DELIVERABLE 5

SiteSync

SiteSync Site Management

Team Number 13

- Anthony Mancia (N01643670)
- Chris Garcia(N01371506)
- Ngoc Le (N01643011)
- Tyler Meira (N01432291)

Name	Student ID	Github ID	Signature	Effort
Tyler Meira	N01432291	95157172		100 %
Chris Garcia	N01371506	196109735		100 %
Ngoc Le	N01643011	157065810		100 %

Anthony Mancia	N01643670	195244864		100 %

10. In the pdf, write the requirement and comment on it for each item. Include the question and the answer.

11. GitHub Repo link. All members must contribute to the repo.

https://github.com/ChristopherGarcia1506/SiteSync

12. Verify the link is working.

13. I will use the following credentials to test your app: Email: aaa@bbb.com Password: Admin101! Also include any other account needed to test, i.e. Admin, customer,…etc.

Admin View: abc@bbb.com Admin101!
Employee View: chris.garcia1000@gmail.com 12345

14. Each member must have a minimum of 20 commits. Counted starting Nov. 17.

15. Marks deducted for members who are not meeting the minimum 20 commits (i.e. if only 16 commits, 20% percent deducted, …etc.), the commits should happen for a duration of minimum of 8 days!

16. Each member must have a minimum of 8 days of commit (Nov. 17 – Nov. 30), otherwise marks deducted.

17. List number of commits for each member for this sprint only. Create a table like below:
CENG-322 2 Team Member Harpreet Eric Arya Nov. 17 4 2 0 …. … … … Total

Team Members	Chris Garcia	Tyler meira	Ngoc Le	Anthony mancia
11/30/2025		2	1	14
11/29/2025	7			
11/28/2025	13	9	16	
11/27/2025		4		
11/26/2025		2		
11/25/2025	1		5	3
11/24/2025		1		2
11/23/2025				
11/22/2025		1		
11/21/2025				
11/20/2025		2		
11/19/2025				
11/18/2025				1
Total:	21	21	22	

18. Sprint 5 goals, list sprint goals and not tasks you worked on.

1. **Improve communication**
2. **Improve user customization**
3. **Consolidate development technicalities**
4. **Complete the Profile Screen with its functionalities**
5. **Give Employees a search feature.**

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

-preform code cleanup
-Optimized Code by consolidating common function to a util file
-Correct Screen Display according to user.

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

Chris Garcia:

I worked on finishing the Profile Screen with its functionalities, such as “edit profile”. Allowing the user to customize fields such as: email, name, phone number, and organization. Furthermore, I implemented a search tool bar feature to allow user employees to search the job they're looking for. Additionally, I moved the functionality of adding a job from a Button to a fab button, then worked on fixing a bug in which an employer would be shown the wrong home screen.

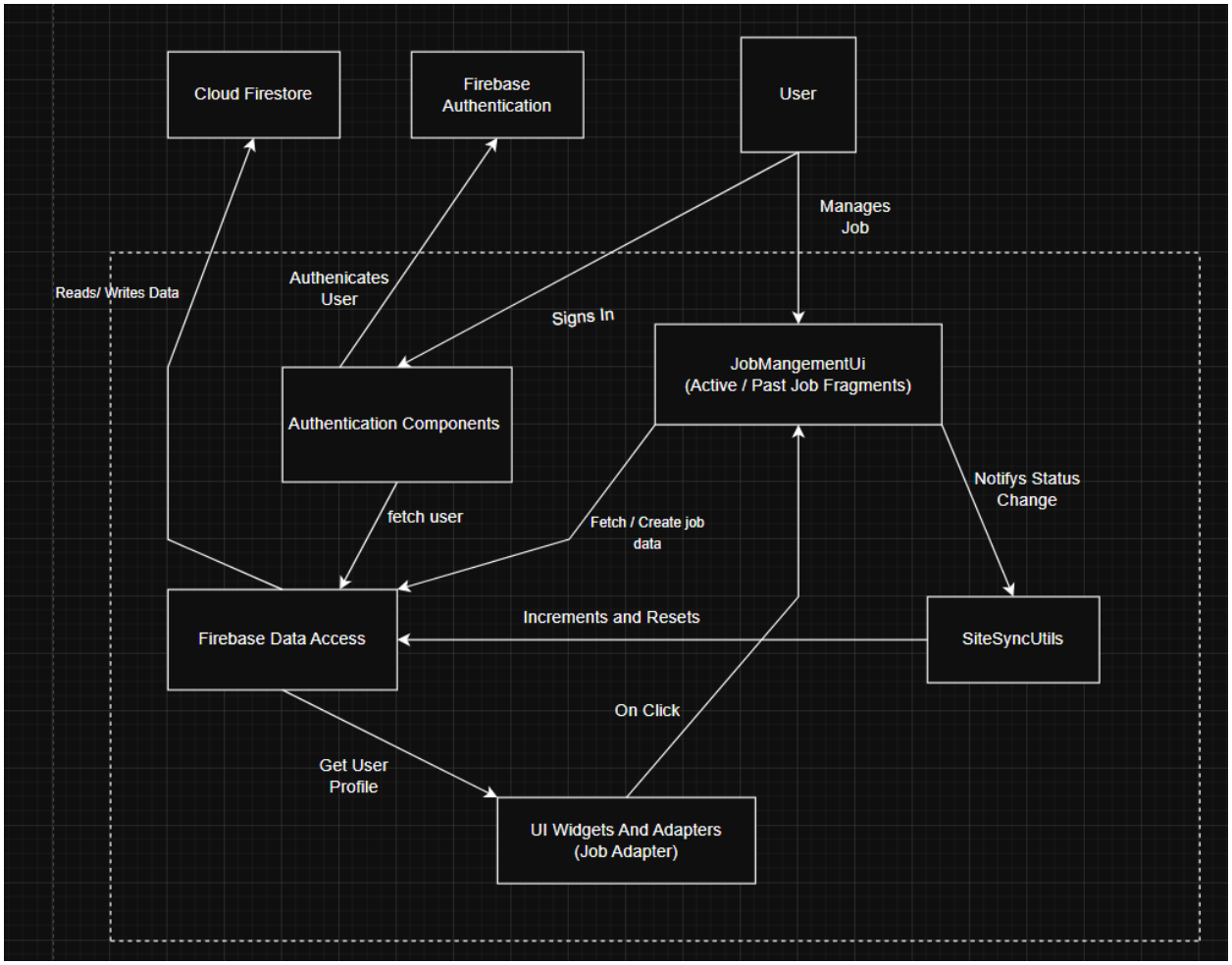
Tyler Meira:

For this sprint I worked on consolidating some code that was reused, across multiple classes, to not only have more concise files, but to eliminate unnecessary bulk of our code. This sprint i also went back through the code to remove any unnecessary leftover code that is not being used, whether this be cleaning up old functionality that we didn't need anymore for the JobListing, PastJob, ActiveJob and JobBoard Fragments, this sprint i also implemented some analytics for the app the current two analytics that i implemented are JobsAccepted and JobsFinished, there are two collections for these in the database, one collection that will reset every 24 hours and one that resets every 7 days, according to the requirements, this sprint i also worked on adding the code for my sensor the the repository.

Ngoc Le: I resolved some technical debts in this deliverable, and made the code look cleaner. I removed repeated validation codes for phone number and email in the register screen and feedback fragment. After that, I stored them in a new class called ValidationUtils. I also fixed some of the code based on previous feedback from the last deliverable about hardcorded strings in the register screen and feedback fragment. I removed duplicated strings in xml files which made the app crash every time it started. I implemented offline functionality for our project so that users can still see their information even if there’s no internet connection.

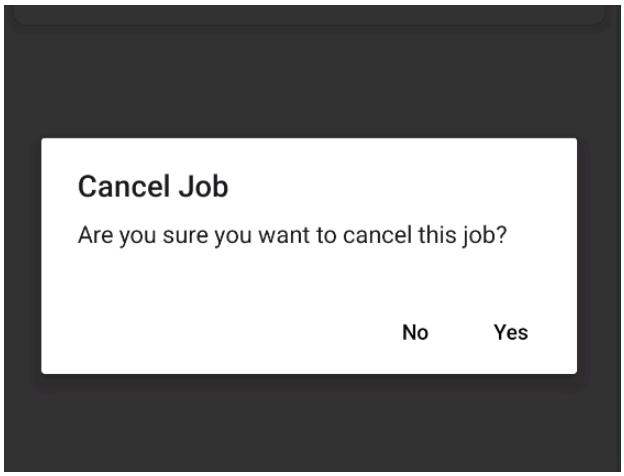
Anthony Mancia: For this Sprint I worked in fixed code from the previous deliverable and added functionality that I was not able to finish in last one like the permission dealing the the toast alert also starting and finishing the help screen in the hamburger menu. Furthermore, worked on the test cases just like last deliverable which were junit, Roblectric, and also Expresso.

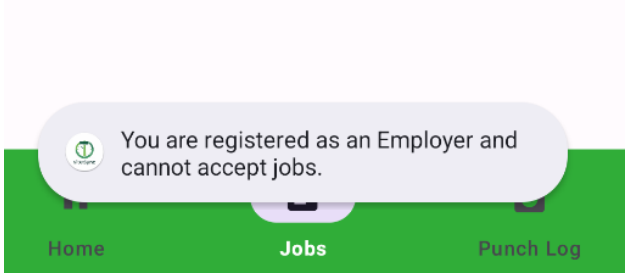
21. Using C4 Model, draw “Component Diagram”. Choose two of the containers in your system. Draw two detailed components diagram.



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

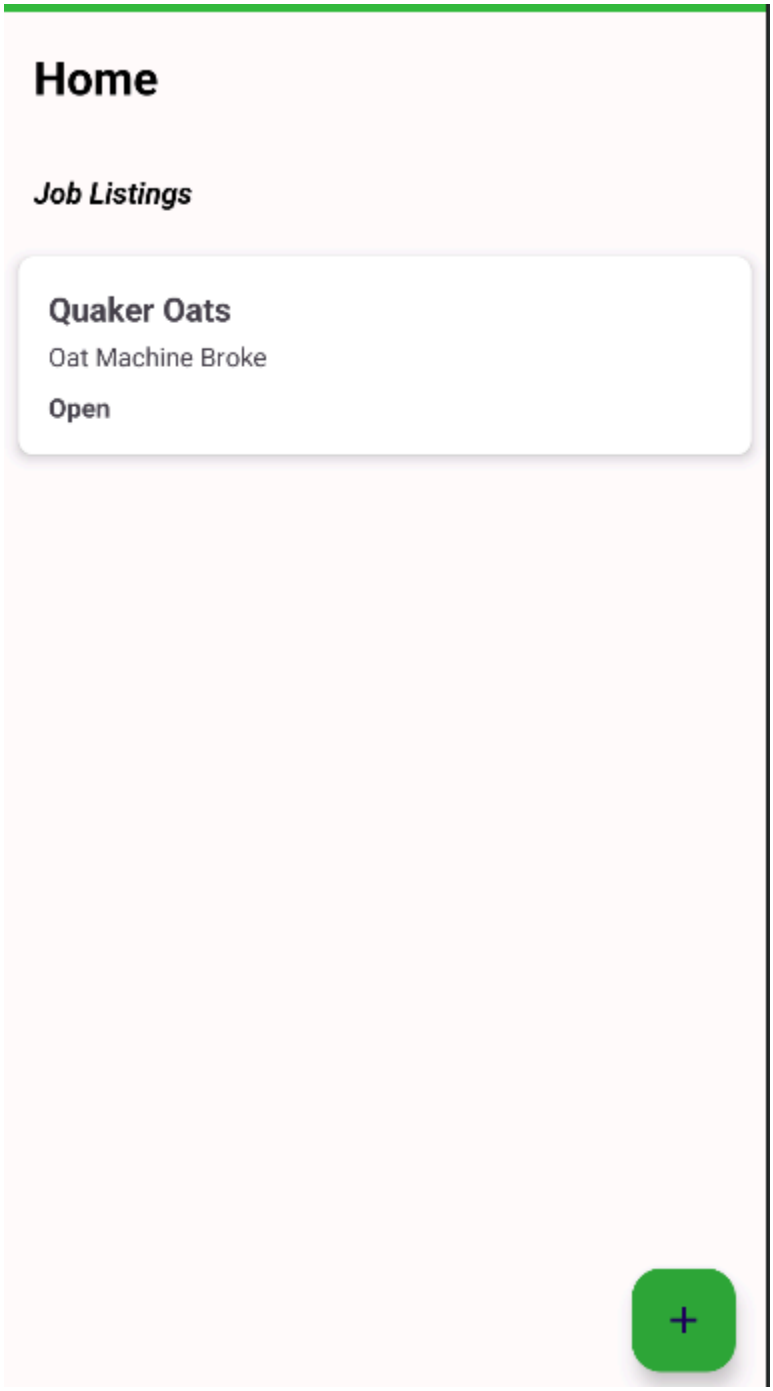
23. Demonstrate the use of all of the following in your app, in the appropriate places: Include screenshots to show. A. AlertDialog B. Snackbar C. Toast D. Notification, handle runtime for API 33 and higher. Seem example below CENG-322 3





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

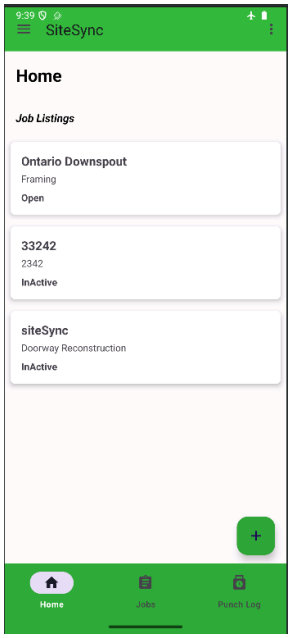
We use the fab button to allow the employer accounts to create new jobs



25. Should implement some functionality for off-line mode. Document what feature will work off-line. Should be meaningful feature. Will be tested, when in Airplane mode. Provide screenshot. Example on how to document offline feature, don't use the same example: CENG-322 4

Before offline functionality

After offline functionality



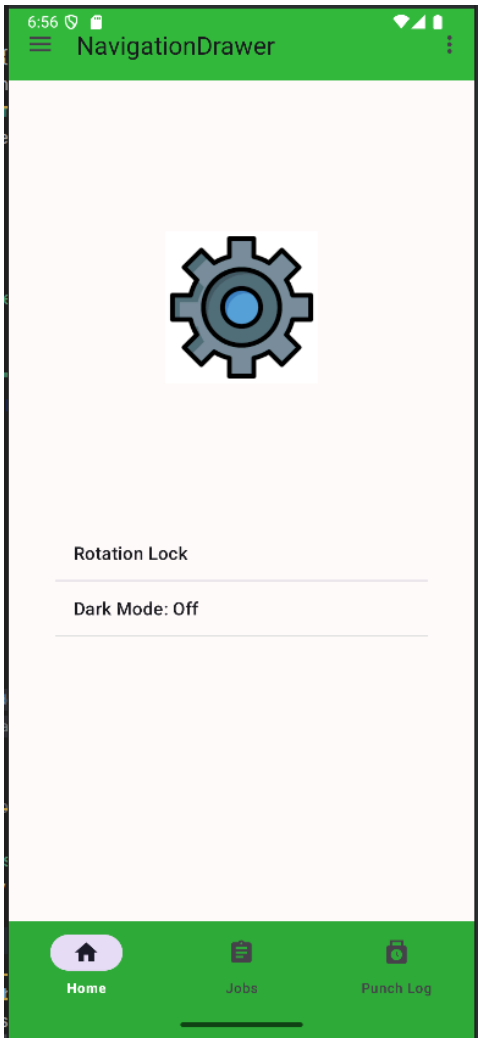
26. Describe how did you handle exceptions, and list the exceptions that you had to handle.

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == RC_SIGN_IN) {
        Task<GoogleSignInAccount> task = GoogleSignIn.getSignedInAccountFromIntent(data);

        try {
            GoogleSignInAccount account = task.getResult(ApiException.class);

            if (account != null && account.getIdToken() != null) {
                firebaseAuthWithGoogle(account.getIdToken());
            } else {
                Alertor.toast(this, "Google Sign-In failed: No account data");
            }

        } catch (ApiException e) {
            Log.w(TAG, "Google sign in failed", e);
            Alertor.toast(this, "Google sign in failed: " + e.getStatusCode());
        }
    }
}
```

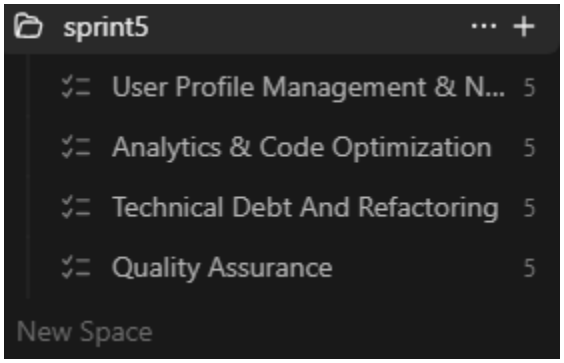
The line GoogleSignInAccount account = task.getResult(ApiException.class); is where the risk lies. The getResult() method attempts to retrieve the signed-in Google account.•If this operation is successful, the code inside the try block continues to execute, proceeding with the firebaseAuthWithGoogle() call.

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

- **Settings Screen to contain Personalized options for the user: Toggle Screen, Dark mode**



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



improve Edit Profile UI Design	CG	Today	Normal
Display Users details in profile screen	CG	Today	High
Develop "Edit Profile Functionality"	CG	Today	High
Develop FAB button functionality	CG	Today	High
Correct Screen display according to user	CG	Today	Urgent

Optimize code by removing unnecessary c...	TM	Today	Normal
Analytics for Jobs Accepted	TM	Today	High
implement data logic to log jobs accepted ...	TM	Today	High
Integrate Sensor in to project	TM	Today	High
perform code cleanup	TM	Today	Normal

General Code Cleanup & Refactor	NL	Today	Normal
Centralize Validation Logic	NL	Today	Urgent
Remove Hardcoded Strings	NL	Today	Low
Fix App Crash	NL	Today	Urgent
Implement Offline Functionality	NL	Today	Urgent

Fix Remaining Previous Deliverable Bugs	A	Today	Normal
Complete Permission Handling Logic	A	Today	Urgent
Implement Toast Alert Functionality	A	Today	Normal
Develop/Finish "Help" Screen	A	Today	Urgent
Update/Expand Test Coverage	A	Today	Urgent

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

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

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

- **Time management was a factor in our development journey in which there needed to be improvement. As a team we managed to pull through each deliverable with improvements in our time management week by week.**

C. Were there issues with the quality or compromises along the way? CENG-322 5

- **Along the way of the SiteSync project there were some quality issues in which we could not fix by the deadline of each sprint. As a team we had to compromise on these issues, and address them in the following sprint. These quality issues comprised of : unfinished functionalities, UI Design, and bugs.**

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

Lessons learned:

- **the key lesson we learned in this project is the importance of communication in a team base scene. We found that in certain instances, our lack of communication in regards to the time of pushing our changes, compromised the work of our colleagues.**

Mistakes:

- **Due to the difficulties of aligning the schedules of the team, we found difficulty in providing the necessary ongoing updates to each team member. This consequently made us to prone to making mistakes such as: committing unfinished code which further delayed our sprint plan**

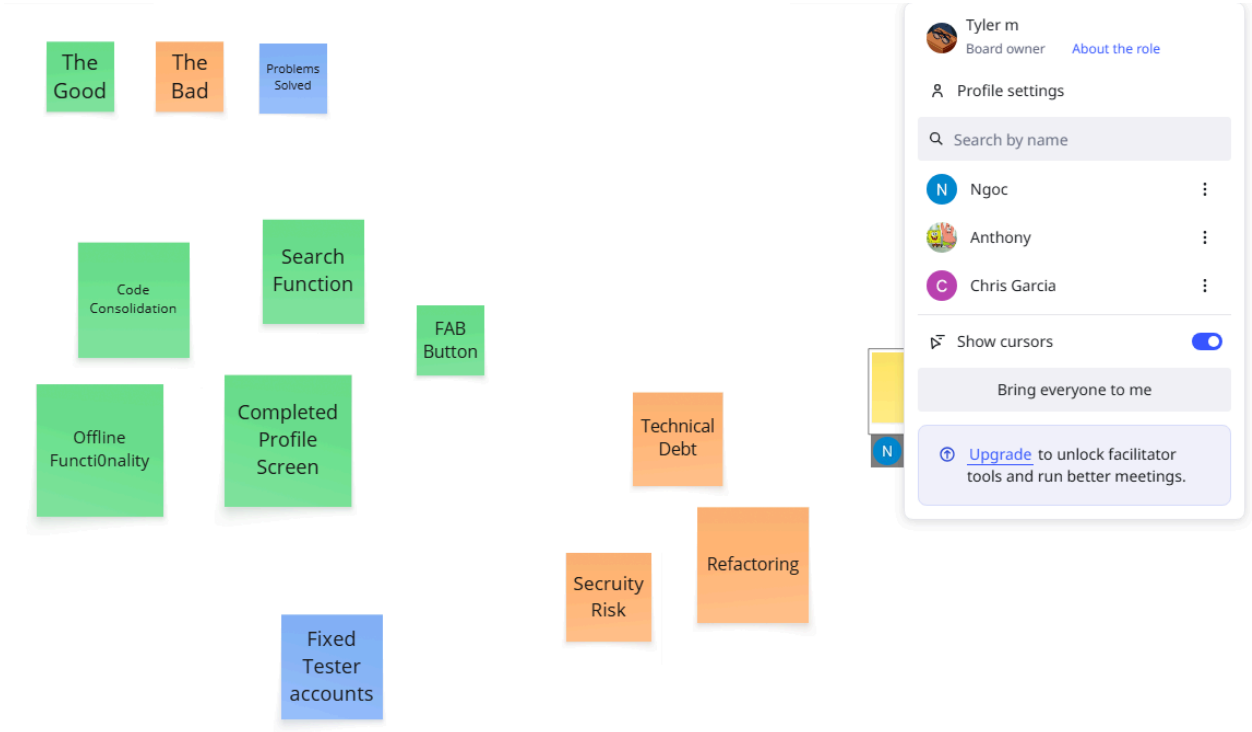
Areas of improvement:

- **Looking back, one of the main areas for improvement resides in our communication protocols. The root cause was our nonaligned schedules, which made spontaneous collaboration difficult.**

E. Who attended and who missed the meeting.

Attendees:
Chris Garcia
Tyler meira
Ngoc le
Anthony mancia

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



31. How did you address technical debt in your project.

We addressed technical debt in our code, over the last sprint by going back to classes developed in other sprints and rewriting / refactoring them to use a utility class if the have similar functions, for example loading a recycler view with jobs in a database is used across multiple files, so in this sprint it was consolidated to FirestoreUtils

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

One area as mentioned above was loading jobs into a recycler view with data from the database, i think we have a total of 4 screens that preform a functionality surrounding this core feature, so that code was

moved into a static utils class so that we can call the util functions from there instead of rewriting the same code over and over again.

33. Describe how DevOps would have helped your project.

Dev ops helped us by helping us break down the tasks/functionality that we created for ourselves or provided for us by the deliverable document, each person on the team would choose the functionality the workout then report when its done so that everyone can pull the changes and test them.

34. What coding standards did you use, and how did you use them.

A coding convection that we used is our naming convention, we something called Camel casing where the first letter of each word in a variable or class is capitalized. We also tried to follow best priacticves and more so this sprint than others so the rewriting of code, and consolidating it into a utility class.

35. What AI tool used and how it helped in your work.

We used gemini to help aid us in our work, it is very useful for very repetitive tasks, such as translating all the the strings in the string.xml file to french, we used ai to also ask for best practices to implement certain functionality, then used a combination of what the ai provided us and our own knowledge through mobile programming to come up with a good solution.

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

A security vulnerability in our app is through our rules in our database, right now this is what are rules are

set to rules_version = '2';

```
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write
    }
  }
}
```

This small piece of code means the our database, is pretty much open to anyone who can get our api key, they would be able to pull data with ease, and with no identification checking. To make sure the person who is pulling the data is not an outside threat actor.

37. Create a table showing some analytics about using your app: by all users (Not a single user) Last 24 hrs Last 7 days Access Home Screen 50 times 200 times Access Settings Screen 20 80 times

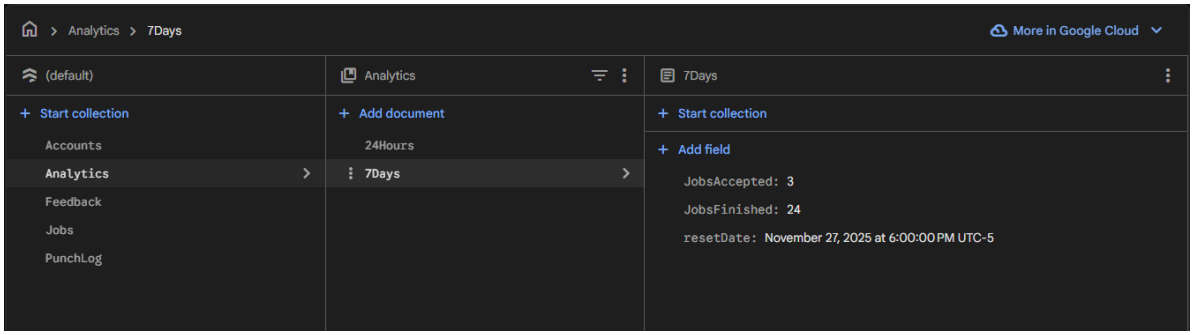
This is some analytics from our app, both collection have the same datafeilds wih the only difference between their functionality is when the counts reset.

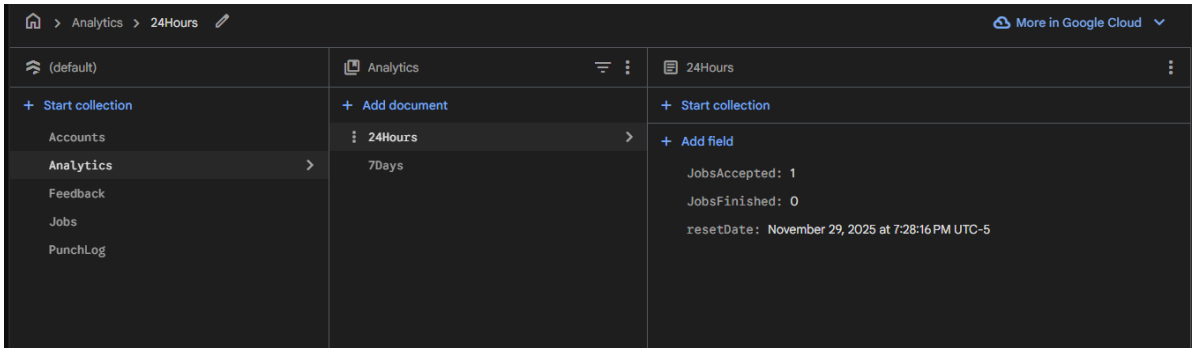
	24Hours	7Days
Jobs Accepted	1	3
Jobs Finished	0	24
resetDate	November 27, 2025 at 6:00:00 PM UTC-5 (timestamp)	November 29, 2025 at 6:00:00 PM UTC-5 (timestamp)

38. Describe your solution to pull the statistic above

In our JobAdapter for our recycler view i had to create a onlcick method that allows the user to interact with the recycler view, because android studio does not provide us with a default solution for this, so when the user onclicks the yes button for the both conditions, they automaticall increment all this code was stored in a separate utility class called SiteSyncUtils and is called from inside the onclick method.

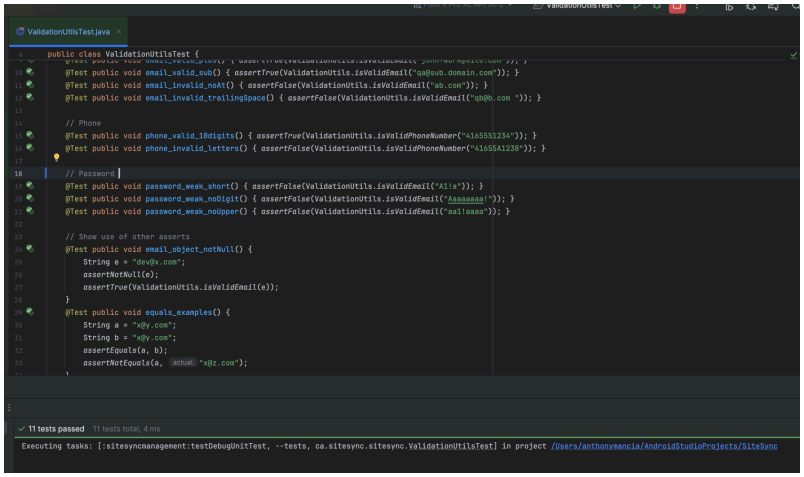
39. Provide Screenshots showing the solution you used.

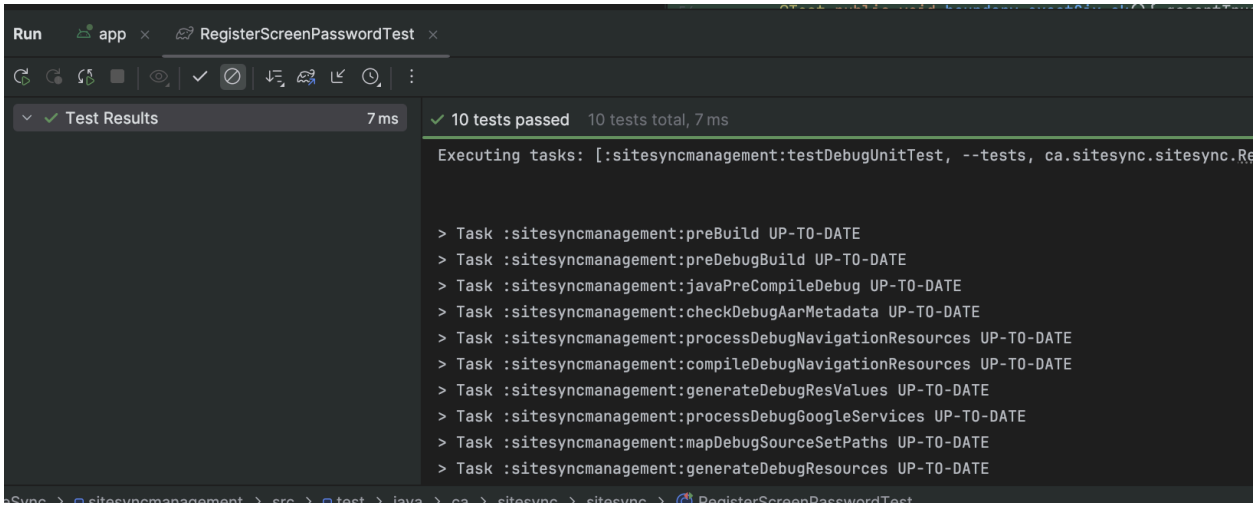


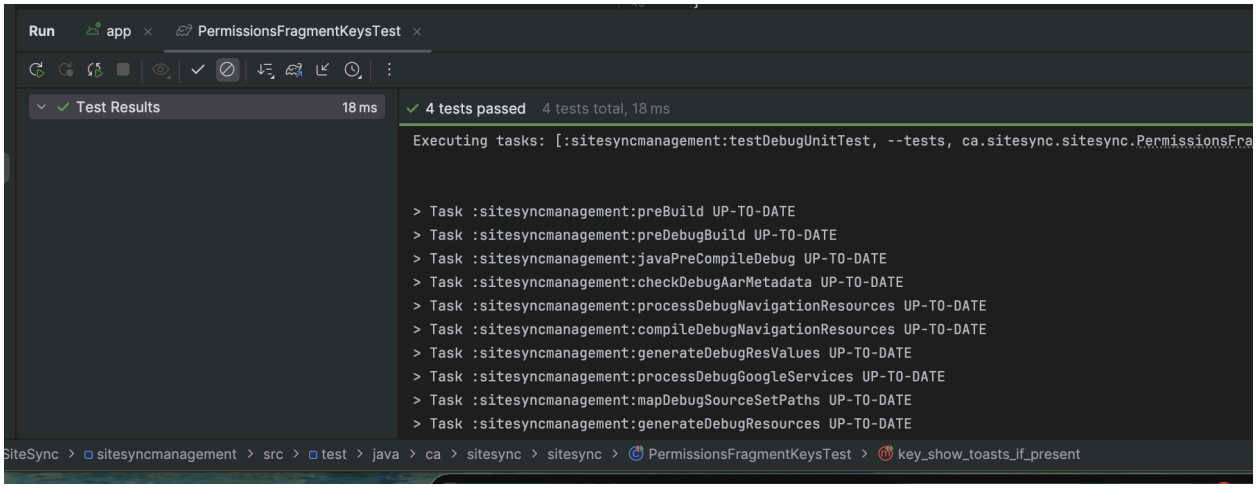


40. Describe your strategy for writing the test cases.

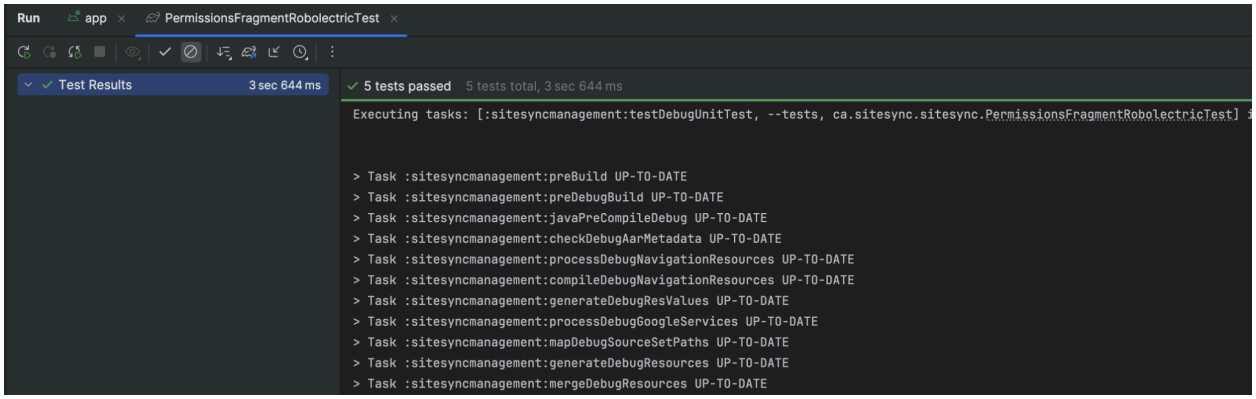
41. Screenshots showing 3 classes junit test cases are passing.



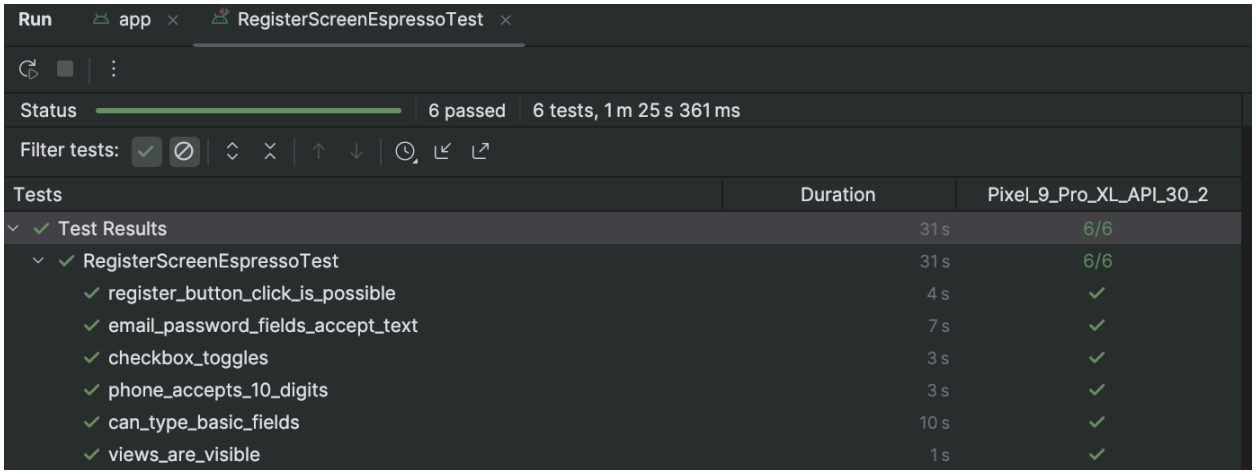




42. Screenshot showing Robleteric test cases are passing



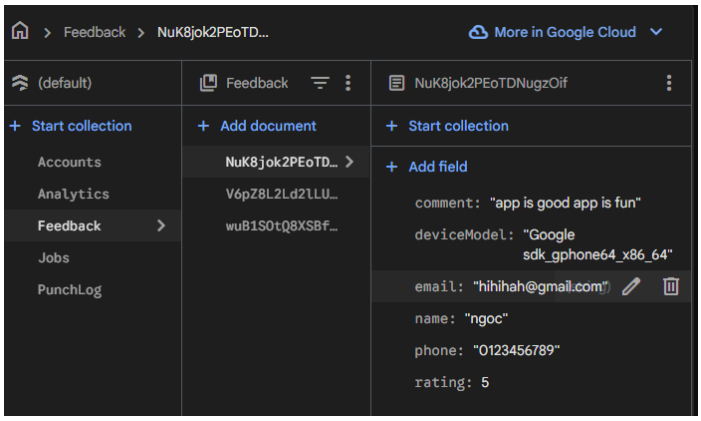
43. Screenshot showing Espresso test cases are passing.

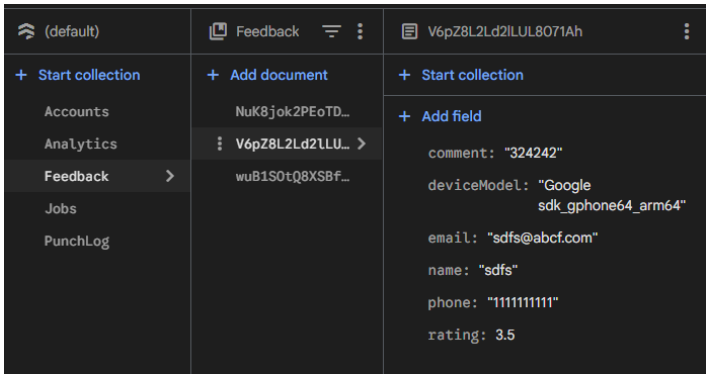


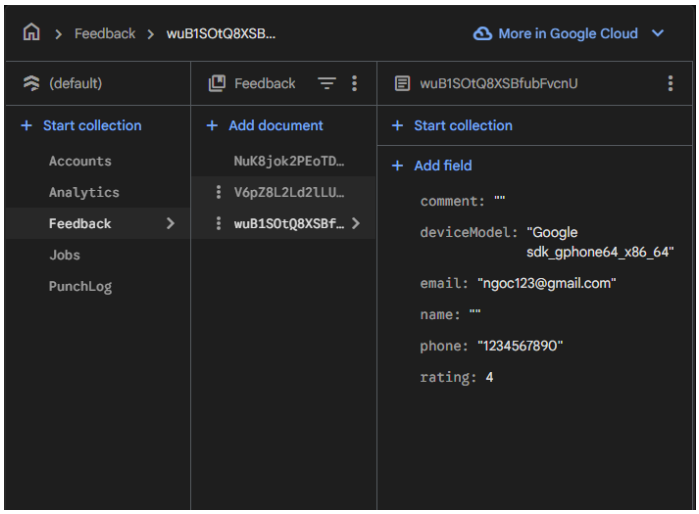
44. Screenshots showing the data in your DB, sensors data (at least 3 different sensors), and login data (including Google login). CENG-322 6

This is going to be the collection for where our sensors put code, but before we can use any of the readings we must develop the software for our devices for any of it to be useful.

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







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

Personally i think we likes that the deliverables gave us direction and what certain things should be completed at certain times in the development process, but that also made it feel if the alot of our creative freedom was slashed because we couldn't add certain things we wanted cause we had to sacrifice some of time to implement certain requirements for the deliverable.

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

48. Commit and push the ppt into deliverable5 folder, this is the only file that can be committed and pushed pass the due and before your presentation.

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

50. Practice on presenting, and be ready, each team member will have 3 - 4 minutes to present.

51. If document not in pdf, or not in the proper name (marks will be deducted!) Please save the PDF document, i.e. TeamName_ProjectName_Deliverable5.pdf 52. One member to submit the pdf file.