A logo for university of australia

Description automatically generated

**COIT20258: SOFTWARE ENGINEERING**

**Term 1, 2024**

**ASSESSMENT 3 – GROUP ASSIGNMENT (AIS-R-ENHANCED)**

**Sydney Campus**

6th June 2024

**Lecturer:** MAHBUB AHMED

**Tutor:** ZAKIULLAH KHAN

**Prepared by:**

TAIMOUR SHAHZAD (12210758) (Leader)

VARIAN VALERIAN RUMAO (12222847)

MILAN KORANGI (12213436)

Table of Contents

[**1.** **Introduction** 1](#_Toc168585297)

[**2.** **Requirement specification** 2](#_Toc168585298)

[**1.1** **Functional Requirements:** 2](#_Toc168585299)

[**1.2 Non-Functional Requirements:** 2](#_Toc168585300)

[**1.3 System Requirements:** 3](#_Toc168585301)

[System Architecture: 3](#_Toc168585302)

[System tools requirement: 4](#_Toc168585303)

[Database script to create schema and tables: 4](#_Toc168585304)

[Running the Project: 4](#_Toc168585305)

[**3.** **Quality Plan** 5](#_Toc168585306)

[**2.1 Quality Assurance Procedures** 5](#_Toc168585307)

[**2.2 Quality Control Measures** 5](#_Toc168585308)

[**2.3 Responsibilities** 6](#_Toc168585309)

[**4.** **Design - UML diagrams:** 6](#_Toc168585310)

[**4.1.** **Context Diagram** 6](#_Toc168585311)

[**4.2.** **Use cases diagram.** 7](#_Toc168585312)

[**4.3.** **Classes diagrams** 8](#_Toc168585313)

[**4.4.** **Sequence diagrams.** 8](#_Toc168585314)

[**4.5.** **ERD for the database** 10](#_Toc168585315)

[**5.** **Test Plan.** 10](#_Toc168585316)

[**6.** **Evidence of Testing** 11](#_Toc168585317)

[**7.** **Conclusion** 12](#_Toc168585318)

# **Introduction**

AI-SPACE, an Australia based intelligent aerospace company, researching about design and manufacture of renewable energy driven drones and robotic space vehicles. The AI-SPACE is looking to onboard new engineering recruits to carry out the research and development in its new R&D facility currently located in Melbourne branch. AI-SPACE has commenced the recruitment process by advertising for applicants to fill positions for the software, aerospace, mechanical and electronics engineering departments of the AI-SPACE. Therefore, AI-SPACE requires the system to manage the recruit details efficiently. This system will perform the several function to keep the details of recruit and current staff. The main aims of the system are the following:

* To store current staff based on role, location and their position type.
* To store data of applicants in the recruitment process.
* To setup interview date and time.
* To make the recruitment operational process easier and efficient.
* To register details by recruiters themselves.
* Assign department to recruiters based on their profile.

This document specifies the details of the AIS-R-Enhanced system and its functionalities. The system is based on Model-View-Controller (MVC) architecture where model lies in multi-threaded server and controller and view lie in GUI client. The overview of the system is highlighted by a context diagram which includes the sub-systems with corresponding functionalities. Moreover, the use case diagram, class diagram, ERD, and sequence diagram will give better understanding of the system with its functionalities of different sub-system with their possible outcomes.

# **Requirement specification**

## **Functional Requirements:**

The AIS-R-Enhanced system should meet the following functional requirements:

* When the multi-threaded server starts for the first time, it should save the contents of 'staff.csv' and 'recruits' into the database.
* The system must include user authentication mechanisms for all users like administrative, Recruits and management staff.
* There should be registration functionalities that allow users to input and modify admin, management and recruits’ data.
* The system should enable management staff to assign recruits to specific one or two engineering departments.
* Reporting functionalities should be included to generate data-driven insights for recruit’s data.
* The system must support loading and saving data to a MySQL database to ensure seamless integration of server with backend storage systems.
* Several clients can perform task con-currently at same time.

## **1.2 Non-Functional Requirements:**

The non-functional requirements for AIS-R-Enhanced focus on performance, security, usability, and maintainability to ensure overall software quality. The following are the non-functional requirements for AIS-R-Enhanced:

1. **Performance:** The system must be responsive and scalable, able to handle numerous concurrent users without a drop in performance.
2. **Security:** Strong security measures must be in place to protect sensitive data. This includes user authentication, data transmission encryption, and access control mechanisms.
3. **Reliability:** The system should maintain high availability and fault tolerance, reducing downtime and ensuring continuous access to essential features.
4. **Scalability:** The system architecture should support future growth and expansion, allowing for the seamless integration of additional features and functionalities as needed.

## **1.3 System Requirements:**

## System Architecture:

The AIS-R system is built on the Model View Controller (MVC) architecture, a commonly used design pattern in software development, particularly for Graphical User Interface (GUI) applications. The architecture involves a multi-threaded server as model, controller and a client as view, detailed as follows:

* **Server (Model)**: The server is multi-threaded and contains all the models and database connectivity. The model (M) encompasses all the business logic, managing the data and interacting with the database. The primary function of model is to accept request of controller and perform business logic or database manipulation and send back response to controller.
* **Client (View and Controller)**: The client is a GUI-based application consisting of controllers (C) and views (V). The controller serves as an intermediary between the model and the view (UI). The view represents the user interface visible to the user. The controller sends data received from UI to the model and supply response received from server to View. The view shows the data received from controller to user and allows user to interact and sends user form data to controller.

A diagram of a software development

Description automatically generated

Figure 1. System Architecture of AIS-R-Enhanced

## System tools requirement:

* MySQL database
* NetBeans
* JDK 17 or above
* JavaFX Senebuilder
* Git and GitHub

## Database script to create schema and tables:

* CREATE SCHEMA `ais-r-db` ;
* CREATE TABLE IF NOT EXISTS Admin (

admin\_id INT AUTO\_INCREMENT PRIMARY KEY,

fullName VARCHAR(50) NOT NULL,

address VARCHAR(50) NOT NULL,

phoneNumber VARCHAR(50) NOT NULL,

email VARCHAR(50) NOT NULL,

password VARCHAR(50) NOT NULL,

level VARCHAR(50) NOT NULL,

branch VARCHAR(50) NOT NUL;

CREATE TABLE IF NOT EXISTS Recruits (

recruit\_id INT AUTO\_INCREMENT PRIMARY KEY,

fullName VARCHAR(100) NOT NULL,

address VARCHAR(100) NOT NULL,

phoneNumber VARCHAR(20),

email VARCHAR(50),

qualification VARCHAR(50),

department VARCHAR(50),

interviewDate DATE;

CREATE TABLE IF NOT EXISTS Management (

manage\_id INT AUTO\_INCREMENT PRIMARY KEY,

fullName VARCHAR(100) NOT NULL,

address VARCHAR(100) NOT NULL,

phoneNumber VARCHAR(50),

userName VARCHAR(100)

password VARCHAR(50) NOT NULL,

position VARCHAR(50) NOT NULL;

## Running the Project:

Always run server first and then client.

* Server Instructions

You can directly compile and run from NetBeans.

* + Open project in NetBeans IDE.
  + Make a database AIS-R-DB.
  + Right click on Server and click on Run.
* Client

You can directly compile and run from NetBeans.

* + Goto Client project.
  + Right click on project and click run.

GitHub link: [cqumilankorangi/AIS-R-Enhanced (github.com)](https://github.com/cqumilankorangi/AIS-R-Enhanced)

# **Quality Plan**

The quality plan for AIS-R-Enhanced outlines the protocols and standards essential for developing a high-quality software prototype. It covers various quality assurance aspects, including thorough requirements validation, rigorous design review, detailed code review, comprehensive testing procedures, and strict documentation standards. The primary quality objectives are to ensure that AIS-R-Enhanced meets all specified functional and non-functional requirements, adheres to industry best practices and coding standards, undergoes extensive testing to identify and resolve any defects or discrepancies, and provides comprehensive documentation to facilitate understanding, maintenance, and scalability. The following techniques, measures, and responsibilities are adopted to maintain the quality of AIS-R-Enhanced:

## **2.1 Quality Assurance Procedures**

* Requirements Validation: The team will review and validate all functional and non-functional requirements to ensure they are clear, complete, and consistent.
* Design Review: The system architecture, including UML diagrams and database schema, will undergo a design review to ensure it adheres to best practices and is scalable.
* Code Review: Team members will conduct regular code reviews to identify and address any coding errors, maintainability issues, or deviations from coding standards.
* Testing Procedures: The system will undergo various levels of testing, including unit testing, integration testing, system testing, and acceptance testing, to verify its functionality, performance, and usability.

## **2.2 Quality Control Measures**

* Version Control: The project will be managed using version control software (e.g., GitHub) to track changes, facilitate collaboration, and ensure code integrity.
* Peer Reviews: Regular peer reviews will be conducted to ensure all code changes meet quality standards and are well-documented.
* Adherence to Coding Standards: The team will follow established coding standards, including naming conventions, formatting guidelines, and best practices for code readability and maintainability.

## **2.3 Responsibilities**

* Team Leader: Oversees the overall quality of the project, coordinates reviews, and ensures quality assurance procedures are followed.
* Development Team: Adheres to coding standards, conducts testing, and participates in code reviews and design reviews.
* Quality Assurance Team: Validates requirements, conducts reviews, and ensures quality control measures are effectively implemented.

# **Design - UML diagrams:**

## **Context Diagram**

The AIS-R-Enhanced system is composed of four sub-systems. Each sub-system has different functionalities the make complete system. The employee staff management system manages the records of staff information. The Recruit management system manages the record of applicant for new recruitment process. The Report and insights management system presents data in graphical view for staff to visualize recruit data. Department management system allocates recruit based on its information to corresponding department.

A diagram of a system

Description automatically generated

Figure 1. AIS-R-Enhanced system context diagram.

## **Use cases diagram.**

The main functional requirements are translated into use case diagram of the AIS-R-Enhanced. Currently, we have three actors in our system. They can register individually, or admin and management can register multiple recruits.

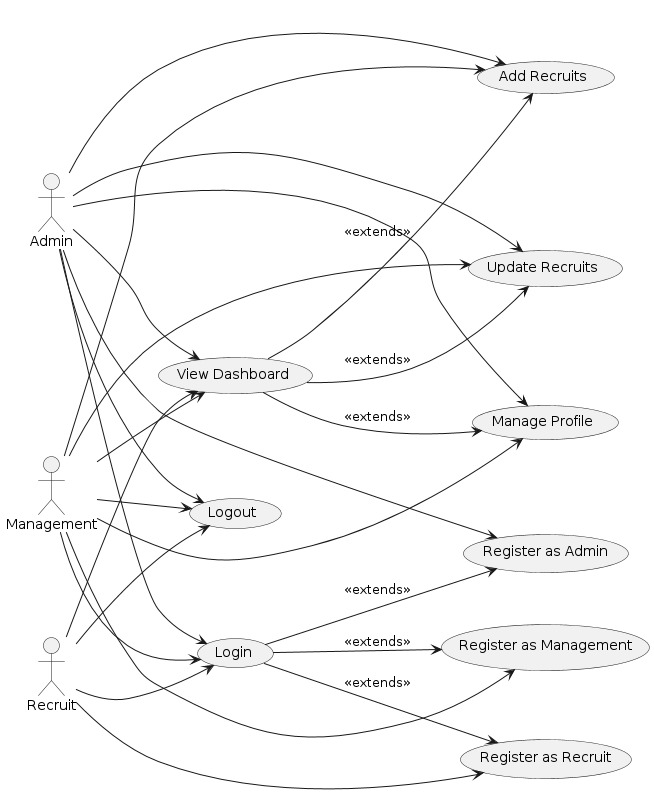


Figure 2. AIS-R-Enhanced Usecase Diagram.

## **Classes diagrams**

A diagram of a diagram

Description automatically generated with medium confidence

Figure 3. AIS-R-Enhanced Class Diagram.

## **Sequence diagrams.**

The following sequence diagram shows the behavioural model of AI-R-Enhanced system for major functions in system.

A white sheet of paper with black text

Description automatically generated

Figure 4. AIS-R-Enhanced Sequence Diagram.

## **ERD for the database**

A diagram of a server

Description automatically generated

Figure 5. AIS-R-Enhanced ERD Diagram.

# **Test Plan.**

4.1 Test plan

|  |  |  |  |
| --- | --- | --- | --- |
| **Test** | **Input** | **Expected Results** | **Actual Results** |
| Admin Registration | Fullname, Username, Password, Email, Contact, Level and Branch | Should be able to register new Admin. | New admin added successfully and redirect to login page. |
| Management Registration | Fullname, Address, Phone Number, Email, Username and Password. | Should be able to register new Management. | New Management added successfully and redirect to login page. |
| Recruit Registration | Fullname, Address, Phone Number, Email, Username, Password, Interview Date and highest qualification level. | Should be able to add recruit details. | New Recruit added successfully by himself or herself and redirect to login page. |
| Admin Login | Username and password | Redirect to Admin Panel | Redirect to Admin Panel |
| Management Login | Username and password | Redirect to Management Panel | Redirect to Management Panel |
| Recruit Login | Username and password | Redirect to Recruit Panel | Redirect to Recruit Panel |
| Dashboard for management |  | Should be visible to management only. | It is visible for Management only. |
| Management Assign department to Recruit | Options: "Software", "Aerospace", "Mechanical", "Electronics Engineering" | Management should be able to one or two department to Recruit | Management assigned department to Recruit. |
| Add/update Recruit by admin and management | Recruit details | Should be able to add/update recruit details | Admin and Management added/update recruit. |
| Update Recruit by Recruit. | Recruit details | Should be able to update own details with the help of otp | Updated own details with the help of otp |
| Report based on Sort | Options: “Sort by lastname in descending order and groupby location” and “Highest level of Qualification” | Recruit data renders according to filter selected. | Recruit data rendered according to filter selected. |
| Csv upload to database | Sfaff.csv and recruits.csv | Should be able to upload all details of both staff and recruits to database when server starts. | When server started it entered staff and recruits’ data into database of corresponding table. |
| Create Table Programmatically |  | Should create table if no table exists. | When server run it checked table and created new table whenever no table existed. |
| Listing Management Staff |  | List of Management staff should be shown in table | List of management staff data is rendered in table. |
| Log out |  | Should take out to login page. | Took out from corresponding Panel |

# **Evidence of Testing**

6.1 Login Testing

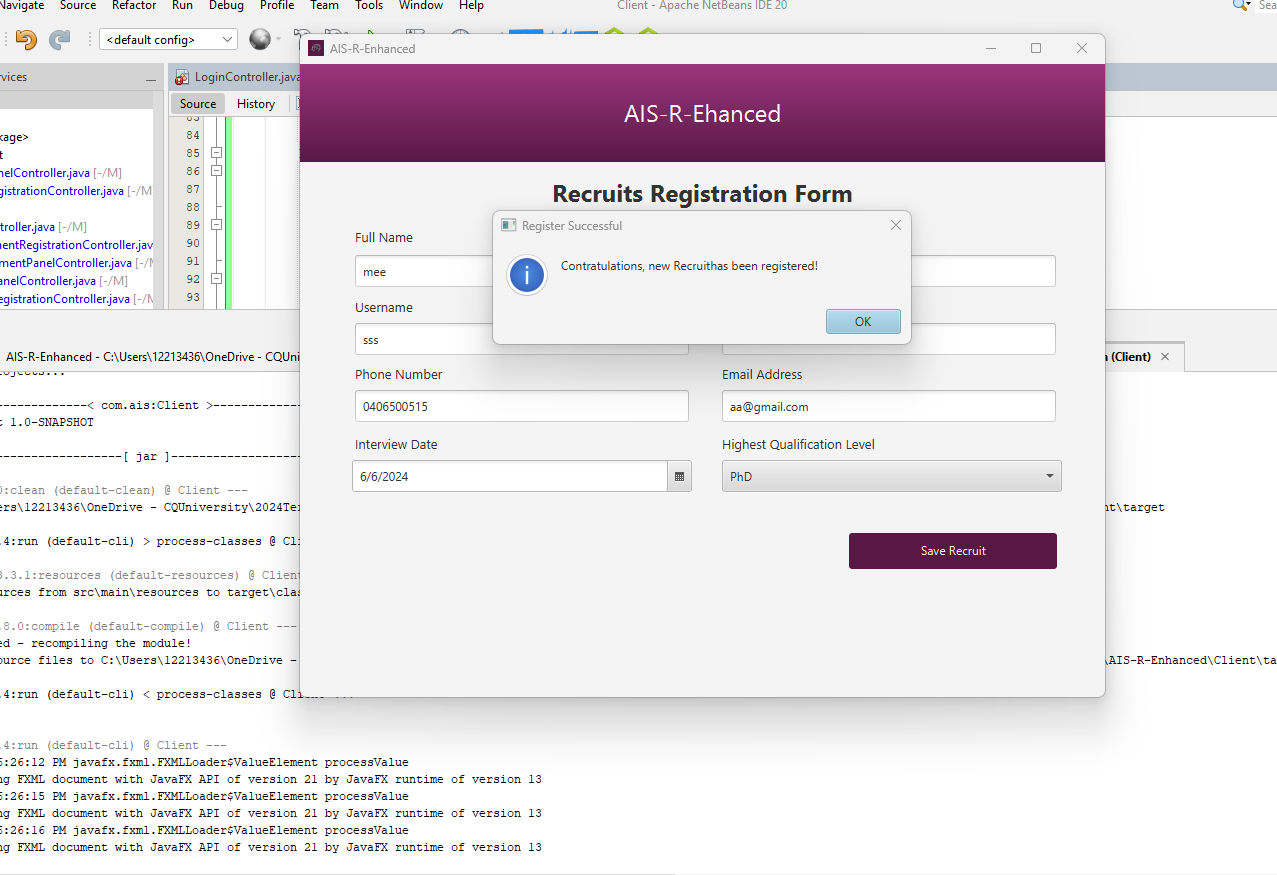
A screenshot of a computer

Description automatically generated

6.2 Registration Testing

A screenshot of a computer

Description automatically generated



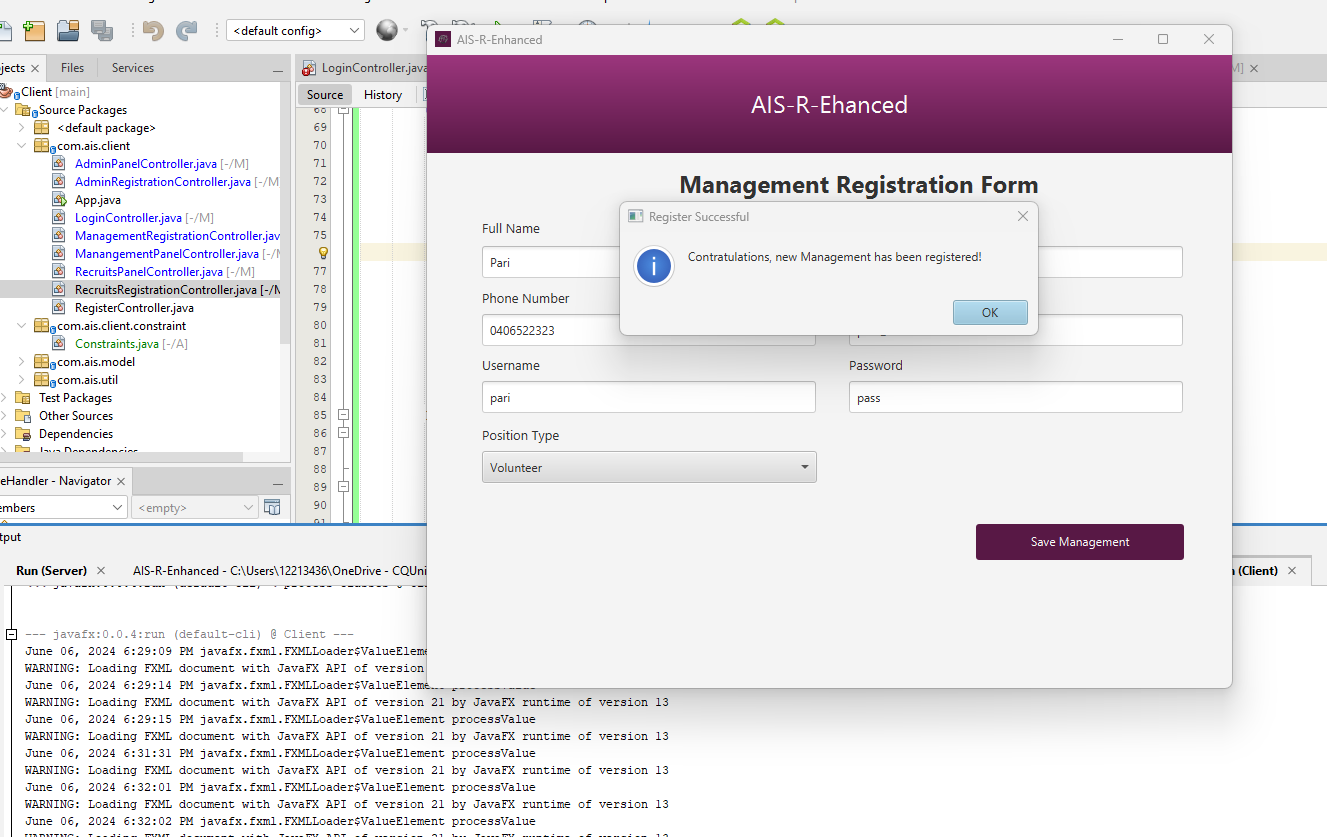


Figure 1. Registration Entry Data Details for all users of AIS-R-Enhanced

6.3 CSV Upload Testing

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

6.4 Add/Update Recruit by Admin

A screenshot of a computer

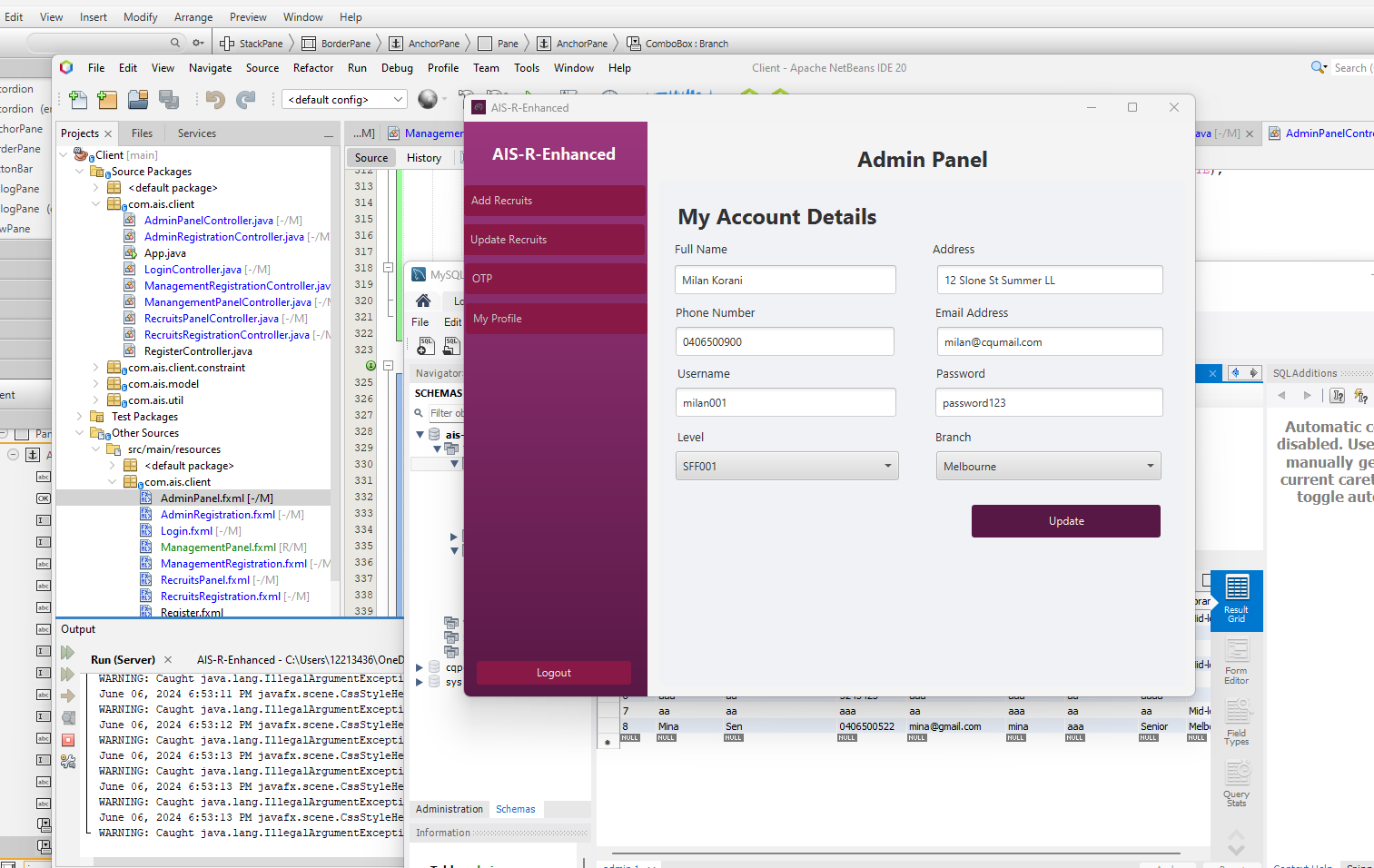
Description automatically generated

6.5 Programmatically database Table creation Testing

A screenshot of a computer

Description automatically generated

6.6 Update Own Profile Testing

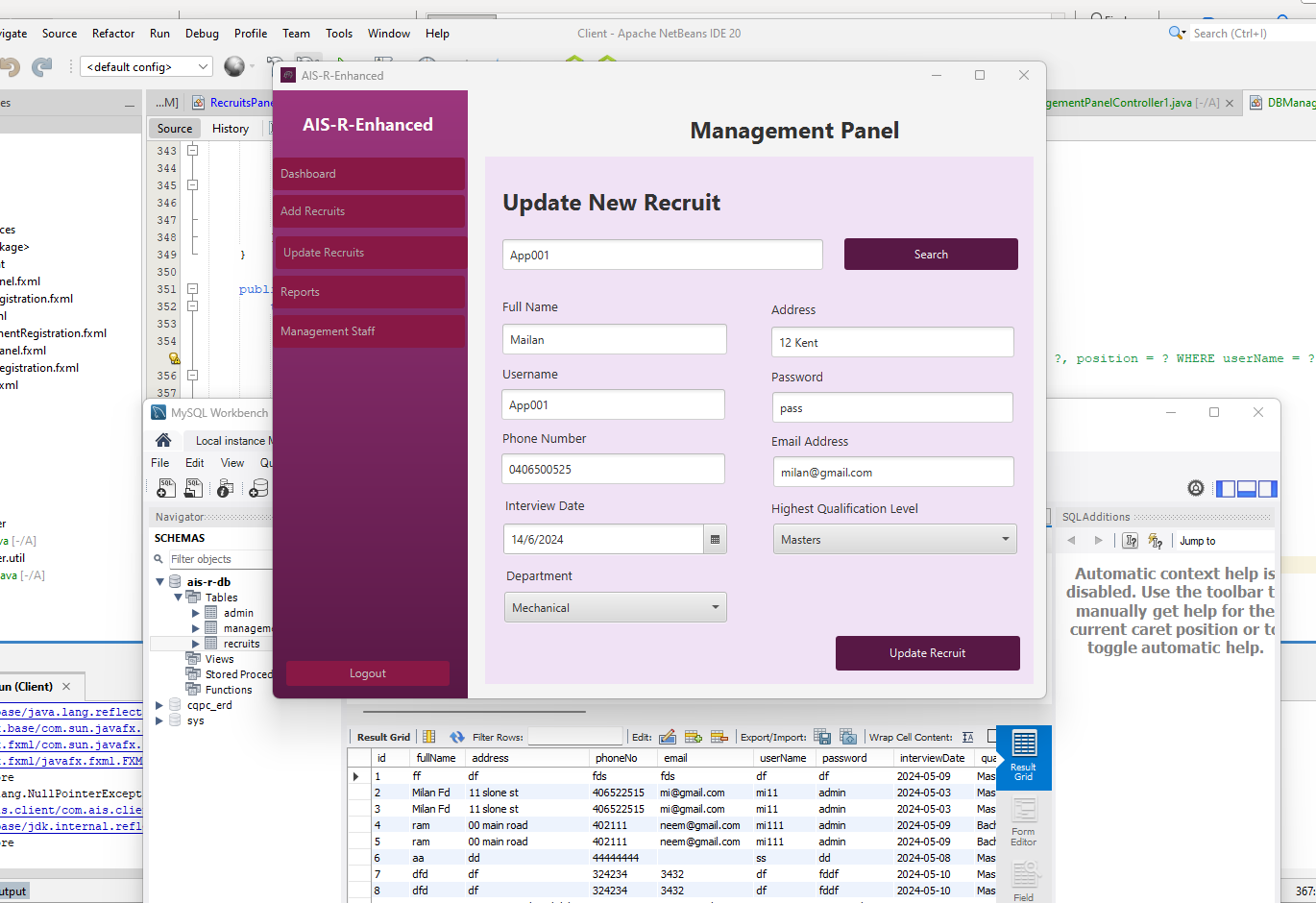


6.7 Update Recruit Data Testing

A screenshot of a computer

Description automatically generated

6.8 Department Assign Testing

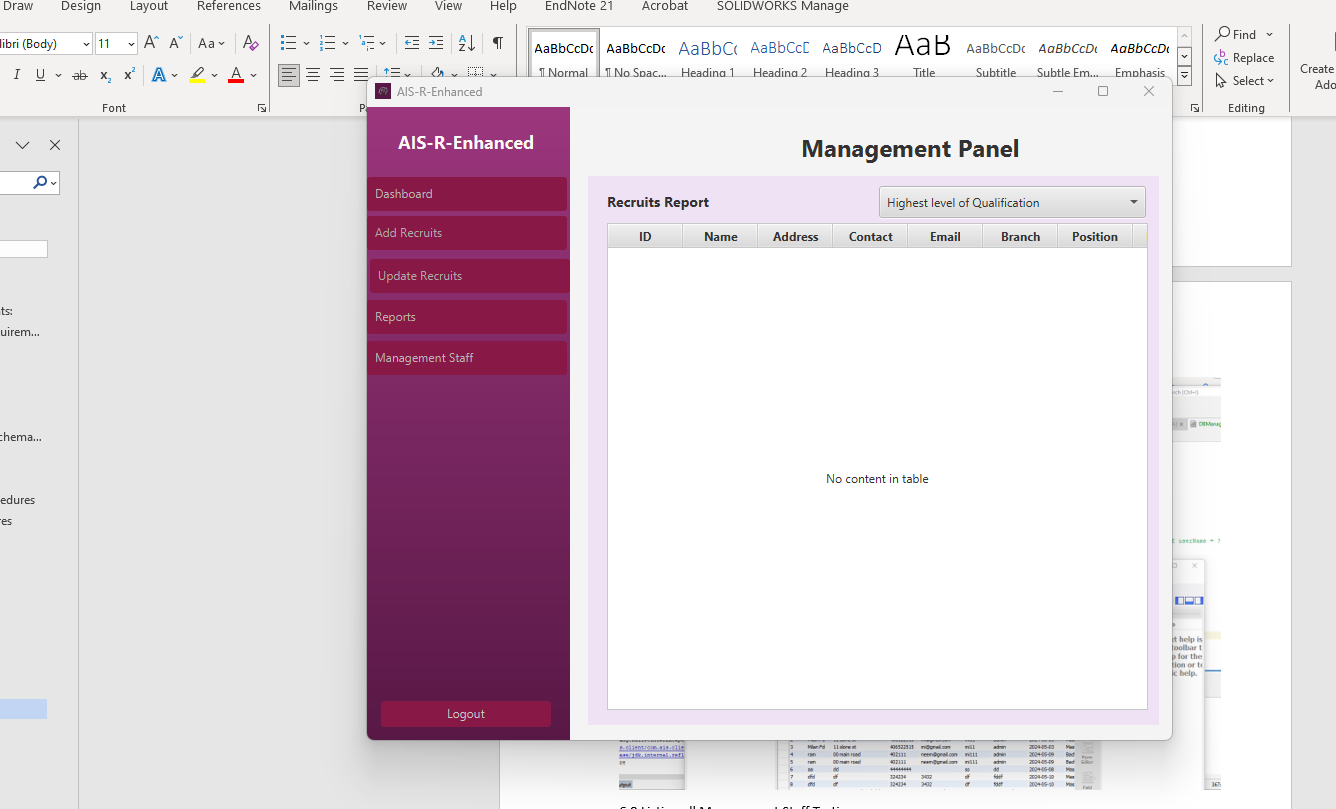


6.9 Listing all Management Staff Testing

A screenshot of a computer

Description automatically generated

6.10 Report



# **Conclusion**

In conclusion, the AIS-R-Enhanced software was developed with a strong focus on quality. We carefully checked the requirements, reviewed designs and code, and performed thorough testing to ensure the software meets all the needed standards. By following industry best practices and coding guidelines, we made the software strong and dependable. We also created detailed documentation to help users understand, maintain, and update the software in the future. AIS-R-Enhanced shows our dedication to providing a high-quality software solution that meets the needs of our users and supports future growth in aerospace technology.

Appendix

A screenshot of a computer

Description automatically generated