SCHOLAR'S UNIFIED PROGRESS & EFFICIENCY RESOURCE (SUPER)

CPIT 498 Report

By

Rawan Almeshari	2005140
Gadah Almuaikel	2005069
Bashaier Alamoudi	2009894
Shatha Alsulami	2006023

supervised By

Dr. Wafaa Alsaggaf

Department of Information Technology

Faculty of Computing and Information Technology

King Abdulaziz University

Jeddah – Saudi Arabia

[Fall 2023]

DECLARATION by AUTHORS

"I/we certify that this work has not been accepted in substance for any degree and is not concurrently being submitted for any degree other than that of BS Information Technology being studied at King Abdulaziz University, Jeddah. I/we also declare that this work is the result of my/our own findings and investigations except where otherwise identified by references and that I/we have not plagiarized another's work".




Rawan Almeshari | Gadah Almuaikel | Bashaier Alamoudi | Shatha Alsulami

---

DECLARATION by SUPERVISOR

I, the undersigned hereby certify that I have read this project report and finally approve it with recommendation that this report may be submitted by the authors above to the final year project evaluation committee for final evaluation and presentation, in partial fulfillment of the requirements for the degree of BS Information Technology at the Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah.

Dr. Wafaa Alsaggaf

Acknowledgment

We extend our profound gratitude to the Faculty of Computing and Information Technology members for their essential support and the wealth of knowledge they have shared, contributing significantly to the completion of our senior project.

Special acknowledgment is due to our supervisor, Dr. Wafaa Alsaggaf, whose invaluable guidance and persistent support have been the cornerstone of our project's development. Her expertise and keen insights were pivotal in refining our concept into the fully realized vision of SUPER.

We also convey our heartfelt thanks to the Information Technology Department faculty, whose expertise in system analysis, design, and implementation provided a robust foundation for our work.

Lastly, our gratitude extends to the senior project committee, whose constructive critiques and recommendations were instrumental in the project's evolution, pushing us to achieve excellence.

Together, these contributions have been integral to the SUPER project's success, and we are eternally thankful.

Abstract

This report presents the development of the Scholar's Unified Progress & Efficiency Resource (SUPER), a centralized web-based platform designed to enhance the PhD journey for students and coordinators at the Faculty of Computing and Information Technology, King Abdulaziz University. SUPER addresses the absence of a unified system for tracking academic progress, providing students with real-time updates on their progress and essential dates while enabling coordinators to access vital statistics and track student progress efficiently. Utilizing technologies like Laravel, Angular, and Bootstrap, SUPER offers a user-friendly interface with features like progress tracking, academic calendar integration, and secure authentication. The project adopts the Waterfall methodology for structured development, ensuring a reliable and effective solution for academic management. Its implementation promises to significantly improve the efficiency and effectiveness of PhD program management at the Faculty of Computing and Information Technology.

TABLE OF CONTENTS

List of Figures

List of Tables

Chapter I: Introduction

1.1 Introduction

In higher education, embarking on a Doctor of Philosophy journey is a noteworthy milestone, marked by its unique challenges. This chapter outlines how to make the PhD journey smoother for students and their coordinators.

1.2 Problem Definition

The main problem for PhD students and coordinators lies in the absence of a unified system to provide all the services they need. This leads to many challenges for the students, such as having no way to track their progress in the PhD program, which makes it hard to follow up on their progress, which may lead the students to exit the PhD program or have an academic suspension. Also, it is hard to know the expected graduation date; on top of that, students do not have notifications to know about important dates, like seminars and comprehensive exam dates.

On the other hand, coordinators also encounter a range of challenges; for example, they do not have a way to obtain statistics about the students, and instead, they need to make the statistics manually, which causes a massive amount of time and effort waste and puts it in the risk of mistakes happening. Also, coordinators cannot clearly track their student's progress, and for them to do that, they need to gather information about every student manually via emails and non-optimal ways, which makes it hard for coordinators to give the required guidance to students properly.

1.3 Project Scope

SUPER is a centralized web-based platform designed to simplify the PhD experience for students and coordinators at the Faculty of Computing and Information Technology (FCIT). This system enables students to track their academic progress and the coordinators to access essential statistics through a clear, organized dashboard by centralizing information and resources.

1.4 Project Aims

We aim to build a website to simplify the PhD process by offering a user-friendly website called Scholar's Unified Progress & Efficiency Resource (SUPER), where students track their progress, and coordinators get clear statistics on the students and the PhD program. The system is designed to make it easier to manage deadlines and requirements, streamlining the journey toward a PhD for all parties involved.

1.5 Project Objectives

- To create a centralized web-based platform that enables PhD students and coordinators to manage academic progress and requirements efficiently.
- To construct a comprehensive database within the system that holds all student-related information, allowing coordinators with the ability to pull various statistical reports.
- To design an interface for PhD students where they can visualize their progress, understand their standing in the program, and see the path to their graduation.
- To incorporate a notification system via e-mails and a calendar informing students about important dates and deadlines.
- To integrate a feature for coordinators to easily update information, input data, and track student progress in real-time.
- To share updates, changes in regulations, and other vital information to avoid any miscommunications.
- To develop a user-friendly and secure login system for students and coordinators to access their profiles and manage personal information effectively.
- To ensure the system is adaptable and scalable to meet future requirements, maintaining its utility and effectiveness over time.

1.6 Proposed Solution

The Scholar's Unified Progress & Efficiency Resource (SUPER) is designed to address the highlighted challenges by introducing a centralized web-based solution that serves as a one-stop hub for PhD students and coordinators. The solution encompasses several key features:

1. **Progress Tracking**: A personalized dashboard for each student showcasing a visual representation of their academic journey from admission to graduation. This tool will highlight completed and ongoing facilitating a clear view of the student's progress.
2. **Academic Calendar Integration**: SUPER will integrate an academic calendar to automatically update students about upcoming events, deadlines, and essential academic milestones.
3. **Statistical Dashboard Tools:** Coordinators will be provided with advanced visual dashboards that offer detailed statistics on student progress and program analytics with just a few clicks.
4. **Notifications**: A notification system will alert students via email about important dates, such as seminar schedules, comprehensive exams, and deadlines for paper submissions, reducing the risk of missed opportunities and about the remaining semesters they have.
5. **Real-time Data Management**: Coordinators can enter and update information in real-time, which is critical for keeping track of student progress and providing necessary academic guidance without delay.
6. **Secure Authentication**: A robust login and authentication system will protect users' privacy and safeguard their personal and academic information.
7. **Data Visualization Tools**: Graphs, charts, and other data visualization tools will be implemented to make understanding complex data more intuitive for coordinators.

By deploying these solutions, SUPER aims to revolutionize the PhD journey at the Faculty of Computing and Information Technology, making it a more manageable and enjoyable experience for students and coordinators alike. The platform will be designed with a focus on user experience, ensuring that all functionalities are accessible, intuitive, and responsive to the needs of its users.

1.6.1 Scenario of the solution

The diagram in Figure 1.1 depicts the SUPER Platform from the coordinator's perspective. Initially, the coordinator accesses the SUPER website and inputs the university ID and password. Upon successful verification, the coordinator gains access to the SUPER platform, enabling them to add and modify student information, conduct specific searches related to students and the doctoral program, filter data, request statistics, and send emails or notifications. The SUPER website is a valuable tool for coordinators to manage the doctoral program and monitor student progress effectively.

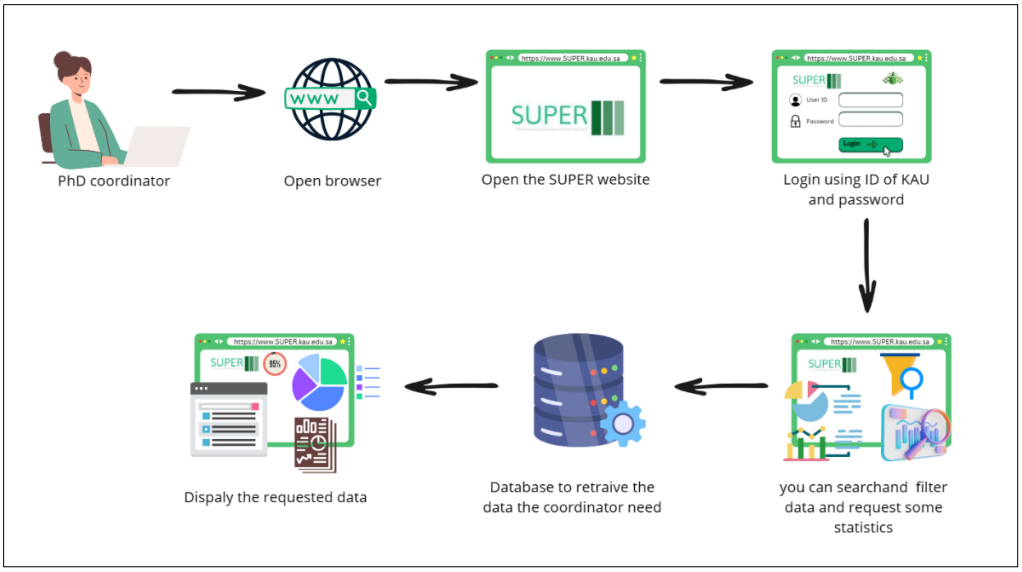

Figure 1.1: The Solution Scenario of SUPER (Coordinator)

On the other hand, in Figure 1.2, from the student's viewpoint, the process begins with the student visiting the SUPER website and entering their university ID and password. Upon successful login, the student can utilize SUPER to track their progress during their doctorate, view completed tasks, and stay informed about important dates. Additionally, the student can upload seminar papers and research papers and access information or guidelines related to the doctoral program. SUPER serves as a guiding tool for students to navigate their path toward graduation, ensuring a smooth doctoral journey.

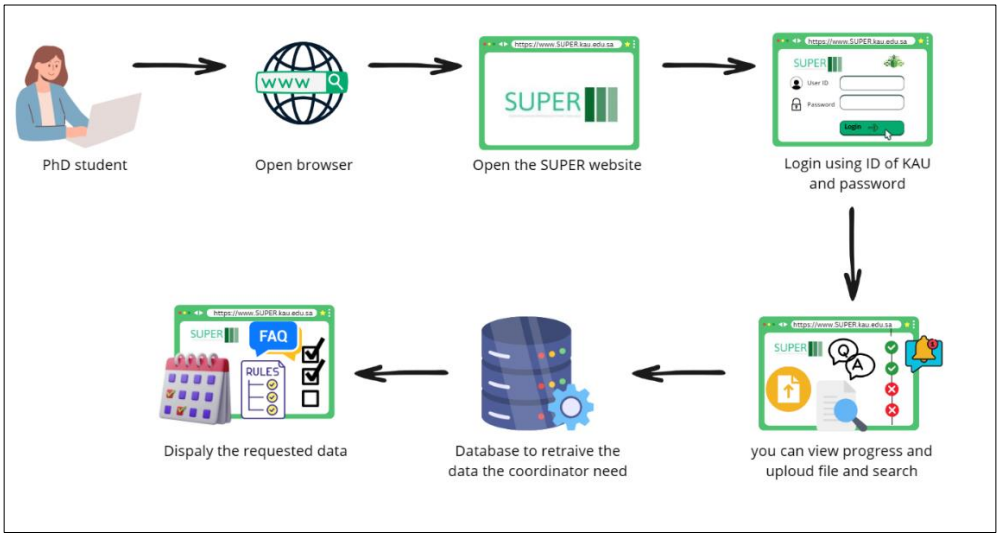

Figure 1.2: The Solution Scenario of SUPER (Students)

1.7 Methodology

The Waterfall software development methodology was chosen for the Scholar's Unified Progress and Efficiency Resource project and can be justified by several key points:

- **Clear Project Requirements**: The scope and requirements of SUPER are clear and detailed, derived from user interviews, making the Waterfall methodology ideal for its predictable and sequential approach.
- **Sequential Progression**: The project's nature allows for a sequential progression where each phase is fully completed before the next begins, aligning with Waterfall's structured framework.
- **Stable Project Scope**: The defined scope of SUPER, determined by specific academic and administrative needs, is stable and well-understood, fitting well with Waterfall's requirement of a stable scope.

The Waterfall model is depicted in Figure 1.3, showcasing the systematic progression of the software development process. This methodological choice is strategic for SUPER, providing a predictable and methodical development pathway. The initial portions of this report delve into the requirements and design phases of the Waterfall methodology, underpinning the critical groundwork for the subsequent development of the project.

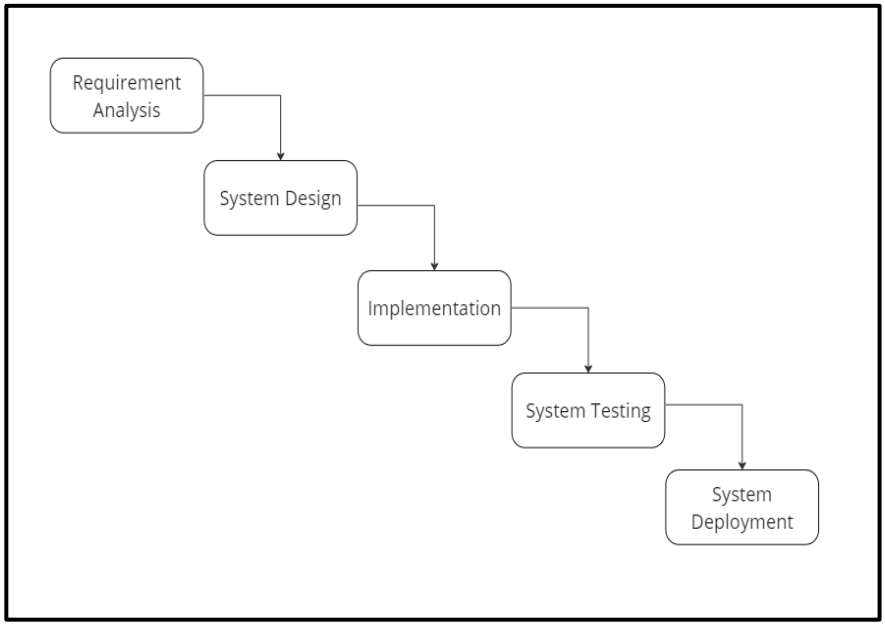

Figure 1.3: The Waterfall Methodology

A Gantt chart in Figures 1.4 and 1.5 is a powerful tool used in project management. It is a visual representation of a project schedule, where each task is displayed as a bar spanning from its start to its end date. The length of each bar reflects the duration of the task.

ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1		**SUPER**	**103 days**	**Sun 13/08/23**	**Thu 28/12/23**	
2		**First Semester**	**103 days**	**Sun 13/08/23**	**Thu 28/12/23**	
3		**Project Scope**	**14 days**	**Sun 13/08/23**	**Tue 29/08/23**	
4		determine the scope	4 days	Sun 13/08/23	Wed 16/08/23	
5		collecting information about similar system	5 days	Wed 23/08/23	Tue 29/08/23	4
6		**First Presention**	**12 days**	**Sun 03/09/23**	**Sat 16/09/23**	
7		Problem Definition	3 days	Sun 03/09/23	Tue 05/09/23	5
8		Similar Projects Review	4 days	Wed 06/09/23	Mon 11/09/23	7
9		IT Technology integration	2 days	Fri 15/09/23	Sat 16/09/23	8
10		Sumbit the first presentation	0 days	Sat 16/09/23	Sat 16/09/23	9
11		**Report**	**49 days**	**Mon 18/09/23**	**Thu 23/11/23**	
12		**Chapter1 [Introduction]**	**14 days**	**Mon 18/09/23**	**Thu 05/10/23**	
13		Problem Definition	4 days	Mon 18/09/23	Thu 21/09/23	01
14		Stakeholders	2 days	Fri 22/09/23	Mon 25/09/23	31
15		Suggested Solution	3 days	Tue 26/09/23	Thu 28/09/23	41
16		Methodology	3 days	Fri 29/09/23	Tue 03/10/23	51
17		Designing Tools	2 days	Wed 04/10/23	Thu 05/10/23	61
18		**Chapter2 [Background and Similar Appl**	**11 days**	**Fri 06/10/23**	**Fri 20/10/23**	
19		Background	3 days	Fri 06/10/23	Tue 10/10/23	71
20		Similar Works/Applications	3 days	Wed 11/10/23	Fri 13/10/23	91
21		Relevance	3 days	Mon 16/10/23	Wed 18/10/23	02
22		Comparative Study	2 days	Thu 19/10/23	Fri 20/10/23	12
23		**Chapter3 [Analysis]**	**13 days**	**Mon 23/10/23**	**Wed 08/11/23**	
24		Functional Requirements	4 days	Mon 23/10/23	Thu 26/10/23	22
25		Non-Functional Requirements	2 days	Fri 27/10/23	Mon 30/10/23	42
26		Initial Design / Analysis	4 days	Tue 31/10/23	Fri 03/11/23	52
27		Mapping Requirements	3 days	Mon 06/11/23	Wed 08/11/23	62
28		**Chapter4 [System Design]**	**8 days**	**Mon 06/11/23**	**Wed 15/11/23**	
29		Design Approach	4 days	Mon 06/11/23	Thu 09/11/23	62
30		Data Model Design	4 days	Fri 10/11/23	Wed 15/11/23	92
31		Appendix A [Data Gathering]	3 days	Thu 16/11/23	Mon 20/11/23	03
32		Final Report Format And Reference	3 days	Tue 21/11/23	Thu 23/11/23	13
33		Sumbit the final report	0 days	Thu 23/11/23	Thu 23/11/23	23
34		**Final Presention**	**4 days**	**Fri 24/11/23**	**Wed 29/11/23**	
35		Computer System Design Tools	2 days	Fri 24/11/23	Mon 27/11/23	33
36		Literature Review	2 days	Tue 28/11/23	Wed 29/11/23	53
37		Sumbit the final presentation	0 days	Wed 29/11/23	Wed 29/11/23	63

Figure 1.4: The Gantt Chart of SUPER Project

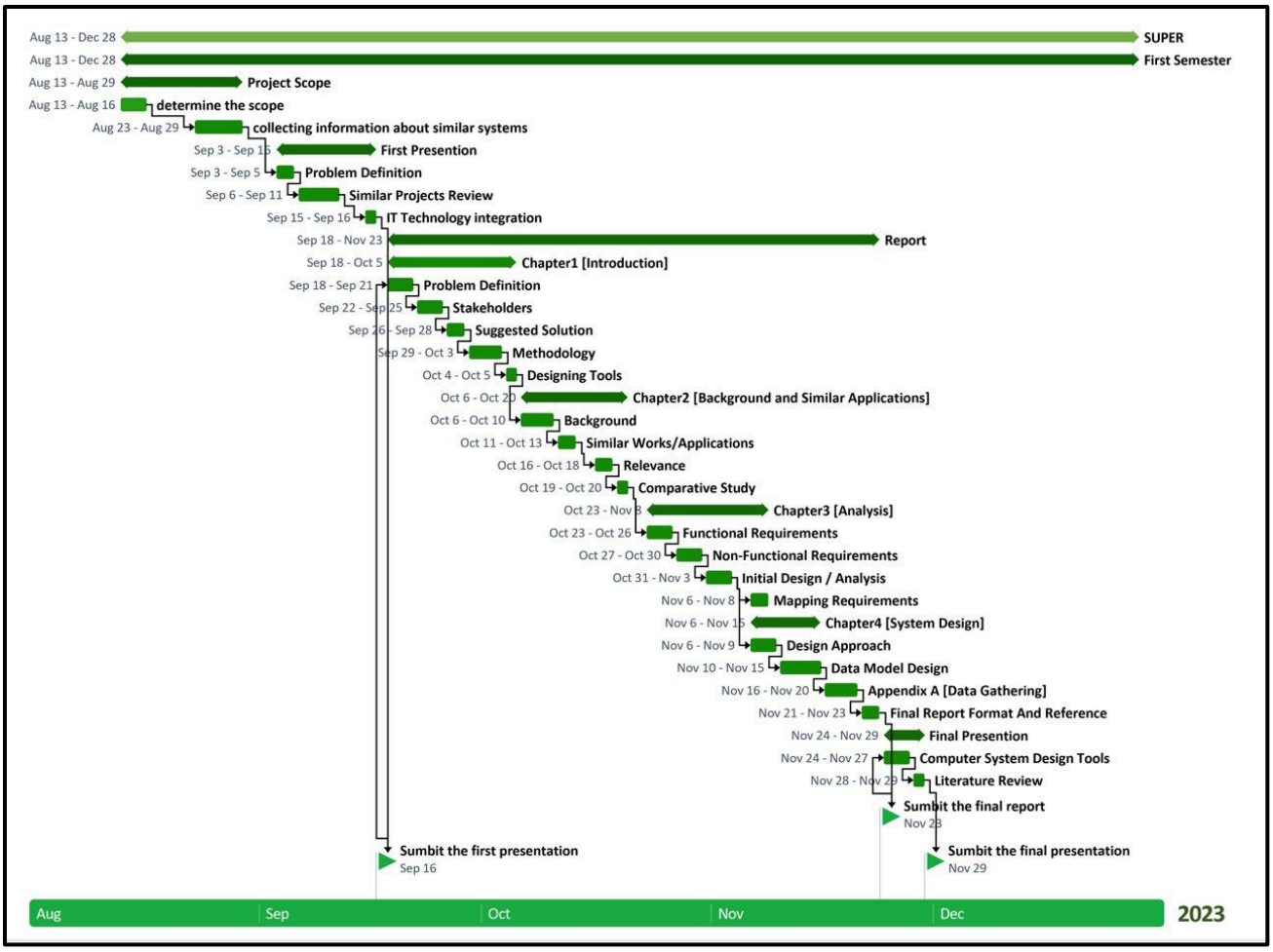

Figure 1.5: The Timeline of the Gantt Chart

1.8 Designing Tools

For the development of the SUPER project, various tools and technologies have been carefully selected for their effectiveness and compatibility with the project's objectives. Below is a list of the chosen tools, alongside their intended purpose within the project and their respective advantages.

- **Laravel (v10)**: To manage server-side operations and RESTful API services.
  Advantages:
  1. Robust security features protect against common web vulnerabilities [1].
  2. Elegant syntax and a rich set of features for rapid application development [1].

- **Angular (version 15):** To construct SUPER's dynamic and interactive user interface.
  Advantages:
  1. Facilitates the development of dynamic, single-page web applications [2].
  2. It offers powerful data binding, which reduces the need for manual DOM manipulation [2].

- **Bootstrap (version 5)**: To design a responsive and uniform user interface.
  Advantages:
  1. Accelerates UI development with a responsive grid system and pre-designed components [3].
  2. Ensures consistency across different browsers and devices with minimal effort [3].

- **Oracle Database (11g)**: For secure and scalable data management.
  Advantages:
  1. High performance and scalability for handling large volumes of data [4].
  2. Advanced features for robust data management and integrity [4].

- **MySQL**: To handle relational data storage and retrieval tasks.
  Advantages:
  1. Popular open-source RDBMS with strong community support [5].
  2. Easy to use and maintain with a comprehensive set of tools for database management [5].

- **Visual Studio Code:** As the code editor for streamlined project development.
  Advantages:
  1. Adaptable coding environment that accommodates a wide array of programming languages and add-ons [6].
  2. Built-in version control and numerous efficiency-enhancing tools simplify the development workflow [6].

- **Cascading Style Sheet:** For styling web pages and controlling layout.
  Advantages:
  1. Enables custom and creative designs for web pages.
  2. Responsive design capabilities ensure proper display on various devices.

- **JavaScript:** To add interactivity and real-time functionality to web pages.
  Advantages:
  1. Creates interactive elements on web pages, enhancing user experience.
  2. Wide support for modern web frameworks and extensive libraries.

- **Hypertext Markup Language**: To structure web content.
  Advantages:
  1. Fundamental building block of the web: defining structure and content.
  2. Seamless integration with other technologies for web development

- **Hypertext Preprocessor**: For scripting backend functionalities.
  Advantages:
  1. Wide hosting compatibility for diverse server environments.
  2. Large community and an extensive array of libraries for various functionalities.

- **TypeScript**: For enhanced code quality in Angular development.
  Advantages:
  1. Provides type safety for JavaScript, reducing runtime errors.
  2. Improves code structuring and readability, which is essential for larger projects.

1.9 Conclusion

In conclusion, Chapter I detailed the challenges in the PhD program and introduced SUPER, a solution aiming to streamline the academic journey through a centralized web platform. Adopting the Waterfall methodology ensures structured development while selecting modern tools and languages promises a secure, user-friendly system.

Chapter II: Background and Similar Applications

2.1 Introduction

This chapter reviews the literature on technologies and applications related to SUPER's domain, focusing on educational management systems for PhD tracking and coordination. This review helps situate SUPER within the existing academic and technological landscape, identifying gaps and informing its development.

2.2 Project Background

In a university setting, tracking programs can be particularly beneficial for both students and coordinators. Students can use these programs to monitor their academic performance, track their degree progress, and stay on top of their graduation requirements. This can help students identify potential challenges or areas for improvement and take proactive steps to address them.

For coordinators, tracking programs provide valuable insights into students' progress and performance. By accessing real-time data on students' academic performance, coordinators can identify at-risk students and provide the necessary support and resources to help them succeed. This proactive approach leads to higher retention rates and improved student outcomes.

For these reasons, we are developing a web platform, but before creating a web app or website, you need to decide what technology you will use. First, we use Laravel, an easy-to-use web framework, to help you create extensible PHP-based websites and web applications at scale [16].

The second is Angular, an open-source JavaScript framework written in TypeScript. Google maintains it, and its primary purpose is to develop single-page applications. As a framework, Angular has clear advantages while providing a standard structure for developers to work with [17].

The last framework is Bootstrap, a free front-end framework to make web development faster and easier. It also includes HTML and CSS-based design templates for forms, typography, buttons, navigation, tables, modals, image carousels, and many other components, along with other optional JavaScript plugins [18].

Finally, Combining Laravel for the backend, Angular for the front end, and Bootstrap for the UI can result in a well-structured and visually appealing web platform.

2.3 Literature Review

2.3.1 FCIT Coop

FCITCOOP, one of the Faculty of Computing and Information Technology central web platforms at King Abdulaziz University, adeptly tracks training progress and fosters seamless communication for administrative tasks. A user-friendly interface enables grade access, proactive learning, and efficient notifications, ensuring timely updates and deadline alerts. It offers a comprehensive overview of the academic journey and is an invaluable asset, harmonizing student engagement and academic oversight [7].

Relevance

- Track progress
- Submit reports.
- Grades page.
- Real-time visual of their training progress.
- Notification.

Difference

- Specific to training.
- Does not calculate expected graduation date.
- Does not calculate any statistics.

2.3.2 eProg (Manchester University)

The system facilitates a comprehensive tracking of the student's academic journey, capturing and documenting progress starting from registration through the submission of the Notice of Submission form. It also oversees the submission to the examination process, extending its monitoring capabilities from the submission of the form to the final ratification of your award. Additionally, the platform offers the convenience of scheduling researcher development training through eProg [8], allowing you to maintain a personal record of attended sessions visible to both you and your supervisors. This record proves valuable for enhancing your CVs or job applications.

Relevance

- Track progress.
- Submit reports.
- Notification.
- Calendar of deadlines.
- Calculate the expected graduation date.
- Show student status.

Difference

- Not for Kau students.
- Not have a dashboard.

2.3.3 ProDoc

The ProDoc system is a special system for the University of Twente (Netherlands) that allows for (and asks for) a yearly update of the Training and Supervision Plan in annual assessment meetings. In this way, the progress of the student- both in terms of research output and educational activities- is frequently monitored, and measures can be taken if progress is lacking [9].

Relevance

- Track progress.
- Submit reports.
- Calendar of deadlines.
- Show the student's status.

Difference

- No notification.
- Not for Kau students.
- Not have a dashboard.

2.3.4 POSTRACKER: Postgraduate Tracking System: Student Research Progress Tracking Tool

This automated tracking system effectively monitors the progress of postgraduate research students, consequently reducing the time required for them to complete their graduation requirements. Through a series of escalating "alerts," such information would notify the administrator if students were achieving their academic goals and allow the administrator to see if they are not progressing well. The tracker additionally benefits the postgraduate department by assisting in organizing educational activities, acting as a centralized source for accessing information on recommended topics available to supervisors, and generating standardized reports for quick and timely decision-making [10].

Relevance

- Track progress.
- Submit reports.
- Calendar of deadlines.
- Expected graduate date.
- The coordinator has a centralized information place.
- Notification.

Difference

- Not for KAU students.
- No grades page.

2.3.5 Graduate Student Tracking System (GSTS)

The GSTS (Graduate Student Tracking System) is a website for Northwestern University that allows programs to keep track of graduate student's academic progress and view academic milestones and related academic activity in one place. GSTS enables both programs and students to access student records information directly from CAESAR, along with supplementary data input by students and programs themselves [11].

Relevance:

- Plan of study, including coursework planned and completed (track progress).
- Annual progress.
- Milestones.
- The coordinator has a centralized information place.

Difference:

- Not for Kau students.
- No expected graduation dates.
- No notification.

2.4 Comparative Study

Table 2.1 presents the differences and similarities between the "SUPER" Website and its counterparts.

Table 2.1: Comparative Study

Features	FCIT Coop	eProg	proDoc	POSTRACKER	GSTS	SUPER
Track progress	✓	✓	✓	✓	✓	✓
Notification	✓	✓	X	✓	✓	✓
Submit reports	✓	✓	✓	✓	✓	✓
Dashboard	X	X	X	✓	✓	✓
Kau students	✓	X	X	X	X	✓
Students Statistical	X	✓	X	✓	X	✓

The in-depth analysis of platforms such as FCIT Coop, eProg, ProDoc, POSTRACKER, and GSTS has provided invaluable insights for the development of the SUPER project. This research has highlighted SUPER's strengths and areas for improvement, particularly in understanding the specific needs of our target users. By examining these systems, we have gained essential knowledge in critical areas like progress tracking, report submission, and deadline management.

The comparative study of these platforms has also revealed opportunities for SUPER to differentiate itself, such as through enhanced user interfaces, real-time notifications, and tailored features for King Abdulaziz University students. This exploration has sparked innovative ideas for streamlining administrative tasks and improving student engagement.

Overall, the insights from this research are crucial in refining SUPER's functionalities, ensuring it meets the unique requirements of its users and establishes itself as a comprehensive and effective academic tracking and management tool.

2.5 Conclusion

In summary, SUPER integrates advanced technologies for efficient PhD tracking and management, distinguishing itself with features like real-time progress tracking, customized for King Abdulaziz University students. Its unique blend of user-friendly interfaces and secure data management positions SUPER as a cutting-edge solution in educational management systems.

Chapter III: Analysis

3.1 Introduction

Chapter III examines the academic management system tailored for PhD students and coordinators. It defines the system requirements for optimal operation, usability, and security. The chapter provides a use case diagram to illustrate user interactions and maps specific functionalities to user roles. It ensures that the system's design meets educational and administrative needs effectively.

3.2 Functional Requirements

User Access & Profile Management (Core Functionality):

- The user shall be able to log onto the platform using their university ID and password.
- The user shall be able to view and modify their personal profile information.
- The user shall be able to recover their forgotten password through their registered email.
- The user shall receive an email containing the login password, which can be used in to log in.

PhD Students Academic Management (Core Functionality for Students):

- The PhD student shall be able to view a real-time visual representation of their academic progress.
- The PhD student shall be able to receive notifications from the coordinator.
- The PhD student shall be able to view the academic calendar on the website.
- The PhD student shall be able to see the semester they finished, the remaining ones, and how many semesters they can postpone.
- The PhD student shall be able to upload and submit their publication.
- The PhD student shall be able to view the upcoming seminars.
- The PhD student shall be able to view the FAQs page.
- The PhD student shall be able to see the expected graduation year.

PhD Coordinator Academic Administration (Core Functionality for Coordinators):

- The PhD coordinator shall be able to access a dashboard that aggregates data about all student's progress, facilitating better decision-making.
- The PhD coordinator shall be able to view a list of all PhD students under their department.
- The PhD coordinator shall be able to enter all the important dates for the academic calendar.
- The PhD coordinator shall be able to enter the information for the upcoming annual seminar.
- The PhD coordinator shall be able to enter student's degrees in the comprehensive exams.
- The PhD coordinator shall be able to add a comprehensive exam track type.
- The PhD coordinator shall be able to assign each student to a comprehensive exam track type.
- The PhD coordinator shall be able to filter statistics and specific student information.
- The PhD coordinator shall be able to view the student's publication.
- The PhD coordinator shall be able to manage the student data, add and edit it, and delete it.
- The PhD coordinator shall be able to provide resources, set regulations, or update any information relevant to the PhD program as a Question-and-answer pair.
- The PhD coordinator shall be able to enter the rules and regulations.
- The PhD coordinator shall be able to search for specific students, view their profiles, and monitor their academic journeys.
- The PhD coordinator shall be able to send notifications for groups of students grouped by standard criteria.
- The PhD coordinator shall be able to accept newly registered students into the system.
- The PhD coordinator can view student information to track their progress status.

3.3 Non-Functional Requirements

3.3.1 Performance

- The website will provide fast response times, ensuring all web pages and functionalities load quickly.
- The website will handle large volumes of data, ensuring smooth performance without delays [12].

3.3.2 Usability

- The website will provide a user-friendly interface for easy navigation.
- The website will provide all the requirements of UI/UX [12].

3.3.3 Compatibility

- The website will be compatible with various operating systems
- The website will be compatible with major web browsers [12].

3.3.4 Security

- The website will ensure data privacy and protection for all users.
- The website will provide User authentication, preventing unauthorized access [12].

3.3.5 Scalability

- The website will be capable of handling an increasing number of users and data without performance degradation.

3.3.6 Maintainability

- The website will have a modular architecture for easy update and maintenance.
- The website will be accompanied by comprehensive documentation for future developers, ensuring ease of understanding and efficient maintenance [12].

3.4 Project Stakeholders

Table 3.1 below identifies the key stakeholders of the SUPER project, detailing their respective roles and the value they bring to the effectiveness and progression of the academic management system.

Table 1: 3.1: Project Stakeholders with Description

Stakeholders	Description
PhD Students	They are the primary users of SUPER, relying on it to track their academic journey, understand requirements, and stay updated on important deadlines and events.
PhD Coordinators	Administrative staff responsible for managing the PhD program. They will use SUPER to monitor student progress, and support students more effectively.
e-college	University leaders who use SUPER's reporting tools to make decisions about the PhD program, allocate resources, and ensure program quality.
Software Developers/ Project Team	The team that develops, updates, and refines SUPER using stakeholder feedback to improve functionality and user experience.
Project Owner	The person or entity responsible for the SUPER project's overall success, providing vision, direction, and resources and making strategic decisions to ensure the project achieves its goals efficiently and effectively.

3.5 Initial Design / Analysis

3.5.1 Use Case Diagram

A use case is a way to organize, define, and identify system needs in system analysis. A use case is a collection of potential interactions between people and systems in a particular setting that aims to achieve a specific objective [13]. The process generates a document listing every user action to finish a task. Figure 3.1 applies this concept to demonstrate how PhD students and coordinators interact with the system.

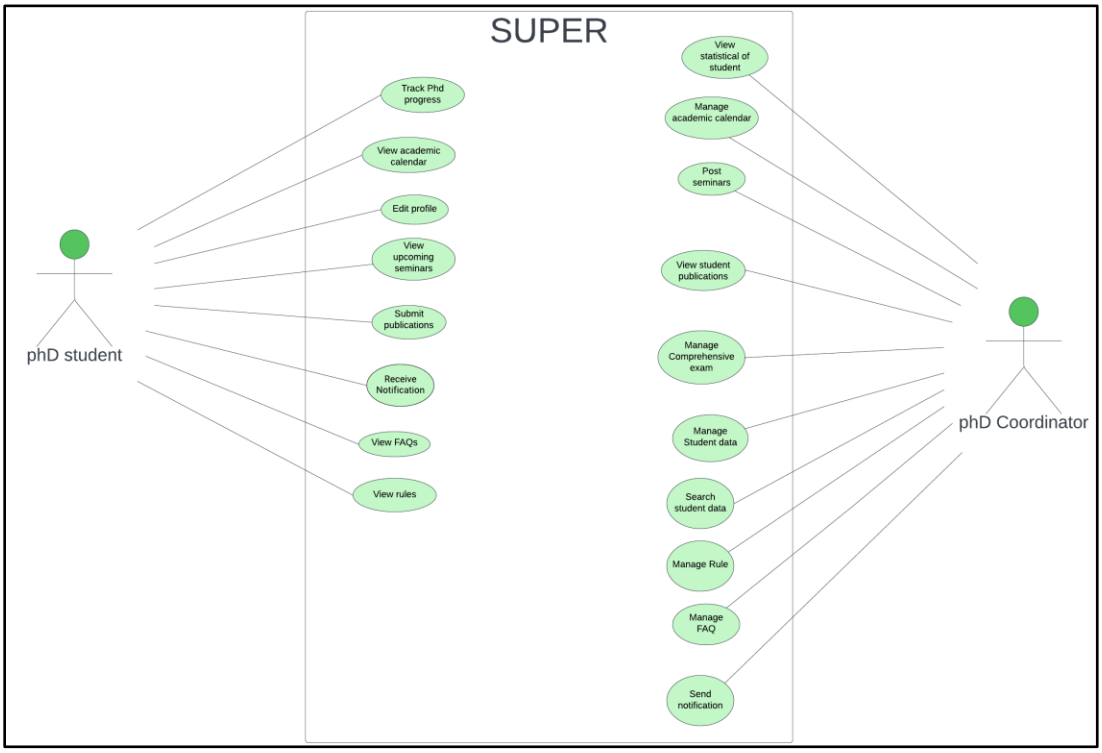

Figure 3.1: Use Case Diagram for SUPER

3.5.1.1 Use Case Description

Actor: PhD Student

1. **Track PhD Progress:** Allows the student to monitor their academic journey, visualizing milestones and completed requirements.
2. **View Academic Calendar:** Students can check the schedule for the academic year, including deadlines and holidays.
3. **Edit Profile:** This function allows students to update their personal and contact information.
4. **View Upcoming Seminars:** Students can see schedules for future seminars that may be relevant to their research or academic interests.
5. **Submit Publications:** Students can submit their research papers or publications through the system.
6. **View Rules:** Students have access to the program's rules and requirements.
7. **View FAQs:** Frequently Asked Questions about the program are available for student reference.
8. **Receive Notification:** Students can receive personalized alerts and messages about their academic progress or program updates.

Actor: PhD Coordinator

1. **Manage Academic Calendar:** The coordinator can update and manage the academic calendar events.
2. **View Statistics of Students:** The coordinator can access statistical data and student progress analytics.
3. **Post Seminars:** Allows the coordinator to add and announce upcoming seminars.
4. **View Student Publications:** Access to view academic papers or research publications submitted by students.
5. **Manage Comprehensive Exam:** The coordinator can oversee and organize details related to comprehensive exams.
6. **Manage Student Data:** This includes adding, editing, or deleting student records in the system.
7. **Search Student Data:** A search function for locating specific student information or records.
8. **Manage FAQs:** The coordinator can update or add to the list of Frequently Asked Questions.
9. **Manage Rule:** The coordinator can edit or update the program's rules and regulations.
10. **Send Notification:** Functionality to send out notifications to students.

3.6 Mapping Requirements

Our academic management system's functional requirements mapping in Table 3.2 clearly outlines the specific actions available to different user roles: PhD Students and the PhD Coordinator. As seen in the accompanying use case diagram, this visual representation ensures that the system meets each group's distinct needs, facilitating student engagement and administrative oversight. It is essential to ensure that the system's design corresponds to the intended academic and administrative functions.

Table 2: 3.2: Mapping Requirements Table

Use Case/Functional Requirements.	User Access Profile Management	PhD Students Academic Management	PhD Coordinator Academic Administration
Track PhD progress		✓	
View academic calendar		✓	
Edit profile	✓		

			Continue
Continuation			
View upcoming seminars		✓	
Submit publications		✓	
View Rules		✓	
View FAQs		✓	
Receive Notification		✓	
View Statistical of students.			✓
Manage academic calendar			✓
Post seminars			✓
View student publications			✓
Manage Comprehensive exam			✓
Manage student data			✓
Search student data			✓
Manage FAQs			✓
Manage Rule			✓
Send Notification			✓

3.7 Conclusion

In summary, Chapter III offers a brief examination of the academic management system, emphasizing the users' requirements and the system's capabilities. It creates a user-centered framework that promises to be effective and simple for coordinators and PhD students to use. The next development phase is made possible by the clearly defined requirements and graphic diagrams, which offer a solid foundation for academic administration.

Chapter IV: System Design

4.1 Introduction

The design chapter is a critical component in developing our proposed system. It is a comprehensive display of the system's design, defined through detailed diagrams. These visual representations, which include a class Diagram, Sequence Diagrams, and an entity-relationship (ER) diagram, are accompanied by descriptive narratives. They provide a straightforward and logical illustration of the system's overall structure and the complex interactions among its various components. This chapter reflects the system's design structure and lays the foundations for its effective implementation.

4.2 Design Approach

The Object-Oriented (OO) design approach was chosen for 'SUPER' due to its excellence in encapsulation and modularity, creating a structured and manageable codebase. This methodology is especially beneficial for complex and scalable projects like 'SUPER,' as it facilitates intuitive system modeling based on real-world scenarios organized into classes and objects [14]. The object-oriented approach's focus on reusability and scalability through class-based structures aids in seamlessly integrating new features and minimizing disruptions to the existing system. Its emphasis on limiting changes to specific objects enhances software quality and error resilience. Adopting Object-Oriented design, renowned for its comprehensive design patterns and best practices, guarantees robust, efficient, and secure development, aligning with 'SUPER's ambitious goals.

4.*2.1 Class Diagram*

Figure 4.1 presents the UML Class Diagram for our project, providing a visual representation of key classes, their relationships, dependencies, and methods. The diagram includes the User class, serving as a generalization of Student and Coordinator subclasses. It also showcases classes like ComprehensiveExam, Rule, Task, Event, FAQ, Seminar, Publication, Notifications, and DatabaseHandler. The DatabaseHandler class connects to all other classes, facilitating data modification and retrieval operations within the system. This diagram simplifies the software structure and highlights interactions between various components.

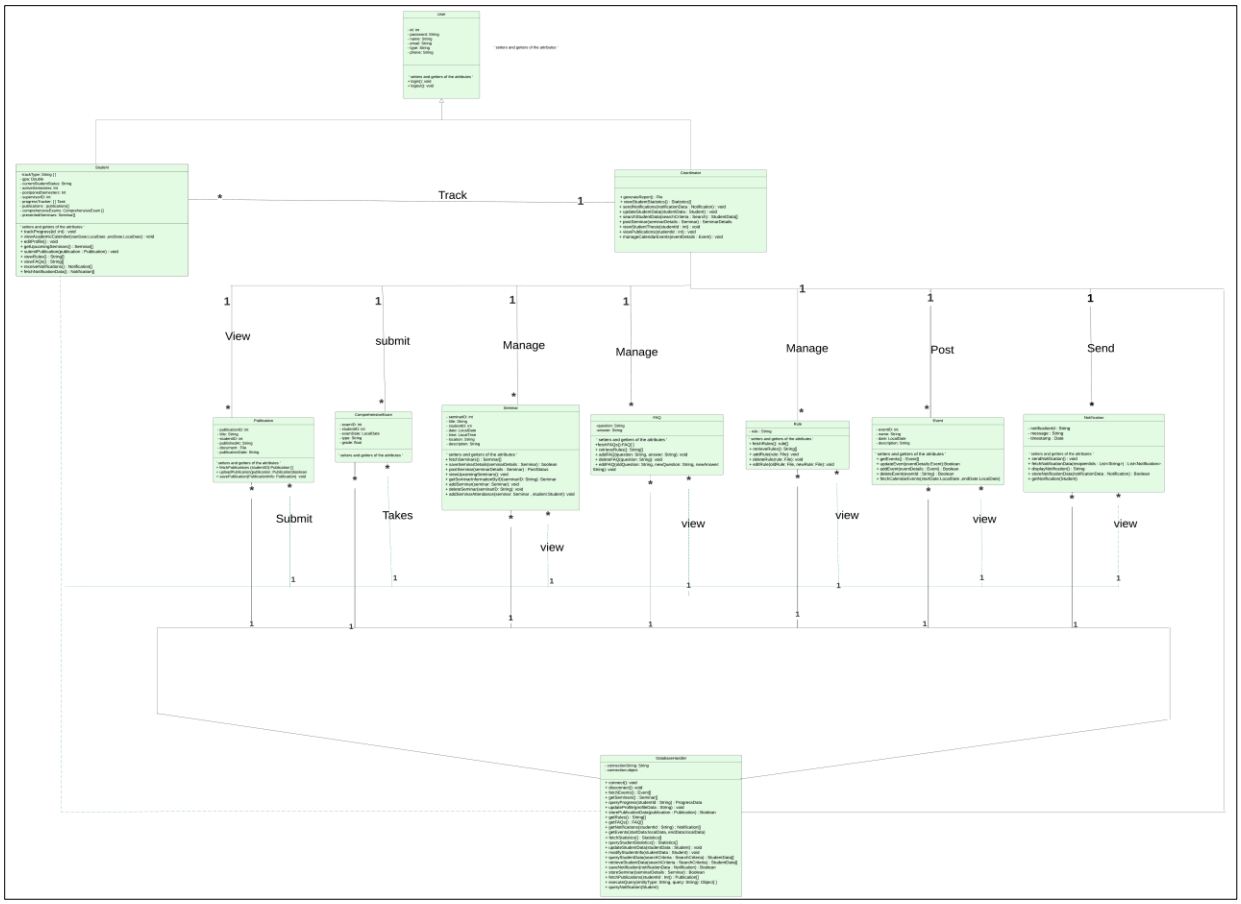

Figure 4.1: Class Diagram

The student class includes methods such as trackProgress, viewRule, and submitPublication, which enable it to interact with other classes and ensure seamless functionality, as illustrated in Figure 4.2.

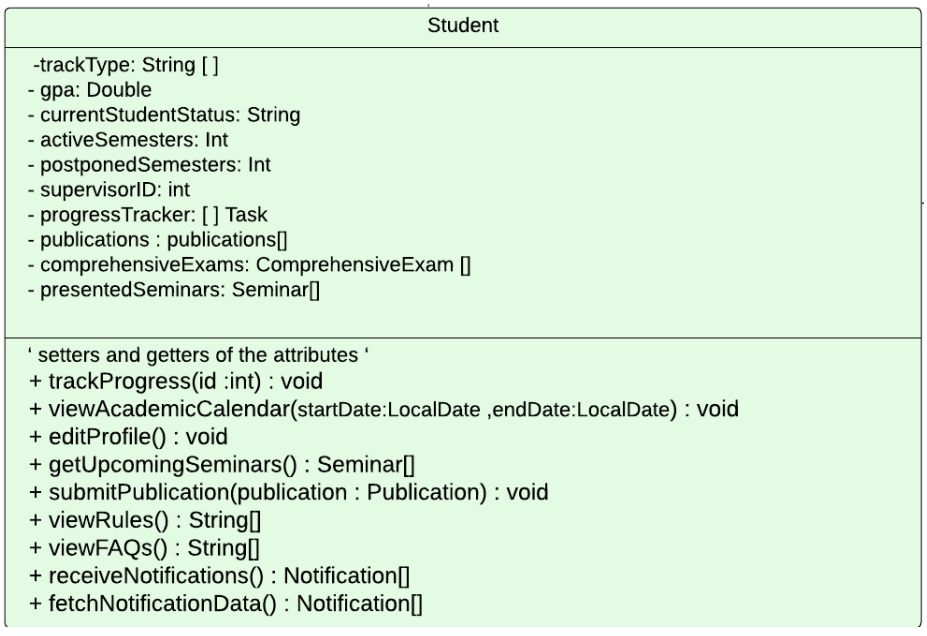

Figure 4.2: Student Class

The coordinator class comprises methods like viewStudentStatistics, PostSeminars, and manageCalendarEvents, as shown in Figure 4.3. This design promotes efficient communication between the Student and Coordinator classes and other system components.

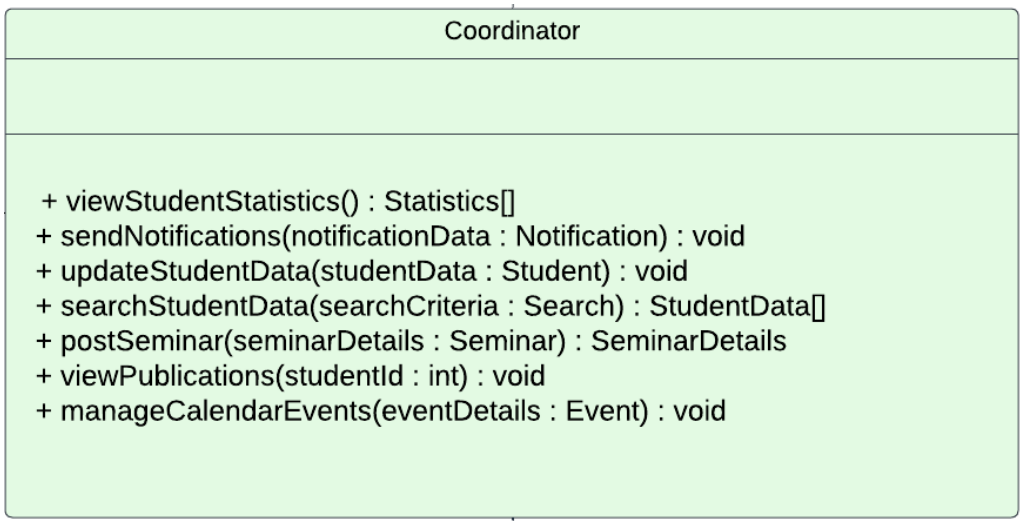

Figure 4.3: Coordinator Class

4.2.2 Sequence Diagram

A sequence diagram is designed to represent a timeline that starts at the top and gradually descends to mark the sequence of interactions. Each object has a column, and arrows represent the messages they exchange. The sequence diagram aided in representing the sequence of interactions of our system's objects and their relationships [19].

4.2.2.1 Coordinator's Sequence Diagram Part One

The sequence diagram in Figure 4.4 illustrates the interactions of a PhD coordinator with various academic subsystems through a Super. The coordinator's role involves querying student statistics, handling event and seminar details, and viewing publication information. Actions performed by the coordinator prompt responses from the Event, Seminars, and Publication subsystems, with a DatabaseHandler facilitating data transactions. This streamlined diagram represents the coordinator's management of critical academic operations within an institutional framework.

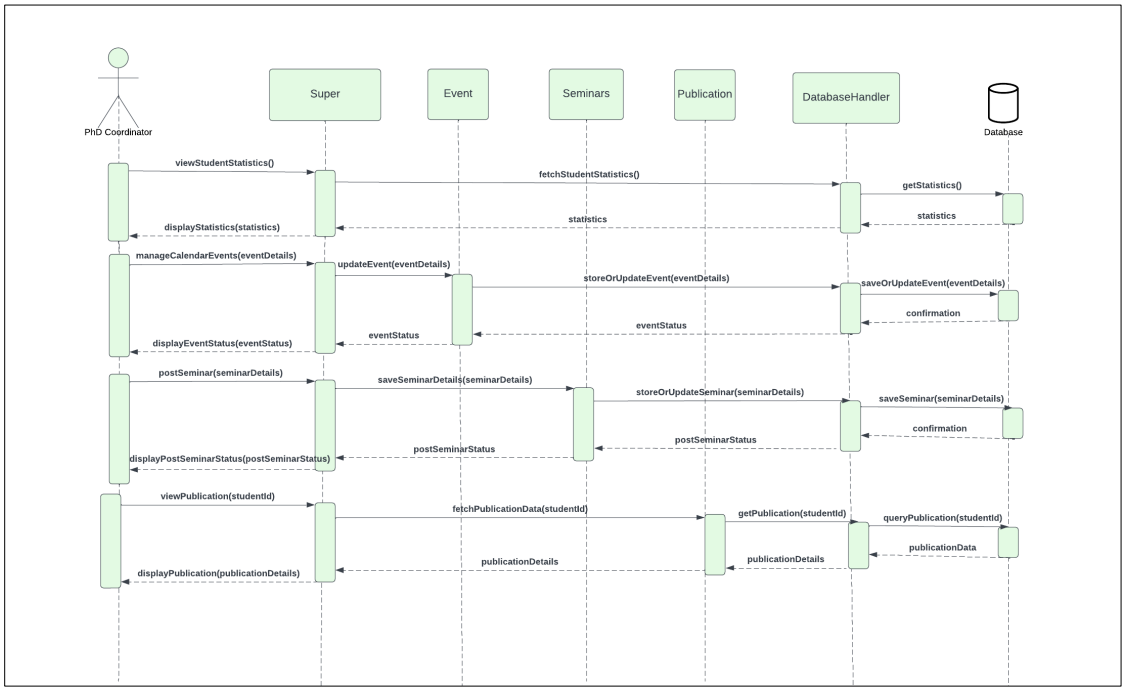

Figure 4.4: Coordinator's Sequence Diagram Part One

4.2.2.2 Coordinator's Sequence Diagram Part Two

The sequence diagram in Figure 4.5 shows how a PhD coordinator uses a Super System to communicate with other academic subsystems. The coordinator oversees the specifics of the comprehensive exam and student data, searches for student data, updates the FAQ, and makes changes to the rules. Every action she takes sets off reactions from subsystems such as comprehensive exams, Rules, and FAQs. The flow is like the previous diagram, "coordinator part one," demonstrating a continuation of the coordinator's management of various academic tasks, with the DatabaseHandler once again mediating data operations. This section expands on the coordinator's responsibilities for maintaining academic standards and communicating information throughout the program.

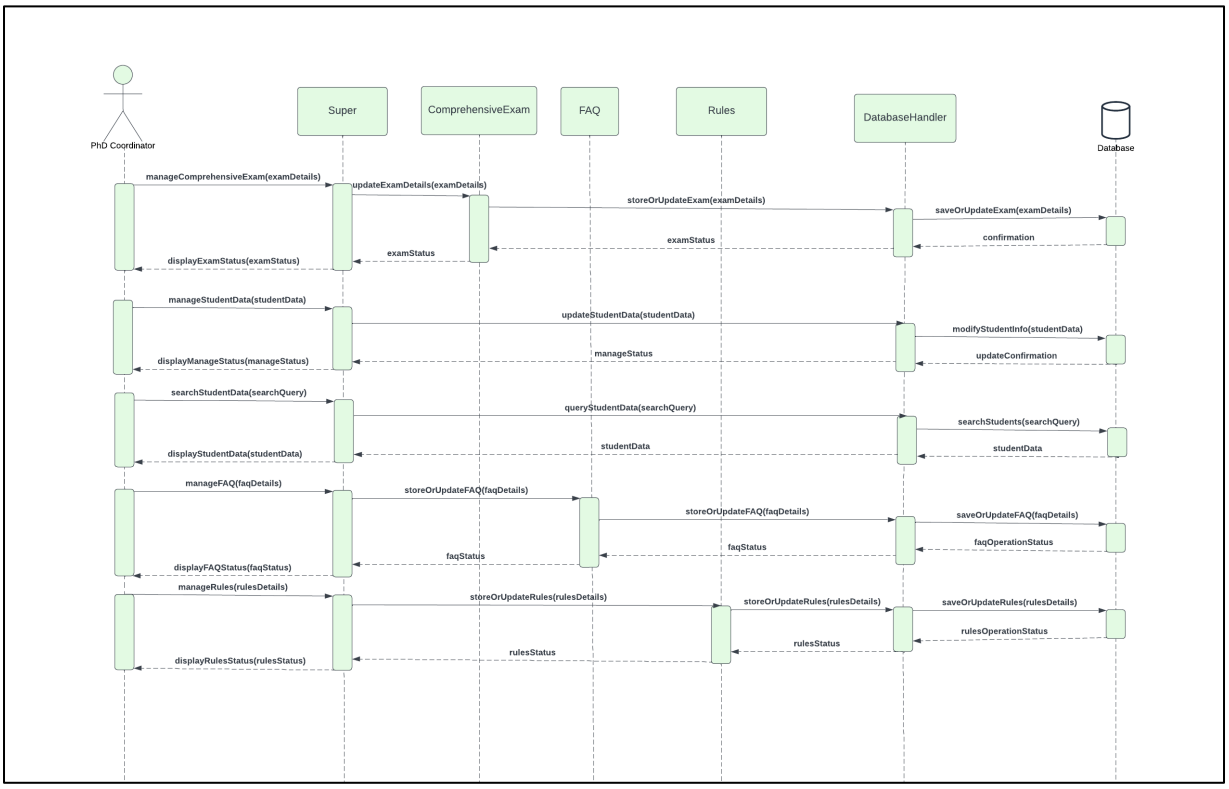

Figure 4.5: Coordinator's Sequence Diagram Part Two

4.2.2.3 Student's Sequence Diagram

In Figure 4.6, the PhD student is the primary actor, interacting with a Super System to perform various academic tasks. The student tracks progress, views academic calendar events, looks up upcoming seminars, submits publications, and reviews academic rules and FAQs. Each action corresponds with a subsystem such as Task, Event, Seminars, Publication, Rules, and FAQ, leading to data retrieval or storage facilitated by the DatabaseHandler. The diagram illustrates a comprehensive system enabling students to manage their academic journey, from tracking progress to accessing crucial academic information.

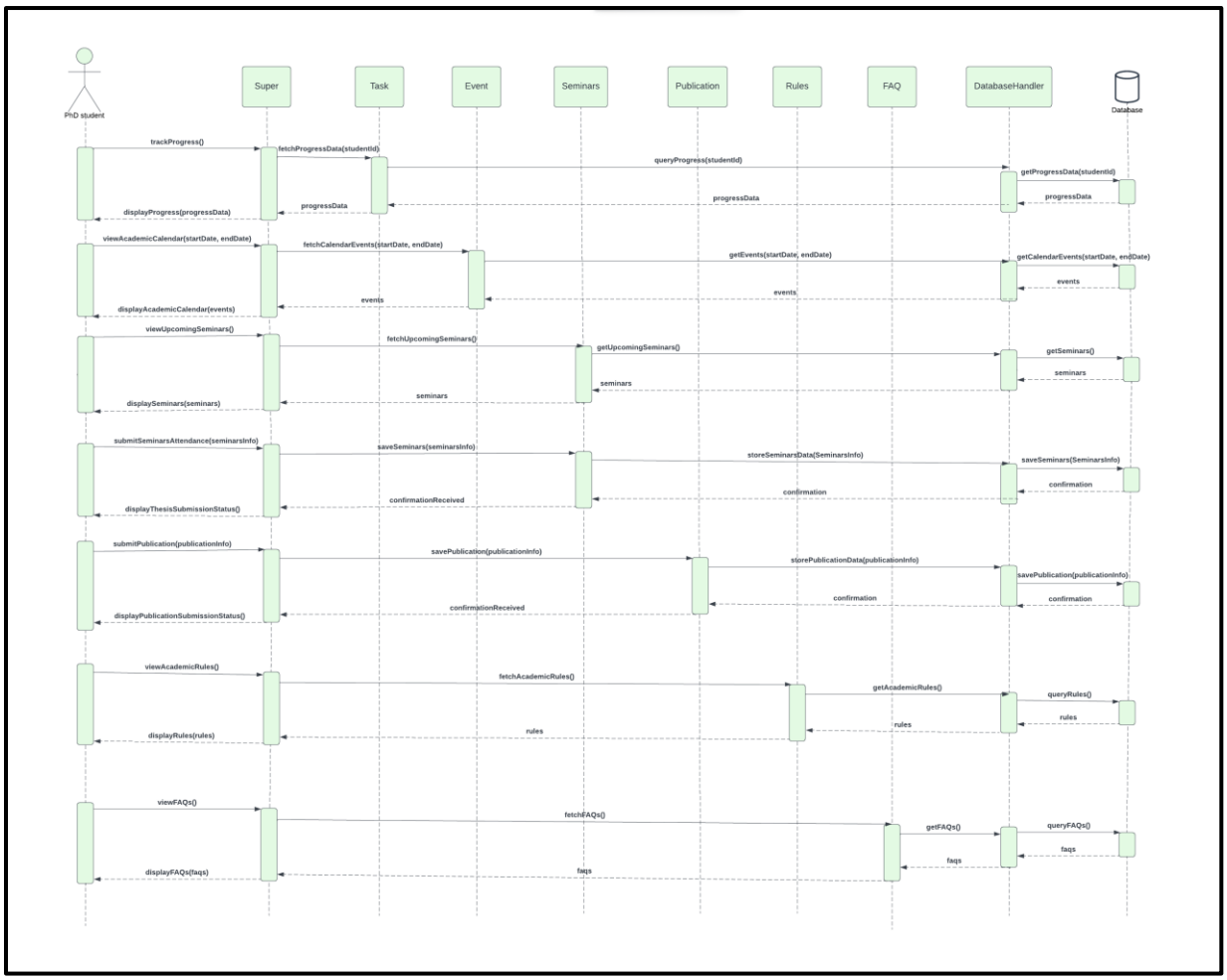

Figure 4.6: Student's Sequence Diagram

4.2.2.4 Notification's Sequence Diagram

The sequence diagram in Figure 4.7 illustrates the process of sending notifications to students. The coordinator can notify students regarding upcoming exams. Upon adding a new exam and enrolling students in the exam, the coordinator can dispatch notifications to all enrolled students. These notifications are transmitted directly through a super system, which interfaces with the pertinent subsystems. Confirmations of these actions are recorded in the database, and a notification system is activated to dispatch emails to doctoral students, ensuring that they stay informed about academic events promptly.

Additionally, the super system also sends notifications to students required to present at an annual seminar. The system autonomously performs calculations to identify the students and subsequently dispatches notifications to them.

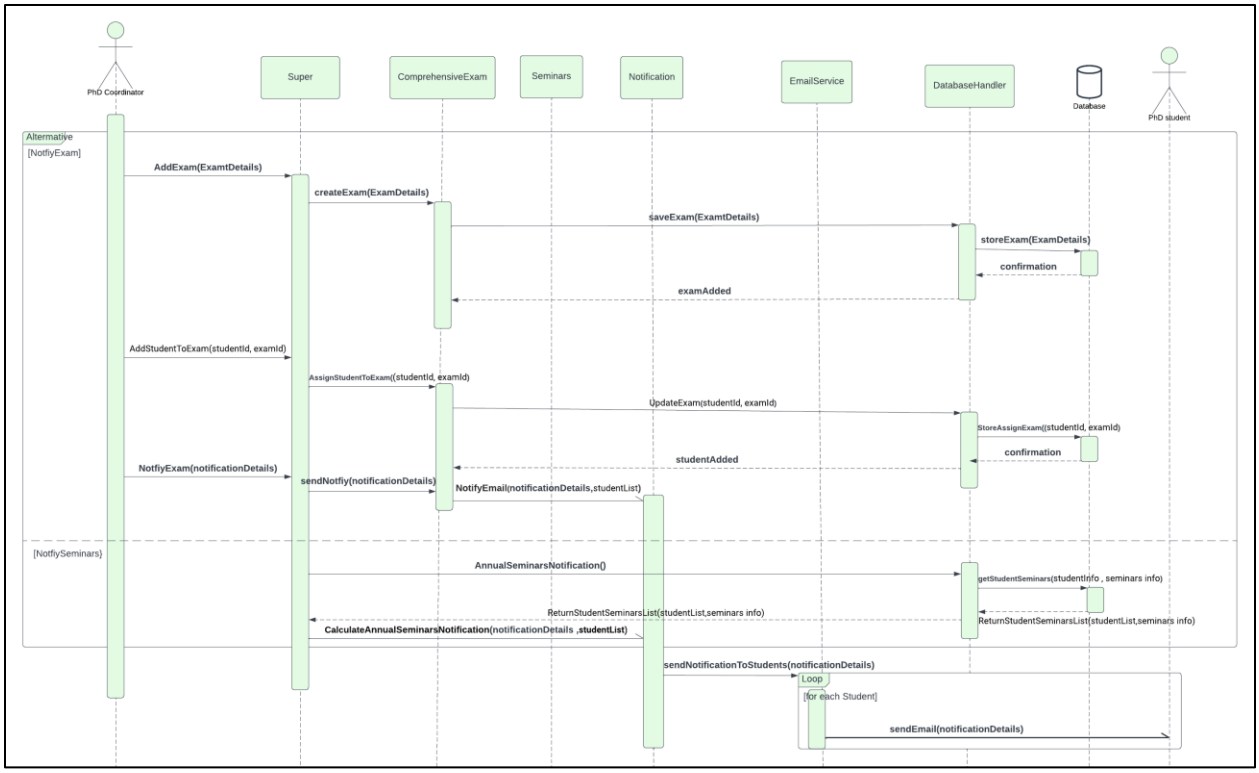

Figure 4.7 Notification's Sequence Diagram

4.3 Data Model Design

4.3.1 Entity Relationship Diagram (ERD)

The virtual database for the SUPER system is displayed in Figure 4.8 The entity relationship diagram [15] displays the tables that make up the system's database and the relationships between them.

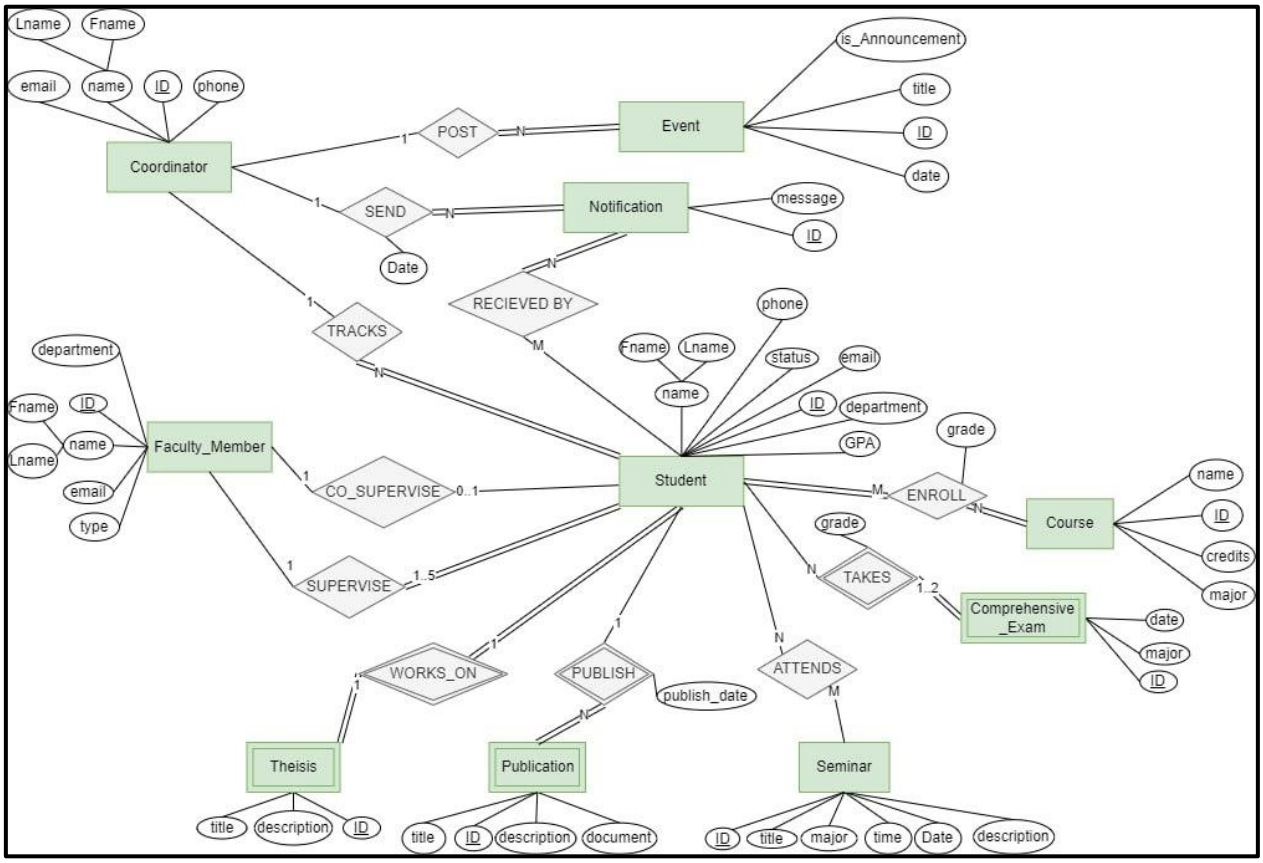

Figure 4.8: Entity Relationship Diagram (ERD) for SUPER

4.3.2 Normalization

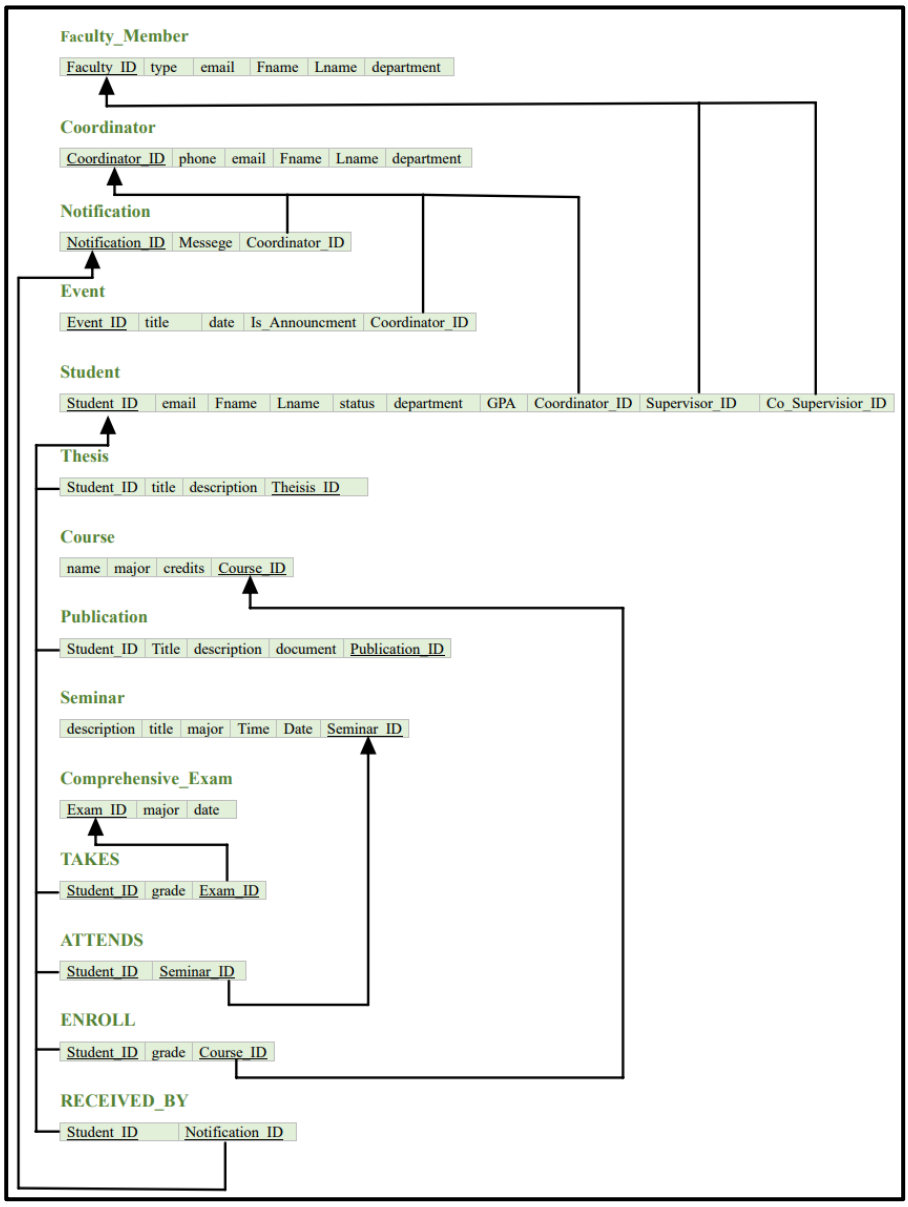

Figure 4.9 Normalized ER Diagram for SUPER

4.4 Conclusion

In conclusion, the design chapter effectively presented 'SUPER's design by utilizing the Object-Oriented approach and various diagrams such as class, sequence, and ER diagrams. These elements, taken together, provide a clear picture of the system's structure and interactions. This foundation lays the foundation for 'SUPER's efficient development and future scalability.

Chapter V: Implementation

5.1 Introduction

The implementation phase of the SUPER is described in this section. The section includes three sections: client-side implementation, database implementation, and server-side implementation.

5.2 Client-Side Implementation

The website was built using the Angular JS framework. It consists of several components that have separate HTML templates that are powered by Angular, CSS styling, and TypeScript to handle logic.

Notable components are:

- **Layout components:** header, sidebar, dialogs, modals
- **Authentication components:** login
- **Student components:** information, seminars, calendar, publications, attendance, FAQ
- **Coordinator components:** calendar, comprehensive exam, dashboard, public seminars, publications, FAQs.

The services are also included to facilitate communication with the server, fetch data, update data, store data, and guard pages for access to relevant users.

The services are:

- Authentication service.
- guard service.
- exam service.

```typescript
import { Injectable } from '@angular/core';
import { Router } from '@angular/router';

@Injectable({
  providedIn: 'root'
})
export class AuthenticationService {

  private roleRouteMapping: { [key: string]: string } = {
    'coordinator': 'Dashboard',
    'student': 'Student/Information',
    // add other roles and their corresponding routes here
  };

  constructor(private router: Router) { }

  login(user: { loginId: number, role: string,name: string }): void {
    console.log(user);
    localStorage.setItem('currentUser', JSON.stringify(user));
    const route = this.roleRouteMapping[user.role] ;
    this.router.navigate(['/' + route]);
```

Figure 5.1 A snippet from authentication service.

The SUPER routes are handled using the RouterModule of the angular router. The routes are protected by guard in Figure 5.2.

```typescript
import { RouterModule, Routes } from '@angular/router';
import { DashboardComponent } from './Coordinator/dashboard/dashboard.component';
import { CalendarComponent } from './Coordinator/calendar/calendar.component';
import { ComprehensiveExamComponent } from './Coordinator/comprehensive-exam/comprehensive-exam.component';
import { PublicationsComponent } from './Coordinator/publications/publications.component';
import { DatabasePanelComponent } from './Coordinator/database-panel/database-panel.component';
import { LoginComponent } from './Components/login/login.component';
import { GuardService } from './Services/guard.service';
import { StudentCalendarComponent } from './Student/student-calendar/student-calendar.component';
import { StudentInformationComponent } from './Student/student-information/student-information.component';
import { StudentPublicationsComponent } from './Student/student-publications/student-publications.component';
import { StudentRulesFAQsComponent } from './Student/student-rules-faqs/student-rules-faqs.component';
import { SignUpComponent } from './Components/sign-up/sign-up.component';
import { PublicSeminarsComponent as CoordinatorPublicSeminar } from './Coordinator/public-seminars/public-seminars.component';
import { StudentPublicSeminarsComponent } from './Student/student-public-seminars/student-public-seminars.component';
import { StudentSeminarsComponent as StudentSeminar } from './Student/student-seminars/student-seminars.component';
import { ResetPasswordComponent } from './Components/reset-password/reset-password.component';
import { RequestComponent } from './Coordinator/request/request.component';
import { RulesFaqsComponent } from './Coordinator/rules-faqs/rules-faqs.component';

import { WaitingComponent } from './Components/waiting/waiting.component';

const routes: Routes = [
  { path: 'login', component: LoginComponent },
  { path: 'waiting', component: WaitingComponent } ,
  {path: 'Coordinator/Rules&FAQs',component:RulesFaqsComponent , canActivate: [GuardService], data: { role: 'coordinator' }},
```

Figure 5.2: Angular Router Configuration

5.2.1 Layout

The SUPER layout consists of a header and sidebar.

5.2.1.1 Header

The header has SUPER logo on the left side, a search bar, a notification icon, a settings icon, and a user profile with a name display in Figures 5.3 and 5.4.

```
<nav class="navbar navbar-expand navbar-light justify-content-between py-4" id="header" style="background-color: white">
  <h1 class="px-5">
    {{header}}
  </h1>
  <div>
    <ul class="navbar-nav d-flex flex-row me-1 mx-3">
      <li class="nav-item me-3 me-lg-0">
        <form class="form-inline d-flex mx-5 align-self-center">
          <input class="form-control mr-1" type="search" placeholder="Search" aria-label="Search">

          <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>
        </form>
      </li>

      <li class="nav-item me-3 me-lg-0 dropdown">
        <a class="nav-link circle-icon" (click)="toggleNotification()">
          <i class="bi  h5"></i>
          <span *ngIf="unlockedCount > 0" class="badge badge-primary badge-position">{{ unlockedCount }}</span>
        </a>
        <div class="dropdown-content" [ngClass]="{'show': showNotification}">
          <div *ngFor="let message of messages" class="message-box"
            [ngClass]="{'unlocked': message.locked, 'locked': !message.locked}">
            {{ message.text }}
          </div>

        </div>
      </li>

      <li class="nav-item me-3 me-lg-0">
        <a class="nav-link circle-icon" href="#"><i class="bi  h5 "></i></a>

      </li>
      <li class="nav-item me-3 me-lg-0">
```

Figure 5.3: HTML template of nav header utilizing bootstrap classes.

```
.circle-icon{
    width:40px;
    height: 40px;
    display: inline-flexbox;
    align-items: center;
    justify-content: center;
    text-align: center;
    cursor:pointer;
    border: none;
    text-decoration: none;
    border-radius: 50%;
    background-color: #E5F4F1;
    color: #707070 ;
    margin-left: 7px;
    margin-right: 7px;
  }
  .oval-icon{
    width:200px;
    height: 40px;
    align-items: center;
    justify-content: center;
    cursor:pointer;
    border: none;
    text-decoration: none;
    border-radius: 6%;
    background-color: #E5F4F1;
    color: #707070 ;
    margin-right: 10px;
    margin-left: 7px;
  }
  a.oval-icon span{
    bottom: 20%;
    padding-bottom: 30%;
```

Figure 5.4: Custom CSS is also implemented to make UI more appealing and smoother.

5.2.1.2 Sidebar

The sidebar is designed to easily navigate the app. The menu on the left contains all necessary paths as shown in Figures 5.5 and 5.6.

```
<div class="sidebar" id="side">
  <div class="header-box px-4 pt-3 pb-4">
   <a href=""  ><img src="../../../assets/img/logosuper.png" class="w-75 p" ></a>

  </div>
  <ul class="list-unstyled px-2">

    <li *ngFor="let link of navLinks">
      <a
      [routerLink]="link.path" routerLinkActive="myactive" [routerLinkActiveOptions]="{exact: true}"
      class="text-decoration-none px-3 py-3 d-block text-black"
      >{{ link.text }}
    </a>
    </li>
  </ul>
</div>
```

Figure 5.5: A student sidebar.

```
<div class="sidebar" id="side">
 <div class="header-box px-4 pt-3 pb-4">
  <a href=""  ><img src="../../../assets/img/logosuper.png" class="w-75 p" ></a>

 </div>
 <ul class="list-unstyled px-2">
  <li>
    <a routerLink="/Dashboard" routerLinkActive="myactive" [routerLinkActiveOptions]="{exact:true}" class="text-
  </li>
  <li>
    <a routerLink="/Calendar" routerLinkActive="myactive" class="text-decoration-none px-3 py-3 d-block text-bla
  </li>
  <li>
    <a routerLink="/Comprehensive-exam"   routerLinkActive="myactive" class="text-decoration-none px-3 py-3 d-bl
  </li>
  <li>
    <a routerLink="/Coordinator-Public-Seminars" routerLinkActive="myactive" routerLinkActive="myactive" class="
  </li>

  <li>
    <a routerLink="/Publications" routerLinkActive="myactive" routerLinkActive="myactive" class="text-decoratio
```

Figure 5.6: Coordinator sidebar.

5.2.2 Authentication (login and sign-up)

A pretty login page is designed to contain the FCIT logo and SUPER logo.

Users can log in using a username and password. Also, a sign-up button is displayed in Figure 5.7.

```
<form class="mx-auto" [formGroup]="loginForm" (ngSubmit)="onSubmit(loginForm.value)">
  <div class="mb-3 mt-5">
    <label for="idInput" class="form-label">User ID</label>
    <input type="text" pattern="[0-9]*" class="form-control" id="idInput" aria-describedby="emailHelp"
      formControlName="loginId">

    <!-- Error Messages for User ID -->
    <div *ngIf="loginId?.errors && (loginId?.dirty || loginId?.touched)">
      <small *ngIf="loginId?.errors?.['required']" class="text-danger">User ID is required.</small>
      <small *ngIf="loginId?.errors?.['pattern']" class="text-danger">User ID should be numbers only.</small>
    </div>
  </div>
  <div class="mb-3">
    <label for="passwordInput" class="form-label">Password</label>
    <input type="password" class="form-control" id="passwordInput" formControlName="password">
    <a [routerLink]="['/reset-password']" class="form-text mt-3 text-decoration-none">Forget Password?</a>
  </div>

  <button type="submit" class="btn btn-primary mt-4">Login</button>

  <p>
  <p>Not registered yet?<a href="javascript:void(0)" (click)="navigateToSignUp()">Register</a></p>
</form>
<p>{{response}}</p>
```

Figure 5.7: The login page form can also display errors.

After the form is submitted in the Figure 5.8. Form data that contains username and password is sent in a POST HTTP request to the server-side using an authentication service.

The server responds with a successful response or error response. If it is asuccess, the user is redirected to the dashboard, and necessary information is stored in app storage such as username, role, etc.

```
import { Injectable } from '@angular/core';
import { Router } from '@angular/router';

@Injectable({
  providedIn: 'root'
})
export class AuthenticationService {

  private roleRouteMapping: { [key: string]: string } = {
    'coordinator': 'Dashboard',
    'student': 'Student/Information',
    // add other roles and their corresponding routes here
  };

  constructor(private router: Router) { }

  login(user: { loginId: number, role: string,name: string }): void {
    console.log(user);
    localStorage.setItem('currentUser', JSON.stringify(user));
    const route = this.roleRouteMapping[user.role] ;
    this.router.navigate(['/' + route]);
  }

  logout(): void {
    localStorage.removeItem('currentUser');
    this.router.navigate(['/login']);
  }
```

Figure 5.8: A snippet from the authentication service.

The sign-up component uses the same technique as in login. A form with validations is designed that takes several inputs and sends on sever sign-up API.

Validations on the sign-up form are as in Figure 5.9.

```
this.signupForm = this.formBuilder.group({
  loginId: ['', [Validators.required, Validators.pattern('^[0-9]{7}$')]],
  nationalId: ['', [Validators.required, Validators.pattern('^[0-9]{10}$')]],
  yearEnroll: ['', Validators.required],
  department: ['', Validators.required],
  firstName: ['', [Validators.required, Validators.pattern(/^[a-zA-Z]+$/)]],
  middleName: ['', [Validators.required, Validators.pattern(/^[a-zA-Z]+$/)]],
  lastName: ['', [Validators.required, Validators.pattern(/^[a-zA-Z]+$/)]],
  email: ['', [Validators.required, Validators.email]],
  section: ['', Validators.required],
  phoneNumber: ['', [Validators.required, Validators.pattern('^[0-9]{10}$')]],
  gpa: ['', [Validators.required, Validators.pattern(/^\d\.\d{1,2}$/)]], // Pattern to ensure the format "X.YY" or "X.Y"
},
```

Figure 5.9: Angular Reactive Form with Validation.

5.2.3 Student Information

On the Student Information Page as in Figure 5.10, information is displayed that is fetched from the server. Information can be edited as well. It also has a student progress that fetches progress descriptions, progress trackers, and student supervisors.

```
22  export class StudentInformationComponent implements AfterViewInit, OnInit {
58    }
59
60    fetchStudentData() {···
71    }
72
73    enableEdit() {···
76    }
77
78    chosenYearHandler(normalizedYear: Date, datepicker: MatDatepicker<Date>) {···
83    }
84
85    cancelEditMode() {···
88    }
89
90    saveEdits() {···
120   }
121
122   fetchProgressDescription() {···
136   }
137
138   fetchProgressTracker() {···
150   }
151
152   loadStudentSupervisors(): void {···
167   }
168 }
```

Figure 5.10: The following functions are used to perform actions.

5.2.4 FAQs & Rules

The coordinator can fetch the list of FAQs & Rules. New FAQs and rules can also be added. Similarly, Students can only fetch and view those FAQs as in Figures 5.11 and 5.12.

Figure 5.11: Functions in coordinator rules and FAQs component to utilize APIs for various purposes

```
<mat-tab-group>
  <mat-tab>
    <ng-template mat-tab-label>
      Rules
    </ng-template>
    <iframe [src]="pdfSrc"> style="width: 100%; height: 100vh;"></iframe>
  </mat-tab>

  <mat-tab>
    <ng-template mat-tab-label>
      Frequently Asked Questions
    </ng-template>
    <mat-accordion>
      <ng-container *ngFor="let element of response">
        <mat-expansion-panel>
          <mat-expansion-panel-header>
            <mat-panel-title>
              {{ element.question }}
            </mat-panel-title>

          </mat-expansion-panel-header>
          <p> {{ element.answer }}</p>
        </mat-expansion-panel>
      </ng-container>
    </mat-accordion>
  </mat-tab>
</mat-tab-group>
```

Figure 5.12: A snippet from Student Rules and FAQs HTML template to display rules and FAQs.

5.2.5 Calendar

Student and Coordinator Calendar components are used to manage events.

Events can fetched and displayed using the full calendar Module as in Figures 5.13 and 5.14.

```
fetchData() {
  this.calendarOptions.update(options => {
    return {
      ...options,
      eventSources: [{
        url: 'http://127.0.0.1:8000/api/event',
        method: 'GET',
        failure: function () {
          alert('There was an error while fetching events!');
        }
      }]
    };
  });

}
```

Figure 5.13: Calendar data is fetched from event API in function.

```
108  dialogRef.afterClosed().subscribe(result => {
109    this.calendarOptions.update(options => {
110      return {
111        ...options,
112        eventSources: [{
113          url: 'http://127.0.0.1:8000/api/event',
114          method: 'GET',
115          failure: function () {
116            alert('There was an error while fetching events!');
117          }
118        }]
119      };
120    });
121    console.log('Dialog was closed', result);
122  }); }
```

Figure 5.14: After creating an event through dialog, it is saved as this.

The calendar view is rendered through the popular Angular full Calendar Module that has plenty of options and customization as in Figure 5.15.

```
<div class="Container">
  <full-calendar [options]='calendarOptions()' >
  </full-calendar>
</div>
```

Figure 5.15: Angular Full Calendar Integration.

5.2.6 Comprehensive Exam

The ComprehensiveExam component in Figure 5.16 enables the coordinator to display exams fetched from the server, edit, or delete them, add new exams, assign students to exams, notify students about exams, and record student grades.

Figure 5.16: Snapshot of the Comprehensive Exam on the Coordinator Side.

5.2.7 Publication

In the Publication component as in Figure 5.17 and Figure 5.18, students can add and edit publications, while the coordinator can view and filter student publications.

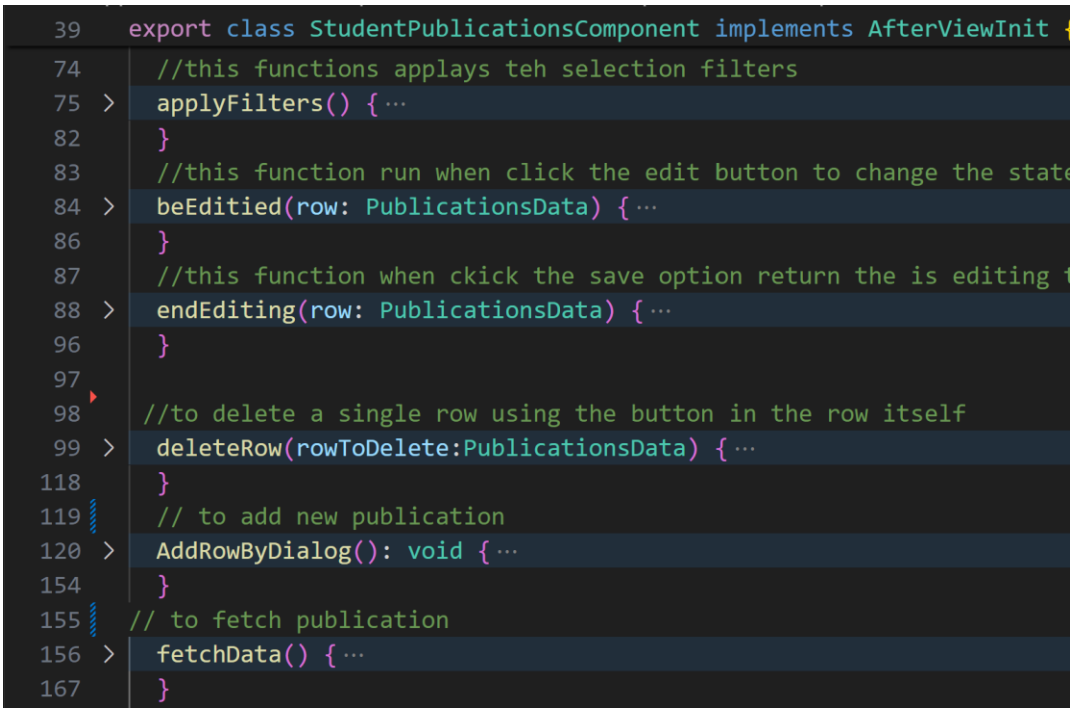

Figure 5.17: Snapshot of the Student Publication Component on the Student Side.

```
<div class="container">
  <mat-form-field >
  <mat-label>Filter</mat-label>
  <input matInput (keyup)="applySearchFilter($event)" placeholder="Ex. ium" #input>
</mat-form-field>

<mat-form-field>
  <mat-label>Date</mat-label>
  <mat-select (selectionChange)="selectedDate = $event.value; applyFilters()">
    <mat-option >All Dates</mat-option>
    <mat-option *ngFor="let element of DatesArray" [value]="element">{{element}}</mat-option>
  </mat-select>
</mat-form-field>

<mat-form-field>
  <mat-label>Field</mat-label>
  <mat-select (selectionChange)="selectedField = $event.value; applyFilters()">
    <mat-option >All Fields</mat-option>
    <mat-option *ngFor="let element of FieldArry" [value]="element">{{element}}</mat-option>
  </mat-select>
</mat-form-field>
</div>
```

Figure 5.18: Snapshot of The HTML of the Publication Component on the Coordinator Side.

5.2.8 Seminar

In the Public Seminars Component as in Figures 5.19 and 5.20, the coordinator can add, edit, and delete seminars, while students can only view upcoming seminars. Additionally, the Seminar Attendances Component allows students to add certificates for the seminars they have attended.

```
34   export class PublicSeminarsComponent implements AfterViewInit{
132
133  //to delete by using the checkbox
134    deleteSelectedRows() {···
155      }}
156    //to add  the seminars using Dialog
157    AddRowByDialog(): void {···
191    }
192
193  //to fetch the seminars
194  fetchData() {···
206  }
```

Figure 5.19: Snapshot of the Public Seminar on the Coordinator Side.

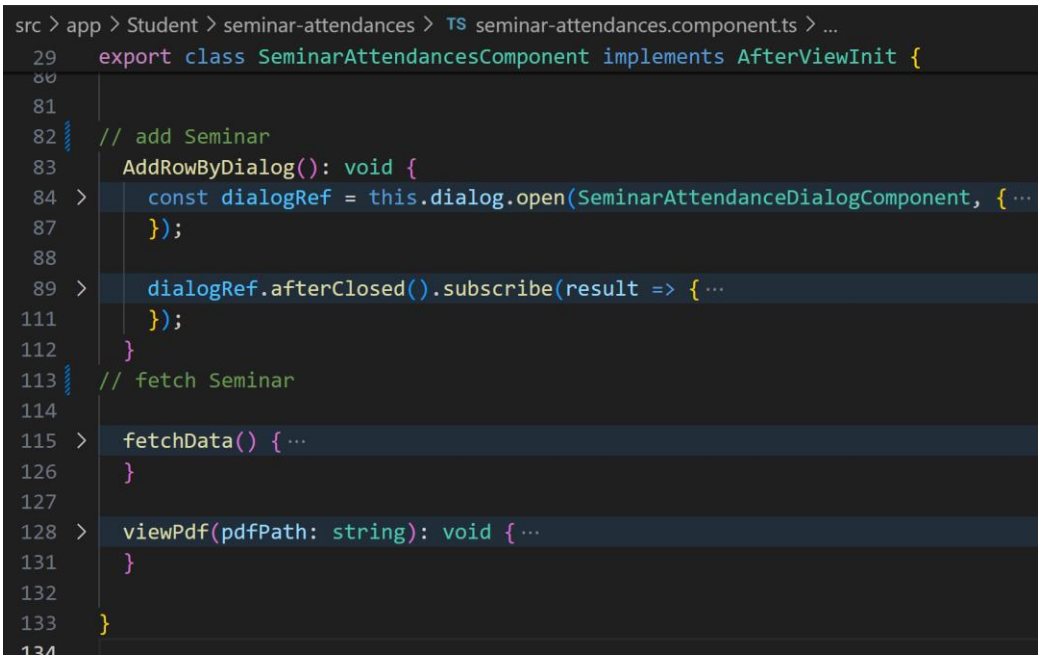

Figure 5.20: Snapshot of the Seminar Attendance on the Student Side.

5.3 Server-Side Implementation

The server-side framework is Laravel. It is a powerful and elegant PHP framework designed for web application development. It follows the Model-View-Controller (MVC) architectural pattern, which helps in organizing and structuring the code efficiently in Figure 5.21.

Key Components Used in Our Website:

5.3.1 Models:

1. **Purpose:** Represent the data structure and business logic of the website.
2. **Usage in Our website:** We use models to interact with the MySQL database, managing data related to PhD students and coordinators, such as their profiles, progress, thesis submissions, and publications.

```php
class User extends Model implements Authenticatable
{
    use HasFactory, AuthenticableTrait;

    protected $table = 'user';
    //protected $primaryKey= 'LoginId';

    protected $fillable = [
        'loginId', 'password', 'firstName', 'middleName', 'lastName', 'phone_number', 'email','gender', 'department', 'role'
    ];

    protected $hidden = [
        'password',
    ];

    public function students()
    {
        return $this->hasMany(Student::class, 'userId'); // Assuming userId is the foreign key
    }

    public function coordinators()
    {
        return $this->hasMany(Coordinator::class);
    }
    public function takenExams()
{
    return $this->hasMany(Taken_Exam::class, 'loginId', 'loginId');
}
public function student()
{
    return $this->hasOne(Student::class, 'userId');
}
public function supervisors()
{
    return $this->belongsToMany(Supervisor::class, 'supervise', 'userId', 'supervisiorId')
                ->withPivot('type');  // Including type data
}
```

Figure 5.21: Example of a User Model.

The models also define relationships. In our Laravel framework, we use Eloquent ORM to define and manage the relationships between different models. These relationships represent the logical connections between various entities on our website, such as students, coordinators, theses, publications, and academic events. Understanding and defining these relationships helps in querying related data efficiently and maintaining data integrity as in Figure 5.22.

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use App\Models\User;
use App\Models\Supervisor;
use App\Models\Supervise;
class Publications extends Model
{
    use HasFactory;

protected $table = 'publication';
protected $primaryKey = 'publicationId';
// Attributes that are mass assignable
protected $fillable = [
    'title', 'field','publicationType','venueName', 'doi','date', 'pdfPath', 'userId'

];
public $timestamps = false;

// Attributes to append to the model's array and JSON form
protected $appends = ['studentName', 'loginId', 'supervisorName'];

/**
 * Define the relationship to the User model.
 */
public function user()
{
    return $this->belongsTo(User::class, 'userId');
}

/**
 * Define the relationship to the Supervisor model through the Supervise pivot table.
 * Assuming there's only one supervisor for each publication and their type is 'main'.
 */
public function supervisors()
```

Figure 5.22: A snippet from Publications Mode that also represents a relationship with the user.

5.3.2 API Routes:

1. **Purpose:** Define the endpoints for the application's API.
2. **Usage in Our Website:** Our website defines routes to handle HTTP requests for various functionalities, such as login, fetching student data, and managing academic calendars.

We have defined several API routes. These routes are consumed by the client side for various functionalities as in Figure 5.23.

```php
<?php

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\ComprehensiveController;
use App\Http\Controllers\SeminarController;
use App\Http\Controllers\PublicationController;
use App\Http\Controllers\StudentController;
use App\Http\Controllers\loginController;
use App\Http\Controllers\EventController;
use App\Http\Controllers\RuleController;
use App\Http\Controllers\signUp;
use App\Http\Controllers\emailcontroller;
use App\Http\Controllers\ProgressDescriptionController;

Route::get('/ComprehensiveExam', [ComprehensiveController::class, 'fetchData']);
Route::post('/ComprehensiveExam/edit/{id}', [ComprehensiveController::class, 'update']);
Route::post('/ComprehensiveExam/delete/{id}', [ComprehensiveController::class, 'delete']);
Route::post('/ComprehensiveExam/add/', [ComprehensiveController::class, 'add']);
Route::get('/ComprehensiveExam/students', [ComprehensiveController::class, 'getStudentsByYearAndSeason']);
Route::post('/ComprehensiveExam/assign-grades', [ComprehensiveController::class, 'assignGrades']);
Route::post('/ComprehensiveExam/assign-grades-and-notify', [ComprehensiveController::class, 'assignGradesAndNotify']);
Route::post('/ComprehensiveExam/new', [ComprehensiveController::class, 'addNewExam']);
Route::get('ComprehensiveExam/exams', [ComprehensiveController::class, 'index']);
Route::post('/ComprehensiveExam/assign-students-to-exam', [ComprehensiveController::class, 'assignStudentsToExam']);
Route::post('/ComprehensiveExam/notify-exam', [ComprehensiveController::class, 'notifyExam']);
Route::post('/ComprehensiveExam/get-students-with-grades', [ComprehensiveController::class, 'getStudentsWithGrades']);
Route::get('/exams/{examId}/students', [ComprehensiveController::class, 'getStudentsByExamId']);

Route::get('/StudentSeminars', [SeminarController::class, 'fetchStudentsData']);
Route::get('/PublicSeminars', [SeminarController::class, 'fetchData']);
Route::get('/UserSeminars/{id}', [SeminarController::class, 'fetchUserData']);
Route::post('/Seminars/edit/{id}', [SeminarController::class, 'update']);
Route::post('/Seminars/delete/{id}', [SeminarController::class, 'delete']);
Route::post('/Seminars/add/', [SeminarController::class, 'add']);


Route::get('/publications', [PublicationController::class, 'fetchData']);
```


Figure 5.23: Laravel API Route Definitions.

5.3.3 Controllers:

1. **Purpose:** Handle the incoming requests, process them, and return appropriate responses.
2. **Usage in Our Website:** Controllers manage the logic for user authentication, handling CRUD operations for student and coordinator data, posting announcements, and managing events like seminars.

A login Controller as in Figure 5.24 that handles login requests from client-side and provides necessary responses .

```php
class LoginController extends Controller
{
    public function __construct()
    {
        $this->middleware('web');
    }
    public function login(Request $request)
    {
        $userId = $request->input('loginId');
        $password = $request->input('password');
        $request->session()->start();

        // Retrieve user from database by username
        $user = User::where('loginId', $userId)->first();

        if (!$user) {
            return response()->json('User not found', 404);
        }

        // Check if password matches
        if ($user->password === $password) {
            // Manually create a session for the user
            $request->session()->put('temp_user_id', $user->id);

            // Retrieve the authenticated user
            $authenticatedUser = User::find($request->session()->get('temp_user_id'));

            // Return the authenticated user as JSON response
            return response()->json($authenticatedUser);
        } else {
            return response()->json('Incorrect password', 401);
        }
    }

}
```

Figure 5.24: Login Controller.

Another Controller example is Event Controller as in Figure 5.25 which handles various functions as fetching all events, adding, updating, deleting, etc.

```php
public function index()
{
    // Fetch all events from the Event model
    $events = event_model::all();  // Fixed variable name and removed unnecessary foreach loop
    $formattedEvents = [];

    foreach ($events as $event) {
        $formattedEvents[] = [
            'title' => $event->title,
            'date' => $event->eventStart,
        ];
    }

    // Return the events as a response (you can customize this)
    return response()->json($formattedEvents);
}
public function fetchData()
{ ···
}

public function add(Request $request)
{
    $event = new event_model([
        'title' => $request['title'],
        'eventStart' => $request['EventStart'],
        'eventEnd' => $request['EventEnd'],
        'description' => $request['Description'],
    ]);
    $event->save();
    return response()->json(['message' => 'Event instance saved successfully']);
}
```

Figure 5.25: Event Controller

5.3.4 Commands:

1. **Purpose:** Handle automated repetitive tasks.
2. **Usage in Our website:** The "Notify Student for Seminar" command notifies students to attend a seminar either two semesters after the start of their dissertation year or one year after their previous seminar as in Figure 5.26.

```php
            })
            ->get();

        if ($enrollments->count() == 3 && $enrollments->last()->semester->isCurrent()) {
            $data = [
                'subject' => 'Your Seminar Is Upcoming!',
                'firstName' => $student->firstName,
                'lastName' => $student->lastName,
                'fullName' => $student->firstName . ' ' . $student->lastName,
                'email' => $student->email
            ];

            Mail::to($student->email)->send(new NotifySeminar($data));
        }
        $this->info("Notified student: {$student->name} (Supervisor: {$student->supervisor})");
    }
```

Figure 5.26: Snapshot of the Commands to Notify Students for Seminars

5.4 Database Implementation

MySQL is a widely used relational database management system (RDBMS) known for its reliability and performance. In Laravel, MySQL is commonly used to store and manage application data. Laravel provides built-in support for MySQL, making it easy to interact with the database using Eloquent ORM and query builder.

Key Features and Usage in Laravel:

Database Configuration:

1. **Description:** Laravel allows you to configure database connections easily.
2. **Usage:** Database settings such as host, database name, username, and password are defined in the .env file.

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=lara
DB_USERNAME=root
DB_PASSWORD=

BROADCAST_DRIVER=log
CACHE_DRIVER=file
FILESYSTEM_DISK=local
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120

MEMCACHED_HOST=127.0.0.1

REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_MAILER=smtp
MAIL_HOST=mailpit
MAIL_PORT=1025
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
MAIL_FROM_ADDRESS="hello@example.com"
MAIL_FROM_NAME="${APP_NAME}"
```

Figure 5.27: Database Configuration.

Eloquent ORM:

1. Description: Eloquent provides an expressive syntax to interact with the database using models.
2. Usage: Define models that map to database tables, allowing CRUD operations and complex queries through simple methods.

```php
public function fetchStudentData($loginId) {
    // Filter publications where the associated user's loginId matches the provided ID
    $publications = Publications::with(['user', 'supervisors'])
                                ->whereHas('user', function($query) use ($loginId) {
                                    $query->where('loginId', $loginId);
                                })
                                ->get();
```

Figure 5.28: ORM in action to find publications in the database for logged-in users with supervisor data.

Query Builder:

1. **Description:** A fluent interface for building and executing database queries.
2. **Usage:** Perform complex SQL queries using a clean and straightforward syntax.

Migrations:

1. **Description:** Version control for the database schema.
2. **Usage:** Create and manage database tables and columns programmatically.

Migrations as in Figure 5.29 ensure the database structure is consistent across different environments.

```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('seminar', function (Blueprint $table) {
            $table->id('seminarId');
            $table->unsignedBigInteger('userId')->nullable(); // Nullable
            $table->string('Name');
            $table->string('type');
            $table->string('title')->nullable();
            $table->string('field')->nullable();
            $table->date('date');
            $table->time('time');
            $table->string('location');

            // Define foreign key constraint for userId, conditional based on type being 'student'
            $table->foreign('userId')
                ->references('id')
                ->on('user')
                ->onDelete('cascade')
                ->when('type', 'student'); // Conditional foreign key constraint
        });
        DB::statement('ALTER TABLE seminar ADD CONSTRAINT check_user_type CHECK ((type = \'student\' AND userId IS NOT NULL) OR (type !=

    }
```

Figure 5.29: A migration that creates a seminar table with structure and field types.

Seeding:

1. Description: Populate the database with test data.
2. Usage: Create seed files to insert sample data into the database for testing and development.

```php
<?php

namespace Database\Seeders;
use Illuminate\Support\Facades\DB;
use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;

class SupervisorTableSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        DB::table('supervisior')->insert([
            [
                'supervisiorId' => 1,
                'firstName' => 'waffa',
                'middletName' => 'A.',
                'lastName' => 'super',
                'phone_number' => '1234567890',
                'email' => 'john.doe@example.com',
                'department' => '',
            ],]);
    }
}

```

Figure 5.30: The supervisor table is seeded with a row defined SupervisorTableSeeder as in the Figure.

Relationships:

1. **Description:** Define relationships between different tables.
2. **Usage:** Establish one-to-one, one-to-many, and many-to-many relationships using Eloquent, enabling efficient querying of related data.

Artisan Commands:

1. **Description:** Command-line interface for managing the database.
2. **Usage:** Run migrations, seeders, and other database-related commands using php artisan commands.

Using MySQL with Laravel provides a powerful and flexible foundation for building robust web applications, leveraging the strengths of both the RDBMS and the framework to handle data efficiently.

These components work together to create a robust and maintainable website for PhD students and coordinators, facilitating features such as tracking progress, managing academic schedules, and handling thesis submissions.

5.5 Code Debugging and Troubleshooting

During the development of our Laravel and Angular websites for SUPER, we encountered several challenges. Here are some key issues we faced and the solutions we implemented:

5.5.1. Angular Design Issues

Challenge: While designing the front end using Angular, we faced issues with responsive design and component communication.

Problem: Some UI elements were not rendering correctly on different screen sizes, and passing data between nested components was causing errors.

Solution:

- **Responsive Design:** We used CSS Flexbox and Grid layout techniques along with Angular's built-in responsive utilities to ensure the UI adapted to various screen sizes.
- **Component Communication:** We utilized Angular's @Input and @Output decorators for parent-child communication and Angular Services with RxJS subjects for more complex data sharing across multiple components.

5.5.2. CORS (Cross-Origin Resource Sharing) Issues

Challenge: When the Angular front-end tried to make HTTP requests to the Laravel back-end, we encountered CORS errors.

Problem: The browser blocked requests from the front-end to the back-end server due to different origins.

Solution:

- **Laravel Middleware:** We installed the Laravel CORS package (barryvdh/laravel-cors) and configured it in config/cors.php to allow the necessary HTTP methods and origins.
- **Environment Configuration:** Ensured the correct environment settings in. env for APP_URL and CORS headers.

5.5.3. Migration Failures

Challenge: Database migrations were failing, preventing the creation of necessary tables and relationships.

Problem: Migrations failed due to syntax errors, missing columns, or incorrect foreign key constraints.

Solution:

- **Debugging:** Checked the migration files for syntax errors and ensured proper ordering of migrations to account for dependencies.
- **Rollback and Redo:** Used PHP artisan migrate:rollback to undo the last batch of migrations, fixed the issues, and reran migrations with PHP artisan migrate.
- **Seeding and Testing:** Created seeders to populate the database with test data and ensure migrations worked correctly.

5.5.4. Controller Issues

Challenge: Some controller methods were not functioning as expected, leading to errors in data processing and response handling.

Problem: Errors in the controller logic, incorrect data validation, and missing or improper route definitions.

Solution:

- **Data Validation:** Used Laravel's built-in validation features in controller methods to ensure data integrity.
- **Route Inspection:** Verified route definitions in routes/api.php and ensured they correctly pointed to the appropriate controller methods.
- **Unit Testing:** Wrote unit tests to cover controller methods, ensuring they handled requests and responses correctly.

5.5.5. Authentication Issues

Challenge: Implementing authentication and ensuring secure login/logout functionality.

Problem: Issues with token generation and management, as well as protecting routes.

Solution:

Route Protection: Used middleware to protect routes that require authentication, ensuring only authorized users could access certain endpoints.

By addressing these challenges systematically, we were able to ensure our website was robust, secure, and user-friendly, providing a seamless experience for PhD students and coordinators.

5.6 Conclusion

In conclusion, this chapter provides code snippets with descriptions and logical reasoning. It addresses debugging and troubleshooting issues with solutions derived from version management. The chapter emphasizes the significance of packaging and documentation for future development and collaboration. Overall, it serves as a valuable resource for understanding and maintaining the code.

Chapter VI: Testing

6.1 Introduction

The testing chapter of the project describes the various types of testing that were performed after the implementation phase to ensure the reliability, functionality, and performance of the code.

6.2 Testing

The testing process involved both unit testing and integration testing to verify the individual components and their interactions within the system. Additionally, compatibility and performance were conducted to assess the system's behavior under different conditions. Usability testing was also performed to evaluate the user experience and ease of use. Finally, system testing was executed to evaluate the overall functionality and user experience of the website.

6.2.1 Unit testing

The unit testing performed for the Super Project focuses on validating various functionalities and ensuring that the endpoints behave as expected. The Laravel testing framework, along with PHPUnit and Laravel's built-in testing tools, was used to facilitate the testing process. The test cases are organized to cover different scenarios, as described below:

1- **Update Seminar Test:**
   - **Test Case:** testUpdateSeminar
   - **Objective:** Verify that the seminar update functionality works correctly.
   - **Details:** A seminar is created using a factory, and then an update request is sent with new data. The test asserts that the response status is 200, indicating success.
   - **Code:**

```php
    public function testUpdateSeminar()
    {
        $seminar = Seminars::factory()->create();        // Create a seminar
        $newData = [        // New data for updating the seminar

            'Title' => 'Updated Title',
            'Field' => 'Updated Field',
            'Location' => 'Updated Location',
            'Date' => '2024-05-15', // New date
            'Time' => '10:00:00', // New time
            'Name' => 'Updated Name'
        ];

        // Call the update seminar endpoint with the new data
        $response = $this->put("/seminar/update/{$seminar->id}", $newData);
        $response->assertStatus(200);        // Assert that the response is successful
    }

}
```

Figure 6.1: Code for the Test Update Seminar.

2- **Fetch Rules Test:**
   - **Test Case:** testFetchRules
   - **Objective:** Ensure that fetching rules returns the correct data.
   - **Details:** A rule is created using a factory, and then a GET request is made to fetch the rules. The test asserts that the response status is 200 and that the response contains the expected rule data.
   - **Code:**

Figure 6.2: Code for the Test Fetch Rules.

3- **Fetch Rules When Empty Test:**
   - **Test Case:** testFetchRulesWhenEmpty
   - **Objective:** Verify the behavior when no rules are present in the database.
   - **Details:** All rules are truncated from the database, and a GET request is made to fetch the rules. The test asserts that the response status is 200.
   - **Code:**

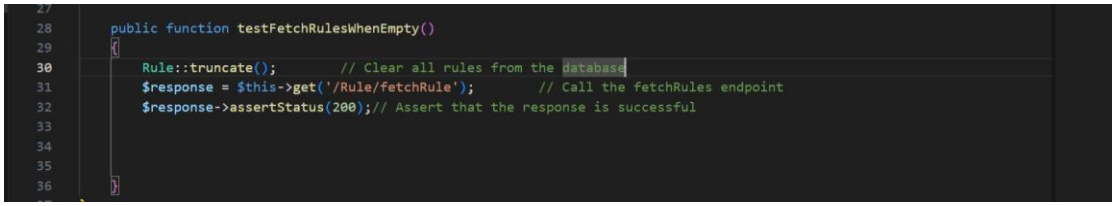

Figure 6.3: Code for the Test Fetch Rules When Empty.

4- **Forgot Password Test:**
   - **Test Case:** testForgotPassword
   - **Objective:** Validate the forgot password functionality.
   - **Details:** The Mail facade is mocked to prevent actual email sending. The test checks that the response body contains the expected data and asserts that the email was sent to the correct recipient.

- **Code:**

```php

    public function testForgotPassword()
    {

        // Mock the Mail facade to prevent actual email sending
        Mail::fake();

        // Assert the response body contains the expected data
        $response->assertJson([
            'subject' => 'New Password For SUPER!',
            'firstName' => 'Bashaier',
            'lastName' => 'Alamoudi',
            'email' => 'Bashaier@gmail.com'
        ]);

        // Assert that the email was sent to the correct recipient with the correct data
        Mail::assertSent(forgotPassword::class, function ($mail) use ($user) {
            return $mail->hasTo($user->email) &&
                   $mail->data['subject'] === 'New Password For SUPER!';
        });
    }
}

```

Figure 6.4: Code for the Test Forgot Password.

5- Add Event Test:

- **Test Case:** testAddEvent
- **Objective:** Confirm that adding an event works as expected.
- **Details:** A POST request is sent to add an event with sample data. The test asserts that the response status is 200.
- **Code:**

```php
class EventTest extends TestCase
{
    use RefreshDatabase; // Use the RefreshDatabase trait to reset the database after each test

    public function testAddEvent()
    {
        $data = [
            'title' => 'Sample Event',
            'EventStart' => '2024-05-15 09:00:00',
            'EventEnd' => '2024-05-15 17:00:00',
            'Description' => 'This is a sample event description.',
        ];

        // Send a POST request to the /event/add endpoint with the sample data
        $response = $this->post('/event/add', $data);

        // Assert that the response status code is 200
        $response->assertStatus(200);
    }
}

```

Figure 6.5: Code for the Test Add Event.

Results for unit tests:

```
PS C:\Users\earth\laravel-1> php artisan test

  PASS  Tests\Feature\EventTest
  ✓ add event                                                    0.01s

  PASS  Tests\Feature\PasswordResetTest
  ✓ forgot password                                              0.01s

  PASS  Tests\Feature\RuleTest
  ✓ fetch rules
  ✓ fetch rules when empty

  PASS  Tests\Feature\SeminarEditTest
  ✓ seminar edit

  Tests:    5 passed (5 assertions)
  Duration: 0.18s
```

Figure 6.6: Result for the Unit Testing.

6.2.2 Integration test

The integration test aims to verify the complete process of creating a new seminar by a coordinator and automatically generating a corresponding event if the seminar type is "public". This test ensures that the created event is correctly displayed on the students' calendar. Additionally, if the seminar type is "student", the seminar will be added to the student's seminar list and displayed to the student who will present it.

Method

The code below was tested during the integration test. It demonstrates the process of creating a new seminar by a coordinator and different test cases its type is student or public:

```php
    }
    public function add(Request $request)
    {
        $validatedData = $request->validate([
            'loginId' => 'nullable',
            'Title' => 'nullable|string|max:255',
            'Field' => 'nullable|string|max:255',
            'Date' => 'required|date',
            'Location' => 'required|string',
            'Time' => 'required',
            'Type' => 'required|string',
            'Name' =>'nullable|string',
        ]);

        // Fetch the user ID based on the provided loginId
        $userId = null;
        if ($validatedData['loginId']) {
            $user = User::where('loginId', $validatedData['loginId'])->first();
            if ($user) {
                $userId = $user->id;
                // If Name is not provided, set it to the user's full name
                $validatedData['Name'] =  ($user->firstName . ' ' . $user->lastName);
            }
        }

        // Create a new seminar object
        $seminar = new Seminars([
            'userId' => $userId,
            'Name' => $validatedData['Name'],
            'title' => $validatedData['Title'],
            'field' => $validatedData['Field'],
            'date' => $validatedData['Date'],
            'time' => $validatedData['Time'],
            'location' => $validatedData['Location'],
            'type' => $validatedData['Type'],
        ]);
        $seminar ->save();

        if ($seminar->type=='public'){
            $title= 'Title of Semiar: '.$validatedData['Title'] ;
            $event = new event_model([
                'title' => $title,
                'eventStart' => $validatedData['Date'],
                'eventEnd' => $validatedData['Date'],
                'description' => "A  Seminar presented by ".$validatedData['Name']." under title ".$validatedData['Title'].
            ]);
            $eventController = new EventController();
            $eventController->addSeminar($event);

        }


    }
```

Figure 6.7: The process of creating a new seminar by a coordinator.

```php
public function addSeminar(event_model $event)
    {
        $event->save();

        return response()->json(['message' => 'Event instance saved successfully']);
    }
```

Figure 6.8: The method in event controller.

Test Case 1: add seminar type public.

- **Input:**

As shown in the figure the input is:

- Type = public
- Name = the presenter's name
- Title = title of the seminar
- Field = field of the presented seminar
- Location = where is the seminar presented
- Time = the time of the seminar
- Date = the date of the seminar

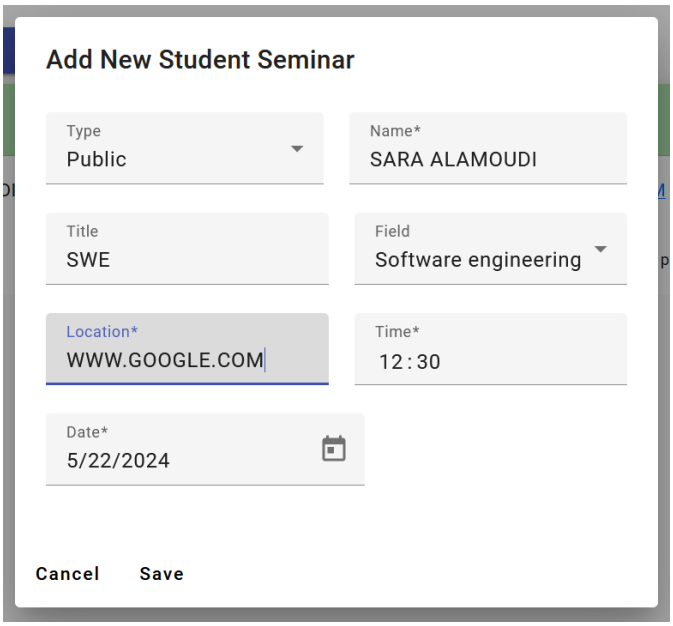

Figure 6.9: the Input of add seminar type public.

- **Expected Output**
  - Seminar Data is saved in the database and shown on both the seminars page and the calendar.
- **Actual Output**

As shown in Figure

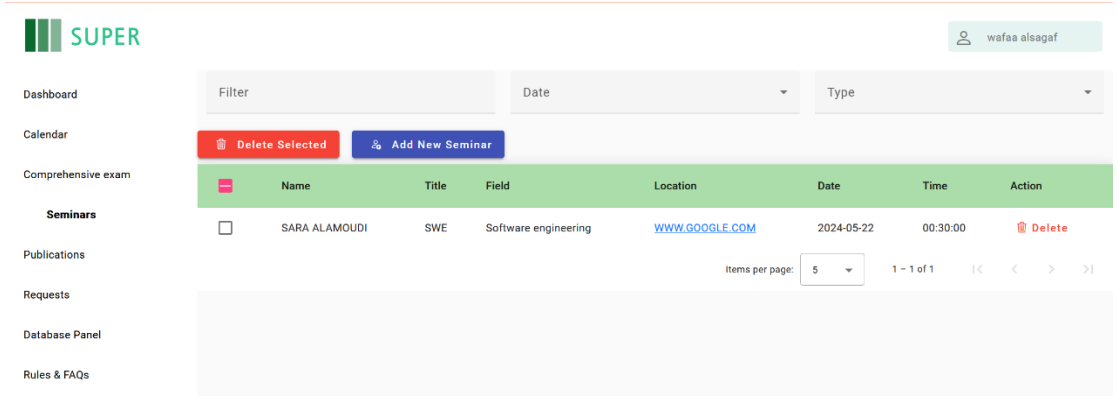

Figure 6.10: The Actual Output of add seminar type public Part 1

Figure 6.11: The Actual Output add seminar type public Part 2

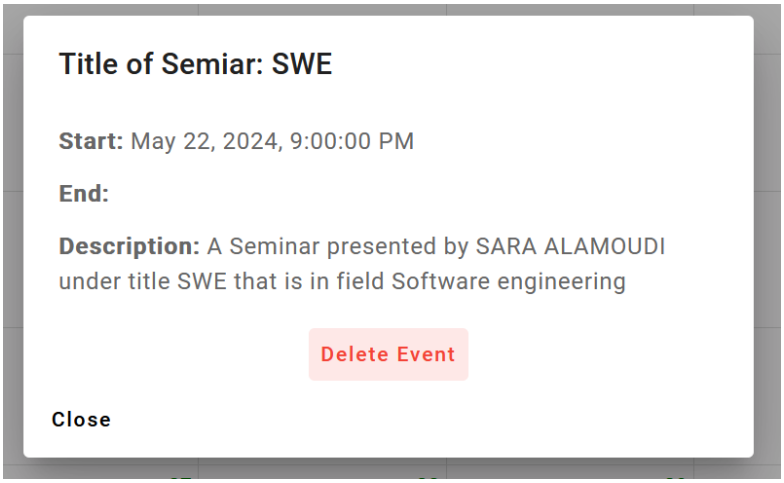

Figure 6.12: The Actual Output of add seminar type public part 3

Test Case 2: Successfully add seminar-type students.

- **Input**

As shown in the figure the input is:

- Type = student
- ID = the student's ID
- Title = title of the seminar
- Field = field of the presented seminar
- Location = where is the seminar presented
- Time = the time of the seminar
- Date = the date of the seminar
-

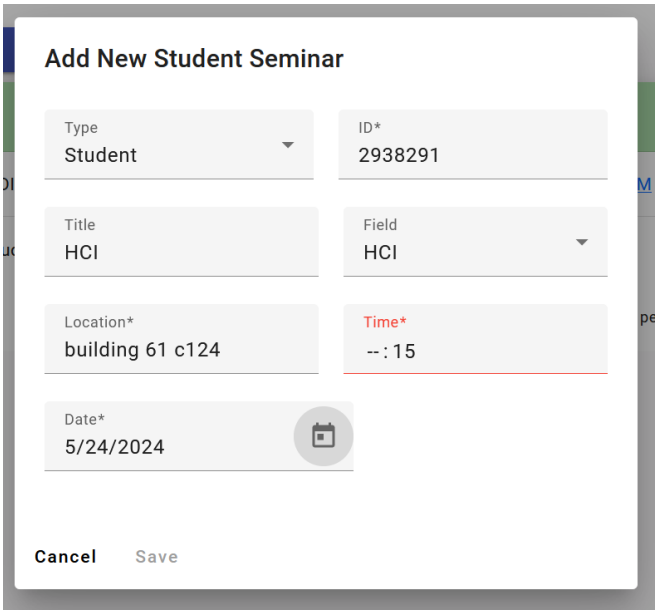

Figure 6.13: The input of Successfully add seminar type student

- **Expected Output**
  - Seminar Data saved in the database for the student and showed on the student page

.

- **Actual Output**

  As shown in Figure

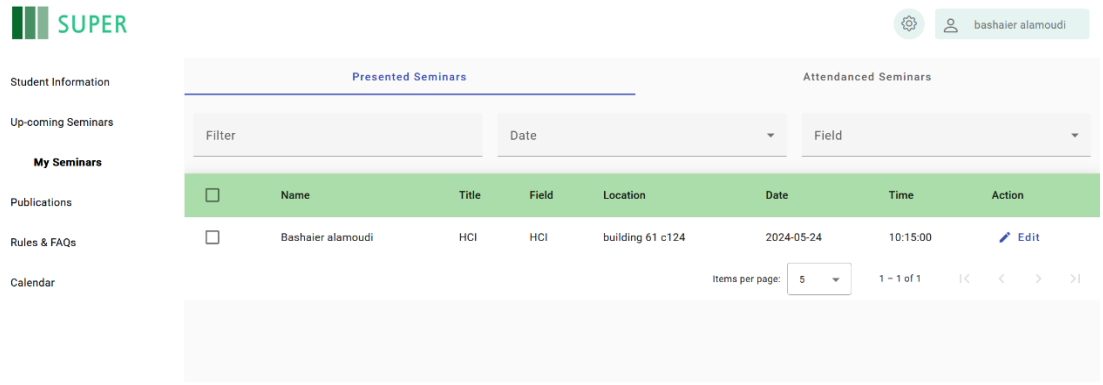

Figure 6.14: Student page

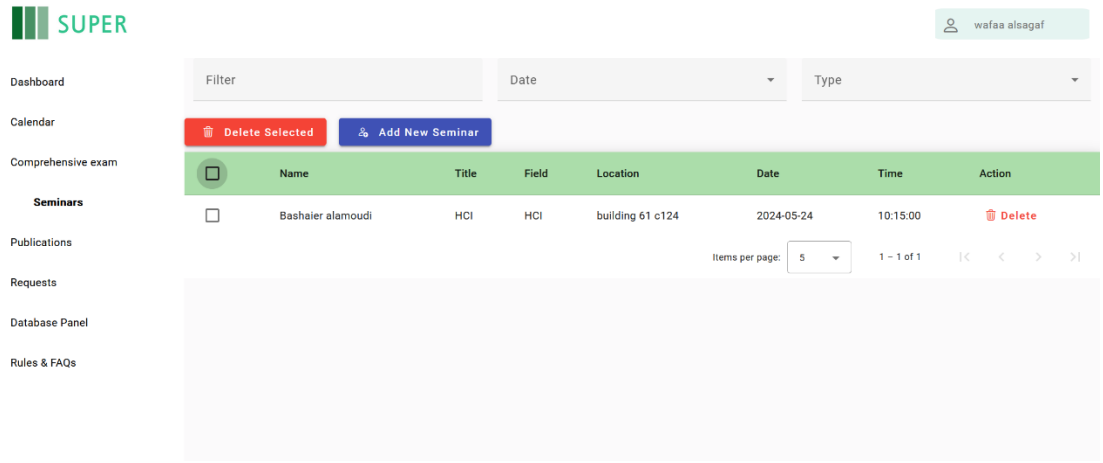

Figure 6.15: Coordinator page

6.2.3 Compatibility Test

Table 6.1 shows the Compatibility test for SUPER.

Table 6.1: Compatibility Test Table

Device	browser	Pass/fail
Windows	chrome 123.0.6312.123 (64-bit)	Pass
Windows	FireFox 125.0.1(64-bit)	Pass
Windows	edge 124.0.2478.80(64-bit)	Pass
Macbook pro	safari 17.4.1	pass

The compatibility test indicates that SUPER functions successfully on both Windows and MacBook Pro devices using the specified versions of these browsers.

6.2.4 System testing

The system test cases, both for the front-end and back-end components, along with their expected and actual outputs, are summarized in Table 6.2. The table demonstrates that all test cases have passed successfully, indicating that SUPER system is functioning as expected.

Table 6.2: System Test Table

Test Case ID	Test Case name	Description	Test Data	Expected output	Actual output	Pass /fail
1	register	New students' signup	Student's all information	"Wait for Approval" page	"Wait for Approval" page	Pass
2	Login	login to the website	Id and password	Login successfully	Login successfully	Pass
3	Add publication	Student adds publication	Publication's data	Publication added successfully	Publication added successfully	Pass
4	Add seminar	Coordinator adds new seminar	Seminar's data	Seminar added successfully	Seminar added successfully	Pass
5	Add event	Coordinator Adds new event	Event's data	Event added successfully	Event added successfully	Pass
6	Add frequently asked questions	Coordinator adds to FAQs page	Question and answer	Added successfully	Added successfully	Pass
7	Add rules file	Coordinators add rule file for students	Rule file	Rule file displayed	Rule file displayed	Pass
8	Add new comprehensive exam	Coordinator adds new exam details	Exam details	New exam created successfully	New exam created successfully	Pass
						Continue

Continuation						
9	Assign student to exam	Assign students id to specific exam	Exam id and students' id	Notify students for the upcoming exam	Notify students for the upcoming exam	Pass
10	Add exam grades	Add students' grades	Exam id Student id grade	Notify students their grades	Notify students their grades	Pass
11	Accept Request	Accept student signUp request	Accept button	Student gets their password by email	Student gets their password by email	Pass
12	Reject Request	Reject student signUp request	Reject button	Notify students to contact coordinator	Notify students to contact coordinator	Pass
13	Edit student profile	Students edit profile	New email	Email changed successfully	Email changed successfully	Pass

6.3 Conclusion

The testing phase of the Super Project encompassed unit testing, integration testing, compatibility testing, and system testing to ensure the system's quality and reliability. These comprehensive testing activities were crucial in validating the functionality and performance of the Super Project. Ongoing testing efforts will be essential for maintaining reliability and delivering a high-quality user experience.

Chapter VII: Results and Discussion

7.1 Introduction

This chapter illustrates the various interfaces of the Scholar's Unified Progress & Efficiency Resource (SUPER) website through descriptive snapshots and detailed explanations. We aim to comprehensively understand the website’s functionality and highlight the objectives achieved during its development. We will also address any limitations encountered during the process.

7.2 Project Navigation Guide

Clear and effective guidance is crucial for users to fully benefit from the project. This section provides a comprehensive guide with step-by-step instructions and visual snapshots, tailored for users of all proficiency levels. The aim is to create a user-friendly experience, ensuring smooth navigation and maximum value from the website.

The activation process for the SUPER system begins with users accessing the website and registering as new students on the "New Student" page. After submitting their registration, students receive an email with their login credentials. They can then use the "Login" page to access the system, with an option to reset their password if forgotten.

The guide covers various interfaces, including the "Student Information" page for viewing and tracking academic progress, and the calendar page, which displays important dates managed by coordinators. The "Comprehensive Exam" and "Seminar" pages allow coordinators to manage exams and seminars, notifying students via email. The "Rules & FAQs" page provides access to essential resources, and the "Publications" page enables students to upload and manage their research outputs. Finally, the "Dashboard" page offers coordinators an overview of student statistics, aiding in quick analysis and data management.

The "New Student" in Figure 7.1 interface allows students to register for the SUPER website. Students must provide their personal information, Once the student fills out the registration form and submits it, their information is sent to the coordinator. The student will receive an email containing their login password if they are accepted. The coordinator will review the student's information and decide whether to accept or reject the student's registration.

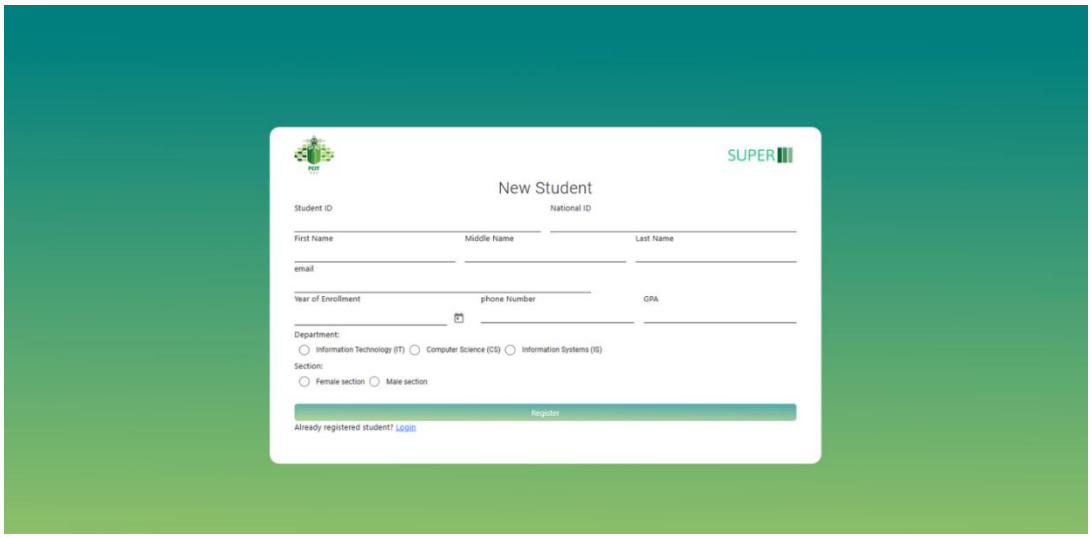

Figure 7.1: New Student Interface.

The "Login" interface in Figure 7.2 enables users to access the SUPER website. Users must enter their User ID and Password to log in. If users forget their password as in Figure 7.3, they can use the "Forget Password?" option to reset it.

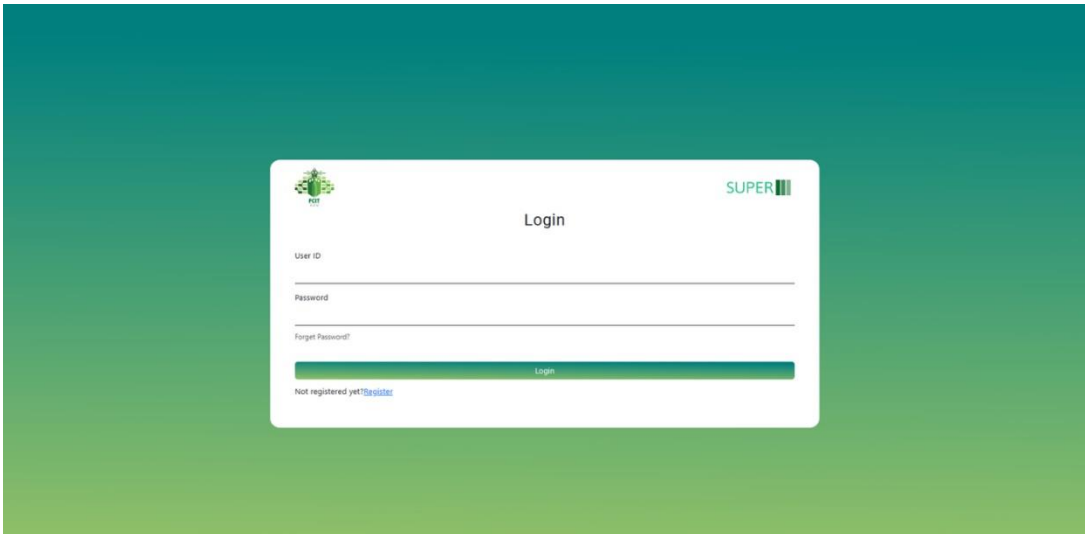

Figure 7.2: Login Interface

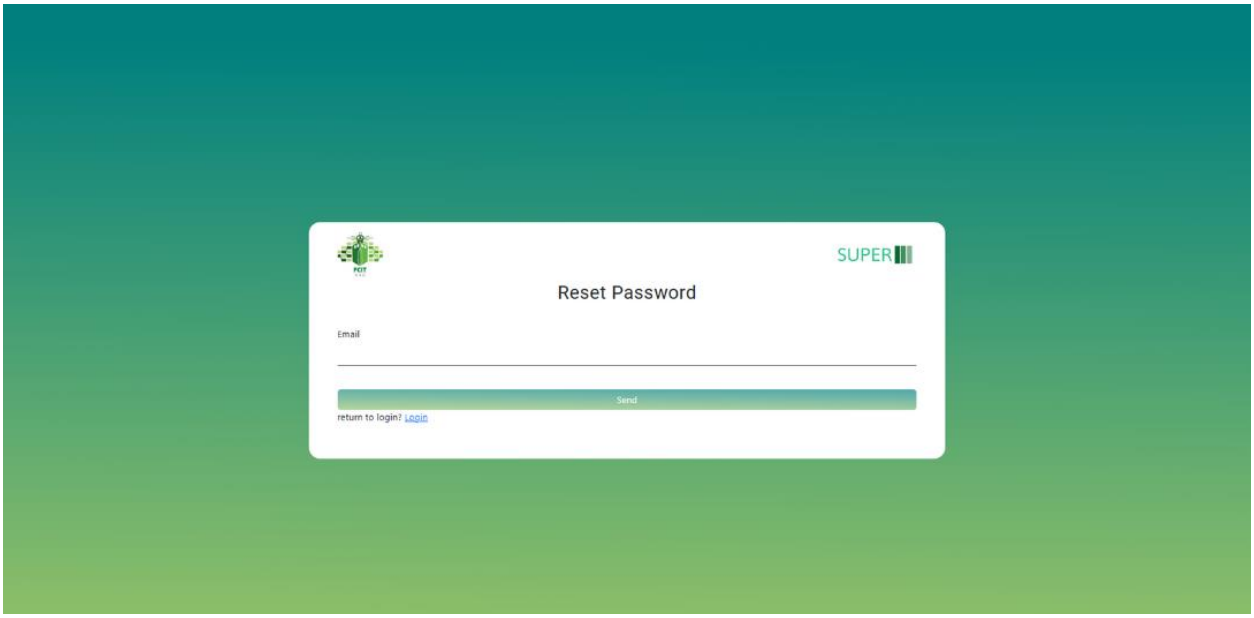

Figure 7.3: Reset Password Interface

Once the student has successfully registered, their information will appear on the coordinator's request page as in Figure 7.4 on the SUPER website. From this page, the coordinator can review the credentials of each registered student as in Figure 7.5 and decide whether to accept them or not, as shown in the figure below.

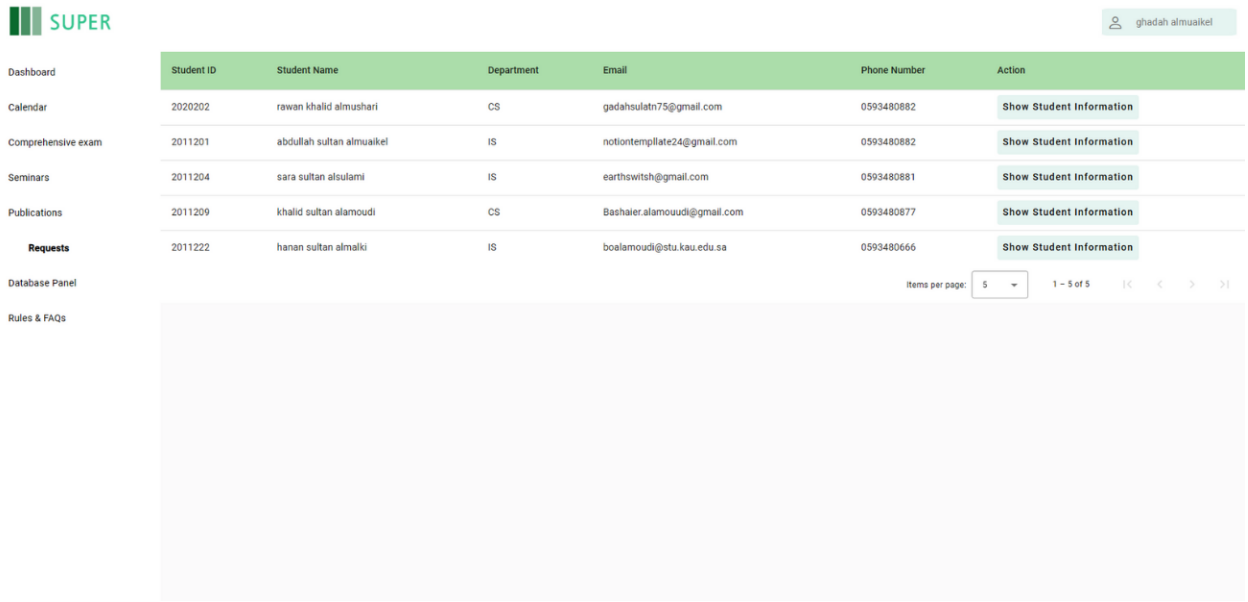

Figure 7.4: Requests Interface

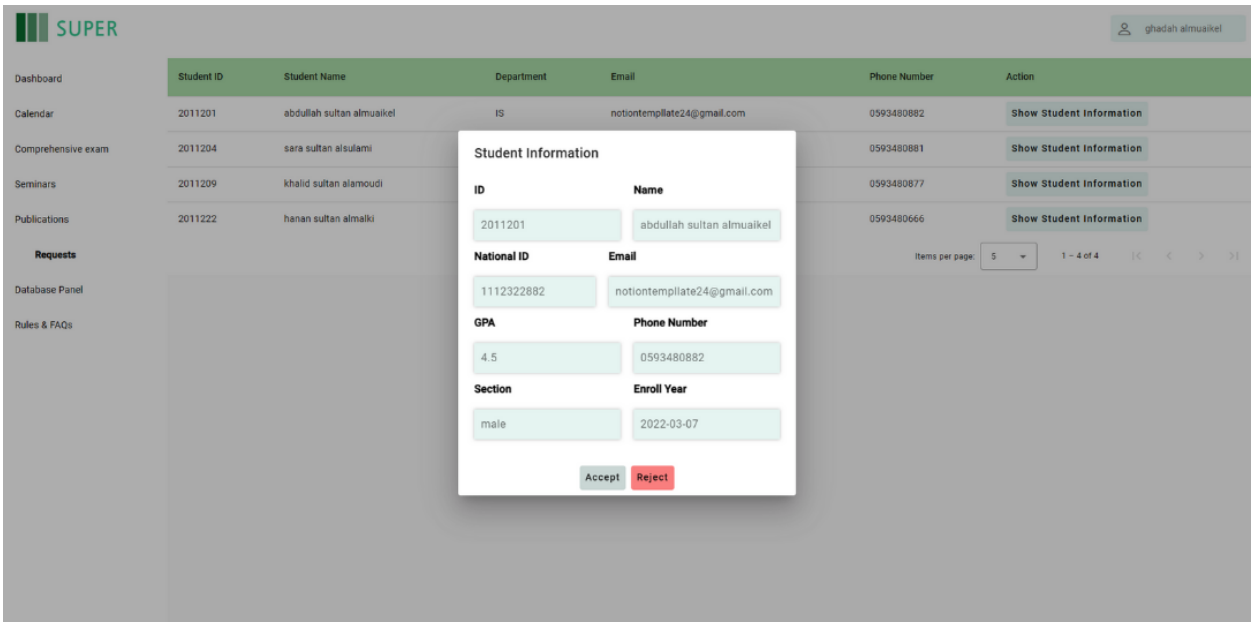

Figure 7.5: The Information Displayed in The Requests Interface.

The "Student Information" page as in Figure 7.6 is designed for students to view all their personal and academic information in one place. This interface not only displays the detailed information provided during registration but also includes features for progress tracking.

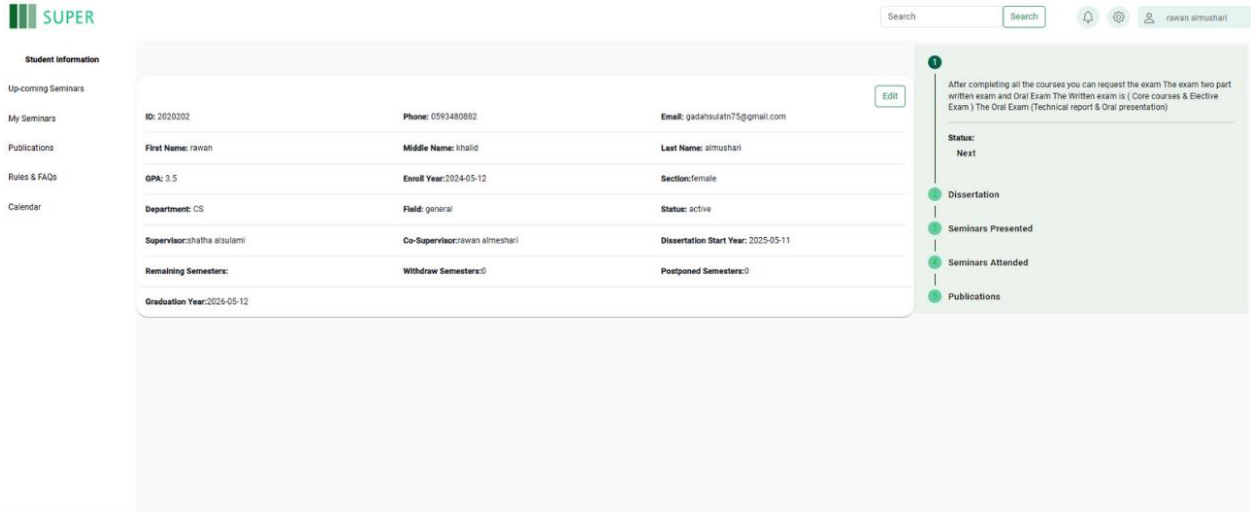

Figure 7.6: Student Information Interface

The calendar page as in Figure 7.7 is accessible to both students and coordinators. It displays all important dates relevant to the academic calendar, However, only coordinators have the authority to add or modify events on the calendar, ensuring that the information remains accurate and up to date. This feature helps both students and coordinators stay organized and informed about key academic milestones.

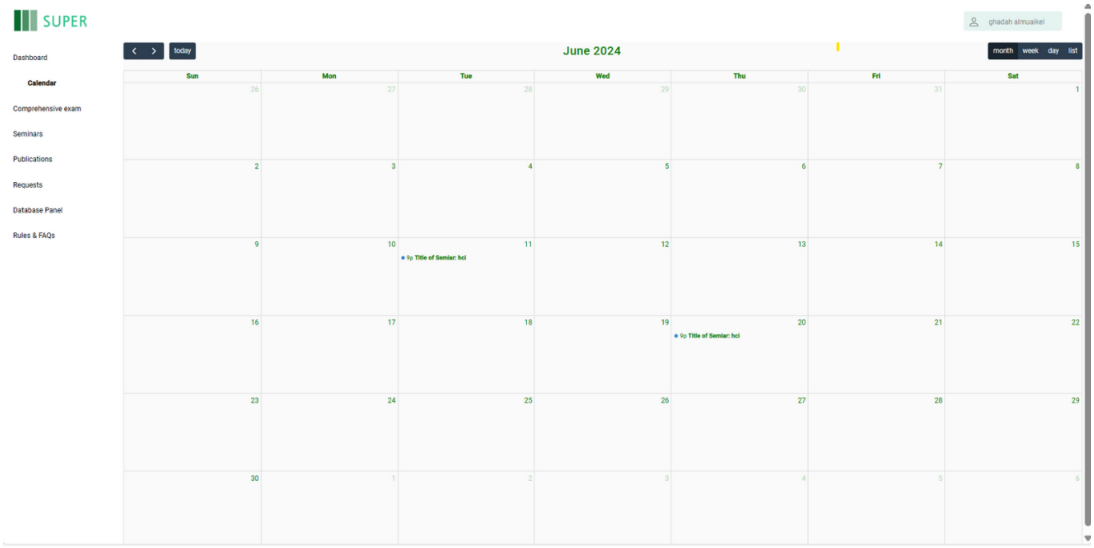

Figure 7.7: Calendar Interface

The "Comprehensive Exam" page as in Figure 7.8 on the coordinator side is designed to manage and grade

comprehensive exams for students. The interface provides a list of students along with their exam details and scores and it will notify the students by their email address.

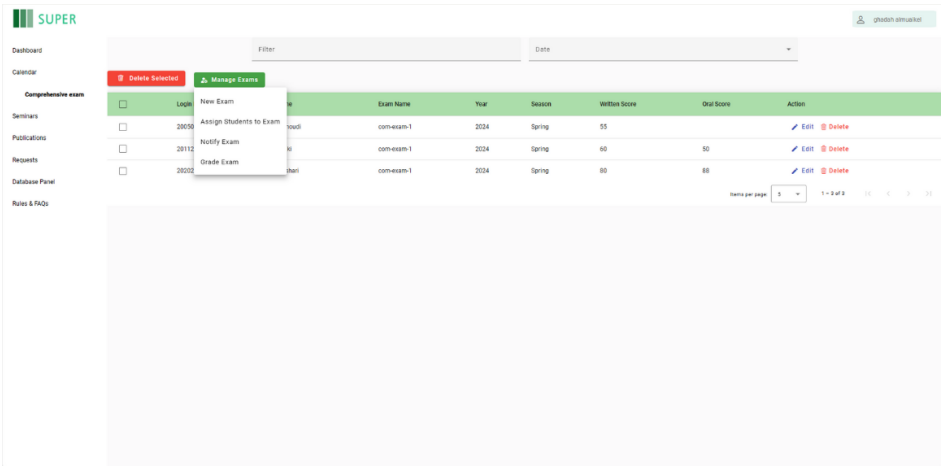

Figure 7.8: Comprehensive Exam Interface

The "Seminar" as in Figure 7.9 page on the coordinator side is designed to manage seminars for students. The interface allows the coordinator to add new seminars, notify students about upcoming seminars, and manage existing seminar entries.

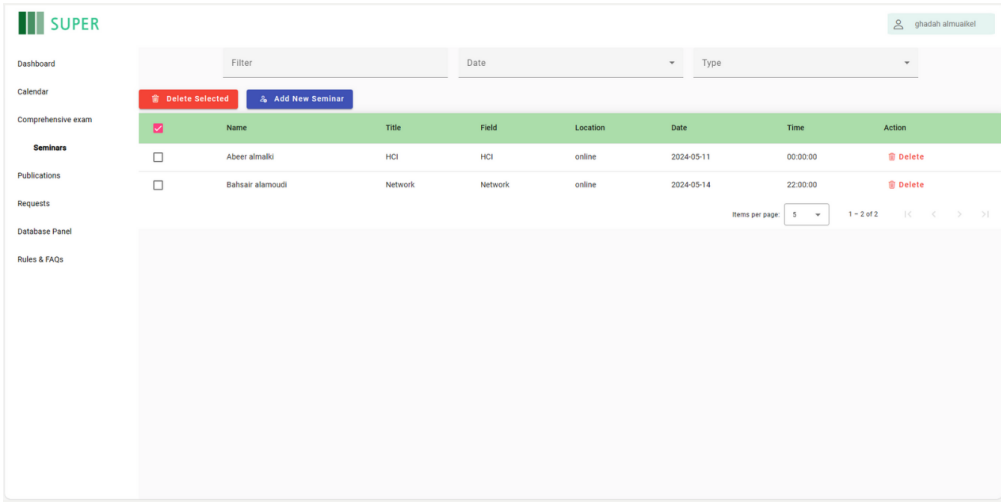

Figure 7.9: Seminars Interface

The coordinator can fill out details for a new seminar as in Figure 7.10, including the type (e.g., Student, public), name if it is public and ID if it is a student, title of the seminar, field of study, location (e.g., an online link), time, and date. Once all details are entered, the coordinator can save the seminar to notify the student and add it to the list of seminars.

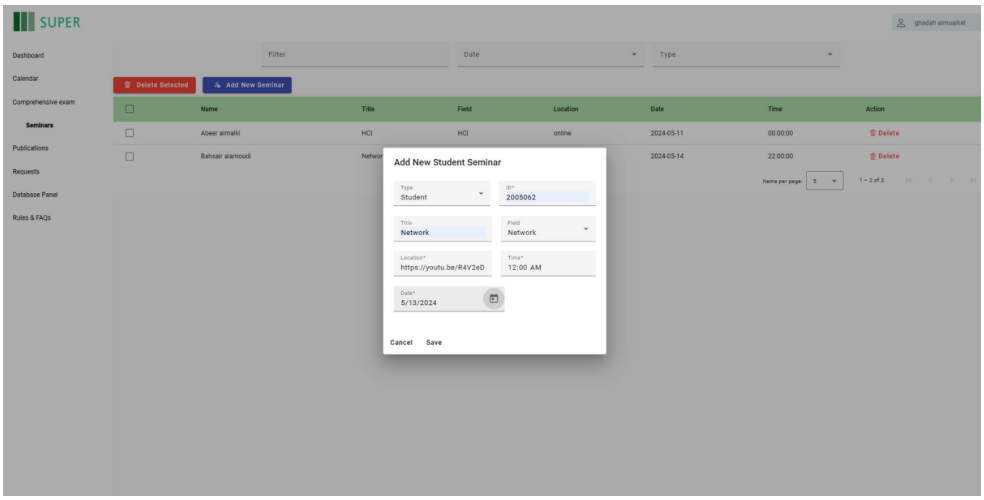

Figure 7.10: Filling the Information of a Seminar

The student side includes "Upcoming Seminars" in Figure 7.12 and "My Seminars" pages in Figure 7.11. "Upcoming Seminars" lists scheduled seminars with details like presenter, title, field, location, date, and time. "My Seminars" has sections for "Presented Seminars" and "Attended Seminars," showing the seminars the student has presented and attended. Students must attend several seminars and present at least two as part of their requirements. These pages help students track and manage their seminar activities efficiently.

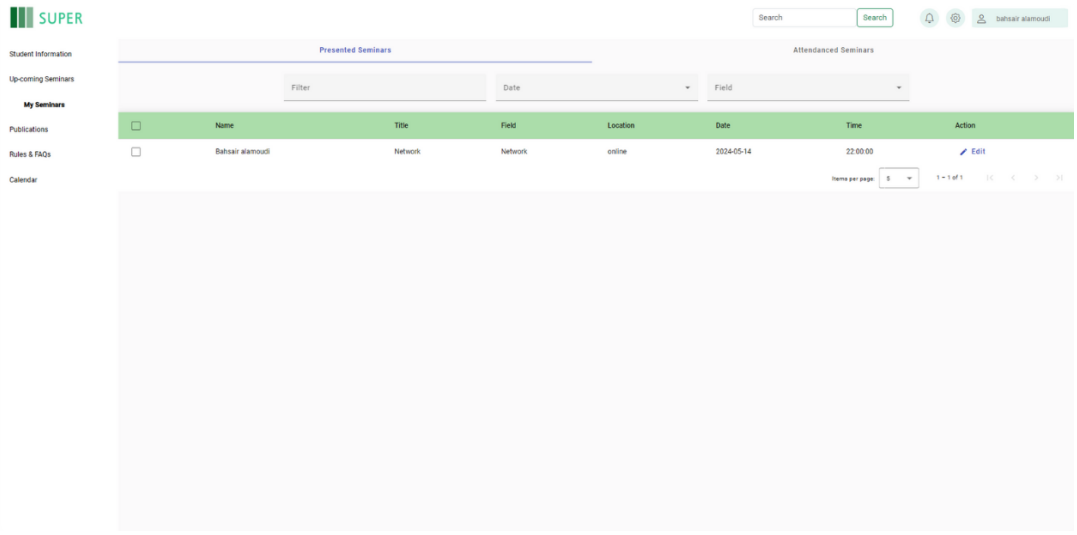

Figure 7.11: My Seminars Interface on the Student Side

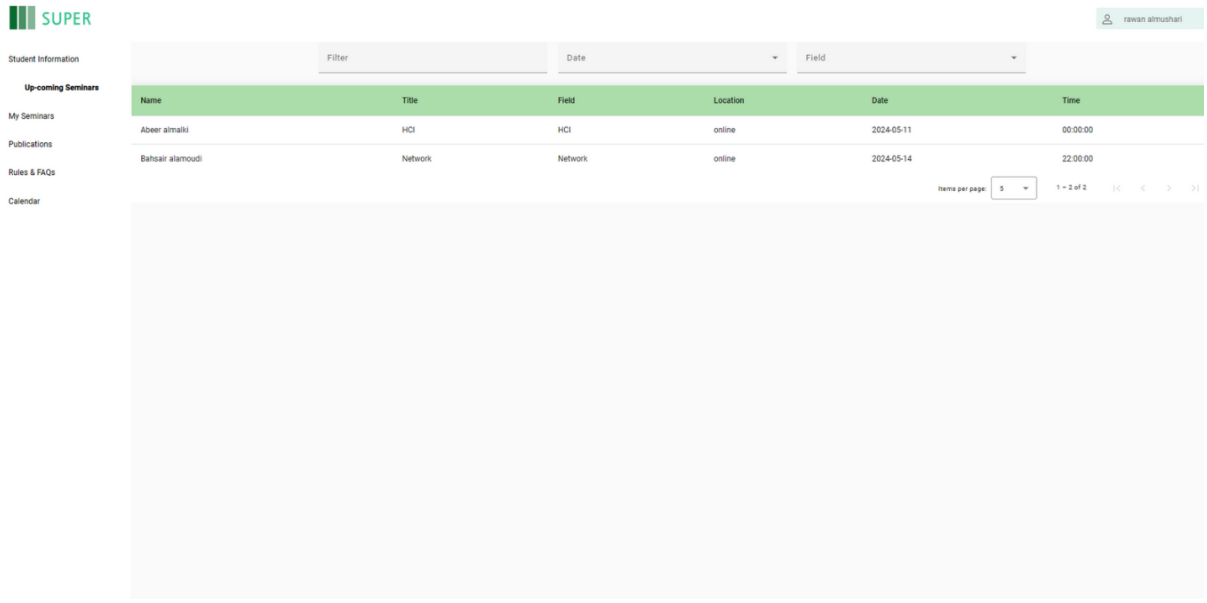

Figure 7.12: My Up-Coming Seminars Interface on the Student Side

"Rules & FAQs" page in Figure 7.13 on the coordinator's interface, which is also visible to students. The page allows coordinators to upload PDFs and add new FAQs for students. It features a Table of Contents on the left side, listing various topics such as student orientation, study plans, and registration. The main panel displays the content of the selected topic, providing detailed information and guidelines for students. This page helps coordinators manage and disseminate important information efficiently, ensuring students have access to essential resources and instructions.

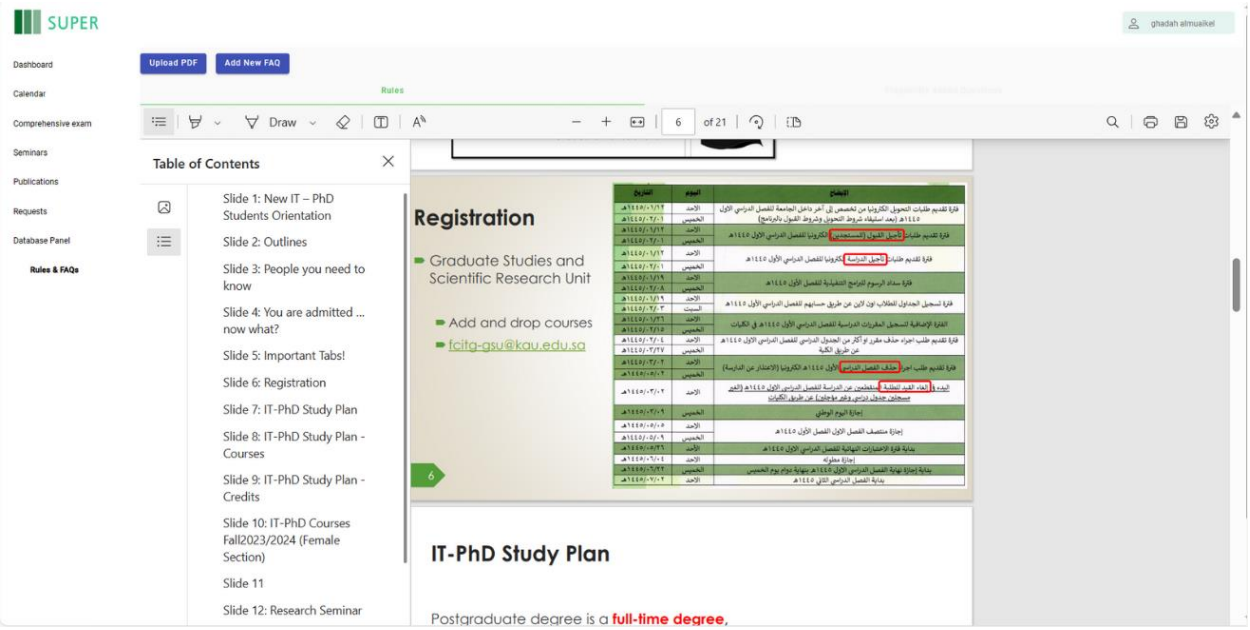

Figure 7.13: Rules & FAQs interface

"Publications" page from both the coordinator in Figure 7.14 and the student in Figure 7.15. Students can upload their publications by providing details such as the title, field, publication type, venue name, DOI, date, and file upload. Coordinators can view, edit, and delete these publications, as well as access detailed information about each publication, ensuring proper management and oversight of student research outputs.

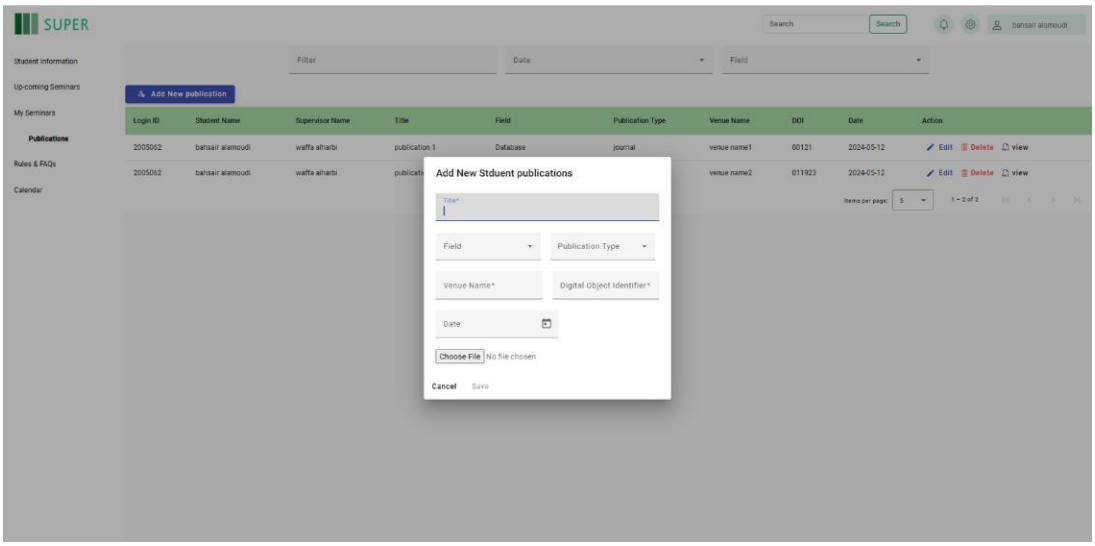

Figure 7.14: Adding a Publication in the Publication Interface.

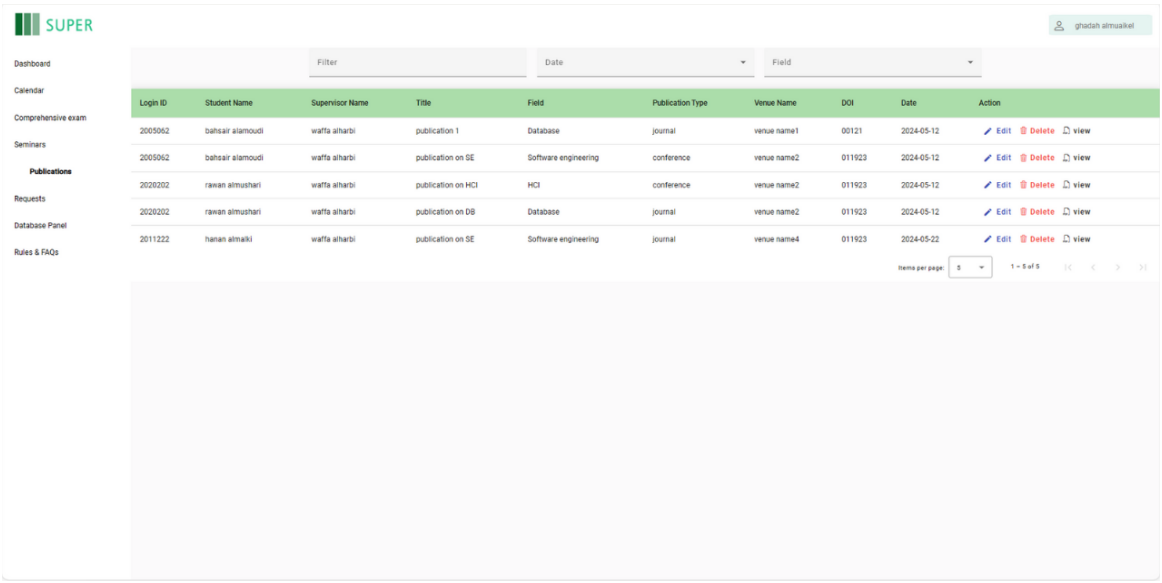

Figure 7.15: The Publication Interface in the Student Side.

"Dashboard" page in Figure 7.16 on the coordinator's interface. The dashboard provides an overview of student statistics, including the total number of students, the number enrolled this year, and those expected to graduate. It features visual representations such as a bar chart showing the status of students (active and not active) and a pie chart depicting the distribution of students across different fields (Database, Software Engineering, HCI). This page helps coordinators quickly access and analyze key student data.

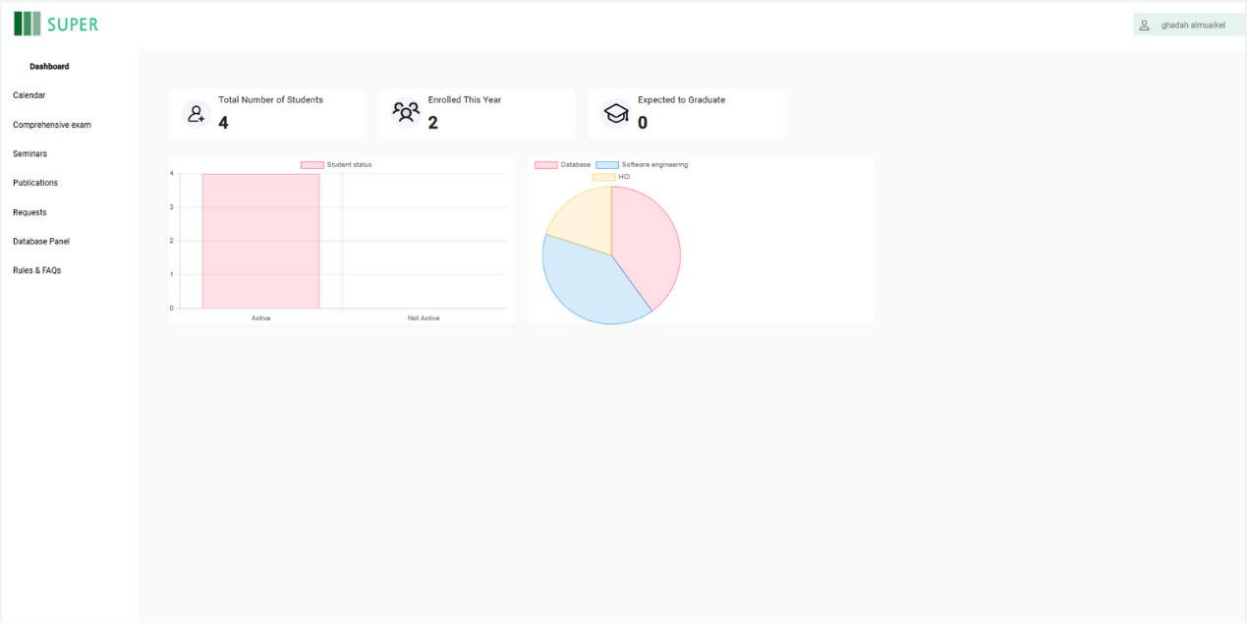

Figure 7.16: Dashboard Interface in the Coordinator Side.

7.3 Achieved Objectives

In our pursuit of creating a centralized and efficient platform for managing PhD academic progress, we set forth a series of ambitious objectives. This section highlights the successful realization of each objective, marking significant milestones in our endeavor to enhance the academic experience for PhD students and coordinators. The achieved objectives encompass the development of a centralized web-based platform, construction of a comprehensive database, design of a student interface, incorporation of a notification system, integration of features for coordinators, effective communication updates, development of a secure login system, and ensuring adaptability and scalability for future requirements.

The first objective centered around the development of a centralized web-based platform that enables PhD students and coordinators to manage academic progress and requirements efficiently. This objective was met successfully as our project team developed a user-friendly interface that provides both students and coordinators with seamless access to critical information and management tools.

Constructing a comprehensive database within the system to hold all student-related information was another pivotal achievement. This database allows coordinators to pull various statistical reports effortlessly, enhancing data management and accessibility. Our commitment to

robust data handling was demonstrated through the efficient organization and retrieval of student information.

Designing an intuitive interface for PhD students where they can visualize their progress, understand their standing in the program, and see the path to their graduation was successfully achieved. This interface empowers students with clear insights into their academic journey, fostering a sense of direction and motivation.

The incorporation of a notification system via emails and a calendar to inform students about important dates and deadlines was implemented successfully. This feature ensures that students remain well-informed about crucial academic events, reducing the risk of missed deadlines and enhancing overall academic planning.

Integrating a feature for coordinators to easily update information, input data, and track student progress in real-time was accomplished with precision. This functionality allows for dynamic and up-to-date tracking of student performance, providing coordinators with a powerful tool for academic oversight.

The objective to share updates, changes in regulations, and other vital information to avoid miscommunications was successfully implemented. This ensures that all stakeholders are consistently informed about the latest developments, fostering a transparent and well-coordinated academic environment.

Developing a user-friendly and secure login system for students and coordinators to access their profiles and manage personal information effectively was another significant milestone achieved. This secure system safeguards user data while providing easy access to personal and academic information.

Ensuring the system is adaptable and scalable to meet future requirements was a key objective that was successfully achieved. The platform's design allows for future enhancements and expansion, maintaining its utility and effectiveness over time.

The achievement of these objectives reinforces our project's commitment to enhancing the management of PhD academic progress. Through the integration of cutting-edge technologies and the meticulous development of user-centric features, our project makes a substantial contribution to the academic experience of PhD students and coordinators.

7.4 Limitations of the Project

The successful implementation of the project, dedicated to enhancing academic progress management for PhD students, is accompanied by inherent limitations that merit acknowledgment. Identifying and understanding these limitations is crucial for providing a comprehensive view of the system's capabilities and pinpointing areas for potential improvement. This section delves into these limitations.

- **Supervisor Access** – The system currently lacks access provisions for supervisors, limiting their ability to directly interact with and monitor the academic progress of their students. This constraint hinders comprehensive oversight and support from supervisors.
- **Language Support** – The platform only supports the English language, posing accessibility challenges for non-English speaking users. This limitation restricts the usability of the system in diverse linguistic environments and necessitates future multilingual support to broaden its accessibility.

Acknowledging these limitations is essential for maintaining transparency and informing users about the current boundaries of the system. These identified limitations can serve as valuable insights for future enhancements and refinements to further fortify the system's effectiveness in managing PhD academic progress.

7.5 Conclusion

In conclusion, this chapter provides a detailed overview of the project navigation guide, achieved objectives, and limitations of the SUPER system. The navigation guide ensures a seamless user experience, while the achieved objectives highlight advancements in managing PhD academic progress. Acknowledging limitations reflects our commitment to continuous improvement. This section serves as a roadmap for future enhancements, ensuring the SUPER system remains an essential tool for supporting students and coordinators effectively

Chapter VIII: Conclusion and Future Work

8.1 Introduction

In this chapter, we encapsulate the core findings and accomplishments of the SUPER project. We provide an overview of the outcomes, draw critical conclusions based on our experiences, and propose future directions to further enhance the platform's capabilities and address its current limitations.

8.2 Conclusion of the Work

The Scholar's Unified Progress & Efficiency Resource (SUPER) project was initiated to tackle the significant challenges faced by PhD students and coordinators at the Faculty of Computing and Information Technology, King Abdulaziz University. These challenges included the absence of a unified system for tracking academic progress, which often led to difficulties in managing deadlines, monitoring progress, and ensuring timely communications between students and coordinators.

The development of SUPER has successfully addressed these issues through the creation of a centralized, web-based platform. Key achievements of the project include:

- **Progress Tracking:** Students now have access to a personalized dashboard that visually represents their academic journey from admission to graduation. This feature highlights completed tasks, ongoing activities, and upcoming deadlines, providing students with a clear view of their progress.
- **Academic Calendar Integration:** The system integrates an academic calendar that automatically updates students about important events, deadlines, and academic milestones. This feature ensures that students are well-informed and can manage their time effectively.
- **Notification System:** The notification system alerts students via email about key dates such as seminar schedules, comprehensive exams, and paper submission deadlines. This reduces the risk of missed opportunities and helps students stay on track.

- **Real-time Data Management:** Coordinators can enter and update information in real-time, which is critical for monitoring student progress and providing timely academic guidance.
- **Secure Authentication:** A robust login and authentication system ensures the privacy and security of users' personal and academic information.
- **Data Visualization Tools:** The implementation of graphs, charts, and other data visualization tools makes it easier for coordinators to understand complex data and derive insights.

These accomplishments have significantly improved the efficiency and effectiveness of PhD program management, creating a supportive and organized academic environment that benefits both students and coordinators.

8.3 Future Work and Handling Limitations

8.3.1 Future Work

- **Implement the Supervisor Side on the Website:** To further enhance the functionality of SUPER, we plan to develop a supervisor-side interface. This will allow supervisors to directly interact with the system, monitor their student's progress, provide timely feedback, and manage their supervisory tasks more effectively. This addition will bridge the gap between students and supervisors, fostering better communication and support.

- **Develop a Version for the Master Program:** Expanding SUPER to include master's programs will extend its benefits to a broader range of graduate students and coordinators. By adapting the system to meet the specific needs of master's students, we aim to provide consistent and efficient academic progress tracking across all postgraduate programs. This expansion will ensure that all graduate students have access to the tools they need to succeed.

- **The Dashboard Will Feature a 'Generate Report' Functionality:** Introducing a comprehensive 'Generate Report' feature will enable users to create custom reports based on specific criteria. This functionality will facilitate more detailed and personalized data analysis, allowing users to generate insights tailored to their needs. This will enhance decision-making processes and provide a deeper understanding of academic progress and program performance.

8.3.2 Handling Limitations

- **Not Having Access as a Supervisor:** One of the current limitations of SUPER is the lack of access for supervisors within the system. By developing a dedicated supervisor module, we will enhance the interactions between supervisors and students, streamline the supervisory process, and ensure that supervisors can provide the necessary support and guidance to their students.

- **Only Support English Language:** At present, SUPER supports only the English language, which limits its usability for non-English speaking users. Future versions should include multi-language support to accommodate a more diverse user base and improve accessibility. By offering language options, we can ensure that the system is inclusive and accessible to all users, regardless of their language preferences.

8.4 Conclusion

The SUPER project has established a robust foundation for efficient PhD program management at King Abdulaziz University. By addressing critical challenges and providing valuable tools for students and coordinators, SUPER streamlines academic progress tracking enhances communication, and offers advanced data analysis tools. Looking ahead, future work will enhance SUPER's capabilities, address current limitations, and expand its scope. Continued development will ensure SUPER remains a cutting-edge, user-friendly solution, ultimately contributing to the success and satisfaction of its users.

Chapter IX: References

[1] "Laravel - The PHP Framework For Web Artisans." https://laravel.com/docs/10.x

[2] "Angular." https://angular.io/docs

[3] M. O. J. T. Contributors and Bootstrap,

"Introduction." https://getbootstrap.com/docs/5.0/getting-started/introduction/

[4] "Learn about the latest Oracle database." https://www.oracle.com/database/technologies/

[5] K. Vyas, "8 Big Advantages of using MySQL | Datamation," *Datamation*, Nov. 01, 2023. https://www.datamation.com/storage/8-major-advantages-of-using-mysql/

[6] "MDN Web Docs," *MDN Web Docs*. https://developer.mozilla.org/

[7] "FCIT coop website," *FCIT COOP*. https://www.fcitcoop.org/Auth

[8] "eProg website," *eProg*. https://app.manchester.ac.uk/

[9] G. van der Steenhoven and P. van Dijk, "ProDoc – A Progress Tracking System for Graduate Students," in *Proceedings of the Seventh Annual Strategic Leaders Global Summit*, University of Twente, Netherlands, April 2007.

[10] S. Hasnan, R. Aziz, and A. A. Hamid, "Postgraduate Tracking System: Student Research Progress Tracking tool," *International Research in Education*, vol. 3, no. 1, Jan. 2015, doi: 10.5296/ire.v3i1.6539.

[11] "Graduate Student Tracking System (GSTS): The Graduate School - Northwestern University." https://www.tgs.northwestern.edu/academic-policies-procedures/graduate-student-tracking-system-gsts.html

[12] Editor, "Non-functional requirements: Examples, types, how to approach," *AltexSoft*, Jul. 26, 2022. https://www.altexsoft.com/blog/non-functional-requirements/

[13] K. Brush, "use case," *Software Quality*, Nov. 28, 2022. https://www.techtarget.com/searchsoftwarequality/definition/use-case

[14] G. Booch, "Object-Oriented Analysis and Design with Applications, Third Edition," *O'Reilly Online Learning*. https://www.oreilly.com/library/view/object-oriented-analysis-

and/9780201895513/#:~:text=by%20Grady%20Booch%2C%20Robert%20A,Wesley%20Professional%0A%0AISBN%3A%209780201895513

[15] "What is Entity Relationship Diagram (ERD)?" https://www.visual-paradigm.com/guide/data-modeling/what-is-entity-relationship-diagram/;WWWSESSIONID=696DEB2424E388F558698E0D49DC7121.www1

[16] A. Jalli, "What is Laravel?," *Built-In*, Dec. 21, 2022. https://builtin.com/software-engineering-perspectives/laravel

[17] C. Deshpande, "What is Angular?: Architecture, Features, and Advantages," *Simplilearn.com*, Jul. 24, 2023. https://www.simplilearn.com/tutorials/angular-tutorial/what-is-angular#:~:text=Angular%20is%20an%20open-source,for%20developers%20to%20work%20with

[18] B. Infinity, "Why should you use Bootstrap? | Board Infinity," *Board Infinity*, Nov. 04, 2023. https://www.boardinfinity.com/blog/why-should-we-use-bootstrap/#:~:text=Bootstrap%20is%20a%20free%20front,with%20other%20optional%20JavaScript%20plugins

[19] "What is Sequence Diagram?" https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/

Appendix A: Data Gathering

A.1 Data Gathering Technique and Discussion

We conducted a series of interviews to understand better the specific needs and concerns of both PhD coordinators and students regarding the current system. This qualitative method was chosen because it could provide specific and complex insights, particularly from a select group of users. PhD coordinators and students were among those interviewed, and they were asked specific questions to elicit detailed feedback on the current system and their hopes for improvements.

The main conclusions drawn from the interviews are as follows:

1. **From the Coordinators' Perspective:**
    - Problems in the old system included manual data collection, time-consuming grading processes, and difficulty in tracking student progress toward graduation.
    - Desired improvements include automated reminders, a more efficient grading system, and better tracking of student progress, including alerts for key academic events.
2. **From the Students' Perspective:**
    - Students faced challenges such as the lack of a centralized system to track academic progress, absence of categorized elective subjects, and insufficient information on seminars.
    - Improvements suggested by students include a more organized system for tracking semesters and withdrawals, categorization of elective subjects, automated updates on seminar attendance, and easy access to seminar information.

General Results:

1. **Lack of Automated Alerts for Academic Milestones and Events:**
    - This resulted in the recommendation to implement a notification system to keep all parties informed on time.
2. **Difficulty in Tracking Students' Graduation Timelines:**
    - Highlighting the need for a system to **Track Academic Progress**, ensuring that students and coordinators can monitor educational milestones efficiently.
3. **No Centralized System for Data:**
    - This leads to the proposal of a centralized platform for **Data Collection and Access** to streamline administrative and academic processes.
    - All the Interviewees agreed to have a PhD website
4. **Requirement of a Dedicated Website for the PhD Program:**
    - Underlining the **Need for a Website for the PhD Program**, which would serve as a hub for all necessary academic information and functions.

Table A.1: Blank Copy of the Coordinator Interview

Interview Outline
Interviewee:
Location:
Objectives: 1. To identify problems experienced by coordinators and students in the existing system. 2. To understand the desired improvements from both the coordinator's and student's viewpoints. 3. To evaluate the necessity and potential impact of a dedicated website for the Ph.D. program. 4. To gather essential insights to guide the design and development of the SUPER project.
Question: 1 What were the problems that bothered you in the old system from the coordinator's perspective?
Question: 2 What were the problems that bothered you in the old system from the student's perspective?
Question: 3 What do you want to improve from the coordinator's perspective?
Question: 4 What do you want to improve from the student's perspective?
Question: 5 Would you recommend creating a dedicated website for the Ph.D. program? (yes/no)

Interview Outline
Interviewe:
Location:
Objectives: 1. To identify problems experienced by students in the existing system. 2. To understand the desired improvements from student's viewpoints. 3. To evaluate the necessity and potential impact of a dedicated website for the Ph.D. program. 4. To gather essential insights to guide the design and development of the SUPER project.
Question: 1 What were the problems that bothered you in the old system from the student's perspective?
Question: 2 Given the absence of a feature displaying the number of semesters taken, remaining, and the available withdrawals or postponements, how do you currently keep track of your academic progress and deadlines?
Question: 3 What do you want to improve from the student's perspective?
Question: 4 To further enhance the seminars page, what specific features or information would you like to see included on this page to support your academic needs?
Question: 5 Would you recommend creating a dedicated website for the Ph.D. program? (yes/no)

Table A.2: Blank Copy of the Student Interview

Table A.3: First Interview

Interview Outline	
Interviewee: Dr. Wafaa Alsggaf	**Interviewers:** Rawan Almeshari Gadah Almuakiel Bashaier Alamoudi Shatha Alsulami
Location: Online/ Google meet	**Appointment Date: 15 August** Start Time: 10:00 AM End Time: 11:30 AM
Objectives: 1. To identify problems experienced by coordinators and in the existing system. 2. To understand the desired improvements from the coordinator's viewpoints. 3. To evaluate the necessity and potential impact of a dedicated website for the Ph.D. program. 4. To gather essential insights to guide the design and development of the SUPER project.	
Question: 1 What were the problems that bothered you in the old system from the coordinator's perspective?	**Answer:** There was no means of collecting data or a unified place to get the data from. One had to collect the data manually.
Question: 2 What were the problems that bothered you in the old system from the student's perspective?	**Answer:** Grading was done manually in Excel sheets, which is a time-consuming process.
	Continue

Continuation	
Question: 3 What do you want to improve from the coordinator's perspective?	**Answer:** The coordinator can view the annual seminars presented by the students and assign the grades through the website. A notification of the grade is sent via email.
Question: 4 What do you want to improve from the student's perspective?	**Answer:** The student can access information about the comprehensive exam: the grade and when it was completed.
Question: 5 Would you recommend creating a dedicated website for the Ph.D. program? (yes/no)	**Answer:** Yes.

Table A.4: Second Interview

Interview Outline	
Interviewee: Dr. Reem Alotaibi	**Interviewers:** Rawan Almeshari Gadah Almuakiel Bashaier Alamoudi Shatha Alsulami
Location: Building 61/ FCIT	**Appointment Date: 3 September** Start Time: 12:00 AM End Time: 12:30 AM
Objectives: 1. To identify problems experienced by coordinators and in the existing system. 2. To understand the desired improvements from the coordinator's viewpoints. 3. To evaluate the necessity and potential impact of a dedicated website for the Ph.D. program. 4. To gather essential insights to guide the design and development of the SUPER project.	
Question: 1 What were the problems that bothered you in the old system from the coordinator's perspective?	**Answer:** The coordinator cannot identify students who are close to graduation, as well as those whose standard duration has ended, to inform them or communicate with them. This used to take a lot of time to resolve.
	Continue

Continuation	
Question: 2 What were the problems that bothered you in the old system from the student's perspective?	**Answer:** There is no service to notify students of important events, like notifying a student who has an oral exam only in case of passing the written exam.
Question: 3 What do you want to improve from the coordinator's perspective?	**Answer:** The existence of an automatic reminder service, which sends the required information when it becomes available.
Question: 4 What do you want to improve from the student's perspective?	**Answer:** Notify the student when the end of the period is approaching, remind the student about the date of the comprehensive exam, inform students who have an oral exam only in case of passing the written exam, and when collecting preferences for the second semester, the system must ensure that the student has studied the core subjects and alert them that it is preferable to monitor the seminars presented each year after registering the thesis.
Question: 5 Would you recommend creating a dedicated website for the Ph.D. program? (yes/no)	**Answer:** Yes.

Table A.5: Third Interview

Interview Outline	
Interviewee: Dr. Manal Kalkatawi	**Interviewers:** Rawan Almeshari Gadah Almuakiel Bashaier Alamoudi Shatha Alsulami
Location: Online/ Google Meet	**Appointment Date: 6 September** Start Time: 10:00 PM End Time: 11:00 PM
Objectives: 1. To identify problems experienced by coordinators and in the existing system. 2. To understand the desired improvements from the coordinator's viewpoints. 3. To evaluate the necessity and potential impact of a dedicated website for the Ph.D. program. 4. To gather essential insights to guide the design and development of the SUPER project.	
Question: 1 What were the problems that bothered you in the old system from the coordinator's perspective?	**Answer:** No Centralized System for The PhD program, so it is difficult to access the data in the regular way which is done manually.
Question: 2 What were the problems that bothered you in the old system from the student's perspective?	**Answer:** The student is unable to know if they are entitled to defer or withdraw from the semester.
	Continue

Continuation	
Question: 3 What do you want to improve from the coordinator's perspective?	**Answer:** Showcasing the number of seminars attended by the student should be sufficient, and it is not necessary to mention those seminars.
Question: 4 What do you want to improve from the student's perspective?	**Answer:** Some features do not always appear, but it depends on their period (choosing the type of comprehensive exam – seminars links)
Question: 5 Would you recommend creating a dedicated website for the Ph.D. program? (yes/no)	**Answer:** Yes.

Table A.6: Fourth Interview

<table>
<tr><th colspan="2">Interview Outline</th></tr>
<tr><td>Interviewee:
Dr. Tareq Mohammed</td><td>Interviewers:
Rawan Almeshari
Gadah Almuakiel
Bashaier Alamoudi
Shatha Alsulami</td></tr>
<tr><td>Location:
Online/ Google Meet</td><td>Appointment Date: 6 September
Start Time: 7:00 PM
End Time: 8:00 PM</td></tr>
<tr><td colspan="2">Objectives:

1. To identify problems experienced by coordinators and in the existing system.
2. To understand the desired improvements from the coordinator's viewpoints.
3. To evaluate the necessity and potential impact of a dedicated website for the Ph.D. program.
4. To gather essential insights to guide the design and development of the SUPER project.</td></tr>
<tr><td>Question: 1

What were the problems that bothered you in the old system from the coordinator's perspective?</td><td>Answer:

The department head is unable to digitally determine the number of students under each supervisor, which complicates the process of understanding the workload.</td></tr>
<tr><td>Question: 2

What were the problems that bothered you in the old system from the student's perspective?</td><td>Answer:

There is no place for uploading research and publications files</td></tr>
<tr><td></td><td>Continue</td></tr>
</table>

Continuation	
Question: 3 What do you want to improve from the coordinator's perspective?	**Answer:** The possibility of forming committees digitally.
Question: 4 What do you want to improve from the student's perspective?	**Answer:** There should be an identification card for the supervisor to help the student in choosing them, and the user should be able to choose the types of notifications they want to receive in their email, with the ability to specify the receiving period.
Question: 5 Would you recommend creating a dedicated website for the Ph.D. program? (yes/no)	**Answer:** Yes.

Table A.7 Fifth Interview

Interview Outline	
Interviewee: Dr. Muna Abu Ghazalah	**Interviewers:** Rawan Almeshari Gadah Almuakiel Bashaier Alamoudi Shatha Alsulami
Location: Online/ Google Meet	**Appointment Date: 5 September** Start Time: 7:00 PM End Time: 8:00 PM
Objectives: 1. To identify problems experienced by students in the existing system. 2. To understand the desired improvements from student's viewpoints. 3. To evaluate the necessity and potential impact of a dedicated website for the Ph.D. program. 4. To gather essential insights to guide the design and development of the SUPER project.	
Question: 1 What were the problems that bothered you in the old system from the student's perspective?	**Answer:** No place shows the number of semesters taken, remaining semesters, and the number of times withdrawals or postponements are available.
Question: 2 Given the absence of a feature displaying the number of semesters taken, remaining, and the available withdrawals or postponements, how do you currently keep track of your academic progress and deadlines?	**Answer:** Currently, I manually keep track of my academic progress and deadlines using a combination of personal calendars and university-provided documents. It's a process that requires constant updating and cross-referencing with official academic schedules.
	Continue

Continuation	
Question: 3 What do you want to improve from the student's perspective?	**Answer:** The elective subjects should be categorized into groups to understand the type of track, along with reminders for the upcoming seminars. Also, to have a page for the seminars.
Question: 4 To further enhance the seminars page, what specific features or information would you like to see included on this page to support your academic needs?	**Answer:** The seminars page should include direct links to upcoming seminars for easy access. Additionally, it would be extremely helpful if the page could automatically update my attendance record when I attend a seminar.
Question: 5 Would you recommend creating a dedicated website for the Ph.D. program? (yes/no)	**Answer:** Yes.

A.2 User Guide

This guide is for users of the Super website, which serves both students and coordinators. The following steps are crucial for a smooth experience.

- Student Side

1- Open Super Website using your preferred web browser.

2- Navigate to the “Register” Page

3- Fill in the required information to create a student account and wait for the approval and your password.

4- Login to Your Account

5- you will be redirected to the student information page, which provides an overview of your activities and your progress.

6- Navigate to the “UP COMING SEMINARS” Page to view all upcoming seminar to attend.

7- Navigate to the “CALENDER” page to view important event and dates the coordinator highlighted.

8- Navigate to the “MY SEMINARS” Page to view the seminar you will present.

9- Navigate to the “PUBLICATIONS” Page to add your final publication for the coordinator to see.

10- Navigate to the “RULES AND FAQs” Page to see frequently asked question and their answers and if coordinator uploaded a file.

- Coordinator Side

1- Open Super Website using your preferred web browser.

2- Coordinator should have an account from the admin.

3- Login to Your Account

4- you will be redirected to the coordinator dashboard, which provides an overview of student’s progress.

5- Navigate to the “SEMINARS” page. Here, you can add new seminars by filling in the necessary details, edit existing seminars, or delete seminars that are no longer needed.

6- Navigate to the “CALENDER” page to add new event for the students.

7- Navigate to the “PUBLICATIONS” page. See students’ publications, edit existing ones, or remove outdated publications.

8- Navigate to the “COMPREHENSIVE EXAMS” page. Schedule upcoming exams, update exam details, assign student to take the exam, and enter or update student grades.

9- Navigate to the “RULES AND FAQs” page. Add new rules and guidelines file, edit existing ones, or delete rules that are no longer applicable.

10- Navigate to the “DATABASE PANEL” page to monitor recent activities and keep track of important updates.