JOB ON THE DOT
(ONLINE JOB BOARD SYSTEM)

FARIS BIN TAMRULAN

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

JOB ON THE DOT
(ONLINE JOB BOARD SYSTEM)

FARIS BIN TAMRULAN

This report is submitted in partial fulfillment of the requirements for the Bachelor of Computer Science (Database Management) with Honours.

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2020/2021

DECLARATION

I hereby declare that this project report entitled

JOB ON THE DOT (ONLINE JOB BOARD SYSTEM)

is written by me and is my own effort and that no part has been plagiarized

without citations.

STUDENT : _____ Date : 5 SEPTEMBER 2021

(FARIS BIN TAMRULAN)

I hereby declare that I have read this project report and found

this project report is sufficient in term of the scope and quality for the award of

Bachelor of Computer Science (Database Management) with Honours.

SUPERVISOR : _____ Date : 5 SEPTEMEBER 2021

(TS. DR. YAHAYA BIN ABD RAHIM)

DEDICATION

My dissertation is dedicated to my family, friends, and lecturers. My heartfelt gratitude goes out to my loving parents, Tamrulan bin Sukhairi and Siti Dahniar binti Sahdan, for their unwavering support and commitment. They also showed me that if you take it one step at a time, even the most difficult work can be completed. I also dedicate my dissertation to my friends, who have been supportive of me during the project's development. Last but not least, I'd like to dedicate my project to Ts. Dr. Yahaya bin Abd Rahim, my supervisor, for his great ideas, inspiration, and continuous encouragement in ensuring that I completed this project on time.

ACKNOWLEDGEMENTS

First and foremost, I want to convey my gratitude to everyone who has contributed to the project's successful completion. My supervisor, Ts Dr. Yahaya bin Abd Rahim, deserves special thanks for guiding me through the project and providing helpful recommendations and direction on how to finish it. Because to his direction, recommendations, motivations, and ideas throughout the project's duration, it was finished effectively.

I'd also like to express my gratitude and special thanks to everyone who has helped me with this project, both directly and indirectly. I am eternally grateful to my loving parents, family, and friends for their unwavering support and encouragement.

ABSTRACT

The project is to develop a web-based application system for job employment which allow people to apply for a job and employer to post a job opening. This system is called Job on the Dot (JOTD). The system allows job seeker to not just apply job but also post their availability such as freelance services to get more attention from employer. The system provides a simple application form and job opening form that can be easily filled by the user. To overcome a fake job listing in the system, the staff will need to filter out all the job opening applied to the system.

ABSTRAK

Projek ini adalah untuk mengembangkan sistem aplikasi berasaskan web untuk pekerjaan yang membolehkan orang melamar pekerjaan dan majikan untuk membuat pembukaan pekerjaan. Sistem ini dipanggil Job on the Dot (JOTD). Sistem ini membolehkan pencari kerja tidak hanya melamar pekerjaan tetapi juga memposting ketersediaannya seperti perkhidmatan bebas untuk mendapatkan lebih banyak perhatian dari majikan. Sistem ini menyediakan borang permohonan sederhana dan borang pembukaan pekerjaan yang dapat diisi dengan mudah oleh pengguna. Untuk mengatasi senarai pekerjaan palsu dalam sistem, kakitangan perlu menyaring semua pembukaan pekerjaan yang berlaku pada sistem.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

PAGE

LIST OF FIGURES

PAGE

LIST OF ABBREVIATIONS

FYP	**-**	**Final Year Project**
ERD	**-**	**Entity Relationship Diagram**
DDL	**-**	**Data Definition Language**
JOTD	**-**	**Job On the Dot**

UTeM
اونيۏرسيتي تيکنيکل مليسيا ملاک
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

CHAPTER 1: INTRODUCTION

1.1 Introduction

In 2020, the world has been hit with a global pandemic. As many as 772,900 people are unemployed in Malaysia. Majority of people have lost their job because their company revenue has been decreased and needs to be downsizing. People have been looking for a job and some even do part time or freelance jobs. According to Danielle Elmers, 70 percent of people who are currently employed are considering a new job. However, with the Malaysian movement control order (MCO) people find it difficult to apply for a job.

Small business is the most affected business during the MCO. This is because they struggle to find employee. This have not to be a problem for big companies as they can afford to advertise their job vacancy on a bigger platform such as an online job board. Some small businesses are now changing their way of hiring people by online instead by word of mouth through their networking. However, small business find is hard to post at an online job board as it requires a lot of form to be fill before can get it posted. Small business owner instead uses their social media account such as Facebook to post their job opening. Job seeker will have difficulties to find the job opening in social media because the post will only allow certain user to view the post.

Job seeker also wants to get hire fast so that they can afford to continue to live. One of the initiatives they make is by posting at their social media account by telling their network that they are available and open for work. As they believe that this is the best way to get hired fast. However, it is only limited to people that follow their social media account.

This project will be developed as a job search portal for job seeker and employer. Job seeker will be able to search job, store their resume and post a freelancer job. From the employer side, they will be able to post their job opening and find job seeker who are available to do a freelance job.

1.2 Problem Statement

Finding a job opening is also as hard as posting a job opening. This is because small company and business have difficulties in posting job at current job searching website because a lot of details need to be fill. Small business owner does not know what to fill in each column of the form.

Furthermore, job seeker lack of features such as posting their availability and freelancing posting. The lack of this feature will be hard for the job seeker to stand out. Besides, a lot of false advertising job shown at current job search portal. Job seeker will struggle in finding a genuine job opening.

1.3 Objective

The objective of this system is:

i. To provide an easy and convenient form for the employer to advertise their job opening.

ii. To create a feature for job seeker to post freelance service and job availability.

iii. To be able to filter out and verify the job opening before advertising it to the public.

1.4 Project Scope

The module that will be develop for this project are:

1) Registration for job seeker and employer.
   - User will need to fill up the basic information and profile picture.

2) Search Module
   - Job seeker can search for their desire jobs.
   - Employer can search for job seeker who are available or offer a freelance service.

3) Job application module
   - Job seeker can apply for job opening by clicking apply for the job.

4) Job Advertisement module
   - Employer will need to fill up a simple application form before can post the job.

The target user for the system:

1) Job Seeker that are looking for job or offering a freelance service.
2) Employer that has small business and need employee immediate.
3) Staff that filters out the job that are genuine.
4) Admin to handle and maintain the system.

1.5 Project Significance

This system will be able to help job seeker to find a real job based on their desired search. The system able to help job seeker to post their availability for a job as this will help the employer to find immediate staff. The freelance feature will help the job seeker to get notice from potential customer. This system will have dual interaction between the employer and job seeker.

1.6 Expected Output

The expected output of this system is job seeker able to find their desired job. Employer able to post their job opening for the public to view. Employer can view all the application made by job seeker on their job opening. Job seeker can offer freelance job to employer who need it.

1.7 Conclusion

This chapter has provided an overview of the system by highlighting the objective and the scope of the system. In doing so, the chapter briefly stated the expected output of the system for the user. The chapter also reflected the problem statement and the background of the system from the current situation in job application world.

CHAPTER 2: LITERATURE REVIEW AND PROJECT METHODOLOGY

2.1 Introduction

This chapter will concentrate on the project methodology and schedule of the project that needs to be carry out. This chapter also discuss on the Database life cycle (DBLC) and the task that needs to be done on each phase of the DBLC. The main objective of this chapter is to make a proper planning and scheduling of the project for it to be able to complete it on time.

2.2 System Review

This section will analyse and compare the current system to identify the specification of the system and weakness of the system. The system that are chosen in this observation are Maukerja, JobStreet and Indeed. After observation of these system, there are some differences between the systems. The results of the observation can be found in the table below:

Table 2.1 Comparison between current online job board

System	Description	Advantages	Disadvantages
Maukerja	A job portal site that posts job opening and seeker can apply for the job.	Have forum that user can interact with each other. Store resume and profile of user.	Job seeker and freelancer cannot share their availability to employer. False job opening.

JobStreet	A job portal site that posts job opening and seeker can apply for the job.	- Employer can filter candidate that have apply for the job. - Can store resume and profile of candidate	- Candidate cannot get share their availability to employer.
Indeed	A job portal site that posts job opening and seeker can apply for the job.	- User can give review to employer or candidate.	- Candidate cannot get share their availability to employer.

2.3 Project Methodology

2.3.1 System Development Life Cycle

1. Planning and Requirement Analysis

At this point, all the project's requirements and risk identification have been completed. It is critical to identify the various technical approaches that can be used to successfully implement the project while minimizing risk. All the requirements are met by conducting research to determine what is required in the system. The product requirement will be documented when the planning and requirement analysis is complete.

2. Design

Based on the requirements identified in the previous stage, the system design will be built to ensure that all requirements are implemented into this system. We use the Data Flow Diagram (DFD) to represent data flow in the system and the Entity-Relational Diagram (ERD) to describe the system's database in this system.

3. Developing the Product

The actual work starts at this stage, and the prototype is developed. The programming code is developed based on the design created during the Design process. Python and MySQL will be used in the coding to ensure the product's performance. Since the design is done in a systematic and structured manner, code generation can be completed quickly.

4. Testing the Product

When the product has built completely, the product needs to be tested to make sure there is no error or problem in the system. All the problem or error found in the product needs to be reported for further action that needs to be fixed before deployment.

5. Deployment

After the product has been fully tested without error, the product will be first released to selected user on the market. The feedback from the user will be collected to improve the product or unchanged it.

2.3.2 Database Life Cycle (DBLC)

In this system, the database life cycle approach is used as the technique to ensure that the system was built efficiently. The DBLC involves database development and resource allocation, database schema and data management, backup and recovery tasks, and database decommissioning. The DBLC consist of six phases which are database initial study, database design, implementation and loading, testing and evaluation, operation, and maintenance and evolution. Each phase needs to be complete before continuing to the next phase.

1. Database initial study

   In this phase, the analysis on the current job portal system is in the work. The main purpose of this phase is to analyse the system

situation, define the problems, constraints, objectives, scope, and its boundaries.

2. Database Design

   The second phase concentrate on the design of the database model that will this system operation and objectives, making sure that the finish product meets the user and system requirements. The conceptual design, logical design and physical design are developed and the selection of the DBMS software to used.

3. Implementation and Loading

   In this phase, the installation of DBMS and creation of the database is done. The data is then loaded or converted into the system.

4. Testing and Evaluation

   During this phase, testing and tweak the database to ensure that it performs as the desired output. This phase occurs as well as applications programming.

5. Operation

   Once the database has passed the evaluation stage, it is considered to be operational. At this point, the database, its management, user and application programs constitute a complete information system. The beginning of the operational phase invariably starts the process of system evolution.

6. Maintenance and Evolution

   In this phase, the database will need to undergo maintenance such as backup, recovery, enhanced performance, adding attribute and entities. The database version will be updated based on the situation in the future.

2.4 Project Schedule and Milestones

This is a project schedule and milestone that must have to followed according to the committee in implementation of PSM, it is to ensure that the project must be completed on time. A timeline schedule is made so that the task can be finish on time.

Table 2.2: Milestone for the project.

WEEK	MILESTONE	EXPECTED DOCUMENTS
1	Introduction and Project Methodology	Proposal PSM Proposal Assessment & Verification
2	Introduction and Project Methodology	Proposal Correction
3	Introduction and Project Methodology	Proposal Presentation & Submission PSM Chapter 1
4	Analysis and Design	Report Chapter 1 & 2
5	Analysis and Design	Report Chapter 1 &2 Correction
6	Analysis and Design	Progress Presentation 1 Report Chapter 3 & 4
7	Analysis and Design	Report 3 & 4 Correction
8	Implementation	Implementation 1
MID SEMESTER BREAK		
9	Implementation	Implementation 1 Project Demo
10	Implementation	Implementation 2 Project Demo
11 & 12	Implementation & Testing	Project Demo Report PSM 1
13 & 14	Testing	Project Demo Report PSM 1 Presentation Schedule
15	Documentation	Final Presentation

16	Documentation	Correction Report PSM 1 Submission of Logbook

Table 2.3: Gantt Chart

Activities	Weeks															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Introduction and Project Methodology																
Analysis and Design																
Implementation																
Testing																
Documentation																

2.5 Conclusion

In a conclusion, through this chapter it was focusing about project methodology that going to be used for the project. Agile Methodology that been used for System Development Life Cycle (SDLC) will be helpful in developing the system organized and followed the phase. Furthermore, the methodology for database in developing Job Portal Management System are based on Database Life Cycle (DBLC) approach. The benefits of using it was the DBLC never ends because database monitoring, modification, and maintenance are part of the life cycle, and these activities continue long after a database has been implemented. It simply put, the DBLC encompasses the lifetime of the database.

CHAPTER 3: ANALYSIS

3.1 Introduction

The method of system development includes analysis, which is the method of analyzing data by applying analytical and logical reasoning to each portion of the data presented. There are explanations about current system analysis and to-be system analysis in the analysis chapter. When doing a research study, this type of study was one of the many measures that had to be done. Data processing, text analysis, business intelligence, and data visualizations are only a few examples of different data analysis methods. In this chapter, the current system as well as the new system that will be developed will be explained.

3.2 Problem Analysis

The importance of a functional and user-friendly system in the development and design of a website or system should not be overlooked. In order to achieve this, conducting a research on all current and existing job portal website such as JobStreet, FastJob and MauKerja as well as interview several job seekers. There are some drawbacks with the current system and can be improvised.

One of the drawbacks faced by job seekers is that they are unable to post their availability to their potential employer. For example, a student who wants to work as a part time during their holidays cannot share their availability on the current job portal website but instead, they need to search and apply for part time job that are available currently.

Besides, job seekers cannot filter the job based on how they want to receive their payment. Job seekers who want to get their payment daily does not want a job that offers monthly payment. This will cause a problem for part timer that wants a payment based on their number of days working.

Next, job seeker who also a freelancer cannot share their freelancing service to potential employer on the job portal website. This will make it difficult for them as they need to post it on other website that they can share their freelancing services such as logo design service.

Lastly, the existence of fake and irrelevant job on job portal website can be a nuisance to the job seeker. This will mislead job seeker into applying these job that will give them false expectation. The existence of a job opening that has been filled or closed also do not help the job seeker in searching for a job.

3.3 The Proposed Improvements / Solutions

Based on the problem faced by the existing system, the implementation of this new system will be able solve the problem. The development of this web-based application system will allow job seekers to apply for job opening as well as posting their availability to potential employer. By simplifying the application form, job seekers will be able to apply for a job easily as they do not need to fill up a complicated form. Job seeker only send an introductory message and apply. Other than that, job seeker can post their availability to which are open to work. Job seeker can also use this function to promote their freelance service such as designing or dispatch delivery.

Besides, the employer has a simple form that requires them to fill in order to post a job. Employer also can search for potential worker by searching at availability that a job seeker posted. Employer can search for freelancer if there is a job that requires it.

In order to filter out the fake job, the role of staff in the system able to reject a job opening application from the system. This will help the system to only

have a genuine job opening so the job seeker can apply for it without feeling it is a fake job.

3.4 Requirement Analysis of the to-be System

With the addition of functionalities and features, the existing system analysis could be improved. The suggested system will analyze and develop the information system using the Structured System Analysis and Design technique.

3.4.1 Functional Requirement

The Functional Requirement describes the service that the system must provide. It refers to a software system or a component of one. A function is nothing more than the software system's inputs, behavior and outputs. The functional requirement can be visualize by the aid of diagram such as context diagram, data flow diagram and use case diagram.

3.4.1.1 Context Diagram

Figure 3.1 showing the context diagram for Job on the Dot system which it contains four entities interact toward system which are seeker, employer, staff and admin. An illustration of the data flow was given to the users as seen in the figure.

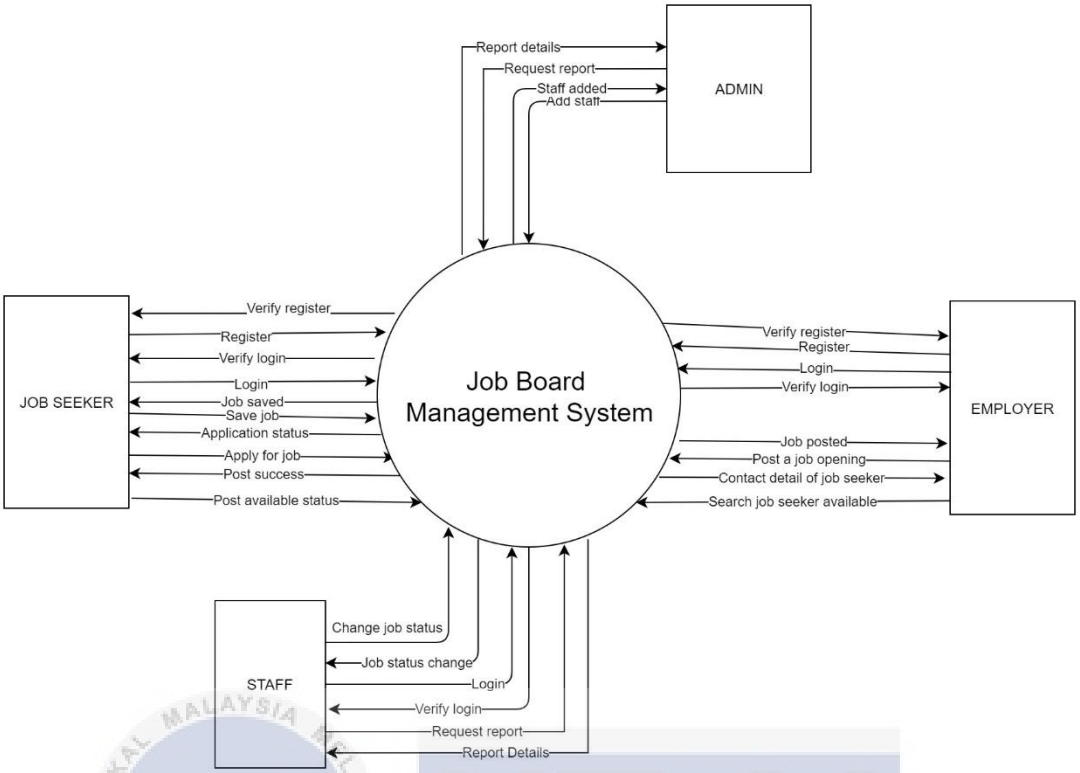

Figure 3.1: Context Diagram for the system.

3.4.1.2 Data Flow Diagram

Figure 3.2 shows Job on the Dot system data flow diagram level 0 contain eight processes. The data flow diagram also consists of four users which is Seeker, Employer, Staff and Admin.

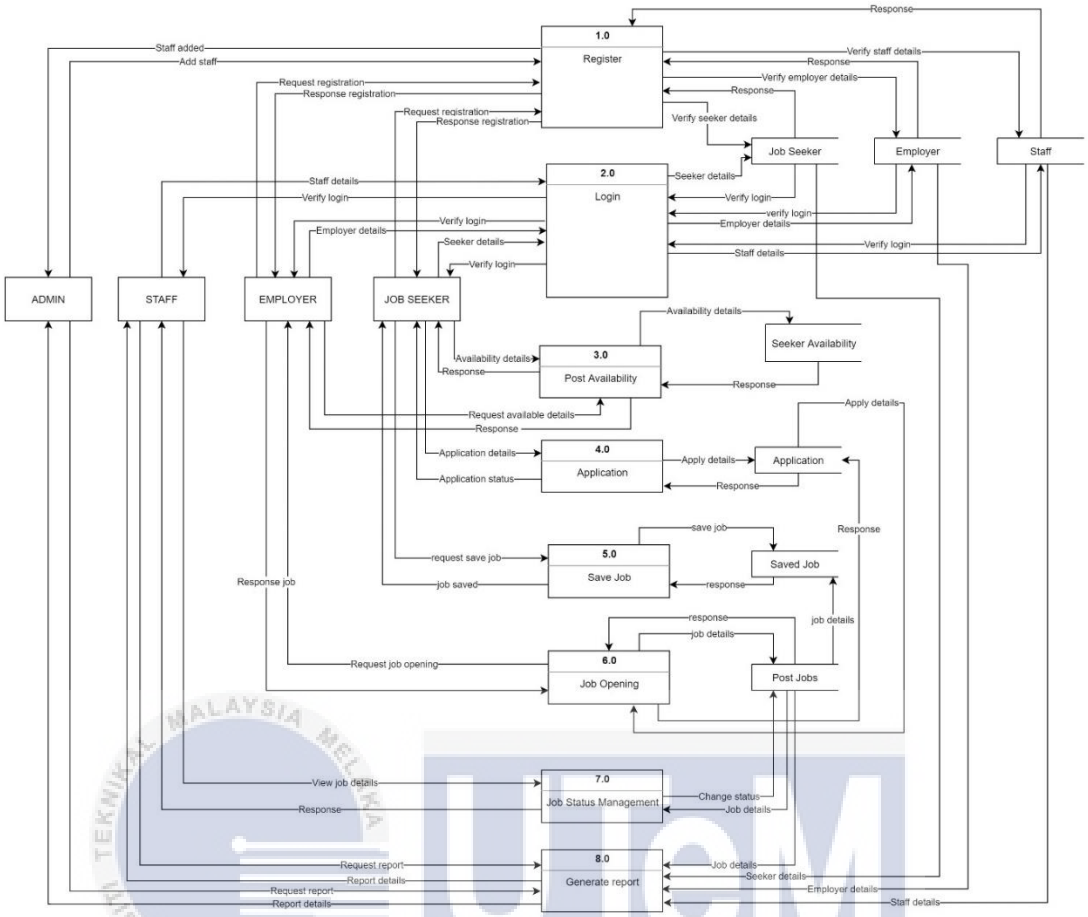

Figure 3.2: Data Flow Diagram Level 0 for the system.

3.4.1.3 Use Case Diagram

Figure 3.3 shows the Use Case Diagram for the system. It shows all the use case involving in the system including with all the actors which are seeker, employer, staff and admin.

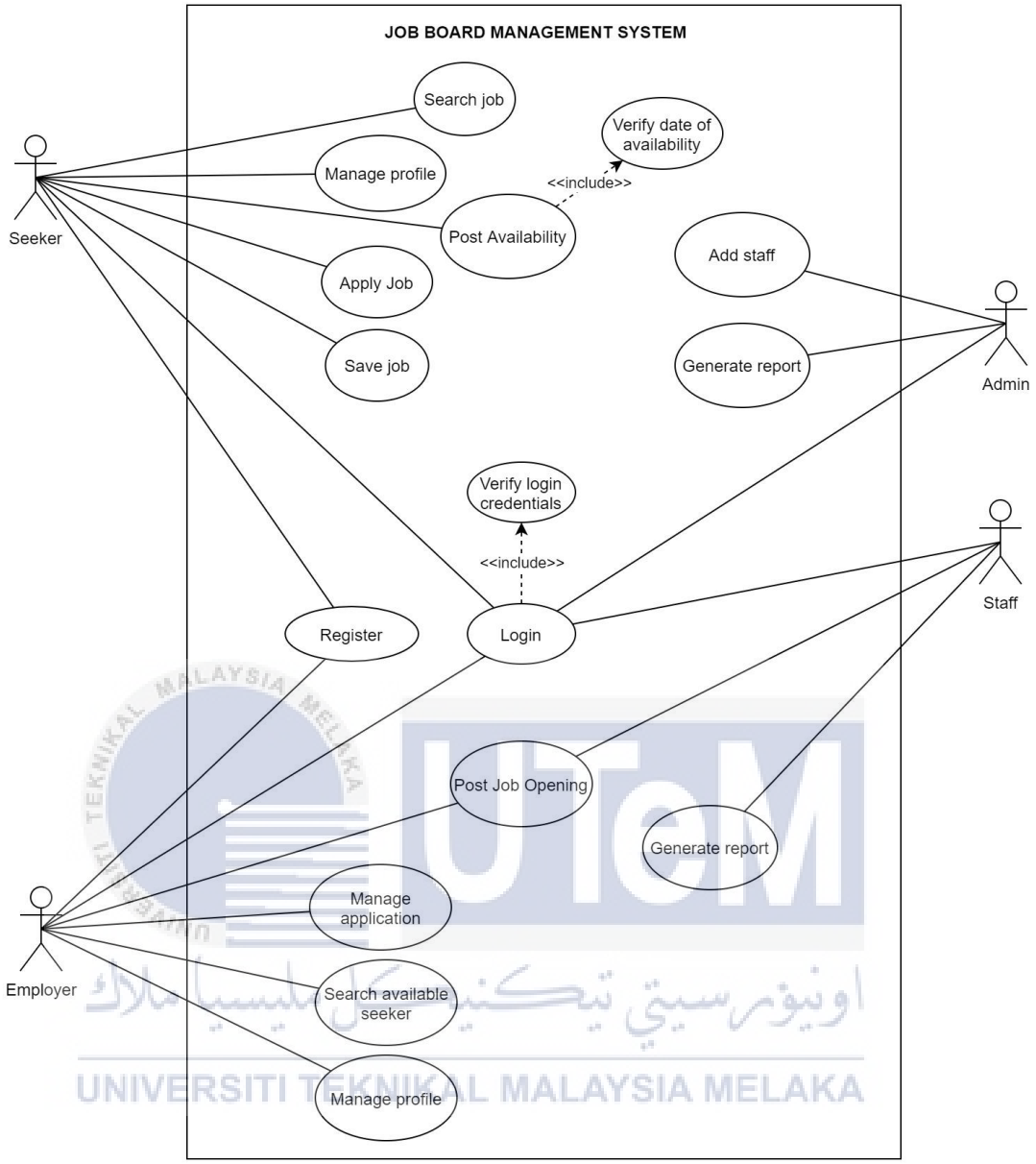

Figure 3.3: Use Case Diagram for the system.

3.4.2 Non-Functional Requirement

The non-functional requirement for this system ensures the usability and effectiveness of the entire system. Failing to meet these requirements may result in system that fail to satisfy the user. The requirements are:

i. Security

Identification: The system able recognize and differentiate the type of users that are register in the system.

Only authorized staff can update the job status of a job based on their assigned location.

ii. Accessibility

Only admin and staff can access the admin page which handle the back-end operation.

New user can access the job search page even without an account however they cannot save job or apply job without an account.

iii. Usability

Learnability: The interface of the system is simple and direct so user can easily understand what it does.

Efficiency: The form for posting a job or applying a job is simple and the user can complete it easily and efficiently.

3.4.3 Other Requirement

This project's requirements are broken into two sections: software requirements and hardware requirements. While the system satisfies the hardware requirements, the application requirements may dictate the application utilized in the system's development.

3.4.3.1 Software Requirement

The software requirement for this system is require in order to run the project system.

Table 3.1: The software requirement of the system.

NO	SOFTWARE	PURPOSE
1	Visual Studio Code	Text editor use for to do the coding for the system.
2	PostgreSQL	The database storage for the system.
3	Python	The server-side scripting language developed for web development.

4	Django	The framework for the python language to run the coding and deploy the server.
5	Draw.io	To create Entity Relationship Diagram (ERD), Flow Charts, and Data Flow Diagram.

3.4.4 Hardware Requirement

The hardware requirement for this system is to determine the hardware that needs to be used in order to be able to run the system smoothly.

Table 3.2: Hardware Requirement

NO	HARDWARE	DESCRIPTION
1	CPU	AMD Quad-Core A8-7410
2	RAM	8 GB DDR3 L Memory
3	Storage	500 GB HDD

3.5 Conclusion

The Context Diagram explained the overall system structure to be developed in general, whilst the Data Flow Diagram detailed more details regarding the system and database's input and output.

Creating data analysis methodologies aids in explaining the structure and importance of data in organizations. The data analysis methodology can be utilized as a first step in translating real-world dynamics into a notion that can be processed on a computer and accessed by multiple people.

This chapter was crucial in designing a more efficient system. The end user will find it efficient, stable, and functional. Furthermore, the system may be easily developed because the developer understands the system's flow. The current system is represented by a data flow diagram, whereas the new system is represented by a data

flow diagram. To conclude, the new system that will be constructed will be able to overcome the problem that the end user encountered with the current system.

CHAPTER 4: DESIGN

4.1 Introduction

The method of establishing a complete database data model is known as database design. The three phases of database design discussed in this chapter are conceptual database design, logical database design, and physical database design.

In the conceptual database design phase, it defines how distinct entities, such as objects and elements, are connected to one another. It also specifies the attributes that each entity possesses. It defines all terms used in the field of application, such as entities and attributes. The information in the users' requirements specification is used to create the data model. A conceptual data model is a source of information for the logical design process. In this approach, an Entity Relationship Diagram is use.

The data is organised into a set of logical relationships called entities and attributes throughout the logical database design phase. A piece of knowledge is represented by an entity. An attribute is a component of an entity that serves to describe the entity's uniqueness. The logical data model is refined from the conceptual data model. The data dictionary is the result of the database's logical design.

Finally, the database will be implemented using a physical database design, which is the goal of this step. Transform the data from the logical design phase into a description of the physical database, including tables and constraints, during this step. Everyone should now be aware of the database management system (DBMS) in use. Distinct DBMS, for example, have different names for data types and data types. A Data Definition Language is the result of this process.

4.2 Database Design

Database design is the second stage in a DBLC. The database design is a set of technique that helps in the designing, development and implementation of database to the system. It can be separated into three level which are conceptual, logical and physical design.

4.2.1 Conceptual Design

The conceptual design is the first stage in the database design process. The conceptual design is to design a database that is independent of database software and physical details. The goal of this design is to be able describe the main data entities, attribute and relationship.

4.2.1.1 Entity Relationship Diagram

The entity relationship diagram (ERD) for the system consists of eight tables. Each table has a relationship between them for data efficiency. The ERD is use as the guidance for the development of the database.

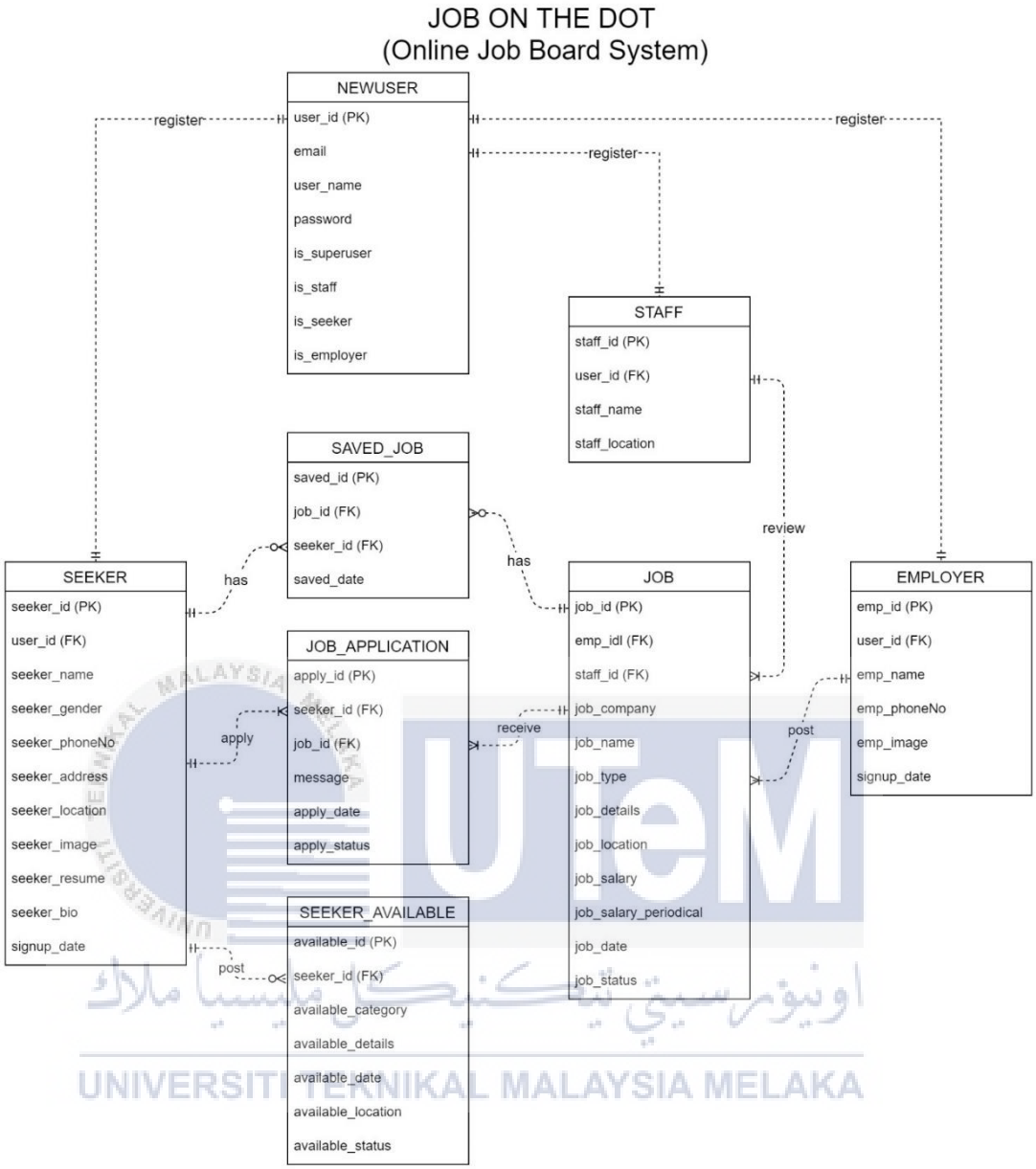

Figure 4.1: Entity Relationship Diagram (ERD)

4.2.1.2 Business Rule

The business rule based on the entity relationship diagram (ERD) of the system as follows:

i. A new user can be assigned only one role which are either seeker, employer or staff.

ii. Seeker can post one or many availabilities. One availability post is posted by only one job seeker.

iii. Employer can post one or many jobs opening. One job opening can only be posted or update by only one employer.

iv. Job Seeker can save one or many jobs opening. A save job can only contain one job seeker and one job opening. A job opening can be saved one or many times.

v. A staff can approve or reject one or many jobs opening application. A job opening application can be updated by only one staff.

vi. A job seeker can apply one or many jobs application. Each job application only contains one job seeker and one job opening. Each job opening can receive one or many jobs application.

vii. A job seeker cannot save or apply for a job opening if it is closed.

4.2.2 Logical Design

The relationship with local logical data will be extracted, and normalization will be utilized to check the relationship, in the second step of the database building approach. Validation of relationships with user transactions is also possible. At this step, the integrity restrictions that apply to attributes and entities are described.

4.2.2.1 Data Dictionary

A data dictionary is used in the system to help maintain the data consistencies across the system. It also uses to define conventions and data that are used in the system. The tables below shows all the data dictionary for each table based on the ERD.

Table 4.1: Data Dictionary for NEWUSER

NEWUSER						
No	Name	Description	Data Type	Required	PK/FK	Reference Table
1	User_id	Unique id for user	int	YES	PK	
2	Email	User email	varchar	YES		

3	User_name	Username	varchar	YES		
4	Password	Password	varchar	YES		
5	Is_superuser	For admin role	Bool	YES		
6	Is_staff	For staff role	Bool	YES		
7	Is_seeker	For job seeker role	Bool	YES		
8	Is_employer	For employer role	bool	YES		

Table 4.2: Data Dictionary for STAFF

STAFF						
No	Name	Description	Data Type	Required	PK/FK	Reference Table
1	staff_id	Unique id for staff profile	int	YES	PK	
2	User_id	Unique id for user	int	YES	FK	NEWUSER
3	Staff_name	Name of staff	varchar	YES		
4	Staff_location	Staff assigned location	varchar	YES		

Table 4.3: Data Dictionary for EMPLOYER

EMPLOYER						
No	Name	Description	Data Type	Required	PK/FK	Reference Table

1	emp_id	Unique id for employer profile	int	YES	PK	
2	User_id	Unique id for user	int	YES	FK	NEWUSER
3	emp_name	Name of employer	varchar	YES		
4	Emp_phoneNo	Employer phone number	varchar	YES		
5	Emp_image	Profile image of employer	varchar	NO		
6	Signup_date	Date of employer create account	date	YES		

Table 4.4: Data Dictionary for SEEKER

SEEKER						
No	Name	Description	Data Type	Required	PK/FK	Reference Table
1	seeker_id	Unique id for seeker profile	int	YES	PK	
2	User_id	Unique id for user	int	YES	FK	NEWUSER
3	seeker_name	Name of seeker	varchar	YES		
4	seeker_phoneNo	Seeker phone number	varchar	YES		

5	Seeker_gender	Gender of seeker	varchar	YES		
6	Seeker_address	Address of seeker home	longtext	YES		
7	Seeker_location	Seeker's location	varchar	YES		
8	Seeker_image	Profile image of seeker	varchar	NO		
9	Seeker_resume	Resume of seeker	varchar	NO		
10	Seeker_bio	short biographical seeker	varchar	NO		
11	Signup_date	Date of seeker create account	date	YES		

Table 4.5: Data Dictionary for SEEKER_AVAILABLE

SEEKER_AVAILABLE						
No	Name	Description	Data Type	Required	PK/FK	Reference Table
1	Available_id	Unique id for available	Int	YES	PK	
2	Seeker_id	Unique id for seeker profile	int	YES	FK	SEEKER
3	Available_category	Category for available	varchar	YES		

4	Available_details	Details of the availability	longtext	YES		
5	Available_date	Start date of availability	date	YES		
6	Available_location	Location of availability	varchar	YES		
7	Available_status	Available status	varchar	YES		

Table 4.6: Data Dictionary for JOB

JOB						
No	Name	Description	Data Type	Required	PK/ FK	Reference Table
1	Job_id	Unique id for job	Int	YES	PK	
2	Emp_id	Unique id for employer profile	int	YES	FK	EMPLOYER
3	Staff_id	Unique id for staff profile	int	NO	FK	STAFF
4	Job_company	Name of the company	varchar	YES		
5	Job_name	Name of the job	varchar	YES		
6	Job_type	Type of the job hiring	varchar	YES		
7	Job_details	Details of the job	longtext	YES		
8	Job_location	Location of the job	varchar	YES		

9	Job_salary	Salary for the job	Decimal	YES		
10	Job_salary_periodical	Type of salary payment	varchar	YES		
11	Job_date	Date posted	date	YES		
12	Job_status	Status of the job	varchar	YES		

Table 4.7: Data Dictionary for SAVED_JOB

SAVED_JOB						
No	Name	Description	Data Type	Required	PK/ FK	Reference Table
1	Saved_id	Unique save job id	Int	YES	PK	
2	Job_id	Unique id for job	Int	YES	FK	JOB
3	Seeker_id	Unique id for available	int	YES	FK	SEEKER
4	Saved_date	Date of saved job	date	YES		

Table 4.8: Data Dictionary for JOB_APPLICATION

JOB_APPLICATION						
No	Name	Description	Data Type	Required	PK/ FK	Reference Table
1	apply_id	Unique job application id	int	YES	PK	
2	Job_id	Unique id for job	int	YES	FK	JOB

3	Seeker_id	Unique id for available	int	YES	FK	SEEKER
4	message	Message to tell	varchar	NO		
5	Apply_date	Date of application	date	YES		
6	Apply_status	Application status	varchar	YES		

4.2.3 Physical Design

The physical database design is the last step in the database design process. The database must be enforced throughout this phase. Transform the data from the logical design step into a description of the physical database, including tables and constraints, during this phase. Everyone should know what database management system (DBMS) is being utilized at this stage. Distinct DBMS, for example, have different names for data types and data types. The logical data model is then transformed into a set of DDL statements that describe the database.

4.2.3.1 Data Definition Language (DDL)

The data definition language (DDL) for each of the table is created using the SQL. The DDL allow the system to be created in the database.

(a) Create table on PostgreSQL

i. Table NewUser

```
CREATE TABLE customuser_newuser
(
    id        integer        NOT        NULL        DEFAULT
      nextval('customuser_newuser_id_seq'::regclass),
    password character varying(128) NOT NULL,
    is_superuser boolean NOT NULL,
    email character varying(254) NOT NULL,
```

```
    user_name character varying(150) NOT NULL,
    is_staff boolean NOT NULL,
    is_employer boolean NOT NULL,
    is_seeker boolean NOT NULL,
    CONSTRAINT customuser_newuser_pkey PRIMARY KEY (id),
    CONSTRAINT    customuser_newuser_email_key    UNIQUE
      (email),
    CONSTRAINT   customuser_newuser_user_name_key   UNIQUE
      (user_name)
)
```

ii. Table Staff

```
CREATE TABLE customuser_staff
(
    id       integer      NOT      NULL      DEFAULT
     nextval('customuser_staff_id_seq'::regclass),
    staff_name character varying(255),
    staff_location character varying(20) NOT NULL,
    user_id integer NOT NULL,
    CONSTRAINT customuser_staff_pkey PRIMARY KEY (id),
    CONSTRAINT    customuser_staff_user_id_key    UNIQUE
      (user_id),
    CONSTRAINT
      customuser_staff_user_id_41603aa8_fk_customuser_new
      user_id FOREIGN KEY (user_id)
        REFERENCES public.customuser_newuser (id)
)
```

iii. Table Employer

```
CREATE TABLE employer_employer_profile
(
```

```
    id       integer       NOT       NULL       DEFAULT
      nextval('employer_employer_profile_id_seq'::regclas
      s),
    "fullName" character varying(255),
    "phoneNo" character varying(11),
    image character varying(100) NOT NULL,
    user_id integer NOT NULL,
    signup_date date NOT NULL,
    CONSTRAINT employer_employer_profile_pkey PRIMARY KEY
      (id),
    CONSTRAINT      employer_employer_profile_user_id_key
      UNIQUE (user_id),
    CONSTRAINT
      employer_employer_pr_user_id_1b83c326_fk_customuse
      FOREIGN KEY (user_id)
       REFERENCES public.customuser_newuser (id)
)
```

iv. Table Seeker

```
CREATE TABLE seeker_seeker_profile
(
    id       integer       NOT       NULL       DEFAULT
      nextval('seeker_seeker_profile_id_seq'::regclass),
    "fullName" character varying(255),
    gender character varying(10),
    "phoneNo" character varying(11),
    address text,
    location character varying(100),
    bio character varying(100),
    resume character varying(100),
    image character varying(100) NOT NULL,
    user_id integer NOT NULL,
    signup_date date NOT NULL,
```

```
    CONSTRAINT  seeker_seeker_profile_pkey  PRIMARY  KEY
      (id),
    CONSTRAINT  seeker_seeker_profile_user_id_key  UNIQUE
      (user_id),
    CONSTRAINT
      seeker_seeker_profile_user_id_48ae0cc5_fk_customuse
      r_newuser_id FOREIGN KEY (user_id)
        REFERENCES public.customuser_newuser (id)
)
```

v. Table Seeker_Available

```
CREATE TABLE seeker_seeker_available
(
    id       integer       NOT       NULL       DEFAULT
      nextval('seeker_seeker_available_id_seq'::regclass)
      ,
    available_category character varying(100),
    available_details text NOT NULL,
    available_date date NOT NULL,
    available_location character varying(50) NOT NULL,
    available_status character varying(20) NOT NULL,
    freelancer_id integer NOT NULL,
    CONSTRAINT  seeker_seeker_available_pkey  PRIMARY  KEY
      (id),
    CONSTRAINT
      seeker_seeker_availa_freelancer_id_f9999149_fk_seek
      er_se FOREIGN KEY (freelancer_id)
        REFERENCES public.seeker_seeker_profile (id)
)
```

vi. Table Post_Job

```
CREATE TABLE employer_post_job
(
```

```
    id        integer        NOT        NULL        DEFAULT
      nextval('employer_post_job_id_seq'::regclass),
    job_company character varying(100) NOT NULL,
    job_name character varying(50) NOT NULL,
    job_type character varying(50) NOT NULL,
    job_details  text  COLLATE  pg_catalog."default"  NOT
      NULL,
    job_location character varying(20) NOT NULL,
    job_salary numeric(6,2) NOT NULL,
    job_salary_periodical character varying(100) NOT NULL,
    job_date date NOT NULL,
    job_status character varying(20) NOT NULL,
    employer_id integer NOT NULL,
    staff_id_id integer,
    CONSTRAINT employer_post_job_pkey PRIMARY KEY (id),
    CONSTRAINT
      employer_post_job_employer_id_d4795bd1_fk_employer_
      FOREIGN KEY (employer_id)
        REFERENCES public.employer_employer_profile (id),
    CONSTRAINT
      employer_post_job_staff_id_id_db943dc4_fk_customuse
      r_staff_id FOREIGN KEY (staff_id_id)
        REFERENCES public.customuser_staff (id)
)
```

vii. Table Saved_Job

```
CREATE TABLE seeker_saved_job
(
    id        integer        NOT        NULL        DEFAULT
      nextval('seeker_saved_job_id_seq'::regclass),
    saved_date date NOT NULL,
    job_id_id integer NOT NULL,
    seeker_username_id integer NOT NULL,
    CONSTRAINT seeker_saved_job_pkey PRIMARY KEY (id),
```

```
    CONSTRAINT
      seeker_saved_job_job_id_id_d9280c29_fk_employer_pos
      t_job_id FOREIGN KEY (job_id_id)
        REFERENCES public.employer_post_job (id),
    CONSTRAINT
      seeker_saved_job_seeker_username_id_9937aa0f_fk_see
      ker_se FOREIGN KEY (seeker_username_id)
        REFERENCES public.seeker_seeker_profile (id)
)
```

viii. Table Job_Application

```
CREATE TABLE seeker_job_application
(
    id        integer       NOT       NULL        DEFAULT
      nextval('seeker_job_application_id_seq'::regclass),
    message character varying(250),
    apply_date date NOT NULL,
    apply_status character varying(20) NOT NULL,
    job_id_id integer NOT NULL,
    seeker_username_id integer NOT NULL,
    CONSTRAINT  seeker_job_application_pkey  PRIMARY  KEY
      (id),
    CONSTRAINT
      seeker_job_applicati_job_id_id_ac8bdbf5_fk_employer
      _ FOREIGN KEY (job_id_id)
        REFERENCES public.employer_post_job (id),
    CONSTRAINT
      seeker_job_applicati_seeker_username_id_f2a4d64d_fk
      _seeker_se FOREIGN KEY (seeker_username_id)
        REFERENCES public.seeker_seeker_profile (id)
)
```

(b) Trigger on PostgreSQL

Trigger is a condition that automatically check after the data manipulation event had happened on the selected table. The trigger creates based on the users request whether they are performed before or after operations occurred. For this system, the trigger function will delete all the save job when the job opening is closed.

```
CREATE FUNCTION public.delete_save_job()
    RETURNS trigger
    LANGUAGE 'plpgsql'
    COST 100
    VOLATILE NOT LEAKPROOF
AS $BODY$
BEGIN
    if NEW.job_status = 'inactive' then
    delete from seeker_saved_job where job_id_id = NEW.id;
    end if;
    RETURN NEW;
END;
$BODY$;
CREATE TRIGGER close_job
    AFTER UPDATE OF job_status
    ON public.employer_post_job
    FOR EACH ROW
    EXECUTE PROCEDURE public.delete_save_job();
```

(c) Stored Procedure in PostgreSQL

A procedure is a PL/SQL statement that can be named and execute in multiple ways. The stored procedure will accept or not the input parameters which can be used over the network by the users using different parameters. An example of stored procedure used in the system is used when the job seeker wants to add data into table seeker_available.

```
CREATE OR REPLACE PROCEDURE public.insert_availability(
```

```
        category character,
        details text,
        apply_date date,
        locations character,
        seeker integer)
LANGUAGE 'plpgsql'
AS $BODY$
BEGIN
        INSERT INTO seeker_seeker_available (
            available_category,          available_details,
        available_date, available_location,
            available_status, freelancer_id
        ) VALUES (
            category,   details,   apply_date,   locations,
        'active', seeker
        );
END;
$BODY$;
```


4.3 Graphical User Interface (GUI) Design

Here is the user interface of the system of job seeker, employer, staff and admin. The interface of the system will differ as they all have different roles and flow that was stated in Chapter 3.

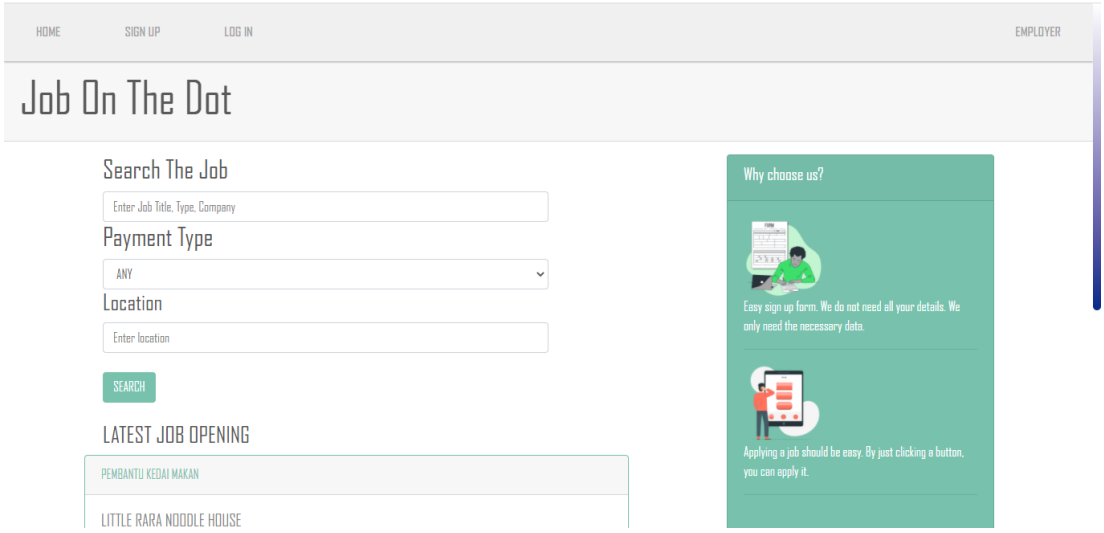

Figure 4.2: shows the landing page of the system

Figure 4.3: shows the sign up form

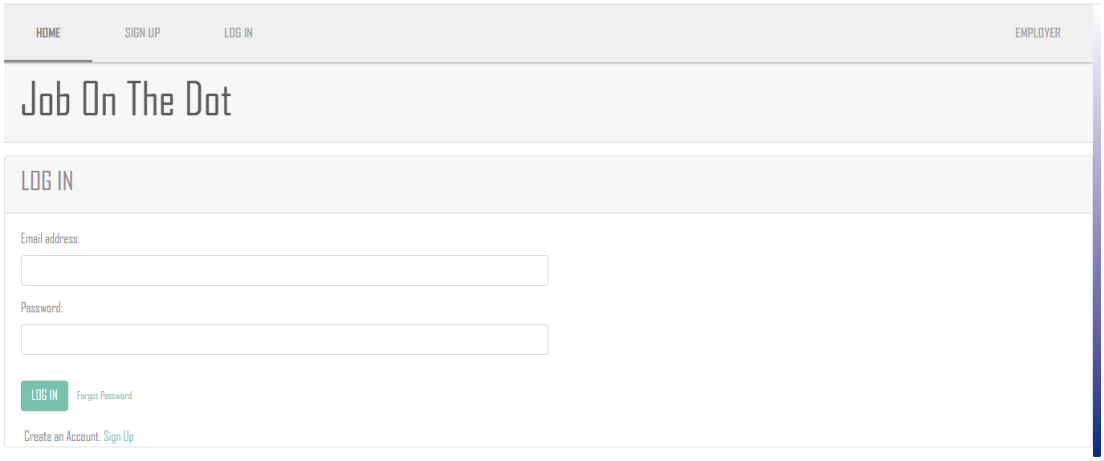

Figure 4.4: shows the login interface

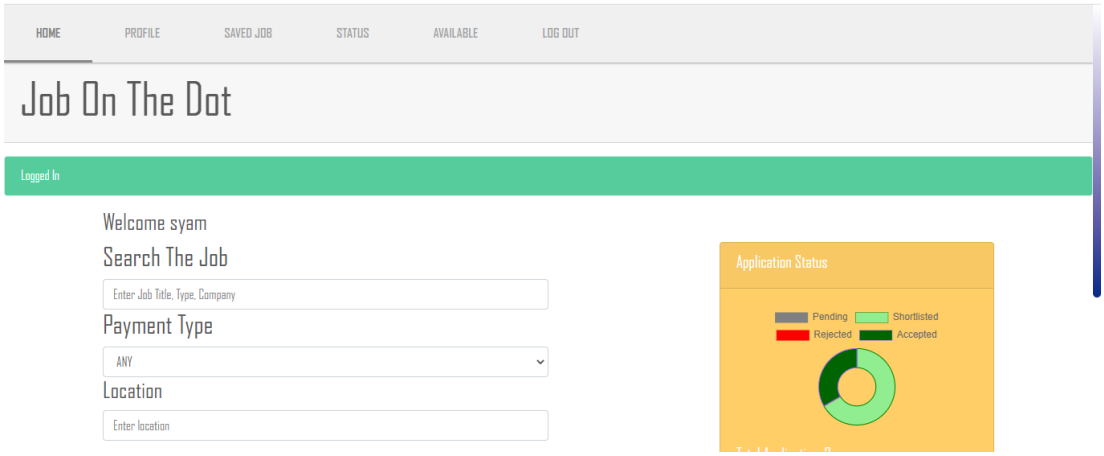

Figure 4.5: shows the homepage for job seeker

Figure 4.6: shows the job opening details that allow job seeker to save or apply the job.

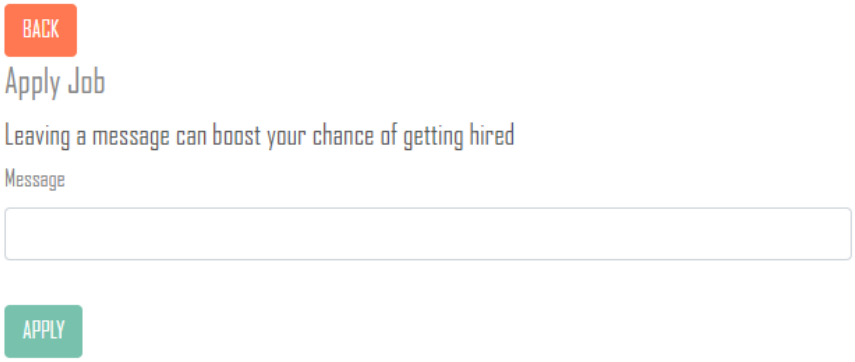

Figure 4.7: shows the job application form for applying the job

Figure 4.8: shows the list of job that job seeker has apply with the status of the application.

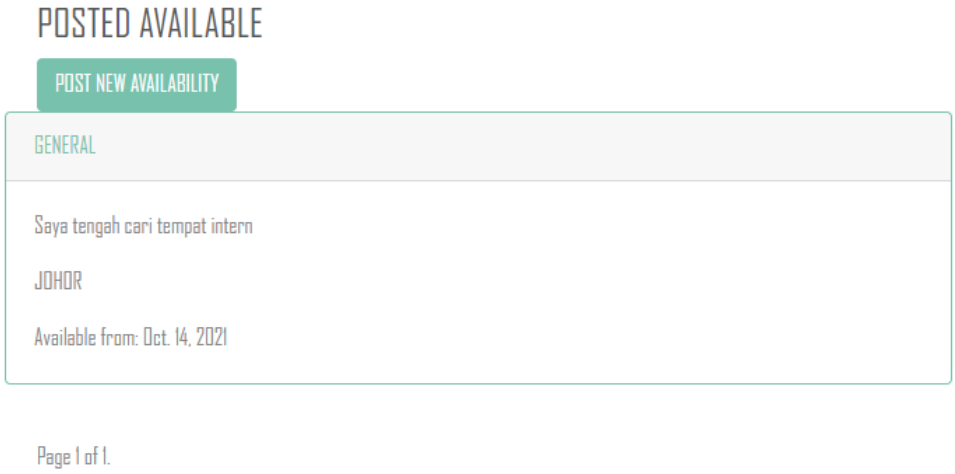

Figure 4.9: shows the available list that job seeker has posted.

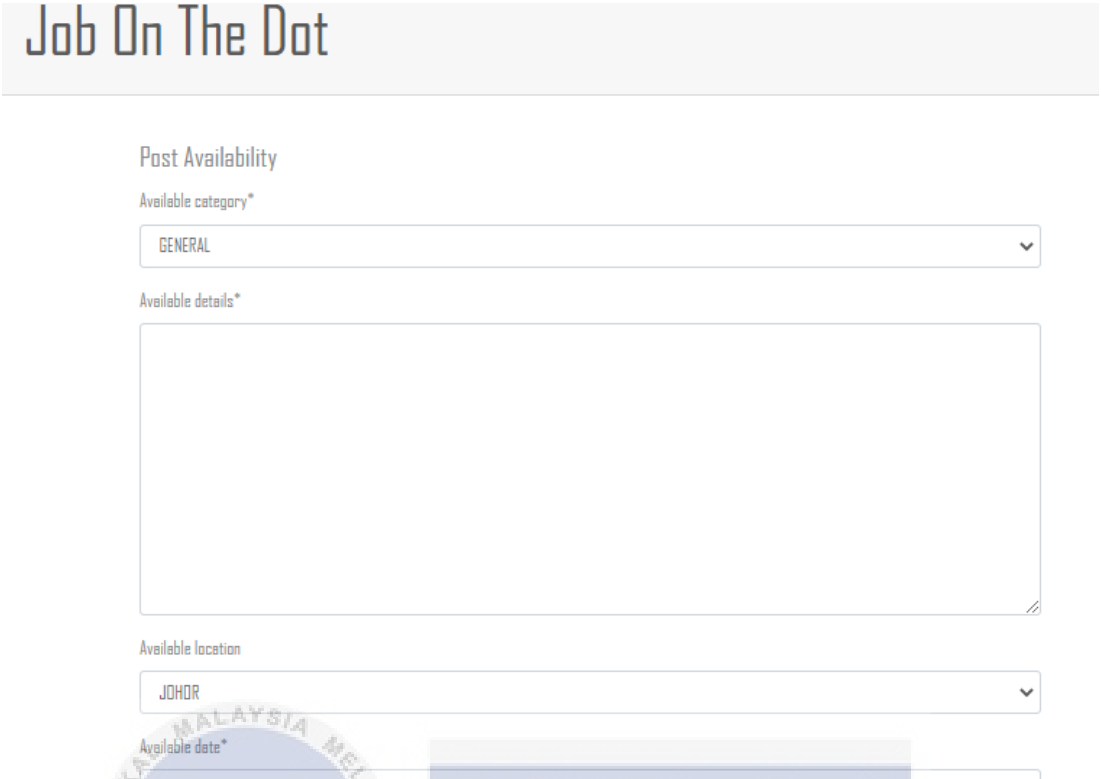

Figure 4.10: shows the available form for job seeker.

Figure 4.11: shows the homepage for employer.

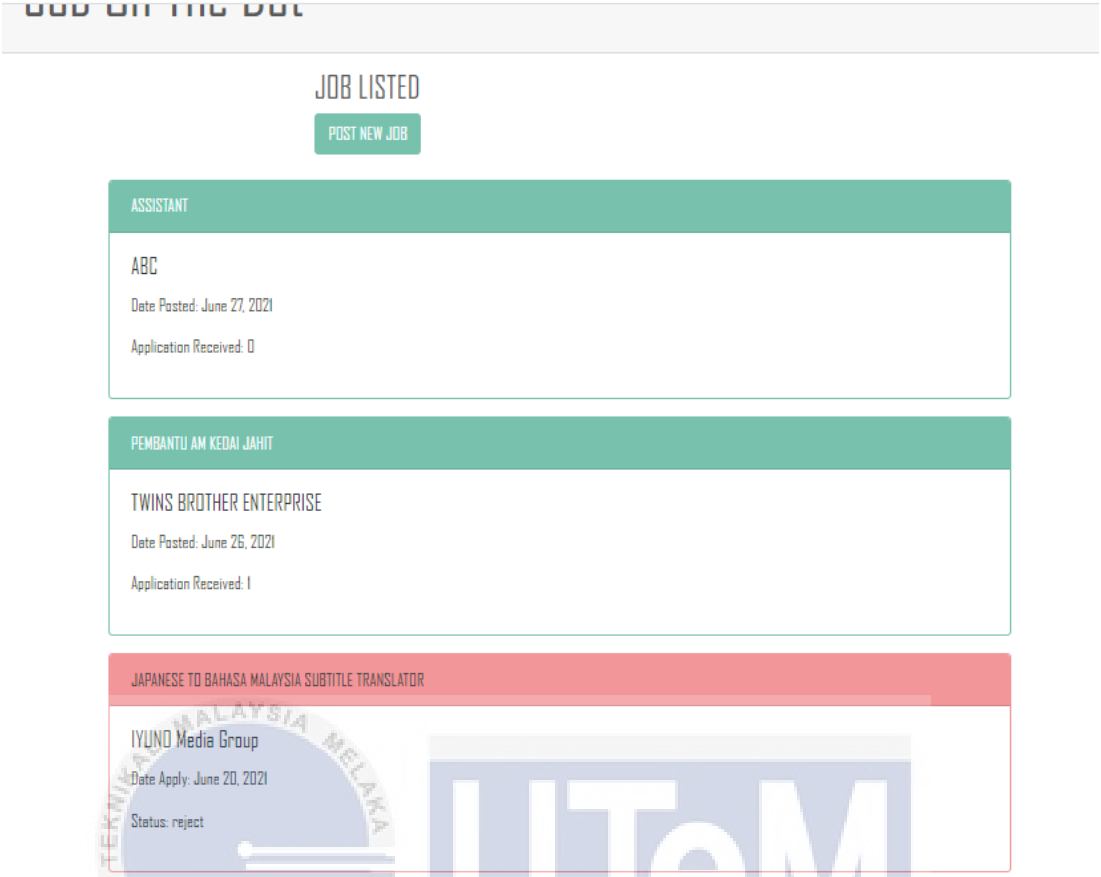

Figure 4.12: shows the list of jobs employer have posted.

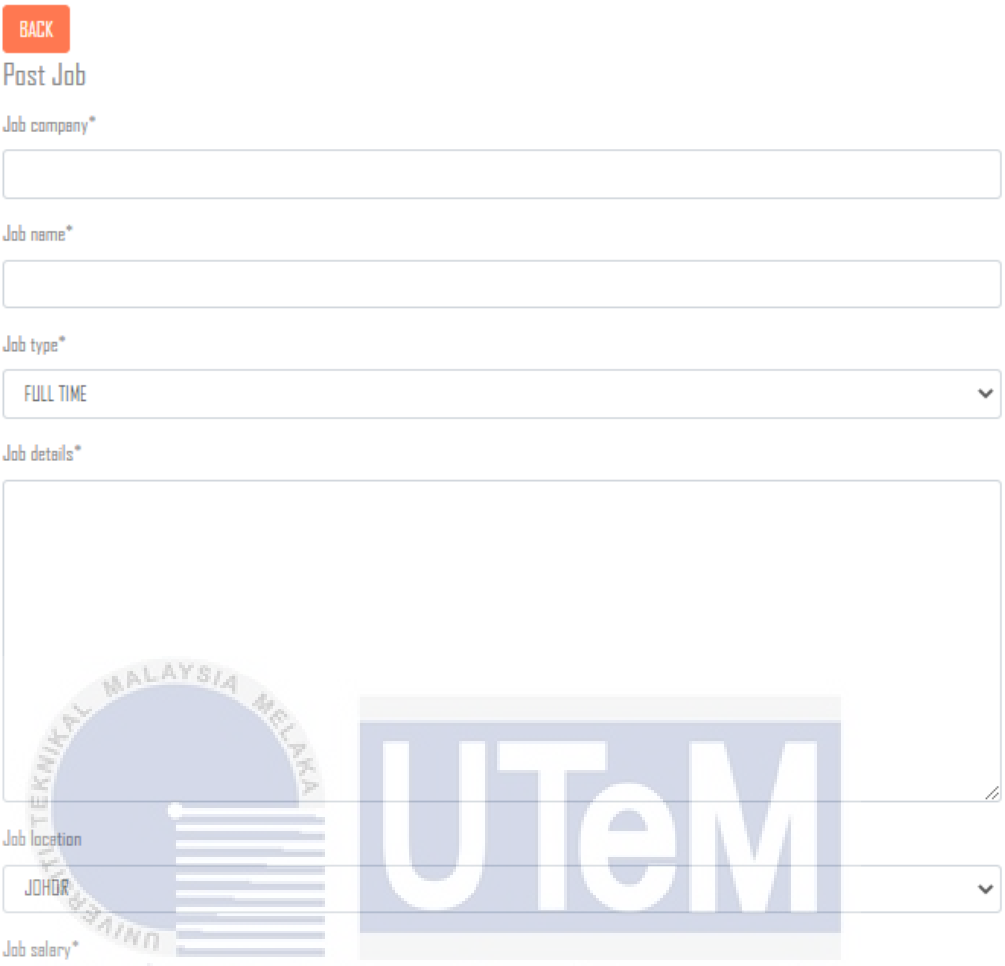

Figure 4.13: shows the job opening form.

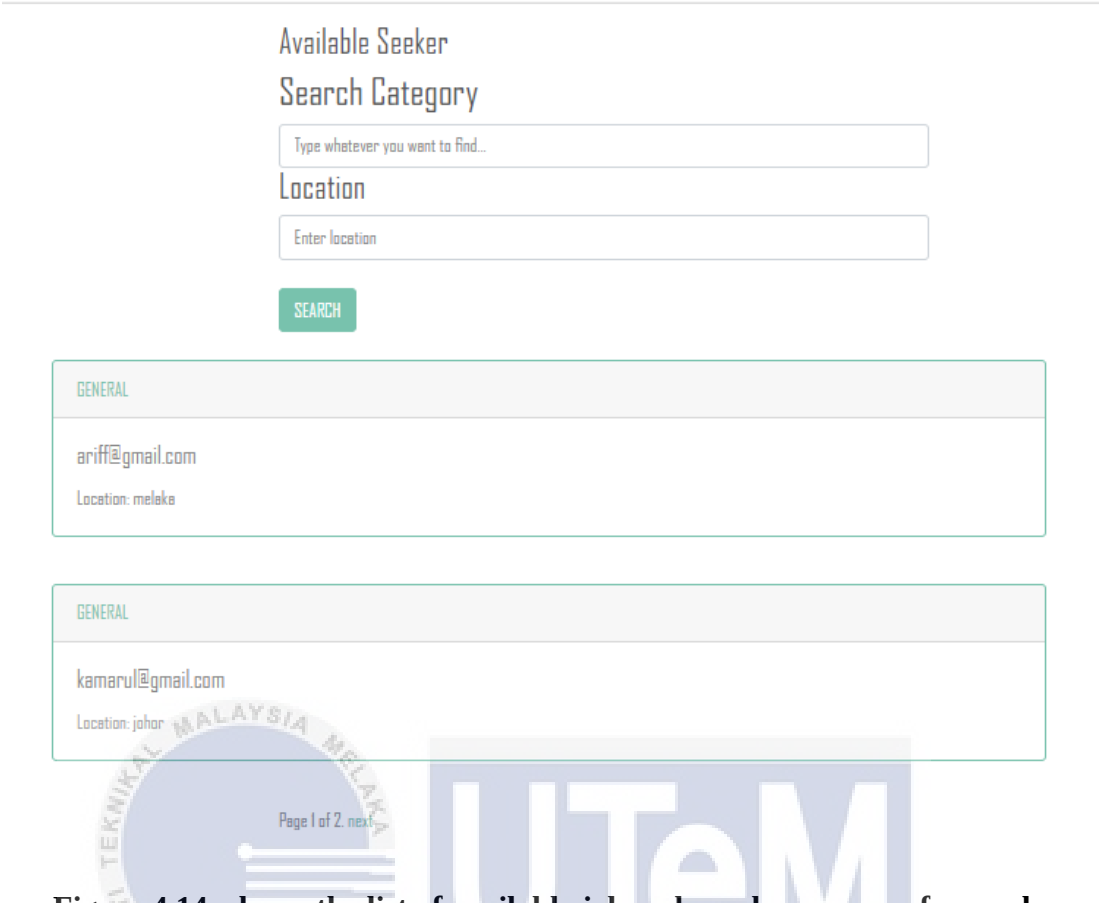

Figure 4.14: shows the list of available job seeker who are open for work.

Figure 4.15: shows the login page for staff and admin.

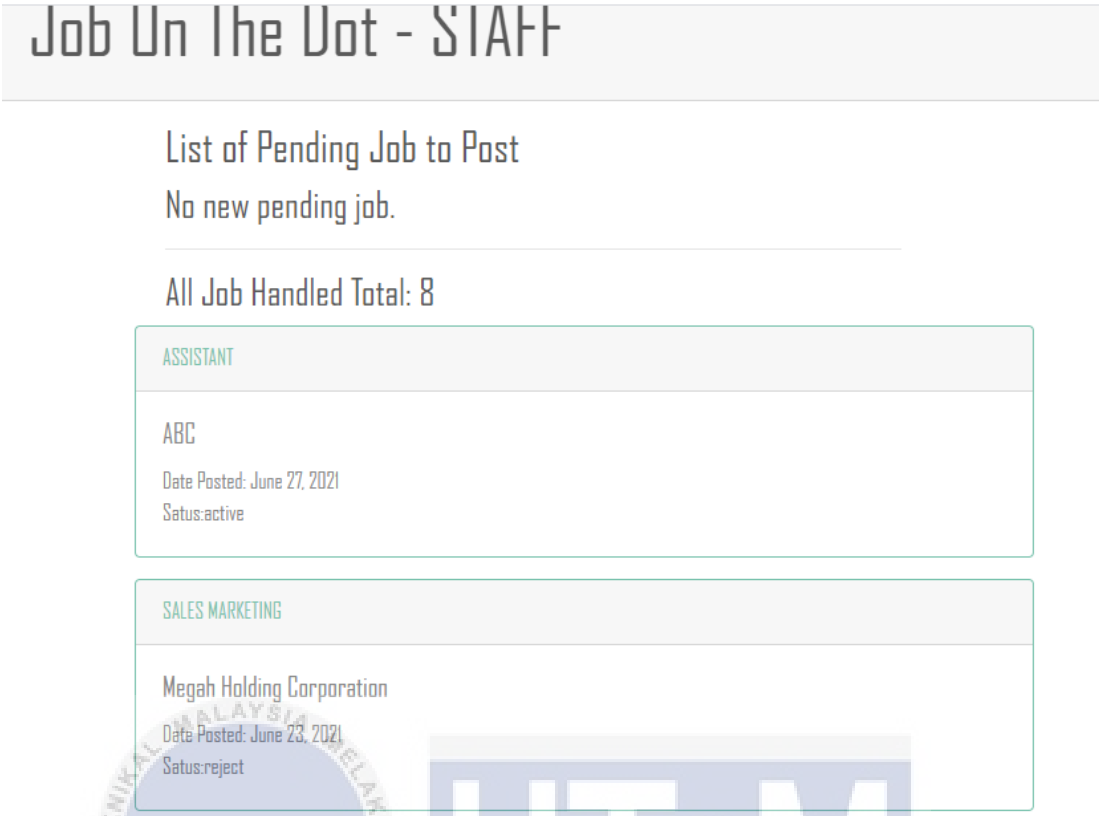

Figure 4.16: shows the list of job opening based on the staff assigned location.

Figure 4.17: shows the job details of the opening that have option to approve or reject

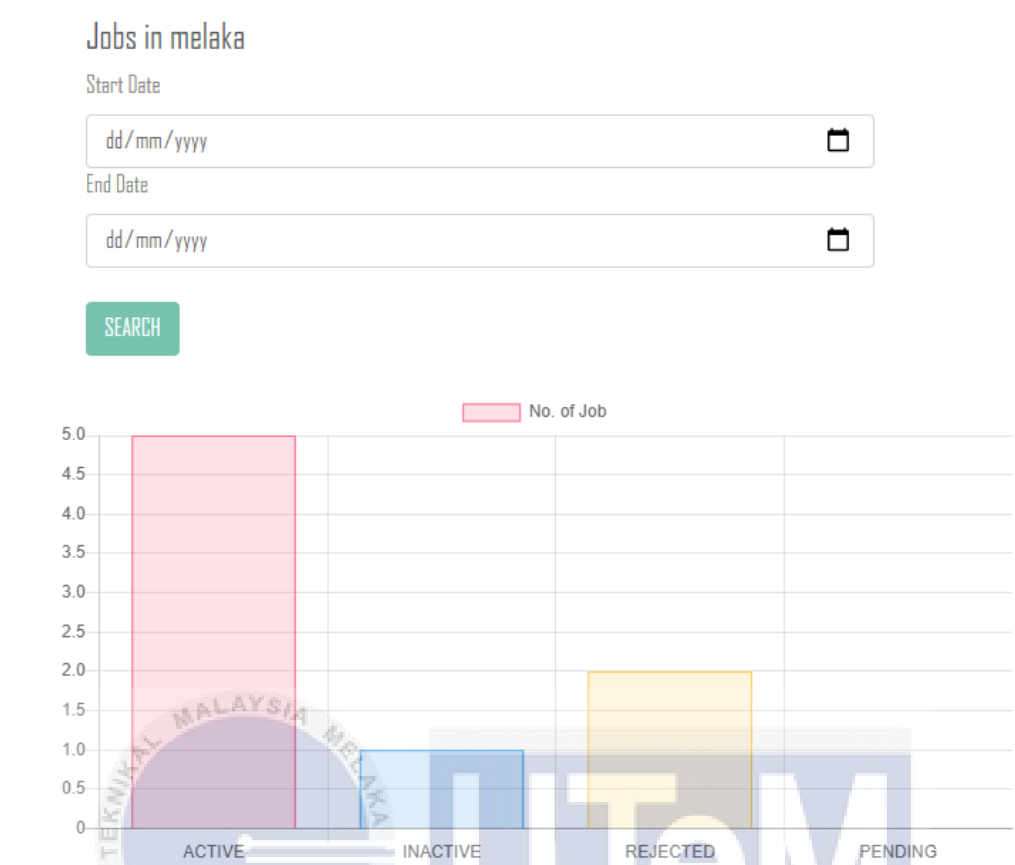

Figure 4.18: shows the report of the job based on the location.

Figure 4.19: shows the list of staff.

4.4 Conclusion

In conclusion, this system met the goal and solved the key problem statement that was studied earlier in this project, however it still has to be improved for better performance. The proposals should be adopted in order to improve the system's reliability and efficiency. Job seekers and employers will benefit the most from this system, which will assist them in all aspects. Overall, this approach will make everyone's job easier, and the quality of work will improve.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

This chapter describes in detail how the software development and system database that were used to construct this project were implemented. The software development environment will be explained by going over the installation process, configuring the program, and running the database server. Furthermore, the database instance as well as the creation of objects such as tables, views, and so on will be described. For database implementation, it includes PostgreSQL Database, Data Definition Language (DDL), Data Manipulation Language, and core processes such as stored procedures and triggers, all of which are implemented using the PLSQL programming language.

5.2 Software Development Environment Setup

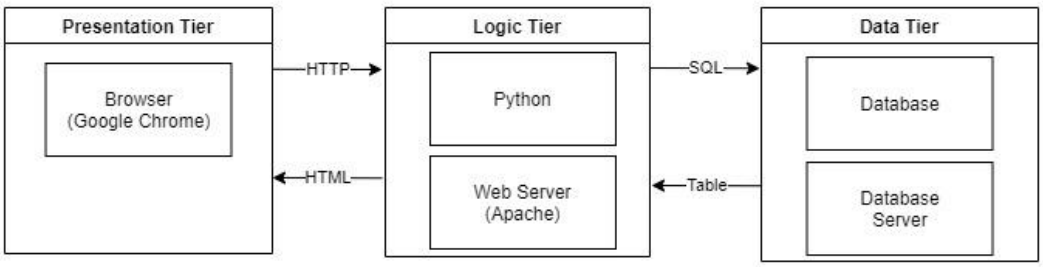

Figure 5.1 : Three-Tier System Architecture

Figure 5.1 shows the three-tier system architecture used to develop the system. Three-tier architecture is an architectural deployment paradigm that divides features into tiers, with each section acting as a layer that may be deployed on its own device. The software is divided into three tiers: presentation, logic, and data in this architecture. As a result, the presentation layer represents the client device and software that communicates with the application, such as Google Chrome. This layer contains the application user interface, which was created to make client-system interaction more efficient.

The logic tier, on the other hand, is in responsible for managing the application's functioning through in-depth analysis. It refers to a programming language as well as a server that allows a user to connect to a web server. The programming language use to build the system is Python programming language and using the Django framework. The Django architecture have three parts for web development which are Template, View and Model.

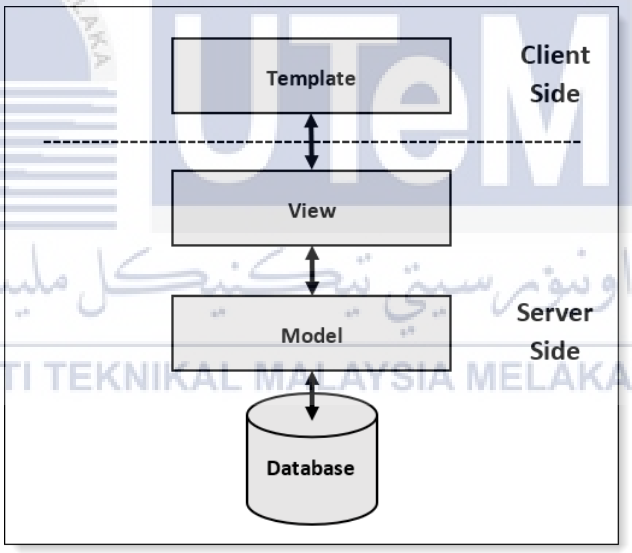

Figure 5.2: Django Architecture

Finally, there is the data tier, which includes the database server, which is the real DBMS access layer. The data in the tables is stored at this layer in a database server, such as a PostgreSQL database.

5.2.1 Software Development Plan

Visual Studio Code is being utilized as a platform for creating Python programming code throughout the development process. The project uses a

PostgreSQL database to store data and database objects as a database management platform.

5.2.1.1 Software Development Setup Python and Django

In order for the system to develop, Python programming language must be installed to the system. After installing Python, the Django framework will also need to be install.

Step 1: Go to https://www.python.org/downloads/ and download the Python installer.

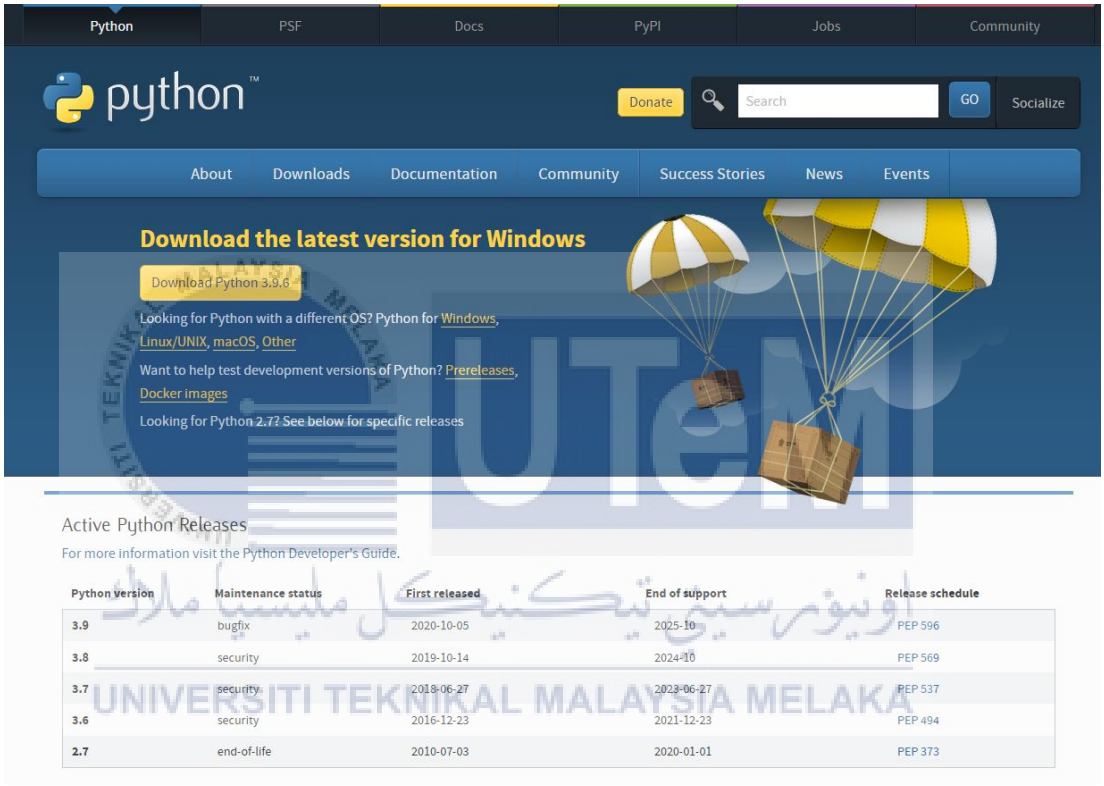

Figure 5.3: Python website

Step 2: Launch the installer. Tick Add Pyhton 3.7 to PATH and then click Install Now.

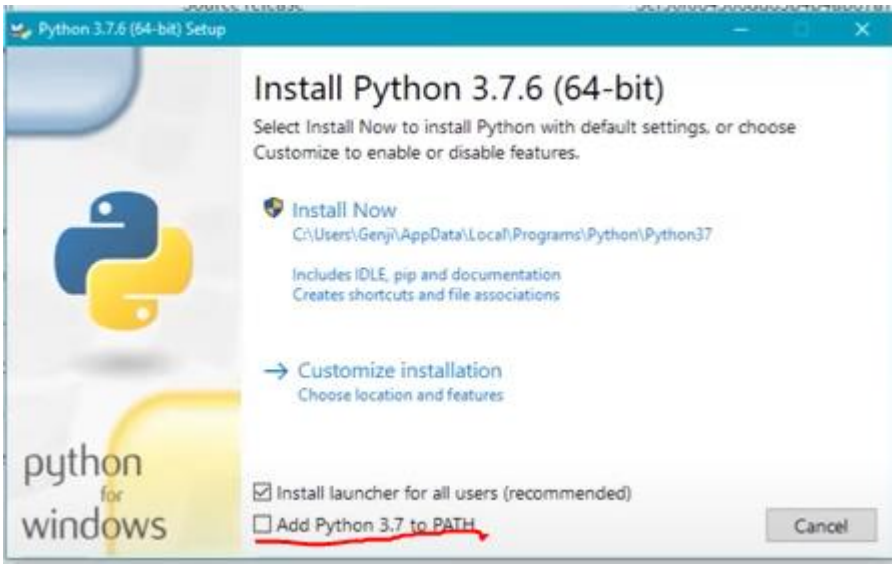

Figure 5.4: Python installer

Step 3: After finish installing to the system, close the installer.

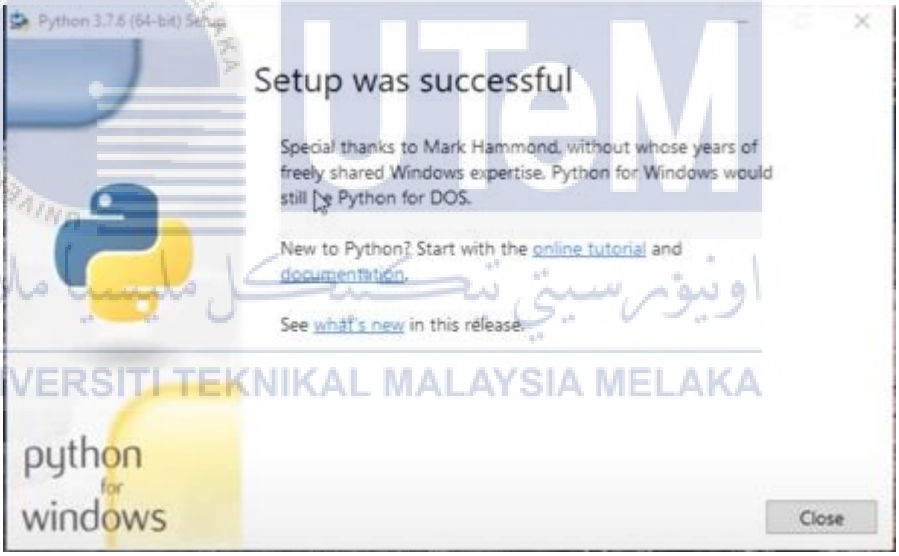

Figure 5.5: Python installation successful

Step 4: Open command prompt, type “pip install Django”. This is to install Django to the system.

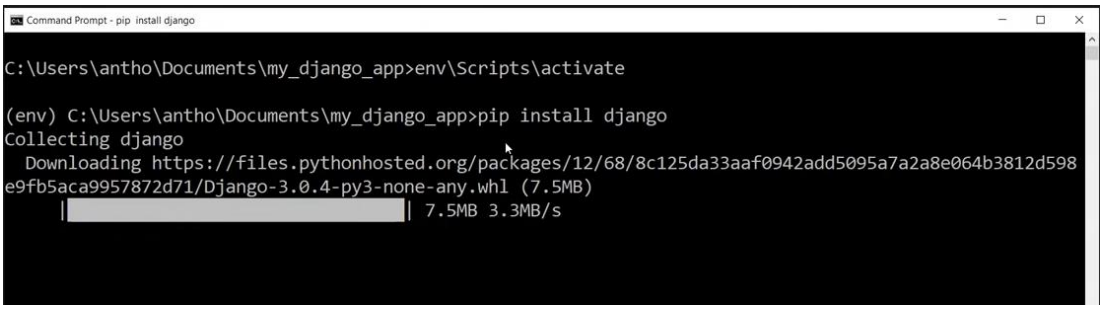

Figure 5.6: Django installation

5.2.1.2 Software Development Setup Database Server

This system uses the PostgreSQL database to store the data. PostgreSQL will need to be install to the system.

Step 1: Go to https://www.postgresql.org/download/ and select the operating system which is Windows. Download the installer packages.

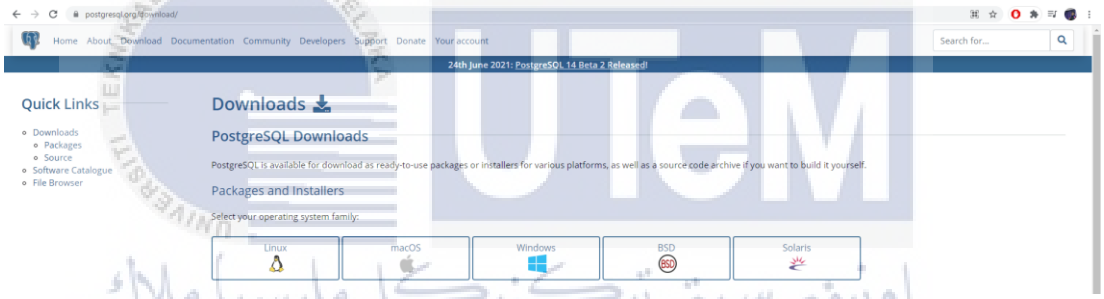

Figure 5.7: PostgreSQL website

Step 2: Open the installer and click next.

Figure 5.8: PostgreSQL installer

Step 3: Set the installation directory and then click next.

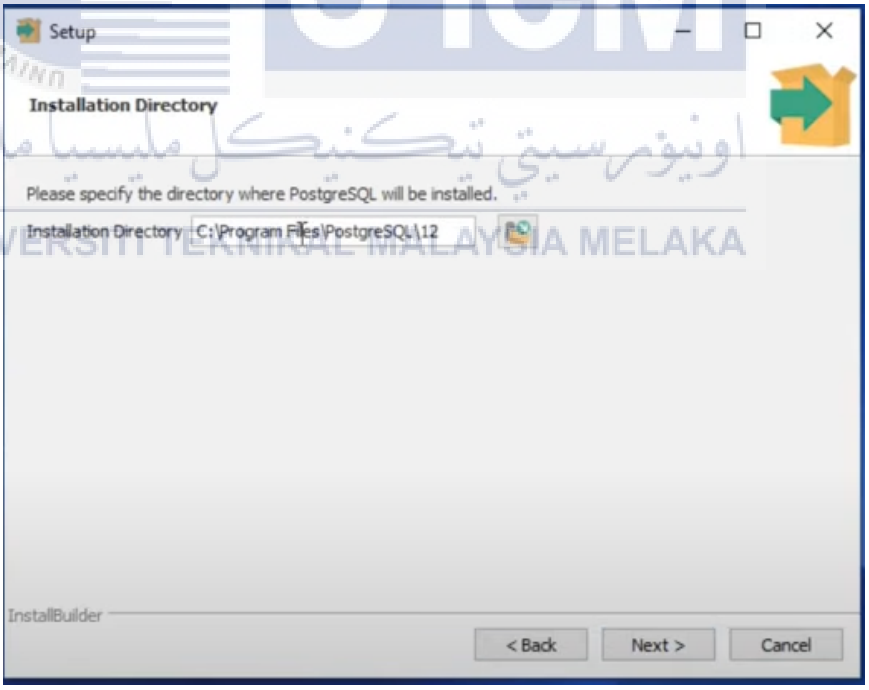

Figure 5.9: PostgreSQL installation directory

Step 4: Tick all the components and click next.

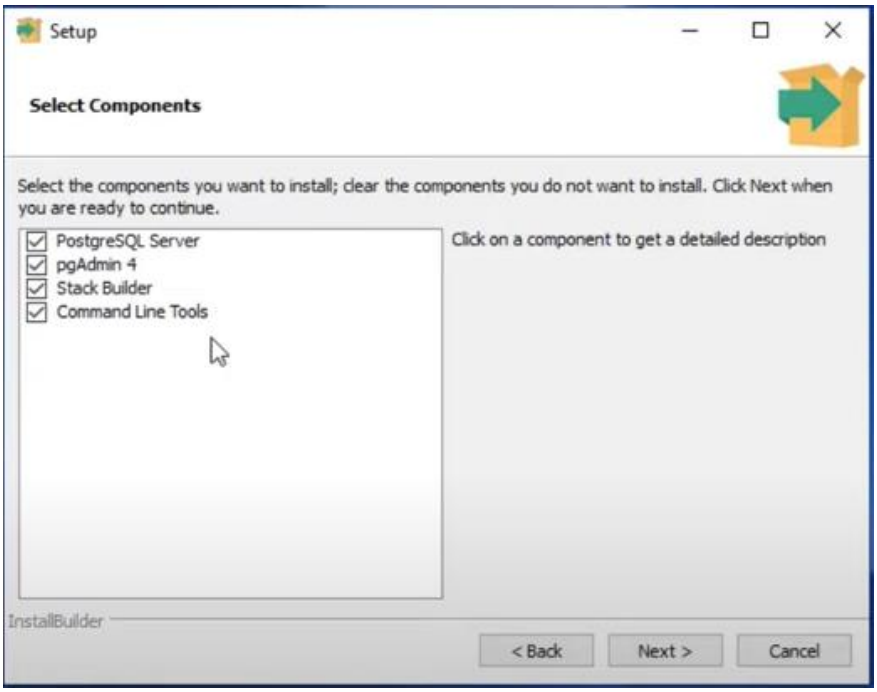

Figure 5.10: PostgreSQL component installation

Step 5: Set the data directory for where the data will be stored and click next.

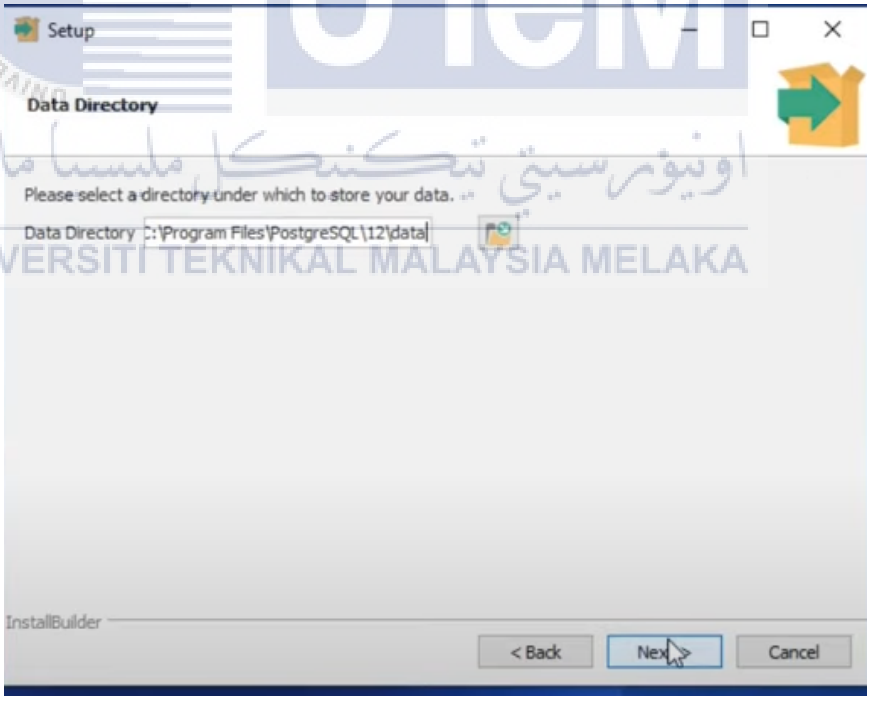

Figure 5.11: PostgreSQL set the data directory

Step 6: After the installation is complete, click finish.

Figure 5.12: PostgreSQL installation successful

5.3 Database Implementation

In order to demonstrate the functioning of this system, information regarding the data definition language, triggers, and stored procedures will be shown in detail in the database implementation. A database stores data in a more organized and efficient manner.

5.3.1 Data Definition Language (DDL)

To begin with, the Data Definition Language was largely used by database administrators during the construction phase of the database to design the database and tables inside it.

5.3.1.1 Database Creation

A database first needs to be created in order to create a database table and database object for the system. The name of the database for the system is “jotd” which is the name of the system.

```
CREATE DATABASE jotd;
```

5.3.1.2 Table Creation

Once the database has been created successfully, the database table will be created next. In order to create a table, the SQL statement that is use 'CREATE TABLE' command followed by the table attribute name, data type, field length and constraint such as primary key and foreign key.

```
CREATE TABLE customuser_staff
(
    id        integer        NOT        NULL        DEFAULT
      nextval('customuser_staff_id_seq'::regclass),
    staff_name character varying(255),
    staff_location character varying(20) NOT NULL,
    user_id integer NOT NULL,
    CONSTRAINT customuser_staff_pkey PRIMARY KEY (id),
    CONSTRAINT    customuser_staff_user_id_key    UNIQUE
      (user_id),
    CONSTRAINT
      customuser_staff_user_id_41603aa8_fk_customuser_new
      user_id FOREIGN KEY (user_id)
        REFERENCES customuser_newuser (id)
)
```

Figure 5.13: Table creation for staff

5.3.1.3 Trigger Creation

Triggers are special stored procedures that run automatically when a database server event happens. Triggers are created in response to a user's request to conduct actions before or after an event has happened. When a user attempts to alter data via a data manipulation language (DML) event, DML triggers are triggered.

In this system, there are several triggers that have been implemented successfully. Each of these triggers plays an important role to ensure that the system runs properly.

Table 5.1: Types of triggers implemented

Type of Trigger	Table	Usage of Trigger
Before Insert	newuser	To check if the email is already existing in the system. If it already exists, the new user cannot be register
Before Update	post_job	To prevent the employer to update the job posted description when the job status is active.
After Update	post_job	To remove the job posted in the job seeker saved job when the job posted is closed.

Figure 5.14 shows the SQL statement for the trigger check_email in the database. This trigger will execute a function to prevent user for creating an account with an existing email in the database. This trigger will be executed before insert data into table newuser.

```
CREATE FUNCTION public.validate_staff_email()
    RETURNS trigger
    LANGUAGE 'plpgsql'
    COST 100
    VOLATILE NOT LEAKPROOF
AS $BODY$
BEGIN
     PERFORM email FROM customuser_newuser WHERE email =
NEW.email;
     IF FOUND THEN
     RAISE EXCEPTION 'New user with this Email address
already exists.';
     ELSE
     RETURN NEW;
     END IF;
END;
$BODY$;
CREATE TRIGGER check_email
    BEFORE INSERT
```

```
    ON public.customuser_newuser
    FOR EACH ROW
    EXECUTE FUNCTION public.validate_staff_email();
```

Figure 5.14: Create before insert trigger on newuser

Figure 5.15 shows the SQL statement for the trigger check_update_detail in the database. This trigger will execute a function to prevent employer from updating their job posted which is in active status. This trigger will be execute before updating the table post_job.

```
CREATE FUNCTION public.bf_update_activejob()
    RETURNS trigger
    LANGUAGE 'plpgsql'
    COST 100
    VOLATILE NOT LEAKPROOF
AS $BODY$
BEGIN
     IF NEW.job_status = 'active' THEN
     RAISE EXCEPTION 'Active Job Cannot be Update!';
     ELSE
     RETURN NEW;
     END IF;
END;
$BODY$;
CREATE TRIGGER check_update_detail
    BEFORE UPDATE OF job_company, job_name, job_type,
job_details,          job_location,          job_salary,
job_salary_periodical
    ON public.employer_post_job
    FOR EACH ROW
    EXECUTE FUNCTION public.bf_update_activejob();
```


Figure 5.15: Create before update trigger on job

Figure 5.16 shows the SQL statement for the trigger close_job in the database. This trigger will execute a function to delete the saved job from the job seeker when a job posted is closed. This trigger will be executed after update the table post_job.

```
CREATE FUNCTION public.delete_save_job()
    RETURNS trigger
    LANGUAGE 'plpgsql'
    COST 100
    VOLATILE NOT LEAKPROOF
AS $BODY$
BEGIN
    if NEW.job_status = 'inactive' then
    delete from seeker_saved_job where job_id_id = NEW.id;
    end if;
    RETURN NEW;
END;
$BODY$;
CREATE TRIGGER close_job
    AFTER UPDATE OF job_status
    ON public.employer_post_job
    FOR EACH ROW
    EXECUTE FUNCTION public.delete_save_job();
```

Figure 5.16: Create after update trigger on job

5.3.1.4 Stored Procedure Creation

A stored method is a collection of SQL statements with the same name that are saved together in a relational database management system to allow multiple functions to be duplicated and transferred. A procedure is a type of PL/SQL statement that may be labelled and run in a variety of ways. A stored process will accept or reject input parameters, and the user can execute it over the network with various parameters.

A few types of stored procedures have been built in the Job on the Dot system for this project. The stored procedure insert_availability is used to insert job seeker availability into seeker available table in the database. Job seeker able to post availability by using the procedure. The insert_savejob procedure is used for the job seeker to save a job. When the job seeker click save job, the job will be insert into the saved job table. Table shows the stored procedure used with its query.

Table 5.2: Type of Procedure Implemented

Procedure Name	Table	Query

insert_availability	Seeker_available	CREATE OR REPLACE PROCEDURE insert_availability(
		category character,
		details text,
		apply_date date,
		locations character,
		seeker integer)
		LANGUAGE 'plpgsql'
		AS $BODY$
		BEGIN
		INSERT INTO seeker_seeker_available (
		available_category, available_details, available_date, available_location, available_status, freelancer_id
		) VALUES (
		category, details, apply_date, locations, 'active', seeker
		);
		END;
		$BODY$;

insert_savejob	Saved_job	CREATE OR REPLACE PROCEDURE insert_savejob(job_id integer, seeker integer) LANGUAGE 'plpgsql'

		AS $BODY$
		BEGIN
		INSERT INTO seeker_saved_job (saved_date, job_id_id, seeker_username_id)
		VALUES (LOCALTIMESTAMP, job_id, seeker);
		END;
		$BODY$;

5.4 Conclusion

In conclusion, the implementation phase is the last stage before moving on to the testing phase, during which the developer is no longer active. The Apache server is used as the server-side programming language, while PostgreSQL is used as the database. A database is a logically connected collection of information or files that are aggregated into a common pool and offer data to one or more users.

CHAPTER 6: TESTING

6.1 Introduction

Before the system is given to users, the final part of the system development life cycle is testing. During this phase, the system is carefully tested, and any flaws discovered will be fixed by the assigned developers. Life-cycle system development testing may be accomplished in a variety of ways. The testing methodologies employed vary based on the model of system development, the stage of the process, and the test procedure's objectives. The purpose of this testing is to assess the system's capacity and determine whether it satisfies all the criteria.

Testing has been carried out on the JOTD System in order to verify and confirm that the system meets the user's needs. This chapter will cover test organizations, which will introduce the tester, test environment, which will go over the installation of the system setup requirements, and test schedule, which will establish the number of cycles and duration of the test to be run. A testing strategy will detail the method used in performing the testing, whereas a test design will describe the testing conducted for each module. Finally, genuine data entered into the system will be used for test data and analysis results in test results and analysis.

6.2 Test Plan

The test plan is a thorough document that focuses on the strategy for the upcoming testing activity. Subtests such as test organization, test timeline, and test environment are included in the test plan. To begin, the testing organization is made up of specialist users who are responsible for ensuring that the system meets a set of standards. Second, the test environment will consist of the hardware, operating system (OS), and software required to run the test. Finally, the test schedule will include the various cycles and deadlines that must be met.

6.2.1 Test Organization

A test organization establishes who is in charge of which operations during the testing process. The test functions, test facilities, and test activities are all described in the organization. It establishes the level of knowledge and experience of the individuals involved. The system was tested by a small group of people, including a system developer, a software tester, and an end user. The system developer is the JOTD System developer. A software tester is someone who has expertise testing and executing user acceptance tests. There will be three end user which are job seeker and employer who will be benefit from using the system while also the staff for the system who has experience in managing and filtering out job advertisement.

Table 6.1 shows the test organization table, which includes information about the individuals engaged in the test stages. They will be in charge of managing, verifying, and performing the test.

Table 6.1: Test Organization Table

Tester ID	Roles	Responsibilities
Tester1	System Developer	Responsible to develop, implement, and test the integration and testing components.
Tester2	Software Tester	Take on the role of an end user, such as an administrator, staff, employer and job seeker. Module for system testing.

		Provide feedback to help the system improve.
Tester3	End User	Responsible in testing as end user which is job seeker.
Tester4	End User	Responsible in testing as end user which is employer.
Tester5	End User	Responsible in testing as end user which is staff.

6.2.2 Test Environment

A test environment is a test setting that takes into consideration test parameters including hardware specifications and software configurations. The test environment is critical in ensuring that the system is evaluated constructively and, more than likely, realistically. Tables 6.2 outline the system software and hardware equipment used by the JOTD System.

Table 6.2: System Software and Hardware

Environment Specification	Description
Operating System	Windows 10 The operating system use to manage system and hardware of the laptop.
Central Processing Unit (CPU)	AMD Quad-Core A8-7410 Use for processing and instruction of a computer program
Random Access Memory (RAM)	4GB and above Use by the central processing unit (CPU) to execute the task.
Database	PostgreSQL Use to store data into table form, that runs on a server

System/Programming Language	Javascript, Python Programming Language, Hypertext Markup Language (HTML) and Cascading Style Sheets (CSS)

6.2.3 Test Schedule

The test activities, start and end dates, as well as the duration, should all be included in the test schedule. It might also describe how the test will be examined, tracked, and accepted. To establish the test schedule, the information obtained for the test plan body is combined with the available resource. During the development of the project, the schedule will help the developer in executing the test exactly over a set period of time.

Table 6.3 shows a test schedule for admin, table 6.4 shows a test schedule for staff, tables 6.5 shows a test schedule for employer and table 6.6 shows a test schedule for job seeker.

Table 6.3: Test Schedule for Admin

Testing Module	Test Activity	Start Date	End Date	Duration
Login (User Authentication)	Unit testing, integration, testing and error handling test.	19 July 2021	19 July 2021	1 Day
Staff Handling	Unit testing, integration, testing and error handling test.	20 July 2021	21 July 2021	2 Days
Job	Unit testing, integration, testing and error handling test.	22 July 2021	22 July 2021	1 Day
Employer	Unit testing, integration, testing and error handling test.	23 July 2021	23 July 2021	1 Day

Seeker	Unit testing, integration, testing and error handling test.	24 July 2021	24 July 2021	1 Day

Table 6.4: Test Schedule for Staff

Testing Module	Test Activity	Start Date	End Date	Duration
Login (User Authentication)	Unit testing, integration, testing and error handling test.	28 July 2021	28 July 2021	1 Day
Job Handling	Unit testing, integration, testing and error handling test.	29 July 2021	30 July 2021	2 Days

Table 6.5: Test Schedule for Employer

Testing Module	Test Activity	Start Date	End Date	Duration
User Registration	Unit testing, integration, testing and error handling test.	2 August 2021	3 August 2021	2 Days
Login (User Authentication)	Unit testing, integration, testing and error handling test.	4 August 2021	4 August 2021	1 Day
Profile	Unit testing, integration, testing and error handling test.	5 August 2021	5 August 2021	1 Day
Job Listed	Unit testing, integration, testing and error handling test.	6 August 2021	8 August 2021	3 Days
Application	Unit testing, integration, testing and error handling test.	9 August 2021	10 August 2021	2 Days

Find Worker	Unit testing, integration, testing and error handling test.	11 August 2021	11 August 2021	1 Day

Table 6.6: Test Schedule for Job Seeker

Testing Module	Test Activity	Start Date	End Date	Duration
User Registration	Unit testing, integration, testing and error handling test.	12 August 2021	13 August 2021	2 Days
Login (User Authentication)	Unit testing, integration, testing and error handling test.	14 August 2021	14 August 2021	1 Day
Profile	Unit testing, integration, testing and error handling test.	15 August 2021	15 August 2021	1 Day
Search Job	Unit testing, integration, testing and error handling test.	16 August 2021	18 August 2021	3 Days
Save Job	Unit testing, integration, testing and error handling test.	19 August 2021	20 August 2021	2 Days
Apply Job	Unit testing, integration, testing and error handling test.	21 August 2021	23 August 2021	3 Days
Available	Unit testing, integration, testing and error handling test.	24 August 2021	25 August 2021	2 Days

6.3 Test Strategy

A test strategy is a plan that describes the test design and specifies how the testing should be carried out. Because the test strategy is simply a subset of the test plan, it is separate from the test plan. White-box testing and black-box testing are the two types of dynamic testing.

White-box testing, also known as structural testing, is a software testing technique that examines the product's fundamental structure, design, and code. It took place in order to determine the system's operating flow and to improve design, usability, and security. Internal knowledge of developing systems and programming abilities are required for this type of testing.

Black-box testing, also known as functional testing, approaches the system under test as a black-box, with the input and output being the only things that are tested. The software interface may be used to test functionality and ensure that it works as anticipated. If internal changes occur but the functionality stays unaltered, the test is judged successful.

6.3.1 Classes of Test

There are many different types of tests that may be utilized during the testing process. In this example, two types of tests were selected: an error handling test and an integration test.

I. Error Handling Test

Error handling testing is a type of software testing that ensures a system's ability to resolve problems that may arise in the future. It will verify that the data provided by the users is correct and valid. Error warnings will appear on the screen to notify the user that part of their input may be incorrect. Essentially, testing is done with the assistance of both developers and testers.

II. Integration Test

Integration testing reveals problems in the interplay of integrated elements. It was to verify that the data input into the system was accurately recorded in the database. To ensure that the process of storing data is successful, the system must interface properly with the database.

6.4 Test Design

The number of testing techniques that will be required, the test criteria, and how the test will be approached are all part of the test design. For each test module that has been created and documented, a test design includes test case identification, test cases, and expected results. In terms of test data, it is necessary to choose either real-life or simulated data.

6.4.1 Test Description

The test case's design and documentation, as well as the intended outcome, are usually included in the test description. The test case is a written list of requirements, procedures, and expected outcomes that the tester uses to ensure that the system satisfies the user's needs. Unit testing and integration testing are the two forms of testing that have been done. Unit testing is a technique of testing that verifies the smallest piece of testable code or each module in a system, while integration testing is a form of testing that verifies the integration between the modules to ensure that they operate correctly.

Table 6.7: Test Description of Unit Testing for Login Module

Test Case ID	Description	Testing Type	Expected Result
TC001_01	Valid email and correct password	Error handling and integration testing	Successful logged in and redirect to homepage.

TC001_02	Invalid email	Error handling test	Unsuccessful login attempt. Display error message.
TC001_03	Valid email and incorrect password	Error handling test	Unsuccessful login attempt. Display error message.

Table 6.8: Test Description of Unit Testing for Staff Handling Module in Admin

Test Case ID	Description	Testing Type	Expected Result
TC002_01	Display list of staff that has been register to the system.	Error handling and integration testing	All staff that has registered into the system will be on the list.
TC002_02	Valid email, username, password, staff name and location with no blanks.	Error handling and integration testing	Successful register of new staff to the system.
TC002_03	Invalid or blank email, username, password, staff name and location	Error handling and integration testing	Unsuccessful registration of new staff to the system.

Table 6.9: Test Description of Unit Testing for Job Module in Admin

Test Case ID	Description	Testing Type	Expected Result
TC003_01	Display list of job that has been added to the system.	Error handling and integration testing	All job that has added into the system will be on the list.
TC003_02	To check search functionality in table	Error handling and integration testing	Table searching work properly and admin

			able to search for job listed in system

Table 6.10: Test Description of Unit Testing for Employer Module in Admin

Test Case ID	Description	Testing Type	Expected Result
TC004_01	Display list of employers that has been registered to the system.	Error handling and integration testing	All employer that has registered into the system will be on the list.
TC004_02	To check search functionality in table	Error handling and integration testing	Table searching work properly and admin able to search for employers in system

Table 6.11: Test Description of Unit Testing for Seeker Module in Admin

Test Case ID	Description	Testing Type	Expected Result
TC005_01	Display list of job seekers that has been registered to the system.	Error handling and integration testing	All job seeker that has registered into the system will be on the list.
TC005_02	To check search functionality in table	Error handling and integration testing	Table searching work properly and admin able to search for job seekers in system

Table 6.12: Test Description of Unit Testing for Job Handling Module in Staff

Test Case ID	Description	Testing Type	Expected Result
TC006_01	Display list of pending job and job handled based on staff location	Error handling and integration testing	All job seeker that has registered into the system will be on the list.
TC006_02	To check button 'REJECT' functionality for response in job pending	Error handling and integration testing	Button work properly and staff able to update the job status to reject.
TC006_03	To check button 'APPROVE' functionality for response in job pending	Error handling and integration testing	Button work properly and staff able to update the job status to active.

Table 6.13: Test Description of Unit Testing for User Registration Module in Employer and Job Seeker

Test Case ID	Description	Testing Type	Expected Result
TC007_01	Valid email, username and password with no blanks.	Error handling and integration testing	Successful register of user to the system.
TC007_02	Invalid or blank email, username and password	Error handling and integration testing	Unsuccessful registration of new user to the system and display error message

Table 6.14: Test Description of Unit Testing for Profile Module in Employer and Job Seeker

Test Case ID	Description	Testing Type	Expected Result
TC008_01	To check the 'Update' button functionality	Error handling and integration testing	Button work properly and employer able to update their profile.

Table 6.15: Test Description of Unit Testing for Job Listed Module in Employer

Test Case ID	Description	Testing Type	Expected Result
TC009_01	No blank in Job company, Job name, Job type, Job details, Job location, Job salary and job salary periodical	Error handling and integration testing	Successful apply for job listing in pending status to the system.
TC009_02	Have blank in either Job company, Job name, Job type, Job details, Job location, Job salary and job salary periodical	Error handling and integration testing	Unsuccessful job listing to the system and display error message.
TC009_03	To check the 'CLOSED' button functionality	Error handling and integration testing	Button worked properly and employers able to change job status to closed.

Table 6.16: Test Description of Unit Testing for Application Module in Employer

Test Case ID	Description	Testing Type	Expected Result
TC010_01	To check the 'RESUME' button functionality	Error handling and	Button work properly and employer able to

		integration testing	view job seeker resume
TC010_02	To check the 'ANSWER' button functionality	Error handling and integration testing	Button work properly and employer able to update their profile.
TC010_03	To check the 'DOWNLOAD' button functionality	Error handling and integration testing	Button work properly and employer able to update their profile.

Table 6.17: Test Description of Unit Testing for Find Worker Module in Employer

Test Case ID	Description	Testing Type	Expected Result
TC011_01	To check the search functionality to find worker.	Error handling and integration testing	Search work properly and able to show available worker

Table 6.18: Test Description of Unit Testing for Search Job Module in Job Seeker

Test Case ID	Description	Testing Type	Expected Result
TC012_01	To check the search functionality to find job.	Error handling and integration testing	Search work properly and able to show the desired job.

Table 6.19: Test Description of Unit Testing for Save Job Module in Job Seeker

Test Case ID	Description	Testing Type	Expected Result
TC013_01	To check the 'SAVE JOB' button functionality	Error handling and integration testing	Button work properly and job seeker can save the job.
TC013_02	To check the 'REMOVE JOB' button functionality	Error handling and integration testing	Button work properly and job seeker can remove the job from saved job.

Table 6.20: Test Description of Unit Testing for Apply Job Module in Job Seeker

Test Case ID	Description	Testing Type	Expected Result
TC014_01	To check the 'APPLY JOB' button functionality	Error handling and integration testing	Button work properly and page will redirect to application page.
TC014_02	Leave blank in message when apply.	Error handling and integration testing	Successfully apply for the job.
TC014_03	Type in message when apply.	Error handling and integration testing	Successfully apply for the job.

Table 6.21: Test Description of Unit Testing for Available Module in Job Seeker

Test Case ID	Description	Testing Type	Expected Result
TC015_01	Valid available date, category, details and location with no blanks	Error handling and integration testing	Successful post availability into the system.
TC015_02	Enter past date in available date	Error handling and integration testing	Unsuccessful post availability and display error
TC015_03	To check the 'UPDATE' button functionality	Error handling and integration testing	Button worked properly and job seeker able to update the availability.

6.4.2 Test Data

Test data is information that has been designated for use in a test, with the expectation that the particular set of input will yield a certain result. Validation, testing, and verification of the software's behavior using actual data given by the user are all part of the test data. Each test data set is based on the test description, which can be found using the test case ID.

Table 6.22 displays the Login module test data which includes the test case ID, test data and test data result.

Table 6.22: Test Data of Login Module

Test Case ID	Test Data	Test Data Result
TC001_01	Email: faris.firo@gmail.com Password: Faris101	User logged in successfully
TC001_02	Email: faris.firo Password: Faris101	Enter a valid email address.

TC001_03	Email: faris.firo@gmail.com Password: 123456789	Invalid Login

Table 6.23 depicts Staff Handling Module test data, which includes the test case ID, test data, and test data result. Tester2 is used to collect the data entered in a test.

Table 6.23: Test Data of Staff Handling Module in Admin to Tester2

Test Case ID	Test Data	Test Data Result
TC002_01	Staff details	List of all staff that has registered to the system
TC002_02	Email: staff_10@jotd.com User name: staff_sabah Password: Sabah101 Staff Name: Oyenze Staff Location: Sabah	Account created successfully.
TC002_03	Email: User name: staff_sabah Password: Sabah101 Staff Name: Oyenze Staff Location: Sabah	Fail to register. Please fill out all the field

Table 6.24 depicts Job Module test data, which includes the test case ID, test data, and test data result. Tester2 is used to collect the data entered in a test.

Table 6.24: Test Data of Job Module in Admin to Tester2

Test Case ID	Test Data	Test Data Result
TC003_01	Job listed details	List of all job that has apply to the system.
TC003_02	User enters ‘Johor’ as a keyword. Keyword: johor	All jobs that contains the keyword ‘johor’ will be listed in the table.

Table 6.25 depicts Employer Module test data, which includes the test case ID, test data, and test data result. Tester2 is used to collect the data entered in a test.

Table 6.25: Test Data of Employer Module in Admin to Tester2

Test Case ID	Test Data	Test Data Result
TC004_01	Employer details	List of all employer that has register to the system.
TC004_02	User enters 'ali' as a keyword. Keyword: ali	All employers that contains the keyword 'ali' will be listed in the table.

Table 6.26 depicts Seeker Module test data, which includes the test case ID, test data, and test data result. Tester2 is used to collect the data entered in a test.

Table 6.26: Test Data of Seeker Module in Admin to Tester2

Test Case ID	Test Data	Test Data Result
TC005_01	Job seeker details	List of all job seekers that has register to the system.
TC005_02	User enters 'Selangor' as a keyword. Keyword: selangor	All job seeker that contains the keyword 'selangor' will be listed in the table.

Table 6.27 depicts Job Handling Module test data, which includes the test case ID, test data, and test data result. Tester5 is used to collect the data entered in a test.

Table 6.27: Test Data of Job Handling in Staff to Tester5

Test Case ID	Test Data	Test Data Result
TC006_01	Job details based on staff location	List of all job listed based on the staff location.

TC006_02	User clicks 'REJECT' button on a job	The job status will be changed to rejected.
TC006_03	User clicks 'APPROVE' button on a job	The job status will be changed to active.

Table 6.28 depicts User Registration Module test data, which includes the test case ID, test data, and test data result. Tester3 is used to collect the data entered in a test.

Table 6.28: Test Data of User Registration Module in Job Seeker to Tester3

Test Case ID	Test Data	Test Data Result
TC007_01	Email: bob@gmail.com Username: AliAfiq Password: Utem2021 Password Confirmation: Utem2021	Account created successfully.
TC007_02	Email: bob@gmail.com Username: AliAfiq Password: Utem2021 Password Confirmation: Utem2020	Fail to register. The two password fields didn't match.

Table 6.29 depicts Profile Module test data, which includes the test case ID, test data, and test data result. Tester3 is used to collect the data entered in a test.

Table 6.29: Test Data of Profile Module in Job Seeker to Tester3

Test Case ID	Test Data	Test Data Result
TC008_01	User update phoneNo	User profile phoneNo updated.

Table 6.30 depicts Job Listed Module test data, which includes the test case ID, test data, and test data result. Tester4 is used to collect the data entered in a test.

Table 6.30: Test Data of Job Listed Module in Employer to Tester4

Test Case ID	Test Data	Test Data Result
TC009_01	Job Company: MABORO MOTOR ENTERPRISE Job Name: Online admin staff Job Type: FULL TIME Job Details: stock keeping, online selling via shopee, lazada, admin work. Expanding new online business line with shopee, Lazada etc Job Location: JOHOR Job Salary: 2000 Job salary periodical: MONTH	Job apply successfully and pending on staff confirmation.
TC009_02	Job Company: MABORO MOTOR ENTERPRISE Job Name: Job Type: FULL TIME Job Details: stock keeping, online selling via shopee, lazada, admin work. Expanding new online business line with shopee, Lazada etc Job Location: JOHOR Job Salary: 2000 Job salary periodical: MONTH	Fail to apply. Please fill out all the field.
TC009_03	User click 'CLOSED' button	The job status change to closed.

Table 6.31 depicts Application Module test data, which includes the test case ID, test data, and test data result. Tester4 is used to collect the data entered in a test.

Table 6.31: Test Data of Application Module in Employer to Tester4

Test Case ID	Test Data	Test Data Result
TC010_01	User clicks 'RESUME' button of applicant	Display the resume of the applicant.
TC010_02	User clicks 'ANSWER' button on applicant	Change the application status of applicant.
TC010_03	User clicks 'DOWNLOAD' button on a job	Automatically download a list of shortlisted applicant to user.

Table 6.32: Test Data of Find Worker Module in Employer to Tester4

Test Case ID	Test Data	Test Data Result
TC011_01	User enters 'programming' as a keyword. Keyword: programming	All available job seeker that contains the keyword 'programming' will be listed.

Table 6.33: Test Data of Search Job Module in Job Seeker to Tester3

Test Case ID	Test Data	Test Data Result
TC012_01	User enters 'part time' as job type. Job Type: part time	All job that is part time job type will be listed.

Table 6.34: Test Data Save Job Module in Job Seeker to Tester3

Test Case ID	Test Data	Test Data Result
TC013_01	User clicks 'SAVE JOB' button on job.	The job will be saved to user and can view on saved job tab.

TC013_02	User clicks 'REMOVE JOB' button on applicant	The job will be removed from the user saved job.

Table 6.35: Test Data of Apply Job Module in Job Seeker to Tester3

Test Case ID	Test Data	Test Data Result
TC014_01	User clicks 'APPLY JOB' button on job.	User will be redirected to apply job form.
TC014_02	User leaves blank in Message.	Successfully apply for the job.
TC014_03	User enters a message. Message: Hi, I would like to apply.	Successfully apply for the job.

Table 6.36: Test Data of Available Module in Job Seeker to Tester3

Test Case ID	Test Data	Test Data Result
TC015_01	Available Category: DELIVERY Available Details: I have a motorcycle and have experience in doing dispatch job before. Available Location: JOHOR Available Date: 08/30/2021	Availability post successfully.
TC015_02	Available Category: DELIVERY Available Details: I have a motorcycle and have experience in doing dispatch job before. Available Location: JOHOR Available Date: 08/30/2020	Fail to post availability. Invalid Date. Date must not be in the past
TC015_03	User updates the available status to inactive and click update.	Update successfully. The available status changed to inactive.

6.5 Test Results and Analysis

This section will provide the results of the system testing that was carried out based on the test data. If the outcome was successful or unsuccessful, the tester will note it in the result field, followed by their test case ID. All test cases were also tested according to the kind of testing, which included unit testing and integration testing.

Table 6.37: JOTD System Test Result

Test Case ID	Tester Identification	Result (Passed or Failed)
Login Module Test Case		
TC001_01	OK	Passed
TC001_02	OK	Passed
TC001_03	OK	Passed
Staff Handling Module Test Case		
TC002_01	OK	Passed
TC002_02	OK	Passed
TC002_03	OK	Passed
Job Module Test Case		
TC003_01	OK	Passed
TC003_02	OK	Passed
Employer Module Test Case		
TC004_01	OK	Passed
TC004_02	OK	Passed
Seeker Module Test Case		
TC005_01	OK	Passed
TC005_02	OK	Passed
Job Handling Module Test Case		
TC006_01	OK	Passed
TC006_02	OK	Passed
TC006_03	OK	Passed
User Registration Module Test Case		
TC007_01	OK	Passed
TC007_02	OK	Passed

Profile Module Test Case		
TC008_01	OK	Passed
Job Listed Module Test Case		
TC009_01	OK	Passed
TC009_02	OK	Passed
TC009_03	OK	Passed
Application Module Test Case		
TC010_01	OK	Passed
TC010_02	OK	Passed
TC010_03	OK	Passed
Find Worker Module Test Case		
TC011_01	OK	Passed
Search Job Module Test Case		
TC012_01	OK	Passed
Save Job Module Test Case		
TC013_01	OK	Passed
TC013_02	OK	Passed
Apply Job Module Test Case		
TC014_01	OK	Passed
TC014_02	OK	Passed
TC014_03	OK	Passed
Available Module Test Case		
TC015_01	OK	Passed
TC015_02	OK	Passed
TC015_03	OK	Passed

6.6 Conclusion

To summarize, defects and mistakes must be discovered during this phase of testing in order to guarantee that the finished product used by consumers includes as few flaws as possible. This phase of testing also guarantees that the user is pleased and that their requirements are met. It will aid in maintaining the system's quality assurance, providing consumers more trust in its usage.

However, one of the primary reasons for the importance of the testing phase is the high maintenance costs. No defects, particularly those that may result in erroneous system outputs, should be tolerated since fixing them afterwards would be prohibitively expensive.

CHAPTER 7: CONCLUSION

7.1 Introduction

The overall conclusion and future prospects of the system, as well as its strengths, flaws, and possibilities for development, will be discussed in this chapter. It's necessary to acknowledge strengths and weaknesses so that appropriate improvisation and recommendations may be made inside the system to reduce the margin of error and make it more user-friendly.

7.2 Observation

After completing this project and having the tester do several tests, the system's strengths and weaknesses are discovered. As a result, all the JOTD System's strengths and weaknesses are listed.

7.2.1 Weaknesses

I. There is no backup and recovery mechanism in the system database, which might result in data loss in the future due to a variety of reasons.

II. The system does not have a direct message feature that allows communication between employer and job seeker as they need to use third party messenger such as WhatsApp and email to communicate.

III. The system does not support video file such as a resume video by the job seeker to show for future employer.

7.2.2 Strengths

I. Employer can easily apply to post a job opening in the system without filling in a complicated form.

II. Job seeker able to post their availability to get more notice from employer who are hiring worker urgently.

III. The system will have only genuine job posting as fake job will be rejected by the system staff who will filter out them.

IV. Job seeker can find job based on the type of payment that they want to receive.

7.3 Propositions form Improvement

There are some suggestions for enhancement that may be applied for long-term usage based on the strengths and weaknesses that have been derived from the observations. The majority of the ideas are made after considering the system's shortcomings. This system is missing a crucial functionality, which is the database backup and recovery procedure. Because this system stores a lot of data, particularly data from job seeker and employer, it will be much better to have the method. Backup and recovery are capable of preventing technical errors from occurring during system operation. There are a number of backup techniques that may be used, including complete or incremental backups that can be scheduled on a regular basis.

Secondly, the system should have a direct message feature that will allow communication between employer and job seeker. This will help the process of hiring much faster and more convenient. Employer can ask the job seeker who post availability easily without using a third-party messenger application.

Lastly, a video support in the system would be much of an improvement. Job seeker can upload their resume video to their profile. This will allow the job seeker chance of getting to hire much higher as it will be able to grab the attention of future employer. A video resumes can help the job seeker to show their enthusiasm and confidence towards the employer.

7.4 Project Contribution

This effort has benefited various parties, including the university, the corporation, and the individual. This project able to contribute to the university in finding new staff. The university can develop and improve the system as this system will be belong to the university.

Besides, for the company, they can post their job opening on the system. This can help the company getting more notice from potential job seeker who have the specialty.

Lastly, the contribution to an individual is that they can find job easier. The individual can also post their availability for future employer to get notice them.

7.5 Conclusion

To conclude, this system achieved its goal and answered the major issue statement that was identified earlier in the project, although it still needs to be improved for better performance. The suggestions suggested should be adopted to improve the system's reliability and efficiency. With JOTD system, job seeker and employer can find and post job easily without a hassle. Overall, this system will make everyone's job easier, and finding of work will be faster.

REFERENCES

Rahim, R. (2020, December 9). Nearly 100,000 Malaysians have lost jobs since start of MCO, says HR ministry. The Star. https://www.thestar.com.my/news/nation/2020/12/09/nearly-100000-malaysians-have-lost-jobs-since-start-of-mco-says-hr-ministry

[Accessed on 20th of March 2021]

Elmers, D. (2020, March 4). *The Job-Search Statistics All Job Seekers Should Know*. TopResume. https://www.topresume.com/career-advice/7-top-job-search-statistics

[Accessed on 20th of March 2021]

Django documentation | Django documentation | Django. (2005). Django. https://docs.djangoproject.com/en/3.2/

[Accessed on 15th of March 2021]

Data, C. (2020). *Building a User Registration Form with Django's Authentication System*. Crunchy Data. https://blog.crunchydata.com/blog/building-a-user-registration-form-with-djangos-built-in-authentication

[Accessed on 25th of March 2021]

Shubham, K. (2021, June 12). *Build a Job Search Portal with Django — Part 1 | Towards Data Science*. Medium. https://towardsdatascience.com/build-a-job-search-portal-with-django-overview-part-1-bec74d3b6f4e?gi=d4f7ca5f186a

[Accessed on 30th of March 2021]

GeeksforGeeks. (2020, August 5). Differences between Black Box Testing vs White Box Testing. https://www.geeksforgeeks.org/differences-between-black-box-testing-vs-white-box-testing/

[Accessed on 20th of August 2021]

Feldmann, J. (2017, May 22). 6 Reasons Why Job Boards Don't Always Work. Undercover Recruiter. https://theundercoverrecruiter.com/job-boards-dont-work/

[Accessed on 10th of August 2021]