**Chapter III**

**DESIGN AND DEVELOPMENT METHODOLOGIES**

**System Design**

The system/project developers created a comprehensive plan to develop a system that is only exclusive for the applicants, agents, and administrators. The reason for this is that when they use their traditional way of recruiting and looking for possible applicants, they spend a lot of effort and money. Which makes it hard for them and it is also costly since they have to travel around to personally meet the applicants. This system will make it easy for the company and their agents when it comes to recruiting applicants. Agents will recruit possible applicants for their job openings. Using the system, they could just send a link on their applicants were they can just register and log-in their accounts, after that they’ll just have to fill-up some forms and send their documents virtually, it will be more cost efficient and less effort for both sides. They could also just check the applicants’ documents and choose from them. They could also just talk to them remotely or just sending them an email for updates.

**Database Design**

The design of the database that includes a lot of entities on the E-recruitment is an important part of the website development. This compiles the information of the applicants that they input on the forms.

This displays the tables along with a description, related data types, and the fields that were used. The process collected, arranged, and verified the meaning of designated data phrases. This further illustrates how the two things are related.

The MySQL RDBMS is used by the developers for database design. With MySQL, we can establish relationships between tables that must be connected to one another. The E-Recruit database will be designed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Default** | **Description** |
| nonlife | varchar | 255 | Default Null | If nonlife |
| life | varchar | 255 | Default Null | If life |
| varlife | varchar | 255 | Default Null |  |
| accaAndHealth | varchar | 255 | Default Null |  |
| othercb | varchar | 255 | Default Null |  |
| othertb | varchar | 255 | Default Null |  |
| agencyname | varchar | 255 | Default Null | Agency Name |
| fname | varchar | 255 | Default Null | Applicant’s Name |
| nickname | varchar | 255 | Default Null | Applicant’s Nickname |
| birthdate | date |  | Current Time Stamp | Birth Date |
| placeOfBirth | varchar | 255 | Default Null | Applicant’s Place of Birth |
| gender | varchar | 10 | Default Null | Applicant’s Gender |
| bloodType | varchar | 5 | Default Null | Applicant’s Blood Type |
| homeAddress | varchar | 255 | Default Null | Applicant’s Home Address |
| mobileNo | varchar | 15 | Default Null | Applicant’s Mobile Number |
| landline | varchar | 15 | Default Null | Applicants’s landline |
| email | varchar | 255 | Default Null | Applicant’s Email |
| citizenship | varchar | 255 | Default Null | Applicant’s Citizenship |
| otherCitizenship | varchar | 255 | Default Null | Applicant’s Other Citizenship |
| naturalizationInfo | varchar | 255 | Default Null |  |
| maritalStatus | varchar | 20 | Default Null | Applicant’s Marital Status |
| maidenName | varchar | 255 | Default Null | Applicant’s Maiden Name |
| spouseName | varchar | 255 | Default Null | Applicant’s Spouse Name |
| sssNo | varchar | 20 | Default Null | Applicant’s SSS Number |
| tin | varchar | 20 | Default Null | Applicant’s TIN Number |

**Table 7. Fields for Aial Form**

Table 7 above contains the field name, data type, size, default, and description of the field in the Aial Tabel. Here, the id is the Primary Key (PK)(**KULANG PA TO, DI KO ALAM KUNG MAY PK BA YUNG TABLE OR WALA, ITATANONG KO PA KAY LES AT JANDEL**)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Data type** | **Size** | **Default** | **Description** |
| id | int |  | Default Null | Applicant’s ID |
| position | varchar | 255 | Default Null | Desired Position |
| preferedArea | varchar |  | Default Null | Preferred Area |
| referral | varchar | 255 | Default Null |  |
| referralBy | varchar | 255 | Default Null |  |
| onlineAd | text | 255 | Default Null |  |
| walkIn | varchar | 255 | Default Null |  |
| othersRef | varchar | 255 | Default Null |  |
| fname | varchar | 255 | Default Null |  |
| nickname | varchar | 255 | Default Null | Applicant’s Nickname |
| birthdate | date | 255 | Current Time Stamp | Applicant’s Brithdate |
| placeOfBirth | varchar | 255 | Default Null | Applicant’s Place of Birth |
| gender | varchar | 255 | Default Null | Applicant’s Gender |
| bloodType | varchar | 255 | Default Null | Applicant’s Blood Type |
| homeAddress | varchar | 255 | Default Null | Applicant’s Home Address |
| mobileNo | varchar | 255 | Default Null | Applicant’s Mobile Number |
| landline | varchar | 255 | Default Null | Applicant’s Landline |
| email | varchar | 255 | Default Null | Applicant’s Email |
| citizenship | varchar | 255 | Default Null | Applicant’s Citizenship |
| otherCitizenship | varchar | 255 | Default Null | Applicant’s Other Citizenship |
| naturalizationInfo | varchar | 255 | Default Null | Applicant’s Naturalization Info |
| maritalStatus | varchar | 255 | Default Null | Applicant’s Marital Status |
| maidenName | varchar | 255 | Default Null | Applicant’s Maiden Name |
| spouseName | varchar | 255 | Default Null | Applicant’s Spouse Name |
| sssNo | varchar | 255 | Default Null | Applicant’s SSS Number |
| tin | varchar | 255 | Default Null | Applicant’s TIN |
| lifeInsuranceExperience | varchar | 50 | Default Null | Applicant’s Life Insurance Experience |
| traditional | varchar | 50 | Default Null |  |
| variable | varchar | 50 | Default Null |  |
| recentInsuranceCompany | varchar | 50 | Default Null | Applicant’s Recently Insurance Company |
| highSchool | varchar | 50 | Not Null | Applicant’s High School |
| highSchoolCourse | varchar | 50 | Not Null | Applicant’s High School Course |
| highSchoolYear | varchar | 50 | Not Null | Applicant’s High School Year |
| graduateSchool | varchar | 50 | Not Null | Applicant’s Graduate School |
| graduateCourse | varchar | 50 | Not Null | Applicant’s Graduate Course |
| graduateYear | varchar | 50 | Not Null | Applicant’s Graduate Year |
| companyName1 | varchar | 50 | Not Null |  |
| position1 | varchar | 50 | Not Null |  |
| employmentFrom1 | varchar | 50 | Not Null |  |
| employmentTo1 | varchar | 50 | Not Null |  |
| reason1 | varchar | 50 | Not Null |  |
| companyName2 | varchar | 50 | Not Null |  |
| position2 | varchar | 50 | Not Null |  |
| employmentFrom2 | varchar | 50 | Not Null |  |
| employmentTo2 | varchar | 50 | Not Null |  |
| reason2 | varchar | 50 | Not Null |  |
| companyName3 | varchar | 50 | Not Null |  |
| position3 | varchar | 50 | Not Null |  |
| employmentFrom3 | varchar | 50 | Not Null |  |
| employmentTo3 | varchar | 50 | Not Null |  |
| reason3 | varchar | 50 | Not Null |  |
| companyName | varchar | 50 | Not Null |  |
| resposition | varchar | 50 | Not Null |  |
| contactName | varchar | 50 | Not Null |  |
| contactPosition | varchar | 50 | Not Null |  |
| emailAddress | varchar | 50 | Not Null |  |
| contactNumber | varchar | 50 | Not Null |  |
| yescureemployed | varchar | 50 | Not Null |  |
| nocureemployed | varchar | 50 | Not Null |  |
| allowed | varchar | 50 | Not Null |  |
| notallowed | varchar | 50 | Not Null |  |
| ifnoProvdtls | varchar | 50 | Not Null |  |

**Table 8. Fields for Life Changer Form**

Table 8 above contains the field name, data type, size, default and description of the field in the Products table. Here, the id is the Primary Key (PK).**(DI KO PARIN ALAM KUNG ALIN PK DITO)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Default** | **Description** |
| id | int |  | Not Null | User’s ID |
| email | text |  | Not Null | User’s Email |
| password | text |  | Not Null | User’s Password |
| role | text |  | Not Null | User’s Role |
| status | text |  | Not Null | User’s Status |
| token | varchar | 50 | Not Null | User’s Token |

**Table 9. Fields of Cart**

Table 9 above contains the field name, data type, size, default and description of the field in the Cart table. Here, the id is the Primary Key (PK), while the userid and menuid is the Foreign Key (FK).**(DI KO RIN ALAM PK)**

**Architectural Diagram/ Block Diagram**

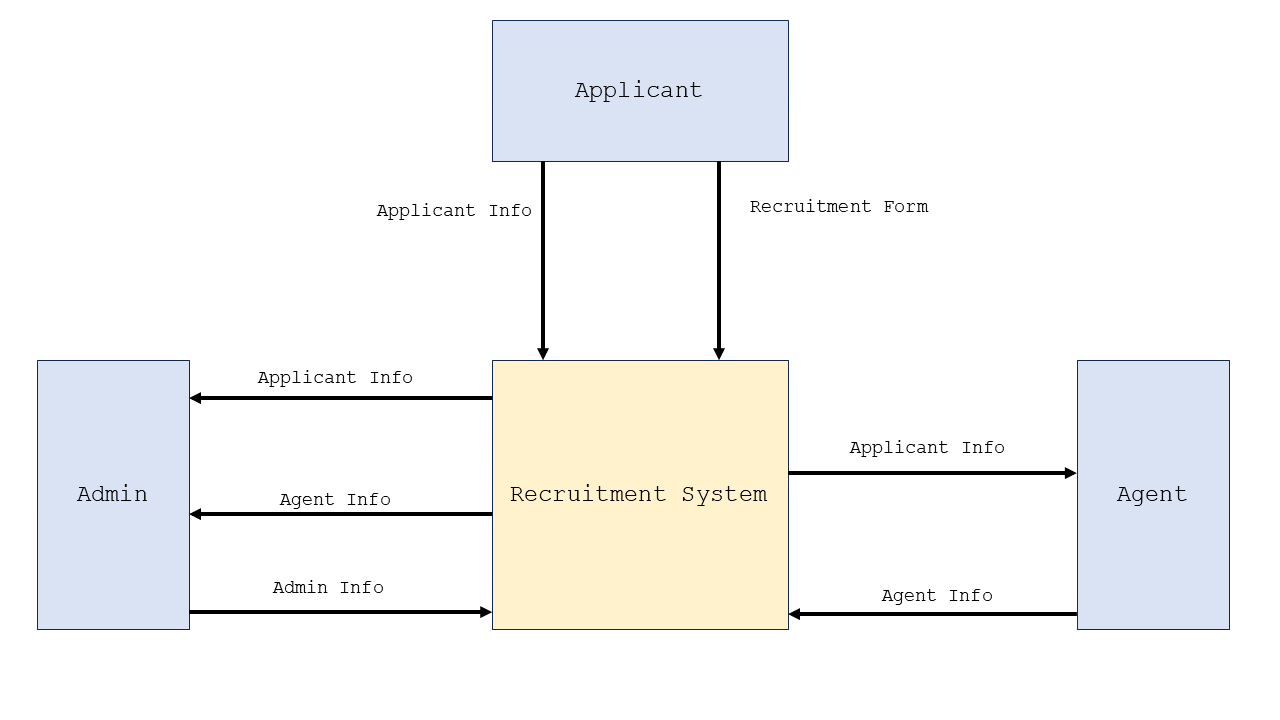
A computer and a computer with a arrow

Description automatically generatedIn this section, system architecture was designed to define the flow and behavior of the system’s functionalities to execute its high-quality performance. This covers the formal illustration and description of the project structure.

**Figure 1. System Architecture of E-Recruit Website**

Figure 1 shows the system architecture of the development of the E-Recruit Website. It displays the flow and how the system work. The researchers show that the internet is needed in order to access the website of both applicant, agent and admin to proceed with the application processes.

**Data Flow Diagram Level 0**

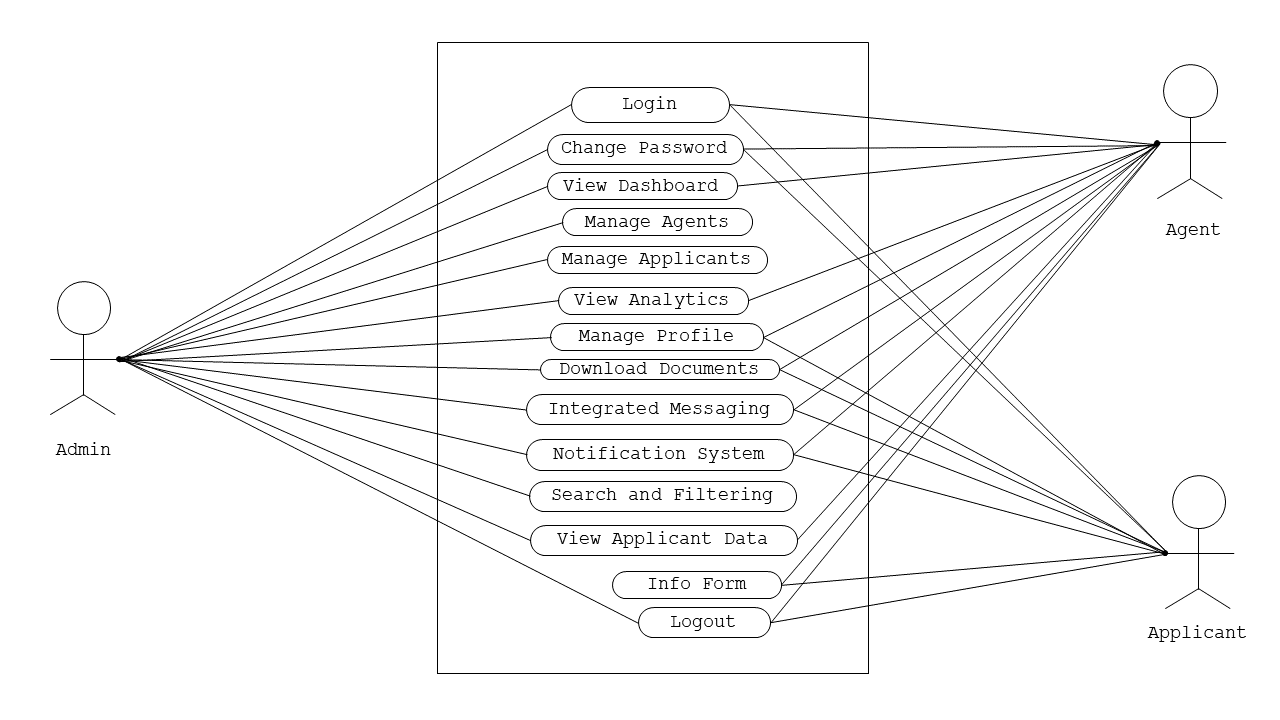
This section shows the Data Flow Diagram Level 0 which is commonly known as an exploded view of the context diagram that shows the detailed process of how the project works.

**Figure 2. Data Flow Diagram Level 0**

Figure 3: DFD Level 0 illustrates the interactions and data flows between the Admin, Agents, and Applicants.

**UML Use-case Diagram**

This section introduces the use case diagram, which provides a high-level overview of functions within a system. It includes a graphic representation illustrating the relationships among key actors such as the system, admin, agent, and users. The diagram serves as a tool for researchers to understand and organize the system's functionality by showcasing how these actors interact.



**Figure 3. UML Use-case Diagram**

Figure 3 shows the roles of the Administrator, Agent, and the Applicants to be executed in the whole process of the system.

**Sample Mock-up**

**A screenshot of a computer

Description automatically generated**A sample mock-up is a visual representation of a website after it is built. It consists of visuals that show how the website should look and its function. It is used to refine the design, identify potential problems, and ensure that the system meets the user's needs and expectations. Below are the system users and admin interface

**Figure 4. User Interface**

**Development Method**

Agile method is chosen for this research, for the reason that it is really well-suited for developing the website for E-recruitment System. Each of which is necessary for an online hiring system to be successful and efficient. The effectiveness and efficiency of an online hiring system depend on its ability to adapt to changing needs and developing technology, which is why this strategy enables continuous improvements and adjustments. Agile's emphasis on customer input and collaboration further guarantees that the E-recruitment System will precisely match the needs and expectations of its users, which enhances the system's A diagram of a process

Description automatically generatedoverall efficacy.

**Figure 5. Agile Methodology Model**

Figure 5 illustrates the researchers' utilization of Agile methodologies as frameworks for project development. This method places emphasis on breaking down large projects into smaller, easier-to-manage activities. Teams participate in all phases of the project lifecycle, from gathering requirements to design, coding, development, and testing, and the tasks are finished in short iterations. The client is presented with and given a demonstration of a working system at the end of this phase.

**Gantt Chart**

In this section, Gantt Chart is presented to show the plans and schedules of the project timeline. All the development stages up to the completion of the project were documented in this chart. This helps the researchers to know the deadlines needed to accomplish and show breakthroughs in various tasks.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Task Name** | **Task Date** | | | | | | | | | |
| Oct | | | | Nov | | | | Dec | |
| Week 1 | Week 2 | Week 3 | Week 4 | Week 1 | Week 2 | Week 3 | Week 4 | Week 1 | Week 2 |
| **1.Planning** |  |  |  |  |  |  |  |  |  |  |
| 1.1 Conduct an interview |  |  |  |  |  |  |  |  |  |  |
| 1.2 Define project objectives |  |  |  |  |  |  |  |  |  |  |
| 1.3 Define project plan |  |  |  |  |  |  |  |  |  |  |
| 1.4 Approval of project plan |  |  |  |  |  |  |  |  |  |  |
| **2.Requirements Gathering** |  |  |  |  |  |  |  |  |  |  |
| 2.1 Data Collection |  |  |  |  |  |  |  |  |  |  |
| 2.2 Functional |  |  |  |  |  |  |  |  |  |  |
| 2.3 Non-Functional |  |  |  |  |  |  |  |  |  |  |
| **3.Design** |  |  |  |  |  |  |  |  |  |  |
| 3.1 Frontend software design |  |  |  |  |  |  |  |  |  |  |
| **4.Development** |  |  |  |  |  |  |  |  |  |  |
| 4.1 Back-end coding |  |  |  |  |  |  |  |  |  |  |
| **5.Testing** |  |  |  |  |  |  |  |  |  |  |
| 5.1 Functionality testing |  |  |  |  |  |  |  |  |  |  |
| 5.2 User interface testing |  |  |  |  |  |  |  |  |  |  |

**Table 5 Group Gantt Chart**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Task Name** | **Task Date** | | | | | | | | | |
| Oct | | | | Nov | | | | Dec | |
| Week 1 | Week 2 | Week 3 | Week 4 | Week 1 | Week 2 | Week 3 | Week 4 | Week 1 | Week 2 |
| **1.Planning** |  |  |  |  |  |  |  |  |  |  |
| 1.1 Conduct an interview |  |  |  |  |  |  |  |  |  |  |
| 1.2 Define project objectives |  |  |  |  |  |  |  |  |  |  |
| 1.3 Define project plan |  |  |  |  |  |  |  |  |  |  |
| 1.4 Approval of project plan |  |  |  |  |  |  |  |  |  |  |
| **2.Requirements Gathering** |  |  |  |  |  |  |  |  |  |  |
| 2.1 Data Collection |  |  |  |  |  |  |  |  |  |  |
| 2.2 Functional |  |  |  |  |  |  |  |  |  |  |
| 2.3 Non-Functional |  |  |  |  |  |  |  |  |  |  |
| **3.Development** |  |  |  |  |  |  |  |  |  |  |
| 4.1 Back-end coding |  |  |  |  |  |  |  |  |  |  |
| **4.Testing** |  |  |  |  |  |  |  |  |  |  |
| 5.1 Functionality testing |  |  |  |  |  |  |  |  |  |  |
| 5.2 User interface testing |  |  |  |  |  |  |  |  |  |  |

**Table 6 Jandel Escalera Gantt Chart**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Task Name** | **Task Date** | | | | | | | | | |
| Oct | | | | Nov | | | | Dec | |
| Week 1 | Week 2 | Week 3 | Week 4 | Week 1 | Week 2 | Week 3 | Week 4 | Week 1 | Week 2 |
| **1.Planning** |  |  |  |  |  |  |  |  |  |  |
| 1.1 Conduct an interview |  |  |  |  |  |  |  |  |  |  |
| 1.2 Define project objectives |  |  |  |  |  |  |  |  |  |  |
| 1.3 Define project plan |  |  |  |  |  |  |  |  |  |  |
| 1.4 Approval of project plan |  |  |  |  |  |  |  |  |  |  |
| **2.Requirements Gathering** |  |  |  |  |  |  |  |  |  |  |
| 2.1 Data Collection |  |  |  |  |  |  |  |  |  |  |
| 2.2 Functional |  |  |  |  |  |  |  |  |  |  |
| 2.3 Non-Functional |  |  |  |  |  |  |  |  |  |  |
| **3.Design** |  |  |  |  |  |  |  |  |  |  |
| 3.1 Frontend software design |  |  |  |  |  |  |  |  |  |  |
| **4.Testing** |  |  |  |  |  |  |  |  |  |  |
| 5.1 Functionality testing |  |  |  |  |  |  |  |  |  |  |
| 5.2 User interface testing |  |  |  |  |  |  |  |  |  |  |

**Table 7 Jef Ramos Gantt Chart**

**Legend:** - Completed/ Done

Table 5 to 8 shows the whole process of developing E-Recruit Website. It displayed the various tasks and marks as completed oats certain date as a group, but the other three tables are for individual gantt charts. Researchers will be kept informed of the progress of the development which will help them not to miss out on steps and differentiate tasks from the amount of time took to complete them.

**Testing and Evaluation**

This section is referring to the testing and evaluation phase of the software development process. During this phase, developers are required to conduct tests on their system to determine its capabilities and limitations. This will allow them to identify any issues or potential problems prior to the production and deployment stages. The tests should include all the requirements outlined in the Requirements Phase, such as design, performance, supportability, etc. The results of these tests will be evaluated to assess the progress of the system and ensure it meets the requirements of the project. The developers considered the following:

1. **Unit Testing**- a type of software testing where individual units or components of a software system are tested. This type of testing is usually done by the developers, as it requires detailed knowledge of the internal structure of the system. Unit Testing is designed to test individual functions, modules, and features of the system, to ensure that they all behave as expected
2. **Component Testing**- involves testing the individual components of the system (modules, classes, objects, and programs) in isolation, without integrating them with other components. This helps to identify any defects or bugs in the individual components before they are integrated into the system. It also helps to identify any unexpected interactions or dependencies between components that could cause problems or errors in the system.
3. **System Testing**- a type of software testing that evaluates the entire system or application and its components to verify that all individual modules are working properly and that data is transferred accurately between modules and the entire system. System testing is meant to ensure that the system meets its requirements, performs as expected, and functions correctly in its intended environment. It is an overall test of the system and its components, and it is typically done after unit and integration testing.

**REFERENCES**

Edirisinghe, S. M. (2020). *Recruitment Management System*. <https://dl.ucsc.cmb.ac.lk/jspui/bitstream/123456789/4486/1/2017%20MIT%20017.pdf>

Grace, M., Ventura, G., & Bringula, R. P. (2013). *Effectiveness of Online Job Recruitment System: Evidence from the University of the East*. www.IJCSI.org

Obipi , I. O., & Kalio, N. (2018). Recruitment Management System and Employee Procurement in the Oil and Gas Sector in Nigeria. *International Journal of Human Resource Management and Research*, *8*(2), 7–18. https://doi.org/10.24247/ijhrmrapr20182

Kmail, A. B., Maree, M., Belkhatir, M., & Alhashmi, S. M. (2016). An automatic online recruitment system based on exploiting multiple semantic resources and concept-relatedness measures. *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, *2016*-*January*, 620–627. https://doi.org/10.1109/ICTAI.2015.95

Ramadhani, F., & Zarlis, M. (n.d.). *Analysis of e-Recruitment System Design*. https://doi.org/10.17706/ijeeee.2019.9.1.38-45

Rosoiu, O., & Popescu, C. (n.d.). *E-recruiting Platforms: Features that Influence the Efficiency of Online Recruitment Systems*. https://doi.org/10.12948/issn14531305/20.2.2016.05

Samoli, C. (2021, May 20). *How management systems drive value in business operations*. Monday.com Blog. https://monday.com/blog/project-management/management-systems/

Sehgal, V. K., Jagtiani, A., Shah, M., Sharma, A., Jaiswal, A., & Mehta, D. (2014). Job portal-A web application for geographically distributed multiple clients. *Proceedings - 1st International Conference on Artificial Intelligence, Modelling and Simulation, AIMS 2013*, 199–204. https://doi.org/10.1109/AIMS.2013.38