Deliverable 4

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.

The user needs to create an account in order to use the blinds app and get most of the app's intended functionality. The test account that is used has two blinds preloaded on the account. There are some preloaded blinds in the database as well the blind keys the user needs to enter is: “0001”, “0002”, “0003”, and”0004”. The credential of the test account:
Username: **admin@mail.com**
Password: **Adminaccount1!**

The app has a total of 7 screens, a homepage, a login page, a register page, a troubleshooting page, a manage blinds page and a schedule blinds page. The homepage will display the location of the blind, the temperature and light reading from the device and two buttons to open and close the blind. In order to use most of the features of the app the user needs to log in to the app, this can be done with

either a username and password or they can sign in with a google account. The troubleshooting page contains instructions on how to use the app as well as solutions to common issues the user might run into while using the app. Also on this page, there is a button to download a .txt file of all the instructions so they can be accessed offline.

The manage blinds page is where users can delete and add blinds to their user profile. On the page, there are two buttons to make the appropriate form appear. To delete a blind the user selects the desired blind from a spinner and then presses the button to delete it. To add a blind the user enters the blind location, the blind key that is registered to the device, and the height of the blind, the data is then stored on the firebase real-time database. The schedule blinds page allows the user to set a time for the blind to either close or open at a certain time. On this page there are two fields where the user enters a date and time and a switch to indicate whether the blind is to be opened or closed. When the user enters the data and presses the submit button it will save to the firebase real-time database to be read from the app.

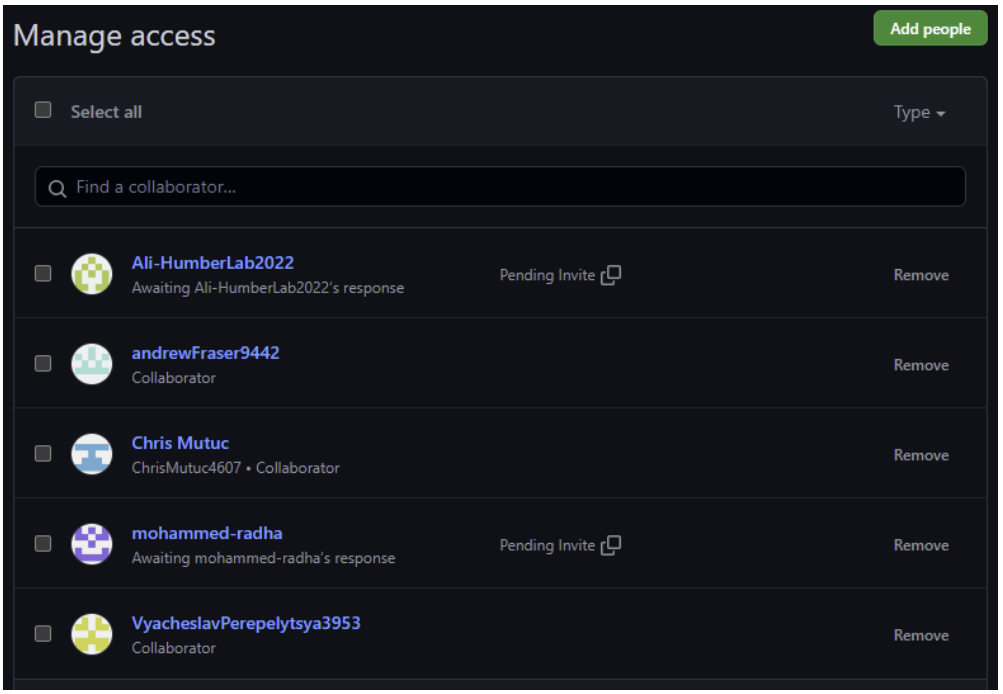
Member info and Participation

Name	**ID**	**Signature**	**Effort**
Amit Punit	n01203930		100%
Andrew Fraser	n01309442		100%
Chris Janelle Mutuc	n01314607		100%

Vyacheslav Perepelytsya	n01133953		100%

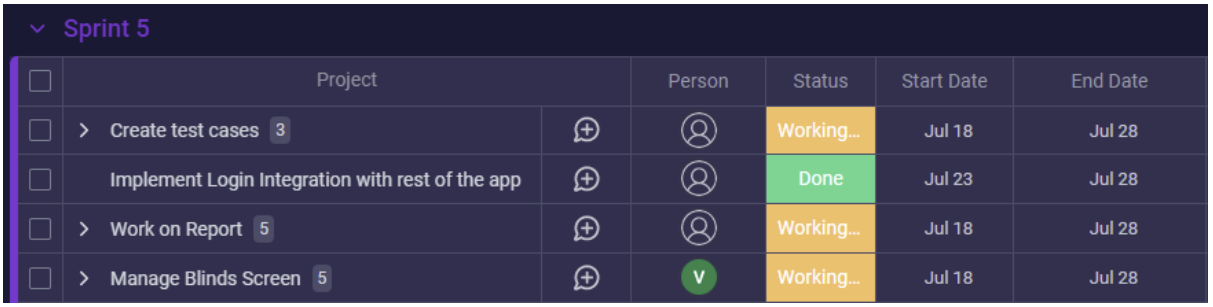
GitHub Project Link

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



Sprint Goals

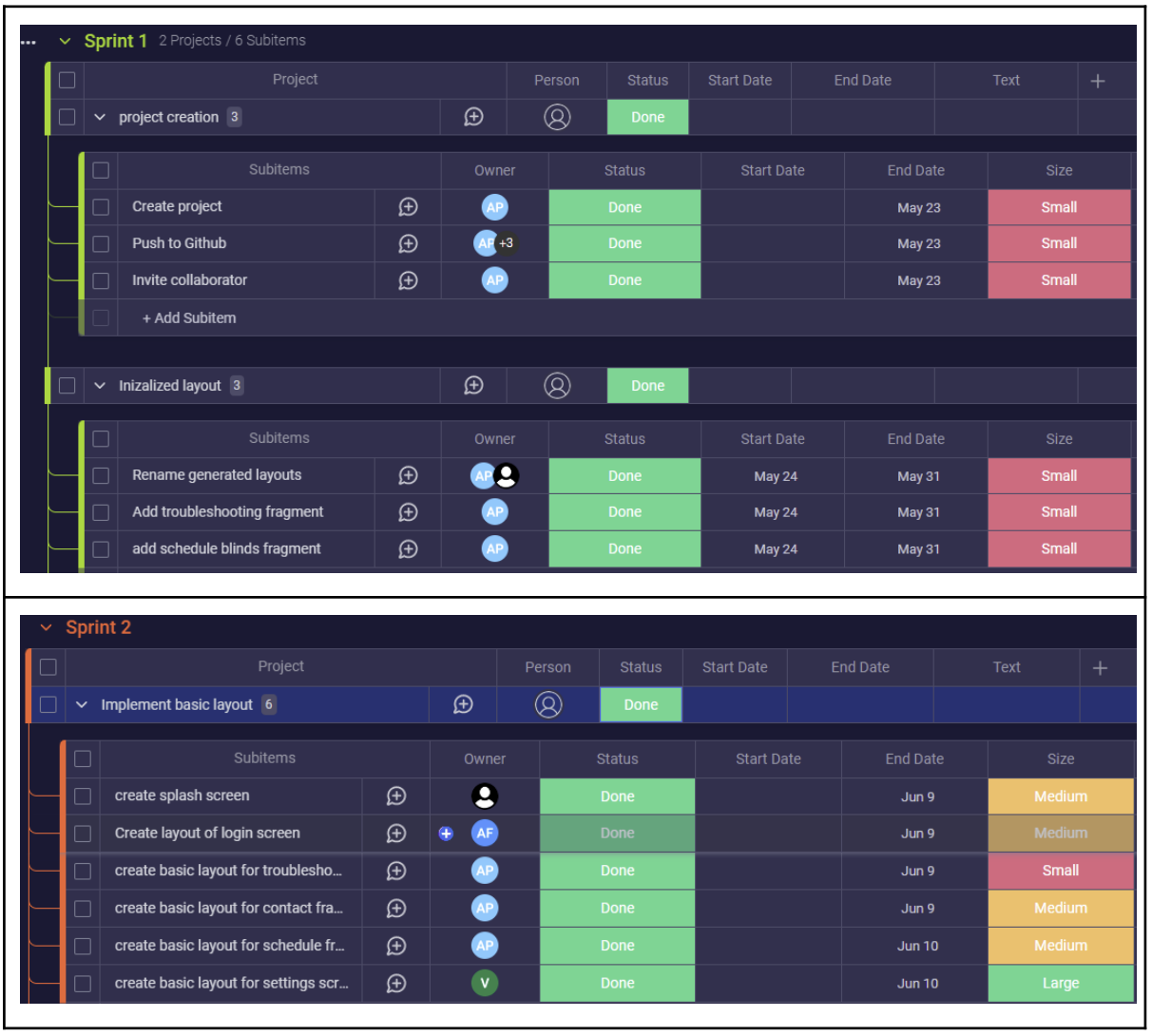
https://view.monday.com/2789650579-eb326b7976f0c1eeeba35b004f81cfa4?r=use1

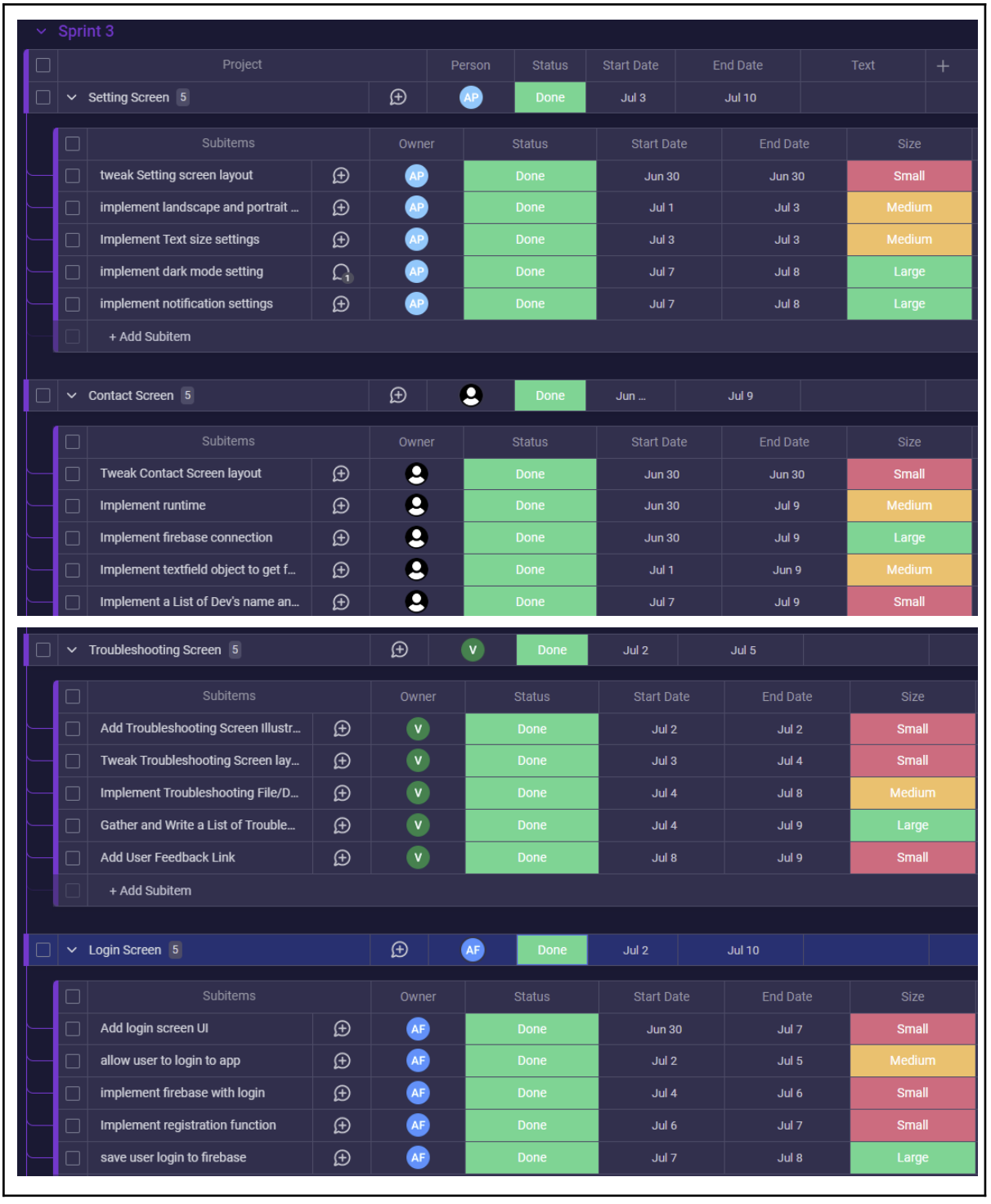


Runtime Permission

Our team has decided that the best possible way to utilize the Runtime-service in our app. Is by using it in our contact screen and we did that by having it function as a call-phone for customer service FAB button. To access the service and validate simply accept it and it should be ready and go when clicked again. Once clicked again it should dial the number we have set in the dialer.

Scrum Dashboard





Sprint 3

Project	Person	Status	Start Date	End Date	Text
Setting Screen 5	AP	Done	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 ...	AP	Done	Jul 1	Jul 3	Medium
Implement Text size settings	AP	Done	Jul 3	Jul 3	Medium
implement dark mode setting	AP	Done	Jul 7	Jul 8	Large
implement notification settings	AP	Done	Jul 7	Jul 8	Large
+ Add Subitem					

Project	Person	Status	Start Date	End Date	Text
Contact Screen 5		Done	Jun ...	Jul 9	

Subitems	Owner	Status	Start Date	End Date	Size
Tweak Contact Screen layout		Done	Jun 30	Jun 30	Small
Implement runtime		Done	Jun 30	Jul 9	Medium
Implement firebase connection		Done	Jun 30	Jul 9	Large
Implement textfield object to get f...		Done	Jul 1	Jun 9	Medium
Implement a List of Dev's name an...		Done	Jul 7	Jul 9	Small

Project	Person	Status	Start Date	End Date	Text
Troubleshooting Screen 5	V	Done	Jul 2	Jul 5	

Subitems	Owner	Status	Start Date	End Date	Size
Add Troubleshooting Screen Illustr...	V	Done	Jul 2	Jul 2	Small
Tweak Troubleshooting Screen lay...	V	Done	Jul 3	Jul 4	Small
Implement Troubleshooting File/D...	V	Done	Jul 4	Jul 8	Medium
Gather and Write a List of Trouble...	V	Done	Jul 4	Jul 9	Large
Add User Feedback Link	V	Done	Jul 8	Jul 9	Small
+ Add Subitem					

Project	Person	Status	Start Date	End Date	Text
Login Screen 5	AF	Done	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	Done	Jul 6	Jul 7	Small
save user login to firebase	AF	Done	Jul 7	Jul 8	Large

Sprint 4

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

Subitems	Owner	Status	Start Date	End Date	Size
configure recycle view layout	AP	Done	Jul 10	Jul 10	Small
populate blind class with methods...	AP	Done	Jul 12	Jul 12	Medium
implement Firebase integration	AP	Done	Jul 14	Jul 14	Large
complete integration with login scr...	AP	Done	Jul 15	Jul 15	Large
Test integration with firebase and ...	AP	Done	Jul 15	Jul 15	Large
+ Add Subitem					

Schedule Blinds Screen 5		Working...	Jul 10	Jul 28	

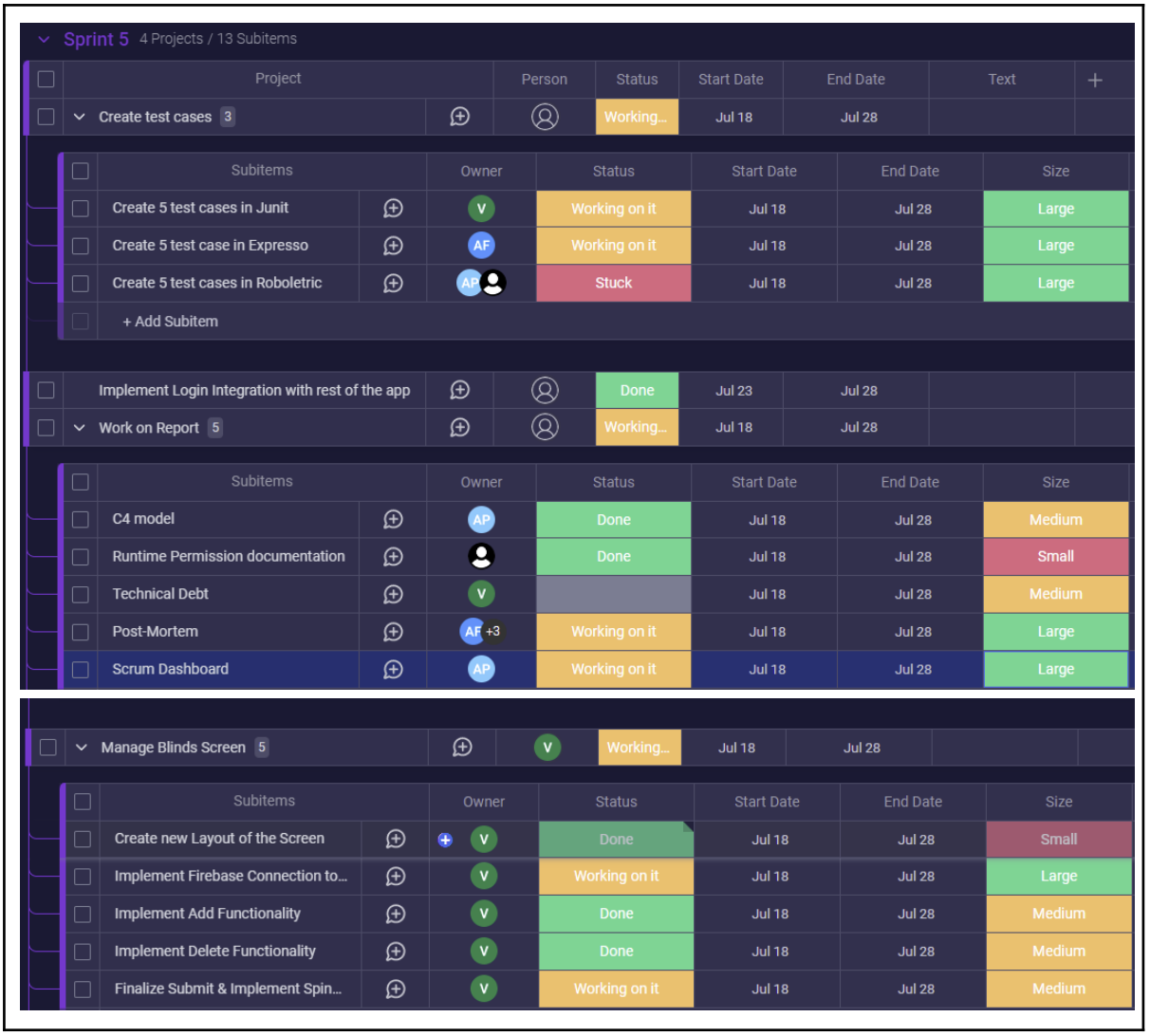
Subitems	Owner	Status	Start Date	End Date	Size
Create & Tweak layout of the screen		Done	Jul 10	Jul 21	Small
Implement firebase connection to ...		Working on it	Jul 10	Jul 28	Large
Implement two text field/buttons f...		Done	Jul 10	Jul 21	Medium
Implement switch button		Done	Jul 10	Jul 21	Small
Implement spinner		Done	Jul 10	Jul 21	Medium

Manage Blinds Screen 5	V	Working...	Jul 10	Jul 17	

Subitems	Owner	Status	Start Date	End Date	Size
Create new Layout of the Screen	V	Working on it	Jul 10	Jul 10	Small
Implement Firebase Connection to...	V	Working on it	Jul 11	Jul 15	Large
Implement Add Functionality	V	Working on it	Jul 15	Jul 16	Medium
Implement Delete Functionality	V	Working on it	Jul 15	Jul 16	Medium
Finalize Submit & Implement Spin...	V	Working on it	Jul 16	Jul 17	Medium
+ Add Subitem					

registration/google authentication 5	AF	Done	Jul 10	Jul 17	

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

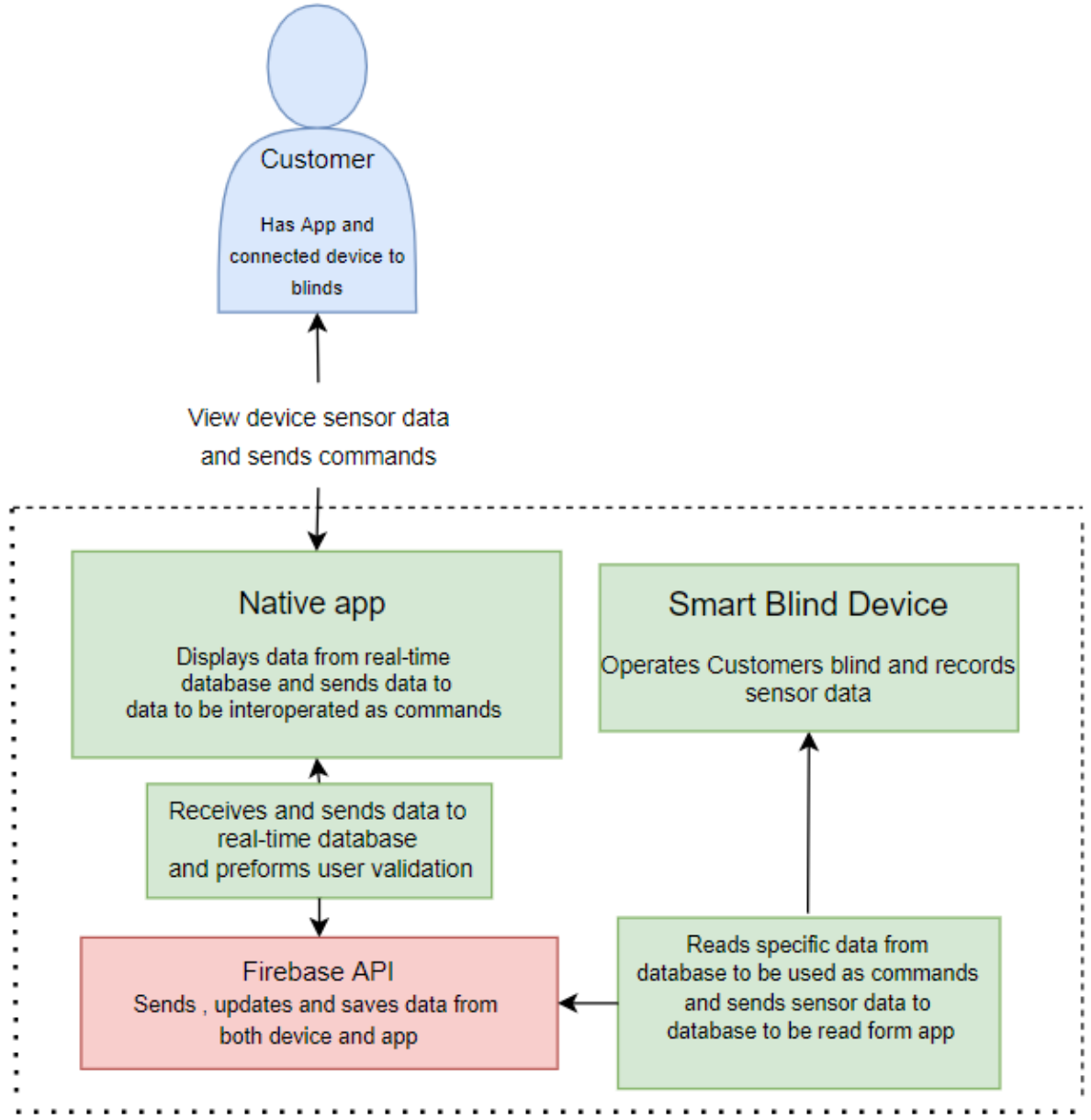


Offline Functionality

The Smart blind add-on app requires an internet connection to accomplish most of the tasks on the app. The offline functionality that the app implements are the changes made at the settings activity and the instruction and tips to use the app on the troubleshooting page.

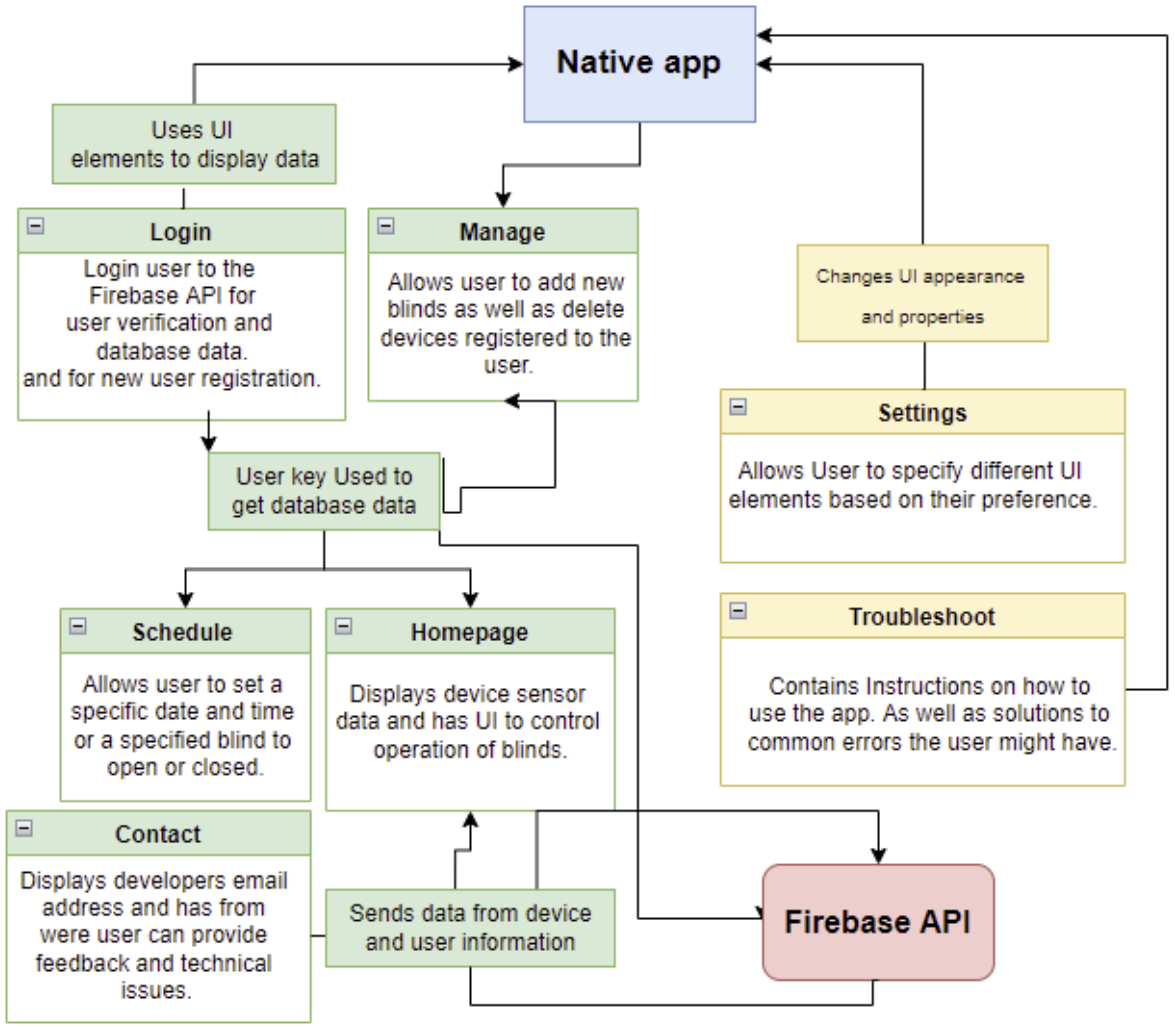
C4 Model

Container Diagram

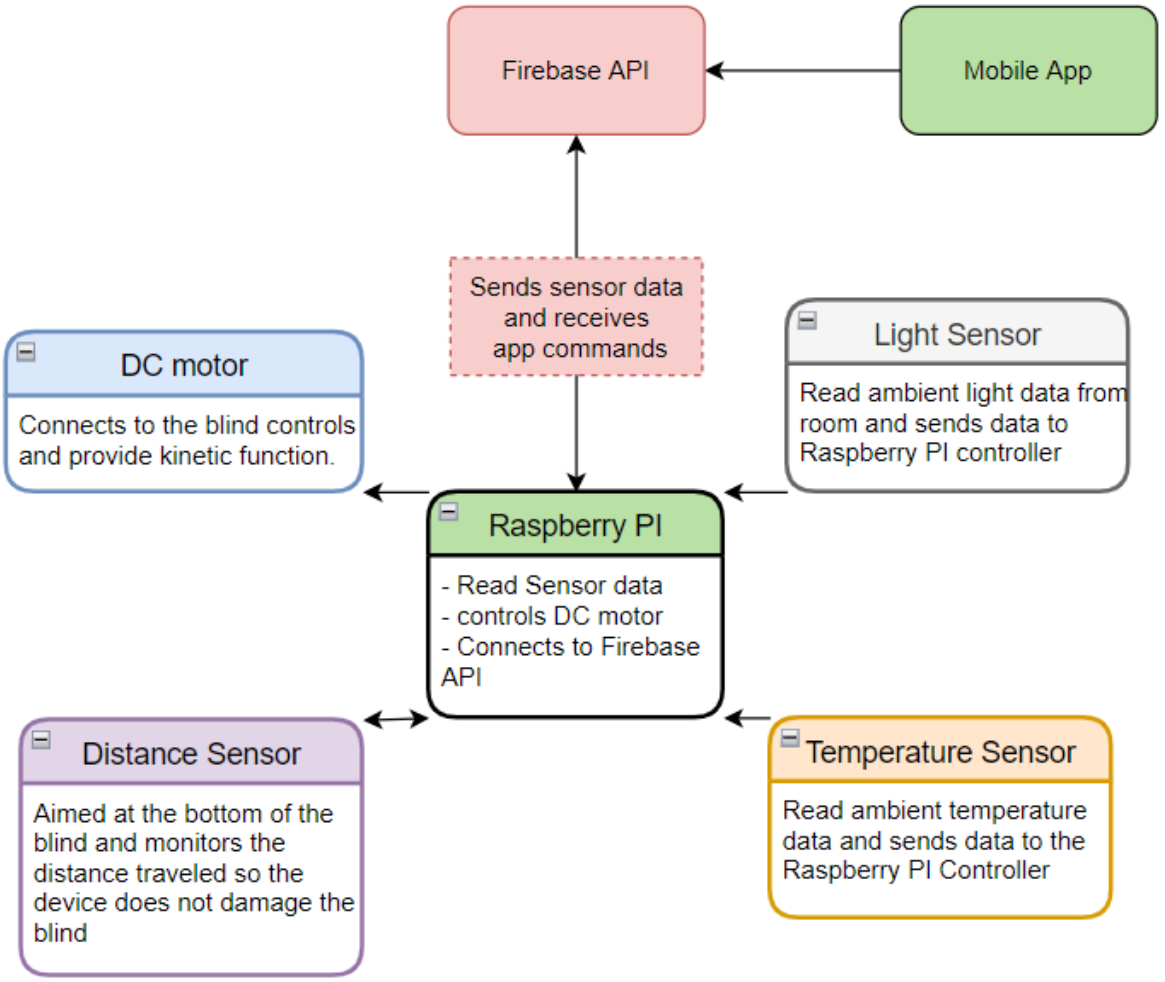


Component Diagrams

Native app



Device



Technical debt

Our most prominent technical debt included a lack of documentation and careful coding around our firebase management on all screens and in general, the login screen registration and troubleshooting download functionality. To understand the technical debt involved we must assess the three issues separately:

- **Firebase management**

  Most of our work on this app was concerned with firebase, and although we haven't been bogged down by it too much due to our care and ingenuity in handling the problems, we had some delays, one prominent technical debt was caused by early implementation of convoluted firebase code without formatting, documenting or commenting it to a sufficient extent. Team members started using different ways to handle the firebase which caused discrepancies among the functions and even technical issues on the app and technical debt as we had to redo a lot of firebase handling code due to this.

Our long-term solution for this was to convene and refactor the code & database to create one solid standard of database handling & coding in the app for all the screens, with sufficient documentation, so that code was more understood by all team members, and so that it became more stable and functional.

-Login Screen Registration

Our login screen registration went through a similar issue as the firebase handling as it was tied to all the other screens due to the credentials being used, and though it worked initially without the database, it needed to be refactored to a code that works with the rest of the screens and the database, hence there was technical debt that we had to address. One unique issue with the login screen was that we didn't notice initially but the registry wasn't working with certain settings, this created the technical debt in the sense that we didn't test the relationship of the settings and registry screen early on in the process - making the issue far bigger in the long run as we accumulated code and it had to be solved with more rigorous debugging, we now know the issue can be avoided by testing any new functionality more thoroughly early on.

- **Troubleshooting Download Functionality**

  The troubleshooting download functionality also was bogged down due to technical debt as the download function was attempted with minimal code, circumventing seemingly more "work" in the code by avoiding the creation of unnecessary threads and permissions and so on. It turns out that this avoidance just wasted more time as early iterations of the download code tended to be buggy and crashed the app and fixations on trying to fix the problem with just one thread just created a messy code - the refactoring of the code and use of two threads and the appropriate permissions and code turned out much more stable and functional.

Post-Mortem

https://lucid.app/lucidspark/d3290b9f-ab6c-4c5c-b89f-963be87908a3/edit?invitationId=inv_f45c4fde-adbd-4a43-a64a-d040d2940dfd#


Project Preformance
Time Management
Quality Issues and Compromises
Lessons Learned and Areas of Improvement
Amit Punit
The project preformance overall meets basic personal standards for a first iteration. getting some of the basic function in working order. To be further improved in terms of future sprints and development time. the apps preformance in terms of cost is great since no montary value was injected in development.
Amit Punit
Time management overall all was reasonably acceptable, meeting target within the appropiate deatlines. However there are some areas of improvment that can be made, specifically with maximizing time given to developing code. For example more time than required was used researching coding issues and implemtations of functions that were later scraped or redone.
Amit Punit
The quality of the App meets an acceptable standard for this scrum. there were some compromises in terms of how user friendly the apps UI is, which is attributed to time and there were some short cuts made when developing code that is relient on the completion of other app functions they would not work or have flaws in their emplementation.
Amit Punit
Some Lessons learned in the development of the Smart Blind Addon app are how changes in the the project scope can be implemented in the planning of each sprint. Also the importance of good team communication for each sprint. Some areas of improvement the need to be made are, having better defintion of done outlines, better management of time, doing more detailed research into stories and tasks, and being more adaptable to changes in project scope
Vyacheslav Perepelytsya
The project performance was of a high standard for me individually and our team as we developed an organizational structure and team members were generally reliable in their roles and time management despite the circumstances, everyone was willing and capable in both creative and performative roles and went beyond their roles to help fellow team members in the spirit of the project.
Vyacheslav Perepelytsya
Time management was acceptable within our circumstances, some difficulties for me personally included other obligations and convoluted instructions and unclear deadlines of the project which were hard to understand at times. Nonetheless the deadlines were met with relative comfort due to exceptional organization and cooperation.
Vyacheslav Perepelytsya
Quality issues and compromises of the app at this point would include instability due to the large scope of the app and general lack of modern UI due to lack of time and investment into it. These shortcomings are easily manageable with further investment and dedication to the app in further updates.
Vyacheslav Perepelytsya
A big app project strives on great organization, cooperation and clarity on goals for it to be succesful. Organization includes the formatting of code for long term use, business end, teamwork end, etc. I would say my personal area of improvement would be to dedicate more time to organize and document the code as it is worth it in the long run and to try to be more vigilant and avoid technical debt in whatever I do.
Chris Janelle Mutuc
The project performance was pretty good as for student devolpers. I can say that i have develop a lot of things as a student coder and had a lot of coding with the people in my group. Overall, the performance is good and everyone is trying their best to achieve best quality of the app.
Chris Janelle Mutuc
I would say time management is great as a team. We manage to have small meetings whenever we need to discuss. Although, For myself I struggle to meet continuous commits since its been hard managing both project and my daily life/work. But i would say as a whole the team did great and manage to finish important deliverables on time.
Chris Janelle Mutuc
Quality issues and compromises of the app, I would say there were a lot of things that we have lack and it would include some shortcuts we took and some features we had to take out due to large scope of the app. Although, I believe we can improve on and accomplish these in the coming future.
Chris Janelle Mutuc
I would say things that i learned from this project would be implementation of firebase along with the app and trying to commit daily and also being with a group of developers that teaches me to be better programmer. Some area of improvement would be commiting daily and learning how to debug certain problems in the app.
Andrew Fraser
The project performance is very well done due to everyone in the group completing their assigned tasks with no final errors. Each tasked was assigned as a team and we all were able to communicate well with one another. If anyone had a problem with the code or needed help with a task, they would send it in the group chat and help would be given by another member.
Andrew Fraser
Time management was a strong suite for our team. We realized early in the project that we would need to work effectively and efficiently in order to complete the project on time. In the end, I do not feel like the group members felt to rushed through out the duration for what we had in mind for this project
Andrew Fraser
I believe the quality for this project is acceptable for the amount of time and the experience level for the team but some things can be improved on. There are a few functions that could have been implemented better that if given more more time, could have been improved upon. We also did not have enough time to add the sensor functionality to the app which would have greatly improved the usability of the app. In the future, our group is planning on implementing some of these features.
Andrew Fraser
Some of the lessons that I have learn throughout the process of developing this app is how important planning is. Planning and discussing ahead of time with the team has greatly helped with the quality and the performance of the project. Another lesson that I have learned is use useful comments and commits. A lot of errors were fixed by reading comment code and pulling the last error free commit. An area of improvement that would have assisted the team while working on the project would be to have more knowledge in developing with android studio. A lot of the functions that our team implemented was the team's first time using and could have been implemented better if our group had more knowledge on it.