Software Requirement Specifications

CONNECT2WORK

Version: [00.01]

Project Code	*F210702*
Supervisor	*Dr. Muhammad Rafi*
Co Supervisor	*Mr. Kashif Noor*
Project Team	*Ishaq Ahmed - K180295* *Umer Alam - k180164* *Burhanuddin - k180213*
Submission Date	*7 / Dec / 2021*

Document History

[Revision history will be maintained to keep a track of changes done by anyone in the document.]

Version	Name of Person	Date	Description of change
01	*Umer*	*30-Nov-2021*	*Changes in interface*
02	*Ishaq*	*3-dec-2021*	*Changes In use case*

Distribution List

[Following table will contain list of people whom the document will be distributed after every sign-off]

Name	Role
Dr. Muhammad Rafi	*Supervisor*
Mr. Kashif Noor	*Co- Supervisor*

Document Sign-Off

[Following table will contain sign-off details of document. Once the document is prepared and revised, this should be signed-off by the sign-off authority.

Any subsequent changes in the document after the first sign-off should again get a formal sign-off by the authorities.]

Version	Sign-off Authority	Sign-off Date

Table of Contents

1. Introduction

1.1. Purpose of Document

The Software Requirement Specification (SRS) is a document that defines the specific nature, software, hardware, and business requirements that lead to the development of a software project, in this case, it is Connect2Work. The guide includes an overview of the thought process, the nature of development, the workflow, the major conceptualizing concepts and driving force, and a list of business satisfying points for which the product attempted to fill gaps.

1.2. Intended Audience

This document is intended for anyone interested in the design and analysis decisions made by the developers, as well as future engineers who want to understand the nature of the project and how it became what it became, with an emphasis on what concepts and features were implemented. This report will go over the product's lower-level details as well as the higher-level business use cases.

1.3 Abbreviations

None used

1.4 Document Convention

Font style: Arial
Font size: 16 for main headings, 14 for sub-headings, 12 for sub-sub-headings and 10 for paragraphs

2. Overall System Description

2.1. Project Background

In today's world, it's becoming difficult to find workers in small household works and also in construction sites, it's becoming difficult to hire workers as there they hire workers in bulk.

There are a lot of questions arising on the recruiter's side such as

- *Is the worker genuine?*
- *Is the worker charging a lot more money than others?*
- *Are there any workers who can come to a place to do work?*

Workers are not able to maintain their daily schedule when they've multiple work. (Lack of Management). Our Aim is to provide a common platform to the workers and recruiters which overcome these problems.

This platform help solves many problems as it helps provide jobs to the workers who are looking for jobs, and the recruiters who need not have to travel places or contact some mediator to find skilled workers. This platform can be used from anywhere and anytime according to our convenience, it is easy to understand to workers who have basic knowledge of technology.

2.2. Project Scope

Connect2Work is a freelancing type site which empowers blue-collar to come forward and take their place in society along with others. It also provides a lot of freedom and flexibility which regular jobs cannot, by giving them freedom to choose.

The system is divided into multiple sub modules. Users of the system are job poster (Recruiter) and job seeker (Worker). Input of the system are details of the Recruiter, Worker, Job. Outputs of the system are rating for a job completed, Recruiter feedback about the job done, Notifications, Weekly Schedule maintenance. Processes of the system are collecting data, analyzing data and job posting, generate weekly schedule and complete the schedule.

2.3. Not In Scope

This project only deals with the jobs categories enlisted and won't work for deviated categories. Moreover, project does not asurity for any loss

2.4. Project Objectives

This platform will provide features that will connect workers & recruiters. It helps to solves many problems as it helps provide jobs to the workers who are looking for jobs, and the recruiters who need not have to travel places or contact some mediator to find skilled workers. This platform can be used from anywhere and anytime according to our convenience, it is easy to understand to the workers who have basic knowledge of technology.

Worker verification will be done by the platform while enrolling them. It simplifies problem to many daily wages workers as they can be hired by recruiters through this platform after a through look of the workers' profile. Workers can also be able to reach the recruiters for the job and so, with the help of this application any layman workers can work without any barrier and will be able to earn their daily wages. This application is easy to use for all educated as well as uneducated background workers.

2.5. Stakeholders

From a business side of perspective, the following list of people are going to be the end users of the application

Some business stakeholders are,

- *A business suit of professionals who need to expand their business and find skillful people. Connect2Work offers little help when it comes to businesses who wish to broaden their network of people and grow with needed related people.*
- *Any person who needs a worker to do their household work.*
- *Worker who has some skills but didn't get much exposure and want to earn money.*
- *The developers who have some experience with the agile method, and the MERN stack, with some basic database intuition about the project.*

2.6. Operating Environment

Not much will be needed to operate the system. A regular and day to day computer with enough bandwidth to connect to the internet as with any application is just fine. The software itself operates on your regular Chrome, Mozilla, Edge browser, which exist on all operating systems and sometimes come pre packed with many new machines.

Thus, there is no,

• Specific version of any OS required – as all current Operating Systems support internet browsers.

• A small enough bandwidth to help you connect to the internet is enough. Nothing too demanding.

2.7. System Constraints

[Describe the constraints imposed on the system by the external environment. External environment may be caused by the stakeholders, business conditions, technical issues, academic requirements etc and may include the following:

2.7.1 Software Constraints

The software needs to deal with more upfront security issues, such as, checking if the email given even exists in the respective domain, thinking about how to deal with people who misuse the system, how to prevent spam and toxic users from jumping onto the platform, basically dealing with the community side of things, and making the perspective clear on how to maintain a healthy and growing community of users who don't abuse or exploit others

2.7.2 Hardware constraints

The application is not supported on mobile at the moment, and only works as intended on desktop systems. Other than that, any decent laptop with a decent bandwidth is enough to run the web application.

2.7.3 Cultural constraints

It's not valuable to the users if we show them posts from a completely different part of the world. The application will not be interesting if the post that the user sees does not even belong to their local area, because the average users wish to relate, which is the hook that keeps them coming back to the platform. In the absence of such a mechanism, a cultural constraint would be to find and fetch relevant information for the user - that is, finding the right profiles, jobs, and interests to keep the user on the platform.

2.7.4 Legal Constraints

Security, privacy, and data mining is a very huge legal challenge. If we put something like this in production, and Connect2Work grows bigger, we seriously would have to answer what data we keep in our system, why we do that, and have a privacy policy ready for us to go for. We also have to answer if our data goes to any third-party vendor, if it does, where is it stored, how secure it is, how easy it would be for a user to wipe their information for the application, and if the data collection we have and the recommendation system, if we introduce one to solve the problem in Cultural Constraints, if that system discriminates or prioritizes people, or if our system prevents people from getting the right exposure they should get for their posts.

2.7.5 Environmental Constraints

There are no such environment constraints that can be thought of at the moment

2.7.6 User constraints

The system wishes to target people starting in the age group adult. The initial release of Connect2Work emphasis on textual interaction, and not so much with graphical interaction.

.

2.7.7 Off the shelf components

It effect the development process in how the project is actually structured and developed. There is a strict procedure about how to add a frontend interface to the application, and as well a procedure for how to add something to the backend side of things for that frontend interface. The file structure is designed and defined to contain a set and fixed idea of how to look at that project.

Google API is used for sending emails to Gmail inboxes, and thus indirectly depends on it to send emails with the respective credentials.

2.8. Assumptions & Dependencies

The list of assumptions at the moment are,

- *A module for verifying the email address exists, but we did not manage the time to integrate this into the actual flow, so we're assuming that the email entered and given to us actually belongs to a user.*
- *A module to verify worker/recruiter identity already does not exist yet, so we're assuming that the information entered and given to us actually belongs to a verified citizen.*

Dependencies:

- *"react": "^17.0.1",*
- *"react-bootstrap": "^1.4.0",*
- *"react-dom": "^17.0.1",*
- *"react-icons": "^4.1.0",*
- *"react-password-strength": "^2.4.0",*
- *"react-password-strength-bar": "^0.3.2",*
- *"react-responsive": "^8.2.0",*
- *"react-responsive-cards": "^3.1.1",*
- *"react-router": "^5.2.0",*
- *"react-router-dom": "^5.2.0",*
- *"react-scripts": "4.0.1",*
- *"react-search": "^2.1.1",*
- *"react-search-autocomplete": "^1.0.6",*
- *"reactstrap": "^8.7.1",*
- *"cors": "^2.8.5",*
- *"express": "^4.17.1",*
- *"mysql": "^2.18.1",*

- *"nodemon": "^2.0.6"*
- *"styled-components": "^5.2.1",*
- *"@fortawesome/fontawesome-free": "^5.15.1",*
- *"@fortawesome/fontawesome-svg-core": "^1.2.32",*
- *"@fortawesome/free-solid-svg-icons": "^5.15.1",*
- *"@fortawesome/react-fontawesome": "^0.1.13",*
- *"@material-ui/core": "^4.11.2",*
- *"@material-ui/icons": "^4.11.2",*
- *"@testing-library/jest-dom": "^5.11.6",*
- *"@testing-library/react": "^11.2.2",*
- *"@testing-library/user-event": "^12.5.0",*
- *"axios": "^0.21.0",*
- *"bootstrap": "^4.5.3*
- *asgiref 3.4.1*
- *Django 3.2.7*
- *django-html5 1.0.0*
- *djangorestframework 3.12.4*
- *gunicorn 20.1.0*
- *pip 21.2.4*
- *pytz 2021.1*
- *setuptools 57.4.0*
- *sqlparse 0.4.2*
- *wheel 0.37.0*

3. External Interface Requirements

There are no external components to this project. Most of the components are internally a part of the software, and which is coded out, without the need for more external devices and machines. The application is based on a Graphical User Interface for the user to interact with by the use of buttons, fields to guide the user.

3.1. Hardware Interfaces

The supported device types are desktops, mobile, and web applications even though the styling and the order of the content might be messed up in all except desktop. Testing for the user interface needs to be done.

3.2. Software Interfaces

The system depends on a list of external packages, listed and updated with the use of the NPM package manager. The technologies used can be classified into three different components depending on the level of their usage. For example,

• Frontend

• Backend

• Deployment

The front-end is connected to the back-end via a list of functions described as the functional components that abstract away the HTTP protocols required to pass data back and from the backend. The back-end system uses those HTTP protocols and uses them to route to the appropriate service defined for it. Each functionality has an existing controller that takes care of the API to connect, receive, and send data to the backend. This effect takes place whenever a page loads to retrieve the context of the information. Thus, each call to the backend bundles up and finishes the database work on the server side, to return everything as a JSON back to the client side to store it in relevant states of the information to give the needed results.

To address the 'Describe the services needed and the nature of the inter-component communications.' we have two views offered by the application. This is for the logged in and logged out sessions that a user can see. For example, whenever a user is logged out, he or she is visited by the home page that gives the overview of the system, and a person who is logged on sees a dashboard that allows them to post a job or view weekly schedule depending on the type of user.

Now, to answer the 'Identify data that will be shared across software components', we can say that most, if not all services and frontend side requests, are primarily based on the MySQL database. Each controller service on the server's side will make sure that the user requesting the service is logged in, to maintain a session, and to not let users reach an undesired location – for example, no logged-out user can post a job.

3.3. Communications Interfaces

The requirements for login emails exist such that the user has given and entered an email that definitely exists on the database on Google for their Gmail service.

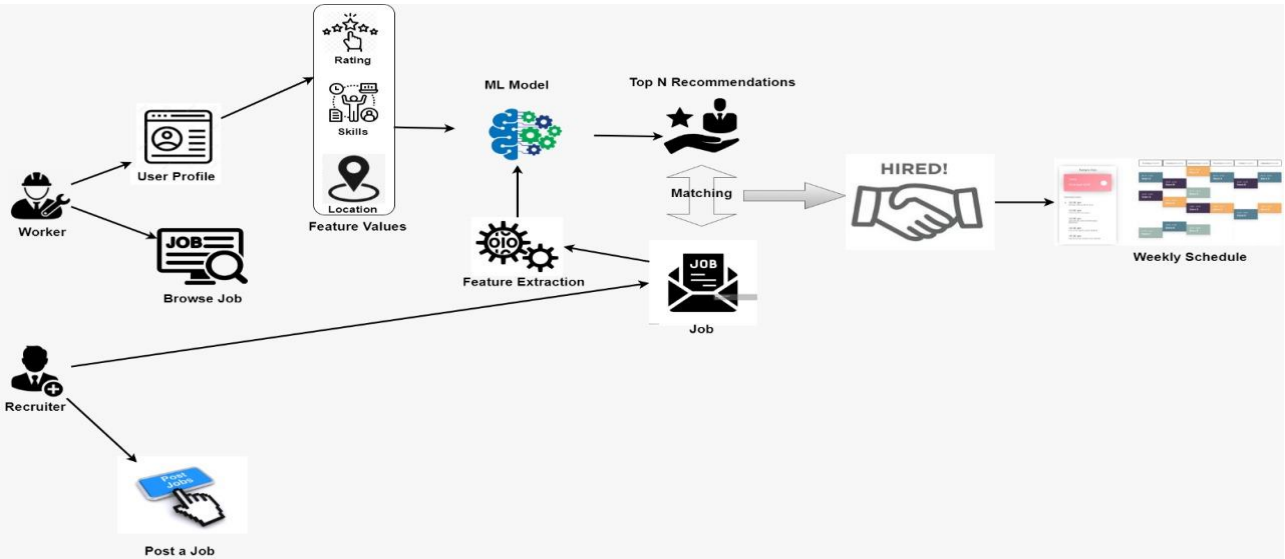
The requirements for login emails exist such that the user has given and entered an email that definitely exists on the database on Meta for their Facebook service.

The frontend will interact with backend database through API to send and receive data.

The system will also have a payment integration service like PayPal, Stripe and Zelle which will be communicated through API also.

4. Functional Requirements

4.1. Functional Hierarchy



4.1.1. Home Page

Any person visiting the website can view the features offered by the website, search for his particular demand, find a correct person for his job and much more. This page leads to many more functionalities like login, signup, about us.

4.1.2. Register Page

This page will add any upcoming new user to the system (Worker or recruiter). This page contains form, upon successful entry of data from the user, a confirmation email will be sent to his email id and upon email confirmation, that user will be added to the system and will be redirected to his/her personal Recruiter/Worker dashboard. The user can update his/her information on profile

4.1.3. Maintain Profile

This page is used to maintain profile for a user. This page also determines that the new upcoming user is a worker or a recruiter, and upon successful entry of information user is redirected to his dashboard.

4.1.4. Login Page

This page contain form to fill in information regarding user login and validate it. This page also redirects the particular user to his particular dashboard (worker / recruiter). It contains many more redirects or login options like.

4.1.4.1. Forget Password

A hyperlink that will redirect the user to the forget password page

4.1.4.2. Login With Facebook

A button which will redirect the user to the login via his Facebook account

4.1.4.3. Login With Google

A button which will redirect the user to the login via his Google account

4.1.4.4. Register Redirect

A hyperlink that will redirect the user to the signup feature

4.1.5. Worker Dashboard

This interface contains summary relating to progress of jobs he applied for, numbers of jobs successfully completed, current month earning, total earning and a navigation bar to look upon other features present. This page also contains a notification button that will show the latest changes/updates occurred regarding to job accepted. Once these changes are finalized by the dashboard user, specific updates will be made to his weekly schedule and summary. The functionalities that the navigation bar redirects to are,

4.1.5.1. Manage Job

Allows the worker to accept/reject new work offers. Cancel a work according to the policy specified by the website. Report a particular offer.

4.1.5.2. Settings

This tab redirects the user to change/edit profile page, which contains many editable input fields that allows the recruiter to update his profile. Attach newly acquired certificates, job experience, skill that relates to his work. Update profile photo and update information that are shown to the public.

4.1.5.3. Logout

Allows the user to logout of his session of login.

4.1.5.4. Weekly Schedule

A redirect tab that will lead to weekly schedule feature of the system.

4.1.6. Recruiter Dashboard

This interface contains summary relating to progress of jobs posted, number of jobs that were successfully completed, no of posted jobs that are not yet accepted, total payout and a navigation bar to look upon other features. This page also contains a notification button that will show the latest changes/updates occurred regarding to job posted. Once these changes are finalized by the dashboard user, specific updates will be made to his weekly schedule and summary. There are few functionalities that the navigation bar redirects to are,

4.1.6.1. Direct Hiring

A unique functionality that will allow the user to directly hire a previous worker with whom he had an amazing experience of the job. This functionality will directly offer the particular worker job, whom the recruiter has chosen and won't have to wait for matchmaking algorithm to show result.

4.1.6.2. Post A Job

This page contain form to fill in information required for his job, then it validates all the details inputted by recruiter and then try to match him with the most suited worker. Basically, a functionality that will allow the recruiter to post his particular requirements and get meet with the most suited worker for his page.

4.1.6.3. Manage Job

Contains many Allows the recruiter to accept/reject proposal made by matchmaking algorithm. Cancel a work, updating job information like changing working hours or day all according to the policy specified by the website.

4.1.6.4. Settings

This tab redirects the user to change/edit profile page, which contains the many editable input fields that allows the recruiter to update his profile. Update profile photo and update information that are shown publicly.

4.1.6.5. Logout

Allows the user to logout of his session of login.

4.1.6.6. Weekly Schedule

A redirect tab that will lead to weekly schedule functionality.

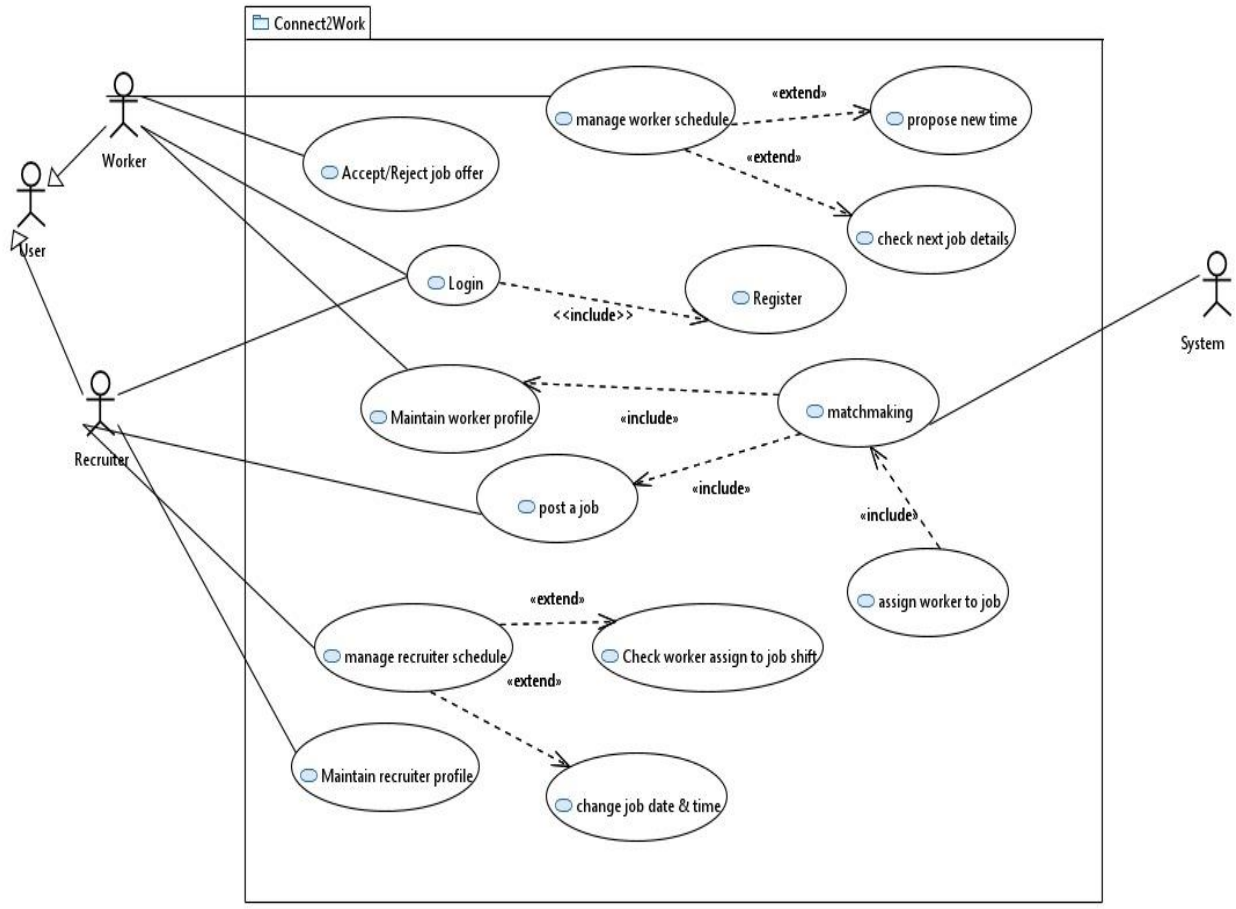
4.1.7. Weekly Schedule

This functionality is unique to our system. This feature shows the weekly schedule for any person using our system. In this way any worker/recruiter keep track of time, appointment and any unfortunate or fortunate updates in the schedule.

4.1.8. Matchmaking Algorithm

This functionality works when a job is posted. System would recommend top N recommendations according to the job posted. System will assign rank score to worker profile by considering different features like rating and reviews, relevant skillset etc. System would generate embeddings for both worker profiles and the job posted. That embeddings/feature vector is used to train the model with different hyperparameters. The ranking model is then evaluated using statistical methods.

4.2. Use Cases



4.2.1. Login

A user will be able to login the website to enjoy more features specific to his requirements, given that he is already registered on the system.

<1: Login>		
Use case Id:	1	
Actors: User		
Feature: *4.1.4 (Login)*		
Pre-condition:	User must have register before he tries to login	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	Enter email	
2.	*Enter password*	
3.	*Click Login*	*Email and password will be validated, if the user exists in the system, then he would be redirected to his / her dashboard*
Alternate Scenarios:		
3a: *If the entered email is wrong, then he will be asked to re-enter valid email* **3b:** *If the entered password is wrong, then he will be asked to re-enter valid password*		
Post Conditions		
Step#	**Description**	
1	Redirection to either recruiter or worker dashboard.	
Use Case Cross referenced	Post A Job	

4.2.2. Register

Feature that will allow a new upcoming user to get himself registered in the network

<2: Register>		
Use case Id:	2	
Actors: User		
Feature: *4.1.2(Register)*		
Pre-condition:	User knows how to input details	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	Enter Username	
2.	Enter Password	
Alternate Scenarios:		

1a: *A user already exists in the system with the given name, so he will be asked to change his username* ***2a:*** *User may be asked to re-enter password as if it not strong enough*

Post Conditions	
Step#	***Description***
1	User is redirected to maintain profile page.
Use Case Cross referenced	Maintain Profile uses this use case

4.2.3. Maintain Profile

Feature that will allow a new upcoming user to maintain his profile information and select his role

<3: Maintain Profile>		
Use case Id:	3	
Actors: *User*		
Feature: *4.1.3(Maintain Profile)*		
Pre-condition:	*User is registered in the system*	
Scenarios		
Step#	***Action***	***Software Reaction***
1.	*Enter Full Name*	
2.	*Enter Address*	
3.	*Enter Email*	
4.	*Selects his role (Worker/ Recruiter)*	*Renders field according to the role confirmed*

4: *It branches then into two sub-categories*
4a: *If user selects his role to be a worker, then he will be asked about to enter his skills, skills category, work experience*
4b: *If user selects his role to be a recruiter, then he will be asked about to enter his business name, office main location, business description.*

Post Conditions	
Step#	***Description***
1.	*User is redirected to his particular dashboard*
Use Case Cross referenced	*Register use case is used by it.*

4.2.4. Post A Job

Allows the recruiter to post a job, given that he is logged into his account. After successful posting, mange job feature is updated with the information of new job posted

<4: Post A Job>		
Use case Id:	4	
Actors: *User (*Recruiter)		
Feature: *4.1.6.2*		
Pre-condition:	*User is logged in*	
Scenarios		
Step#	***Action***	***Software Reaction***
1.	Enter Job Title	
2.	*Enter Job Start Time*	
3.	*Enter Job End Time*	
4.	*Enter Job Category*	
5.	*Enter Wage*	
6.	*Enter Job Description*	
7.	*Post Job Button Is Clicked*	*Notification of job posted successfully is issued.*
Alternate Scenarios:		
7a: *Notification is not issued; fields are marked red that don't have correct input*		
Post Conditions		
Step#	***Description***	
1.	*Job posted is made visible for worker to browse*	
2.	*A new job is added in manage job*	
Use Case Cross referenced	*Login*	

4.2.5. Accept/Reject Job Offer:

Allows the worker to either accept or reject a proposed job by the system. If accepted, its schedule will change accordingly.

<5: Accept/Reject Job Offer >	
Use case Id:	5
Actors: *User (Worker)*	
Feature: *4.1.5.1*	
Pre-condition:	*Woker is logged in and a new job proposal seeking choice is present*
Scenarios	

Step#	***Action***	***Software Reaction***
1.	*User view job offer*	
2.	*User accepts or rejects*	*Weekly schedule is updated*
Alternate Scenarios:		
Post Conditions		
Step#	***Description***	
1.	Weekly schedule of both the recruiter and worker are updated	
Use Case Cross referenced	Login, Post A Job	

5. Non-functional Requirements

5.1. Performance Requirements

Database would be designed in such a way that there will be no redundancy of data and no chance of producing anomalies. This would be done by normalizing the tables. This will reduce the query execution time and hence increases the performance.

5.2. Safety Requirements

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed-up log, up to the time of failure.

5.3. Security Requirements

Data of the Users would remain confidential and secure by the administrator.

5.4. User Documentation

At the moment, there is no such thing, but we can plan on delivering a tutorial on how to use the platform for new users unfamiliar with the platform to get them started quickly.

6. References

For reference purposes we visited many similar products currently available on website and a research paper similar to our app.

[1] WORKSAPP, May 2020, International Research Journal of Engineering and Technology (IRJET), https://www.irjet.net/archives/V5/i1/IRJET-V5I1143.pdf

2. www.taskrabbit.com

3. www.zaarly.com

4. www.needto.com

7. Appendices

We conducted research about the existing features of the same product as ours and concluded it in a matrix that will show what features we will be trying to give to the users and what others do not.