Gyg

Shawn Mitchell

Isuru Nanayakkara

Jonathan Luetze

J. Anthony Timberlake

Andrew Vivian

Outline

- Overall Description
- Functional Requirements
- Nonfunctional Requirements
- Diagrams
- Timeline

Overall Description

Gyg makes it easy to:

- Find simple jobs to do in your area.
- Post new jobs that you need help with.
- Establish a trustworthy medium for getting tasks done.
    - As opposed to sketchy non-community based applications.

How it works

ACTION 1: FIND GYGS

1. Find Gygs that they want to do for someone else
    a. Filter by User and/or Area
2. Post a Gyg that they want someone else to do for them
3. View Gygs
    a. Past and upcoming to do’s
    b. Posted Gygs

How it works

ACTION 2: COMPLETE GYGS

1. User selects a Gyg that they want to do
2. Both users scan a QR code to start and finish a Gyg
3. User that was helped can comment on the user who worked

How it works

ACTION 3 : VIEW PROFILE

1. Name
2. Picture
3. General Working Area
4. Skillset
5. Description
6. View past transactions

Front End and Back End Expectations

- **Front End**
  - Material Design guidelines
  - User friendly layout
  - Simple UI
- **Back End**
  - Firebase - Database storage and transaction records.
  - Node.js - For data crunching and networking
  - Paypal / Plaid APIs for financial processing.

Nonfunctional Requirements

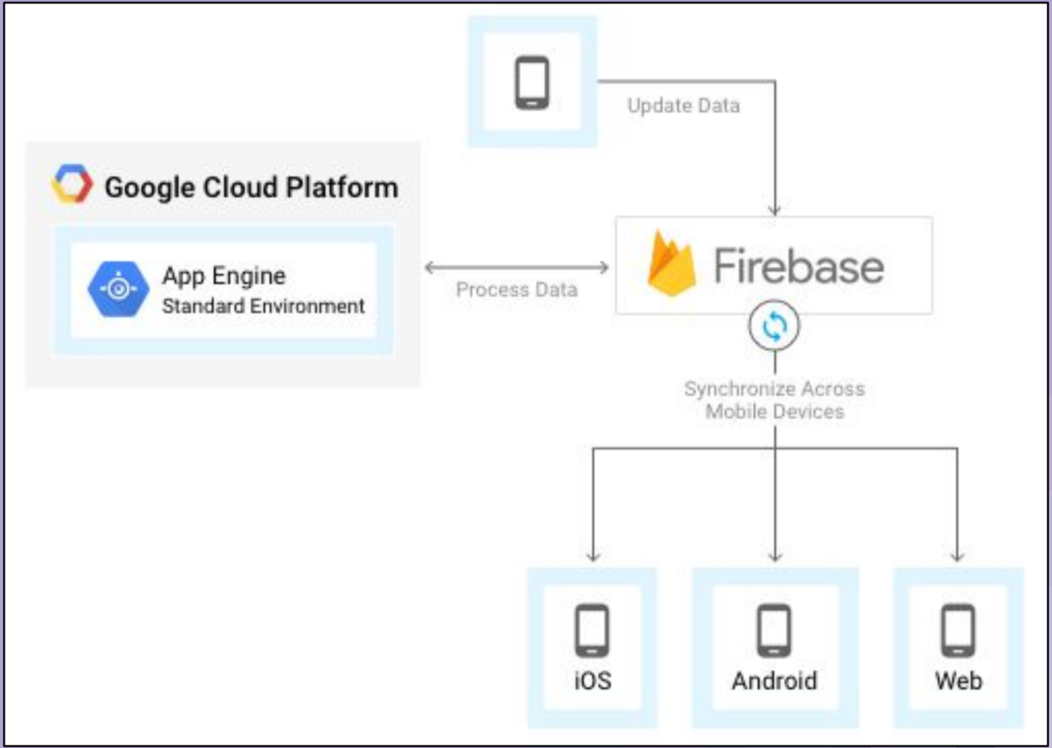
- Map using Google Maps API
- QR Scanner
- Scalability
- Filtering system based on jobs
- Usability

Functional Requirements

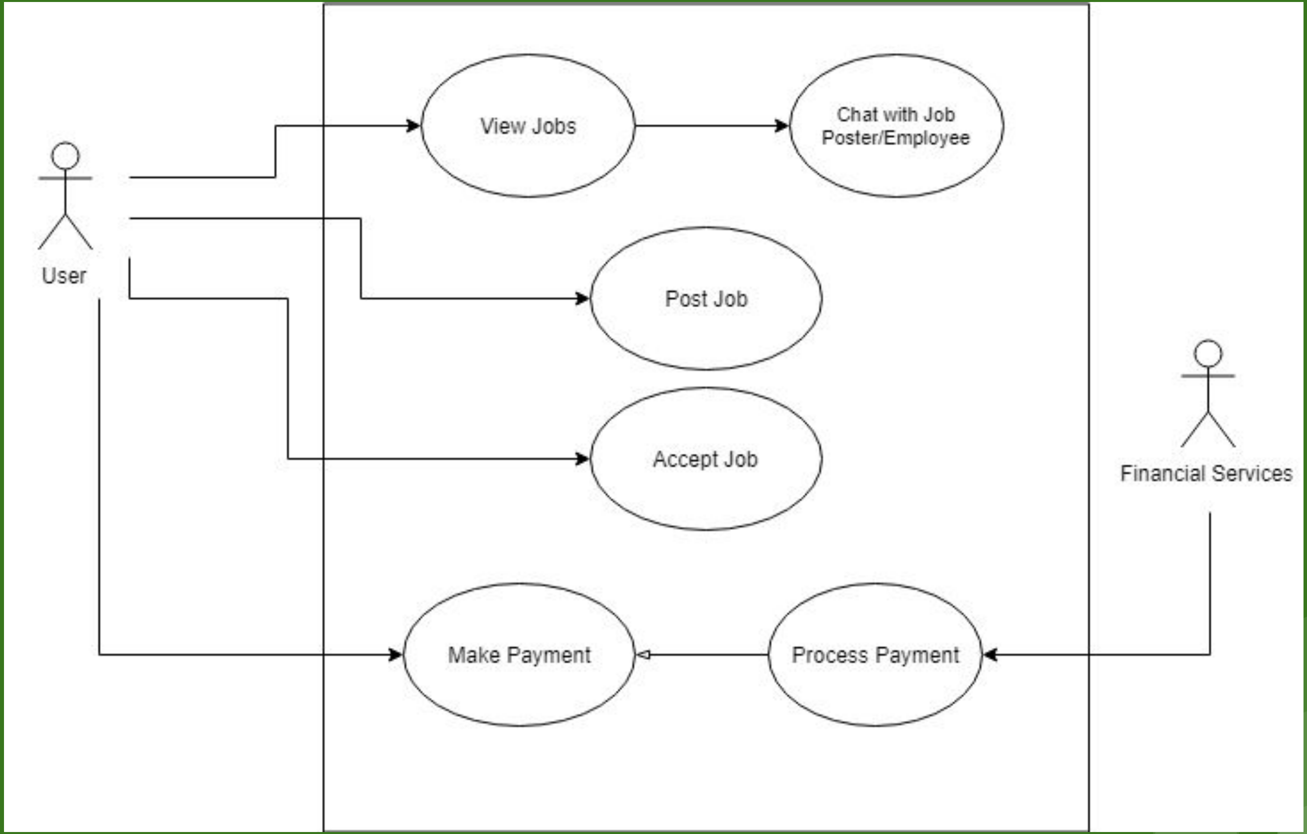
- **Focus:** Find or perform small side jobs easily.
- **Reliable and Fair:** App will have features to ensure job completion.
- **Outsourcing:** Have external API’s deal with financial operations.
- **Server:**
  - Must account for delays.
  - Schedule tasks with appropriate priority.

App Engine + Firebase

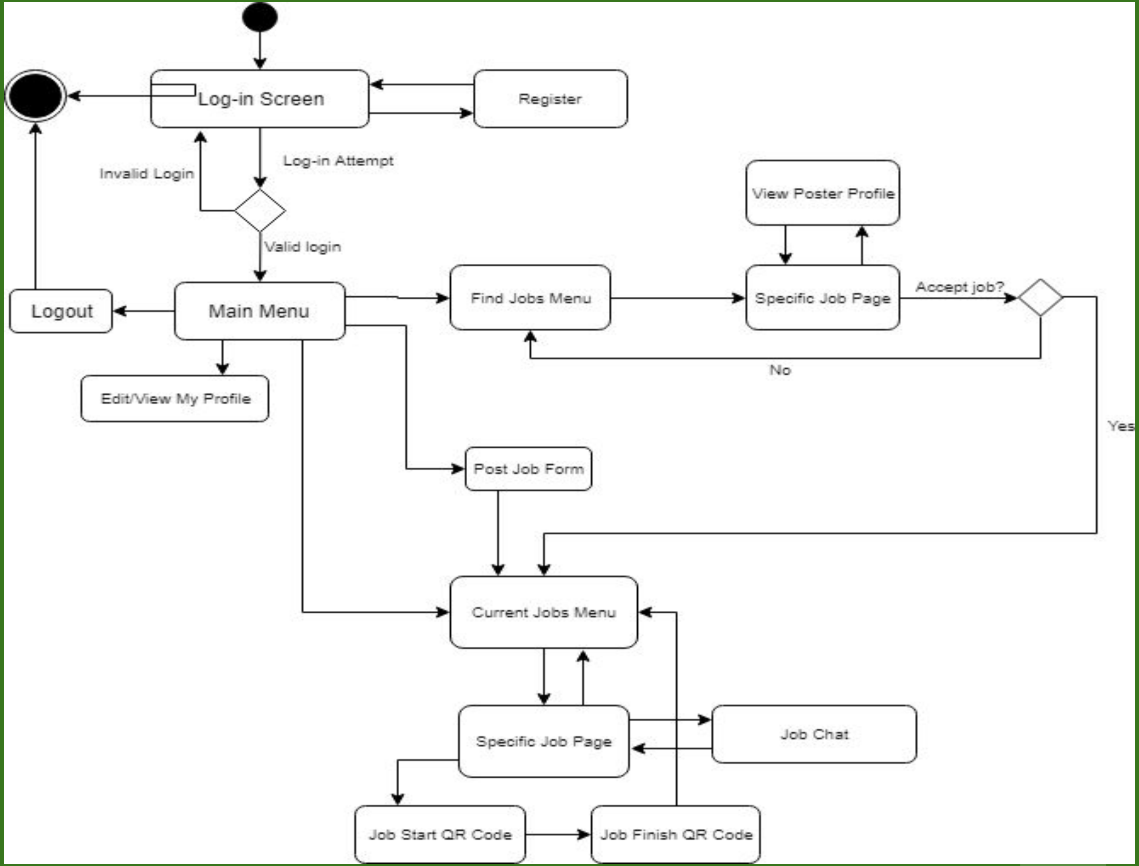
- Firebase will be the database where all data will be stored.
- App Engine will be listening to Firebase data and process accordingly.
- App Engine will be using Node.js.



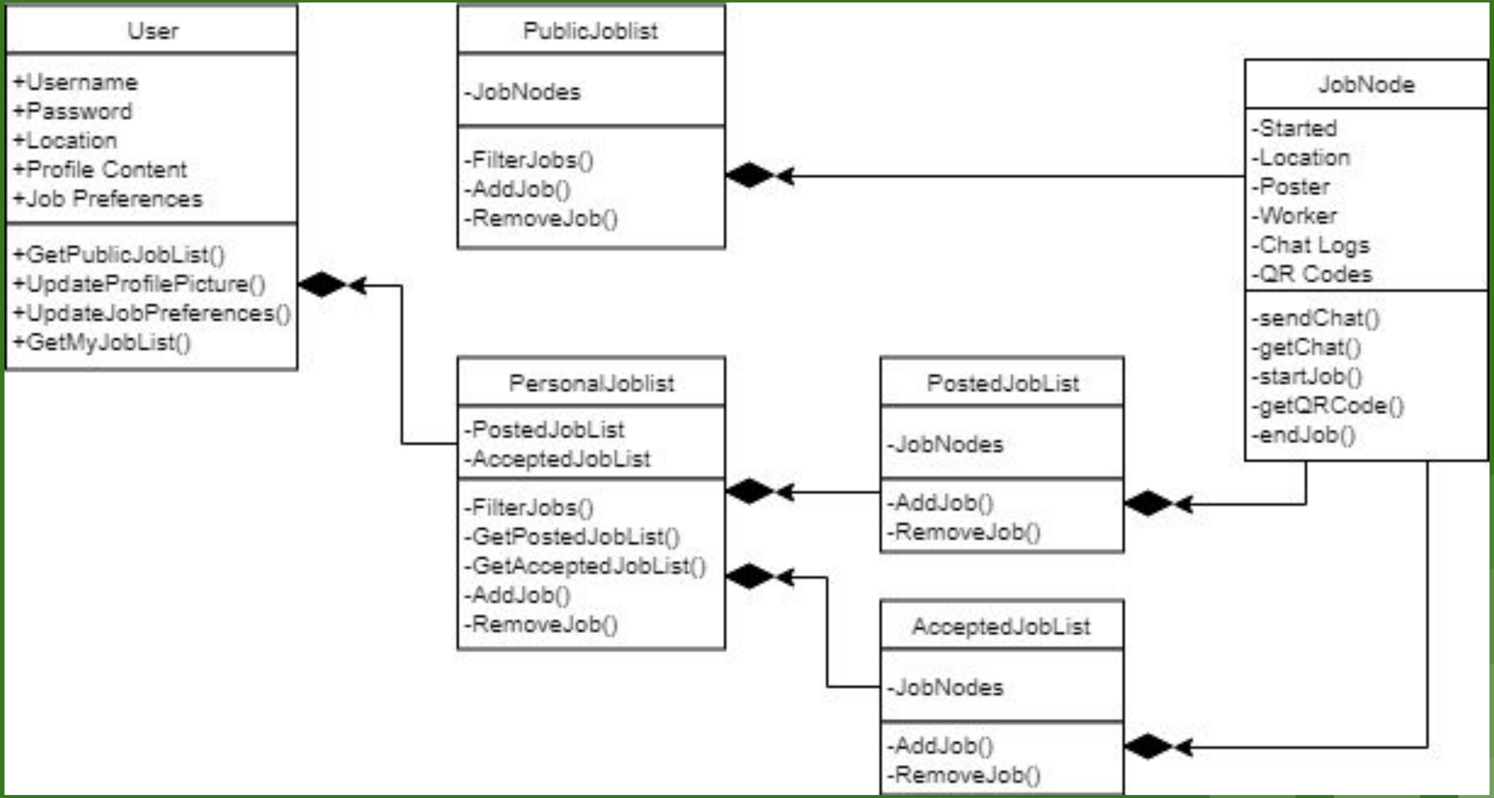
Use Cases


User
View Jobs
Chat with Job Poster/Employee
Post Job
Accept Job
Make Payment
Process Payment
Financial Services

User Activities



Class Diagram



Risks

- The back-end can get messy if not structured properly.
- Financial transactions need to be carefully and securely implemented.
- Direct interactions between users cannot be controlled. Cannot guarantee safety.

Possible Future Features

Assignment 1

- Bid Style Interactions

Assignment 2

- Social media integration

Assignment 3

- Recommended gygs

03.04.18

- Begin setting up Front End and Back-End skeleton.

03.20.18

FE: Design Layouts, Navigation

BE: Design a RESTful connection protocol.

03.30.18

- Begin to connect the Front end with the backend.
- Optimize synchronization.

04.15.18

- Refine the UI and performance as best as possible.
- Extensive testing

04.30.18

- Delivery