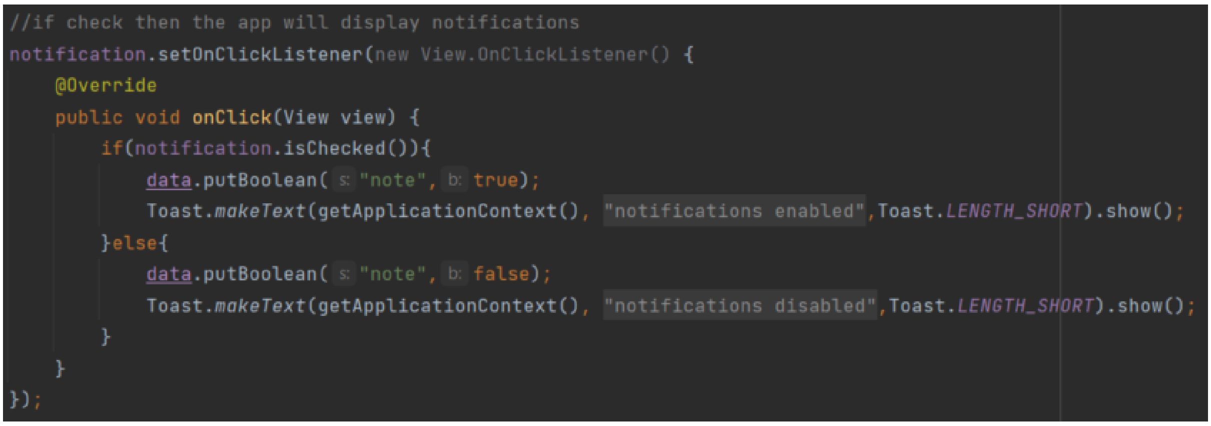
Implementing Shared Preferences



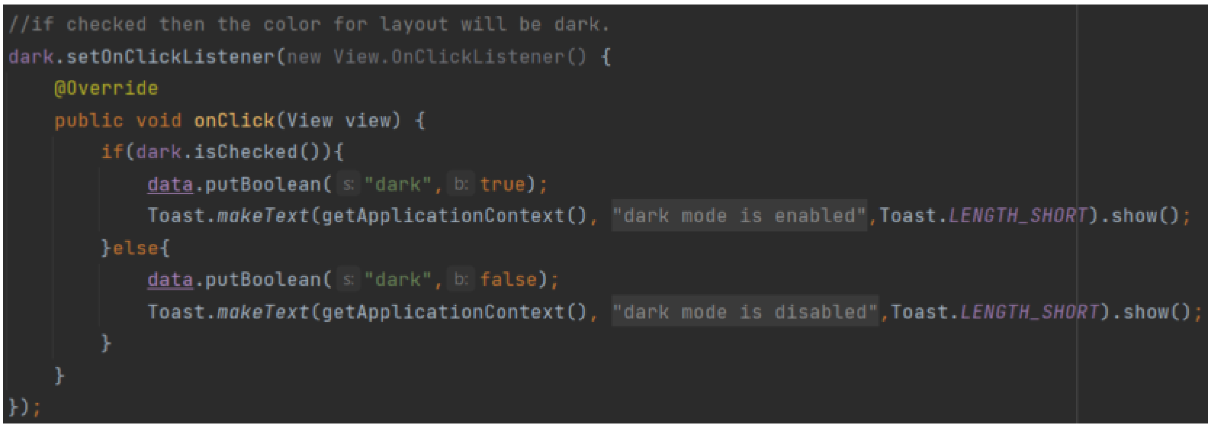
```
textsize.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(RadioGroup radioGroup, int i) {
        int selected = textsize.getCheckedRadioButtonId();
        switch (selected){
            case R.id.small_text:
                //
                data.putString( s: "size", s1: "small");
                Toast.makeText(getApplicationContext(),"Small",Toast.LENGTH_SHORT).show();
                break;
            case R.id.medium_text:
                //
                data.putString( s: "size", s1: "medium");
                Toast.makeText(getApplicationContext(),"Medium",Toast.LENGTH_SHORT).show();
                break;
            case R.id.large_text:
                //
                data.putString( s: "size", s1: "large");
                Toast.makeText(getApplicationContext(),"Large",Toast.LENGTH_SHORT).show();
                break;
        }
```



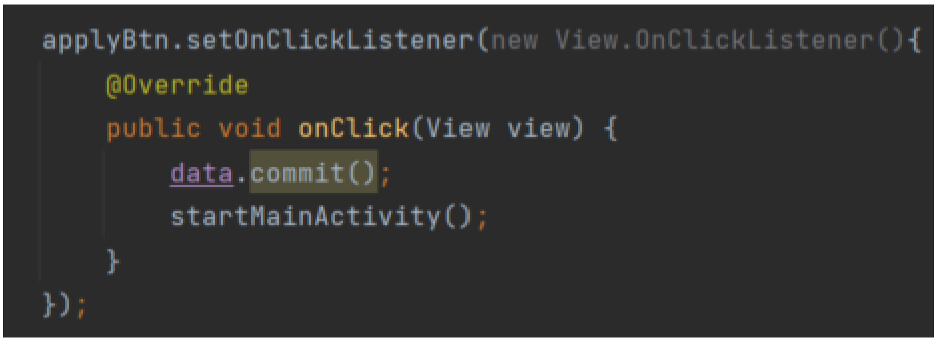
```
//if enable the app will be locked in portrait mode
portrait.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if(portrait.isChecked()){
            data.putBoolean( s: "portrait", b: true);
            Toast.makeText(getApplicationContext(), "portrait mode enabled",Toast.LENGTH_SHORT).show();
        }else {
            data.putBoolean( s: "portrait", b: false);
            Toast.makeText(getApplicationContext(), "portrait mode disabled",Toast.LENGTH_SHORT).show();
        }
        //data.commit();
    }
});
```



```
//if check then the app will display notifications
notification.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if(notification.isChecked()){
            data.putBoolean( s: "note", b: true);
            Toast.makeText(getApplicationContext(), "notifications enabled",Toast.LENGTH_SHORT).show();
        }else{
            data.putBoolean( s: "note", b: false);
            Toast.makeText(getApplicationContext(), "notifications disabled",Toast.LENGTH_SHORT).show();
        }
    }
});
```

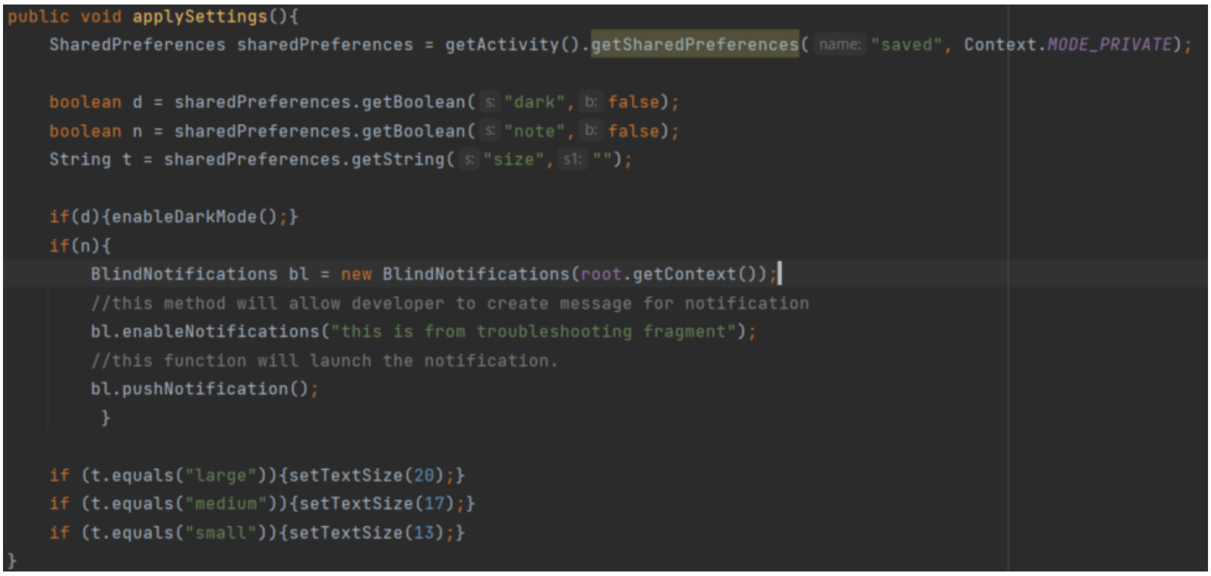


```
//if checked then the color for layout will be dark.
dark.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if(dark.isChecked()){
            data.putBoolean( s: "dark", b: true);
            Toast.makeText(getApplicationContext(), "dark mode is enabled",Toast.LENGTH_SHORT).show();
        }else{
            data.putBoolean( s: "dark", b: false);
            Toast.makeText(getApplicationContext(), "dark mode is disabled",Toast.LENGTH_SHORT).show();
        }
    }
});
```

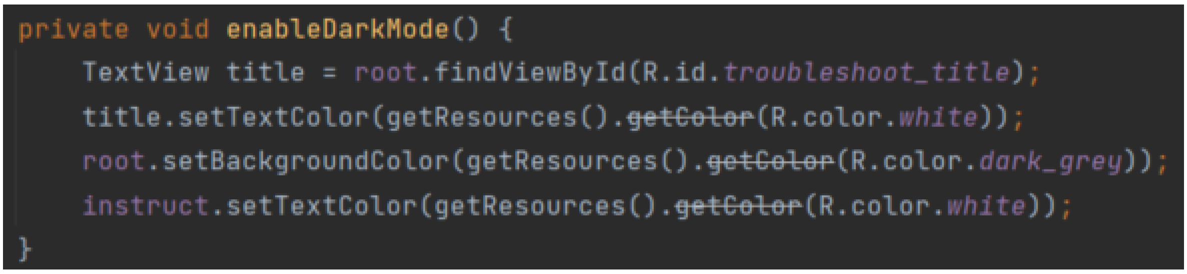


```
applyBtn.setOnClickListener(new View.OnClickListener(){
    @Override
    public void onClick(View view) {
        data.commit();
        startMainActivity();
    }
});
```

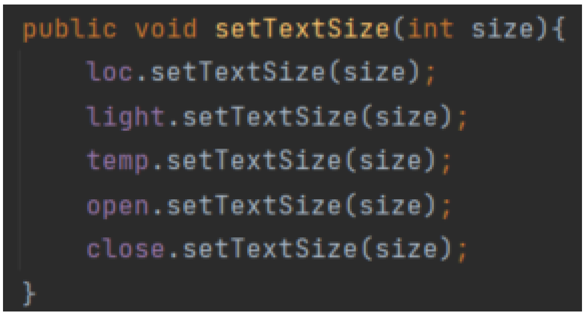
Using Shared Preferences



```
public void applySettings(){
    SharedPreferences sharedPreferences = getActivity().getSharedPreferences( name: "saved", Context.MODE_PRIVATE);

    boolean d = sharedPreferences.getBoolean( s: "dark", b: false);
    boolean n = sharedPreferences.getBoolean( s: "note", b: false);
    String t = sharedPreferences.getString( s: "size", s1: "");

    if(d){enableDarkMode();}
    if(n){
        BlindNotifications bl = new BlindNotifications(root.getContext());
        //this method will allow developer to create message for notification
        bl.enableNotifications("this is from troubleshooting fragment");
        //this function will launch the notification.
        bl.pushNotification();
         }

    if (t.equals("large")){setTextSize(20);}
    if (t.equals("medium")){setTextSize(17);}
    if (t.equals("small")){setTextSize(13);}
}
```



```
private void enableDarkMode() {
    TextView title = root.findViewById(R.id.troubleshoot_title);
    title.setTextColor(getResources().~~getColor~~(R.color.white));
    root.setBackgroundColor(getResources().~~getColor~~(R.color.dark_grey));
    instruct.setTextColor(getResources().~~getColor~~(R.color.white));
}
```



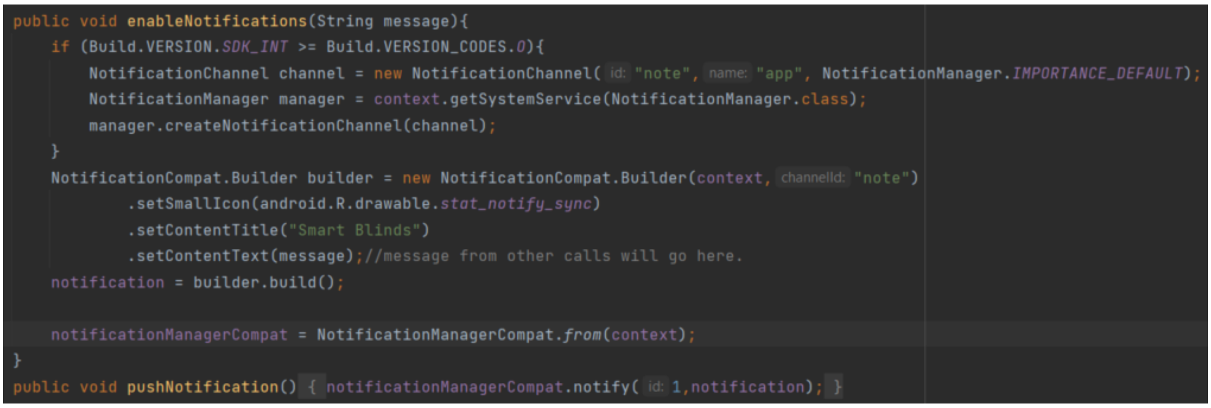
```
public void setTextSize(int size){
    loc.setTextSize(size);
    light.setTextSize(size);
    temp.setTextSize(size);
    open.setTextSize(size);
    close.setTextSize(size);
}
```

Builder

The builder creational pattern is used by a class that is created in order to create notifications and set custom messages that the developer would like to set. The builder is used to set icons that the notification will display. Also, the title of the notification will be set on the builder as well, which in this case is the name of the app. Next is the message that the notification will display, which is an argument that is passed by the developer. Once those elements of the builder are set the data is passed to a notification object variable where it is called in another method to be executed. Below is the code on how to use the builder.

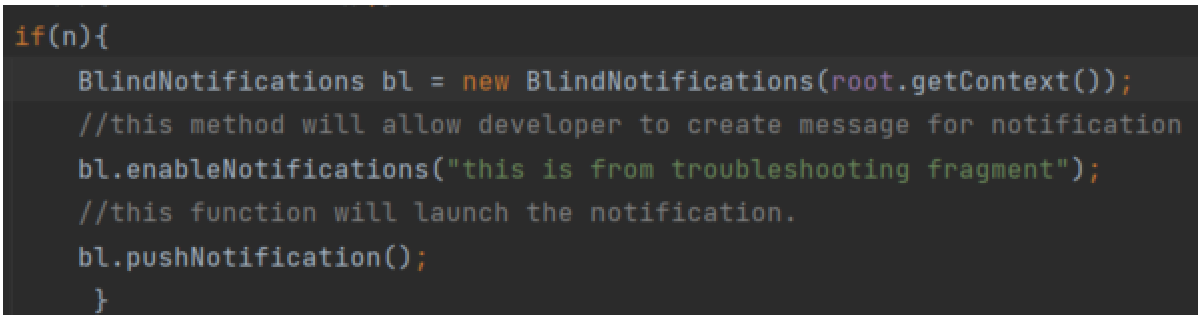
Code Example

Class Implementation



```
public void enableNotifications(String message){
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O){
        NotificationChannel channel = new NotificationChannel( id: "note", name: "app", NotificationManager.IMPORTANCE_DEFAULT);
        NotificationManager manager = context.getSystemService(NotificationManager.class);
        manager.createNotificationChannel(channel);
    }
    NotificationCompat.Builder builder = new NotificationCompat.Builder(context, channelId: "note")
            .setSmallIcon(android.R.drawable.stat_notify_sync)
            .setContentTitle("Smart Blinds")
            .setContentText(message);//message from other calls will go here.
    notification = builder.build();

    notificationManagerCompat = NotificationManagerCompat.from(context);
}
public void pushNotification() { notificationManagerCompat.notify( id: 1,notification); }
```

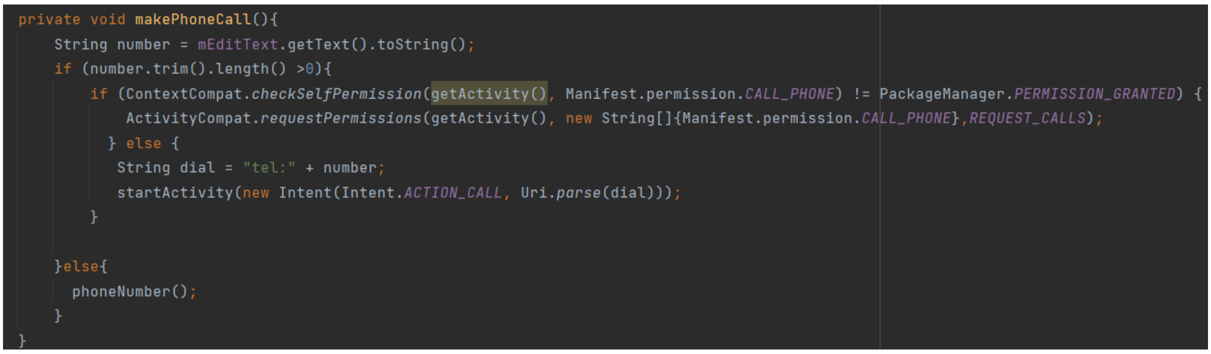
Execution in-app



```
if(n){
    BlindNotifications bl = new BlindNotifications(root.getContext());
    //this method will allow developer to create message for notification
    bl.enableNotifications("this is from troubleshooting fragment");
    //this function will launch the notification.
    bl.pushNotification();
     }
```

Runtime permission

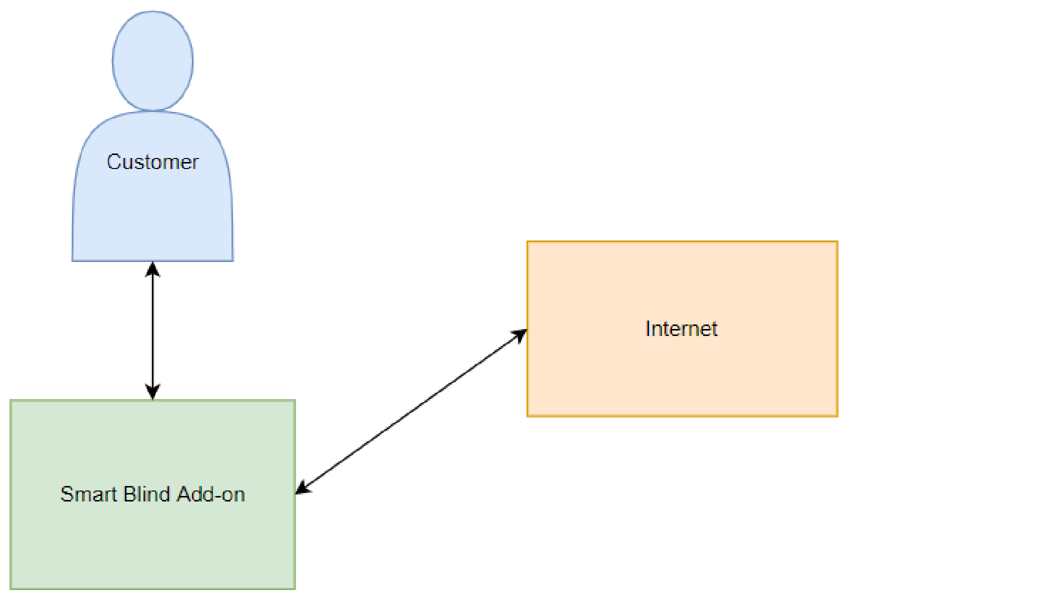
The runtime permission implemented in the app is for the app to make a phone call to a customer service number when the user pressed a button on the Contact Fragment.



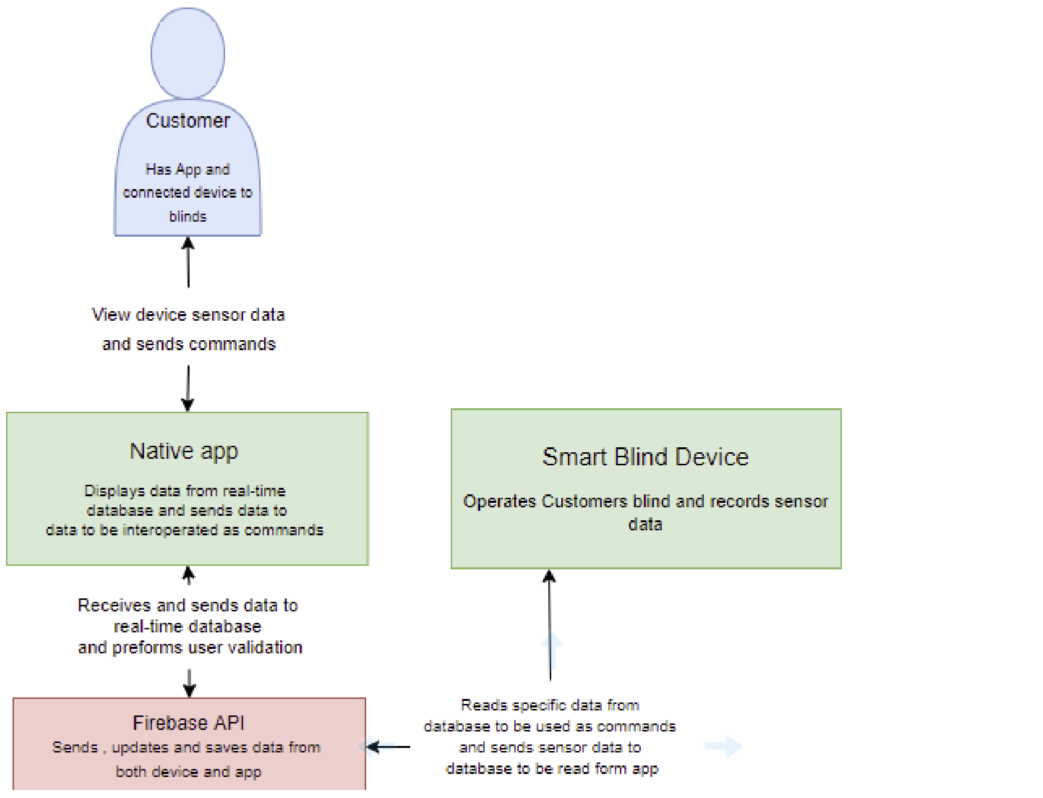
```
private void makePhoneCall(){
    String number = mEditText.getText().toString();
    if (number.trim().length() >0){
        if (ContextCompat.checkSelfPermission(getActivity(), Manifest.permission.CALL_PHONE) != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(getActivity(), new String[]{Manifest.permission.CALL_PHONE},REQUEST_CALLS);
          } else {
           String dial = "tel:" + number;
           startActivity(new Intent(Intent.ACTION_CALL, Uri.parse(dial)));
        }

    }else{
      phoneNumber();
    }
}
```

C4 Model

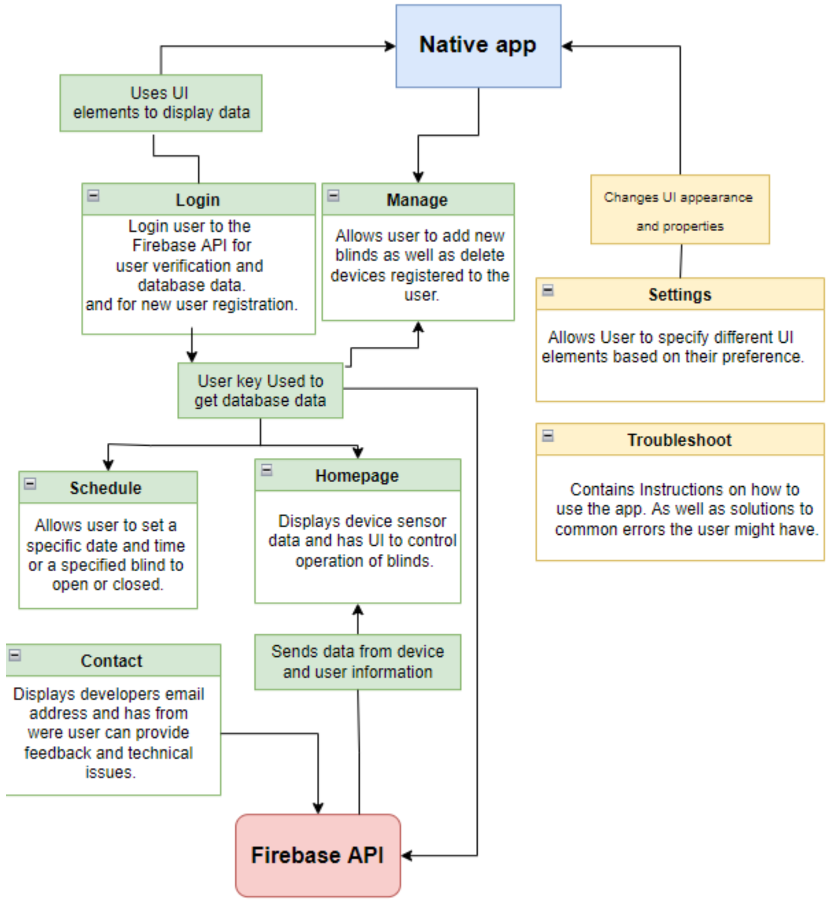
Level 1



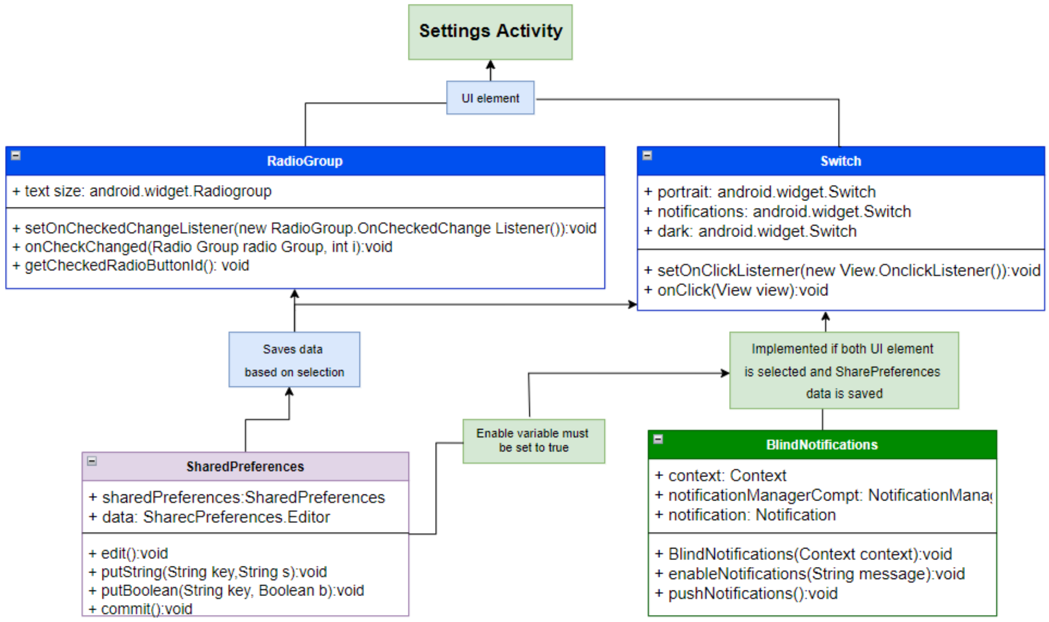
Level 2


Customer
Has App and connected device to blinds
View device sensor data and sends commands
Native app
Displays data from real-time database and sends data to data to be interoperated as commands
Smart Blind Device
Operates Customers blind and records sensor data
Receives and sends data to real-time database and preforms user validation
Firebase API
Sends , updates and saves data from both device and app
Reads specific data from database to be used as commands and sends sensor data to database to be read form app

Level 3


Native app
Uses UI elements to display data
Login
Login user to the Firebase API for user verification and database data. and for new user registration.
Manage
Allows user to add new blinds as well as delete devices registered to the user.
Changes UI appearance and properties
Settings
Allows User to specify different UI elements based on their preference.
User key Used to get database data
Troubleshoot
Contains Instructions on how to use the app. As well as solutions to common errors the user might have.
Schedule
Allows user to set a specific date and time or a specified blind to open or closed.
Homepage
Displays device sensor data and has UI to control operation of blinds.
Sends data from device and user information
Contact
Displays developers email address and has from were user can provide feedback and technical issues.
Firebase API

Level 4


Settings Activity
UI element
RadioGroup
+ text size: android.widget.Radiogroup
+ setOnCheckedChangeListener(new RadioGroup.OnCheckedChange Listener()):void
+ onCheckChanged(Radio Group radio Group, int i):void
+ getCheckedRadioButtonId(): void
Switch
+ portrait: android.widget.Switch
+ notifications: android.widget.Switch
+ dark: android.widget.Switch
+ setOnClickListerner(new View.OnclickListener()):void
+ onClick(View view):void
Saves data based on selection
Implemented if both UI element is selected and SharePreferences data is saved
Enable variable must be set to true
SharedPreferences
+ sharedPreferences:SharedPreferences
+ data: SharecPreferences.Editor
+ edit():void
+ putString(String key,String s):void
+ putBoolean(String key, Boolean b):void
+ commit():void
BlindNotifications
+ context: Context
+ notificationManagerCompt: NotificationMana
+ notification: Notification
+ BlindNotifications(Context context):void
+ enableNotifications(String message):void
+ pushNotifications():void

Coding Progress

Developer	Date	Progress
Vyacheslav Perepelytsya	July 3rd, 2022	Set up better-troubleshooting page layout skeleton for future in portrait and landlocked
Amit Punit	July 3rd, 2022	Implemented Shared preferences functionality on the setting screen. Implemented portrait lock for the app.
Chris Mutuc	July 3rd, 2022	Implement list view widget to view dev's contact emails.
Vyacheslav Perepelytsya	July 4th, 2022	Added/Refactored image to Troubleshooting page
Amit Punit	July 4th, 2022	Implemented Text size setting option on the app. Implemented Test data on Firebase database.
Chris Mutuc	July 4th, 2022	Implemented runtime service to call phone and applied for a default service phone number. Implemented Rating Alert Dialog.
Vyacheslav Perepelytsya	July 5th, 2022	Added Download File button and pseudocode, then, further functional code in TroubleshootFragment
Amit Punit	July 5th, 2022	Implemented Dark mode setting on the app
Chris Mutuc	July 5th, 2022	Implement multiple edit text fields for the feedback forum. Tweaked the Contact screen Layout.
Vyacheslav Perepelytsya	July 6th, 2022	Debugging/Merging for project to work, debugging download file functionality. Converted method to thread for download file functionality. Added necessary AndroidManifest Permissions.

Amit Punit	July 6th, 2022	Implemented notification settings Added back button on the login activity and new user activity. Made layout tweaks to the new user page.
Chris Mutuc	July 6th, 2022	Implement Landscape Orientation and remove hardcoded string text.
Chris Mutuc	July 7th, 2022	Initialised firebase connection. Created a getter and setter for Contact Screen
Vyacheslav Perepelytsya	on July 8th, 2022	Implemented and verified successful download. Pseudocode to show file automatically after downloading in future. Created boolean system to avoid app crashes from calling the same thread twice on troubleshooting fragment. Added Toasts for UI.
Amit Punit	July 8th, 2022	Fix errors in values.xml preventing Gradle build. Implemented back press to main activity on the login screen.
Chris Mutuc	July 8th, 2022	Implemented a functional firebase to store data from users. Did a final check of every feature and fix any problems it has.(Contact Screen)

Vyacheslav Perepelytsya	July 9th, 2022	Create and populate troubleshooting spinner accordingly, code functionality to provide instructions with selection, fix layout view for a spinner, and create constraints for it. Tweaked layout to fit all troubleshooting instructions. Finalize and polish all functionality and UI for sprint 3 for troubleshooting fragments.

Andrew Fraser	July 3rd, 2022	-implement google play service dependencies
Andrew Fraser	July 4th, 2022	-customize login screen and add more functionality such as the google sign in button
Andrew Fraser	July 5th, 2022	-fix some bugs causing the program to crash when starting a new activity
Andrew Fraser	July 6th, 2022	-import google login functionality when the user presses the google sign-in button and return to the main page
Andrew Fraser	July 7th, 2022	-Collect the user’s Google email and name from firebase and display it in the navigation bar header
Andrew Fraser	July 8th, 2022	-Allow the user to sign in and out of the application -clean up code