

Iteration 3

Group 13

Christopher Phillips

Haoran Zhang

Khaleda Efat Jahan

Mohammed Al-Ajmi

Ruizhe Wu

Seif Elbayomi

Vishwa Patel

Date: April 9th, 2023.

Iteration Plan

1. User Stories & Acceptance Criteria

Provide your git project dashboard link: <https://git.cs.dal.ca/alajmi/group13project.git>

GitLab repo: <https://git.cs.dal.ca/alajmi/group13project.git>

2. Tasks Assignment: Acceptance test and Engineering task

[Necessary Field --> labels,task description, member assigned, start date, end date(expected),
status [pending,completed,in progress]

Corresponding User Story and Acceptance Criteria	Task(s)	Estimate	Member Assigned	Start	Expected	Status

US19: As an employee, I want to receive a notification whenever a new job is posted so that I can apply as early as possible.	1. UX/UI Design of the push notifications. 2. Create code to extract the push notifications' content from the database (whenever a job is posted). 3. Set up Firebase Cloud Messaging Services	5	Christopher Phillips	Mar 5, 2023		Not completed
AT19.1 I will never receive a job ad which I can't apply for.	User can close the ads they don't want to see with a button	5	Christopher Phillips	Mar 5, 2023		Not completed
AT19.2 I can set up the condition which related to the kinds of jobs I prefer.	Users can change the performance of that in the Personal Information page	5	Christopher Phillips	Mar 5, 2023		Not completed
US 37 : As an employee, I want to be able to apply for the job I want so that I can get in contact with the employer and start the job.	1-Employer can see the applicants for a specific job 2-Applicant information is sent to the databas 3-Employee Can apply for a job	6	Mohammed Jasim Al-Ajmi	Mar 25, 2023	April 9, 2023	Completed
AT 37.1: I can check the phone number on every job description pages	We can add one more text information for the contact number	6	Mohammed Jasim Al-Ajmi	Mar 25, 2023	April 9, 2023	Completed

AT 37.2: the employer can reject the application from others who they don't want to hire.	There should be a button for employer to refuse the call from the employees they don't want.	6	Mohammed Jasim Al-Ajmi	Mar 25, 2023	April 9, 2023	Completed
US 39: As an employee, I want to get recommended jobs based on my preferences so that I can find jobs easily.	1. Added a facility to recommend a jobs based on user preference in a separate activity. 2. Used the Relatime Database to store these preferences.	5	Khaleda Efat Jahan	March 25 th , 2023	April 9 th , 2023	Completed
AT 39:1 A user can set his/her job preference so that he/she can find the relevant jobs and update the information.	3.To update the job preferences, I have added a menu button named Edit Preference in RecommendedJobsActivity.	5	Khaleda Efat Jahan	March 25 th , 2023	April 9 th , 2023	Completed
AT 39:2 Upon launching the activity, the jobs already available will go through some conditional checking. If any of	4. I have added a FloatingActionButton in the EmployeeHomeActivity activity to launch this activity.	5	Khaleda Efat Jahan	March 25 th , 2023	April 9 th , 2023	Completed.

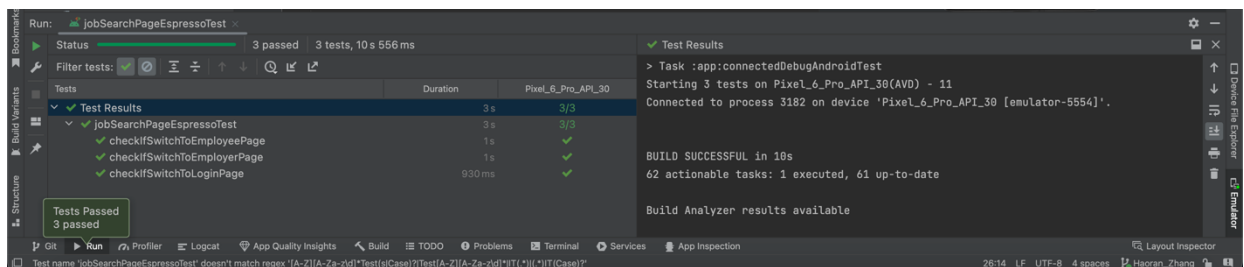
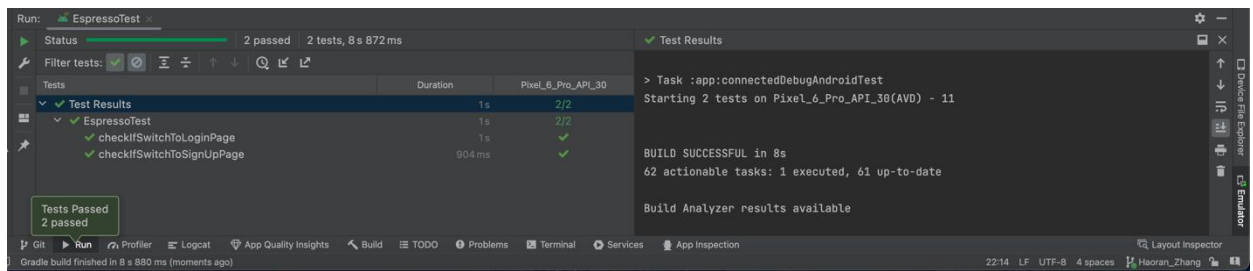
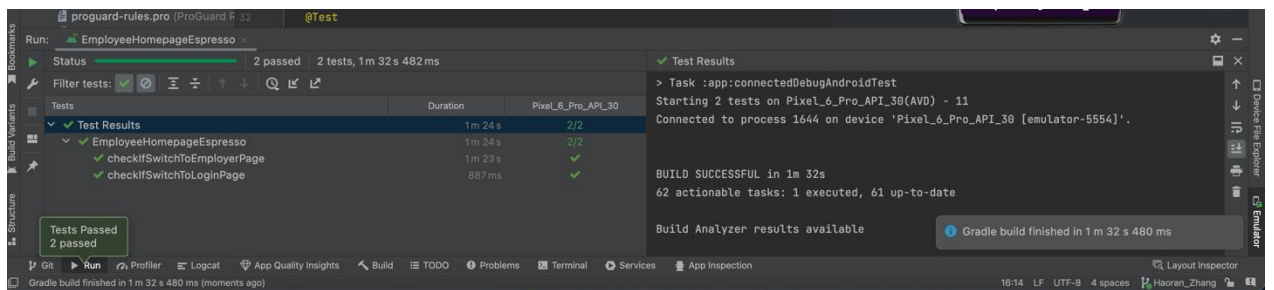
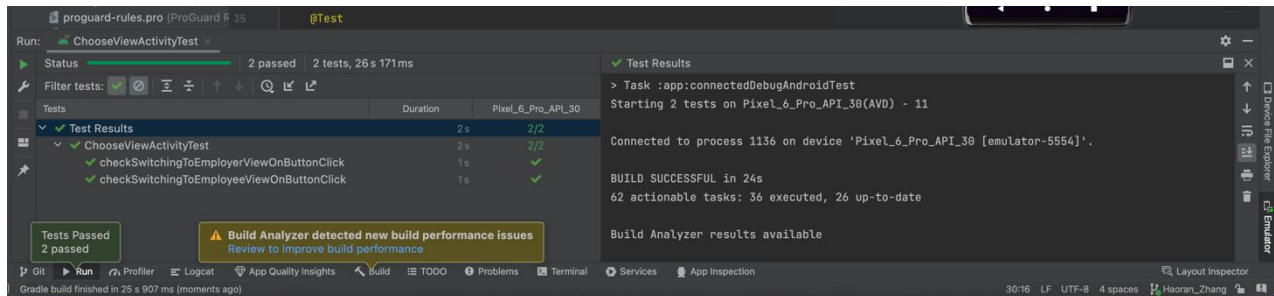
those jobs meet any criteria of the job preference, then they will be displayed in the list view; otherwise, they will not be displayed.						
US40: As a user, I want to be able to visualize the job history, income, and reputation of other users so that I know which applicant has better rating and trustworthy.	1. Integrate the new feature with the existing application. 2. Create a user interface to display the job history, income, and reputation of users. 3. Retrieve the job history, income, and reputation of users from the database.	8	Seif ELBayomi	Mar 26, 2023	April 9 th ,2023	Completed.
AT40:1: as a user, I want to see the rank of jobs from high to low	Make a button (rank it from high to low) on the job posting page	8	Seif ELBayomi	Mar 26, 2023	April 9 th ,2023	Completed.
AT 40.2 : the employee can also see the rating of their own, and make comments under them	Create a text box that employers can edit to allow them to edit comments	8	Seif ELBayomi	Mar 26, 2023	April 9 th ,2023	Completed.
US 36: As a user, I want to be able to pay/receive payments for jobs	1. Integrated PayPal into the application where the employer	6	Vishwa Patel	April 6 th ,2023	April 9 th ,2023	Completed

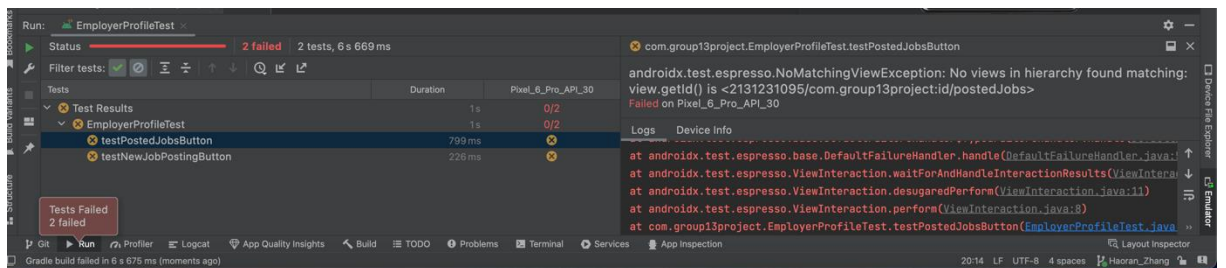
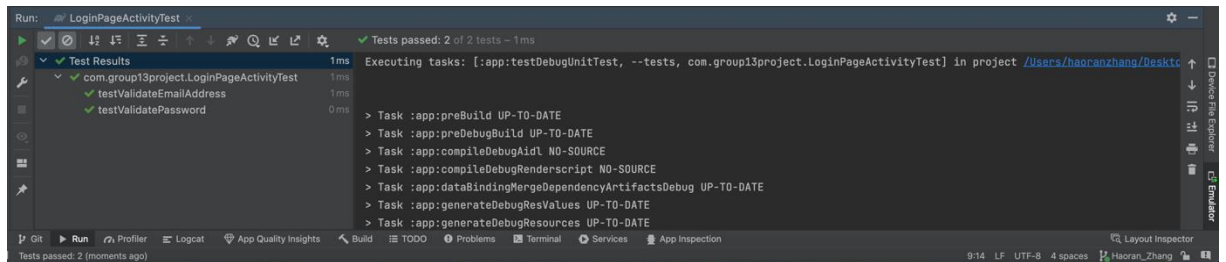
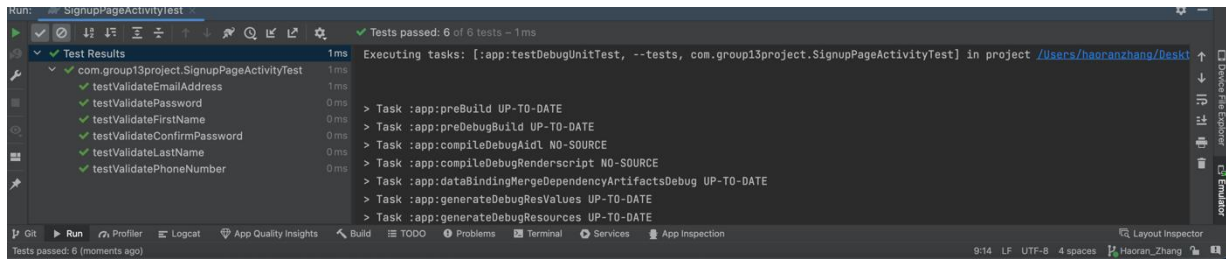
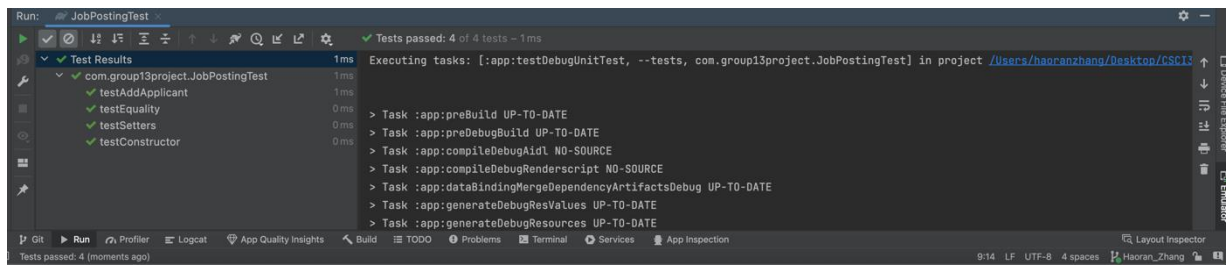
so that I get/give compensations for jobs immediately.	can enter their card details and pay compensation instantly to the applicant, they want to hire					
AT 36.1: as a user, I want the app owner to take the deposit to protect the rights of us, (the employee may give less money)	The employer should pay a deposit of a certain amount at the time of confirming the employment relationship	2	Vishwa Patel	April 6 th ,2023	April 9 th ,2023	Completed
AT 36.2: as a user I want set the payment method as default, so that I don't need to input the numbers frequently.	Create a checkbox next to the payment method and let the user decide if they need to set it as the default	4	Vishwa Patel	April 6 th ,2023	April 9 th ,2023	Completed
US 14: As a user, I want to be able to switch between employer view and employee view so that I can switch back and forth between two functionalities	<ol style="list-style-type: none"> 1. Adding new button call: my profile button on the choose view activity. So user can enter their own profile directly from the main page. 2. Added Espresso test. 3. Link button to the profile page. 4. Deleted my profile button in employer view activity 	3	Ruizhe Wu	April 1	April 6 th	Completed

Progress Charts

Testing Reports

Screenshots of the testing reports





Manual Tests:

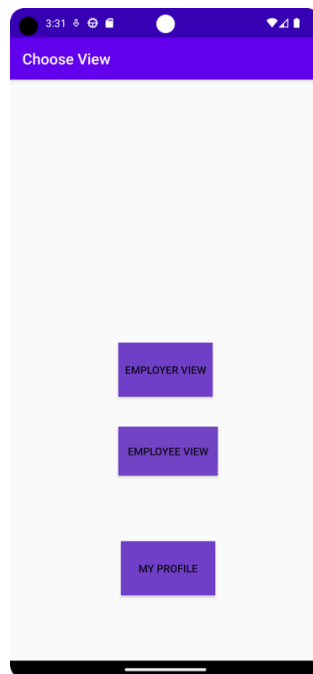
Users sign up for the application:

A screenshot of a mobile app interface showing a sign-up form. At the top, a purple header bar contains the "Quick Cash" logo and the text "Please sign up here". The form fields are: "First name" (Ruizhe), "Last name" (Wu), "Email address" (rz347594@dal.ca), "Phone number" (9020987654), "Password" (masked with dots), and "Confirm password" (masked with dots). A yellow "SIGN UP" button is at the bottom.

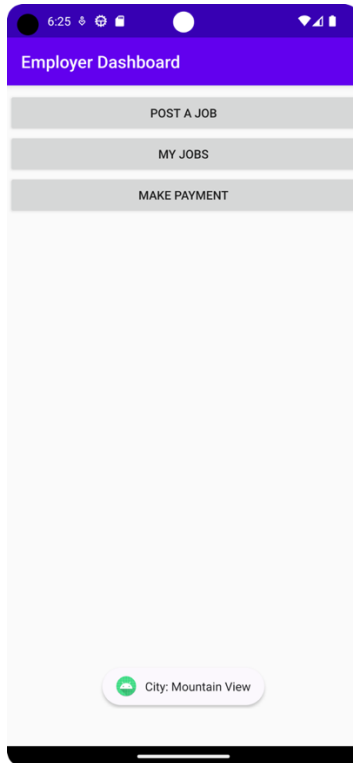
Login:

A screenshot of a mobile app interface showing a login form. At the top, a purple header bar contains the "Quick Cash" logo. Below the header, there are two input fields: one for email (rz347594@dal.ca) and one for password (masked with dots). A blue "LOG IN" button is at the bottom.

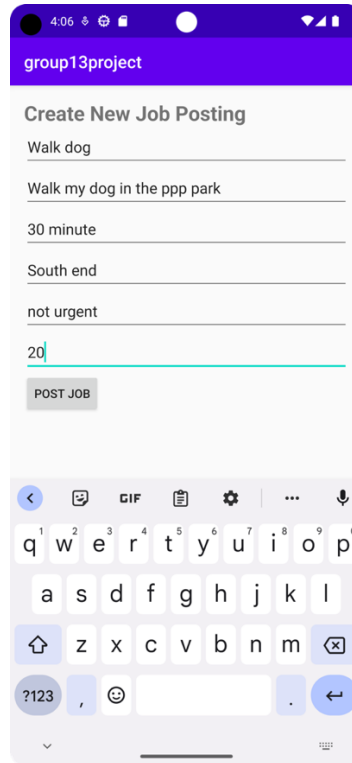
The choose view page:



Employer page:



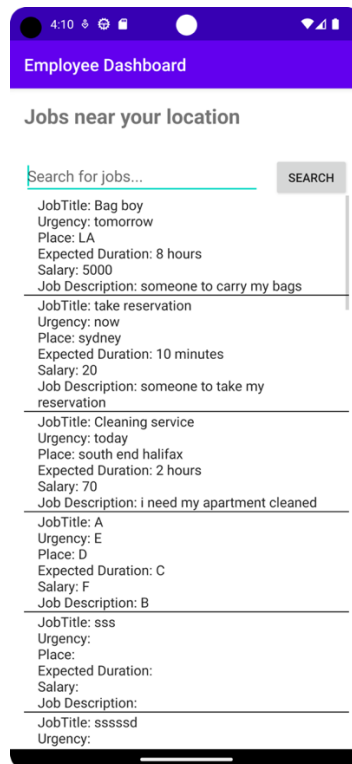
Creating new Job:



New Job created:



Employee page:



Search for job:

Employee Dashboard

Jobs near your location

Walk dog

SEARCH

JobTitle: Walk Dog
Urgency: not urgent
Place: south end
Expected Duration: 30 minute
Salary: \$20
Job Description: Walk my dog in the ppp park

dog dogs

q w e r t y u i o p
a s d f g h j k l
z x c v b n m
?123 , .

Apply for job:

Description

JobTitle: Bag boy
Urgency: tomorrow
Place: LA
Expected Duration: 8 hours
Salary: 5000
Job Description: someone to carry my bags

Confirm Application

Are you sure you want to apply for this job?

NO YES

APPLY

Fill in Job preferences:

My Job Preference

Expected Duration
30

Expected Job Location
south end

Expected Salary
20

SAVE PREFERENCE

< GIF

q w e r t y u i o p
a s d f g h j k l
z x c v b n m
?123 , .

Recommended Jobs base on preferences:

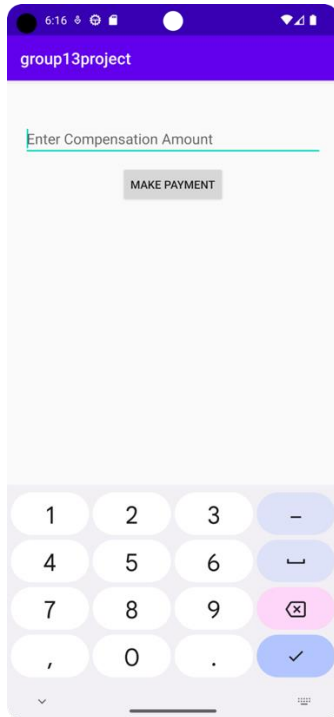
Recommended Jobs

JobTitle: take reservation
Urgency: now
Place: sydney
Expected Duration: 10 minutes
Salary: 20
Job Description: someone to take my reservation

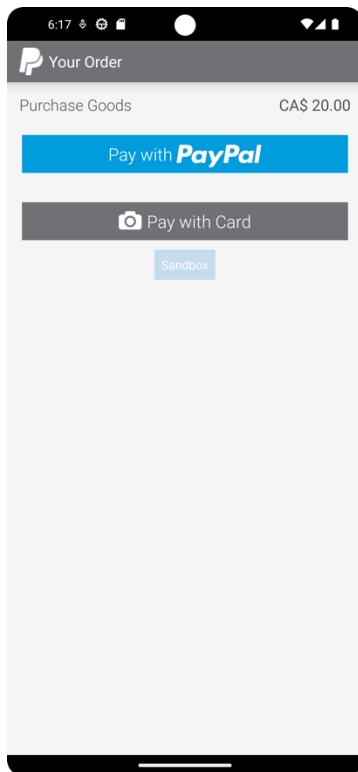
JobTitle: Walk Dog
Urgency: not urgent
Place: south end
Expected Duration: 30 minute
Salary: \$20
Job Description: Walk my dog in the ppp park

City: Mountain View

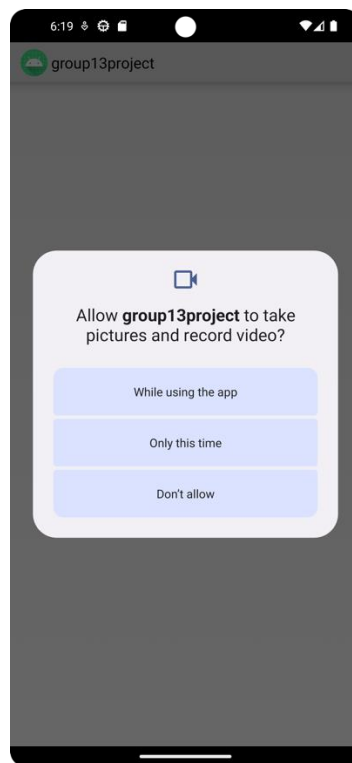
Make payment:



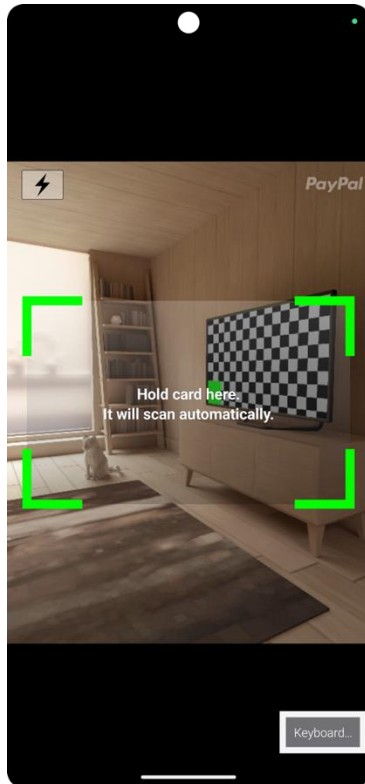
Select how to input card information:



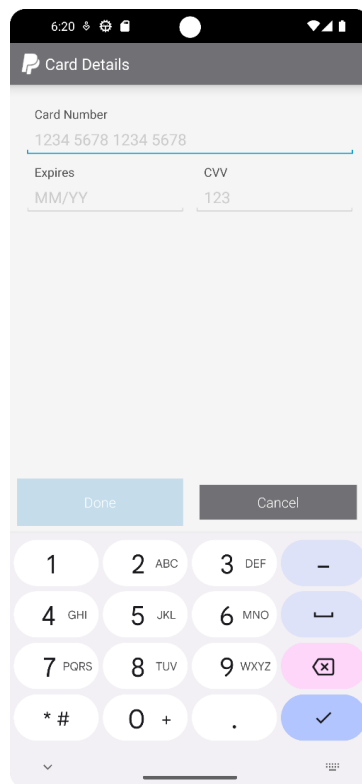
Ask for permission:



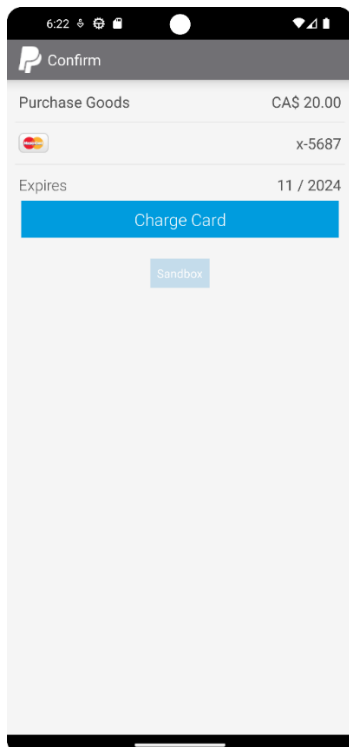
Take picture of the credit card:



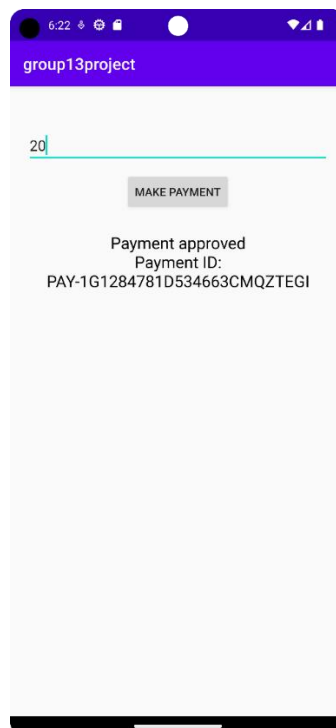
User enter the details of the card:



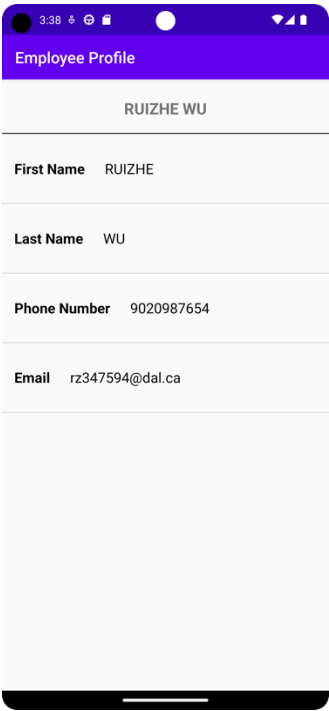
App confirm the card is Valid:



Payment approved:



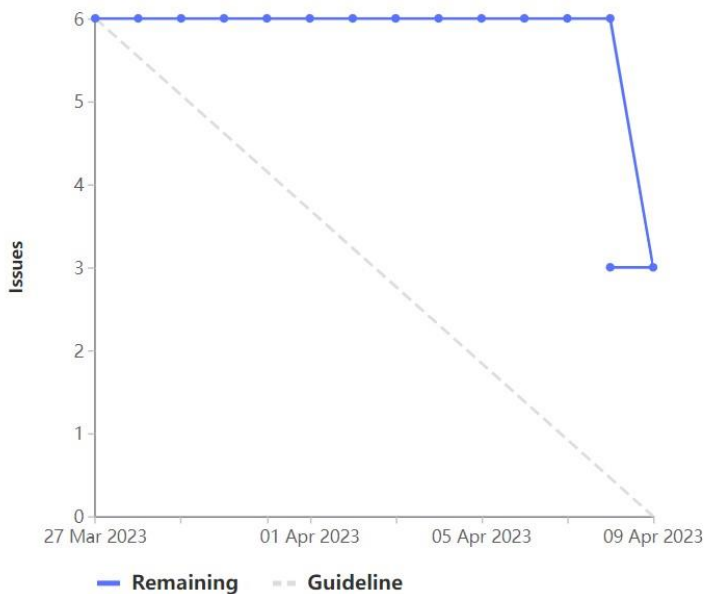
User's profile



Burndown Chart

As we discussed in the client TA meeting, we were splitting each user story into tasks, and each team member was assigned a user story consisting of multiple tasks. Whenever we finished a task, it was marked as completed. However, the task's completion times are not reflected on the burndown charts. Thus, the burndown chart looks flat, despite the team having continuous progress.

Burndown chart



Velocity Chart

Commit statistics for **Development** Jan 28 - Apr 09

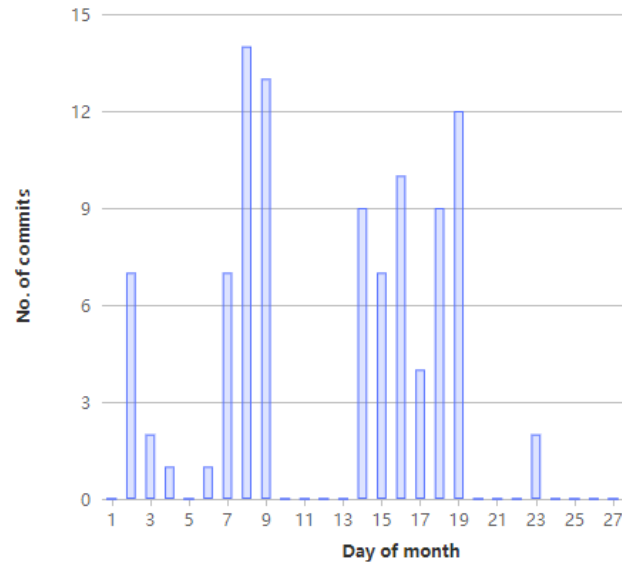
Excluding merge commits. Limited to 2,000 commits.

- Total: **102 commits**
- Average per day: **1.4 commits**
- Authors: **10**

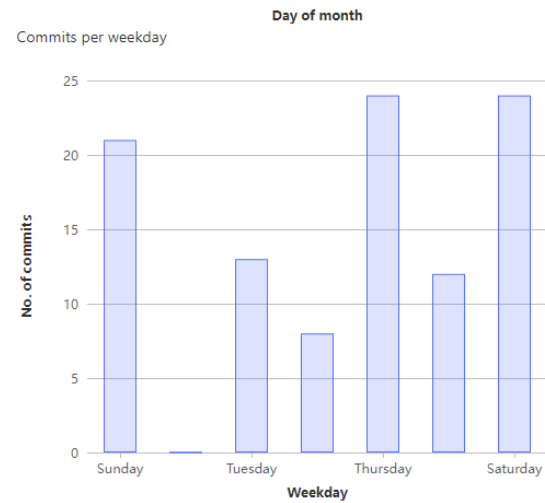
Development

group13project

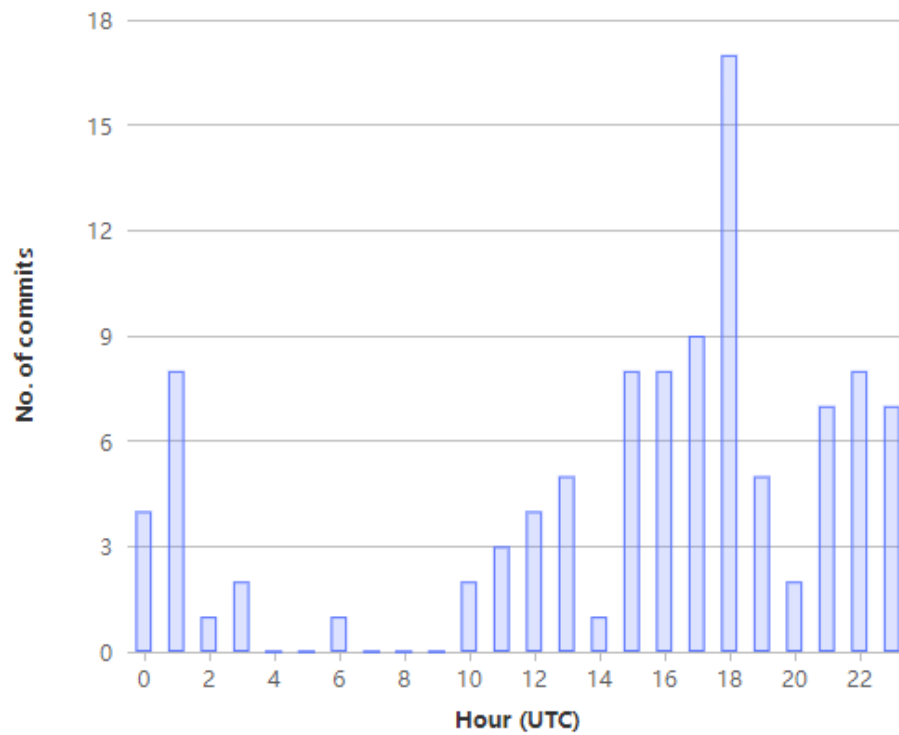
Commits per day of month



Commits per weekday



Commits per day hour (UTC)



Unstarted Issues (open and unassigned)

0

Ongoing Issues (open and assigned)

1 5

As an employee, I want to receive a notification whenever a new job is posted so that I can apply as early as possible.

#19 In progress 5

Completed Issues (closed)

6 34

As a user, I want to be able to visualize the job history, income, and reputation of other users so that I know who are good ones and trustworthy.

#40 8

As an employee, I want to get recommended jobs based on my preferences so that I can find jobs easily.

#39 3

As a user, I want to be able to chat with other employees/employers so that I can discuss the details of the job.

#38 6

As an employee, I want to be able to apply for the job I want so that I can get in contact with the employer and start the job.

#37 In progress 6

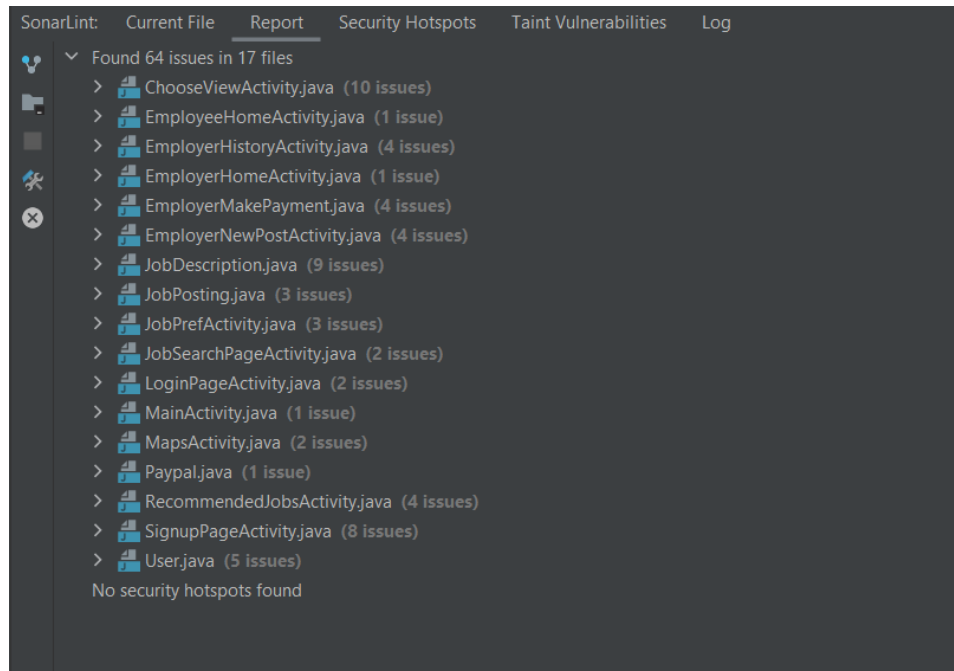
As a user, I want to be able to pay/receive payments for jobs so that I get/give compensations for jobs immediately.

#36 6

As an employee, I want to be able to store my preferences for job types so that I can be alerted when a job of my preference is posted.

#35 Cancelled Task 5

SonarLint and Code Refactoring



Violations:

SonarLint identified several violations in the codebase. After analyzing the violations, we found the following three violations to be the most important:

1. Not using the `@Override` annotation: This violation was identified as a critical issue, as it can lead to subtle bugs when a method is intended to override a superclass method but does not due to a typo or a mismatch in method signatures. By using the `@Override` annotation, the compiler ensures that the method is actually overriding a superclass method, reducing the risk of these types of errors.
2. Anonymous inner class containing only one method should become lambda: This violation was identified as a major issue, as it can reduce code readability and maintainability. Before Java 8, the only way to partially support closures in Java was by using anonymous inner classes, but their syntax may seem unwieldy and unclear. With Java 8, most uses of anonymous inner classes should be replaced by lambdas to highly increase the readability of the source code.
3. Not merging collapsible if statements: This violation was identified as a minor issue, as it can affect code readability. Merging collapsible if statements can simplify the code and improve its readability, making it easier for developers to understand and maintain.

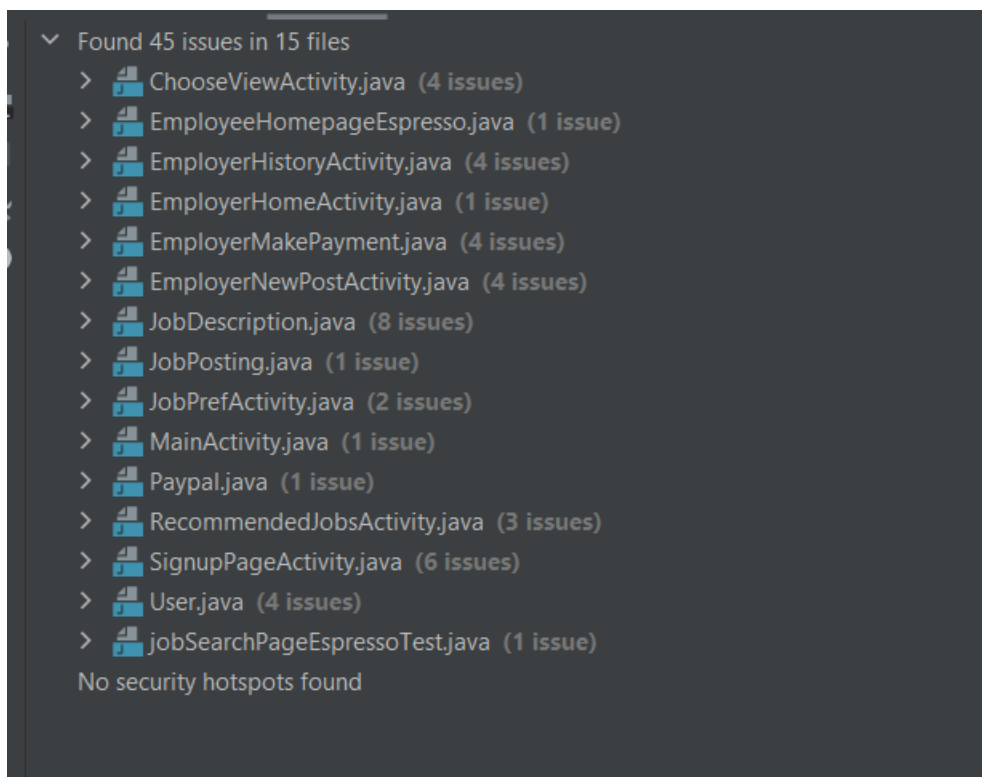
Actions taken:

To address the identified violations, we took the following actions including but not limited to:

- We added the `@Override` annotation to methods that were intended to override superclass methods, ensuring that the compiler checks for proper overriding and reduces the risk of errors.
- We replaced anonymous inner classes containing only one method with lambda expressions, significantly improving the readability and maintainability of the source code.
- We merged collapsible if statements to simplify the code structure and improve its readability.

You can find a lot of the code refactoring done under the SonarLint commit.

Screenshot after refactoring and decreasing the number of issues from 64 to 45:



SOLID Principles

Single Responsibility Principle (SRP):

We have implemented the Single Responsibility Principle by designing each class with a particular unique purpose, ensuring that each class has its own specific properties and methods that should be changes only in that class. For example, our EmployerHistoryActivity class functions only getting all the job posting from the database for that particular employer and display it on its separate page in the application. Similarly, RecommendedJobsActivity focuses on only receiving all the preferences of the employee to search for the job from the list of jobs displayed on Employee Homepage and recommend them on separate page to apply. By following the Single Responsibility Principle, we have ensured that our code is modular and easy to maintain, and every activity class is for a specific purpose.

Open/Closed Principle (OCP):

The Open/Closed Principle was considered when designing our application. Our classes are available for extension but closed for alteration, as we have made sure. As an illustration, the JobApplication class may be expanded by adding new functionality to a new class that inherits from it without changing the JobApplication class itself. With the help of this strategy, we were able to develop an expandable and adaptable architecture.

Liskov Substitution Principle (LSP):

We have also made sure that the Liskov Substitution Principle is followed by our code. We have utilized interfaces, such as JobPref in the JobPrefActivity class, to ensure that our classes deliver the same behavior as JobPref as stated by the interface even if there are no explicit inheritance connections in our code. Similarly, JobApplication is used in JobDescription class too. To provide a constant and dependable application behavior, this enables our classes to be utilized interchangeably with other classes that implement the same interface.

Design Patterns

Singleton (Creational):

The Singleton pattern is utilized in the project for managing the FirebaseDatabase instance. In the EmployerHistoryActivity and EmployeeHomeActivity classes, the FirebaseDatabase instance is retrieved using the following line of code:

```
FirebaseDatabase database = FirebaseDatabase.getInstance();
```

FirebaseDatabase uses a Singleton pattern to ensure that only one instance of the database is created and shared across the entire application. This helps maintain data consistency and reduce resource usage.

Adapter (Structural):

The Adapter pattern is used in the JobDetailAdapter class, which is a custom ArrayAdapter for displaying job postings in a ListView. The Adapter pattern allows the ListView to work with the JobPosting objects without modifying their structure. The JobDetailAdapter class extends ArrayAdapter<JobPosting> and overrides the getView method to create and populate the custom view for each item in the ListView.

Observer (Behavioral):

The Observer pattern is used in both the EmployerHistoryActivity and EmployeeHomeActivity classes with the ValueEventListener instances. ValueEventListener acts as an Observer, monitoring changes in the Firebase Realtime Database.

For example, in EmployerHistoryActivity, the ValueEventListener for job postings (jobListener) is defined and attached to the DatabaseReference (jobRef).

Similarly, in EmployeeHomeActivity, ValueEventListener is used for monitoring job postings.

The Observer pattern allows the ValueEventListener instances to be automatically notified when the data in the Firebase Realtime Database changes. This pattern helps keep the UI up-to-date without manual intervention, improving the user experience.

Regular Standup Meetings

1. General Meeting: 2023-03-21

Time and location: Teams

Attendance: Ruizhe Wu, Mohamed Jasim, Seif ELBayomi, Haoran Zhang, Christopher Phillips, Khaleda Efata Jahan, Vishwa Patel, Christopher Phillips.

Documented by: Khaleda Efata Jahan.

Summary: (30 mins)

Our group discussed and agreed to work on the unfinished tasks from iteration 2 instead of waiting for the new requirements to be announced for iteration 3. We agreed to finish previous tasks before Thursday's meeting so that we can be ready to work on the new requirements.

What to do after meeting:

1. Mohammed will work on employee view of all job postings.
2. Mohammed will work on employee search functionality for job postings.
3. Christopher will continue to work on Notifications.
4. Editing the location field of the job posting class. This will by default be the user's location provided by their device, meaning they will now be able to choose the location of the post, it will be set to their device's location.
5. Moving the user profile from the employee view to the choose view activity because it is not specific to the employee.
6. Improving the visual appearance of the app.

2. General Meeting: 2023-3-23

Time and location: Teams.

Attendance: Ruizhe Wu, Mohamed Jasim, Seif ELBayomi, Haoran Zhang, Christopher Phillips, Khaleda Efata Jahan, Vishwa Patel, Christopher Phillips.

Documented by: Khaleda Efata Jahan.

Summary: (25 mins)

We discussed with the TA and team members regarding our user stories and how many tasks are left to finish the project. We also discussed the previous iteration and the unfinished tasks with the TA.

What to do after meeting:

- Discussion with the TA regarding functionalities of previously assigned tasks as well as upcoming tasks.
- Discussion of doing medication of the functionalities of the app.
- Clarifications from the TA of what is left to do for iteration 3.
- Discussed to the TA regarding iteration 3 rubric.
- For the rest of the meeting, everyone discussed their progress.
- Decided the next date to arrange a meeting give user stories and tasks to each other for iteration 3.

3. General Meeting: 2023-3-25

Time and location: Teams.

Attendance: Ruizhe Wu, Mohamed Jasim, Seif ELBayomi, Haoran Zhang, Christopher Phillips, Khaleda Efata Jahan, Vishwa Patel, Christopher Phillips.

Documented by: Khaleda Efata Jahan

Summary: (32 mins)

We assigned user stories and tasks to each team members on Gitlab. Additionally, we read the documents together and explained each of the user stories individually for better understanding.

What to do after meeting:

- Ruizhe will work on his previous task as well as being assigned to work on the alert system for job preference.
- Vishwa was assigned to work on PayPal integration.
- Khaleda will work on the recommended job functionality based on users' preferences.
- Mohammad will work on the job applying functionality of the app so that the employee can contact the employer.
- Seif will work on the functionality of filtering job seekers and providers based on given criteria.
- Christopher will continue to work on the notification function to receive new job alerts.
- Haoran will work on the chat feature of the app.
- Let the TA know regarding the assigned user stories.

4. General Meeting: 2023-3-28

Time and Location: Teams

Attendance: Ruizhe Wu, Mohamed Jasim, Seif ELBayomi, Haoran Zhang, Christopher Phillips, Khaleda Efata Jahan, Vishwa Patel.

Documented by: Khaleda Efata Jahan

Summary (33 min):

We reported on each member's work and reconfirmed each person's assigned user stories and sent out a list of invitations to each member on firebase.

What to do after meeting:

- Everyone discussed their progress of the user stories.
- Helped solve difficulties in each other's tasks.
- Each member should focus on completing their user story.
- Discussed about availability option for the presentation date.

5. General Meeting: 2023-4-4

Time and Location: Teams

Attendance: Ruizhe Wu, Mohamed Jasim, Seif ELBayomi, Haoran Zhang, Christopher Phillips, Khaleda Efata Jahan, Vishwa Patel.

Documented by: Khaleda Efata Jahan

Summary (27 min):

Everyone reported what they have accomplished so far and discussed the merge request activities.

What to do after meeting:

- Everyone discussed their progress of the user stories.
- Discussed merge conflicts and acceptance.
- Each member should focus on completing their user story.
- Planned to start working on the report.

6. General Meeting: 2023-4-9

Time and Location: Teams

Attendance: Ruizhe Wu, Mohamed Jasim, Seif ELBayomi, Haoran Zhang, Christopher Phillips, Khaleda Efat Jahan, Vishwa Patel.

Documented by: Ruizhe Wu

Summary (50 min):

Report what has been done for the app and discuss what kind of final improvements we need.

What to do after the meeting:

- Mohamed will be working on the Job application from the Employer side
- Seif will do the test for Mohamed's classes.
- Seif will be working on allowing the employer to view the employee profile
- Seif will work on sonarlint
- Vishwa will keep working on Paypal.
- Khaleda will move the view of the preferred job to a separate activity
- Christopher will keep working on the notification
- Haoran will write the documentation of SOLID principle
- Haoran will improve some XML files
- Ruizhe will add user stories and acceptance tests for all tasks of this iteration.

Iteration Post-mortem Review

Issues With Development

1. Sometimes the program is not running properly for some members when they try to push their code to the group repository.
2. Some of the members were having issues with firebase connection.
3. Completing assigned tasks on time.
4. Employer and employee has sperate profile.
5. Some of the classes have too much functionality.
6. Some members cannot finish their own tasks.
7. Some members are assigned tasks that are not in the requirement or are already included in other member's tasks.

Solutions

1. Some member has to change their Emulator on the machine and some member ha to create a new file and pull the project again. Most importantly is for each member to only commit the file that they made changes on instead of committing every single file in their project.
2. Delete the member from the firebase and invite again.
3. Communica more frequently, we add another meeting on every Saturday night to report project progress and help with troubleshooting.
4. Create an button on choose view activity to direct the user to the profile page.
5. Sperate them into smaller subclass and make sure the code pass the testes.
6. Pass their task to other members with the ability to accomplish it and assign new tasks to the member.
7. Cancel the task and assign a new task to the member.

Goals for Upcoming Iteration

If this project had a future iteration, we would have had the following goals:

- 1- Improve version control and collaboration:
 - Ensure all team members have a clear understanding of Git best practices to avoid conflicts when pushing code to the group repository.
- 2- Strengthen Firebase connectivity and troubleshooting:
 - Provide team members with documentation and resources on Firebase connection and setup.
- 3- Enhance time management and task completion:
 - Encourage team members to communicate proactively if they anticipate delays or require assistance.
- 4- Refine employer and employee profiles:
 - Review the user experience and interface design for employer and employee profiles.
 - Gather feedback from team members to identify areas for improvement and implement necessary changes.
- 5- Optimize code structure and modularity:
 - Encourage team members to follow the Single Responsibility Principle and keep classes focused on specific functionalities.
 - Conduct code reviews to identify opportunities for refactoring and improving code quality.