**Graphical user interface, text

Description automatically generatedKing Saud University**

**College of Computer and Information Sciences**

**Department of Information Systems**

**1444-1445 / 2022-2023**

**First Semester**

GRADUATION PROJECT I (IS498)

BS Project Committee Assistance Software

Icon

Description automatically generated

Prepared by:

|  |  |  |
| --- | --- | --- |
| Name | Email | ID |
| Abdulaziz Aldaoud | [441100737@student.ksu.edu.sa](mailto:441100737@student.ksu.edu.sa) | 441100737 |
| Musab Alghamdi | [438102298@student.ksu.edu.sa](mailto:438102298@student.ksu.edu.sa) | 438102298 |
| Naif Alsadoun | [441101373@student.ksu.edu.sa](mailto:441101373@student.ksu.edu.sa) | 441101373 |

Supervised by:

Dr. Bader Alkhamees

## **Abstract**

In this project we aim to aid the IS department project committee in automating their work of managing the proposals and assigning each group to a project, so the website will organize committee’s work and make it easier by providing an easy interface and automatic ranking system and other features, the website will calculate the priority of each group according to their GPA average and assign them a project to work on so they can finish IS 498 course and inform the IS department of each group and their respective project and faculty member, we used the waterfall methodology to complete our work and used: Use case diagram, Entity relationship diagram, Activity diagrams, Sequence diagrams and Class diagram to illustrate our work and better understand it, and we will be implementing the website in the next semester in graduation project II (IS499).

**Table of contents:**

[**Abstract** 2](#_Toc118730691)

[CHAPTER 1: INTRODUCTION 6](#_Toc118730692)

[**1.1 Introduction:** 7](#_Toc118730693)

[**1.2 Motivations:** 7](#_Toc118730694)

[**1.3 Workflow of Current System:** 8](#_Toc118730695)

[**1.4 Objectives:** 9](#_Toc118730696)

[**1.5 Positional Users:** 9](#_Toc118730697)

[**1.6 Gathering Data:** 9](#_Toc118730698)

[**1.7 Gantt Chart:** 10](#_Toc118730699)

[CHAPTER 2: LITERATURE REVIEW 11](#_Toc118730700)

[**2.1 Literature Review:** 12](#_Toc118730701)

[CHAPTER 3: SYSTEM DEVELOPMENT METHODOLOGY & REQUIREMENTS 15](#_Toc118730702)

[**3.1 System Development Methodology:** 16](#_Toc118730703)

[**3.2 Waterfall vs Agile:** 18](#_Toc118730704)

[**3.3 Why Did We Decide To Use The Waterfall Model?** 18](#_Toc118730705)

[**3.4 Functional Requirements:** 19](#_Toc118730706)

[**3.5Non-Functional Requirements:** 20](#_Toc118730707)

[CHAPTER 4: SYSTEM ANALYSIS 21](#_Toc118730708)

[4.1 Use Case Diagram: 22](#_Toc118730709)

[**4.2 Use Case Description:** 23](#_Toc118730710)

[**4.3 Activity Diagram:** 41](#_Toc118730711)

[CHAPTER 5: SYSTEM DESIGN 48](#_Toc118730712)

[**5.1 Entity Relationship Diagram:** 49](#_Toc118730713)

[**5.3 Sequence Diagram:** 51](#_Toc118730714)

[**5.4 Class Diagram:** 56](#_Toc118730715)

[CHAPTER 6: 58](#_Toc118730716)

[**6.1 Social Impact:** 59](#_Toc118730717)

[**6.2 Ethical Impact:** 59](#_Toc118730718)

[**6.3 Legal Impact:** 59](#_Toc118730719)

[**6.4 Global Impact:** 59](#_Toc118730720)

[**6.5 Security Impact:** 59](#_Toc118730721)

[**6.6 Conclusion:** 60](#_Toc118730722)

[**6.7 Reference List:** 61](#_Toc118730723)

**Table of Figures:**

[*Figure 1: Workflow of current system* 8](#_Toc118725317)

[Figure 2: Gantt chart 10](file:///C:\Users\musab\Downloads\Report%20(3).docx#_Toc118725318)

[Figure 3: Consolidated electronic admission 12](#_Toc118725319)

[Figure 4: Consolidated electronic admission (2) 12](#_Toc118725320)

[Figure 5: Ranking majors 13](#_Toc118725321)

[Figure 6: Noor changing region 14](#_Toc118725322)

[Figure 7: Agile model 16](#_Toc118725323)

[Figure 8: Waterfall model 17](#_Toc118725324)

[Figure 9: Use case diagram 22](#_Toc118725325)

[Figure 10: Authorize students Activity Diagram 41](#_Toc118725326)

[Figure 11: Add proposal Activity Diagram 42](#_Toc118725327)

[Figure 12: Set load Activity Diagram 43](#_Toc118725328)

[Figure 13: Create group Activity Diagram 44](#_Toc118725329)

[Figure 14: Respond to invitation Activity Diagram 45](#_Toc118725330)

[Figure 15: Leave group Activity Diagram 46](#_Toc118725331)

[Figure 16: Contact group Activity Diagram 47](#_Toc118725332)

[Figure 17: Entity Relationship Diagram 49](#_Toc118725333)

[Figure 18: Relational Schema 50](#_Toc118725334)

[Figure 19: Registration Sequence Diagram 51](file:///C:\Users\musab\Downloads\Report%20(3).docx#_Toc118725335)

[Figure 20: login 52](file:///C:\Users\musab\Downloads\Report%20(3).docx#_Toc118725336)

[Figure 21: Proposal addition Sequence Diagram 53](file:///C:\Users\musab\Downloads\Report%20(3).docx#_Toc118725337)

[Figure 22: Assigning Proposals Sequence Diagram 54](#_Toc118725338)

[Figure 23: Group interactions Sequence Diagram 55](file:///C:\Users\musab\Downloads\Report%20(3).docx#_Toc118725339)

[Figure 24: invitation response 56](file:///C:\Users\musab\Downloads\Report%20(3).docx#_Toc118725340)

[Figure 25: Class Diagram 57](#_Toc118725341)

**Table of Tables:**

[Table 1: Positional Users: 9](#_Toc118725493)

[Table 2: Waterfall vs Agile 18](#_Toc118725494)

[Table 3: Faculty member Functional requirements 19](#_Toc118725495)

[Table 4: Student functional requirements 19](#_Toc118725496)

[Table 5: Committee member functional requirements 20](#_Toc118725497)

[Table 6: Non-Functional requirements 20](#_Toc118725498)

[Table 7: Register use case description 23](#_Toc118725499)

[Table 8: Log-In use case description 24](#_Toc118725500)

[Table 9: publish proposals use case description 25](#_Toc118725501)

[Table 10: Set load use case description 26](#_Toc118725502)

[Table 11: Set student count use case description 27](#_Toc118725503)

[Table 12: Set deadline use case description 28](#_Toc118725504)

[Table 13: Authorize student use case description 29](#_Toc118725505)

[Table 14: Add proposal use case description 30](#_Toc118725506)

[Table 15: Delete proposal use case description 31](#_Toc118725507)

[Table 16: Submit evaluation use case description 32](#_Toc118725508)

[Table 17: View group use case description 33](#_Toc118725509)

[Table 18: Contact group use case description 34](#_Toc118725510)

[Table 19: Submit report use case description 35](#_Toc118725511)

[Table 20: Rank proposals use case description 36](#_Toc118725512)

[Table 21: Create group use case description 37](#_Toc118725513)

[Table 22: Leave group use case description 38](#_Toc118725514)

[Table 23: Send invitation use case description 39](#_Toc118725515)

[Table 24: Respond to invitation use case description 40](#_Toc118725516)

# CHAPTER 1: INTRODUCTION

## **1.1 Introduction:**

Every year the IS department project committee goes through a long and tedious process of assigning each group to a faculty member the process as is, wastes time and students may need a lot of guidance on how the process works.

The process is very prone to human error, so we want to automate this process and make it as easy as possible, and ultimately limit unnecessary email usage for both the students and faculty members.

We also plan on providing the students with guidelines, FAQ, and the project handbook and provide the IS department with the collected data form these processes ready to be analyzed to help them make better decisions.

## **1.2 Motivations:**

We ourselves have faced difficulties in understanding the process of assigning each group to a proposal and it’s respective faculty member because it is not clear and not practical enough, and we find that some groups don't get any of the projects they have chosen in their top 10 selections because a faculty member submitted a proposal when he has reached his maximum workload, which means the proposal isn’t valid anymore, another problem is that we usually see some students who might not find a group that matches their skills and knowledge or may not find a group at all then they would go to the committee for help or may ask redundant questions that may waste the committee’s time which can be easily answered in one place, so we want to fully automate this process and hopefully make it a better experience for all parties involved.

## **1.3 Workflow of Current System:**

Diagram

Description automatically generated

Figure 1: Workflow of current system

It’s clear that the current manual system is not practical enough, our website aims to facilitate and automate the entire process for committee members and avoid any errors that may occur from human element.

## **1.4 Objectives:**

* Limit user error
* Limit Email usage
* Easier work and less workload for committee members
* Organizing committee’s work
* Provide a chat room for the group and their supervising faculty members
* Provide data ready to be analyzed
* Provide automatic ranking system
* Provide an easy interface
* Check faculty member’s loads to determine if they are available
* Make project details and requirements clear
* Make handbook, Guideline, and FAQ easily accessible

## **1.5 Positional Users:**

|  |  |
| --- | --- |
| User | Description |
| Students | Who will use the website to view project proposals details and rank their top 10 desired proposal. |
| Faculty members | Who will post proposals and wait to be assigned a group. |
| Committee member | Who will manage the system: Authorize students, manage proposals, etc. |

### *Table 1: Positional Users:*

## **1.6 Gathering Data:**

We collected information related to this project from our advisor Dr. Bader Alkhamees who was previously head of the project committee, and we interviewed Dr. Ashraf, they answered our questions about the committee’s work and walked us through the entire process of the current system, including how the committee members, faculty members and students play their parts in the current system and the problems that could occur since the system is impractical and manual,

They also told us the features that could better automate the committee’s work.

## **1.7 Gantt Chart:**

Chart

Description automatically generated

Figure 2: Gantt chart

# CHAPTER 2: LITERATURE REVIEW

## **2.1 Literature Review:**

1. Consolidated electronic admission

[****](https://www.rbu-admit.edu.sa/)

Figure 3: Consolidated electronic admission

**Graphical user interface, text

Description automatically generated**

Figure 4: Consolidated electronic admission (2)

Our project is very similar to this website, used to help

high school graduates rank and apply to colleges in their region, it’s very easy and intuitive and we aim to replicate that.

1. Universities desirable majors ranking

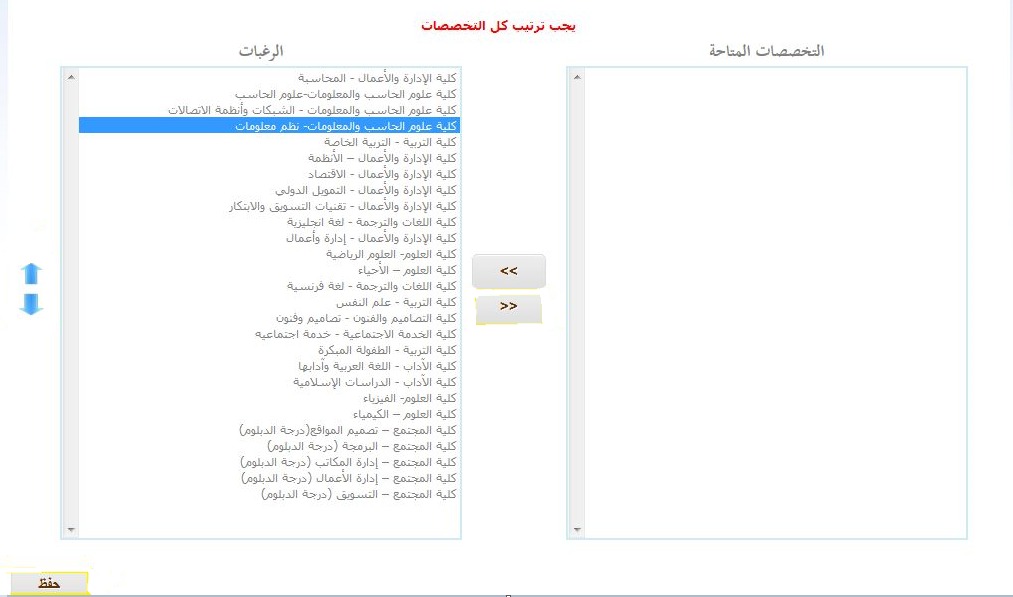
****

Figure 5: Ranking majors

This is another example very similar but instead it is used to priorities majors after finishing the first common year, it calculates

your GPA plus some other parameters that will be used in a set formula that is very much different for every university to determine your major.

1. Noor:

****

Noor is a system provided by the Ministry of Education to be the intermediate between the ministry and related individuals: students and teachers.

****

Figure 6: Noor changing region

Noor allows teachers to rank their preferred regions to teach in, and the system ranks the teachers based on their job performance, years of experience and whether they are needed in a specific region or not, to give them priority on the regions they chose.

We would like to take inspiration from these examples, so we hope that our contribution can be used to make a better system in the future.

# CHAPTER 3: SYSTEM DEVELOPMENT METHODOLOGY & REQUIREMENTS

## **3.1** **System** **Development Methodology:**

**Agile:**

Diagram

Description automatically generated

Figure 7: Agile model

The Agile model was primarily designed to help a project to adapt to change requests quickly. So, the main aim of the Agile model is to facilitate quick project completion. Sometime agility is required to accomplish tasks. Agility is achieved by fitting the process to the project, removing activities that may not be essential for a specific project. Also, anything that may waste time and effort is avoided.

In the Agile model, the requirements are decomposed into many small parts that can be incrementally developed. The Agile model adopts Iterative development. Each incremental part is developed over an iteration. Each iteration is intended to be small and easily manageable and can be completed within a couple of weeks only. At a time one iteration is planned, developed, and deployed to the customers. Long-term plans are not made.

**Waterfall:**

Diagram

Description automatically generated

Figure 8: Waterfall model

The Waterfall methodology is a sequential development process that flows like a waterfall through all phases of a project (analysis, design, development, and testing, for example), with each phase completely wrapping up before the next phase begins.

The Waterfall methodology depends on the belief that all project requirements can be gathered and understood upfront. The project manager does their best to get a detailed understanding of the project sponsor’s requirements.

## **3.2 Waterfall vs Agile:**

|  |  |  |
| --- | --- | --- |
|  | Waterfall | Agile |
| Sequential | ✓ |  |
| Flexible |  | ✓ |
| Defined requirement | ✓ |  |
| Suited to small projects | ✓ |  |
| Continually evolving |  | ✓ |
| Fixed project scope | ✓ |  |

### *Table 2: Waterfall vs Agile*

## **3.3 Why Did We Decide to Use The Waterfall Model?**

We believe that the waterfall is more suitable for our project because of these reasons:

* The project doesn’t have ambiguous requirements.
* We offer a clear picture of how things will proceed from the outset.
* We have clients who seem unlikely to change the scope of the project once it is underway.

Since we have clearly defined processes, design, and time requirements, then the Waterfall method is the way to go, if the project itself is conducive to those constraints.

## **3.4 Functional** **Requirements:**

**Faculty member:**

|  |  |  |
| --- | --- | --- |
| # | Requirement | Description |
| 1 | Register | Faculty members should be able to register into the website by entering his KSU email, password**.** |
| 2 | Log-In | Faculty members should be able to login into the website by entering his KSU email, password**.** |
| 3 | Add proposal | Faculty members should be able to add proposals for the groups to choose from**.** |
| 4 | View group | Faculty member should be able to view the groupmembers assigned to him**.** |
| 5 | Contact group | Faculty member should be able to contactgroupmembersassigned to him**.** |
| 6 | Delete proposal | Faculty members should be able to delete proposals**.** |
| 7 | Submit evaluation | Faculty members should be able to submit evaluation of the group members**.** |

### *Table 3: Faculty member Functional requirements*

**Student:**

|  |  |  |
| --- | --- | --- |
| # | Requirement | Description |
| 1 | Register | The student should be able to register into the website n by entering his KSU email, password**.** |
| 2 | Log-In | The student should be able to login into the websiteby entering his KSU email, password**.** |
| 3 | Rank proposals | The group leader **r**anking proposal from 10 to 1 (priority list)**.** |
| 4 | View group | Student should be able to view his groupmembers. |
| 5 | Contact group | Group members should be able to contact each other, and thefaculty member assigned to thegroup**.** |
| 6 | Create group | Student should be able to create a group**.** |
| 7 | Send Invitation | Send invitation to other students to join the group**.** |
| 8 | Leave group | Student should be able to leave his group**.** |
| 9 | Respond to invitation | Student should be able to accept or decline invitation**.** |
| 10 | Submit report | Group leader will be able to submit the report after the group have finished working on the project. |

### *Table 4: Student functional requirements*

**Committee member:**

|  |  |  |
| --- | --- | --- |
| # | Requirement | Description |
| 1 | Log-In | The committee member should be able to login into the website by entering his KSU email, password. |
| 2 | Set load | The committee member will enter faculty members workload for this semester. |
| 3 | Set student count | The committee member will decide how many students per group. |
| 4 | Set deadline | The committee member will enter deadline for registration period. |
| 5 | Publish proposals | The committee member selects desired proposals based on the load table. |
| 6 | Authorize student | The committee members will enter students who are permitted to register in the system to deny students who didn’t sign up for the course. |

### *Table 5: Committee member functional requirements*

## **Non-Functional Requirements:**

|  |  |  |
| --- | --- | --- |
| # | Requirements | Description |
| 1 | Security | * The system is protected by username and password with at least 8   Characters have at least one capital char and special char.   * Email authentication will only allow university’s emails to be used with matching id. |
| 2 | Availability | The system will run 24/7 till registration period ends. |
| 3 | Usability | The system will provide a user friendly interface. |
| 4 | Performance | The system will be light and responsive. |
| 5 | Reliability | The system will run consistently with no errors. |
| 6 | Integrity | The system will use military grade encryption. |

### *Table 6: Non-Functional requirements*

# CHAPTER 4: SYSTEM ANALYSIS

# **4.1 Use Case Diagram:**

A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. The system performs a set of actions, services, and functions.

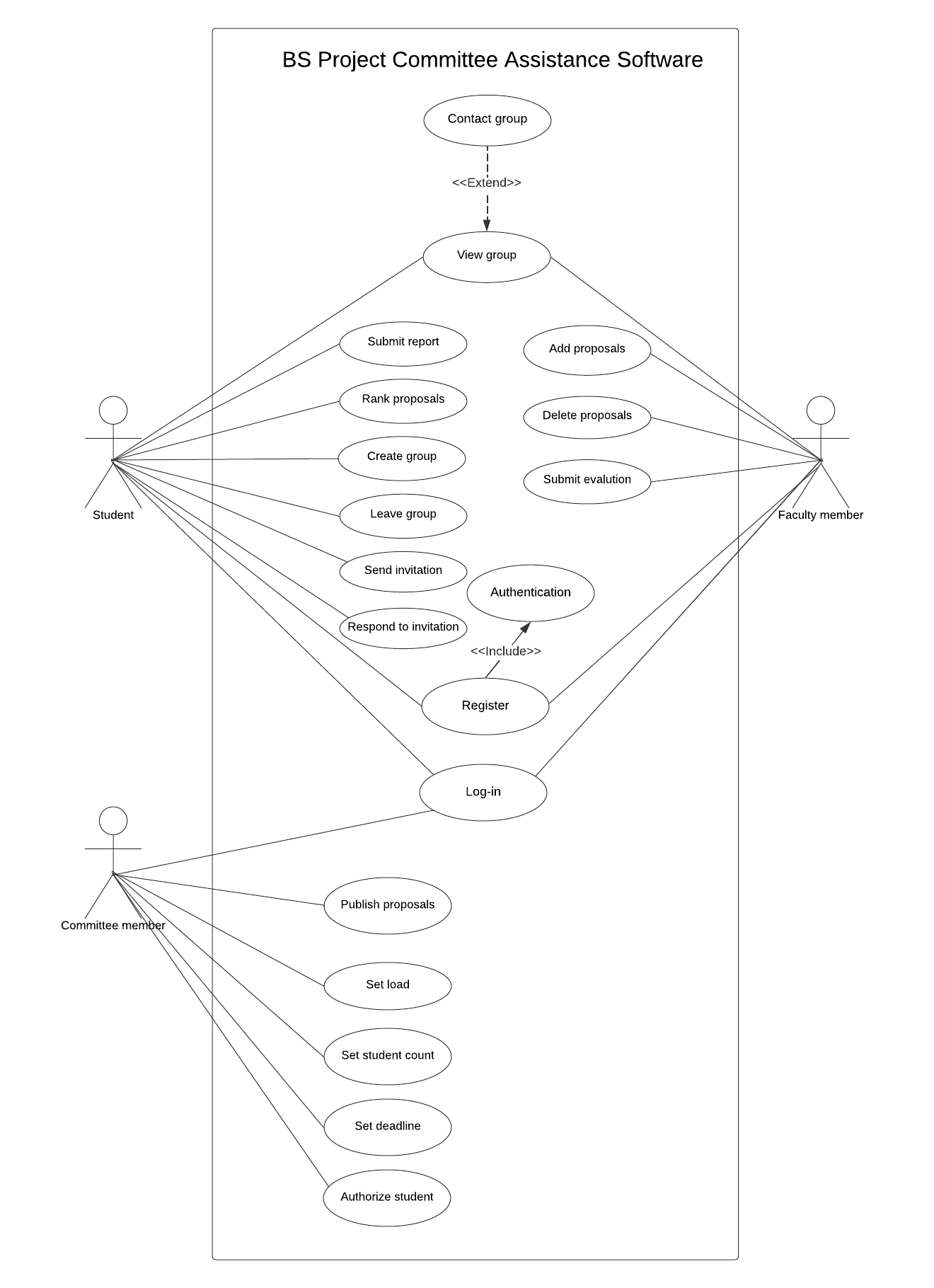


Figure 9: Use case diagram

## **4.2 Use Case Description:**

A use case is a written description of how users will perform tasks on your website. It outlines, from a user’s point of view, a system’s behavior as it responds to a request. Each use case is represented as a sequence of simple steps, beginning with a user's goal, and ending when that goal is fulfilled.

|  |  |  |
| --- | --- | --- |
| Use Case Name | Register | |
| Triggering Event | User wants to register. | |
| Brief Description | Fill in the required information and the system will validate it. | |
| Actors | Student, Faculty member. | |
| Related Use Cases | Login, authenticate. | |
| Stakeholders | Student, Faculty member. | |
| Preconditions | User must be authorized by Committee member.  User must provide valid information. | |
| Postconditions | User will be able to login. | |
| Flow of Activities | Actor | System |
| 1. User will fill in required information | * 1. System will validate the information   2. System will inform the user that an account has been created |
| Exception Conditions | User already exists  Incomplete required information | |

### *Table 7: Register use case description*

|  |  |  |
| --- | --- | --- |
| Use Case Name | Log-In | |
| Triggering Event | User wants to login | |
| Brief Description | Fill in the required information and the system will validate it. | |
| Actors | Student, faculty member, & Committee member. | |
| Related Use Cases | Register | |
| Stakeholders | student, faculty member, Committee member. | |
| Preconditions | User account must exist. | |
| Postconditions | The system display home page. | |
| Flow of Activities | Actor | System |
| 1. User will fill in required information | 1.1 System will validate the information   * 1. System will take the user to the homepage |
| Exception Conditions | Username or password incorrect | |

### *Table 8: Log-In use case description*

|  |  |  |
| --- | --- | --- |
| Use Case Name | Publish proposals | |
| Triggering Event | The system receives the load table from Load committee member. | |
| Brief Description | Committee member selects desired proposals based on the load table. | |
| Actors | Committee member. | |
| Related Use Cases | Set deadline, Add proposal, & Rank proposal. | |
| Stakeholders | Student. | |
| Preconditions | Deadline must be set, Proposals must be available, Load must be set. | |
| Postconditions | Students will be able to view and rank proposals. | |
| Flow of Activities | Actor | System |
| 1. User will publish proposals | * 1. System will make Proposals viewable to the students |
| Exception Conditions | Proposals not added. | |

### *Table 9: publish proposals use case description*

|  |  |  |
| --- | --- | --- |
| Use Case Name | Set load | |
| Triggering Event | Load committee finalizes faculty member’s load table. | |
| Brief Description | Committee member will enter how many hours faculty member is working this semester. | |
| Actors | Committee member. | |
| Related Use Cases | None. | |
| Stakeholders | Faculty members. | |
| Preconditions | None. | |
| Postconditions | System will decide if a faculty member should be relived or not . | |
| Flow of Activities | Actor | System |
| 1. User will enter hours | 1.1 System will show faculty members who can’t be assigned a group to work on the proposal |
| Exception Conditions | None. | |

### *Table 10: Set load use case description*

|  |  |  |
| --- | --- | --- |
| Use Case Name | Set student count | |
| Triggering Event | Committee member wants to set the number of members in group after proposal submission deadline. | |
| Brief Description | Choose the number of members in a group. | |
| Actors | Committee member. | |
| Related Use Cases | None. | |
| Stakeholders | Student. | |
| Preconditions | None. | |
| Postconditions | System sets group member count limit. | |
| Flow of Activities | Actor | System |
| 1. User will fill desired number | * 1. System will calculate the number of groups   2. System will display number of groups to user   3. System will take the user to the homepage |
| Exception Conditions | Less than 2. | |

### *Table 11: Set student count use case description*

|  |  |  |
| --- | --- | --- |
| Use Case Name | Set deadline | |
| Triggering Event | Committee member wants to set the date of last day of entry | |
| Brief Description | Select the desired date for faculty members adding proposals and for group rankings, etc. | |
| Actors | Committee member. | |
| Related Use Cases | None. | |
| Stakeholders | Faculty members, Student. | |
| Preconditions | None. | |
| Postconditions | System set date of last chance to entry. | |
| Flow of Activities | Actor | System |
| 1. User will fill desired dates | 1.1 System will update and display the dates on the homepage for all related users  1.2 System will take the user to the homepage |
| Exception Conditions | Less than 4 weeks from current date. | |

### *Table 12: Set deadline use case description*

|  |  |  |
| --- | --- | --- |
| Use Case Name | Authorize student | |
| Triggering Event | Registration committee provides table of students who have graduation project course this semester. | |
| Brief Description | Committee member wants to allow only the student who are taking this course to register. | |
| Actors | Committee member. | |
| Related Use Cases | Register. | |
| Stakeholders | Student. | |
| Preconditions | None. | |
| Postconditions | System allows only these students to register | |
| Flow of Activities | Actor | System |
| 1. User will enter students id | 1.1 System will allow these students to register |
| Exception Conditions | None. | |

### *Table 13: Authorize student use case description*

|  |  |  |
| --- | --- | --- |
| Use Case Name | Add proposal | |
| Triggering Event | Project committee sends request for proposal to faculty members before each semester starts. | |
| Brief Description | Faculty will add a proposal from the system. | |
| Actors | Faculty member. | |
| Related Use Cases | None. | |
| Stakeholders | Student, faculty members. | |
| Preconditions | Within specified submission time window. | |
| Postconditions | System displays these proposals to the student | |
| Flow of Activities | Actor | System |
| 1. User will add proposal | 1.1 System will add the proposal to the database  1.2 System will take the user to the homepage |
| Exception Conditions | Faculty member can’t add more than 2 proposals.  Can’t add the same proposal twice. | |

### *Table 14: Add proposal use case description*

|  |  |  |
| --- | --- | --- |
| Use Case Name | Delete proposal | |
| Triggering Event | Faculty member will want to delete his proposals. | |
| Brief Description | Faculty will delete a proposal from the database. | |
| Actors | Faculty member. | |
| Related Use Cases | None. | |
| Stakeholders | Student, faculty members. | |
| Preconditions | Faculty member should have at least one submitted proposal. | |
| Postconditions | Proposal is deleted. | |
| Flow of Activities | Actor | System |
| 1. User will select proposal | * 1. System deletes proposal from database   2. System confirms deletion   3. System will take the user to the homepage |
| Exception Conditions | Deadline has passed. | |

### *Table 15: Delete proposal use case description*

|  |  |  |
| --- | --- | --- |
| Use Case Name | Submit evaluation | |
| Triggering Event | The faculty member wants to submit group evaluation. | |
| Brief Description | After the group members have finished working on the project the faculty member evaluates their work and send the evaluation to the system. | |
| Actors | Faculty member. | |
| Related Use Cases | None. | |
| Stakeholders | Student, faculty member. | |
| Preconditions | Before deadline. | |
| Postconditions | Notify group members of evaluation. | |
| Flow of Activities | Actor | System |
| 1. User enter group evaluation | 1.1 System assigns the evaluation to the group members  1.2 System will take the user to the homepage |
| Exception Conditions | Group already evaluated. | |

### *Table 16: Submit evaluation use case description*

|  |  |  |
| --- | --- | --- |
| Use Case Name | View group | |
| Triggering Event | Faculty member/ student would want to view the group he is assigned/invited to. | |
| Brief Description | Faculty member/student will view the members of the group he is assigned/ invited to. | |
| Actors | Faculty member, student. | |
| Related Use Cases | None. | |
| Stakeholders | Student, faculty members. | |
| Preconditions | A group must be assigned first. | |
| Postconditions | None. | |
| Flow of Activities | Actor | System |
| 1. User will select view group | 1.1 System will display member and supervisor of the group to user |
| Exception Conditions | None. | |

### *Table 17: View group use case description*

|  |  |  |
| --- | --- | --- |
| Use Case Name | Contact group | |
| Triggering Event | Faculty member/ student wants to contact the members of the group he is assigned to. | |
| Brief Description | Faculty member/student will send a message to the members of the group he is assigned to. | |
| Actors | Faculty member, student. | |
| Related Use Cases | View group. | |
| Stakeholders | Student, faculty members. | |
| Preconditions | A group must be assigned first. | |
| Postconditions | A message will be sent to the selected member. | |
| Flow of Activities | Actor | System |
| 1. User will select group 2. User will type a message and send it | * 1. System will display the group to user   2.1 System will send the message to group members |
| Exception Conditions | None. | |

### *Table 18: Contact group use case description*

|  |  |  |
| --- | --- | --- |
| Use Case Name | Submit report | |
| Triggering Event | The group leader wants to submit project report. | |
| Brief Description | The group leader will submit the report after the group members have finished working on the project. | |
| Actors | Student. | |
| Related Use Cases | None. | |
| Stakeholders | Student, Faculty member. | |
| Preconditions | Before deadline. | |
| Postconditions | None. | |
| Flow of Activities | Actor | System |
| 1. User submits the report | 1.1 System assigns the report to the group  1.2 System will take the user to the homepage |
| Exception Conditions | Report already submitted. | |

### *Table 19: Submit report use case description*

|  |  |  |
| --- | --- | --- |
| Use Case Name | Rank proposals | |
| Triggering Event | Group members want to choose project proposal to work on. | |
| Brief Description | Ranking top 10 proposals according to group desire. | |
| Actors | Student. | |
| Related Use Cases | Add proposal, Publish proposal. | |
| Stakeholders | Student, faculty members. | |
| Preconditions | Student must be in a group. | |
| Postconditions | Assigning proposal to group. | |
| Flow of Activities | Actor | System |
| 1. User ranks top 10 desired proposals   2. User confirm changes | 1.1 System displays ranking  1.2 Asks user to commit changes |
| Exception Conditions | Ranking less or more than 10 proposals. | |

### *Table 20: Rank proposals use case description*

|  |  |  |
| --- | --- | --- |
| Use Case Name | Create group | |
| Triggering Event | Student wants to create a group. | |
| Brief Description | Student will create a group to work with for the project. | |
| Actors | Student. | |
| Related Use Cases | Send invitation. | |
| Stakeholders | Student, Faculty member. | |
| Preconditions | The student must not be a member of an existing group. | |
| Postconditions | Student can’t create another group. | |
| Flow of Activities | Actor | System |
| 1 User will select create group tab | * 1. System will create a group   1.2 System will display the group interface to the user |
| Exception Conditions | None. | |

### *Table 21: Create group use case description*

|  |  |  |
| --- | --- | --- |
| Use Case Name | Leave group | |
| Triggering Event | User wants to leave a group. | |
| Brief Description | Student will leave his group. | |
| Actors | Student. | |
| Related Use Cases | Respond to invitation, create group. | |
| Stakeholders | Student. | |
| Preconditions | The student must be a member in an existing group. | |
| Postconditions | Student will no longer be a part of a group. | |
| Flow of Activities | Actor | System |
| 1. User will select leave group tab 2. User will confirm | 1.1 System will prompt user to confirm  2.1 System will remove user from the group |
| Exception Conditions | None. | |

### *Table 22: Leave group use case description*

|  |  |  |
| --- | --- | --- |
| Use Case Name | Send invitation | |
| Triggering Event | User wants to add members to his group. | |
| Brief Description | Student will send an invitation to a desired student | |
| Actors | Student. | |
| Related Use Cases | Respond to invitation | |
| Stakeholders | Student. | |
| Preconditions | The student must be a leader in an existing group, The recipient shouldn’t be in a group. | |
| Postconditions | None. | |
| Flow of Activities | Actor | System |
| 1. User will choose the student he would like to join his group | * 1. System will send prompt to the other student   2. System will notify the group leader of the response |
| Exception Conditions | The group is full. | |

### *Table 23: Send invitation use case description*

|  |  |  |
| --- | --- | --- |
| Use Case Name | Respond to invitation | |
| Triggering Event | User will want to interact with an invitation | |
| Brief Description | Student will either accept or decline the invitation. | |
| Actors | Student. | |
| Related Use Cases | Send Invitation. | |
| Stakeholders | Student. | |
| Preconditions | None. | |
| Postconditions | Change student status | |
| Flow of Activities | Actor | System |
| 1. User will select a response | * 1. System will inform the sender of the response   2. System will update student status |
| Exception Conditions | Student not responding in the allotted time. | |

### *Table 24: Respond to invitation use case description*

## **4.3 Activity Diagram:**

An activity diagram visually presents a series of actions or flow of control in a system like a flowchart or a data flow diagram, Activity diagrams are often used in business process modeling.

Diagram

Description automatically generated

Figure 10: Authorize students Activity Diagram

Diagram

Description automatically generated

Figure 11: Add proposal Activity Diagram

Diagram

Description automatically generated

Figure 12: Set load Activity Diagram

Diagram

Description automatically generated

Figure 13: Create group Activity Diagram

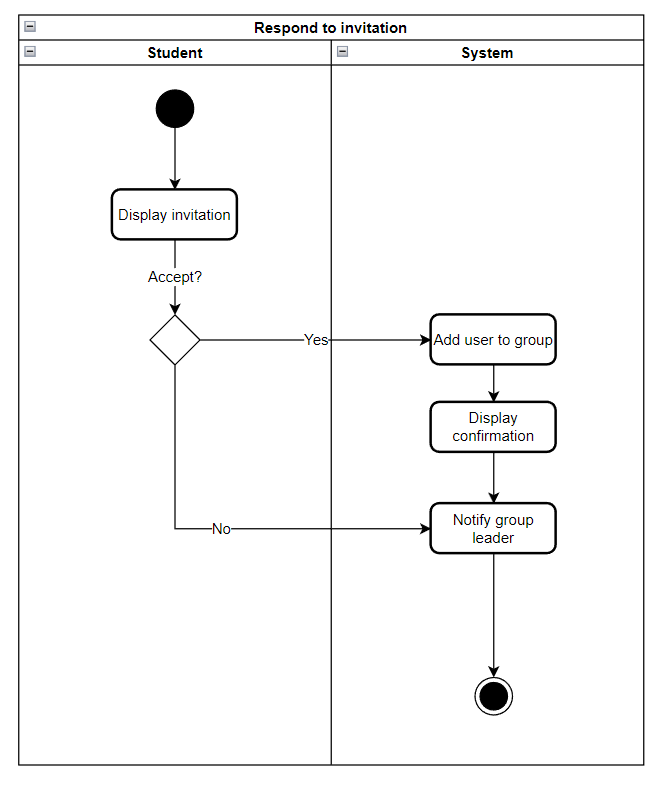


Figure 14: Respond to invitation Activity Diagram

Diagram

Description automatically generated

Figure 15: Leave group Activity Diagram

Diagram

Description automatically generated

Figure 16: Contact group Activity Diagram

# CHAPTER 5: SYSTEM DESIGN

## **5.1 Entity Relationship Diagram:**

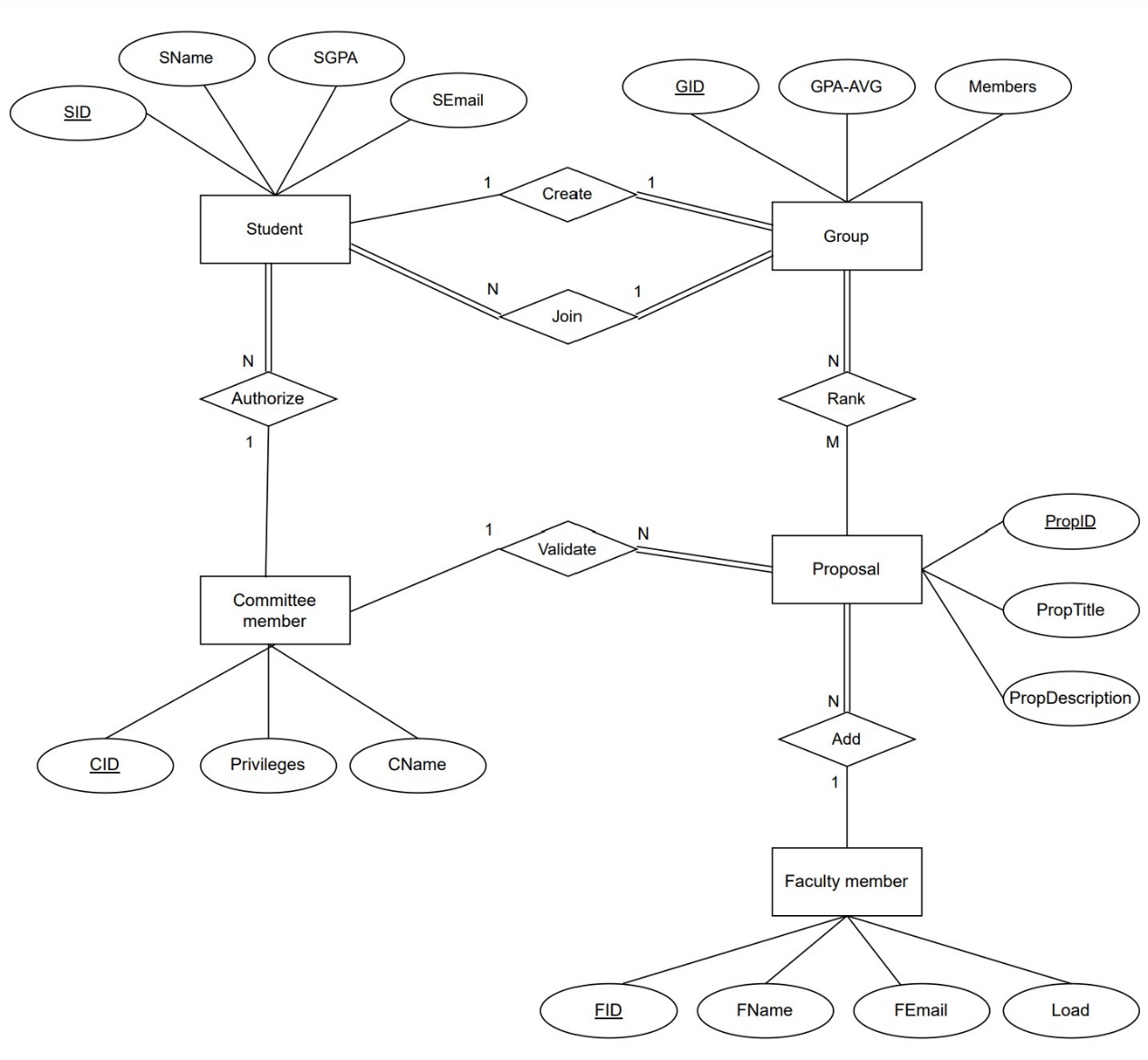


Figure 17: Entity Relationship Diagram

**5.2 Relational Schema:**

A picture containing chart

Description automatically generated

Figure 18: Relational Schema

## **5.3 Sequence Diagram:**

Sequence Diagrams are interaction diagrams that detail how operations are carried out, Sequence Diagrams are time focus, and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

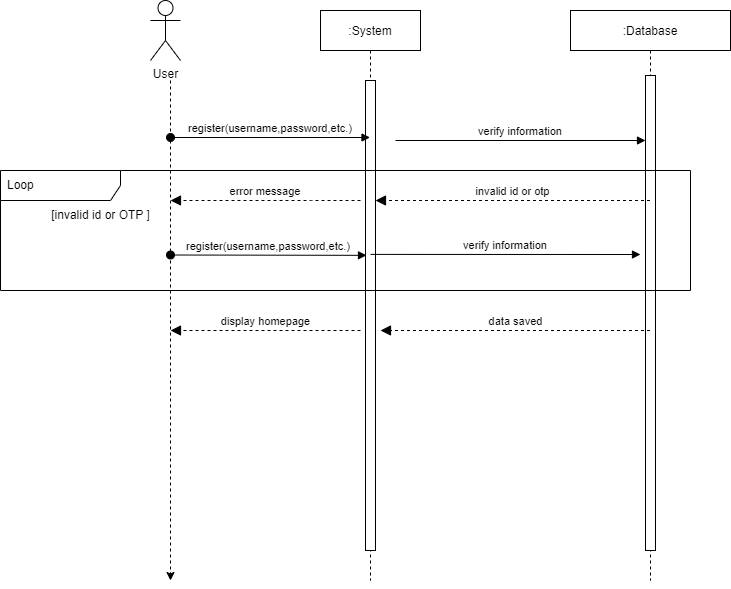


Figure 19: Registration Sequence Diagram

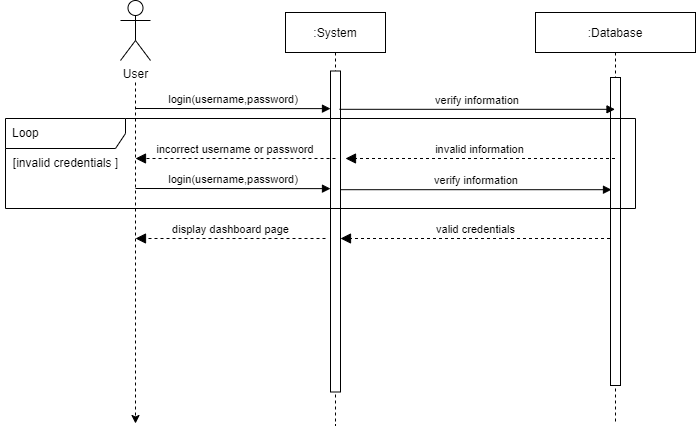


Figure 20: login

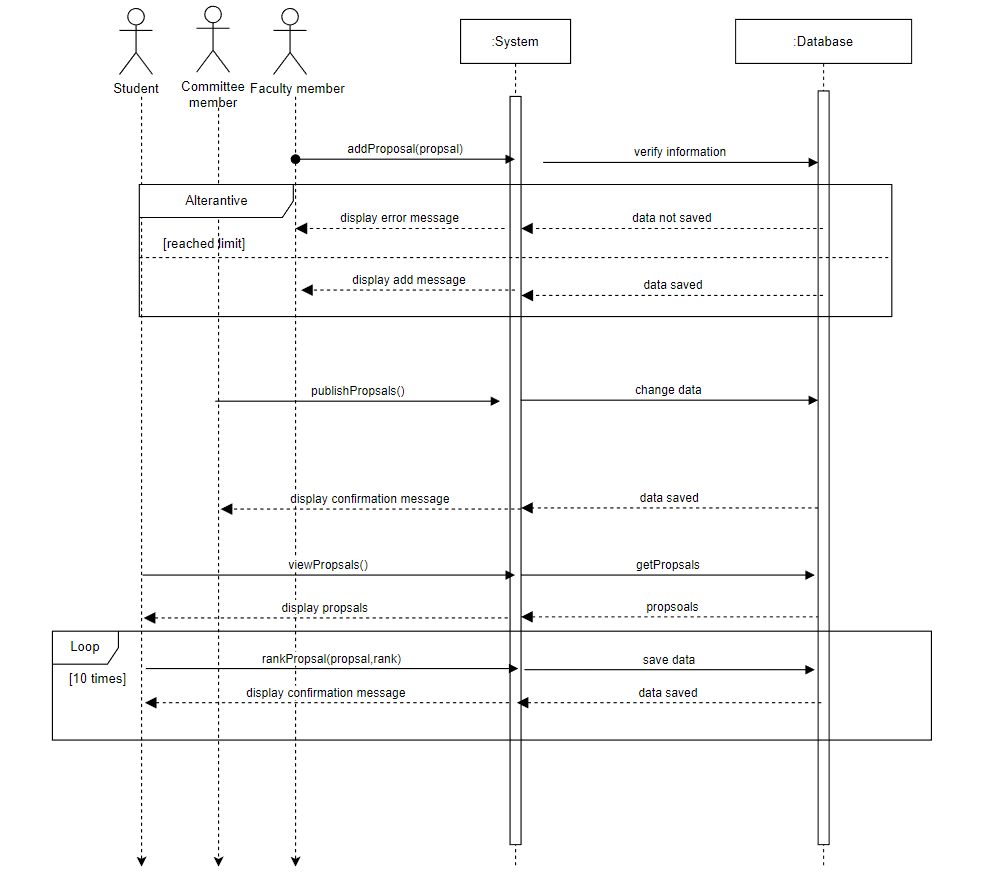


Figure 21: Proposal addition Sequence Diagram

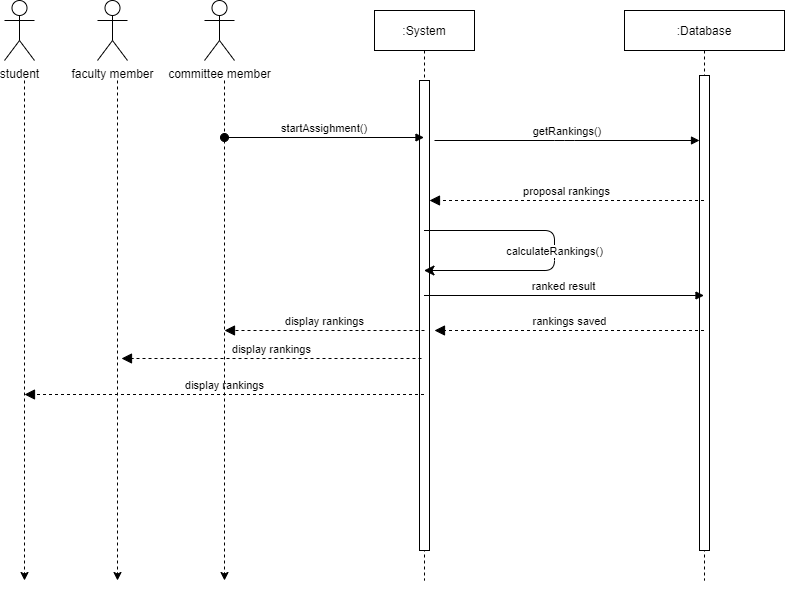


Figure 22: Assigning Proposals Sequence Diagram

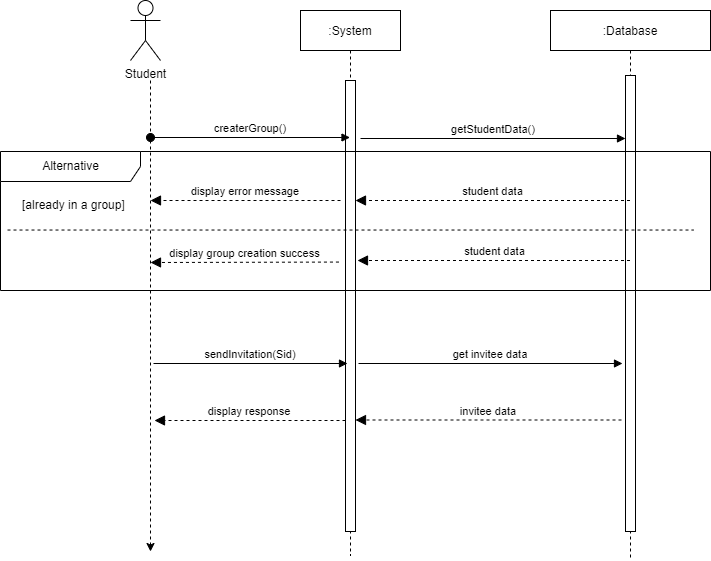


Figure 23: Group interactions Sequence Diagram

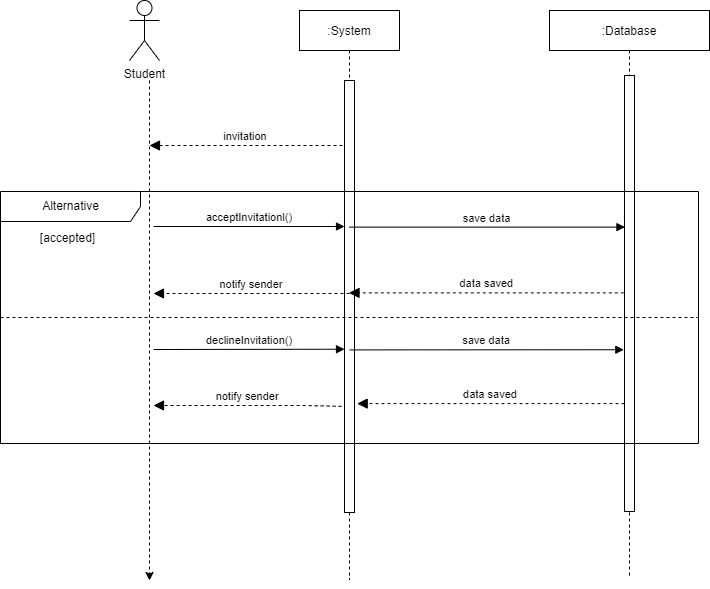
**

Figure 24: Invitation Response

## **5.4 Class Diagram:**

The class diagram is static, it represents the static view of an application, the class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

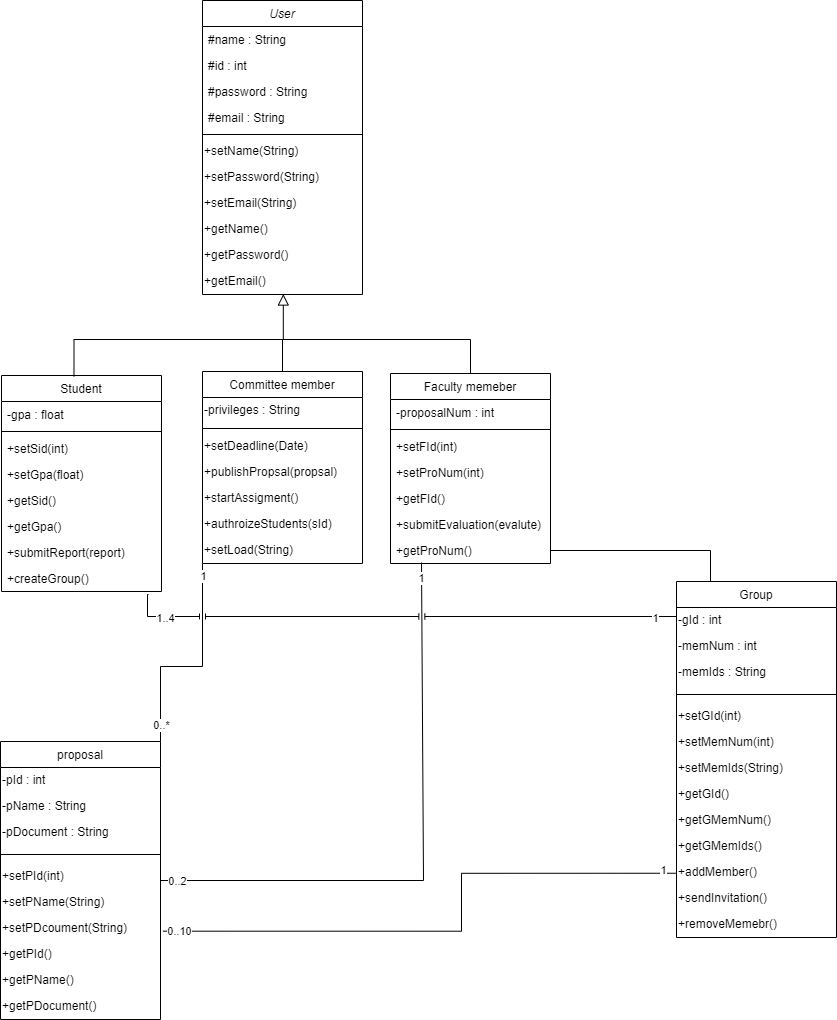


Figure 25: Class Diagram

# CHAPTER 6:

CONCLUSION

## **6.1 Social Impact:**

Our project would make communications between the committee, faculty members, and students much easier than before, and allow students to pair up more effectively.

## **6.2 Ethical Impact:**

Protecting the rights of the committee members, faculty members and students by providing a safe website for the process of registration and log-in, ensuring the information they provide about them is not used for anything unethical.

## **6.3 Legal Impact:**

This website is compatible with the bylaws and regulations of the College of Computer and Information Sciences, as well as the university.

## **6.4 Global Impact:**

After we apply and run our website on the College of Computer and Information Sciences, we hope to transfer this experience to the level of universities in Saudi Arabia, and then to all universities in the world, and that will create a great impact around the world.

## **6.5 Security Impact:**

No person will be allowed to access another user's information and its credentials and data linked to the user account, whether it was a faculty member or a student, committee members will have more privileges than the other users, but we will keep logs of every transaction to ensure security on the website.

## **6.6 Conclusion:**

In this part of the project, we have studied the requirements, analyzed, and designed the system. Going forward we hope that our vision can be realized in the second half of the project

(Implementation phase), to achieve that and meet the committee's time expectations we have decided to go with python's framework flask, it's lightweight, scalable, and uses a single page approach that is known to be very fast and responsive, we are looking forward to the next part of this project, we hope it offers us unique challenges and more opportunities to learn and advance our skillset.

## **6.7 Reference List:**

1. <https://business.adobe.com/blog/basics/waterfall>
2. <https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban>
3. <https://www.rbu-admit.edu.sa/>
4. <https://noor.moe.gov.sa/>
5. <https://dev.to/amigosmaker/what-is-flask-used-for-2do5>
6. <https://www.justinmind.com/blog/single-page-vs-multi-page-websites-design-battle/>