**SC2006 Software Engineering**

****

**Lab#1 Deliverables**

Lab Group A29

Group Name: Mala

Lab Supervisor: Zhang Chongzhi

| Group Members | |
| --- | --- |
| Name | Matriculation No. |
| Pavanraj Selvaraju | U2122616H |
| Ng Yao Ming | U2122168F |
| Low Yong Keng | U2122645L |
| Teo Kai Jie Keith | U2121074G |
| Oak Soe Khant | U2121901K |

**Table of Contents**

[**1.0 Project Introduction**](#_heading=h.gjdgxs) **4**

[1.1 Purpose](#_heading=h.30j0zll) 4

[1.2 Intended Audience](#_heading=h.1fob9te) 4

[**2.0 Documentation of Functional and Non-functional requirements**](#_heading=h.3znysh7) **5**

[2.1 Non-Functional Requirements](#_heading=h.jnmag6txxq8w) 5

[2.2 Non-Functional Requirements](#_heading=h.4d34og8) 8

[**3.0 Data Dictionary**](#_heading=h.2s8eyo1) **9**

[**4.0 Initial Use Case Model**](#_heading=h.3rdcrjn) **10**

[4.1 Use Case Diagram](#_heading=h.26in1rg) 10

[4.2 Use Case Descriptions](#_heading=h.lnxbz9) 11

[4.2.1 Use Case 1](#_heading=h.u9opbrr5wg6p) 11

[4.2.2 Use Case 2](#_heading=h.vl40c2o3byjj) 13

[4.2.3 Use Case 3](#_heading=h.sdc5a91bwhyd) 14

[4.2.4 Use Case 4](#_heading=h.k6d0luuc3qs2) 15

[4.2.5 Use Case 5](#_heading=h.oq8wlblivt4x) 16

[4.2.6 Use Case 6](#_heading=h.hh1q362jni6g) 17

[4.2.7 Use Case 7](#_heading=h.y6ee8vr7h2ye) 18

[4.2.8 Use Case 8](#_heading=h.o1rphbl993er) 19

[4.2.9 Use Case 9](#_heading=h.wsypy65100t5) 20

[4.2.10 Use Case 10](#_heading=h.y6v588sde2in) 21

[4.2.11 Use Case 11](#_heading=h.fpjsctywsmci) 22

[4.2.12 Use Case 12](#_heading=h.hmh8mr3wmad1) 23

[4.2.13 Use Case 13](#_heading=h.98gpbby3rhge) 24

[4.2.14 Use Case 14](#_heading=h.egfxeh4mzus5) 25

[4.2.15 Use Case 15](#_heading=h.yzaoxfho7h89) 26

[**5.0 UI Mock-Ups**](#_heading=h.t5pf51wovmp) **27**

[5.1 Home Page (not logged in)](#_heading=h.1ksv4uv) 27

[5.1.1 Login Page](#_heading=h.6t2n3e2db9hk) 28

[5.1.2 Registration Page](#_heading=h.2jxsxqh) 28

[5.2 Dashboard Page (after logged in)](#_heading=h.z337ya) 29

[5.3 Tracker Page (main page)](#_heading=h.3j2qqm3) 29

[5.3.1 Tracker Page (show more details on application)](#_heading=h.1y810tw) 30

[5.3.2 Tracker Page (job details)](#_heading=h.4i7ojhp) 30

[5.4 Searcher Page (initial set up)](#_heading=h.2xcytpi) 31

[5.4.1 Searcher Page (collapsed card)](#_heading=h.1ci93xb) 31

[5.4.2 Searcher Page (expanded card)](#_heading=h.wtqa744j9xoa) 32

[5.5 Forum Page](#_heading=h.wywsdiq0uvz3) 32

[5.6 Settings Page (edit profile)](#_heading=h.3as4poj) 33

# 1.0 Project Introduction

## 1.1 Purpose

Searching for and tracking job applications is tedious, and one needs to be meticulous in order to not get overwhelmed. Our group, Mala, aims to create a job platform that helps job seekers find job listings and keep track of job applications, all in one place. The Job Searcher shows a list of jobs that best match the user in a "dating app" manner, allowing the user to quickly and effectively browse and select those that interest them. The Job Tracker assists them with organising all their job applications and also improves resumes based on the job listing, streamlining the process of finding a job.

## 1.2 Intended Audience

Our intended audience for the application is anyone who is seeking an all-in-one platform to organise and source for jobs so as to increase their likelihood of finding one that they are pleased with and at the same time reduce the trouble they would have to go through in the process.

# 2.0 Documentation of Functional and Non-functional requirements

## 2.1 Functional Requirements

1. Homepage

1.1. Users must be able to access the login and register page from the home page.

2. Account

2.1. Users must be able to register for a new account at the register page.

2.1.1. System must create a new account for the user once all requirements are fulfilled.

2.1.1.1. System must prompt the user to re-enter the username if it is taken.

2.1.1.2. System must prompt the user to re-enter the password if it didn’t adhere to the requirements: at least 8 characters, password must be a combination of letters and numbers, 2 passwords do not match.

2.1.1.3. System must prompt the user to re-enter the email address if it is already linked to an existing account.

2.1.2. System must set a default display picture and bio for the user when the account is created successfully.

2.1.3. System must display a message saying the account is created when users successfully register for a new account.

2.2. Users must be able to login to the account with username and password at the login page.

2.2.1. System must validate the user's account when users enter their username and password.

2.2.2. System must redirect users to the dashboard after their account is validated.

2.3. Users must be able to view their profile at the profile page.

2.4. Users must be able to edit their account information at the edit profile page.

2.4.1. Any changes made by users must fulfil the requirements according to (2.1.1.).

2.4.2. System must be able to update the user's email when the user changes their email.

2.4.3. System must be able to update the user's password when the user changes their password.

2.4.4. System must be able to update the user's biography when the user changes their biography.

2.4.5. System must be able to update the user's display picture when the user changes their display picture.

2.5. Users must be able to reset the password if they had forgotten their passwords.

2.5.1. System must send an email to users to reset the password.

2.5.2. Systems must prompt the user to re-enter the password if it did not adhere to the requirements: at least 8 characters, password must be a combination of letters and numbers.

3. Dashboard

3.1. Users must be able to access the profile page, job tracker page, job search page and forum page from the dashboard.

4. Forum

4.1. Users must be able to access the dashboard, job tracker, job search and profile page from the forum page.

4.2. Users must be able to create forum threads at the forum page.

4.2.1. System must create forum threads for users when requirements are fulfilled.

4.2.1.1. Forum threads must not exceed 10000 words.

4.3. Users must be able to edit their forum threads.

4.3.1. Any edits must adhere to the requirements in (4.2.1.).

4.4. Users must be able to delete their forum threads.

4.4.1. System must delete forum threads when users delete it.

4.5. Users must be able to post on forum threads.

4.5.1. System must create posts for users when requirements are fulfilled.

4.5.1.1. Posts must not exceed 1000 words.

4.6. Users must be able to edit their posts on forum threads.

4.6.1. Any edits must adhere to the requirements in (4.4.1.).

4.7. Users must be able to delete their posts on forum threads.

4.7.1. System must delete posts when users delete it.

5. Job Search

5.1. Users must be able to access the profile page, job tracker page, dashboard and forum page from the job search page.

5.2. System must survey users the first time they access this page and hold the information.

5.2.1. Users must be able to update information.

5.3. System must show jobs suitable to users based on the information held from the survey.

5.4. Users must be able to browse through jobs at the job search page.

5.4.1. Job information must be displayed.

5.4.1.1. Job name must be displayed.

5.4.1.2. Job description must be displayed.

5.4.1.3. Job location must be displayed.

5.4.1.4. Job working hours must be displayed.

5.4.1.5. Job salary range must be displayed.

5.4.1.6. Job required skills must be displayed.

5.5. Users must be able to add jobs they are interested in to a wish-list in the job tracker page.

6. Job Tracker

6.1. Users must be able to access the profile page, job search page, dashboard and forum page from the job tracker page.

6.2. Job tracker page must have a page containing a list of tracked applications and a separate page containing a list of jobs wish-listed from the job search page.

6.3. Users must be able to create new job entries.

6.4. Users must be able to create an entry for a job application based on jobs wish-listed from the job search page.

6.5. Users must be able to update their job application entries.

6.6. Users must be able to delete their job application entries.

6.7. Users must be able to view their job application entries.

6.8. System must suggest edits to the user's resume if requested by the user.

## 2.2 Non-Functional Requirements

1. Usability
   1. The system should include help documentation and tutorials that can help users understand how to navigate the pages.
   2. Users can access every other page from any page.
2. Reliability

2.1. After a system reboot, the full system functionality must be restored within 5

minutes.

1. Performance
   1. The application entries shall load in within 2 seconds.
   2. The sign up and sign in page shall load in within 1 second.
   3. The updating of password shall take at most 1 second to update in the database.
   4. The email for the resetting of password when users forgot their password shall arrive within 1 min.
   5. Each function module shall load in within 1 second.
   6. System must add the new job application entry to the tracker listing within 1 minute.
   7. Details of the job shall load within 2 seconds.
   8. The application shall exit within 2 seconds.
   9. All reminding messages shall load in within 0.5 second.
2. Supportability

4.1. The database must be replaceable with any commercial product supporting

standard SQL queries.

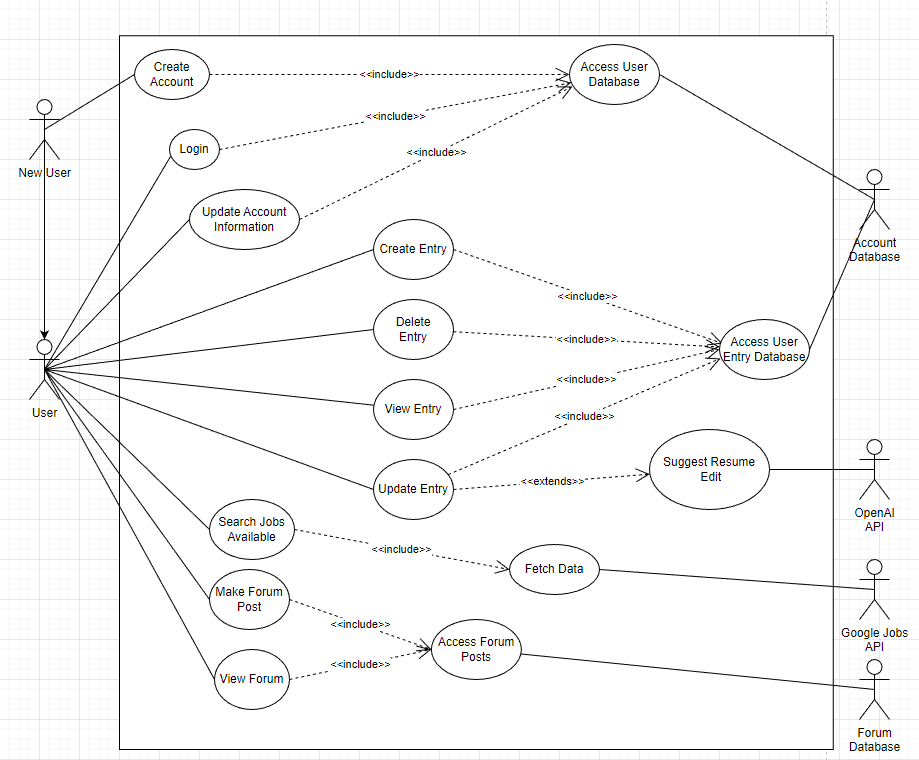
1. Safety Requirements
   1. The team shall maintain the system regularly.
   2. The team shall test and debug the application beforehand.
2. Security Requirements
   1. If the user enters the wrong password more than 3 times consecutively, the account will be locked.
   2. Users must log in their account to access all the other pages.
   3. The system shall handle exceptions.
      1. Reminding message shall display when the user enters two different passwords when setting or resetting the password.
      2. Reminding message shall display when the user inputs the wrong verification code.
   4. The application must not leak user information.
      1. User information must not be seen or used by others.
      2. User information must be deleted if the user account is deleted.

# 3.0 Data Dictionary

| Term | Definition |
| --- | --- |
| User | A person who has a registered account and is using the services provided by the application. |
| Account | The profile of the user in our system which contains details (Name, Username, Email) associated with them. |
| Username | A unique name that is associated with the user. |
| Password | A string of characters (letters, numbers, special symbols) that allow the user to access their account |
| Dashboard | The default page when users are logged in. Access to the profile, job tracker, job search and forum page can be made here. |
| Job | An instance of a job from the Google Job Search API with details like the job’s name, description, salary range, location, skills. Users can add jobs from the job search page to the wish-list in the job tracker page. |
| Job Entry | An instance of a job entry of the user with details like the job’s name, description, salary range, location, skills. |
| Job Searcher | The page where users can browse through different jobs recommended to them based on the information gathered by the system from the survey. |
| Survey | A questionnaire to the users to indicate their interests and skills so that the recommendations from the job search page can be streamlined to suit their needs. |
| Match | A match is the case where a job is compatible with a user based on the surveyed information. |
| Job Tracker | A page where users can find a list of all the jobs they applied to and their current statuses. |
| Wish-List | A list containing the jobs added by the users from the job search page which they are interested in. |
| Forum | A community where users can interact with one another and share their thoughts regarding job applications. |
| Post | A post is either the starting post of a thread or replies to it. |
| Thread | A thread is a series of posts about a single topic. A user can make a new thread and users can reply to it. |
| Suggest | Generate a resume for the users based on the resume the user inputted with edits by ChatGPT using the OpenAI API. |

# 4.0 Initial Use Case Model

## 4.1 Use Case Diagram



## 

## 

## 4.2 Use Case Descriptions

## 4.2.1 Use Case 1

| Use Case ID: | **UC\_NewUser\_1** | | |
| --- | --- | --- | --- |
| Use Case Name: | **Create Account** | | |
| Created By: | **Yong Keng** | Last Updated By: | **Yong Keng** |
| Date Created: | **27.01.2023** | Date Last Updated: | **27.01.2023** |

| Actor: | **New User** |
| --- | --- |
| Description: | **Create Account allows new users to create an account in our system database. To create an account, users must input Name, Username, Email, Password.** |
| Preconditions: | 1. **Username must not be taken.** 2. **Passwords must meet the requirements:**     1. **At least 1 special character**    2. **At least 1 capital character**    3. **At least 8 characters.** 3. **Email must be unique.** |
| Postconditions: | 1. **System sends a confirmation link to the user's email account.** 2. **System redirects the user to the main page.** |
| Priority: |  |
| Frequency of Use: | **Low** |
| Flow of Events: | 1. **User inputs required information: Name, Username, Email and Password.** 2. **System validates the information.** 3. **System creates the account.** 4. **System updates account information in database.** 5. **System directs the user to the main page.** |
| Alternative Flows: | **User account already exists: Email taken.**   1. **System displays “User account already exists”** 2. **System prompts users to create an account or log in.** 3. **If an account is created, then continue from main flow step 1.** 4. **If log in, then system directs to log in page to continue from Login (USER\_UC\_2)**   **Username already taken**   1. **System displays “Username already taken”** 2. **System prompts users to input a different username.** 3. **Continue from main flow step 1.**   **Passwords do not meet requirements.**   1. **System displays “Please input a password of at least 1 special character, mixed case and at least 8 characters long” message.** 2. **Continue from main flow step 1.** |
| Exceptions: |  |
| Includes: | **Access User Database (UC\_ACCDB\_1)** |
| Extends: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

## 4.2.2 Use Case 2

| Use Case ID: | **UC\_USER\_1** | | |
| --- | --- | --- | --- |
| Use Case Name: | **Login** | | |
| Created By: | **Yong Keng** | Last Updated By: | **Yong Keng** |
| Date Created: | **27.01.2023** | Date Last Updated: | **27.01.2023** |

| Actor: | **User** |
| --- | --- |
| Description: | **Login allows users to enter the app to use features catered for registered users of the app. To login to the system, the user must input an existing account details which consists of a username and password.** |
| Preconditions: | **User must have an existing account in the entry database** |
| Postconditions: | **System directs user to main page** |
| Priority: |  |
| Frequency of Use: | **High** |
| Flow of Events: | 1. **User enters username and password on login form.** 2. **User submits details.** 3. **System validates the user’s username and password.** 4. **System shows the main page.** |
| Alternative Flows: | **Invalid Username**   1. **System cannot find the username in the database.** 2. **System displays an error message.** 3. **System prompts user to input username and password.** 4. **Use case resumes at main flow step 1.**   **Invalid Password**   1. **Password does not match for unique username.** 2. **System displays an error message.** 3. **System prompts user to input username and password.** 4. **Use case resumes at main flow step 1.**   **Missing Username/ Password**   1. **System displays an error message.** 2. **System prompts user to input username and password.** 3. **Use case resumes at main flow step 1.** |
| Exceptions: |  |
| Includes: | **Access User Database (UC\_ACCDB\_1)** |
| Extends: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

## 4.2.3 Use Case 3

| Use Case ID: | **UC\_USER\_2** | | |
| --- | --- | --- | --- |
| Use Case Name: | **Update Account Information** | | |
| Created By: | **Yong Keng** | Last Updated By: | **Yong Keng** |
| Date Created: | **27.01.2023** | Date Last Updated: | **27.01.2023** |

| Actor: | **User** |
| --- | --- |
| Description: | **Update Account Information allows users to edit account information such as username, name, profile picture, password.** |
| Preconditions: | **User must login first.** |
| Postconditions: | **User will have updated their account information.** |
| Priority: |  |
| Frequency of Use: | **Low** |
| Flow of Events: | 1. **User selects the field to update.** 2. **User inputs updated details.** 3. **System updates information in the database.** |
| Alternative Flows: | **User selects to change passwords.**   1. **If the user selects the password field to update, the user must input the old password and new password.** |
| Exceptions: |  |
| Includes: | **Access User Database (UC\_ACCDB\_1)** |
| Extends: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

## 4.2.4 Use Case 4

| Use Case ID: | **UC\_USER\_3** | | |
| --- | --- | --- | --- |
| Use Case Name: | **Create Entry** | | |
| Created By: | **Yong Keng** | Last Updated By: | **Yong Keng** |
| Date Created: | **27.01.2023** | Date Last Updated: | **27.01.2023** |

| Actor: | **User** |
| --- | --- |
| Description: | **In order for users to look for desired jobs, they have to first create an entry indicating their fields of interests so that the system can run its algorithm and show the user the lists of jobs available for them.** |
| Preconditions: | 1. **Users must be logged in.** |
| Postconditions: |  |
| Priority: |  |
| Frequency of Use: | **Moderate** |
| Flow of Events: | 1. **User chooses to create an entry.** 2. **System prompts users to select the fields of interest, salary range, location, specific company (optional).** |
| Alternative Flows: | **Fields not fully filled up.**   1. **System prompts users to fill up the remaining fields before proceeding.** 2. **User details will be updated once all fields are filled.** |
| Exceptions: |  |
| Includes: | **Access User Entry Database (UC\_ACCDB\_2)** |
| Extends: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

## 4.2.5 Use Case 5

| Use Case ID: | **UC\_USER\_4** | | |
| --- | --- | --- | --- |
| Use Case Name: | **Delete Entry** | | |
| Created By: | **Yong Keng** | Last Updated By: | **Yong Keng** |
| Date Created: | **27.01.2023** | Date Last Updated: | **27.01.2023** |

| Actor: | **User** |
| --- | --- |
| Description: | **Delete Entry allows users to remove an existing application that the user created previously.** |
| Preconditions: | 1. **Entry must exist before allowing users to delete** |
| Postconditions: |  |
| Priority: |  |
| Frequency of Use: | **Moderate** |
| Flow of Events: | 1. **User selects Delete Entry and chooses the desired application to be removed.** 2. **System removes previously created entries that the user chose.** |
| Alternative Flows: | **Entry does not exist.**   1. **Delete button to be greyed out to prevent users from using.** |
| Exceptions: |  |
| Includes: | **Access User Entry Database (UC\_ACCDB\_2)** |
| Extends: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

## 4.2.6 Use Case 6

| Use Case ID: | **UC\_USER\_5** | | |
| --- | --- | --- | --- |
| Use Case Name: | **View Entry** | | |
| Created By: | **Yong Keng** | Last Updated By: | **Yong Keng** |
| Date Created: | **27.01.2023** | Date Last Updated: | **27.01.2023** |

| Actor: | **User** |
| --- | --- |
| Description: | **Users can view all the entries that the user has created.** |
| Preconditions: | 1. **Existing entries created by the user** |
| Postconditions: |  |
| Priority: |  |
| Frequency of Use: | **High** |
| Flow of Events: | 1. **User selects View Entry.** 2. **System displays the list of entries that the user created.** |
| Alternative Flows: | **The user does not have any entry which will display an empty list.**   1. **Prompts users to create entries in order for the system to show jobs to users.** 2. **Else displays an empty list.** |
| Exceptions: |  |
| Includes: | **Access User Entry Database (UC\_ACCDB\_2)** |
| Extends: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

## 4.2.7 Use Case 7

| Use Case ID: | **UC\_USER\_6** | | |
| --- | --- | --- | --- |
| Use Case Name: | **Update Entry** | | |
| Created By: | **Yong Keng** | Last Updated By: | **Yong Keng** |
| Date Created: | **27.01.2023** | Date Last Updated: | **27.01.2023** |

| Actor: | **User** |
| --- | --- |
| Description: | **Updates user’s previously created entries if the user did not input the correct details.** |
| Preconditions: | 1. **Must have at least 1 entry created by the user.** |
| Postconditions: |  |
| Priority: |  |
| Frequency of Use: | **Moderate** |
| Flow of Events: | 1. **User selects the Update Entry and chooses the fields to edit.** 2. **System checks and updates the selected entry.** |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: | **Access User Entry Database (UC\_ACCDB\_2)** |
| Extends: | **Suggest Resume Edit (UC\_JOBSAPI\_1)** |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

## 4.2.8 Use Case 8

| Use Case ID: | **UC\_USER\_7** | | |
| --- | --- | --- | --- |
| Use Case Name: | **Search Jobs Available** | | |
| Created By: | **Keith** | Last Updated By: | **Keith** |
| Date Created: | **27.01.2023** | Date Last Updated: | **27.01.2023** |

| Actor: | **User** |
| --- | --- |
| Description: | **Allows users to search for jobs based on their survey results. If the user is not logged in, search is done by keywords in search only.** |
| Preconditions: | 1. **User must be logged in to use personalised search results** |
| Postconditions: |  |
| Priority: |  |
| Frequency of Use: | **1** |
| Flow of Events: | 1. **User enters search terms into the search bar and presses the search button.** 2. **Results using search terms and survey results are returned.** |
| Alternative Flows: | **User is not logged in.**   1. **User enters search terms into the search bar.** 2. **Results using only search terms are returned.** |
| Exceptions: |  |
| Includes: | **Fetch Data (UC\_JOBSAPI\_1)** |
| Extends: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

## 4.2.9 Use Case 9

| Use Case ID: | **UC\_USER\_8** | | |
| --- | --- | --- | --- |
| Use Case Name: | **Make Forum Post** | | |
| Created By: | **Keith** | Last Updated By: | **Keith** |
| Date Created: | **27.01.2023** | Date Last Updated: | **27.01.2023** |

| Actor: | **User** |
| --- | --- |
| Description: | **Allows users to make a post on the forum.** |
| Preconditions: | **Users must be logged in to make a post.** |
| Postconditions: |  |
| Priority: |  |
| Frequency of Use: | **Moderate** |
| Flow of Events: | 1. **User navigates to the forum tab.** 2. **User clicks on the “New Post” button and enters the post contents, then presses the “Post” button .** |
| Alternative Flows: | **User is not logged in.**   1. **The make post button is greyed out.** |
| Exceptions: |  |
| Includes: | **Access Forum Posts (UC\_FORUMDB\_1)** |
| Extends: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

## 4.2.10 Use Case 10

| Use Case ID: | **UC\_USER\_9** | | |
| --- | --- | --- | --- |
| Use Case Name: | **View Forum** | | |
| Created By: | **Keith** | Last Updated By: | **Keith** |
| Date Created: | **27.01.2023** | Date Last Updated: | **27.01.2023** |

| Actor: | **User** |
| --- | --- |
| Description: | **Allows users to view posts on the forum.** |
| Preconditions: |  |
| Postconditions: |  |
| Priority: |  |
| Frequency of Use: | **Moderate** |
| Flow of Events: | 1. **User navigates to the forum tab to see the latest forum threads.** 2. **Clicks on individual threads to see the posts it is made out of.** |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: | **Access Forum Posts (UC\_FORUMDB\_1)** |
| Extends: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

## 4.2.11 Use Case 11

| Use Case ID: | **UC\_ACCDB\_1** | | |
| --- | --- | --- | --- |
| Use Case Name: | **Access User Database** | | |
| Created By: | **Keith** | Last Updated By: | **Keith** |
| Date Created: | **27.01.2023** | Date Last Updated: | **27.01.2023** |

| Actor: | **Account Database** |
| --- | --- |
| Description: | **API that allows access to user information for login, account creation and updating information.** |
| Preconditions: |  |
| Postconditions: |  |
| Priority: |  |
| Frequency of Use: | **High** |
| Flow of Events: | 1. **Frontend sends JSON request to account database for account information** 2. **Account database replies with relevant information** |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Extends: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

## 4.2.12 Use Case 12

| Use Case ID: | **UC\_ACCDB\_\_2** | | |
| --- | --- | --- | --- |
| Use Case Name: | **Access User Entry Database** | | |
| Created By: | **Keith** | Last Updated By: | **Keith** |
| Date Created: | **27.01.2023** | Date Last Updated: | **27.01.2023** |

| Actor: | **Account Database** |
| --- | --- |
| Description: | **API that allows access to user entries for job applications currently tracked.** |
| Preconditions: |  |
| Postconditions: |  |
| Priority: |  |
| Frequency of Use: | **High** |
| Flow of Events: | 1. **Frontend sends JSON requests to the account database for account entries information.** 2. **Account database replies with relevant information.** |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Extends: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

## 4.2.13 Use Case 13

| Use Case ID: | **UC\_JOBSAPI\_1** | | |
| --- | --- | --- | --- |
| Use Case Name: | **Suggest Resume Edit** | | |
| Created By: | **Keith** | Last Updated By: | **Keith** |
| Date Created: | **27.01.2023** | Date Last Updated: | **27.01.2023** |

| Actor: | **OpenAI API** |
| --- | --- |
| Description: | **OpenAI API takes in general text prompts and returns answers based on its NLP engine. This is used to suggest edits to the resume of the user.** |
| Preconditions: |  |
| Postconditions: |  |
| Priority: |  |
| Frequency of Use: | **Low** |
| Flow of Events: | 1. **Frontend sends API requests to OpenAI containing user resumes and prompts for improvements.** 2. **API responds with suggestions.** |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Extends: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

## 4.2.14 Use Case 14

| Use Case ID: | **UC\_JOBSAPI\_1** | | |
| --- | --- | --- | --- |
| Use Case Name: | **Fetch Data** | | |
| Created By: | **Keith** | Last Updated By: | **Keith** |
| Date Created: | **27.01.2023** | Date Last Updated: | **27.01.2023** |

| Actor: | **Google Jobs API** |
| --- | --- |
| Description: | **Fetches a list of jobs from API based on user job search query and/or survey** |
| Preconditions: |  |
| Postconditions: |  |
| Priority: |  |
| Frequency of Use: | **High** |
| Flow of Events: | 1. **Search Jobs Available use case sends API requests to Google Jobs API.** 2. **API responds with a list of results** |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Extends: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

## 4.2.15 Use Case 15

| Use Case ID: | **UC\_FORUMDB\_1** | | |
| --- | --- | --- | --- |
| Use Case Name: | **Access Forum Posts** | | |
| Created By: | **Keith** | Last Updated By: | **Keith** |
| Date Created: | **27.01.2023** | Date Last Updated: | **27.01.2023** |

| Actor: | **Forum Database** |
| --- | --- |
| Description: | **Retrieves past posts for viewing and supports creating of new posts.** |
| Preconditions: |  |
| Postconditions: |  |
| Priority: |  |
| Frequency of Use: | **High** |
| Flow of Events: | 1. **Frontend sends API requests to Forum Database containing action and relevant data.** 2. **API executes relevant action.** |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Extends: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

# 

# 

# 

# 

# 

# 5.0 UI Mock-Ups

## 5.1 Home Page (not logged in)



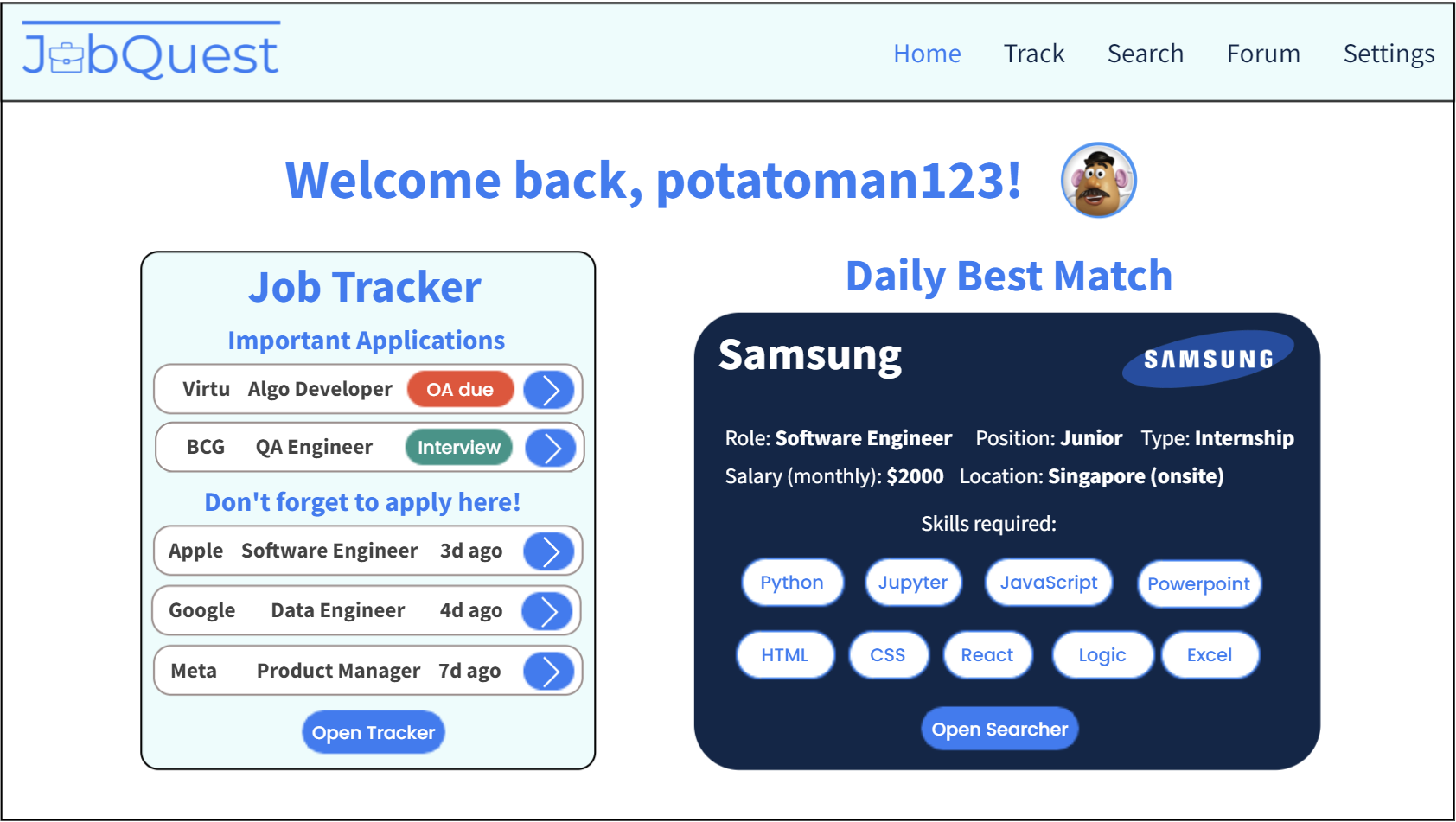
## 5.1.1 Login Page



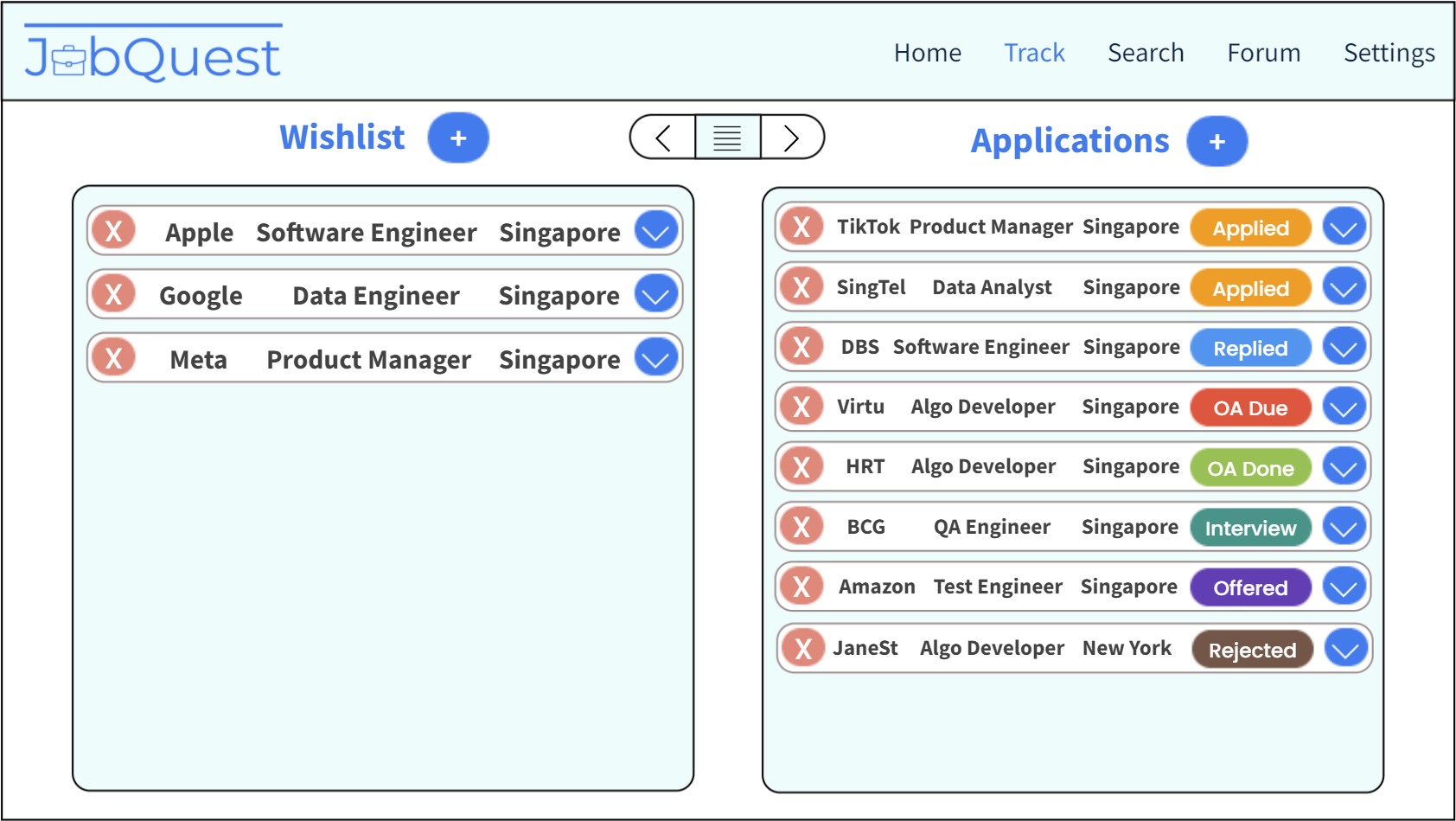
## 5.1.2 Registration Page



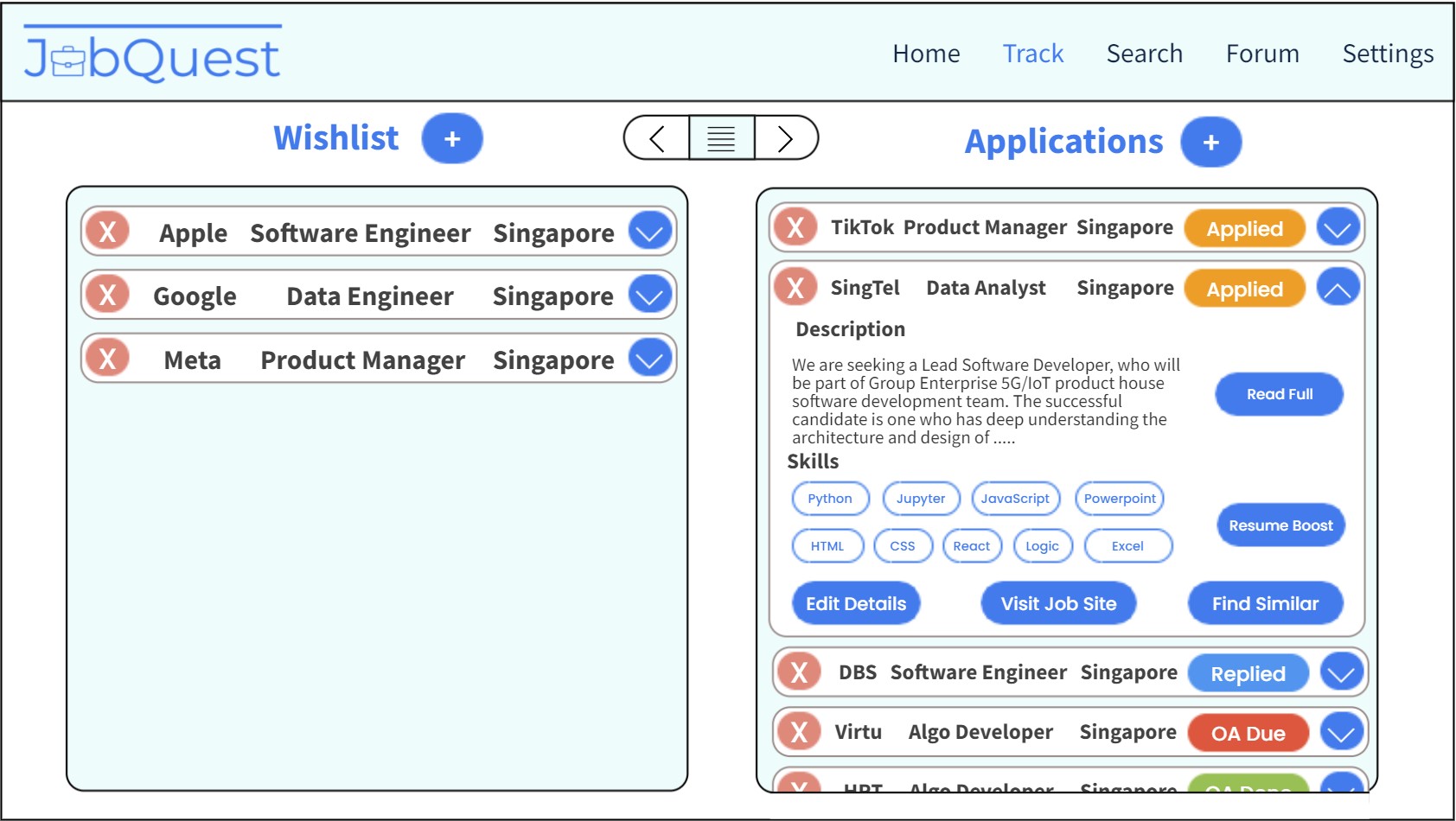
## 5.2 Dashboard Page (after logged in)



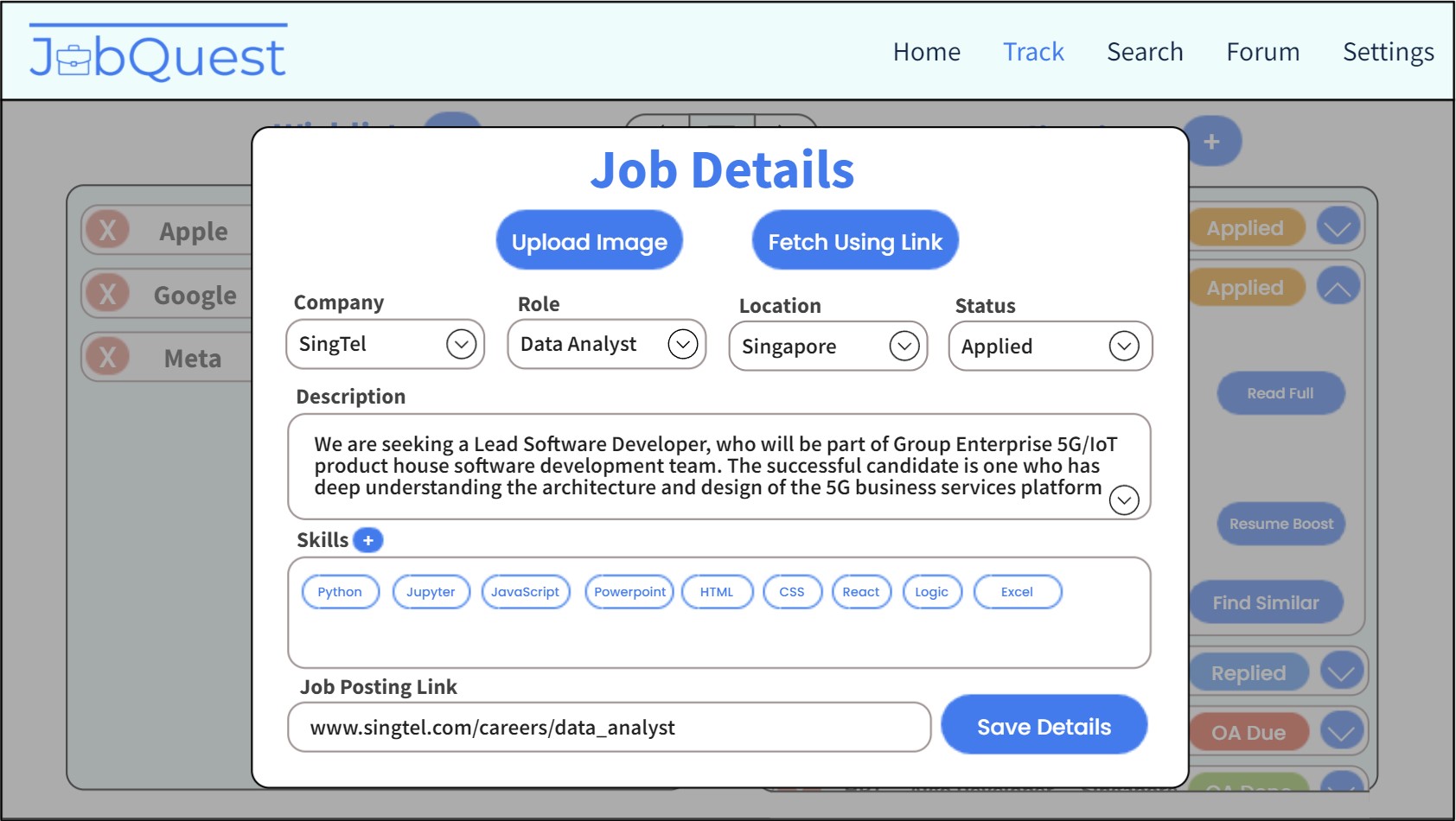
## 5.3 Tracker Page (main page)



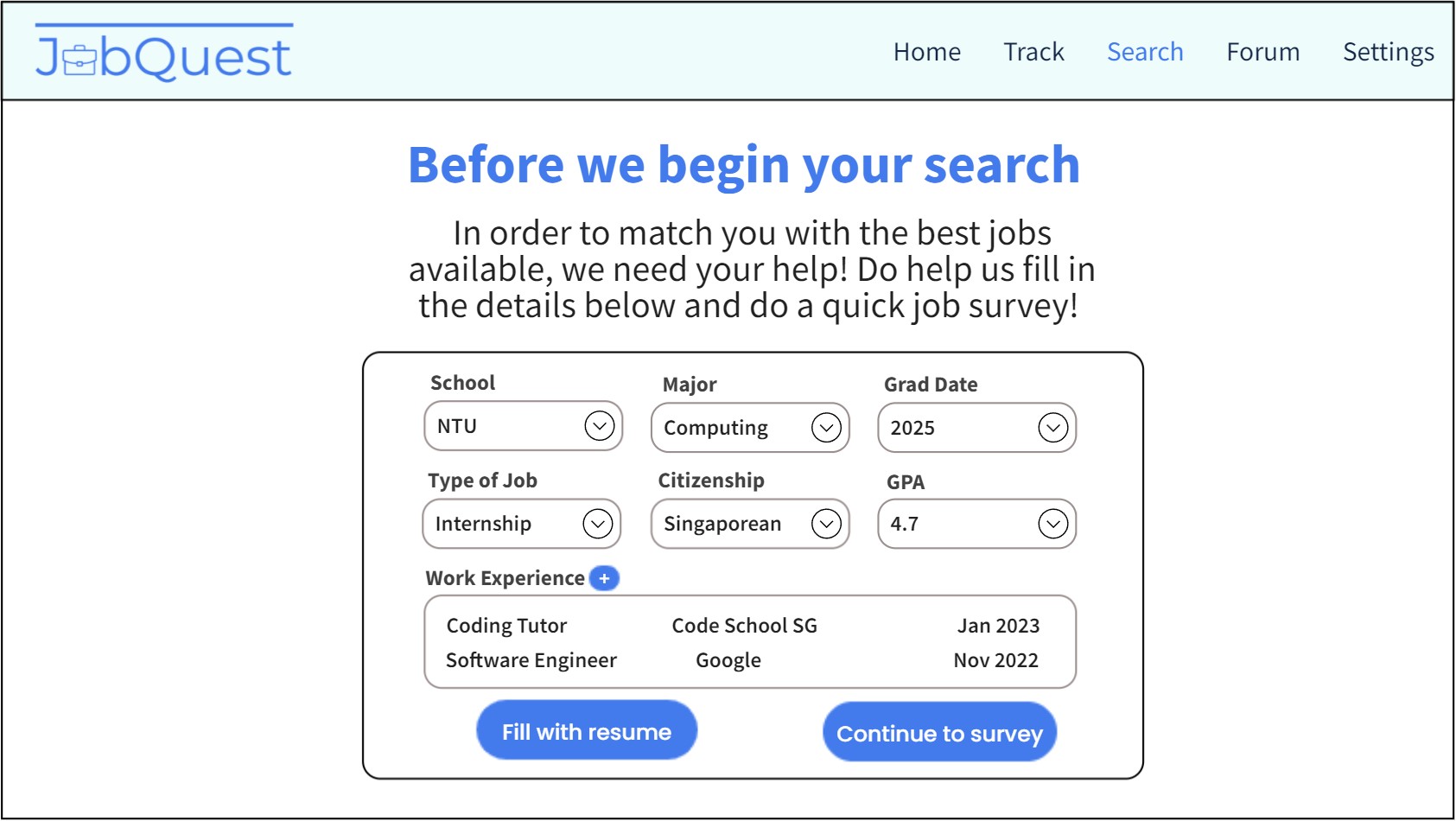
## 5.3.1 Tracker Page (show more details on application)



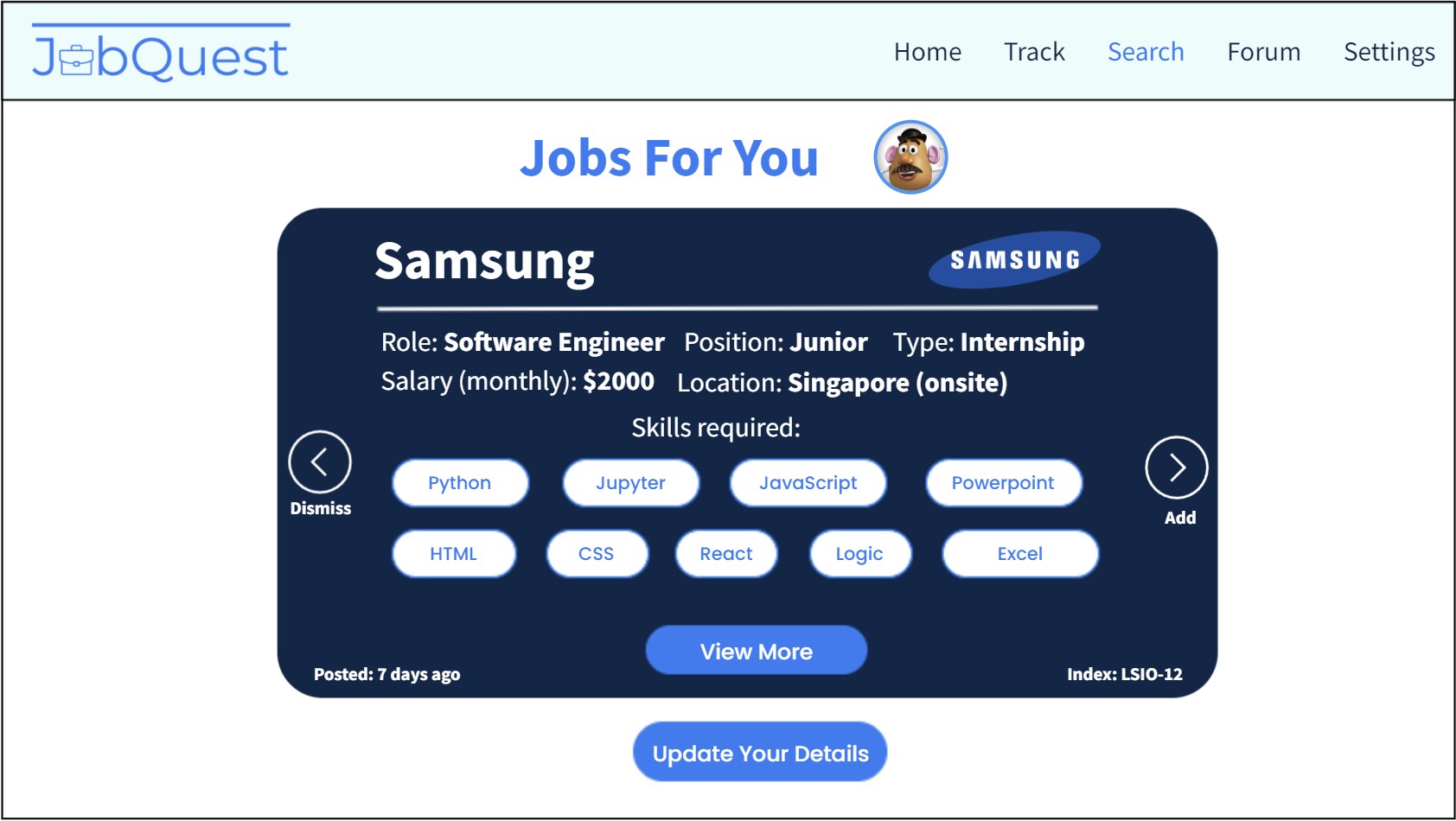
## 5.3.2 Tracker Page (job details)



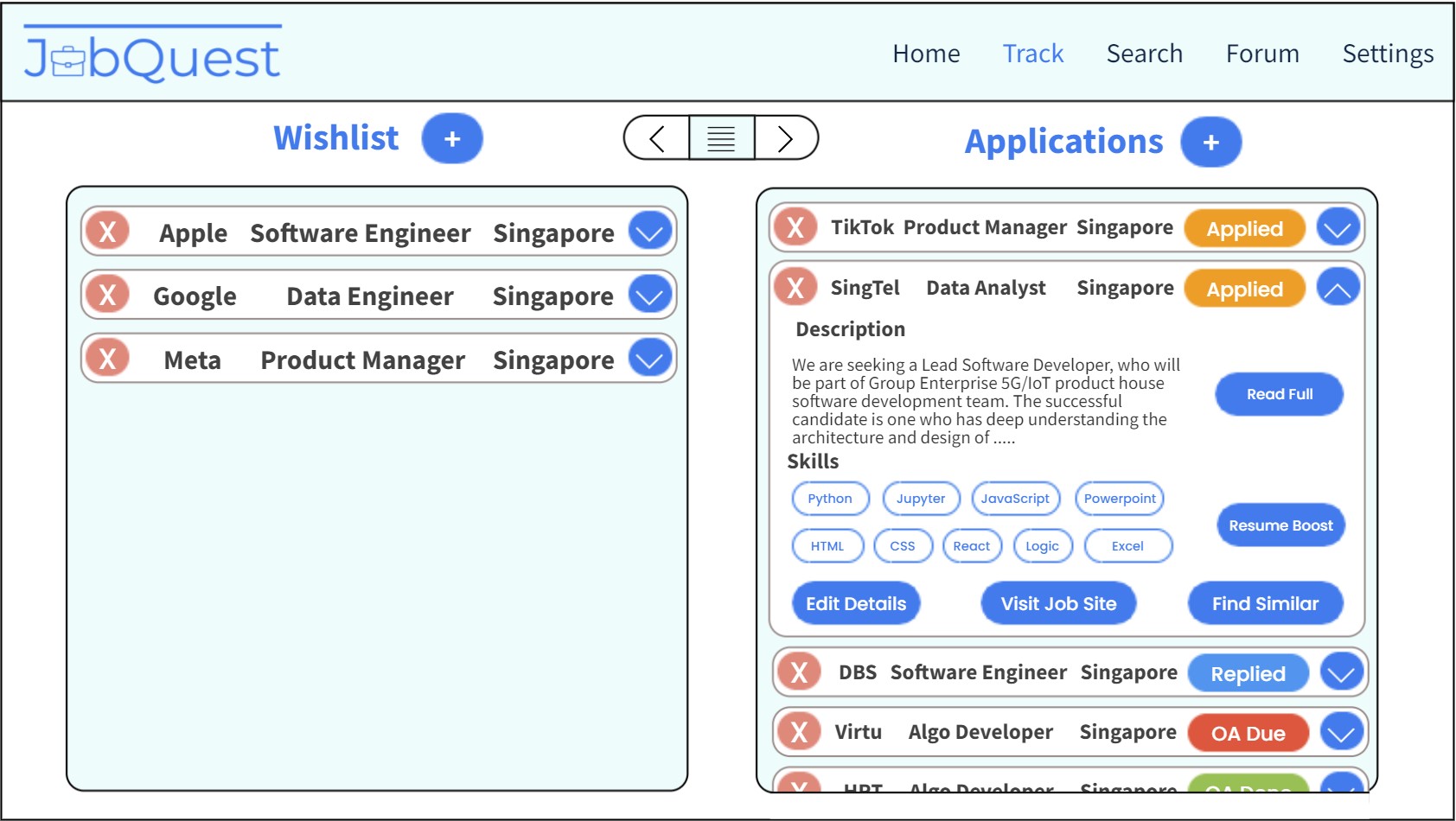
## 5.4 Searcher Page (initial set up)



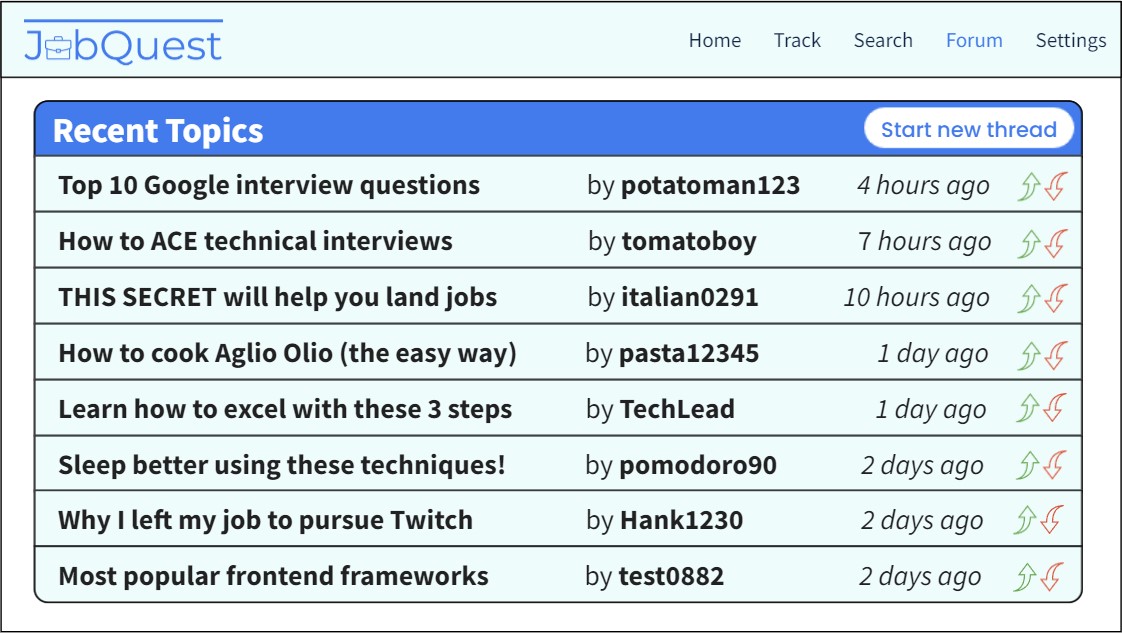
## 5.4.1 Searcher Page (collapsed card)



## 5.4.2 Searcher Page (expanded card)



## 5.5 Forum Page



## 5.6 Settings Page (edit profile)

