**National University of Computer and Emerging Sciences, Lahore**


# FASTGlide

Asmar Shahid 21L-1754 BS(CS)

Abdul Wahab 21L-5291 BS(CS)

Muhammad Saad Sohail 21L-5344 BS(CS)

Supervisor: Dr. Zeeshan Ali Khan

Final Year Project

March 21, 2025

# Anti-Plagiarism Declaration

This is to declare that the above publication was produced under the:

**Title: FASTGlide**

is the sole contribution of the author(s), and no part hereof has been reproduced as it is the basis (cut and paste) that can be considered Plagiarism. All referenced parts have been used to argue the idea and cited properly. I/We will be responsible and liable for any consequence if a violation of this declaration is determined.

Date: December 8, 2024

Name: Asmar Shahid

Signature: .....

Name: Abdul Wahab

Signature: .....

Name: Muhammad Saad Sohail

Signature: .....

---

## Author's Declaration

This states Authors' declaration that the work presented in the report is their own, and has not been submitted/presented previously to any other institution or organization.

## Abstract

The Final Year Project (FYP) is a significant aspect of a student's degree. It tests whether the student has successfully learnt the concepts and skills that were taught during the degree program. It also evaluates the student's work ethic, technical skills, and domain knowledge. However, the entire process of registering for the FYP , finding a group, choosing a supervisor, and completing other requirements is hectic and requires frequent trips to the university. Supervisors and coordinators also face challenges in managing all their tasks. FASTGlide aims to assist all those involved in the FYP process by automating and streamlining the entire process from registration to project completion.

## Executive Summary

The Final Year Project (FYP) stands as a compulsory academic requirement worked towards by students undergoing their final year of undergraduate studies. In computing programs, for instance, students could either develop programs addressing real life problems or explore various advanced computer science areas or work on other goals oriented towards mastering subjects offered in their academic years.

At FAST-NUCES, and like most other universities, the FYP is equally important throughout the period of undergraduate studies. FYP involves several steps including registration, submission of interim project reports and it ends with evaluation. In each of these situations, students, and faculty members including FYP coordinators, have to deal with many issues and administrative problems.

Usually, students experience problems on how to group together without a specific platform for this purpose. They either use their connections or turn to the FYP coordinators asking for help in sending out a mass email to find group members. After groups when formed, other than being challenged by the logistics of finding members, students are also faced with the problem of how to approach a supervisor which often means movement from one teacher's classroom to another making a shy pitch of the project to each one of them – a resource wasting activity which often leads to several visits.

On the other hand, supervisors get so many emails and see many students who want to be advised on some issues, or to get a decision on submitting a project. Once they agree to be the supervisors of some groups, handling several groups of students becomes a problem. FYP coordinators, who manage the process as a whole, are also equally worse off as they have to deal with updating group information or ensuring the right format is used for every submitted work in a very manual manner.

Through the use of the latest web technologies, our project FASTGlide endeavours to resolve these problems by transforming and improving the Final Year Project Process. FASTGlide plans to reduce the effort of all the parties by reorganizing the processes of forming groups, assigning supervisors, submission of deliverables and their follow ups to make the proceedings more organized and easy.

On the technical side of things, the FASTGlide software development is based on the MERN (MongoDB, Express, React, Node.js) technology stack which enables dynamic and scalable web applications. The front-end is developed using React which makes the interface easy to use by providing a simple interaction with the users, whereas the back-end is developed using Express and Node.js which implements the API and the server-side logic. The database used in this solution is MongoDB which is a NoSQL database that allows for agile development of applications with complex data types such as students, supervisors, and their deliverables.

As part of our research work, we also explored other similar applications like Google Classroom, Base-

camp, and MS Teams, among others. While these offer features for project management and team collaboration, none of them was developed primarily to support the FYP process. In contrast, FAST-Glide is a solution which takes care of the particular issues that students, supervisors, and coordinators at FAST-NUCES encounter during the FYP cycle.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1 Introduction

Final Year Project (FYP) is a required academic assessment undertaken by students in their last year of undergraduate studies. Students studying in computing programs create applications for real-world problems, research in advanced subjects of computer science, or work on some other goal with the purpose of demonstrating their proficiency in the subjects they studied during their degree.

FAST-NUCES, like other universities, also has the FYP requirement for all undergraduate students. The FYP process begins with registration, consists of submissions of multiple deliverables, and culminates with the final evaluation. Throughout the process, all the people involved in it have to undergo a lot of tedious work.

The students have to worry about finding like-minded individuals to form groups with. There is no channel for them to do so and they have to rely on asking their friends or the FYP coordinators to send emails to help them find group members. Once this task is done with, the group members have to find a supervisor for their project. They have to visit the offices of different teachers multiple times, and have to explain their project idea separately to all of them.The teachers who take the role of FYP supervisor are disturbed by constant emails and officevisits by students looking for a supervisor. After they have finalized their groups, they find it hard to keep track of all of them. The FYP coordinators, in particular, have to do a lot of work manually. For example, if any student wishes to change their group details, the coordinators have to manually make the changes. They have to go through every single deliverable to verify that it follows the correct format.

Our project FASTGlide seeks to resolve the above-mentioned issues and modernize the FYP process through modern web technologies. It will solve the above-mentioned problems and make the process easier.

## 1.1 Purpose of this Document

The purpose of this document is to outline the various challenges faced in the FYP process and how our application FASTGlide aims to solve those challenges. It clarifies the goals and objectives of our project, as well as the various technologies and strategies we will be using to fulfill those. Throughout this report, we will be discussing the scope of our project, overview of related applications, and the detailed design and functionalities of our application.

## 1.2 Intended Audience

The document's intended audience include all the individuals involved in some way in the FYP registration and supervision processes. These include the supervisors, coordinators, and students undertaking the FYP. This document will help them understand how making use of this web application can significantly reduce their tedious work and streamline the required processes.

## 1.3 Definitions, Acronyms, and Abbreviations

Important definitions, acronyms, and abbreviations used in this document are listed below.

**FYP**: Final Year Project
**UI**: User Interface
**UX**: User Experience
**REST**: Representational State Transfer
**ORM**: Object-Relational Mapping
**CRUD**: Create, Read, Update, Delete
**API**: Application Programming Interface
**AI**: Artificial Intelligence
**SDG**: Sustainable Development Goal
**JWT**: JSON Web Token
**RBAC**: Role Based Access Control

## 1.4 Conclusion

The first chapter of this report gives an introduction to FASTGlide, its intended audience, and the multiple terms, acronyms, and abbreviations used throughout the whole report. The second chapter will clarify the vision and scope of our project, as well as any business oppurtunities alongwith the stakeholders' details. The goals and objectives of our project are also clearly defined. The third chapter involves thorough research of related applications in the market. The fourth chapter outlines the functionalities, requirements, and use cases of FASTGlide. The fifth and last chapter elaborates on the design and architecture of the application and illustrates it with multiple diagrams.

# Chapter 2 Project Vision

The proposed solution of FASTGlide is built upon the observation that the Final Year Project (FYP) system at FAST-NUCES is cumbersome and time-consuming. The goal of the platform is to practically eliminate all the intermediaries and make all stages starting from registration and ending up with the completion of the FYP as smooth as possible for students, their supervisors and FYP coordinators. To achieve its stated goals, FASTGlide seeks to utilize current-generation web technologies and artificial intelligence in enhancing collaboration between the various entities.

## 2.1 Problem Domain Overview

There are some problems existing in the current FYP process at FAST–NUCES. Some challenges students experience include; formation of working groups, identification of supervisors and assurance of compliance with production standards. Supervisors can fail to monitor the progress of several projects and interruptions from students. On the other hand, FYP coordinators find themselves dealing with several manual processes, including checking of deliverables and making group alterations which is inefficient and can be very incorrect.

## 2.2 Problem Statement

The FYP process as it is practiced at FAST-NUCES is somewhat lengthy and convoluted at the moment. The challenges range from group formation, identification of supervisors and on the other side the supervisors struggle with managing different projects. FYP coordinators receive the burden of entering their own data, verification of students' deliverables and updating the group information. This calls for an automated system that will handle the entire process.

## 2.3 Problem Elaboration

Currently, there exists no formal system that can help students to find group members. When the groups are formed students are forced to go and explain their ideas to the supervisors severally. As a result, for supervisors it results in confusion and time management issues. Coordinators are responsible for manually entering group details adjustments and changes to formats of the submissions, thus contributing to an inefficient and error-prone procedure. These inefficiencies harm everyone involved, which is why they require an improved approach.

## 2.4 Goals and Objectives

The goal of our project is to create an automated workflow for FYP submission in FAST. We aim to achieve the following objectives:

- To help the students undertaking FYP in finding groups and relevant supervisor for their project
- To assist the supervisors in managing the projects they are supervising and monitor their progress.
- To automate checking of deliverables and other tasks which are currently performed manually by FYP coordinators

## 2.5 Project Scope

FASTGlide will be a web-based application, making it accessible across platforms. The project aims to benefit final-year students, supervisors, and the FYP committee at FAST-NUCES School of Computing. It will assist students in forming project groups, supervisors in overseeing multiple projects efficiently, and coordinators in automating the verification of deliverables and other repetitive tasks. Through this system, the FYP process will be more organized, reducing the workload for all involved parties and enhancing the overall experience.

## 2.6 Sustainable Development Goal (SDG)

FASTGlide is targeting the Sustainable Development Goal (SDG) of Quality Education. FASTGlide aims at enhancing the efficiency and accessibility of the process of Final Year Project (FYP) for students. It eases the load of managing and coordinating interactions among students, supervisors, and administrators, ensuring smooth education processes. Easy interaction, receiving and giving constructive criticism, and tracking the progress of the project in this case FASTGlide helps to eliminate impediments to access quality learning and enhances creativity in managing academic projects and initiatives.


**Figure 2.1: Quality Education**

## 2.7 Constraints

Every one of FASTGlide's potentialities mentioned earlier has its drawbacks too. First, there are budgetary limits that may slow down project development and limiting project scope as with insufficient money it may be difficult to acquire resources. Resources may be some constraints to the development of the project, especially when it comes to the technical side of the platform and making the same available on all devices. It is possible that there may be other hurdles resulting from internal governing policies, such as regulation of academic administration. Lastly, it is important to emphasize the need of users - it may take time to convince students, supervisors, and coordinators to shift from manual work to computerized systems.

## 2.8 Business Opportunity

Through the boosting of project inefficiencies, FASTGlide brings a very lucrative market opportunity relating to the FYP process. The system has potential for wider usage in other universities as it provides them with a good project management system. Besides, FASTGlide is such a well-known system that it aims offering additional services, including but not limited to supervisors' performance analytics and students' AI help in finding project ideas which creates possibilities for launching new subscription-based or licensing models of the platform in the future.

## 2.9 Stakeholders Description/ User Characteristics

FASTGlide is participative in a number of relevant stakeholders who contribute significantly to the Final Year Project (FYP) process and these are students, supervisors and FYP coordinators. The system, the role played by each individual as well as guidelines on how each of them interacts with the system differs quite a lot.

### 2.9.1 Stakeholders Summary

#### 2.9.1.1 Students

Students remain the first and frequent users of the FASTGlide platform. They are mostly in their undergraduate final year and have to fulfill the FYP requirement for graduation. In theirs, they seek to meet and recruit persons for project groups, a suitable project supervisor and timely submission of project deliverables. At present, students are forced to use informal groups and contacts or the help of the coordinators to join groups. This procedure, FASTGlide provides as an opportunity to search for future

group members through targeted interest or skills. In addition, students will have a single application where they will perform all actions related to the FYP coordination – submitting project ideas, project deliverables or feedback requests and tracking supervisor's comments.

#### 2.9.1.2 Supervisors (Faculty Members)

The involvement of faculty members as project supervisors is necessary for steering students throughout the FYP process. Supervisors have issues in terms such as trying to manage student groups that are in the same level and evaluating the extent of their work and giving the necessary comments. In the case of FASTGlide, supervisors will have a dashboard that will ease the management of all the groups that have been assigned to them. They will manage deliverables, and even track the milestones that have been achieved, and interact with students easily. Further, the system will assist the supervisors in time management by cutting down on repeated visits to the office and individual emails from students.

#### 2.9.1.3 FYP Coordinators

The activities of the students are from the time when the students are registered to the time the projects are physically completed and that is the FYP coordinators' duty. Nowadays, coordinators are greatly overburdened because paperwork that is supposed to be automated is still performed manually. For instance, capturing changes in the group members, confirming whether the task was completed, and making sure that the work meets university standards; all these officials still perform them manually. Majority of these activities will be handled by FASTGlide and therefore the workload for the coordinators is bound to reduce. The site will give guidance on the correct procedures for submitting group documents and checking the final submission. This automation will lower the number of routine tasks done and also help to keep the FYP process on track.

### 2.9.2 Key High-Level Goals and Problems of Stakeholders

Different stakeholder groups have different goals and challenges, which FASTGlide will endeavor to understand and tackle:

#### 2.9.2.1 Students

**Goals:** Create a central web platform that makes the process of finding groups, getting a supervisor, and submitting the project deliverables easier.

**Problems:** The current situation is rather inefficient and takes quite a lot of time. It forces students to come to university several times and manually communicate with students and faculty.

#### 2.9.2.2 Supervisors (Faculty Members)

**Goals:** Enhancing their ability to multitask on projects, monitor student activities, and return comments on time

**Problems:** Emailing back and forth or making a trip to the office is an extremely inefficient system which has room for errors leading to missed appointments or deadlines.

#### 2.9.2.3 FYP Coordinators

**Goals:** Replace these manual processes, for example, verification of deliverables, making updates to group databases, as well as report writing by employing FYP system.

**Problems:** Inconvenience posed by manual handling of task processes, risk of underperformance due to human factor.

# Chapter 3 Literature Review / Related Work

In the domain of educational technology, there are a number of applications that are designed to make the process of learning, working together and managing projects more efficient. This section covers a number of tools that are already available in the academic or professional use and focuses on their advantages as well as the relation of these tools to the concept of FASTGlide, which is intended to revolutionize and improve the Final Year Project (FYP) process.

## 3.1 Definitions, Acronyms, and Abbreviations

Important definitions, acronyms, and abbreviations used in this section are listed below.

**FYP**: Final Year Project
**AI**: Artificial Intelligence
**LMS**: Learning Management System

## 3.2 Detailed Literature Review

### 3.2.1 Google Classroom

In educational institutions, Google Classroom [**?** ] is an online course management system that allows making coursework, assignments, and communication easier. It also enables educators to set up classes, send the coursework to students, mark it and interact among students and instructors. It is a Google workspace integrated system; it provides collaborative working business tools such as Docs, Sheets, and Slides for group work, plus a cloud net storage system called Google Drive.

#### 3.2.1.1 Critical analysis of the related work

There is no doubt that google classroom can help one manage their assignments as well as submit them, however it does not have the adequate tools to support the intricacies involved with undertaking a final year project. This general purpose features, although are practical do not provide the necessary essential elements of project management such as the tracking of progress, management of milestones achievements, and faculty and students communication that is needed for projects like the FYPs. It also lacks provision for advanced analytics and an effective proposal submission and approval process with well-defined hierarchies, which in fact are central in achieving the objectives of FASTGlide.

#### 3.2.1.2 Relationship to the proposed related work

Google Classroom and FASTGlide share the common objective of easing the students' academic burden, however, Google Classroom is more course management oriented. FASTGlide on the other hand is designed for the FYP processes and provides functionalities like milestone management, communication between supervisors and students, and automating proposal processes making it suitable for the academic project management cycle.

### 3.2.2 Gradescope

Gradescope [**?** ] is a system that assists in marking quizzes and tests, tailored towards enhancing the efficiency of the grading processes for the instructors. It incorporates tools for AI grading, assessing using a grading rubric and LMS integration.

#### 3.2.2.1 Critical analysis of the related work

For large classes or courses with complex grading tasks, Gradescope shines in grading speed. The tool, however, does not reach to areas such as coordination of students and faculties in relation to projects in real time, which is a very crucial aspect in an FYP. Furthermore, its main focus is on evaluation processes rather than the management and automation of the project life cycle.

#### 3.2.2.2 Relationship to the proposed related work

As much as Gradescope concentrates on automating the grading process, FASTGlide goes a step further and integrates proposal focusing on the project management aspect of a project in its proposal, milestone reporting, and collaboration. There is a broader range of tasks performed in FASTGlide other than grading which includes project performance measurements at various levels which are not covered by Gradescope.

### 3.2.3 Microsoft Teams

Microsoft Teams [**?** ] is a versatile platform for communication, enabling instant messaging, video conferencing, data storage as well as integration of other applications. It is often referred to as a team working space within educational institutions and workplaces, with its ability to support team interactions and sharing of files and information supported by other activities such as management of tasks and scheduling.

#### 3.2.3.1 Critical analysis of the related work

Despite the fact that Teams allows video conferences, real time chats and exchange of documents, it does not have any feature designated for management of a final year project. It does not have a well-defined submission process for the project deliverables. It does not have any mechanism for tracking project milestones, nor does it assign any task to a supervisor or a student for that matter. While it helps in collaborative work, it does not address the education processes in final year projects that FASTGlide is to address.

#### 3.2.3.2 Relationship to the proposed related work

There are effective collaboration tools in Microsoft Teams application but FASTGlide has its focus on automating tasks that are relevant to FYPs. Approval processes and specialized workflows of FASTGlide are meant to enhance the particular project management functions that an academic institution has, for which Teams lacks inbuilt functionality.

### 3.2.4 Basecamp

Basecamp [**?** ] is an application that helps teams in managing projects by breaking down tasks, allocating them to the relevant persons, and monitoring the developments. Such facilities include to-do lists, message boards, schedules and file sharing in order to help all the members of the team to work as one.

#### 3.2.4.1 Critical analysis of the related work

Relatively, Basecamp is very good in generic project management but it does not have particular academic functionalities that are crucial for FYPs such as communication with the professors or even grading. Moreover, it does not have the management and statistics features that are necessary for proper control of educational projects.

#### 3.2.4.2 Relationship to the proposed related work

Basecamp's project management characteristics closely fit systems in the project-tracking category of FASTGlide. Nevertheless, FASTGlide comes with additional academic features such as grading systems, activity tracking systems, and faculty systems that simplify endorsements but Basecamp does not include such modules.

### 3.2.5 Moodle

Moodle [**?** ] is a popular open-source LMS learning platform that is mostly utilized by educational institutions. It helps teachers to build online course, conduct assessments, and monitor the academic performance of students.

#### 3.2.5.1 Critical analysis of the related work

Moodle has robust capacity for online learning and assessment. However, it does not in any way support the work processes of the final year programme (FYP). It is able to run courses, but there are no means of automating the project acceptance or the requisition for feedback which are core to FYPs. Moreover, the elements of the system are quite complicated as well as its use for assisting the specific requirements of a final-year student.

#### 3.2.5.2 Relationship to the proposed related work

Moodle covers a wide spectrum of online education, while FASTGlide strives to enhance only the FYP flow. FASTGlide's benefits are appreciated in the components which are aimed at controlling the processes of FYPs and are not provided fully through a wide range of Moodel's LMS offered solutions.

### 3.2.6 Canvas

Canvas [**?** ], like many others, is yet another learning management system that is used in most educational institutions. It is mostly course management oriented and provides support in organizing assignments, grades and feedback for students.

#### 3.2.6.1 Critical analysis of the related work

Canvas, like many others, is yet another learning management system that is used in most educational institutions. It is mostly course management oriented and provides support in organizing assignments, grades and feedback for students.

#### 3.2.6.2 Relationship to the proposed related work

Canvas is adequate in terms of managing academic activities in general. Canvas does not possess the tools that deal with the intricacies of final year projects, hence FASTGlide is more configurable in that that it provides tools for final year project activities.

### 3.2.7 Sakai

Sakai [**?** ] is an open source learning management system is adopted by many educational institutes for online courses, grading of students, and students' interactions. It accentuates adaptability and personalization to different learning environments.

#### 3.2.7.1 Critical analysis of the related work

Although Sakai's design allows for an easy incorporation of many factors, it may have to be extensively redesiged in order to effectively manage projects such as students FYPs. Basic workflow communications such as project and milestone approvals, FYP faculty to student communications tools, and other controls as required in an FYP, are not provided.

#### 3.2.7.2 Relationship to the proposed related work

FASTGlide offers better tailored academic project management solution that centers around FYP specifications. Sakai's broader course management functionalities present good chances of enhancing the efficacy of managing processes with final year students and their supervising professors but fail to meet the general requirements of the users.

### 3.2.8 Blackboard

Blackboard [**?** ] is one of the most popular learning management systems available as it facilitates the management of courses, assignments, and interaction with students for educational institutions. It enables grading, assessment, and interaction between students and educators.

#### 3.2.8.1 Critical analysis of the related work

Blackboard has provided an all-encompassing academic management suite of tools however this has not been designed for workflows that are strictly project based such as FYPs. Further it lacks automated process to facilitate project completion, monitoring of milestones achieved and performance statistics which are critical management activity of academic projects.

#### 3.2.8.2 Relationship to the proposed related work

The LMS functions of the Blackboard overlap with the functions of many other systems including that of FASTGlide project management system, which is specifically designed for FYPs. FASTGlide provides a solution to the functionality of Blackboard by allowing management of final year projects which have their own complexities in a simplified manner.

### 3.2.9 Edmodo

Edmodo [**?** ] is a learning management system (LMS) that promotes interaction between students and teachers. It is a platform where educators can prepare their assignments, quizzes, and polls, while allowing students to engage, turn in their work, and get resources for study. The instructor can also grade the work and carry out assessments of each student's performance based on predefined metrics.

#### 3.2.9.1 Critical analysis of the related work

Although Edmodo is a satisfactory general purpose classroom engagement tool that incorporates such features as submitting assignments and communicating with learners, it does not have the necessary features to manage final year projects (FYPs) for colleges and universities. It does not allow for tracking of milestones, facilitate an approval workflow process automatically, or even provide any form of analytics as far as academic projects focused on FYP's are concerned. More attention is paid on operating within manageable units of a day or two in class than organizing and coordinating work over a couple of weeks or months.

#### 3.2.9.2 Relationship to the proposed related work

Edmodo may allow submission of assignments and interaction with the students but FASTGlide aims to make the FYP process quicker and more efficient. There are specific offerings tailored for FYP management like tracking of proposal submissions, controlling of milestones and student's supervisors which makes this system better fitted for a very particular purpose unlike Edomo where the focus is on management of classroom activities.

### 3.2.10 Brightspace by D2L

Brightspace [**?** ] is a comprehensive LMS for higher learning institution with content management, quizzes, grading, and analytics, among others. It endorses blended learning and has course management, learning progress, and student interaction tools.

#### 3.2.10.1 Critical analysis of the related work

Brightspace has a massive potential in terms of coursework and student performance management but focuses mainly on general systems and does not suit the precise management of the FYP. Further, it does not have features such as automated systems for FYP approvals, timelines for the completion of project phases, and various supervisor-to-student interactions focused on the academic nature of projects. Also, although it is true that Brightspace does have analytic aspects with regards to course performance, it

does not have any of the pertinent FYP analytics.

#### 3.2.10.2 Relationship to the proposed related work

While Brightspace has got all the essential components of an LMS which enhances academic management systems, FASTGlide focuses solely on FYPs. FYP-specific project submission processes for example, supervisor assessment and milestone tracking are features that are unavailable in Brightspace but are present in the software. The design of FASTGlide captures this unique need of final year students and their supervisors hence it is more preferred and focused on the management of final year projects.

## 3.3 Literature Review Summary Table

**Table 3.1: Summary of Related Work**

| Application | Features | Relevance | Limitations |
|---|---|---|---|
| Google Classroom [? ] | User-friendly interface for managing coursework, assignments, and class communications | Streamlines the process of class management, enabling efficient communication and assignment distribution among students and instructors | Lacks specialized tools for tracking progress and managing the complexities of Final Year Projects (FYP), which may hinder effective project management |
| Gradescope [? ] | AI-assisted grading | Streamlines grading, allowing instructors to evaluate large numbers of assignments efficiently | Focuses on grading, not project management or collaboration needed for FYPs. |
| Microsoft Teams [? ] | Comprehensive collaboration platform | Enhances communication and teamwork among students and faculty | Lacks FYP-specific features like milestone tracking and project oversight. |
| Basecamp [? ] | Project management tool | Provides a structured approach to managing tasks and projects | Does not cater to the academic environment, lacking educational compliance features. |
| Moodle [? ] | Open-source learning management system | Facilitates course creation and student tracking | Lacks specific FYP workflows and automated processes for project submissions. |
| Canvas [? ] | User-friendly course management tools | Supports efficient management of assignments and grading | Does not specifically address the unique requirements of FYPs or their management. |
| Sakai [? ] | Flexible course management platform | Customizable for different academic needs | Requires significant setup time for customization to meet FYP requirements. |

*Continued on the next page*

| Application | Features | Relevance | Limitations |
|---|---|---|---|
| Blackboard [**?** ] | Comprehensive academic management tools | Streamlines course administration | Lacks dedicated project management features for Final Year Projects. |
| Edmodo [**?** ] | Classroom management tools | Enhances classroom interaction and communication | Focuses on day-to-day activities rather than long-term project management. |
| Brightspace [**?** ] | Advanced learning management system | Offers robust analytics and course management features | Lacks dedicated FYP management capabilities and detailed supervisor-student interaction tools. |

## 3.4 Conclusion

The previous chapters examine the usage of diverse computer software within the academic context, concentrating mainly on the organization of the coursework, the interaction between students and instructors, as well as the evaluation of the students' achievements. Google Classroom [**?** ] and Microsoft Teams [**?** ] provide users with features that facilitate the teaching and the collaborative aspects of the courses, however, they usually do not include the tools that are needed for the successful FYP management. There are such applications as Gradescope [**?** ] or Moodle [**?** ] that simplify the grading process and organization of the course but do not help with the specific aspects of the FYP workflow. Furthermore, solutions such as Blackboard [**?** ] and Canvas [**?** ] assist in the promotion of academic activities coordination however, they lack project control and task scheduling features that are fundamental to FYPs. From this analysis, it is evident that there is a requirement for such a solution, which covers the limitations of existing ones at the same time improving the experience of all participants during the FYP process for instance students, supervisors and coordinators.

# Chapter 4 Software Requirement Specifications

This chapter highlights important features of the projects. It also includes functional and non- functional requirement of the project, defines database design and risk analysis involves in the project.

## 4.1 List of Features

The system will support the following core features:

- Allow students to search for FYP group members.
- Notify students about FYP group requests.
- Enable students to search for FYP supervisors.
- Facilitate online FYP registration.
- Enable communication between students and their group members, as well as supervisors.
- Allow students to schedule meetings (online or physical) with their supervisor.
- Enable students to submit their final deliverable for format verification.
- Notify supervisors of FYP registration requests.
- Allow supervisors to post FYP project ideas.
- Notify supervisors of meeting requests.
- Facilitate communication between supervisors and FYP groups.
- Allow Coordinators to view the deliverables submissions.

## 4.2 Functional Requirements

The functional requirements fully describe the external behavior of the system. Each functionality is identified and briefly described below, along with the respective users.

### 4.2.1 Functional Requirements for Students

The system will allow students to:

- **Log in to the FYP portal**: Students will be able to securely log in to access the portal.
- **Complete their profile with relevant information**: Students can update their profile with essential details, including academic and personal information.

- **View project ideas posted by supervisors**: Students can browse and view FYP project ideas posted by different supervisors.
- **Connect with other students to form FYP groups**: The system allows students to find and connect with peers to form FYP groups.
- **Reach out to supervisors regarding FYP projects**: Students can communicate with supervisors to discuss project ideas and potential collaboration.
- **Register their FYP project**: Students can officially register their FYP project through the portal.
- **View FYP registration details**: Registered students can view the details of their FYP registration.
- **Communicate with their FYP group members**: A communication feature allows students to stay connected with their group members.
- **Schedule meetings with their supervisor**: The system enables students to schedule either online or physical meetings with their supervisor.
- **Submit the FYP deliverables**: Students can upload their final FYP deliverables for review and approval.

### 4.2.2 Functional Requirements for Supervisors

The system will allow supervisors to:

- **Post FYP project ideas**: Supervisors can post FYP project ideas for students to review and consider.
- **View student requests to join an FYP project**: Supervisors can see requests from students interested in joining their project.
- **Accept or reject student requests**: Supervisors have the ability to approve or decline student requests to join their project.
- **Register FYP groups**: Supervisors can formally register FYP groups within the system.
- **Communicate with the members of the FYP groups**: Supervisors can directly communicate with the FYP group members.

### 4.2.3 Functional Requirements for Coordinators

The system will allow coordinators to:

- **View the deliverables submitted by the student**: Coordinators can review the final deliverables

submitted by students.

- **Set the deadlines for the deliverables**: Coordinators can establish and manage deadlines for the submission of FYP deliverables.
- **Post any announcement for the students**: Coordinators can publish announcements and updates for students through the system.

### 4.2.4 Functional Requirements of the System

The system will:

- **Recommend potential group members to students for FYP**: The system will suggest possible group members for students based on relevant criteria.
- **Suggest suitable supervisors for students**: Based on project interests, the system will recommend appropriate supervisors to students.
- **Handle FYP registration**: The system will manage the registration process for FYP projects, ensuring accurate tracking of project registrations.
- **Notify both students and supervisors about updates and requests**: The system will send notifications to keep both students and supervisors updated on requests and changes.
- **Facilitate meeting scheduling between students and supervisors**: The system will assist in scheduling meetings between students and their supervisors.
- **Verify the formatting of the submitted final deliverables**: The system will check the format of submitted deliverables to ensure they meet the required standards.

## 4.3 Non-Functional Requirements

This section describes the non-functional requirements, including performance, reliability, usability, security, scalability, maintainability, and portability of the system.

### 4.3.1 Performance

- The system should allow users to access the FYP portal and load pages within 5-10 seconds under normal conditions.
- It should be able to handle concurrent usage by a realistic number of users (students, supervisors) without noticeable lag or degradation in performance.

### 4.3.2 Reliability

- The system should aim for 99% uptime, ensuring high availability throughout the FYP submission period.
- Backup and recovery mechanisms should be in place to minimize data loss in case of system failure.

### 4.3.3 Usability

- The system will provide an intuitive, user-friendly interface that allows users to navigate and use all core functionalities (FYP registration, group formation, communication, etc.) without extensive training.
- The interface should be self-explanatory, enabling users to complete tasks in a few minutes.
- It should also be accessible on a wide range of devices, such as laptops, tablets, and smartphones.

### 4.3.4 Security

- User data such as profile information, FYP details, and communications must be securely stored, ensuring confidentiality and privacy.
- The system should implement secure authentication and authorization mechanisms to prevent unauthorized access to sensitive information.
- All data exchanges, especially during communication and registration, should be encrypted (e.g., using SSL/TLS).

### 4.3.5 Scalability

- The system should be scalable to support an increasing number of students and supervisors over time without performance degradation.
- It should be easy to upgrade or modify features to accommodate future requirements, such as adding new types of FYP projects or expanding to other departments.

### 4.3.6 Maintainability

- The system will follow modular design principles, making it easier to maintain and update.
- The code should be well-documented to allow future developers to easily understand and make modifications as needed.

### 4.3.7 Portability

- The system should be designed to run on various operating systems and browsers that support JavaScript and modern web technologies.
- It should be deployable on different environments (on-premise or cloud-based) without significant changes.

## 4.4 Assumptions

The following assumptions have been made for the system specification:

- The end users have access to a browser with JavaScript compatibility installed on their devices.
- A stable internet connection is available for all users to interact with the system.
- Users are expected to have a basic understanding of common website functionalities, such as navigation and form submission.

## 4.5 Use Cases

This section lists relevant use cases that represent central functionalities of our system encompassing all stakeholders.

**Table 4.1: Login Process Details**

| | |
|---|---|
| **Name** | Login |
| **Actors** | Students, Teachers |
| **Summary** | The user shall provide their email and password on the login form, and after successful verification, redirect the user to the home page. |
| **Pre-Conditions** | The user must be in the database records, either added by any of the authorized users or added manually by a developer.<br>The user must not be logged in. |
| **Post-Conditions** | The user's session is successfully established and shall be redirected to the home page. |
| **Special Requirements** | None |

| **Basic Flow** | | | |
|---|---|---|---|
| **Actor Action** | | **System Response** | |
| 1 | The user opens the login page. | 2 | The login page is displayed asking for email and password. |
| 2 | The user enters valid email and password. | 4 | The system verifies the email and password, establishes a session for the user, and redirects the user to the home page. |
| **Alternative Flow** | | | |
| 3 | The user enters invalid email or password. | 4-A | The system responds with an error message: Incorrect email or password entered. |

**Table 4.2: Logout Process Details**

| | | | |
|---|---|---|---|
| **Name** | Logout | | |
| **Actors** | Students, Teachers | | |
| **Summary** | The user shall click on the ”Logout” button and will be redirected to the login page. | | |
| **Pre-Conditions** | The user must be logged in. | | |
| **Post-Conditions** | The user’s session is successfully ended and the user is redirected to the login page. | | |
| **Special Requirements** | None | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The user clicks on the ”Logout” button. | 2 | The system terminates the user’s session and redirects the user to the login page. |
| **Alternate Flow** | | | |
| No Alternate Flow | | | |

**Table 4.3: Add FYP Post Description Process**

| | | | |
|---|---|---|---|
| **Name** | Add FYP Post Description | | |
| **Actors** | Teachers | | |
| **Summary** | The user shall click on the ”Add Post” button and a pop-up will appear asking for the details of the post. | | |
| **Pre-Conditions** | The user must be logged in. | | |
| **Post-Conditions** | The new FYP post description is stored in the database. | | |
| **Special Requirements** | None | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The user clicks on the ”Add Post” button. | 2 | The ”Add New Post Description” popup is displayed. |
| 2 | The user fills out the necessary fields. | 3 | The system validates the entered data. |
| 3 | The user clicks the ”Post” button. | 4 | The system stores the new FYP post description in the database and confirms the successful addition. |

| | Actor Action | | System Response |
|---|---|---|---|
| **Alternate Flow** | | | |
| 2a | The user enters invalid data. | 4-A | The system responds with an error message: Incorrect data entered. |

**Table 4.4: Analyzing Process**

| | | | |
|---|---|---|---|
| **Name** | Analyzing Process | | |
| **Actors** | Students | | |
| **Summary** | The User selects the document for running the analyzing process. | | |
| **Pre-Conditions** | The user must be already logged in. | | |
| **Post-Conditions** | The uploaded document feedback is displayed on the screen | | |
| **Special Requirements** | None | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The user navigates to ”Deliverable checkup” section. | 2 | The deliverable checkup section is displayed |
| 2 | The user clicks the Upload document button | 3 | The system opens the tab to select documents from device |
| 3 | The user selects the document | 4 | The system uploads the document and stores them temporarily for screening on server side and confirms the successful addition |
| 3 | The User presses the “Check” button. | 4 | The System analyzes the uploaded document on server side and shows the feedback of that document on the screen |
| **Alternative Flow** | | | |
| 3a | The user enters invalid data. | 4-A | The system responds with an error message. |

**Table 4.5: Upload FYP Deliverables Process**

| | | | |
|---|---|---|---|
| **Name** | Upload FYP Deliverables | | |
| **Actors** | Students | | |
| **Summary** | The user uploads FYP Deliverable | | |
| **Pre-Conditions** | The user must be already logged in. | | |
| **Post-Conditions** | The uploaded document is temporarily stored in the application | | |
| **Special Requirements** | None | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The user navigates to ”Deliverable checkup” section. | 2 | The deliverable checkup section is displayed |
| 2 | The user clicks the Upload document button | 3 | The system opens the tab to select documents from device |
| 3 | The user selects the document | 4 | The system uploads the document and stores them temporarily for screening on server side and confirms the successful addition |
| **Alternative Flow** | | | |
| 3a | The user enters invalid data. | 4-A | The system responds with an error message. |

**Table 4.6: Delete FYP Post Description Process**

| | | | |
|---|---|---|---|
| **Name** | Delete FYP Post Description | | |
| **Actors** | Teachers | | |
| **Summary** | The user shall be able to delete an existing post from the database. | | |
| **Pre-Conditions** | The user must be already logged in. | | |
| **Post-Conditions** | The system deletes the specified FYP post from the database. | | |
| **Special Requirements** | None | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The user navigates to the posts section. | 2 | The system displays the user’s posts. |

| | Actor Action | | System Response |
|---|---|---|---|
| 2 | The user clicks the delete icon for a specific post. | 3 | The system displays a confirmation pop-up. |
| 3 | The user clicks the “Confirm” button. | 4 | The system deletes the FYP post from the database. |

**Table 4.7: Edit FYP Post Description Process**

| **Name** | Edit FYP Post Description |
|---|---|
| **Actors** | Teachers |
| **Summary** | The user shall be able to edit an existing FYP post in the system. |
| **Pre-Conditions** | The user must be already logged in. |
| **Post-Conditions** | The system stores the updated FYP post description in the database. |
| **Special Requirements** | None |

| **Basic Flow** | | | |
|---|---|---|---|
| | **Actor Action** | | **System Response** |
| 1 | The user navigates to the Posts section. | 2 | The system displays the user’s posts. |
| 2 | The user selects the edit icon for a specific post. | 3 | The system displays the current post description. |
| 3 | The user edits the necessary fields. | 4 | The system validates the entered data. |
| 4 | The user clicks the ”Edit” button. | 5 | The system stores the updated post in the database and confirms the update. |
| **Alternative Flow** | | | |
| 3a | The user enters invalid data. | 4-A | The system responds with an error message. |

**Table 4.8: Meeting Scheduling**

| | | | |
|---|---|---|---|
| **Name** | Meeting Scheduling | | |
| **Actors** | Students, Supervisors | | |
| **Summary** | The user shall be able to schedule meetings with supervisors or team members. | | |
| **Pre-Conditions** | The user must be logged in and have access to the meeting scheduling tool. | | |
| **Post-Conditions** | The meeting is successfully scheduled and notifications are sent to all participants. | | |
| **Special Requirements** | The meeting time must not conflict with other scheduled events for participants. | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The user navigates to the meeting scheduling page. | 2 | The system displays the meeting scheduling interface. |
| 2 | The user selects a date and time for the meeting. | 3 | The system checks for any conflicts with existing schedules. |
| 3 | The user adds participants to the meeting. | 4 | The system sends invitations to the selected participants. |
| 4 | The user confirms the meeting details and schedules the meeting. | 5 | The system finalizes the meeting and sends notifications to all participants. |
| **Alternative Flow** | | | |
| 2a | The user selects a time slot that conflicts with another meeting. | 3-A | The system displays an error message: ”Selected time slot is unavailable, please choose another.” |
| 3a | The user fails to add participants to the meeting. | 4-A | The system prompts the user to add at least one participant before scheduling the meeting. |

**Table 4.9: Supervisor Selection by Post**

| | | | |
|---|---|---|---|
| **Name** | Supervisor Selection by Post | | |
| **Actors** | Students, Supervisors | | |
| **Summary** | The User selects a supervisor by posting a request. | | |
| **Pre-Conditions** | The user must be logged in to the system, and the supervisor's post must be available. | | |
| **Post-Conditions** | The selected supervisor is assigned to the user. | | |
| **Special Requirements** | The system must ensure the supervisor is available for new students. | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The user navigates to the supervisor post section. | 2 | The system displays the list of available supervisors. |
| 2 | The user selects a supervisor from the list. | 3 | The system confirms the selection and updates the supervisor's availability. |
| 3 | The user posts a request to the selected supervisor. | 4 | The system sends a notification to the supervisor about the request. |
| **Alternative Flow** | | | |
| 1a | If no supervisors are available in the list. | 2-A | The system informs the user and suggests checking back later. |
| 2a | The user tries to select a supervisor who is no longer available. | 3-A | The system displays an error message. |

**Table 4.10: FYP Group Selection**

| | | | |
|---|---|---|---|
| **Name** | FYP Group Selection | | |
| **Actors** | Students | | |
| **Summary** | The user shall be able to select a group for the FYP project. | | |
| **Pre-Conditions** | The user must be logged in and have access to group selection. | | |
| **Post-Conditions** | The selected group is confirmed and members are notified. | | |
| **Special Requirements** | All group members must accept the selection to proceed. | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The user navigates to the FYP group selection page. | 2 | The system displays available groups for selection. |
| 2 | The user selects a group for the FYP project. | 3 | The system validates the selection and checks group availability. |
| 3 | The user confirms the group selection. | 4 | The system sends notifications to all group members for approval. |
| **Alternative Flow** | | | |
| 2a | The user selects an already full group. | 3-A | The system displays an error message |
| 3a | One or more members reject the group selection. | 4-A | The system informs the user and requests a new group selection. |

**Table 4.11: FYP Registration**

| | | | |
|---|---|---|---|
| **Name** | FYP Registration | | |
| **Actors** | Students, Supervisors | | |
| **Summary** | The user shall be able to register for the FYP project. | | |
| **Pre-Conditions** | The user must be logged in and have the necessary prerequisites for registration. | | |
| **Post-Conditions** | The user is successfully registered for the FYP project. | | |
| **Special Requirements** | The registration must be approved by the supervisor. | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The user navigates to the FYP registration page. | 2 | The system displays the registration form. |

| | Actor Action | | System Response |
|---|---|---|---|
| 2 | The user fills in the required details for FYP registration. | 3 | The system validates the entered information. |
| 3 | The user submits the registration form. | 4 | The system confirms the submission and sends the details to the supervisor for approval. |
| **Alternative Flow** | | | |
| 2a | The user provides incomplete or incorrect information. | 3-A | The system displays an error message requesting corrections to the form. |
| 3a | The supervisor rejects the registration request. | 4-A | The system notifies the user about the rejection and provides feedback. |

**Table 4.12: Communication**

| **Name** | Communication | | |
|---|---|---|---|
| **Actors** | Students, Supervisors, Admins | | |
| **Summary** | The user shall be able to communicate with supervisors and other team members. | | |
| **Pre-Conditions** | The user must be logged in to the system and have access to communication tools. | | |
| **Post-Conditions** | The messages are successfully delivered to the intended recipients. | | |
| **Special Requirements** | The system must ensure message security and privacy. | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The user navigates to the communication section. | 2 | The system displays the messaging interface. |
| 2 | The user selects a contact or group to communicate with. | 3 | The system opens the chat window for the selected contact or group. |
| 3 | The user types a message and clicks the send button. | 4 | The system sends the message and displays it in the chat window. |
| **Alternative Flow** | | | |
| 2a | The user tries to send a message without selecting a contact. | 3-A | The system displays an error message: ”Please select a contact to send a message.” |
| 3a | The system fails to deliver the message due to network issues. | 4-A | The system notifies the user that the message could not be delivered and suggests retrying. |

**Table 4.13: FYP Group Creation**

| | | | |
|---|---|---|---|
| **Name** | FYP Group Creation | | |
| **Actors** | Students | | |
| **Summary** | The User shall be able to create groups for FYPs with other students either sending them request | | |
| **Pre-Conditions** | The user must be already logged in. | | |
| **Post-Conditions** | The system sends the request to other students along with details and notifies them. | | |
| **Special Requirements** | None | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The user navigates to “Search” section. | 2 | The Search section opens. |
| 2 | The User searches the student either by their name or by interest matched. | 3 | The System displays the relevant areas for searching. |
| 3 | The User selects the student. | 4 | A popup appears requiring details to be filled required for sending request to student. |
| 3 | The User fills the required details. | 4 | The System validates the entered details. |
| 3 | The User clicks send request button. | 4 | The System stores the request details in the system and notifies the student about the request with details. |
| **Alternative Flow** | | | |
| 3a | The user enters invalid data. | 4-A | The system responds with an error message. |

**Table 4.14: Deadlines Setup**

| | | | |
|---|---|---|---|
| **Name** | Deadlines Setup | | |
| **Actors** | Students, Supervisors | | |
| **Summary** | The user shall be able to set deadlines for project submissions. | | |
| **Pre-Conditions** | The user must be logged in and have the necessary permissions to set deadlines. | | |
| **Post-Conditions** | The deadlines are saved, and notifications are sent to the relevant users. | | |
| **Special Requirements** | The system must allow setting of multiple deadlines for different deliverables. | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The user navigates to the deadlines setup page. | 2 | The system displays the current deadlines and options to add new ones. |
| 2 | The user enters the details for the new deadline. | 3 | The system validates the entered information for correctness. |
| 3 | The user saves the new deadline. | 4 | The system updates the deadline list and confirms the changes to the user. |
| **Alternative Flow** | | | |
| 2a | The user enters an invalid date or time. | 3-A | The system displays an error message: ”Please enter a valid date and time.” |
| 3a | The user tries to set a deadline that conflicts with existing deadlines. | 4-A | The system notifies the user of the conflict and prompts for a new deadline. |

**Table 4.15: Deliverables Submission**

| | | | |
|---|---|---|---|
| **Name** | Deliverables Submission | | |
| **Actors** | Students, Supervisors | | |
| **Summary** | The user shall be able to submit deliverables for their FYP project. | | |
| **Pre-Conditions** | The user must be logged in and have the deliverables ready for submission. | | |
| **Post-Conditions** | The deliverables are successfully submitted and confirmation is sent to the user. | | |
| **Special Requirements** | The system must support multiple file formats for deliverable submissions. | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The user navigates to the deliverables submission page. | 2 | The system displays the submission form. |

| Actor Action | | System Response | |
|---|---|---|---|
| 2 | The user uploads the required deliverable files. | 3 | The system validates the file format and size. |
| 3 | The user submits the deliverables. | 4 | The system stores the files and sends a confirmation to the user. |
| **Alternative Flow** | | | |
| 2a | The user uploads a file in an unsupported format. | 3-A | The system displays an error message: ”Unsupported file format, please upload a valid file.” |
| 3a | The file size exceeds the maximum allowed limit. | 4-A | The system prompts the user to reduce the file size before submitting. |

**Table 4.16: Edit FYP Registration**

| **Name** | Edit FYP Registration |
|---|---|
| **Actors** | Students, Supervisors |
| **Summary** | The user shall be able to edit the FYP registration details. |
| **Pre-Conditions** | The user must be logged in and have an existing FYP registration. |
| **Post-Conditions** | The FYP registration details are updated successfully. |
| **Special Requirements** | The system must ensure that the updates do not conflict with existing submissions. |

| **Basic Flow** | | | |
|---|---|---|---|
| **Actor Action** | | **System Response** | |
| 1 | The user navigates to the FYP registration edit page. | 2 | The system displays the current registration details for editing. |
| 2 | The user modifies the required fields. | 3 | The system validates the updated information for correctness. |
| 3 | The user submits the edited registration details. | 4 | The system updates the registration and confirms the changes to the user. |
| **Alternative Flow** | | | |
| 2a | The user enters invalid data while editing. | 3-A | The system displays an error message: ”Please correct the highlighted fields.” |
| 3a | The user tries to submit changes that conflict with existing data. | 4-A | The system notifies the user of the conflict and prompts for a resolution. |

**Table 4.17: Display Deliverables**

| | |
|---|---|
| **Name** | Display Deliverables |
| **Actors** | Students, Supervisors |
| **Summary** | The user shall be able to view the deliverables submitted for their FYP project. |
| **Pre-Conditions** | The user must be logged in to the system. |
| **Post-Conditions** | The system displays the list of submitted deliverables to the user. |
| **Special Requirements** | The deliverables must be displayed in an organized manner, including status updates. |

| **Basic Flow** | | | |
|---|---|---|---|
| **Actor Action** | | **System Response** | |
| 1 | The user navigates to the deliverables section. | 2 | The system displays the list of deliverables submitted by the user. |
| 2 | The user selects a deliverable to view details. | 3 | The system shows detailed information about the selected deliverable, including submission date and feedback. |
| **Alternative Flow** | | | |
| 1a | The user navigates to the wrong section or encounters an error. | 2-A | The system displays an error message: ”Please select the correct section to view deliverables.” |
| 2a | The system fails to retrieve the deliverables due to a network issue. | 3-A | The system notifies the user: ”Unable to retrieve deliverables at this time, please try again later.” |

## 4.6 Hardware and Software Requirements

Hardware and Software requirements of the project are:

### 4.6.1 Hardware Requirements

The following are the hardware requirements for the project:

- Desktop PC/Laptop
- Internet connection

### 4.6.2 Software Requirements

The following are the software requirements for the project:

- VS Code
- React JS
- MongoDB
- Node JS

## 4.7 Graphical User Interface

The following are the GUI designs of the FASTGlide system.

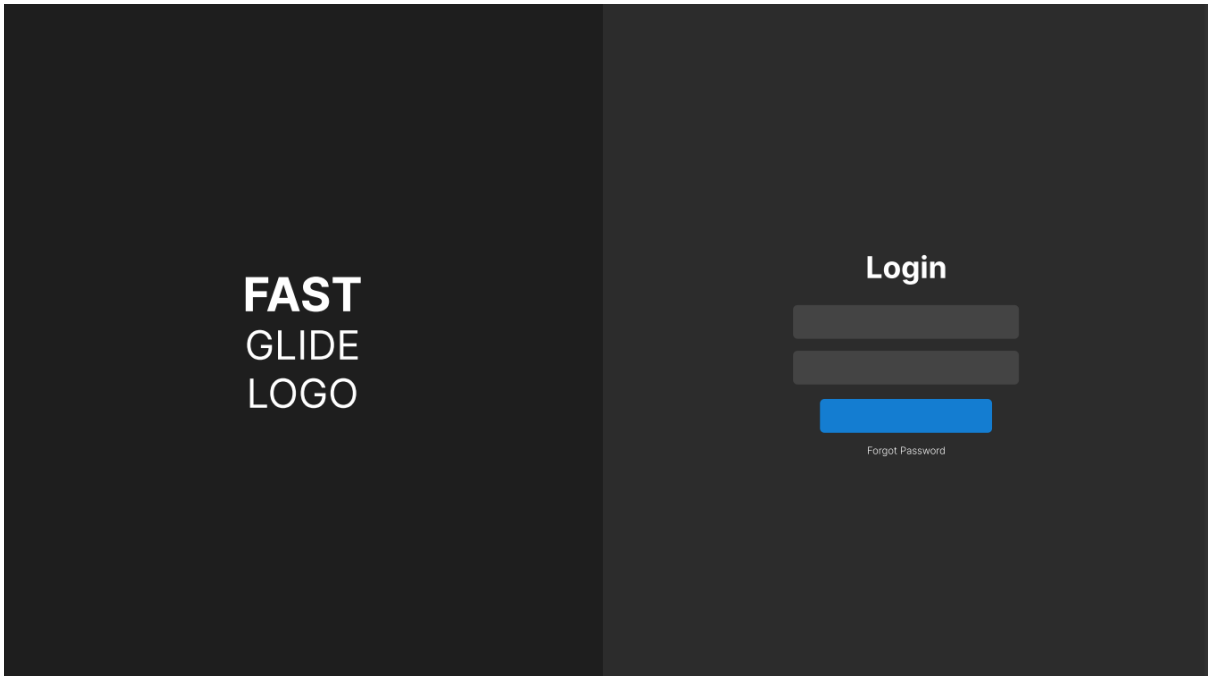


**Figure 4.1: Login Page Design**

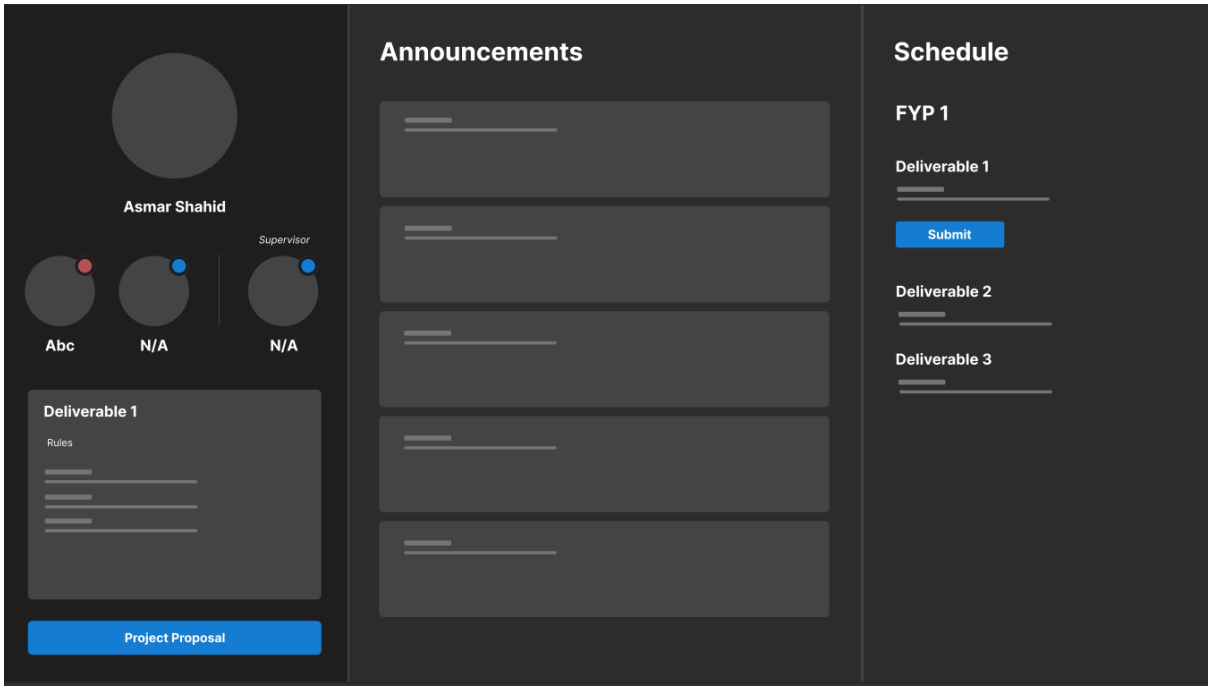


**Figure 4.2: Dashboard Interface**

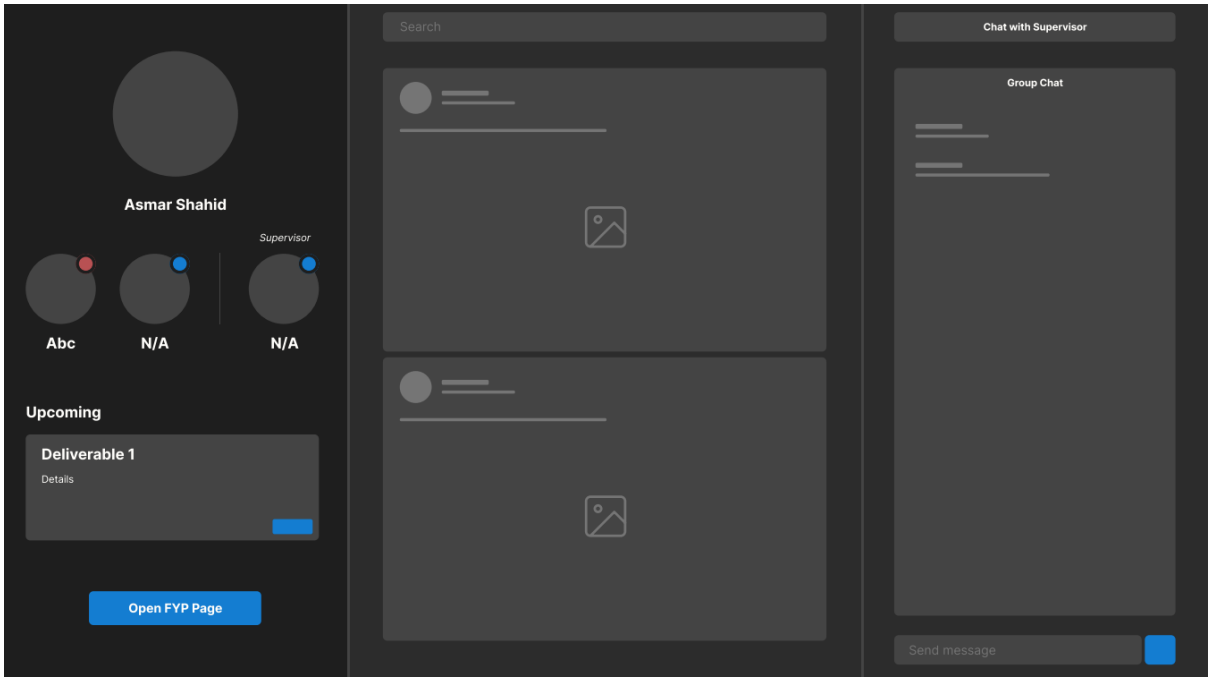


**Figure 4.3: Post Creation Page**

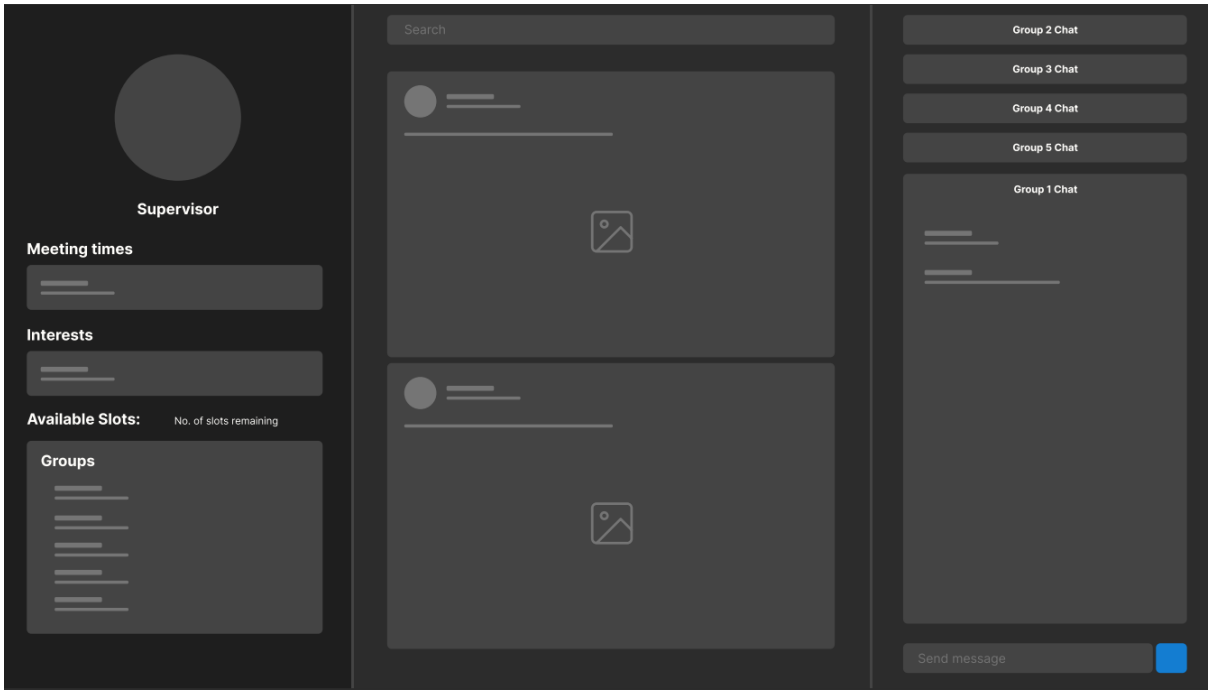


**Figure 4.4: FYP Group Creation Interface**

## 4.8 Database Design

### 4.8.1 ER Diagram

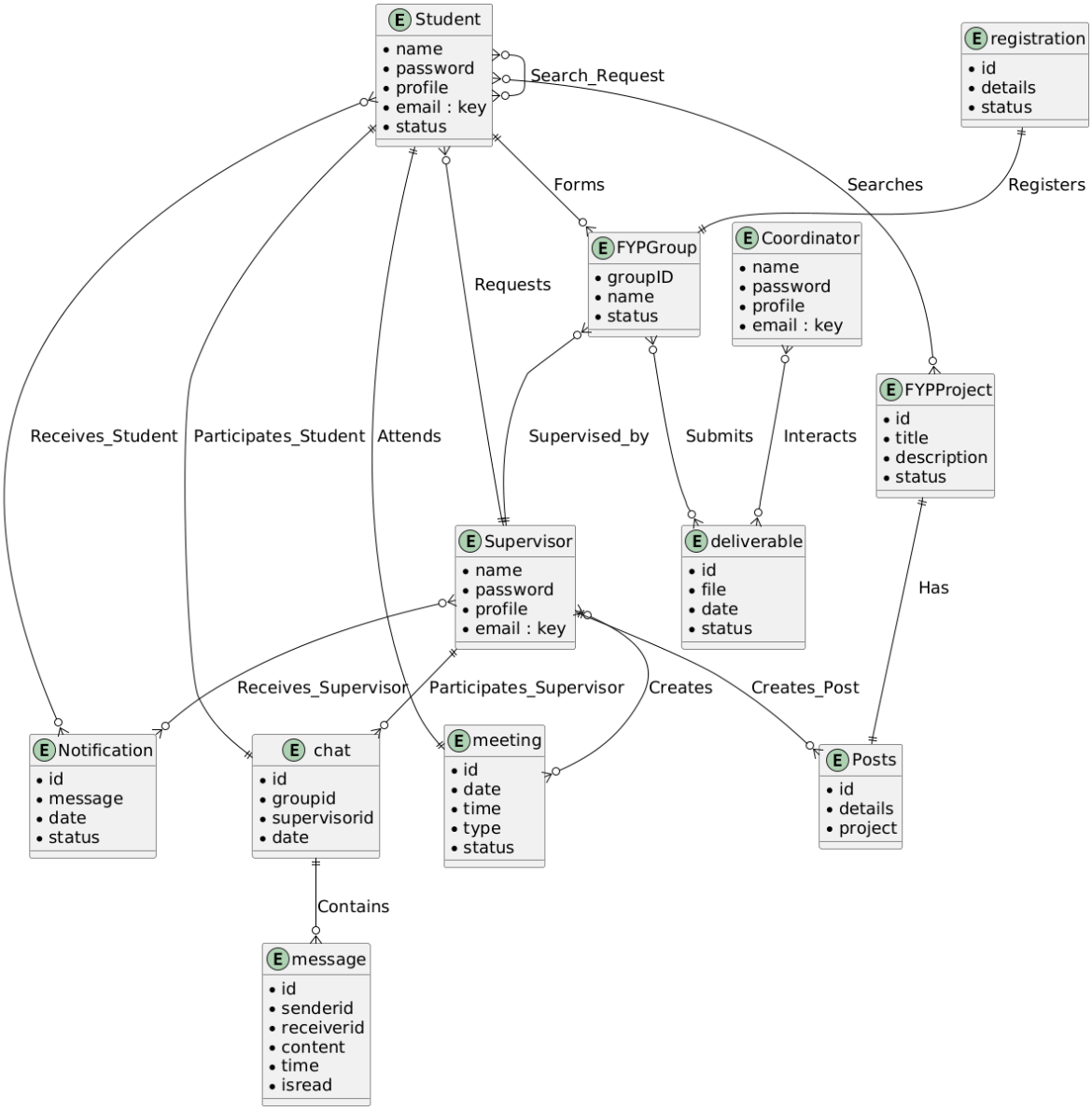


**Figure 4.5: Entity Relationship Diagram**

### 4.8.2 Data Dictionary

**Table 4.18: Data Dictionary**

| Entity | Attribute | Data Type | Nullable | Relation To | Relation Type | Description |
|---|---|---|---|---|---|---|
| Student | ID | String | No | Supervisor, FYP Group, FYP Project, Meeting, Chat, Notification | Many to One, Many to Many, Many to One, Many to One, Many to Many | Primary key |
| | Name | String | No | - | - | Full Name |
| | Email | String | No | - | - | Email ID |
| | Status | Boolean | No | - | - | Status |
| | Profile | Object | No | - | - | Profile Info |
| | Password | String | No | - | - | Password |
| Supervisor | ID | String | No | Student, FYP Group, Meeting, Notification, Chat, Posts | One to Many, One to Many, One to Many, Many to Many, One to Many, One to Many | Primary key |
| | Name | String | No | - | - | Full Name |
| | Email | String | No | - | - | Email ID |
| | Password | String | No | - | - | Password |
| | Profile | Object | No | - | - | Profile Info |
| | Slots | int | Yes | - | - | Slots |
| Coordinator | ID | String | No | Deliverable | Many to Many | Primary key |
| | Name | String | No | - | - | Full Name |
| | Password | String | No | - | - | Password |
| | Email | String | No | - | - | Email ID |

*Continued on the next page*

| Entity | Attribute | Data Type | Nullable | Relation To | Relation Type | Description |
|---|---|---|---|---|---|---|
| FYP Group | ID | String | No | Student, Supervisor, Registration, Deliverable | One to Many, Many to One, One to One, One to Many | Primary key |
| | Name | String | No | - | - | Group Name |
| | Status | Boolean | No | - | - | Status |
| FYP Project | ID | String | No | Student, Posts | Many to Many, One to One | Primary key |
| | Title | String | No | - | - | Project Title |
| | Description | Object | No | - | - | Project Description |
| | Status | Boolean | No | - | - | Status |
| Registration | ID | String | No | FYP Group | One to One | Primary key |
| | Details | Object | No | - | - | Registration Details |
| | Status | Boolean | No | - | - | Registration Status |
| Deliverable | ID | String | No | FYP Group, Coordinator | Many to One, Many to Many | Primary key |
| | File | Pdf | No | - | - | Deliverable File |
| | Date | Date | No | - | - | Submission Date |
| | Status | Boolean | No | - | - | Deliverable Status |
| Chat | ID | String | No | Student, Supervisor, Message | One to Many, Many to One, One to Many | Primary key |
| | Group ID | Object | No | - | - | Group ID |
| | Supervisor ID | Object | No | - | - | Supervisor ID |
| | Date | Date | Yes | - | - | Chat Date |

*Continued on the next page*

| Entity | Attribute | Data Type | Nullable | Relation To | Relation Type | Description |
|---|---|---|---|---|---|---|
| Message | ID | String | No | Chat, Student | Many to One, Many to One | Primary key |
| | Sender ID | Object | No | - | - | Sender ID |
| | Receiver ID | Object | No | - | - | Receiver ID |
| | Content | String | No | - | - | Message Content |
| | Time | Time | No | - | - | Message Time |
| | Read Status | Boolean | No | - | - | Read Status |
| Notification | ID | String | No | Student, Supervisor | Many to Many, Many to Many | Primary key |
| | Content | String | No | - | - | Notification Content |
| | Date | Date | No | - | - | Notification Date |
| | Status | Boolean | No | - | - | Notification Read Status |
| Posts | ID | String | No | Supervisor, FYP Project | Many to One, One to One | Primary key |
| | Details | String | No | - | - | Post Text |
| | Project | Object | Yes | - | - | Linked Project |
| Meeting | ID | String | No | Student, Supervisor | One to Many, Many to One | Primary key |
| | Time | Time | No | - | - | Meeting Time |
| | Date | Date | No | - | - | Meeting Date |
| | Type | String | Yes | - | - | Meeting Type |

*Continued on the next page*

| Entity | Attribute | Data Type | Nullable | Relation To | Relation Type | Description |
|---|---|---|---|---|---|---|
| | Status | Boolean | No | - | - | Meeting Status |

## 4.9 Risk Analysis

Risk analysis is an important part of the project to identify factors or risks that may adversely affect the design, implementation, and success of the FASTGlide web application. Below details the main risks that have been pinpointed in the project, their potential effects, and ways to control them:

### 4.9.1 Data Security Risks

- The web application contains sensitive data such as students' details, supervisors' information, and even the final project submissions. Any compromise to this data could lead to information abuse and violations of rights.
- A data breach could also lead to the loss of users' confidence and result in administrative and legal consequences for the university.

**Mitigation:**

- Use high-level encryption programming for data transmission processes (HTTPS) and incorporate JSON Web Tokens (JWT) for secure login.
- Implement Role-Based Access Control (RBAC) to restrict access based on user categories (e.g., Students, Supervisors, Coordinators).

### 4.9.2 System Downtime or Performance Issues

- As the number of users increases, especially during peak times (e.g., final submission deadlines), the system may experience slowdowns or downtime, affecting usability.
- If the system goes down during critical times, such as the final project submission period, it could lead to frustration and missed deadlines.

**Mitigation:**

- Use performance optimization techniques, such as caching and asynchronous processing, to handle high loads.

### 4.9.3 File Format and Submission Issues

- Students may occasionally fail to submit valid files, or the files may contain bugs that would impair the web application's core function of validating the format of LaTeX project deliverables.
- Failure to correctly validate the format may result in confusion among stakeholders and delays in project approval.

**Mitigation:**

- Conduct a thorough testing process of the LaTeX format validation component to ensure the system can effectively detect and flag invalid files.
- Educate students on the requirements for presenting LaTeX deliverables by providing guidance on what is expected and how to format their outputs.
- Offer a "test submission" option for students to verify the formats and composition of their files before final submission.

### 4.9.4 Integration with University Systems

- Since FASTGlide lacks the ability to query the university's internal databases and systems, issues may arise during integration into the existing network infrastructure, particularly for authentication or communication.
- The dependency on university email for authentication may pose risks if seamless integration is not achieved, or if there are prolonged downtimes of email systems.

**Mitigation:**

- Implement a robust and independent authentication method using university email addresses, such as through OAuth.
- Design the system with contingency measures to address scenarios where email verification fails or login processes encounter issues.

## 4.10 Conclusion

The SRS chapter has discussed the main features and the other defining characteristics of the FASTGlide system, which facilitates the Final Year Project (FYP) for students, supervisors and coordinators. The functional requirements specify that the users will be able to use the modules of the system starting from the registration of FYP, completing the communication and submitting the deliverables which

is comprehensive and user friendly. Non-functional requirements, such as performance, scalability, security, and maintainability envision that the system will continue to function well as well as work as intended in the beginning even as time progresses and the number of users increases. All in all, the thorough requirements in this chapter form the basic structure that helps attain the development and the operationalization of the FASTGlide system.

# Chapter 5 High-Level and Low-Level Design

This chapter discusses both the high-level and low-level design of FASTGlide, a web-based application aimed at solving the major problem that students face during their FYP process. In this section we will explore different stages involved in the achieving the desired functionality of the application by discussing key design decisions, strategies and tactics.

## 5.1 System Overview

FASTGlide is a web application, which is developed to ease the FYP (Final Year Project) process for students, supervisors, and coordinators as well. It allows students to look out for an FYP group members, find supervisors, register their projects and finally submit final report. The application helps interact all group members with their supervisors, schedule a meeting (online or physical), and send notifications and requests about project initiatives. In addition, supervisors can share projects, approve group requests, and review submitted works, and coordinators can control how the submissions are made. The application is designed in a way that all future growth targets are reached providing a good flow of use in all components that it presents.

## 5.2 Design Considerations

This section describes many issues that need to be addressed or resolved before attempting to devise a complete design solution.

### 5.2.1 Assumptions and Dependencies

The following assumptions and dependencies associated with our web application need to be addressed:

#### 5.2.1.1 User Access to Internet

It is assumed that all users—students, supervisors, and coordinators—will have regular access to a stable internet connection. Since the system is web-based, a reliable internet connection is essential for accessing features such as FYP registration, communication, and notifications.

#### 5.2.1.2 Basic Technological Proficiency

We assume that users will possess basic skills in using computers and the internet. This foundational knowledge will enable them to navigate the system effectively and utilize features like scheduling meetings and communicating with others.

#### 5.2.1.3 Use of University Email

We assume that all users will have active FAST email accounts, which will be utilized for user authentication and communication through the platform. This integration ensures that users can log in with their university credentials and receive notifications and updates related to their FYP activities.

#### 5.2.1.4 Availability of Mobile Devices

We assume that many users may access the system through their mobile devices. Therefore, the app will be designed to be mobile-friendly, ensuring it works effectively on various screen sizes.

#### 5.2.1.5 Meeting Platforms

The system facilitates both online and physical meetings between students and supervisors. For online meetings, it assumes users can integrate commonly used platforms like Zoom, Microsoft Teams, or Google Meet. These platforms should be accessible to users for scheduling and conducting meetings effectively.

#### 5.2.1.6 Browser Compatibility

The app assumes that users will be accessing it through standard web browsers such as Chrome, Firefox, or Safari. Ensuring compatibility across major browsers is crucial for delivering a consistent user experience.

#### 5.2.1.7 Possible and/or Probable Changes in Functionality

The FYP process may change over time, with changes in university requirements or submission procedures. Therefore, the application is designed to be flexible and adaptable to accommodate future changes in the FYP structure or university guidelines.

### 5.2.2 General Constraints

#### 5.2.2.1 Internet Dependency

A continuous and stable internet connection is required for smooth operation of the application without encountering disruptions that may hinder functionality.

#### 5.2.2.2 Device Compatibility

Device compatibility is necessary for allowing users to have a good experience instead of restricting them to a single device. Therefore, it must function smoothly across different devices such as desktops,

laptops, tablets, and smartphones.

#### 5.2.2.3 Browser Compatibility

In addition to device compatibility, the app should also be browser-compatible for a consistent user experience. It should support major browsers such as Chrome, Firefox, and Safari.

#### 5.2.2.4 User Proficiency

Users are assumed to have basic technological skills, which may limit functionality if users are unfamiliar with web applications.

#### 5.2.2.5 Email Requirements

All users must have valid university email accounts for authentication, which could restrict access for those lacking such accounts.

#### 5.2.2.6 Data Integrity

The system must maintain accurate and up-to-date information regarding users, FYP projects, and deliverable submissions to prevent errors and confusion.

#### 5.2.2.7 Institutional Policies

The web app must comply with the university's policies regarding student data privacy, project submission guidelines, and academic integrity.

#### 5.2.2.8 Load Handling

The web app should be able to handle peak loads during critical times such as registration periods without significant slowdowns or outages.

### 5.2.3 Goals and Guidelines

When designing FASTGlide, the main goal is to make the FYP process smoother and more manageable for everyone involved—students, supervisors, and coordinators. The system should be intuitive and easy to use, so that users can quickly navigate and engage with its features without confusion. Flexibility and scalability of the platform in addition to its intended use should also be taken into consideration. Future growth may change how university policies or guidelines affect high usage times during activities such as when project registration or submission deadlines are in place. In this domain of sensitive

information pertaining to the student and project, importance of the security of the system also is a higher priority. Lastly, FASTGlide will be designed to work well on any device meaning a responsive application, whether it's a laptop or smartphone, ensuring students can access it anytime, anywhere, without unnecessary barriers.

### 5.2.4 Development Methods

The project shall be managed by adopting the Scrum methodology. The Scrum project management method is very self-explanatory and flexible. Scrum utilizes short periods of time called sprints with the aim of producing working software through frequent evaluation and modification. A sprint meeting is held every day, focusing on the work that is needed to be completed on that particular day. Additionally, there is a meeting at the end of every sprint to review the work done in that incremental stage of the sprint. Furthermore, a sprint retrospective meeting is carried out before the start of the new sprint to understand what goals or other improvements need to be accomplished in the next sprint.

## 5.3 System Architecture

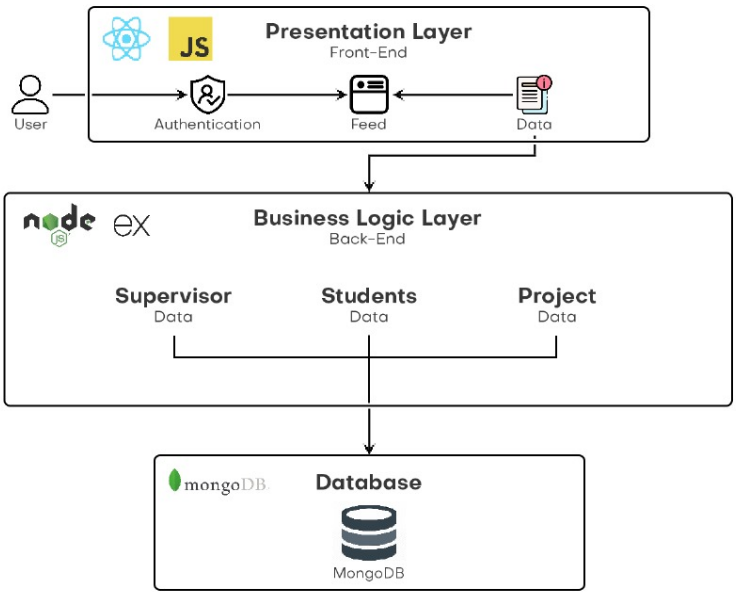


**Figure 5.1: High-Level System Architecture Diagram**

This section discusses the architecture of our system. The FASTGlide web app is designed using a layered architecture that incorporates client-server architecture. Each layer has a defined role and interacts with other layers to ensure the smooth workflow of our web app's functionality.

The architecture is divided into three main layers:

**1. Presentation Layer (Front-End):** This layer is built using React and JavaScript, handling the user interface and rendering the app's UI components such as authentication forms, login, user feeds, format checking, communication components, and submission components. Users interact with the system through this layer, performing actions like logging in, searching for FYP groups and supervisors, communicating, checking formats, and submitting deliverables. The Presentation Layer communicates with the backend through API calls to facilitate data operations.

**2. Business Logic Layer (Back-End):** Utilizing Node.js and Express.js, this layer manages data processing and user requests, including authentication, communication between students and supervisors, FYP registration, scheduling, and format verification of deliverables. It consists of modules for managing supervisor data, student data, FYP project data, registration data, and submissions, ensuring proper handling of user interactions and secure, efficient data processing. For format checking, the backend processes the LaTeX files of deliverables to ensure compliance with the university's formatting guidelines (e.g., margins, font size, section structure, image captions, proper hierarchy). Node.js handles this through libraries or custom scripts that parse LaTeX documents and validate their structure.

**3. Database Layer:** The system employs a NoSQL database, specifically MongoDB, responsible for storing all web app data, including student profiles, supervisor information, FYP project details, communication logs, and deliverables. MongoDB's flexible document structure allows for easy scaling as the amount of data grows, making it suitable for storing complex, interconnected data like FYP projects and group collaborations.

By integrating these layers, FASTGlide provides a seamless and efficient experience for users while ensuring data integrity and adherence to university guidelines.

### 5.3.1 Subsystem Architecture

Each of the system's layers contains subsystems that interact with one another to fulfill the system's functionalities. The main subsystems include:

#### 5.3.1.1 Authentication Subsystem

The Authentication Subsystem is responsible for managing user logins using university email accounts. It ensures secure access to the platform through token-based authentication methods, likely utilizing JWT (JSON Web Tokens). This subsystem also handles user registration and password recovery processes, enhancing user management and security.

#### 5.3.1.2 Student, Supervisor, and Project Data Subsystem

This subsystem manages data related to students, supervisors, and FYP projects, facilitating CRUD (Create, Read, Update, Delete) operations on user and project details. It ensures data consistency across the different user types and their associated projects, minimizing errors and ensuring accurate data representation. Additionally, it manages relationships between users and their respective projects, ensuring that information is accessible and organized.

#### 5.3.1.3 Notification and Communication Subsystem

The Notification and Communication Subsystem is responsible for sending notifications to students and supervisors about project updates, meeting requests, and registration statuses. It manages communication between students and their group members, as well as interactions between students and supervisors, facilitating chat-like conversations and meeting scheduling. This subsystem also logs communication history, providing users with an easily accessible record of interactions.

#### 5.3.1.4 Deliverable Submission and Format Subsystem

This subsystem allows students to submit their final project deliverables for review and checks for format verification against established guidelines. It facilitates the submission process by sending these files to coordinators and providing real-time feedback to students if submissions fail to meet the criteria. This proactive feedback mechanism helps students correct errors before final submission, improving the quality of deliverables.

#### 5.3.1.5 File Upload and Processing Workflow

The File Upload and Processing Workflow is designed to efficiently handle LaTeX documents, ensuring they are properly converted or compiled as needed for format verification. This workflow includes the steps of uploading the file to the server, parsing it, and running validation checks to ensure compliance with formatting guidelines. By addressing LaTeX format-specific issues on the back-end, the architecture ensures a thorough, scalable, and flexible validation process that can accommodate future changes to formatting standards.

## 5.4 Architecture Strategies

The design of the FYP Management App follows several key strategies to ensure scalability, maintainability, and user satisfaction.

### 5.4.1 Modularity

The architecture is organized into three distinct layers: presentation, business logic, and database. This division promotes modularity, enabling easier maintenance and scalability of the application. Each layer can be developed, tested, and deployed independently, which enhances collaboration among development teams.

### 5.4.2 RESTful API Design

The application utilizes a RESTful API approach for communication between the front-end and back-end. This design ensures that data requests and responses are handled in a structured manner, which simplifies both development and troubleshooting processes. It adheres to standard HTTP methods (GET, POST, PUT, DELETE), promoting a clear and predictable interface for developers.

### 5.4.3 Responsive UI

The front-end is designed to be responsive, ensuring the application functions smoothly across a variety of devices, including desktop PCs and smartphones. This responsiveness enhances usability and accessibility, allowing users to engage with the system seamlessly, regardless of their device. Techniques such as fluid grids, flexible images, and CSS media queries are employed to achieve a dynamic layout.

### 5.4.4 Scalability

MongoDB, a NoSQL database, has been selected for its flexibility and scalability. It can efficiently handle increased data loads as the number of users (students, supervisors, and projects) grows without performance degradation. The schema-less nature of MongoDB allows for easy modifications to the database structure, facilitating future changes and additions to the system.

### 5.4.5 Security

The architecture incorporates best practices for security, particularly in user authentication and data protection. Secure token mechanisms (such as JWT) are used for authentication, ensuring that user sessions are handled securely. Communication between different layers is protected using HTTPS protocols, and sensitive data stored in the database is encrypted to prevent unauthorized access.

### 5.4.6 Agile Development

The system architecture supports agile development methodologies, allowing for incremental improvements based on user feedback. This approach fosters a flexible development process, enabling quick

adaptations to changing user needs without necessitating extensive rewrites or major structural changes. Regular sprint reviews and retrospectives help the team identify areas for enhancement and streamline future development efforts.

## 5.5 Domain Model/Class Diagram

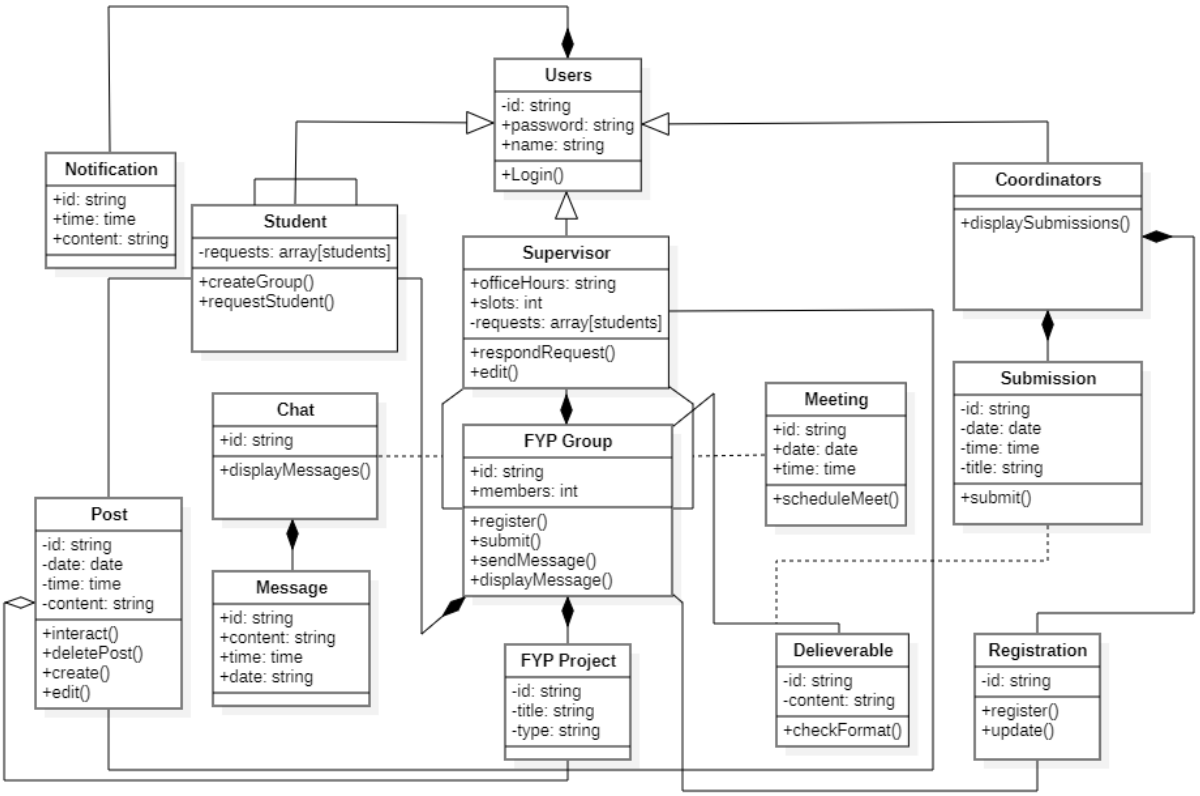


**Figure 5.2: Class Diagram**

## 5.6 Policies and Tactics

The design and development of the FASTGlide web app follows the following policies and tactics.

### 5.6.1 Security Policy

Given the nature of the web application, which revolves around sensitive student and faculty information, security is of utmost importance. Communication between the client and the server is secured over HTTPS, ensuring that data is always encrypted during transmission. For user authentication, we will implement JWT (JSON Web Tokens), a widely accepted paradigm that restricts system resources to authenticated users. Role-based access control (RBAC) will be utilized to veil resources stored in the database, allowing only certain user roles to access their respective information. This helps maintain the integrity and confidentiality of user data.

### 5.6.2 Performance Optimization Tactics

The application is designed to maintain a high level of usability, even with an increasing number of users. To achieve this, the server employs asynchronous execution and caching techniques to reduce server workload and improve response speed. Specific features, such as LaTeX file submissions for deliverables, are optimized to handle simultaneous user interactions, ensuring that the system remains responsive even during heavy transactions.

### 5.6.3 Tools

FASTGlide utilizes various tools to facilitate the development process and enhance the quality of the final deliverable. For frontend design, ReactJS libraries are employed to create an interactive user interface. The component-based structure of ReactJS promotes code reusability and maintainability, making the application easier to develop and manage over time. On the backend, Node.js and Express.js are used to build a high-performance, scalable server. To address storage needs, we utilize the document-oriented database, MongoDB, which is well-suited for the evolving nature of our data, including student, supervisor, and project information. Throughout the development process, Visual Studio Code IDE is employed as the main software for code writing and management, enhancing workflow with its fast performance and rich features.

### 5.6.4 Development Tactics

The development process is organized into sprints according to the Scrum methodology, allowing for incremental implementation of functionality. Development is continuously monitored through frequent sprint reviews and retrospectives, ensuring that the project remains on track and that any problems or obstacles are addressed promptly. This agile approach fosters collaboration and adaptability, enabling the team to incorporate user feedback and adjust project priorities as needed.

## 5.7 Conclusion

This section provides a thorough architectural framework for the construction of the FASTGlide system. The design employs a layered architecture with three levels, namely: presentation, business logic and database, which enhances modularity, scalability and maintainability of the components. The hight level design describes structure of the entire system allowing users to interact with the platform seamlessly and it narrows down into the low level how the detailed components ,subsystems and data runs. This particular architecture gives the basis of the key functions of the FYP management system, that is, communication between students and supervisors, submissions of deliverables, and progress of students

but at the same time leaves room for future developments and additional features. The design discussed in this chapter provides a sound technical base which makes the system potential and effective during its operations.

# Chapter 6 Implementation and Test Cases

This section talks about the development and implementation details of our FASTGlide modules.It mentions different stratgies and techniques used to implement the feature of our web app.

## 6.1 Implementation

The implement details of our modules are given below.

### 6.1.1 Authentication module

The Authentication module is very important part of the FASTGlide. Since FASTGlide is dedicated purely to students and teachers of FAST university only, any external access will be strictly prohibited. Additionally, safeguarding confidential information about students and their FYP projects is paramount, ensuring that no unauthorized users, including unverified students, can access the system. The front end for the login is built using ReactJS, while the back end uses Node.js and Express for handling authentication logic. To prevent unauthorized access, the system employs JSON Web Tokens, a widely used and highly secure method of authentication. Tokens will have a short expiry time to minimize the risk of session hijacking and ensure users reauthenticate frequently. Student and teacher credentials will be preloaded into the database in an encrypted form. Bcryptjs a javascript library for hashing will be used for passwords, ensuring they are stored securely and resistant to brute force attacks. A unique password generated by script will be sent to teachers and student through email with an alert to change the password immediately for enhancing security. The token will be securely stored in cookies, preventing client side scripts from accessing sensitive token data.

### 6.1.2 FYP Management module

Since the main objective of FASTGlide is to streamline the Final Year Project (FYP) process by helping students find supervisors, explore project ideas, and connect with group members. The FYP Portal allows students to search for other students who share similar interests, with filtering options to narrow down the search. Similarly, students can explore a list of supervisors in another section, where they can search by expertise, department, or availability and send requests for collaboration. The main page of the platform features a ”Wall” where supervisors post project ideas, encouraging students to explore and collaborate on topics of mutual interest. For searching process, a GET request will be send to server along with search query since the return data of student and supervisor will be large, we will have to implement efficient queries and employ pagination and infinite scroll strategy to optimize the whole process. Similarly for implementing wall the infinite scroll strategy will be used. A separate registra-

tion section allows students to easily create FYP groups by providing all necessary details required by university. Once a group is registered and linked to a supervisor, the system facilitates communication through features like group chats with the assigned supervisor and integrated meeting scheduling. These functionalities are implemented using simple POST requests to the server with the corresponding data. Upon successful validation, the server sends notifications to the respective students and the supervisor. For meetings, Google Meet can be used, as the university has a contract with Google that provides access to premium features of Google Meet. Data related to students, supervisors, project ideas, and group details is stored in a MongoDB database, ensuring flexibility and scalability as data grows. RESTful APIs connect the front-end with the back-end, providing a seamless flow of data.

### 6.1.3 Chat module

The Chat module in FASTGlide enables real time communication between students and their supervisors. The Chat module is implemented using WebSockets, which provides full duplex communication channels over a single, long-lived connection. This enables real-time message exchange without the need for continuous page refreshes. A WebSocket client is integrated into the React components using a WebSocket library such as Socket.IO. When a user sends a message, it is emitted to the server, and any responses from the supervisor or group members are immediately received and displayed in the chat window. On the back end, nodejs with Socket.IO is used to handle web socket connections. The server manages multiple chat rooms, where each room corresponds to a specific FYP group or a student-supervisor conversation. When a user logs into the platform, they are connected to their relevant chat rooms based on their active groups or assigned supervisors. The web socket server listens for incoming connections, and when a message is received from a user, it is broadcast to all users in the same chat room. To ensure security, the system uses json web token authentication for validating users before allowing them to join chat rooms. Messages sent within the chat are stored in a MongoDB database to maintain chat history.

### 6.1.4 Latex Checking module

The LaTeX Report Checking Module in FASTGlide is designed to ensure that student FYP submissions are according to university formatting guidelines. The module utilizes node and express to manage file uploads, and multer library to handle the extraction of ZIP archives containing LaTeX files. Once the ZIP file is extracted, the module checks the LaTeX content for several critical formatting issues, such as verifying that student names are properly listed in the required section, ensuring references in the .bib file follow the correct format, and checking that figures include captions, ensuring they do not start with ”This figure...”. The system also verifies that all required figures and signatures are present.

Additionally, the LaTeX document is compiled using pdflatex, and if compilation fails, the log file is analyzed for errors, providing detailed feedback on the issue. The module also includes functionality to extract student names from the LaTeX content and checks for correct inclusion within specified sections. We are currently using Regex for parsing the latex file. There are libraries available as well for parsing like latex.js or TeXParser that can help break down LaTeX into a more structured representation. We will be using computer vision techniques a library such as OpenCV for detecting layout issues like incorrect spacing or margin problems in the compiled PDF along with library pdf2text to analyze the spacing and layout in a more programmatic manner. The results and findings will be sent back to client in a response body which will be displayed to user in defined UI.

# Chapter 7 Conclusion and Future Work

The web-based application FASTGlide is being developed to make the Final Year Project process easy, providing a platform for students to find supervisors, form groups, and collaborate on projects. In this phase, we have focused primarily on developing the LaTeX module, which aims to automate the checking and validation of LaTeX document formats in accordance with university guidelines. This module addresses key issues faced by evaluators and coordinators who currently must manually check each student's FYP report for common formatting and structural errors. By automating this process, our module helps students identify and correct major mistakes before submission, significantly reducing the workload for evaluators and ensuring that reports adhere to university standards. The current implementation provides and solves the basic validation issues faced for LaTeX documents, such as checking the format for student names, figure captions, and references, using regex and simple parsing techniques. However, this phase has been limited to addressing the most common issues, and we plan to extend the module's capabilities in future phases. The next step will include more rigorous and strict checking mechanisms, ensuring that the LaTeX file complies fully with university guidelines, such as proper spacing, consistent formatting, and accurate citation styles. This will involve integrating advanced LaTeX parsing libraries and employing image processing tools to check for spacing and layout errors, further improving the accuracy and comprehensiveness of the validation. Looking ahead, the FYP-2 phase will focus on completing the remaining functional requirements of the system, including the FYP management module, chat module and authentication module. Additionally, the LaTeX checking module will be enhanced to provide more detailed feedback and automatic correction suggestions for students, enabling a smoother and more efficient submission process. Testing and refinement will be conducted to ensure the system is scalable, robust, and user-friendly. In conclusion, the work completed so far lays a solid foundation for FASTGlide. The LaTeX module addresses critical pain points faced by evaluators and students by automating the document validation process, helping students avoid major errors before submission. Future work will focus on improving the more in depth checking of LaTeX files, aligning them more closely with university guidelines, and further enhancing the system's overall functionality. This will bring us closer to delivering a comprehensive and efficient solution for FYP management and evaluation.