2022/2023

“Job / Employment Recommendation System”

[Graduation Project]



FCAI - Helwan university

2022/2023

“Job / Employment Recommendation System”

[Graduation Project]

**-TEAMWORK-**

1. Ahmed Adel El-Sayed Sabry
2. Alaa Mustafa Ahmed Moati
3. Ahmed Sharaf Mohamed Hasan
4. Mayar Sayed Hasan Mahmoud
5. Mohamed Sayed Mohamed Saleh
6. Taher Ahmed Rashad Aweis

Submitted in partial fulfilment of the requirements for the degree of Bachelor of Science in Computers & Artificial Intelligence, at the department of

**I N F O R M A T I O N - S Y S T E M**

Faculty of Computers & Artificial Intelligence, Helwan University

**Supervised by: Associate. Prof. Dr. (Mohamed Marie)**

FCAI - Helwan university



*In this chapter we are going to discuss and go deeper in the overview of the project and know more about its scope and limitations and explain some terminologies we will find throughout the document.*

**(Chapter 1) - Introduction -**

* 1. **- Overview.**
  2. **- Objectives.**
  3. **- Purpose.**
  4. **- Scope.**

**1.5 - Advantages of the system.**

**1.6 - Project Methodology.**

* 1. **- Overview:**

The System is a web-based application system that links the client who wants to accomplish a specific task or work and the performer who seeks to complete the work in the best way.

After completing the work, he receives an evaluation from the performer and the financial compensation he deserves or agreed upon.

Also, with the recommendation system on the site, both job provider and the performer can find each other so fast and easily! Where the system must suggest to the performer the appropriate job provider based on the skills that the performer possesses, which will be registered in the personal page of the performer.

The project must also be of a high degree of Security, especially about financial transactions and personal information of the performer and the job provider.

The project solves a modern problem, which is finding job opportunities. And it makes it easier for users to communicate with each other, so that the provider presents his request and sets its price, as there is no negotiation, only the provider writes the price, and the users who can complete this work will “submit” an application to perform the work and obtain money from the job provider then.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. **- Objectives:**
* Developing a system that provides a straightforward way of communicating job providers and performers.
* The end user got what they needed, or what they wanted.
* The quality, performance, security, or other attributes of what the project made are acceptable to the end user.
* This makes it easier and faster to recruit an employee to finish your work.
* Reduces time wasted searching for jobs or for employees.
* Completing the work in a reasonable amount of time.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. **- Purpose:**

**The site aims to help job providers and performers:**

* Find a new job easily.
* Easy way to connect with the job provider and performer.
* Make the user confident when he chooses anyone to work, by giving it all the information he
* will need is through a CV.
* Reduces time wasted searching for a job.
* This project provides every employee with a specialized recruitment system to passionately help throughout the full hiring journey until the perfect employee is hired.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. **- Scope:**

The scope of this project is to link the job provider and the performer until a certain task is accomplished in return for a fee, which in turn will benefit both parties, so generally the site is the link between the performer and the job provider.

**Out of scope:**

Employment process, payment methods and agreements and deals between the performer and job provider are considered out of the scope of the project

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. **- Advantages of the System:**
* **It is Accessible:**

No matter where you are in the world – if you have an internet-enabled device and connection, you can perform all those management tasks described above. You can modify your job posting, see how many replies you have and even communicate with candidates directly.

#### Easy and Timesaving:

Online job searches are very quick and easy and do not involve hours spent on filling out physical forms and going to the post office. You can also get immediate online feedback from the job advertisers too.

#### Economical:

There is no hassle of printing resumes and sending them by courier to multiple employers. With the full job application process going online, applying to jobs is now more economical too.

* **Elimination of the role of mediators:**

This saves the contracting company from paying additional expenses**.**

* **Effective filtering of candidates:**

E-recruitment websites have facilitated the process of communication between employers and job seekers. Employers can filter and choose the best ones based on their CV**.**

* **It is Flexible:**

The internet gives you plenty of flexibility with regards to controlling your posts and the applications you receive. If you post in a newspaper and want to amend the job advert, though, you will need to pay for an entirely new ad. With online posts, most platforms will allow you to edit, update or remove your job post whenever you wish.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. **– Project Methodology:**
* **Planning:**
* Choosing the appropriate method for data pre-processing.
* Determining the functional and non-functional requirements.
* Setting a Gantt chart for the project
* Determining the resources of the team.
* Prepare a UI\UX prototype.
* Dividing the tasks on team
* **Designing:**
* Determining the diagrams to be carried out within the project:

✓ Activity Diagram

✓ Class Diagram

✓ Use-case Diagram

✓ Sequence Diagram

* **Coding:**
* The main supposed functions to be coded in this App are: -

✓ Register

✓ Log in

✓ Log out

✓ Publish job

✓ Apply job

✓ Accept job request

✓ Edit job

✓ Delete job

✓ Search

✓ View profile

✓ Edit profile

✓ Get notified

✓ Chat

✓ Delete user profile

* **Testing:**
* Functional Testing: -
* Unit testing
* Integration testing
* Additional testing

*In this chapter we’re going to discuss and go deeper in how we planned the project and show the steps and the instructions that we’ve followed to plan the application.*

**(Chapter 2) - Project Planning -**

**2.1 - Feasibility Study.**

* 1. **- Estimated Cost.**
  2. **- Software & Hardware Requirements.**
  3. **– Gantt Chart.**

**2.5 – Constraints**

* 1. **- Feasibility Study:**

**A Feasibility study is used to determine if a business or a specific project is achievable, so for determining the achievability of our project we will go deeper in the following points:**

* **Market Feasibility:**
* Nowadays, people have become lazier to find an offline job, while communication has become quite easy, so people find it easier to work online.
* This application can help these people to find a job faster, just with one click and from their house!
* This application contains job offers -not specialized for some fields- but for many skills a person can have, so people can easily find a suitable job for their own skills.
* **Communication:** The application provides an effortless way of communication between users.
* **Time:** Saves users time -with the suitable recommendation- instead of spending a lot of time searching for an appropriate job.
* **Operational Feasibility:**
* In our project we provide some rules and services to our employees such as: -
  + 1. Environment suitable for work.
    2. Commitment to the deadline for delivering tasks.
    3. Good salaries.
* We plan to support and retain our customers by Providing them with all the services that can be provided, following them up, and being important as a companion to his companion.
* **Technical Feasibility:**
* Technical feasibility involves study to establish the technical capability of the system being created to accomplish all requirements to the user.
* The system should be capable of handling the proposed volume of data and provide users and operating environment to increase their efficiency.
* **Accessibility:** Users can access the application anytime and there will be no time constraints.
* **Economic Feasibility:**
* Economic feasibility involves study to establish the cost benefit analysis.
* Money spent on the system must be recorded in the form of benefit from the system.
* **Cost:** The application has not any cost -completely free- the user can find or publish a job without paying.
* **Organizational Feasibility:**
* The number of employees in our system:
* We have 6 employees:

✓ 1 Admin   
✓ 2 Front-End developers   
✓ 2 Back-End developers   
✓ 1 Tester

* Payment for your employees?   
   ✓ 8,000 L.E for each Developer and Tester.   
   ✓ 6,000 L.E for each Admin.
* The coverage we need to generate and support our sales:
* We upgrade the application by the developers.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. **- Estimated Cost:**

**A cost estimate approximates a program, project, or operation's cost. the cost estimate is the product of the cost estimating process and our estimated cost for this project comes as following: -**

* We did not have any financial problems.
* We spent a lot of time and effort in self-studying about our technologies and searching for the best packages and service to be included in our app.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. **- Software & Hardware Requirements:**

Software Requirements:

|  |  |
| --- | --- |
| Programming Language: | Python (Django platform) |
| Web Server: | IIS |
| Database: | Oracle 9i |
| Web Technologies: | HTML, CSS, Java Script, Bootstrap |
| Operating System: | Windows XP, 98, 2000 |

Hardware Requirements:

|  |  |
| --- | --- |
| Processor: | Intel Pentium III or IV |
| Memory: | 2GB RAM or Greater |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. **- Gantt Chart:**

**A picture containing graphical user interface

Description automatically generatedGraphical user interface, table

Description automatically generated**

**Graphical user interface, table

Description automatically generatedA picture containing graphical user interface

Description automatically generatedGraphical user interface, table

Description automatically generated**

**Graphical user interface, application, Word

Description automatically generated**

* 1. **– Constraints:**
* **General constraints:**
* Tasks division which can be not fair enough.
* Indiscipline “Human factor” like being late in delivering tasks or attending meetings.
* Hesitation, especially when it is related to taking a serious step or learning a new technique.
* Time management.
* Learning new technologies may take much time.
* Underestimating the objectives that may not lead to realistic or achievable function.
* **Software constraints:**
* Making rules and conventions commonly agreed to in each programming environment explicit and automatically checkable.
* In frontend, we used HTML, CSS, JavaScript, & Bootstrap.
* In Backend, we used Python with Django framework.
* **Hardware constraints:**

*In this chapter we’re going to discuss and go deeper in how we made the suitable analysis to the project and show the real need to it and the limitation on it.*

**(Chapter 3) - Project Analysis -**

**3.1 - Analysis and Limitation of existing system.**

* 1. **- Need for the system.**

**3.3 - Analysis of the system.**

**3.4 - Risk and Risk Managements.**

**3.1 - Analysis and Limitation of existing system:**

* **Difficult to Navigate "Ease of Use":**

Navigation should let users know:

1.where they are

2.where they are going

3.where they have been

Navigations should be placed in a conspicuous place on the

website.

Single menus level is better than many smaller menus.

* **Limit Purchased Ways:**

Many systems today have only a few payment methods to get their services, and this is not compatible with some users and forces them to move to other sites.

* **Legal Considerations:**

Today's websites have become more regulated due to their widespread. Governments today are placing restrictions on these sites and how they work.

* **Limit Focusing on SEO:**

Optimize websites for SEO purposes. This will help the site rank higher in search engines and will help the website reach a larger audience.

* **Browser Compatibility:**

You must make sure that the system works well on many operating systems such as iOS, Mac, Windows, Linux, and android in their various browsers such as “Google, Safari, Edge, and Firefox.... etc.” This helps the site to spread faster. And get the most benefit.

* **Limit on Service Area:**

As an excellent site, services must be available in more than one country and in several languages, and this is not available in all employment sites.

**Things make the system operate slowly:**

* **Graphics**

Unoptimized Images:

High resolution images can consume lots of bandwidth while loading. Uploading larger sized images and then scaling them down can unnecessarily increase the size of your web page causing your website to load slowly.

* **Too Many Adds:**

Advertisements are additional HTTP requests and slow down page loading time.

* **Unclean Code:**

Use excessive white spaces, inline stylings, empty new lines, and unnecessary comments on website can make the website stylesheet grow larger, keeping page loading slow.

**3.2 - Need for the system:**

Today

1. **Redundancy:**

Manual records tend to contain data which is redundant. This is because Normalization followed is consistent data base is missing feature in manual records.

1. **Inconsistency:**

Because of redundant data, there is no consistency in the data.

1. **Time Consuming:**

Maintaining each user data with many numbers of fields is a tedious process and takes more time to retrieve back.

1. **Tedious:**

An organization database goes through many numbers of updates each day which is hard to maintain in registers.

1. **Insecurity:**

The website should be more secure against hackers to keep the users’ data safe, so it should be supported by vendors and constantly upgraded against the latest malware threats.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**3.3 - Analysis of the system:**

3.3.1 User requirements:

**Describe Functional & Non-Functional requirements, so that they are understandable by system users who do not have a detailed technical knowledge.**

* 1. The user shall register his\her data in the system.
  2. The user shall login to the system with his username and password.
  3. The user shall enter his\her profile and change privacy information.
  4. The user can enter another user’s profile.
  5. The users can communicate with each other via chat corner.
* **As a Job provider:**
  1. The user shall publish a job offer.
  2. The user can modify any job offer.
  3. The user shall get a notification if someone confirmed their job.
  4. The user shall accept the request from the performer.
* **As a Performer:** 
  1. The user shall search for any job from the home page.
  2. The user shall apply for the proper job.
* **Optional Requirements:**
  1. The user can upload his\her own CV.

\_\_\_\_\_\_\_\_\_\_\_\_\_

3.3.2 System Requirements:

**Statements that set out detailed descriptions of the system functions, services, and operational constraints.**

**And they define what should be implemented so it may be part of a contract between client and contractor.**

* 1. The system should receive the data from the new user.
  2. The data will be stored in the Database.
  3. The user will have full access to his account.
  4. The user will enter his login details to enter the website.
  5. Username and password will be checked in the Database if found login process done successfully.

3.1 The user will have full access to his/her account.  
 3.2 The user can change his/her information.

* 1. The user can change the privacy information whether it is

password or email connected with the account.

* 1. The user can visit other users' profiles.
  2. The user can see their public information.
  3. The user can see their job offers.
  4. The user can also see their CV.
  5. The user can send a request to connect with others.
  6. If there is connection between two users, they will be able to chat together.
  7. The user publishes his\her job offer with the description and requirements.
  8. The user can edit the job offer description anytime with any additional detail.

7.1 Users can easily access their published offers on their profiles

and modify them by clicking on “Edit post.”

7.2 Users can also delete any job offer if it is not needed anymore.

8.1 When a request from a performer is sent to the job provider,

he\she can easily receive a notification saying that a request is

received.

9.1 The provider enters the request and then enters the

performer’s profile.

9.2 The provider then checks the performer profile and CV, and

checks whether he\she is qualified to do it.

9.3 The provider shall accept or reject the request then.

10.1 Users in the system can upload their own CV on their profile.

10.2 The user enters his\her profile and click on “upload CV,” then

Successfully uploaded it.

11.1 After login into the system, user shall be in the home page

directly.

11.2 Users can click on the search icon or the search box to start

Searching for suitable jobs.

11.3 In the homepage will be there those jobs recommended by

the system according to the data of the user to match his/her

category and industry.

12.1 When a user meets the proper job and after reading all

Details, he shall send a request by clicking on “request job.”

12.2 The user shall wait till the provider accept his\her request the

Start connecting with each other and confirm the job.

* **Additional Requirements**
* **Web Browser**

The Website works on any type of browsers, we used Django framework that supports all browsers.

* **Internet Connection**

The website -as any website- needs a connection to work for sure, so users can be able to view it.

\_\_\_\_\_\_\_\_\_\_\_\_\_

3.3.3 Domain Requirements:

**Derived from the application domain and describes the system’s characteristics and features that reflect the domain.**

* Our site revolves 2 parties:

- Admin

- User (Job Provider or Performer)

* A person who wants to complete a job that can be done remotely (Provider) communicates with another person who wants to complete this work.
* To accomplish his task, The user also registers his data to create his own profile.
* The site must have update capabilities for the future models.
* The database should be backed up occasionally in case the original does become corrupt.
* The site must verify all values before making the change in the database.

\_\_\_\_\_\_\_\_\_\_\_\_\_

3.3.4 Functional Requirements:

**Statements of services the system should provide, how the system should react to specific inputs and how the system should behave in specific situations.**

* The user must be able to create a profile and upload a picture in it.
* The user must be able to see certain things from other users' profiles.
* The user must be able to rate the performer after the work has been completed.
* The user should be able to easily modify personal information.
* The system must contain means to protect the rights of both the client and the performer.
* The system must contain the means to protect property rights from any attack.
* The system must contain means of protection for personal information.
* The system must allow users to log in with their Google accounts.
* The system must allow users to reset their password by clicking on "I forgot my password" and receiving a link to their verified email address.
* system must suggest to the performer the appropriate job provider based on the skills that the performer possesses.
* The system must contain means of communication between users.
* The system should contain a simple website interface that highlights job offers in simple leaflets below each post the skills required to complete the work.
* The user must be able to access the previous projects of the performer and the ratings for each project.
* Users should be allowed to share your website’s content on social media.

**-User Scenario-**

|  |  |
| --- | --- |
| **Name** | Register |
| Actor | User |
| Input | Fill Registration form.  (E-mail, Username, Password) |
| Output | User will go directly to login page |
| Description | User can create an account to be a part of the application, and login to the application for the second time he opens the application with no need to register again |
| Pre-conditions | Open the application, client should enter valid data to create an account |
| Post-conditions | The User should have an account and go to the login page |

|  |  |
| --- | --- |
| **Name** | Login |
| Actor | User & Admin |
| Input | E-mail & Password |
| Output | Send user to home page |
| Description | User should enter his E-mail and password for authorizing that he has an account |
| Pre-conditions | Open the application, enter valid E-mail and password, click login button |
| Post-conditions | Logged into the system then redirected the home page |

|  |  |
| --- | --- |
| **Name** | View profile |
| Actor | User & Admin |
| Input | Click on user’s account name or profile’s button. |
| Output | User’s profile will be displayed |
| Description | User clicks on the profile’s button or name and view it |
| Pre-conditions | Open the application, user must have an account, login, then go to the personal account and view it |
| **Post-conditions** | The profile of the user selected will be displayed and can see all their shared data |

|  |  |
| --- | --- |
| **Name** | Edit profile |
| Actor | User |
| Input | Click “Edit” button |
| Output | Personal information is modified |
| Description | Any user can edit his own profile by going to the personal account and then click on edit |
| Pre-conditions | Open the application, user must have an account, login, go to the personal account and click on “edit” button |
| **Post-conditions** | The profile will be successfully updated with the new information |

|  |  |
| --- | --- |
| **Name** | **Search** |
| Actor | User & Admin |
| Input | Text to be searched |
| Output | The profiles or jobs that match with the text |
| **Description** | Function that helps retrieve the data needed by the users |
| Pre-conditions | Open the application, user must have an account, login, data should be existed |
| Post-conditions | The data will be retrieved |

|  |  |
| --- | --- |
| **Name** | **Publish a job** |
| Actor | User |
| Input | Click “add new job”, then enter job description and needs |
| Output | The job has been published and will be displayed to other users and on personal profile |
| Description | Function that allows users to publish their own jobs to help both individuals and professionals in mutual ways |
| Pre-conditions | Open the application, user must have an account, login |
| Post-conditions | The new job is successfully added and displayed to others |

|  |  |
| --- | --- |
| **Name** | **Apply a job** |
| Actor | User |
| Input | Click on “Apply for the job” button |
| Output | A notification will be sent to the job provider with the request |
| Description | Applying for the job that is previously published by the job provider |
| Pre-conditions | Open the application, user must have an account, login, Job published |
| Post-conditions | A notification will be sent to the job provider that a user has applied for the job and waiting for their response |

|  |  |
| --- | --- |
| Name | Accept job request |
| Actor | User |
| Input | Click “accept” button |
| Output | Notification with the acceptance will be sent to the accepted performer |
| Description | The job performer chooses the suitable one from the performers and accept him based on their C.V. and information |
| Pre-conditions | Performers have applied for the job |
| **Post**-**conditions** | The performer will be notified by the acceptance and start to connect with the provider using the chat and arrange for the job starting |

|  |  |
| --- | --- |
| **Name** | Edit a job |
| Actor | User |
| **Input** | Click on “edit” button from the job options list and edit the job description. |
| Output | The job will be edited. |
| Description | The job provider can edit the published job anytime by clicking on the “edit” button. |
| Pre-conditions | Open the application, user must have an account, login, user must have published jobs that can be edited |
| Post-conditions | The job will be displayed for the users and in personal profile with the new updates. |

|  |  |
| --- | --- |
| **Name** | Delete a job |
| Actor | User & Admin |
| Input | Click on “delete” button from the job options list. |
| Output | The job will be deleted |
| **Description** | Any job provider -anytime- can delete the published job. |
| Pre-conditions | Open the application, user must have an account, login, go to the personal account, select the needed job, and then click on “delete” button! |
| **Post-conditions** | The specific job will be deleted successfully. |

|  |  |
| --- | --- |
| **Name** | Get notified |
| Actor | User |
| Input | A user clicks apply to the job button |
| Output | New notification |
| Description | Users can get any new notification from the app if the job was applied, or a job was requested |
| Pre-conditions | Open the application, login.  User should publish or request a job to get a notification |
| Post-conditions | The provider chooses the suitable one from the performers who applied for the job.  And the performer will be notified though that he was accepted |

|  |  |
| --- | --- |
| **Name** | Chat |
| Actor | User |
| Input | Click “chat” icon |
| Output | A conversation between users |
| **Description** | Each user in the system can communicate with the others through the chat box. |
| Pre-conditions | Open the application,  Each user should have an account, login, users should have accepted a performer as a provider or accepted from a provider as a performer to be able to chat with each other and arrange for the job starting. |
| **Post-conditions** | Performer and job provider have a deal |

|  |  |
| --- | --- |
| **Name** | Delete user profile |
| Actor | Admin |
| Input | Click delete profile button. |
| Output | User’s profile is deleted |
| Description | The admin can delete any user’s profile if it goes against the system standards |
| Pre-conditions | Open the application, login.  The admin should go to the user’s profile then delete it |
| Post-conditions | The user’s account will be deleted. |

|  |  |
| --- | --- |
| **Name** | Logout |
| Actor | User & Admin |
| Input | Click logout button |
| Output | End user session |
| Description | After logging in to the application, they can logout from the application easily |
| Pre-conditions | Open the application, login, press logout button |
| Post-conditions | User will logout successfully and couldn’t use the application without logging in again |

3.3.5 Non- Functional Requirements:

**specify system/software properties (such as reliability and safety, security, etc..), and constraints on the services.**

1. **Easy to Use:**

## The system is easy to use by members of the public who might not read English.

## The system is easy to use on the first attempt by a member of the public without training.

## It is possible to use the system to pay in different currencies.

1. **high-Performance:**

## Users of the Job system can authenticate themselves using their username and password.

## The system should be organized in such a way that user errors are minimized.

## The system shall handle up to 100 users simultaneously.

* This project provides every employee with a specialized recruitment system.

1. **Reliability:**

## The system is able to be modified to cope with minor changes to European law that occur every six months on average.

## The language used in the interface is formal and polite.

1. **Availability:**

## The system shall, on average, operate without failure for 30 days (about 4 and a half weeks).

1. **Security:**

## The system ensures that only authorized users can gain access.

## The system distinguishes between authorized and non-authorized users.

1. **Robustness:**

* The system can handle error conditions gracefully, without failure. includes a tolerance of invalid data, software defects, and unexpected operating conditions.

1. **Scalability:**

* The website can handle a wide variety of system configuration sizes, by adding new features.
* Ease to use requirements address the factors that constitute the capacity of the software to be understood, learned, and used by its intended users.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. **- Risk and Risk Managements:**

1. **User risks**

* Every new user should register on site by e-mail and password. Poor password may cause risk (vulnerable to cyber-attacks). So, this risk management is to use a combination of (upper-case, lower-case, numbers and special characters), Make your password longer than 8 up to 20 characters.
* Password forgotten It might be difficult for users to remember their passwords all the time, such a problem will be solved by the reset password function. It will allow users to insert a new password once they prove their email address is still under their authority or using their personal phone number.
* The user also may not be normal user, as he may be a hacker.

1. **Operational Risk**

* An operational risk includes risks from poor implementation and process problems such as (database used in the system can’t process as many transactions per second as expected or Software components that should be reused contain defects that limit their functionality).

1. **Schedule Risk**

* Schedule related risks refers to time related risks or project delivery related planning risks. The wrong schedule affects the project development and delivery.

**Some reasons for Schedule risks:**

* Illness of a member of the team, which leads to delay in carrying out his tasks on the project.
* Time is not estimated perfectly.
* Poor tasks division

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*In this chapter we are going to discuss and go deeper in the project’s design and present its diagrams and database.*

**(Chapter 4) – Software Design –**

**(Chapter 3) – Software Design -**

**(Chapter 4) – Software Design -**

**(Chapter 3) – Software Design -**

**(Chapter 4) – Software Design -**

**(Chapter 3) – Software Design -**

**(Chapter 4) – Software Design -**

**(Chapter 3) – Software Design -**

**4.1 - Use case diagram.**

**4.2 - Activity diagram.**

**4.3 - Sequence diagram.**

**4.4 – Class diagram.**

**4.1- Use case diagram:**

Diagram

Description automatically generated

**4.2 - Activity diagram:**

1. Admin Activity:

Diagram

Description automatically generated

1. User Activity:

Diagram, engineering drawing

Description automatically generated

* 1. **- Sequence diagram:**

1. Login & Register:

Chart, bar chart

Description automatically generated with medium confidence

1. Admin Login:

Chart, bar chart

Description automatically generated

1. View Profile:

Diagram, schematic

Description automatically generated

1. Edit Profile:

Chart

Description automatically generated with medium confidence

1. Upload CV:

Diagram, schematic

Description automatically generated

1. Search:

A picture containing bar chart

Description automatically generated

1. Publish job:

Diagram, schematic

Description automatically generated

1. Apply job:

Timeline

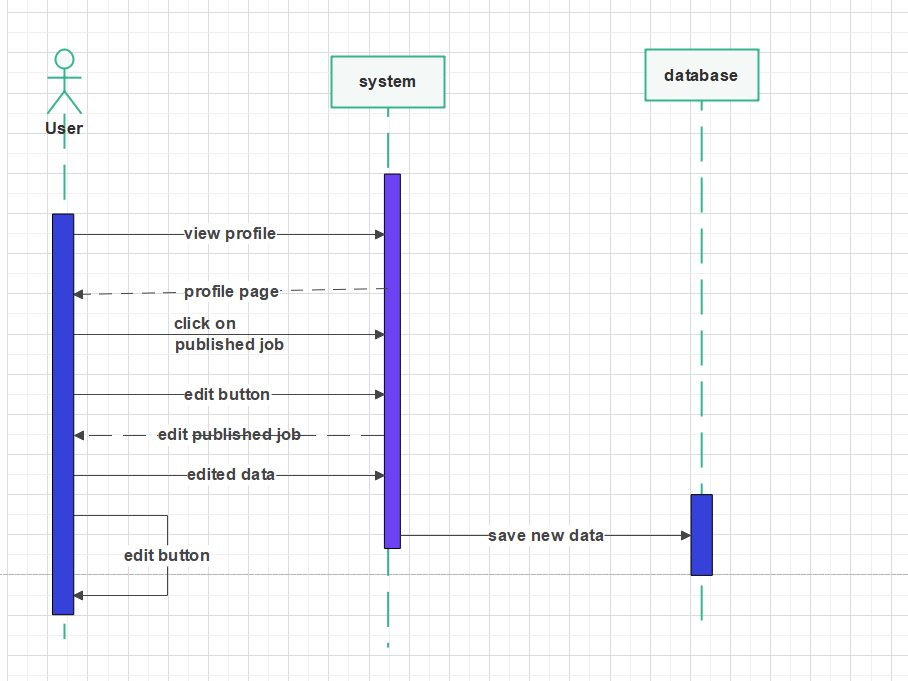
Description automatically generated with medium confidence

1. Accept job request:

A picture containing bar chart

Description automatically generated

1. Edit a job:



1. Delete a job:

Diagram, schematic

Description automatically generated

1. Get notified:

Diagram

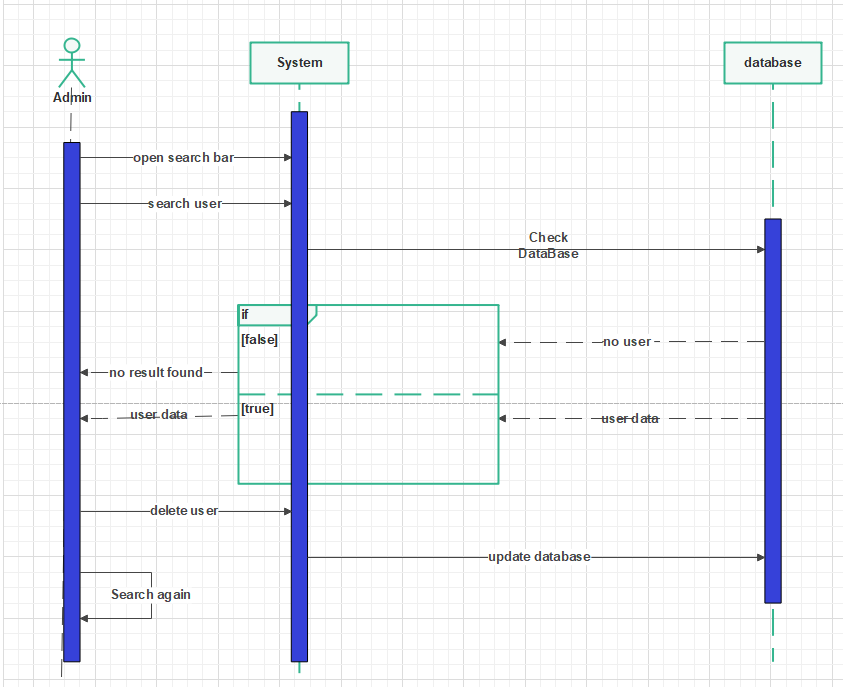
Description automatically generated

1. Chat:

Diagram

Description automatically generated

1. Delete user profile:



1. Logout:

Diagram, schematic

Description automatically generated

* 1. **- Class diagram:**

Diagram

Description automatically generated

Dd

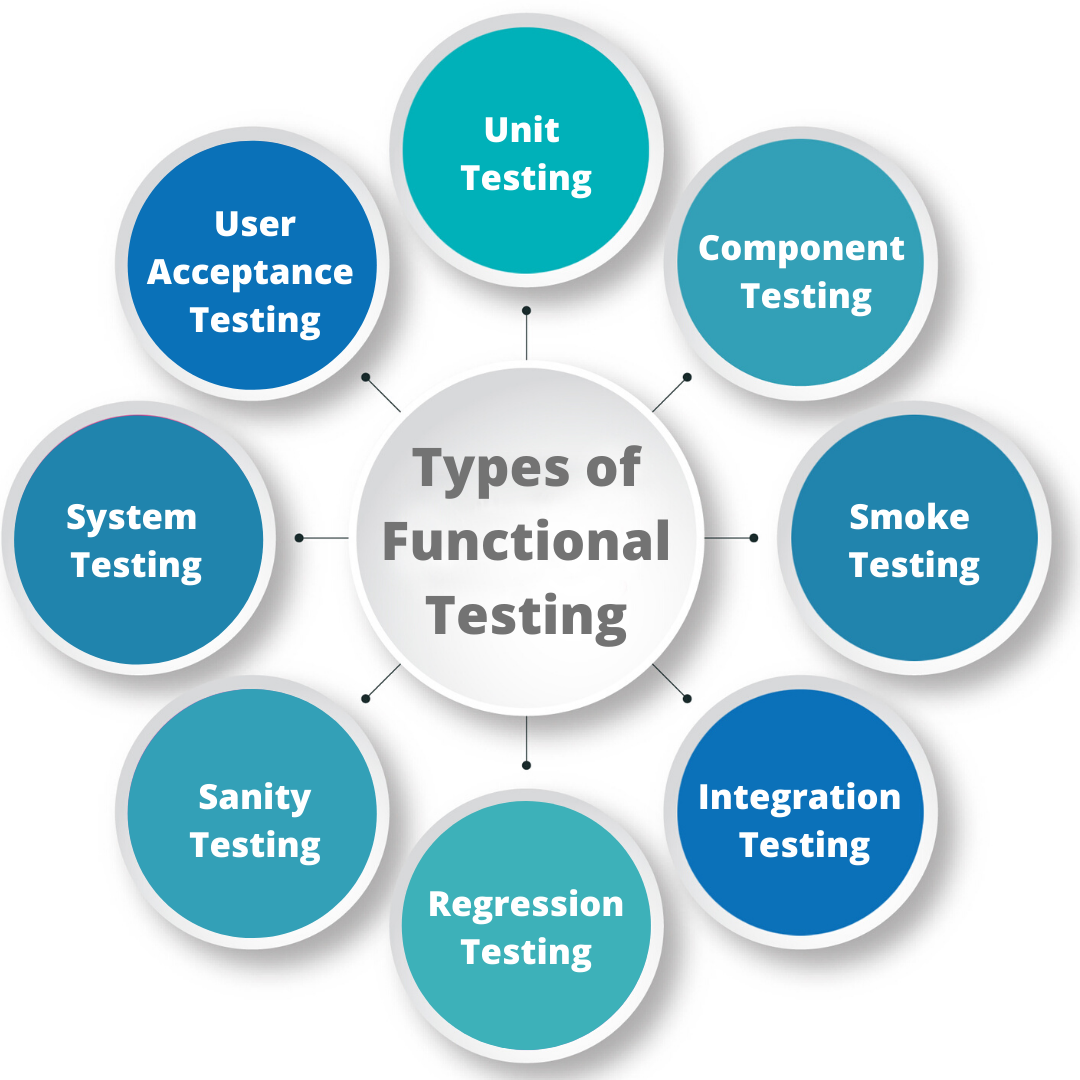
*In this chapter we are going to discuss and go deeper in Joblance website implementation and present its code and the libraries used to build it.*

**(Chapter 5) - Implementation -**

*In this chapter we are going to discuss and go deeper in the project testing and present the types of testing to be used and test cases we examined our application through.*

**(Chapter 6) – Testing -**

* **Functional Testing:**



*Figure “”*

* **Unit Testing**

Testing of individual items (e.g., modules, programs, objects, classes, etc.) Usually as part of the coding phase, in isolation from other development item sand the system as a whole.

* **Integration Testing**

Testing the interfaces between major (e.g., systems level application modules) and minor (e.g., individual programs or components) items within an application which must interact with each other.

* **System Testing**

Testing a system behavior as a whole when development is finished, and the system can be tested as complete entity. Regression Testing To check older functionality after integrating new functionality.

* **Acceptance Testing**

Testing to ensure that a development is ready to be deployed into the business, operational or production environment

* **Component Testing**

Testing to validate an individual component of the application before testing the entire application. As a consequence, component testing finds bugs and verifies the functionality of software modules/programs which are individually testable.

* **Smoke Testing**

 Testing to verify that the most important functionality is working. determines whether the deployed software build is stable or not.

* **Regression Testing**

Testing to practice that ensures an application still functions as expected after any code changes, updates, or improvements.

It is responsible for the overall stability and functionality of the existing features. Whenever a new modification is added to the code, regression testing is applied to guarantee that after each update, the system stays sustainable under continuous improvements.

* **Sanity Testing**

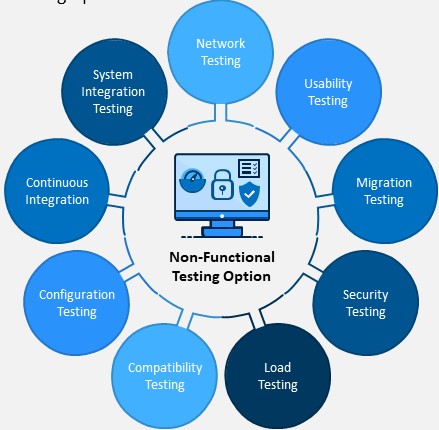
Testing that is performed to ensure that the code changes that are made are working as properly.

Sanity testing is a stoppage to check whether testing for the build can proceed or not.

The focus of the team during this testing process is to validate the functionality of the application and not detailed testing.

It is generally performed on build where the production deployment is required immediately like a critical bug fix.

**Non-Functional Testing:**



*Figure “”*

* **Network Testing**

Similar to Software Testing, is the process of analyzing and testing your network using a network performance test to identify bugs and performance issues, evaluate large network changes, and [measure network performance](https://obkio.com/blog/how-to-measure-network-performance-metrics/).

* **System Integration Testing**

The overall testing of the whole system which is composed of many sub-systems.

The main objective of SIT is to ensure that all software module dependencies are functioning properly and that data integrity is preserved between distinct modules of the whole system.

* **Usability Testing**

Accomplished a designated function regarding processing time and through put rate.

* **Migration Testing**

This testing is a**verification process** of migration of the legacy system to the new system with minimal disruption/downtime, with data integrity and no loss of data, while ensuring that all the specified functional and non-functional aspects of the application are met post-migration.

* **Security Testing**

Testing how well the system protects against unauthorized internal or external access

* **Load Testing**

Measuring the behavior of within creasing load which can be handled by the component or system.

* **Configuration Testing**

Testing technique in which the software application is tested with multiple combinations of software and hardware in order to evaluate the functional requirements and find out optimal configurations under which the software application works without any defects or flaws.

* **Compatibility Testing**

**Testing to check whether your software is capable of running on different hardware, operating systems, applications, network environments or Mobile devices.**

* **Continuous Testing**

Testing where the applications are tested continuously throughout the entire software development life cycle (SDLC).

*In this chapter we are going to find out the results of the project whether they are achieved or not and the differences between the desired results and the actual ones.*

**(Chapter 7) – Results and Discussion -**

7.1 Results

* Expected Result
* Actual Result

7.2 Discussion