

Software Design and development Project

ASSIGNMENT-4 Group & Individual Report



Submitted by

Bishal Budhathoki (12116421)

Laxman Khanal (12123129)

Niraj Prasad Timila (12125320)

Raken Shahi (12119317)

Submitted to

Lecturer/Tutor: Indra Seher

Unit Coordinator: Lily Li

This assignment was done using Agile approach.

Table of Contents

[Introduction 2](#_Toc75034691)

[Design Implementation – User Interface 2](#_Toc75034692)

[Design Implementation – Data Base Interface 14](#_Toc75034693)

[Design Implementation – Core functions 19](#_Toc75034694)

[Testing Plan, results, and discussion 23](#_Toc75034695)

[Overall quality of work 29](#_Toc75034696)

[Link to our deployed website 29](#_Toc75034697)

[Individual Part 29](#_Toc75034698)

[Project in Requirement Stage 29](#_Toc75034699)

[Project in Design Stage 30](#_Toc75034700)

[Project in Implementation 31](#_Toc75034701)

[Project in Testing 31](#_Toc75034702)

[Lessons learned from the project 32](#_Toc75034703)

[Peer Review: 32](#_Toc75034704)

[References: 34](#_Toc75034705)

# Introduction

Our project Will Job Finder is a responsive web-based application which help job seeker to search and apply for the job. End user being Student and Employer, they both can have their own profile created and search for a job and recruit an individual for the required job, respectively. Basically, user can search job based on four categories: Software Development, Network, Business Analysis and Project Management.

This report represents the final report this project has to offer. Hence, in this report there are design implementation of User Interface (UI), data base interface and core functions. This report also explains how phases like requirement, design, implementation went. At the end this report also includes some of the lessons learned and the quality of work of the project.

# Design Implementation – User Interface

1. Dashboard

Graphical user interface, application

Description automatically generated

Figure 1 – Dashboard

Above given image shows the main page for our system. It includes a “Dashboard”, “Jobs”, “Users”, “Profile” and “Logout” in the side tool bar. These side tool bar changes according to the type of user i.e., **Admin** or **Employer** or **Job** **Seeker** Main panel consist of search text field and list of all unapproved and approved jobs. When someone search for the job according to the category then related job is displayed.

Text field and a button. In this page, the user can search for the category of vacancy. When user provides the job category and press the search button. Then the list of jobs is displayed.

1. Login Page

Graphical user interface, application

Description automatically generated

Figure 2 – Login Page

The above given page is for the login page. It includes input for email and password. When the user provides their credentials then the credentials are verified, and the user is authorized to the system.

1. Registration Page

Graphical user interface, application

Description automatically generated

Figure 3 – Registration Page

The above given page is for the Registration page. It includes the fields required for the registration of the user such as Email address, First name, Last name and password of the user. While registering for the account one must select whether the user is “Job Seeker” or “Employer” with the help of buttons given. Here the email address must be unique for each user.

1. Add Job Vacancy

Graphical user interface, application, Teams

Description automatically generated

Figure 4 – Create Vacancy Page

The above given page is for creating the job vacancy. It includes input for job details such as, job title, company, salary, job description and expiry date.

1. Apply Job

Graphical user interface, application, Teams

Description automatically generated

Figure 5 – Apply Job Page

Graphical user interface, application, Teams

Description automatically generated

Figure 6: Apply Job Form.

Graphical user interface, text, application, email

Description automatically generated

Fig 7: Web page after applying job.

The above given page is for the Apply Job page. Only the approved jobs which are verified by admin can be applied. Here when a student selects for a job, then he/ she is redirected to this Apply Job form page. And they can apply the job.

1. List All users as admin.

Graphical user interface, application, Teams

Description automatically generated

Figure 8- List Users Page.

This page is to list out all the registered user from the database. This page is only accessible to admin.

1. Delete users as admin.

Graphical user interface, application, Teams

Description automatically generated

Figure 9 - Delete user page.

This page is used to delete the user from the database by taking the reference of their user id.

1. Approve Job Vacancy and Delete Job as Admin

Graphical user interface, application, Teams

Description automatically generated

Figure 10 – Delete Job vacancy page as Admin.

This page is used to approve the pending jobs which are posted by employer. Admin first verifies the job and hits the approve job button to approve the job. And if there are invalid or spam jobs then admin has the right to delete those jobs by clicking on delete button.

1. Edit and Delete Job Vacancy as Employer and view applicants.

Graphical user interface, application, Teams

Description automatically generated

Figure 11- Delete Job vacancy page as Employer.

In this page, the employer can edit and delete the job which are stored on the database. Besides that, the applicant can view the applicants who have applied on that specific job. The plus icon is used to add new job post.

1. List Candidates

Graphical user interface, application

Description automatically generated

Figure 12 – List Candidate Page

The above given page is for listing those candidates who have applied for the specific job. When the employer wants to check those students, who applied on their specific vacancy can check form this page. And if the job seeker has uploaded their CV then the employer can download it via “DOWNLOAD CV” button.

1. Profile Page

Graphical user interface, application

Description automatically generated

Figure 13 - Profile Page of the current user.

This page gives the detail information of the user, and they can update their profile if they want by clicking “EDIT PROFILE INFO” button. Users can drop their CV / RESUME here and later this CV / RESUME can be downloaded by the employer.

# Design Implementation – Data Base Interface

Database interface’s design implementation reflects the database design of the system. We have used and designed the database previously, but we are going to look at it more.

There are some tables to store the data in the database with all possible primary keys, foreign keys to manage their relation between entities. Some of the tables in the database are described as below:

**Database Connectivity:**

The database used in this project is MYSQL and the server used to deploy this project are “Heroku” and “Google Cloud”. Client and Server is uploaded on Heroku server whereas for storing database, Google Cloud is used.

The following image shows the database and the required table fields.

Graphical user interface, text, application

Description automatically generatedFigure 14: wiljob database.

**USER Table:** This table has fields like user\_id, first\_name, middle\_name, last\_name, email, phone\_number, date\_of\_birth, user\_type, password, description, skills, interests and download\_src.

This table stores the information of the user such as whether the user is admin or student or employer, user’s interest, skill and description of the user. This table consist of the major data of the users and consist of email and password. The email must be unique for each user and email id and password is used for the user login. In this table user\_id is the primary key.

Graphical user interface, application

Description automatically generated

Figure 15 – USER Table

**Job Table:** This is Job table that has job\_id (Primary key), user\_id (foreign key) post\_title, post\_description, post\_salary, post\_date, expiry\_date. Every user who is recruiter has one individual entry in this table where user\_id is the unique id of the recruiter. Following is the image of table Job.

Graphical user interface, text, application, email

Description automatically generated

Figure 16 – Job Table

**Appliedjobs Table:** This table has fields like applied\_id, applied\_user\_id, job\_id and cover\_letter. Here applied\_id is the primary key while applied\_user\_id and job\_id both are foreign key referencing the primary key of user table and job table.

This table stores the information regarding the job applied by the student. If student applies for a job then his/her id is taken and id of the selected job is taken and stored in this table with cover letter.

Graphical user interface, text, application

Description automatically generated

Figure 17 – Appliedjobs Table

**Users\_file Table:** This table has fields like id, file\_src and cv\_user\_id. Here id is the primary key and cv\_user\_id is the foreign key referencing student.

This table stores the resume or CV address of the student and store them according to the id of the student.

Graphical user interface

Description automatically generated

Figure 18 – Users\_file Table

# Design Implementation – Core functions

To make this project happens, there are some backends code that is needed for the project to function well. Some of the APIs used are described below:

1. Creating user:

With this function, user is created and stored into the user table of the database. It takes different details of the user from the front end and proceed to store to database via insert query. If the process is successful, then it will send successful message else it will log insert error.

Text

Description automatically generated

Figure 19: Post users API.

1. Get all user:

With this function, all the users stored in the user table are retrieved.

Text

Description automatically generated

Figure 20: Get users API.

1. Login:

This function is to use for authentication of the user. Email and password are provided from the front end. If the given email and password is correct, then the user can login on to the system.

Text

Description automatically generated

Figure 21: Login API.

1. Create Job Vacancy:

With this function, the employer posts the job vacancy and is stored in the job table of the database.

Text

Description automatically generated

Figure 22: Create Job API.

1. List all jobs:

This function lists out all the jobs stored in the job table and users can check the jobs.

Text

Description automatically generated

Figure 23: List All Jobs API.

1. Search Job:

This function is used to search the job according to the type such as software development, Network, Business Analysis and Project Management.

A screenshot of a computer

Description automatically generated with medium confidence

Figure 24: Job Search API

1. Apply Job:

With this function, the user can apply for the job and the details such as user id, job id and cover letter details are then stored in the appliedjobs table.

Text

Description automatically generated

Figure 25: Apply Jobs API

# Testing Plan, results, and discussion

Testing is the crucial phase in the development of the software. A document that depicts the software testing scope and activities is a testing plan (Test Plan - SOFTWARE TESTING Fundamentals, 2021). This is something that is done before deploying the software and after integrating all modules and prototypes.

**User testing:**

The process of testing interface and function of a software, website and more by a real user to perform specific required tasks in realistic conditions is user testing (What is User testing? Definition - Omniconvert, 2021). For this project, all the testing were done by the developers and the end user being the tutor in the demonstration. The outcome that we received were a successful one with friendly user interface and simple navigation to let the user do their required task. Please see the given table for the results:

**Admin:**

Table for the requirement of Admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task** | **Input** | **Expected Result** | **Obtained Result** | **Screenshot of result obtained** |
| Homepage | URL | Webpage | Successful webpage UI. |  |
| Login | Email address and password | Validate credentials and then login. | Expected outcome. |  |
| Approve Job | Job Id | Confirmed Vacancy | Same as expected. |  |
| Edit Profile | Provide personal details of the user. | Successfully update the personal profile. | Expected outcome. |  |
| List Users | User id | Show all the users. | Expected outcome. |  |
| Delete Users | User Id | Delete the expected user. | This feature is yet to be lived in next version. |  |
| Delete Job | Job Id | Job successfully deleted. | Expected outcome. |  |

**Employer:**

Table for the requirement of employer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task** | **Input** | **Expected Result** | **Obtained Result** | **Screenshot of result obtained** |
| Homepage | URL | Webpage | Successful webpage UI. |  |
| Login | Email address and password | Validate credentials and then login. | Expected outcome. |  |
| Add Job | Provide Job details | Job successfully added | Expected outcome. |  |
| Delete Job | Press Delete button. It takes job id and deletes the selected job. | Job successfully deleted. | Expected outcome. |  |
| List candidates | Job Id | List candidates who applied specific job. | Expected outcome. |  |
| Edit Profile | Provide personal details of the user. | Successfully update the personal profile. | Expected outcome. |  |

**Student:**

Table for the requirement of Student:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task** | **Input** | **Expected Result** | **Obtained Result** | **Screenshot of result obtained** |
| Registration | Name, Email, Password, Address, User type, phone number, date of birth. | User successfully created | New user |  |
| Login | Email address and password | Validate credentials and then login. | Dashboard page |  |
| Apply Job | Select the appropriate job | Successfully apply for the job. | Required outcome. |  |
| Edit Profile | Provide personal details of the user. | Successfully update the personal profile. | Expected Outcome. |  |

# Overall quality of work

Although we had a bumpy journey as one of our team members had major health issue, we as a team did not let it affect in our project quality from implementation to documentation. We had set a time frame and set tasks to every individual. Generating designs to make prototypes, integrating them, and making final product with every documentation were done. From the extensions provided by the Unit Co-ordinator, we were able to catch up on our milestones and ready the product on time. Every code used are formatted, indented and with comments so every person seeing the code understands what is going with the code. We used the tools like VSCode, MySQL Workbench, POSTMAN, Heroku and Google Cloud. VSCode is the code editor where coding and in line documentation is done. MySQL Workbench is the database server and used to store our data and information. Postman is a collaboration tool for API Development. In context of language, we used React JS for the front end and Node JS for the back end. Heroku and Google Cloud is used for deployment of the website. We had regular meeting, zoom call and WhatsApp so we could know where every individual is with the milestone and help out if required. With rigorous testing, documentation, prototyping and reporting we as a team gave our best effort to maintain the quality of project and is phenomenal.

# Link to our deployed website

* Link: <https://wiljobui.herokuapp.com/>
* **Default User Credentials**
  + Admin User Credentials: Email: [admin@gmail.com](mailto:admin@gmail.com) Password: admin
  + Employer User Credentials. Email: [employer@gmail.com](mailto:employer@gmail.com) Password: employer
  + Job Seeker User Credentials. Email: [jobseeker@gmail.com](mailto:jobseeker@gmail.com) Password: jobseeker

# Individual Part

# Project in Requirement Stage

There is some need that requires a product or a service. That need is a requirement for a product or a service. This is also a stage where we choose the software development model. For this project we choose Agile Development Model as per the requirements. Every aspect in the development of the project is discussed here. All requirements are analysed thoroughly to make discission for other stages like tools, language to use, database to use, deployment tools, design tools and much more.

Functional requirement determines the basic system behaviour that explains what to do and what not to do for a system. When an input is given system interacts with that data and provide the output. The interaction of data is a business process, and the output determines it is intended or not (Functional vs Non-Functional Requirements: The Definitive Guide - QRA Corp, 2021).

Non-functional requirements are to specify how the system should do it for better usability. This does not affect the products results, behaviour, features. It is more focused on product properties and user expectations (Functional vs Non-Functional Requirements: The Definitive Guide - QRA Corp, 2021).

For this project, WilJobFinder, there are functional and non-functional requirements.

Functional are: description of data to enter

Who can post job vacancy?

Description of business operation performed by each screen

Non-functional are: Student cannot see who has applied for the job.

Only recruiter can post job.

|  |  |
| --- | --- |
| **Use Case** | **Functional Requirements** |
| View home page | Admin, Student, and employer can visit homepage and select the required options. |
| Login / Registration | Every new user can register or login into the system with username and password whereas there can only be specified admin |
| View Jobs Posted | Student must be able to see the job posted by the employer |
| View Jobs Applied Students | Employer must be able to see the job applied students |
| Delete Job | Admin and employer have right to delete the job vacancy post |
| Handle complete application | Admin can manage complete application and the data with it. |
| Handle login | Admin has access to login credentials and handle users with their username and password with a unique number assigned to everyone |
| Handle registration | Admin is to handle the registration process with all data provided stored in the database |

# Project in Design Stage

Before the development of the project itself, various requirements of the client were collected and then diagrams, prototypes were made. The diagrams made and used are architecture of the software, Work Break-Down Structure, Activity Diagram, Gantt Chart, Class Diagram.

All these diagrams are made using draw.io All the diagrams were easy to draw but lot of time were invested in making them. Class diagram took a while as well as WBS. In the development of the products, many mistakes were realised and were corrected. Protypes were made using Figma. This is a very easy tool to use. Every design diagram made has been beneficiary to us and for the project.

# Project in Implementation

Various research, brain storming, user requirements validations are to be seen in this stage. This stage is an end part and for this task were divided among them members where one creates a base file that contains dummy user interface, create database that everyone can reference to and make required tables and a file/folder for backend APIs, make prototype design for the UI.

Another member is fully focused on creating user interface using react also making sure it fulfills the user requirements. Another two members are focused on building core backend APIs using NodeJS, and libraries. This also include creating database on the go when needed.

We as a team communicated with each other in every phase of the project development, shared individual ideas, researched for some issues faced during the development. This has led to successful development and deployment of the project.

# Project in Testing

Minor testing was done when every APIs were built. But as a complete system it was done after the system was ready as per the work break down structure. Multiple iteration of testing was done to verify that all functions and tasks required are done completely and correctly. Requirements were checked as specified by the client and all requirements are fulfilled. There were some issues like database not found even when connected, reference errors, library reference error and many more. Multiple code testing was also done to confirm the correct output. There were rigorous testing and results obtained were all success as shown in given testing tables in the individual section 5 and as per the requirements. The system was not only tested by the developing team that is us but also by external user that consider the system as simple yet effective as a user and for the recruiter. Overall testing performed are as below those covers the system to verify use case implementation.

**Student/Jobseeker:**

|  |  |
| --- | --- |
| **Use Case** | **Test Result** |
| Login | Success |
| Registration | Success |
| Profile Edit | Success |
| Job Search | Success |
| Resume upload | Success |

**Employer:**

|  |  |
| --- | --- |
| **Use Case** | **Test Result** |
| Login | Success |
| Registration | Success |
| Job Post | Success |
| Update Profile | Success |
| Edit Profile | Success |
| Manage Student Application Details | Success |

**Admin:**

|  |  |
| --- | --- |
| **Use Case** | **Test Result** |
| Approve Post | Success |
| Delete Post | Success |
| Delete users | Success |

# Lessons learned from the project

This project did not go as we anticipated as it would go, and there were various ups and downs. We all have different skill sets and we were determined that everyone has their contribution in every phases so that everyone knows what they are doing and increase their skill set. Hence, we all have increased our skill set, programming ability, coding styles, documentation, reporting and communication skill and problem handling and debugging. We all usually communicated online through video call and audio call using Zoom and WhatsApp. From this communication skill has been greatly improvised. We also need to go to work and then also manage time for the project and maintain social balance. This project certainly taught us the time management skill. As we had work together in a project in different unit, we did know each other skills and had proper communication that leads us to better teamwork. This teamwork also taught us multitasking as we also had to handle a major project in different unit, handle social life and one team members health was not on top and had to go to hospital in regular basis. Yet we believed in him and supported him in his hard time, and he managed to get on his milestone.

# Peer Review:

Our team consist of 4 members including me. All have their say in this project and I am reviewing them in terms of punctuality, contribution and more as given below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Team Member | Punctuality Rating | Contribution | Professionalism Rating | Communication Rating | Quality of work |
| Raken Shahi | 5 / 5 | High level Backend APIs, code maintenance and some part of report | 5 / 5 | 5 / 5 | Delivered high level of quality of work that might need some polishing |
| Niraj Timila | 4.8 / 5 | Created base file and database required in initial phases, some backend APIs, building prototype | 4.5 / 5 | 4 / 5 | All provided files were highly reliable |
| Laxman Khanal | 4 / 5 | Front end User interface | 4 / 5 | 4 / 5 | Reliable user interface, simple, clear, and concise |

Raken has always been punctual in every meeting whether in video call or audio call or face to face meeting. Even when at work he replies to the text message when anyone is stuck with an issue. So, there is no doubt on his communication and professionalism skills. Every work done by him has been of high quality.

Niraj has given high priority to this project and has been available for any task that needs attention and has been assigned to him. If any issue he has communicated very well so we all work on that issue if required. His skills and efforts are tremendous for this project and has not said no to any task assigned.

Laxman is a smart guy and with some of his professional experience he has helped in user interface development and guided us major decision taking phases, backend API guidance. He was not available always as he must manage a lot of things like professional life, projects, other units, and his social life too.

# References:

SOFTWARE TESTING Fundamentals. 2021. *Test Plan - SOFTWARE TESTING Fundamentals*. [online] Available at: <https://softwaretestingfundamentals.com/test-plan/> [Accessed 9 June 2021].

Omniconvert. 2021. *What is User testing? Definition - Omniconvert*. [online] Available at: <https://www.omniconvert.com/what-is/user-testing/> [Accessed 9 June 2021].

QRA Corp. 2021. *Functional vs Non-Functional Requirements: The Definitive Guide - QRA Corp*. [online] Available at: <https://qracorp.com/functional-vs-non-functional-requirements/> [Accessed 16 June 2021].