

DELIVERABLE 3

SiteSync

SiteSync Site Management

Team Number 13

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

Table Of Contents

- 6) The siteSync Management project is a project to provide construction workers a platform in which they can find, manage, and track their jobs around the GTA. It aims to eliminate the hassle of looking for jobs aimlessly, and simplifies tracking hours worked for employers to employee transparency.

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

<https://github.com/ChristopherGarcia1506/SiteSync>

11. Verify the link is working.

The screenshot shows the GitHub repository page for 'SiteSync' owned by 'ChristopherGarcia1506'. The repository has 5 branches and 0 tags. The master branch is selected, showing a list of recent commits. One commit by 'TylerMeira2291' is highlighted, showing a merge pull request from 'ChristopherGarcia1506/Tylers-Branch'. The commit message is 'Merge pull request #13 from ChristopherGarcia1506/Tylers-Branch'. The commit was made by 'd5base4' last month with 66 commits. The commit details show changes to 'idea', 'Docs', 'app', 'gradle', '.gitignore', 'README.md', 'build.gradle.kts', 'gradle.properties', 'gradlew', 'gradlew.bat', and 'settings.gradle.kts'. The commit message also mentions 'Added Firestore database connection, with somesample fun...'. On the right side of the page, there is an 'About' section describing SiteSync as a time management system for construction sites, warehouses, or residential job sites. It highlights portability and helping connect employers and employees. There are sections for 'Readme', 'Activity', 'Contributors' (4), 'Releases' (none), and 'Packages' (none). A 'Create a new release' button is available under releases.

12. Login functionality, I will use the following credentials to test your app: Email: aaa@bbb.com Password: Admin101!

Created(this is for employer POV)

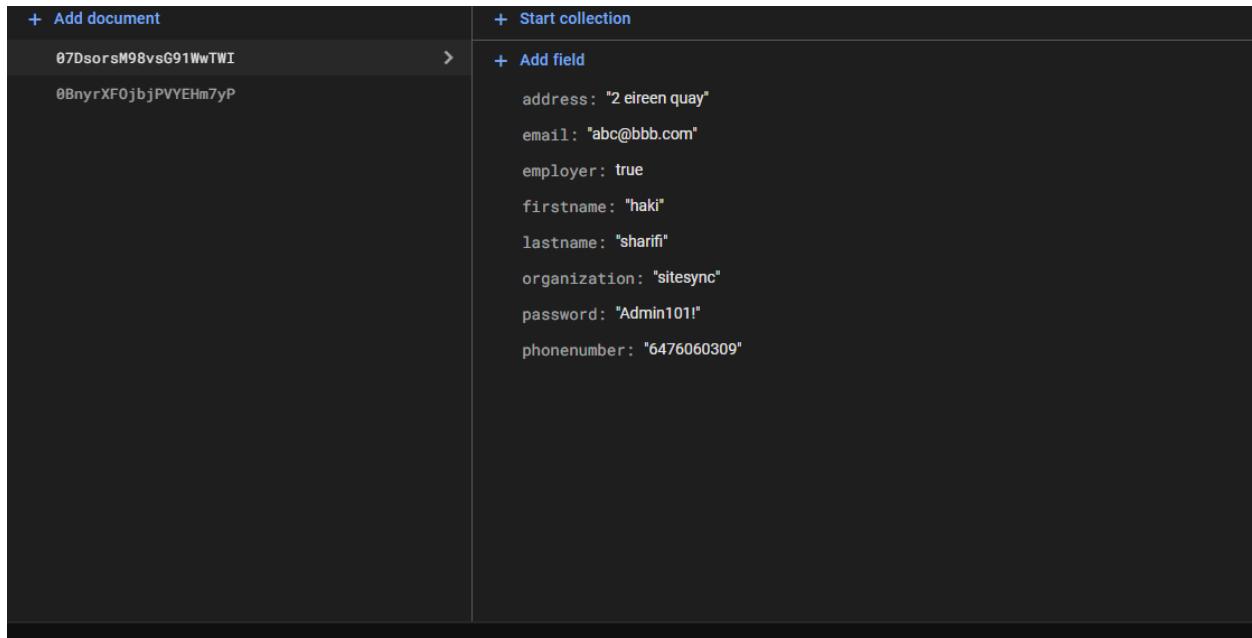
13. Document any other login credentials I should use i.e. Admin vs. regular user. I should not need to send you an email to login to your app, or create an account.

Regular user credentials

email:chris.garcia1000@gmail.com

Password: 12345

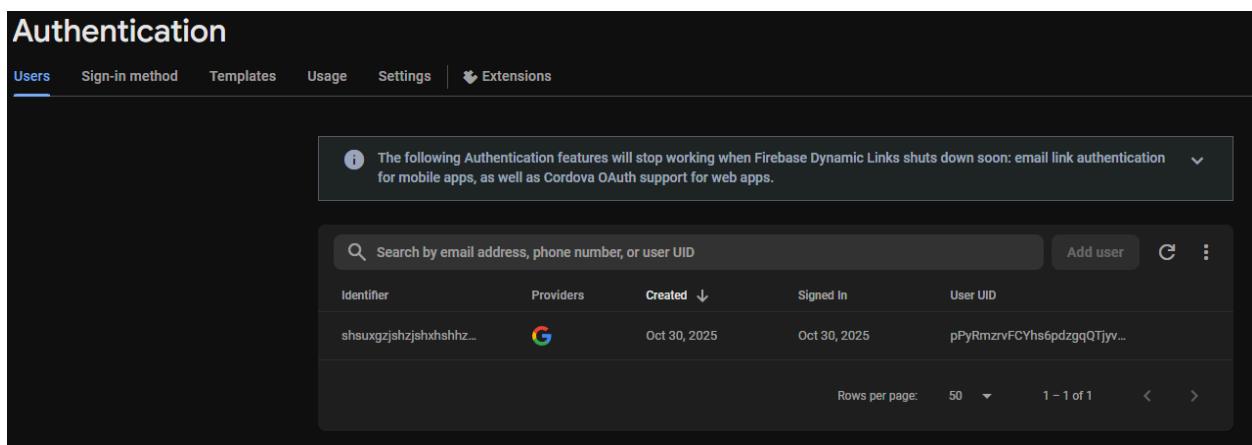
14. Verify you had created an account in the DB with credentials above. Take a screenshot showing the account above in the DB.



A screenshot of the Firebase Firestore interface. On the left, there's a sidebar with a '+ Add document' button and a list of document IDs: '07DsorsM98vsG91WwTWI' and '0BnyrXF0jbjPVYEhm7yP'. The main area shows a document with the following fields:

- address: "2 eireen quay"
- email: "abc@bbb.com"
- employer: true
- firstname: "haki"
- lastname: "sharifi"
- organization: "sitesync"
- password: "Admin101!"
- phonenumer: "6476060309"

15. Screenshot showing the Authentication and users logged in using gmail or their email accounts, i.e. below: some used their credentials, some used gmail account to login into the app.



A screenshot of the Firebase Authentication console under the 'Users' tab. The interface includes a navigation bar with 'Users', 'Sign-in method', 'Templates', 'Usage', 'Settings', and 'Extensions'. A message box states: 'The following Authentication features will stop working when Firebase Dynamic Links shuts down soon: email link authentication for mobile apps, as well as Cordova OAuth support for web apps.' Below this is a search bar and an 'Add user' button. The user list table has columns: Identifier, Providers, Created, Signed In, and User UID. One visible row shows:

Identifier	Providers	Created	Signed In	User UID
shsuxgzjshzjshhhz...	G	Oct 30, 2025	Oct 30, 2025	pPyRmzrvFCYhs6pdzgqQTjyv...

At the bottom, there are pagination controls for 'Rows per page: 50', '1 - 1 of 1', and navigation arrows.

16. You are working on sprint 3.

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

Chris Garcia:

Implemented the following functionalities: google sign in, remember me button when user logs in, employer Home Screen and post listing UI, added sign out button.

Updated Home Screen icon, login and Register UI, app name,

Anthony mancia:

Added the overflow menu which includes permissions(moved from settings) to make it so it can be changed within that screen ,feedback(moved form hamburger viewer).

Updated the colours mainly used from orange to green so it is able to fit with our logo and have a more uniform look.

Tyler Meira :

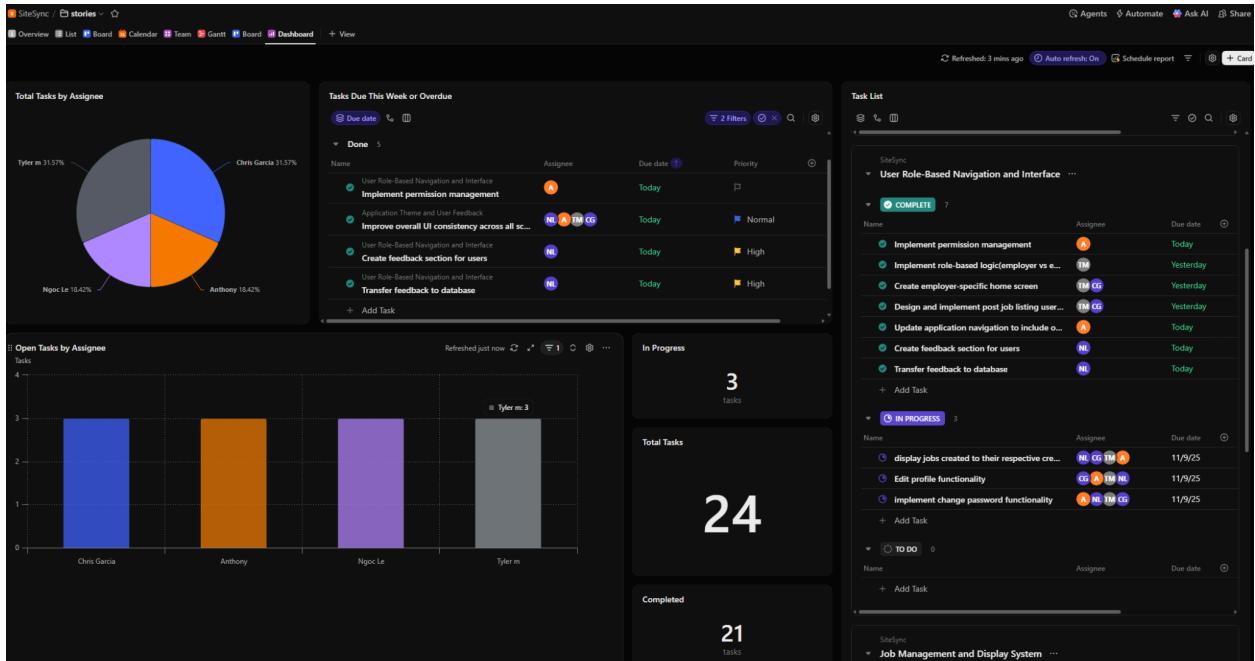
Implemented the display all active jobs in the job board fragment, via a recycler view, with information pulled from our database, and also implemented the functionality of which fragments are display depending on which type of user is logged in, an employer and employee, implemented the functionality of the post a job fragment, created a new page in the database of the new job being created. Changed where the user gets asked for notification permission to main activity instead of before they are logged in. I also fixed and updated some implementations of the settings fragment.

Ngoc Le:

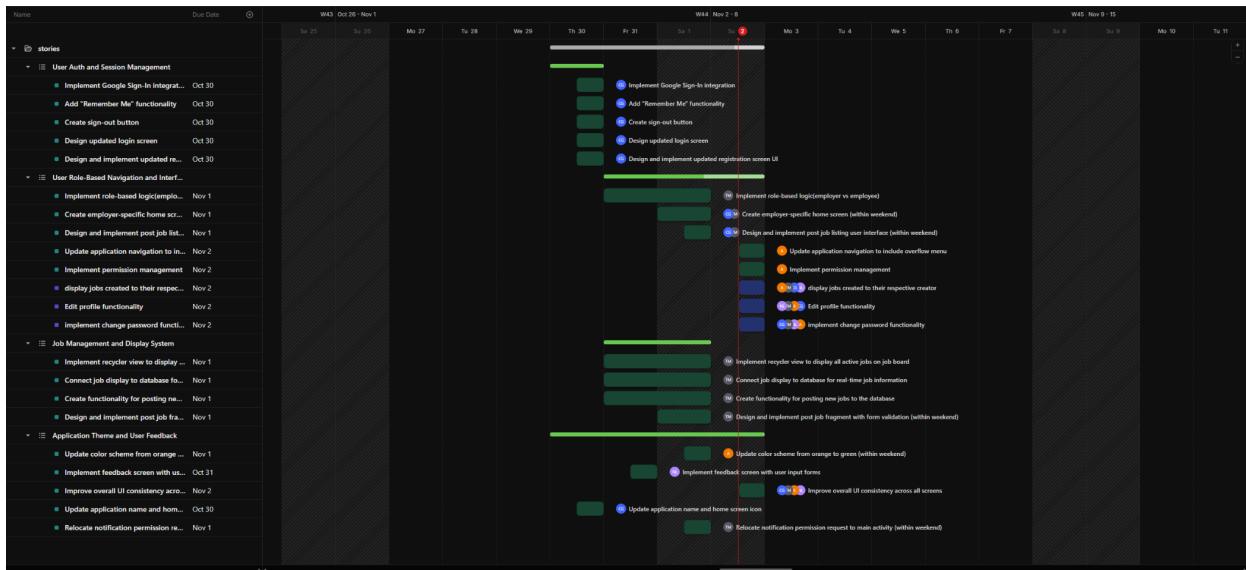
Created Feedback Screen and its functionalities including name, email, phone number, rating bar and feedback with a submit button. Feedback is now stored in the database. Moved profile from bottom navigation menu to sliding drawer and changed its icon.

18. Each member must have a minimum of 10 commits, counted starting Oct. 19 (Two-week Sprint).
19. Marks deducted for members who are not meeting the minimum 10 commits (i.e. if only 8 commits, 20% percent deducted, ...etc.).
20. Number of commits by each member for this sprint only.
 - Anthony Mancia (N01643670) = 16 commits
 - Chris Garcia (N01371506) = 17 commits
 - Ngoc Le (N01643011) = 12 commits
 - Tyler Meira (N01432291) = 18 commits
21. Sprint goals, list sprint goals.
 - 1) User authentication
 - 2) UI complete
 - 3) Upload Jobs Functionality
 - 4) Regular User and admin separation
 - 5) Update Application color theme
 - 6) Active jobs displayed
 - 7) Create feedback screen
22. Update the Sprint dashboard, showing Sprint 3 with closed tasks and tasks you did not complete.
23. Dashboard, should clearly show task, owner, status, start date, end date, priority and size.
24. Must use a tool such as Trello, Monday ...etc. Word, Excel, ... etc. will not be accepted.
25. Please use Scrum and not Kanban.

Scrum dashboard



26. Take a screenshot showing clearly the stories and breakdown of tasks with all details for sprint 3 only.
27. Must have a minimum of 4 stories and 5 tasks per story.
28. Update gantt chart showing main milestones and work progress. You can use <https://try.airtable.com/gantt2> or any other tool. Do the research and select the proper tool. Excel or word will not be accepted.
29. The gantt chart must show the work for a minimum of 10 components and the timelines. Focus on main components and not every small task.
30. Add a column to show the person assigned. An example of a Gantt chart:
31. Gantt chart should reflect the entire project from start to end. One of the main goals for gantt chart to know the final delivery date for this term.
32. Take a screenshot of the gantt chart.



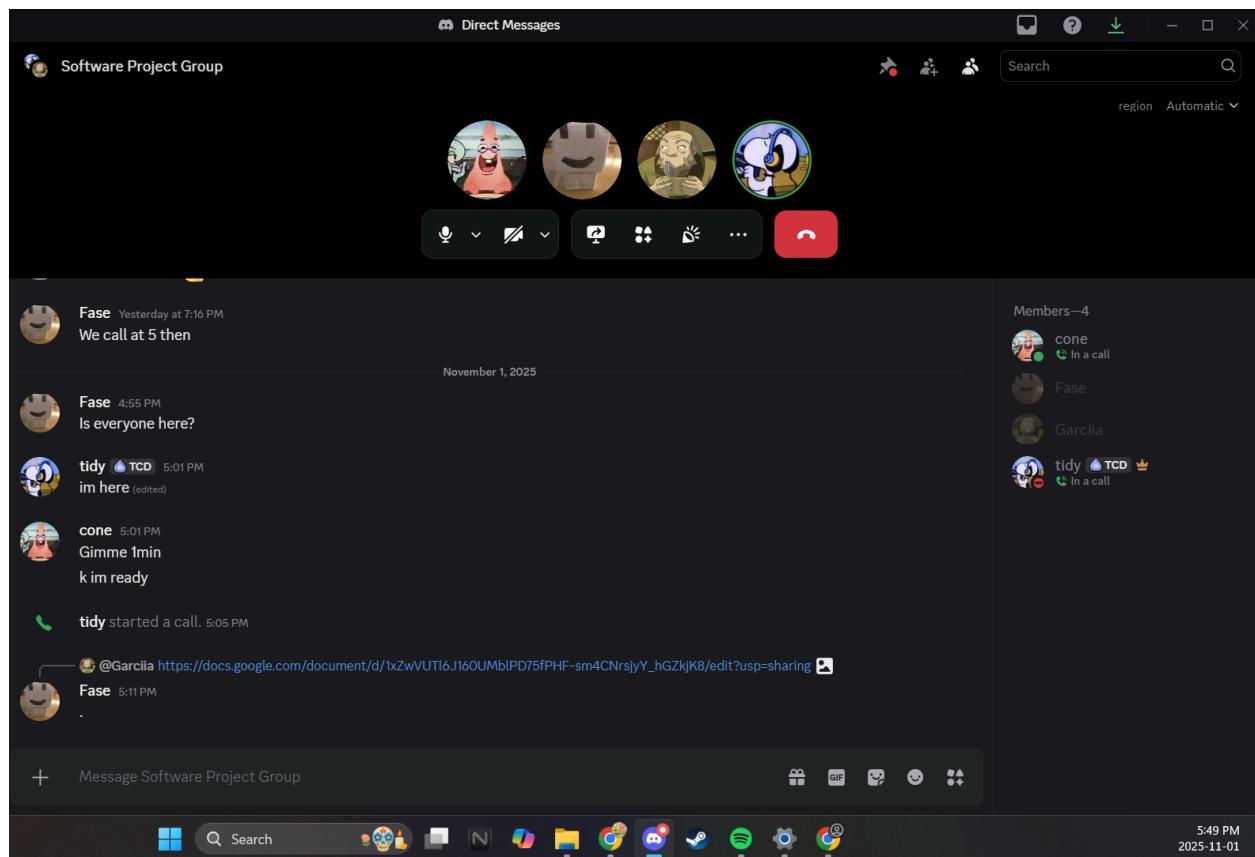
35. Record a minimum of 3 meetings.

October 31, 2025	Tyler Meira:	Chris Garcia	Ngoc Le	Anthony Mancia
What did you do yesterday?	Started implementing the job board fragment to display job from the database.	Worked on google sign in functionality, Log in and sign up UI, remember me functionality,	Checked feedback from last deliverable	Reviewed progress made and also review the things that needed to be changed
What will you do today?	today i will constitute working on the same functionality	Worked on Employer home screen UI, Employer Post Job UI	Improve the app based on feedbacks	Look through what need to be altered
What (if anything) is blocking your progress?	When initially creating the data base we added empty characters to the collections page, which caused errors when retrieving data from the collection	Implementation of both employee and employer functionality at sign in	Nothing is blocking	N/A

November 1, 2025	Tyler Meira:	Chris Garcia	Ngoc Le	Anthony Mancia
What did you do yesterday?	Finished implementing the job board	Worked on Employer home screen UI, Employer Post Job UI	Improved the app based on feedbacks	Looked over the
What will you do today?	today i will start working on the post job functionality	Completed UI for Post Job UI, and worked on profile Tab UI	Create Feedback screen ui for the app	Add the overflow menu and with four options to be able to click on
What (if anything) is blocking your progress?	road block was redefined the job adapter to make sure that all our new fields would make sense with the adapter	Struggled to implement the edit profile functionality for the user	Nothing is blocking	Nothing is being blocked

November 2, 2025	Tyler Meira:	Chris Garcia	Ngoc Le	Anthony Mancia
What did you do yesterday?	Finished implementing the job post functionality	Completed UI for Post Job UI	Created Feedback screen ui for the app	Finished adding the overflow menu and added permissions, about, and feedback screens in it (Permission is missing full functionality).
What will you do today?	today i will finish working on the job posting functionality and move the permissions code.	Worked on screen flow from employers home screen to post jobs and vise versa, and	Complete the functionalities for feedback screen and store it in database	Finish working on the overflow menus and try to add functionality to the permissions screen and also find thing of a 4th and final screen to place in the overflow menu

		worked on Employers home screen.		
What (if anything) is blocking your progress?	no road blocks today	Found it difficult to link the job posted by the admin to match the account that created it.	Nothing is blocking	Nothing was getting in the way of progress



36. Document two different design principles used in the code. Copy the code you used, and add your explanations.

1. In the code below we have used the design principle Singular responsibility principle. This is because, the task of this class is to fetch and load the jobs on to the employers UI, this code does not deal with permissions or other unrelated task.

```
private void loadJobs() {  
  
    db.collection(getString(R.string.jobs))  
  
        .whereEqualTo(getString(R.string.email),  
getString(R.string.test_company_com)) // Same email as PostJobsFragment  
  
        .get()  
  
        .addOnCompleteListener(task -> {  
  
            if (task.isSuccessful()) {  
  
                jobList.clear();  
  
                for (QueryDocumentSnapshot document : task.getResult()) {  
  
                    String company =  
document.getString(R.string.company);  
  
                    String description =  
document.getString(R.string.description));  
  
                    String status =  
document.getString(R.string.status));  
  
                    JobItems job = new JobItems(company, description,  
status);  
  
                    jobList.add(job);  
  
                }  
  
                jobAdapter.updateJobList(jobList);  
  
            }  
        }  
}
```

```
});
```

2.

37. Document one design pattern used in the code. Copy the code you used, and add your explanations.

- 1) This code changes how the back button normally works, by asking the user if they're sure they wish to exit the application. Contrary to what normally happens when the back button is pressed. The design pattern used here is the strategy pattern.

```
OnBackPressedCallback callback = new OnBackPressedCallback(true) {  
  
    @Override  
  
    public void handleOnBackPressed() {  
  
        new AlertDialog.Builder(MainActivity.this)  
  
            .setTitle(R.string.exit_application)  
  
            .setMessage(R.string.exitMsgMain)  
  
            .setIcon(R.drawable.sitesynclogo)  
  
            .setPositiveButton(R.string.YesButton, new  
DialogInterface.OnClickListener() {  
  
                @Override  
  
                public void onClick(DialogInterface dialog, int which) {  
  
                    finish(); // Closes the current activity  
  
                }  
  
            })  
  
            .setNegativeButton(R.string.NoButton, new  
DialogInterface.OnClickListener() {  
  
                @Override  
  
                public void onClick(DialogInterface dialog, int which) {  
  
                }  
  
            })  
  
    }  
  
};
```

```
        dialog.dismiss();

    }

} )

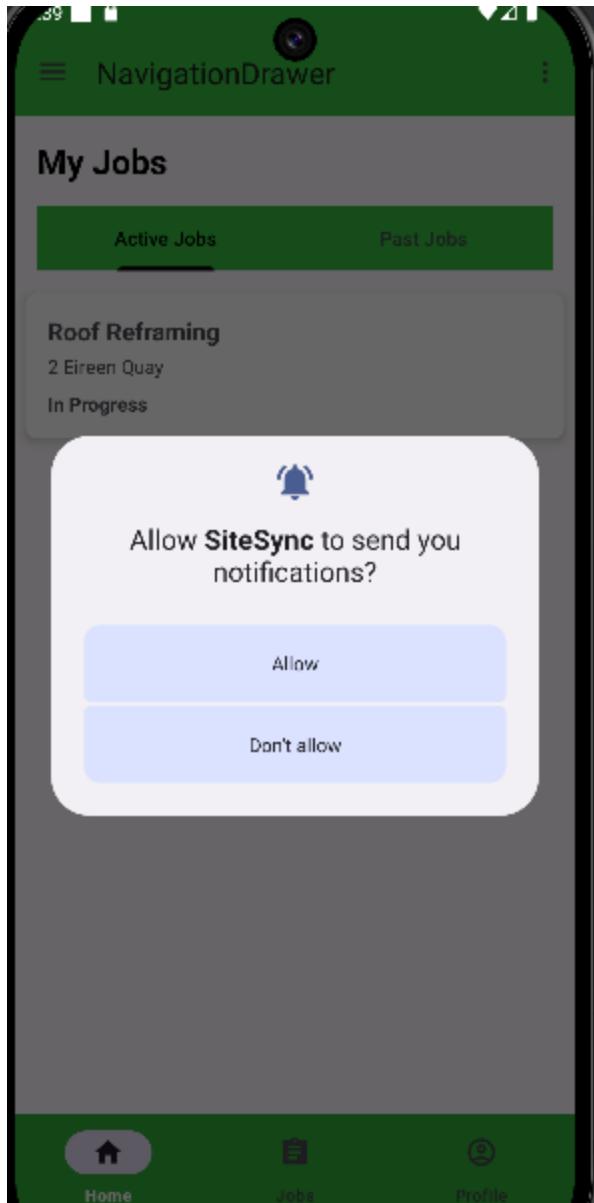
.show();

}

};

getOnBackPressedDispatcher().addCallback(this, callback);
```

38. Document what runtime permission you have implemented, with a screenshot.



39. Document the two main functionality of your app that was implemented in this release. Take screenshots of the two features. Comment on the two screenshots.

- 1) Create a listing

X

Listing details

Post

air canada

concrete work

200

pearson airport

2) display listing on home screen and job board

Home +

Job Listings

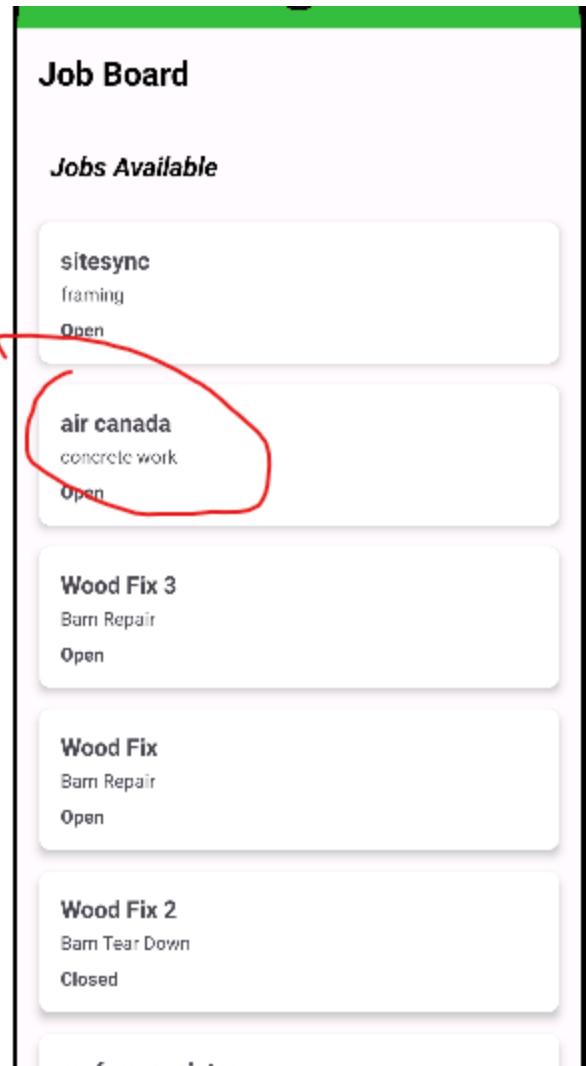
sitesync
Framing
Active

air canada (circled)
concrete work
Active

Wood Fix 3
Barn Repair
Active

Wood Fix 2
Barn Tear Down

rooferassociates
shingle repair
Active



40. Provide screenshot how the data from the Customer Feedback Screen stored in the DB. See example.

The screenshot shows the Google Cloud Firestore interface. On the left, there's a sidebar with a 'Feedback' collection and a sub-document 'w90xoKiPvrJclf...'. The main area displays the document details:

(default)	Feedback	w90xoKiPvrJclf...
+ Start collection	+ Add document	+ Start collection
Accounts	w90xoKiPvrJclf...	+ Add field
Feedback >		comment: "hello"
Jobs		deviceModel: "Google sdk_gphone64_x86_64"
		email: "123456@gmai.com"
		name: "ngoc"
		phone: "123456"
		rating: 4.5

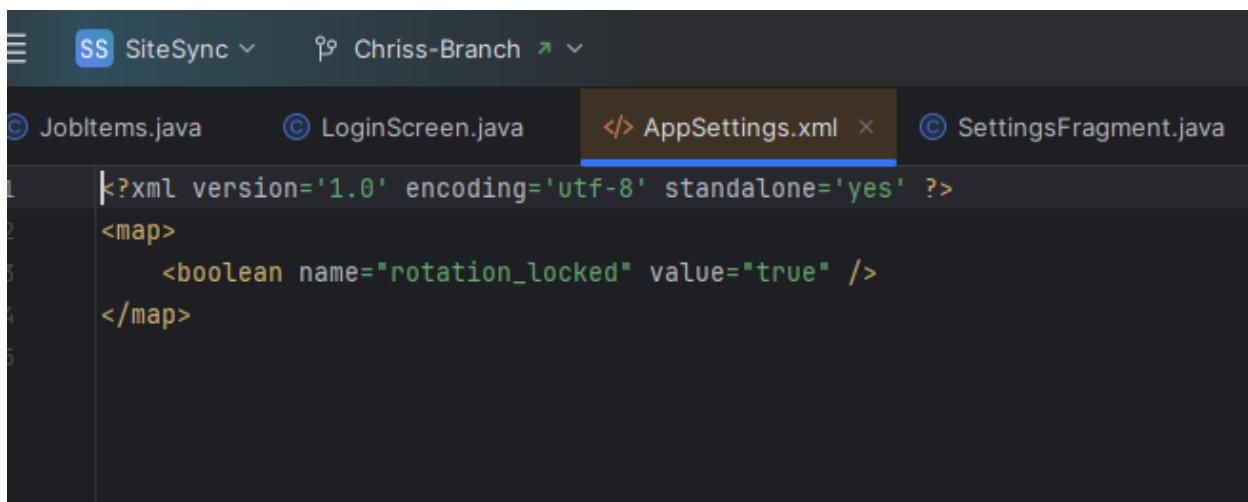
41. Create a subfolder called deliverable3 under docs in the repo and add the pdf file.
42. Push the pdf into github.

DELIVERABLE 3 PART 2

20- Should allow for configuration setting and remember user selections. Store data into SharedPref and read it back, i.e. if user selects Dark Theme, this option should be maintained when app restarts, device power off/on, ...etc. CENG-322 8 21- Take screenshot showing the data stored in the SharedPreference

21- Take screenshot showing the data stored in the SharedPreference

Rotation lock preference saved



The screenshot shows the Android Studio interface with the following details:

- Project Bar:** SiteSync is selected.
- Branch:** Chriss-Branch is shown.
- File List:** JobItems.java, LoginScreen.java, AppSettings.xml (highlighted with a blue bar), and SettingsFragment.java.
- Code Editor:** The AppSettings.xml file is open, displaying the following XML content:

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <boolean name="rotation_locked" value="true" />
</map>
```