Faculty of Informatics Engineering

Department of Software Engineering

Managing student projects system based on cloud computing using scrum methodology

A junior project report - submitted to complete the requirements

for obtaining a Bachelor's Degree in informatics engineering

Prepared by

Raghad Alhossny Kasem Alkelani

Supervised by

Dr. Mouhib Alnoukari

Eng. Anas Abdulaziz

2023-2024

SUPERVISION CERTIFICATION

ABSTRACT

At the Faculty of Computer and Information Engineering in the Syrian Private University, registering student projects (junior, senior 1, senior 2) is currently done manually and can be unorganized at times. Students who want to register for projects must also be present with their project supervisors, before the official start of each semester to sign the project form and obtain approval.

This project aims to automate the different aspects of the project registration process, combine them into a consistent and integrated software system, and manage the process more accurately. This will make it easier for students, supervisors, and administrators responsible for project registration.

Additionally, it demonstrates the significance of cloud computing and its diverse and extensive services and shows the impact of using cloud computing on building software systems with strong features in terms of performance, scalability, and more.

Finally, illustrates the development of software systems according to a flexible and comprehensive methodology. The Scrum development methodology will be used, which works iteratively and focuses on rapid product delivery and continuous improvement. This helps develop software systems that significantly meet desired requirements, by providing prompt feedback and utilizing observations to make cost-effective modifications. This is in contrast to various traditional development methodologies such as waterfall.

Ultimately, our goal is to develop a consistent software system that provides services and meets specified requirements using important methodologies and technologies.

ملخص

في الجامعة السورية الخاصة – كلية هندسة الحاسوب والمعلوماتية تكون عملية تسجيل المشاريع الطلابية (فصلي – تخرج 1 -تخرج 2) ورقية ويسبقها خطوات عديدة لإتمام العملية، تتم بشكل غير منظم احياناً، بالإضافة لأنها تتطلب حضور الطلاب الذين يرغبون بتسجيل مشاريع مختلفة في نفس الوقت مع أعضاء الهيئة التدريسية "مشرفي المشاريع" لتوقيع استمارة المشروع والموافقة عليه في الجامعة ومن ثم تسجيله، وذلك قبل الدوام الرسمي لكل فصل دراسي.

نهدف في هذا المشروع اولاً أتمتة الجوانب المختلفة لعملية تسجيل المشاريع في كلية الهندسة المعلوماتية - الجامعة السورية الخاصة، وجمعها في نظام برمجي متسق ومتكامل، لتنظيم وادارة العملية بشكل أكثر دقة، وتسهيل خطواتها على الطلاب والمشرفين والإداريين القائمين على تسجيل المشاريع.

بالاضافة الى توضيح اهمية الحوسبة السحابية وخدماتها الواسعة والمختلفة في الوقت الحاضر، ومدى تأثير استخدام احدى هذه الخدمات على بناء أنظمة برمجية تتمتع بصفات قوية من اداء وقابلية توسع وغيرها.

واخيرا تقديم مثال عملي على كيفية تطوير الأنظمة البرمجية وفق منهجية تطوير مرنة وشمولية، تعمل بنمط تكراري وتركز على التسليم السريع للمنتجات والتحسين المستمر، وهي منهجية التطوير Scrum. التي بدورها تفيد في تطوير انظمة برمجية تحقق بشكل كبير المتطلبات المرادة، وذلك من خلال التسليم السريع لاخذ تغذية مرتدة عن المشروع وملاحظات تستخدم لتعديل المشروع تعديل غير مكلف بطريقة مبالغة ليصبح أكثر كفاءة وتحقيقاً لمتطلباته، وذلك ما لا تقدمة منهجيات التطوير التقليدية المختلفة من مثل waterfall.

لنحصل في النهاية على نظام برمجي متسق يقدم الخدمات ويحقق المتطلبات المحدده له وفق استخدام منهجيات وتقنيات مهمة.

Table of Contents

List of Tables

List of Figures

List of abbreviations

Abbreviation	Definition
IT	Information Technology
UML	Unified Modeling Language
SPU	Syrian Private University
IaaS	Infrastructure as a Server.
SaaS	Software as a Service.
PaaS	Platform as a Service.
NoSQL	Not only Structured Query Language.
JSON	JavaScript Object Notation
APIs	Application Programming Interfaces.
RTM	Requirement Traceability Matrix
DRF	Django Rest Framework
pk	Primary key

Chapter1 Introduction

1. Introduction:

In this chapter, we will introduce our project, discussing the main issues and reasons for building this system. We will also explain the objectives and goals we aim to accomplish with this system. Finally, we will provide an overview of the main concepts and tools used and required in this project.

2. Problem Definition:

The Computer and Informatics Engineering faculty at the Syrian Private University needs to improve the process of registering student projects. Automation is the solution to this problem, which involves using technology and software to carry out tasks with minimal human intervention. The current process for project registration at the university involves communication and collaboration between various entities and members. However, some of the steps are still paper-based which can cause delays.

The first step is for supervisors to suggest projects for students. These suggestions are then reviewed and approved by the manager before being presented to the students. Once the students have reviewed the suggestions, they select the project that best suits them and submit a request for registration with their team members. To ensure that the supervisor is present at the university, all students must submit their registration requests on a specific day to obtain the supervisor's signature and approval. After this, the students will submit their requests to the responsible employee. The responsible employee will then check if the students have fulfilled some

faculty roles before approving the registration. These roles include finishing the “application course” and having more or equal to 100 completed hours. The team members must also have hours close to each other, if they meet those conditions the project will then register to the university system “Learnata”.

Automating and organizing this process will make it more efficient and effective for everyone involved.

3. Project objectives

Our project aims to create a system that will handle student project registration. We want to make the process more efficient for everyone involved, including managers, supervisors, students, and responsible employees.

In addition to this, we want to highlight the importance of cloud computing and how applying its services can positively impact the systems, by using the cloud storage service “Realtime Database Service” provided by the Firebase from Google.

Lastly, we aim to demonstrate how to develop a software system using the scrum methodology. This approach allows for continuous improvement from one sprint to the next by accepting changes and delivering demos for feedback. By applying scrum techniques, we will significantly improve the system.

4. Concepts

4.1. Cloud computing:

Cloud computing is the delivery of computing services - like storage, databases, networking, software, analytics, and intelligence - over the internet ("the cloud") to offer faster innovation, flexible resources, and economies of scale. You typically pay only for cloud services you use, helping you lower your operating costs, run your infrastructure more efficiently, and scale as your business needs change.

Cloud computing is a big shift from the traditional way businesses think about IT resources, common reasons organizations are turning to cloud computing services:

- Cost: moving to the cloud helps companies optimize IT costs. This is because cloud computing eliminates the capital expense of buying hardware and software and setting up and running onsite data centers.

- Reliability: cloud computing makes data backup, disaster recovery, and business continuity easier and less expensive because data can be mirrored at multiple redundant sites on the cloud provider's network.

- Security: many cloud providers offer a broad set of policies, technologies, and controls that strengthen your security

posture overall, helping protect your data, apps, and infrastructure from potential threats.

- Performance: the biggest cloud computing services run on a worldwide network of secure data centers, regularly upgraded to the latest generation of fast and efficient computing hardware. This offers several benefits over a single corporate data center, including reduced network latency for applications and greater economies of scale.

Types of cloud services:

IaaS

- The most basic category of cloud computing services. infrastructure as a service (IaaS) works by renting IT infrastructure—servers and virtual machines (VMs), storage, networks, and operating systems from a cloud provider on a pay-as-you-go basis.

PaaS

- Platform as a service refers to cloud computing services that supply an on-demand environment for developing, testing, delivering, and managing software applications.

SaaS

- Software as a service is a method for delivering software applications over the internet, on-demand, and typically on a subscription basis. With SaaS, cloud providers host and manage the software application and underlying infrastructure, and users connect to the application over the internet, usually with a web browser.

4.2. Firebase:

Firebase is a cloud-based platform developed by Google. It utilizes cloud technology to provide a range of services and tools for building web and mobile applications. The cloud aspect of Firebase refers to the utilization of cloud infrastructure, where the services are hosted and managed on remote servers. This allows developers to access and leverage these services over the internet without the need to establish and maintain their infrastructure.

4.3. Scrum methodology:

Scrum is an agile project management framework that helps teams structure and manage their work through a set of values, principles, and practices. The definition of scrum is based on empiricism and lean thinking. Empiricism says that knowledge comes from experience and that decisions are made based on what is observed. Lean thinking reduces waste and focuses on essentials.

Agile vs. Scrum: scrum is a framework for getting work done, whereas agile is a philosophy. The agile philosophy centers around continuous incremental improvement through small and frequent releases. You can't really "go agile", as it takes dedication from the whole team to change the way they think about delivering value to your customers. But you can use a framework like Scrum to help you start thinking that way and to practice building agile principles into your everyday communication and work.

Scrum sprints:

With Scrum, a product is built in a series of iterations called sprints that break down big, complex projects into bite-sized pieces. A sprint is a short, time-boxed period when a scrum team works to complete a set amount of work. Sprints are at the very heart of scrum and agile methodologies

Scrum artifacts: Scrum artifacts are important information used by the scrum team that helps define the product and what work to be done to create the product.

- Product Backlog is the primary list of work that needs to get done and maintained by the product owner or product manager. This is a dynamic list of features, requirements, enhancements, and fixes that act as the input for the sprint backlog.

- Sprint Backlog is the list of items, user stories, or bug fixes, selected by the development team for implementation in the current sprint cycle. Before each sprint, in the sprint planning meeting (which we'll discuss later in the article) the team chooses which items it will work on for the sprint from the product backlog.

- Increment (or Sprint Goal) is the usable end-product from a sprint.

Chapter2 Project Management

1. Introduction:

In this chapter, we will introduce the proposed system of the project, its outlines, the project plan we are working on, and the system requirements. We will also discuss how we are gathering these requirements.

2. Proposed System:

To address the problems outlined in the previous chapter and to achieve the system objectives, we will develop a system based on a web application. This system will be linked to a cloud database through the use of Firebase Realtime database services. The development and management of this project will be based on the principles of the Scrum methodology, enabling us to achieve a better improvement process. For the backend development, we will use the Django framework for Python programming language and for the frontend development, we will use the React framework.

3. Project planning:

Gantt chart: a Gantt chart, commonly used in project management, is one of the most popular and useful ways of showing activities (tasks or events) displayed against time.

Figure 1 Gantt chart

As shown in the diagram, the first step is to collect all the requirements for the project, which will represent the “project backlog”. After gathering the requirements, we will select a few of them based on their priority, and develop them in the next sprint. This selection of requirements is known as the "sprint backlog". During each sprint, we will carry out analysis, design,

implementation, and testing. Once we have completed a sprint, we will plan for the next one, taking into consideration any bugs or improvements that were identified after the previous sprint.

4. Requirements Elicitation:

We have interviewed the employee who is responsible for the registration process at the Faculty of Computer and Informatics Engineering "Mohammed Othman", the purpose of the interview was to identify the process steps in the university, the requirements of the system, and the necessary features that are needed for the system to be effective.

As a result of this interview, we have identified the functional requirements list that needs to be established for this project to achieve the goals and objectives we are aiming for.

Requirements Database "Project Backlog":

Table 1 Requirements database

Req-ID	Title	Description	Type	Priority
Req-01	the system must allow the admin to make an account for the supervisor, manager, and employee by a unique ID and password.	The system must have student data from the university.	Functional	3
Req-02	the system must allow the students to make an account by	Sign in	Functional	2

	their university ID (unique account).			
Req-03	the system must be able to check if a student belongs to the university by comparing some entered data with the student data.	The system must have student data from the university (university id, first name, last name, GPY, completed hours, and some courses informations).	Functional	2
Req-04	The system must allow users to log in to their accounts with an ID and password	Every user will use his university ID	functional	1
Req-05	The system must allow users to change their profile photo		functional	1
Req-06	The system must allow users to change their account password	Must be a strong password	functional	1
Req-07	The system must allow a supervisor to add a project suggestion.	By completing the project form (title, description, goal, department)	functional	1
Req-08	The system must allow a supervisor to edit or delete suggestion.	Before it gets approved or rejected by the manager.	functional	1
Req-09	The system must allow users who request to track their requests state.		Functional	1
Req-10	The system must be able to inform the manager of all projects suggestions.	To either accept or reject the suggestions.	functional	1

Req-11	The system must allow the manager to accept or reject project suggestion		functional	1
Req-12	The system must be able to inform a request maker of the response.	As a notification on their accounts	Functional	1
Req-13	The system must be able to display the suggestions list for users.		functional	1
Req-14	The system must be able to display the suggestions list filtered by supervisors or department.		functional	1
Req-15	The system must allow students to request a project.	Students can add other students (team).	functional	2
Req-16	The system must be able to check if a student and a team met the project's registration conditions.	By checking the student's data from the university.	functional	2
Req-17	The system must be able to get the acceptance of all team members for a request.		functional	2
Req-18	The system must be able to inform the supervisor about the requests made for his project suggestions.		functional	2
Req-19	The system must allow a student who request to delete his request	Before it accepted by his team member.	functional	2
Req-20	The system must allow supervisors to either accept or reject a project request.		functional	2

Req-21	The system must inform the employee of the projects that are ready for registration.		functional	2
Req-22	the system must be able to inform the students if their project has been registered.	If the whole process is done and the project now registered on the university system	functional	2
Req-23	The system must be able to display registered project list.	To all users.	functional	2
Req-24	The system must be able to display registered project list filtered by supervisors or departments.		functional	2
Req-25	The system must allow students to make a new project suggestion and send it to a supervisor they choose.	Project doesn't exist in the suggestion list.	functional	3
Req-26	The system must allow the manager to set a supervisor as the head of the evaluation process for a specific department.	Set new responsibilities for a supervisor.	functional	3
Req-27	The system must allow the manager and the head of the evaluation process to set an advertisement.	To be shown for all users.	functional	3
RQ-28	the system must allow the head of the evaluation team and the manager to upload files with an advertisement.	To be Shown for all users.	Functional	3
RQ-29	The system must log all the events that occur on the system.	Log the event with the one who made it, and	functional	3

		display it to the system admin.		
Rq-31	The system must be user-friendly.	It must be easy to use and understand	Non-functional	3
Req-32	the system must be secure.	determine a level of complexity for passwords, encrypt any password before storing it	Non-functional	3

Chapter 3 system analysis, design and implementation using scrum methodology

1. Introduction:

In this chapter, we will introduce how to develop software system using scrum methodology.

Scrum, as an agile framework, advocates for iterative and incremental development, allowing for the rapid delivery. This approach contrasts with traditional waterfall methodologies, where each phase in the SDLC is typically executed sequentially and with limited interaction between phases

2. Sprint #1

Sprint#1 analysis:

In this section, we will introduce the analytical study for the first sprint using the needed UML diagrams.

1. Sprint Backlog:

   The Requirement list we will complete for this sprint:

   - ✓ Req-01: The system must allow users to log in to their accounts with an ID and password.
   - ✓ Req-02: The system must allow a supervisor to add a project suggestion.
   - ✓ Req-03: The system must allow a supervisor to edit or delete suggestion.

- ✓ Req-04: The system must allow users who request to track their requests state.
- ✓ Req-05: The system must be able to inform the manager of all projects suggestions
- ✓ Req-06: The system must allow the manager to accept or reject project suggestion
- ✓ Req-07: The system must be able to inform a request maker of the response.
- ✓ Req-8: The system must be able to display the suggestions list for users.
- ✓ Req-9: The system must be able to display the suggestions list filtered by supervisors or department.
- ✓ Req-10: The system must allow users to change their profile photo.
- ✓ Req-11: The system must allow users to change their account password.

2. Initial Requirements traceability Matrix – Sprint1:

   a document that demonstrates the relationship between requirements and other artifacts. It's used to prove that requirements have been fulfilled. And it typically documents how requirements and connect it with each phase (analysis, design, implementation, testing).

Req-id	Title	Analysis	Detailed design	coding	App user interfaces	Test cases
Req-01	The system must allow users to log in to their accounts with an ID and password.					
Req-02	The system must allow a supervisor to add a project suggestion.					
Req-03	The system must allow a supervisor to edit or delete suggestion.					
Req-04	the system must allow users who request to track their requests state.					
Req-05	The system must be able to inform the manager of all projects suggestions					
Req-06	The system must be able to inform the manager of all projects suggestions					
Req-07	The system must be able to inform a request maker of the response.					
Req-08	The system must be able to display the suggestions list for users.					
Req-09	The system must be able to display the suggestions list					

	filtered by supervisors or department.					
Req-10	The system must allow users to change their profile photo.					
Req-11	The system must allow users to change their account password.					

3. Requirements Modeling:
    - Use Case Diagram: use-case diagrams model the behavior of a system and help to capture the requirements of the system.

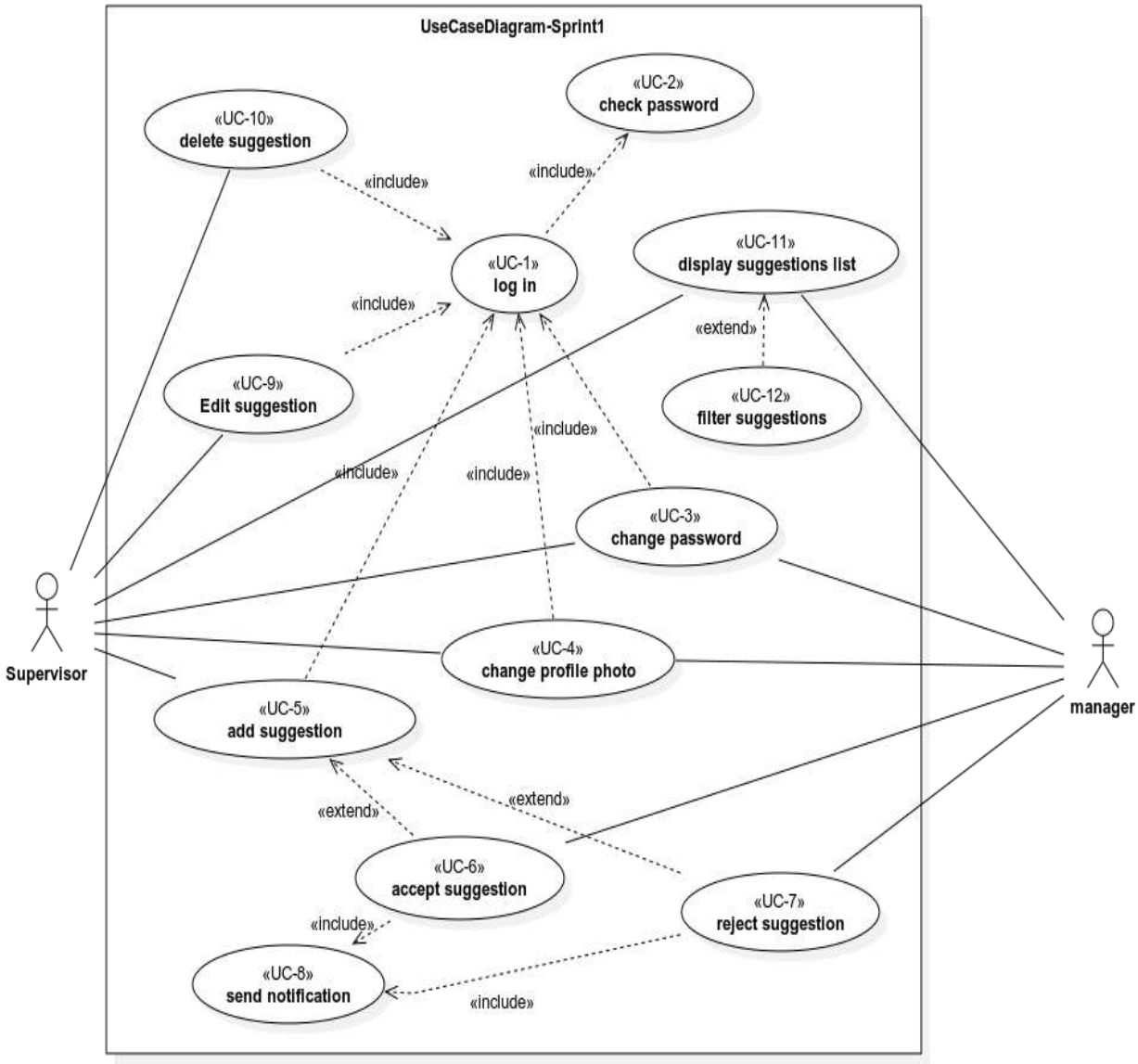


Figure 2 sprint#1 use case diagram

- Use Case Specification:

Table 2 sprint#1 log in specification

Use case title:	Log in
Participating actors:	initiated by all users
The flow of events:	1. The User first will enter the website. 2. The system will show the login form. 3. The user will enter his ID and password(form). 4. The system checks the entered ID and searches for the account. 5. And the system will validate the entered password with the stored password for this user. 6. If the password is correct the web app will open to the main page for the user account.
Alternative flow:	first alternative flow-A1: start in step 4 in the main flow: 5. if the id is not founded. 6. the system will show an error message for ID to the user and ask him to reenter it. and it will back to step 5 in the main flow.
	Second alternative flow-A2: start at the step 5 in the main flow: 6. if the password is not correct. 7. the system will show an error message for the password and ask the user to reenter it. and the flow will go back to step 6 in the main flow.
Entry condition	user has an account on the system.
Exit conditions	the user enters the system

Table 3 sprint#1 change password specification

Use case name:	change password.
Participating Actors:	initiated by all users.
The flow of events:	1. The actor selects the "Change Password" option from the user interface 2. The system will display a form containing ID, old password, new password, and confirm new password fields. 3. The actor will fill in the required fields. 4. The system will check if the ID matches the actor's account ID. 5. The system validates the entered data: - Verifies that the ID corresponds to the actor's account. - Verifies that the old password matches the current password for the account. - Verifies that the new password meets the system's password requirements (construct of 8 characters and contain letters and numbers). - Verifies that the new password and the confirmed password match. 6. If the data is valid, the system updates the actor's password with the new password. 7. The system displays a success message indicating that the password has been changed.
Entry condition	user has an account on the system.
Exit conditions	password changed.

Table 4 sprint#1 add suggestion specification

Use case name:	Add suggestion
Participating Actors:	initiated by: supervisor. manager
The flow of events:	1. the supervisor choice to add new project suggestion. 2. The system displays the project form. 3. The supervisor will complete all fields in the project form, and choice apply. 4. The system check if all the field completed, then send the suggestion to the manager and show message the suggestion applied successfully. 5. If the manager choice to accept the suggestion: • The system will send the response to the supervisor as a notification. • The system will add the suggestion to the suggestions list.
Exception flows:	First exception flow: start at step 4 from the main flow, if the supervisor choice to reject the suggestion: • The system will send the response for the supervisor as a notification. • And the system will delete the suggestion from the requests, and the use case will fail.
Entry condition	Supervisor and manager had log in the system
Exit conditions	the suggestion request is handled.

Table 5 sprint#1 edit suggestion specification

Use case name:	Edit suggestion
Participating Actors:	initiated by: supervisor
The flow of events:	1. The supervisor chooses a suggestion from his suggestions page, that manager still don't accept or reject it. 2. The system gives two options. 3. The supervisor choice to edit this suggestion. 4. The system will display the project form to the supervisor. 5. The supervisor will recomplete the form. 6. The system check if all field are completed and resend the new suggestion to the manager.
Entry condition	The supervisor already had suggestions, that doesn't accept or rejected yet.
Exit conditions	The suggestion information had updated.

Table 6 sprint#1 delete suggestion specification

Use case name:	Delete suggestion
Participating Actors:	initiated by: supervisor
The flow of events:	1. The supervisor chose a suggestion from his suggestion list. 2. The system shows two options. 3. The supervisor chose to delete suggestion. 4. The system will ask the supervisor to confirm the decision he made. 5. The supervisor chose yes. 6. The system will delete the suggestion from the supervisor and manager.
Exception flows:	First exception flow-E1: start at the step 4 in the main flow: 5. if the supervisor chooses to change his decision.

	6. the system will close and exit the delete project interface, and the use case will fail.
Entry condition	The supervisor log in and already had suggestions, that doesn't accept or reject yet.
Exit conditions	The suggestion has deleted.

Table 7 sprint#1 change profile photo specification

Use case title:	change profile photo
Participating users:	initiated by all users.
The flow of events:	1. The user will choose to change his profile photo. 2. The system will ask the user to upload a photo from his device 3. The user will upload a photo 4. The system will check the photo format. 5. if it fits the system format 6. The system will replace the old photo with the new photo and send a successful message.
Alternative flow:	first alternative flow-A1: start in the step 4: 5. if the format does not match the system-determined format the system will send an error message to the user and ask him to upload another image with the correct format. And the flow will go back to step 6 in the main flow.
Entry condition:	The user log in
Exit conditions:	profile photo changed.

Table 8 sprint#1 display suggestion list specification

Use case name	display suggestions list
Participating actors	initiated by all users.
Flow of events	1. The actor selects the "Display suggestions List" option from the user interface. 2. The system will display the suggestions list. 3. If the user chooses to filter the list. 4. The system will show two options. 5. If the user chooses "by department". 6. The system will filter the list and display it. 7. If the user chooses "by supervisor". 8. The system will filter the list and display it.
Entry conditions:	user log in
Exit conditions:	Suggestions list displayed

- Activity Diagram: a type of UML flowchart that shows the flow from one activity to another in process.

  - Use case – login:

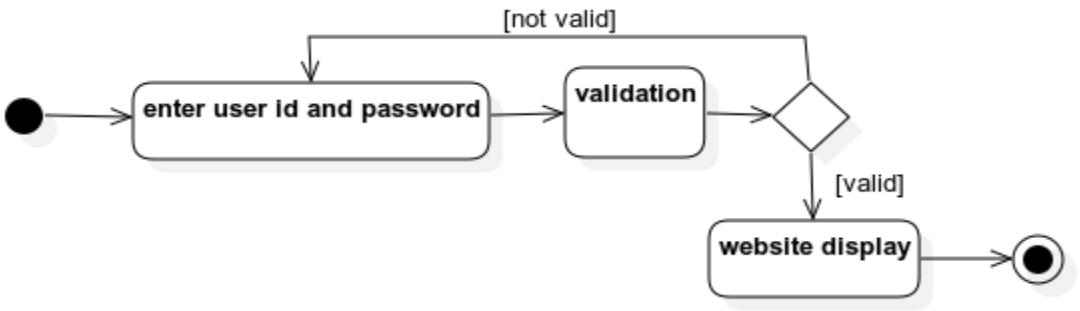


Figure 3 sprint#1 login activity

- Use case – change password:

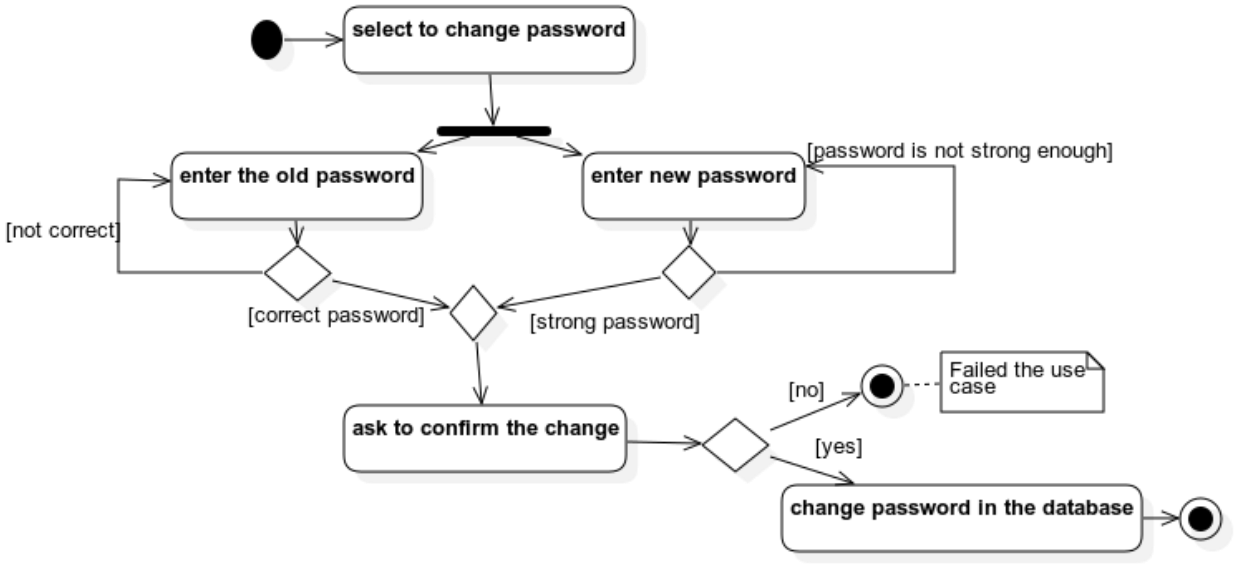


Figure 4 sprint#1 change password activity

- Use case – add suggestion:

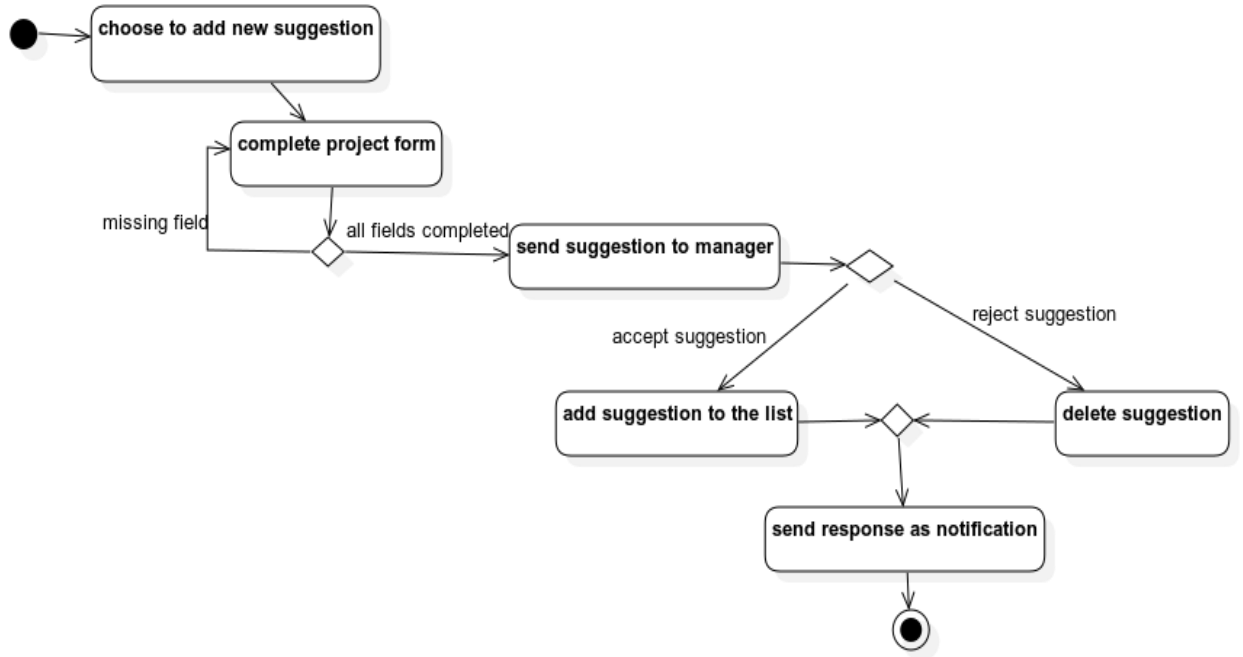


Figure 5 sprint#1 add suggestion activity

- Use case – delete project suggestion:

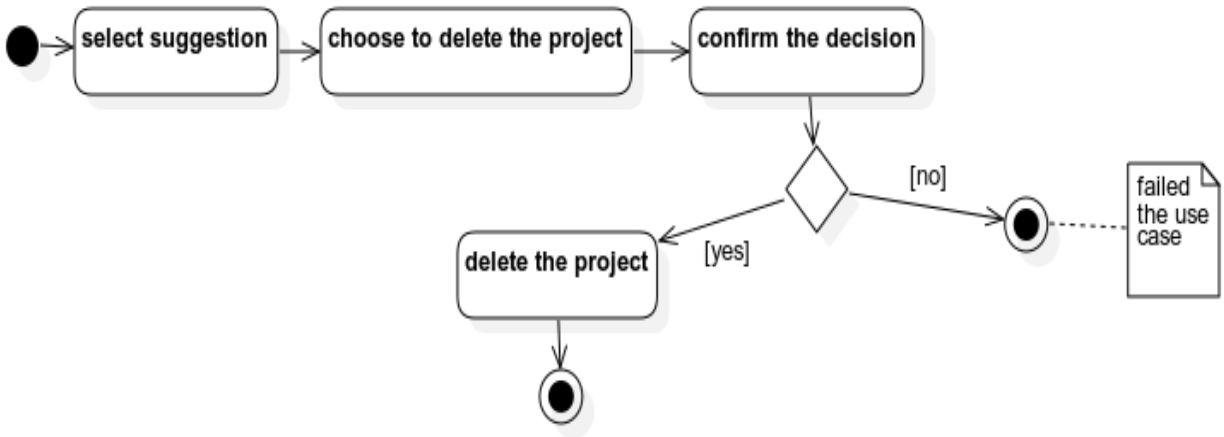


Figure 6 sprint#1 delete project suggestion activity

- Use case – edit project suggestion:

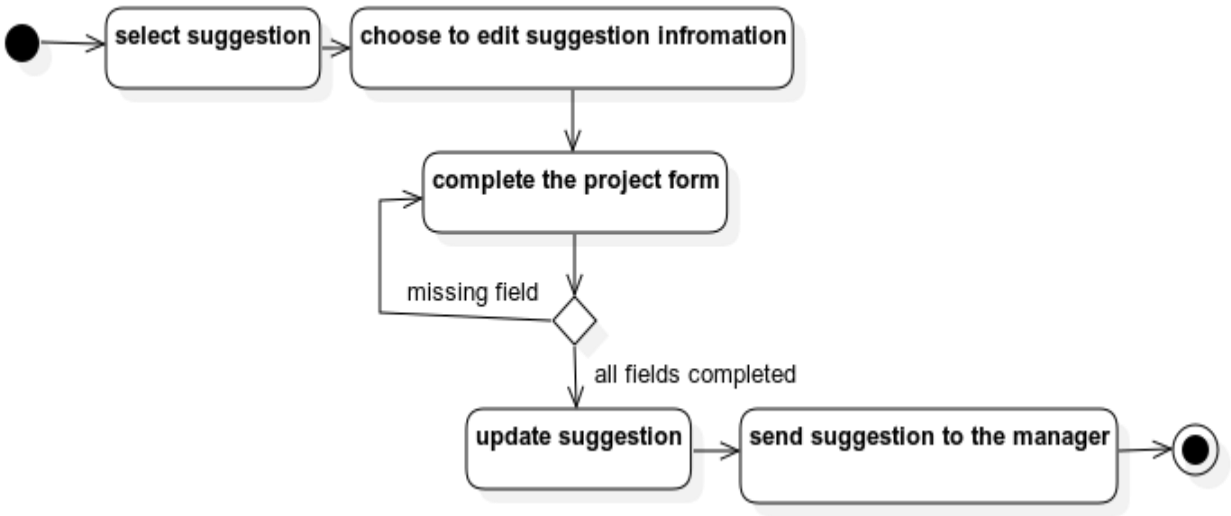


Figure 7 sprint#1 edit project suggestion activity

- Use case - Change profile photo:

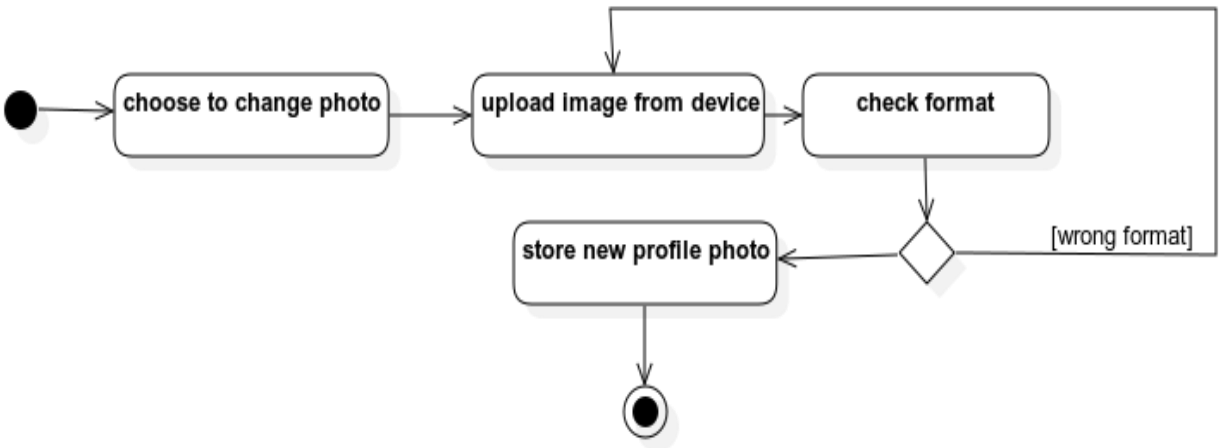


Figure 8 sprint#1 change profile photo activity

- Use case – display suggestions list:

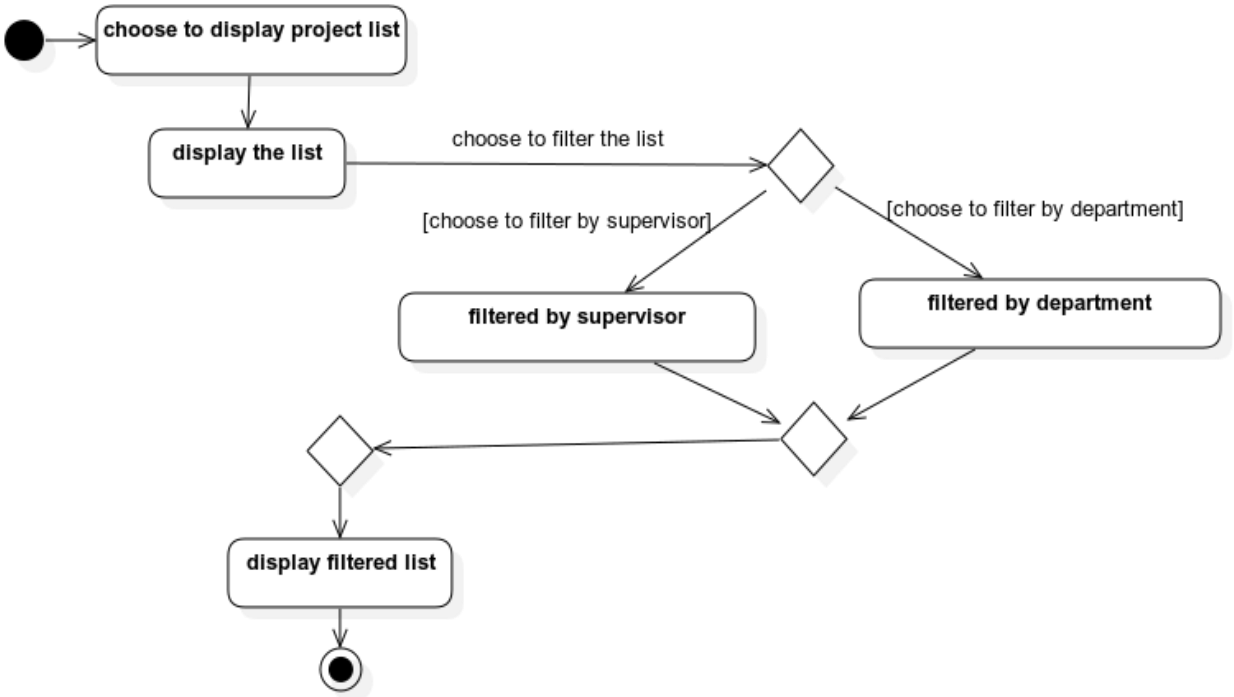


Figure 9 sprint#1 display suggestion list activity

- Sequence Diagram: a sequence diagram is a UML diagram that illustrates the sequence of messages between objects in an interaction.

  - Use case – login:

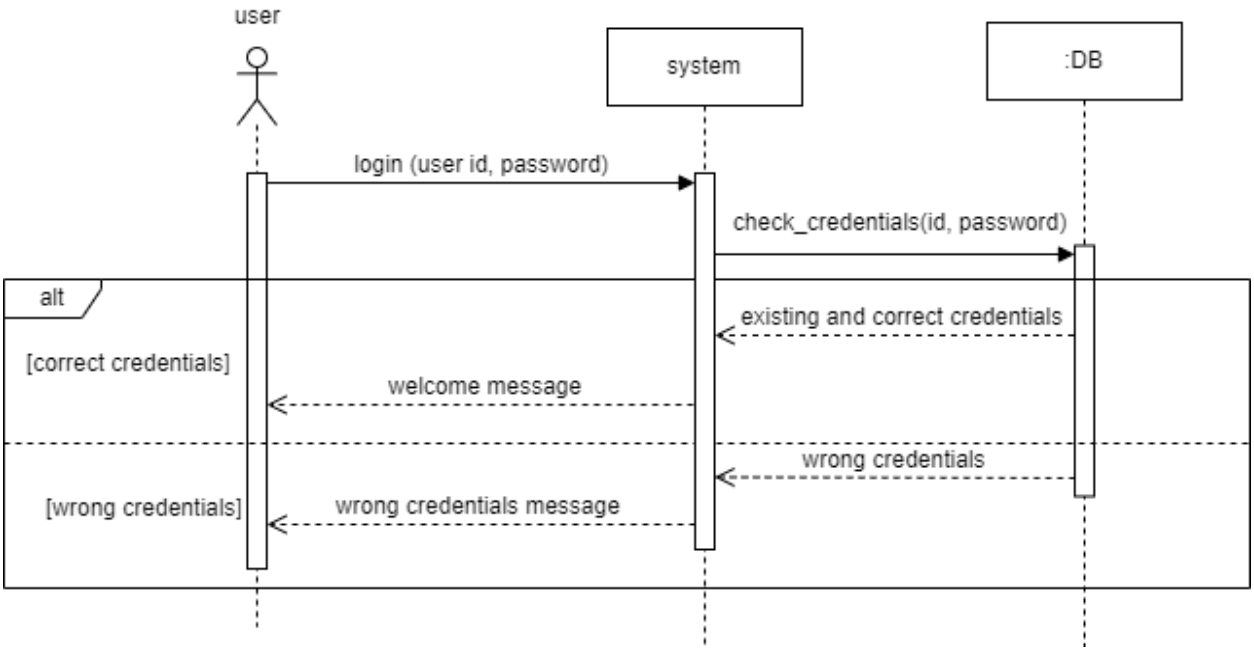


Figure 10 sprint#1 login sequence

  - Use case – display suggestion list:

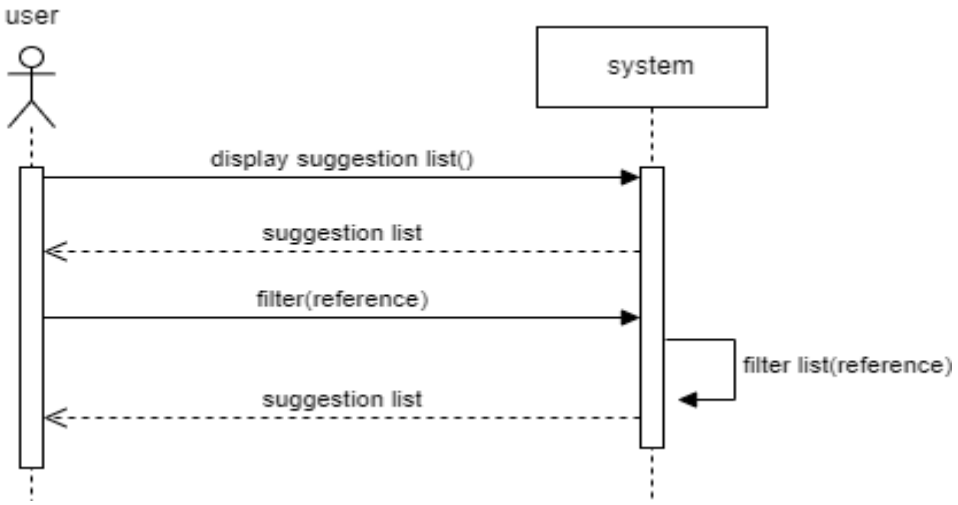


Figure 11 sprint#1 display suggestion list sequence

- Use case – change password:

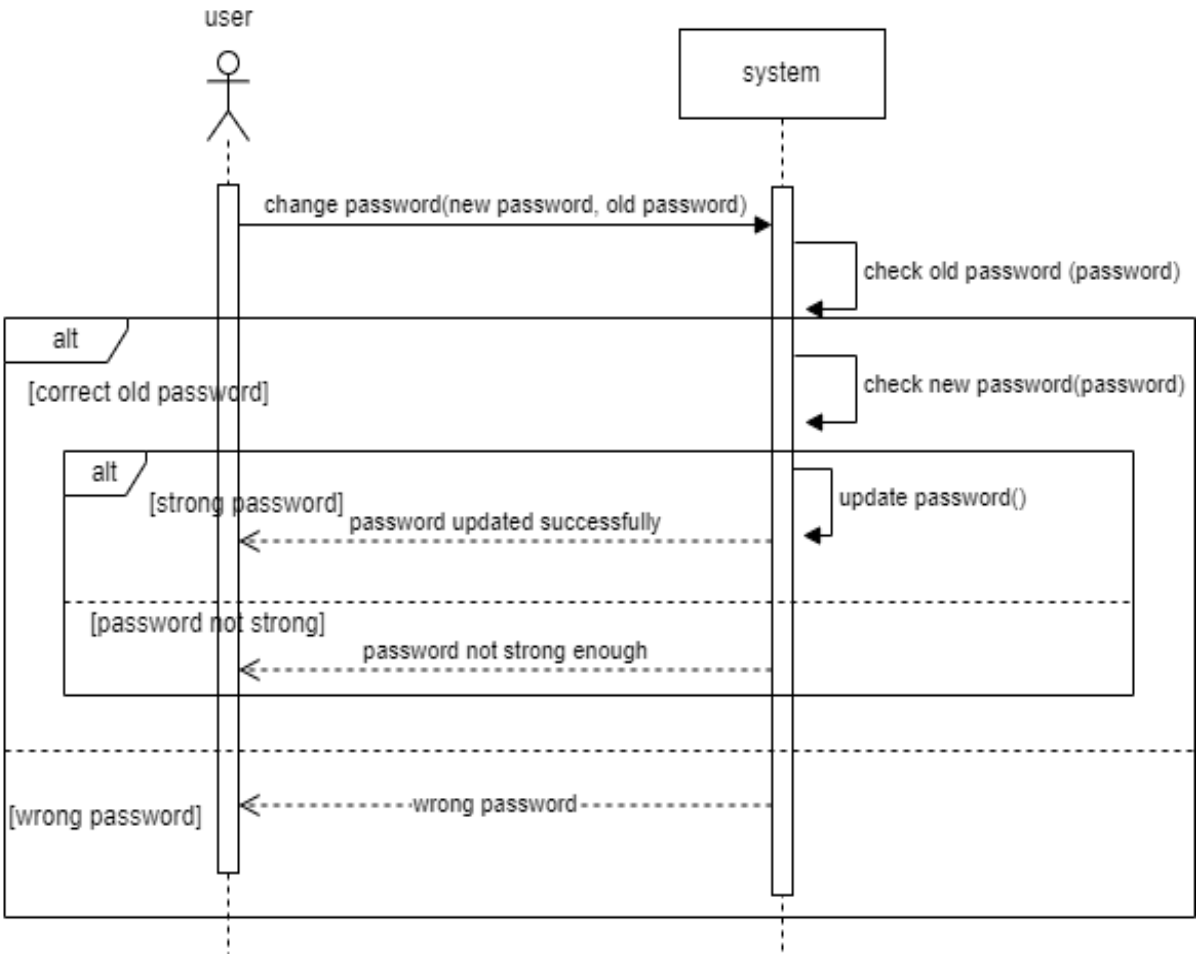


Figure 12 sprint#1 change password sequence

- Use case – edit suggestion:

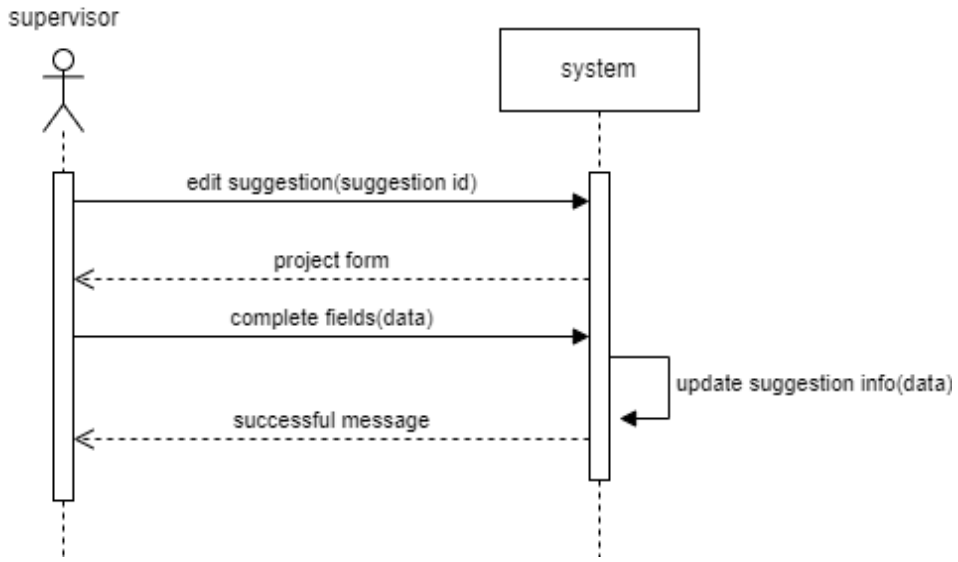


Figure 13 sprint#1 edit suggestion sequence

- Use case – delete suggestion:

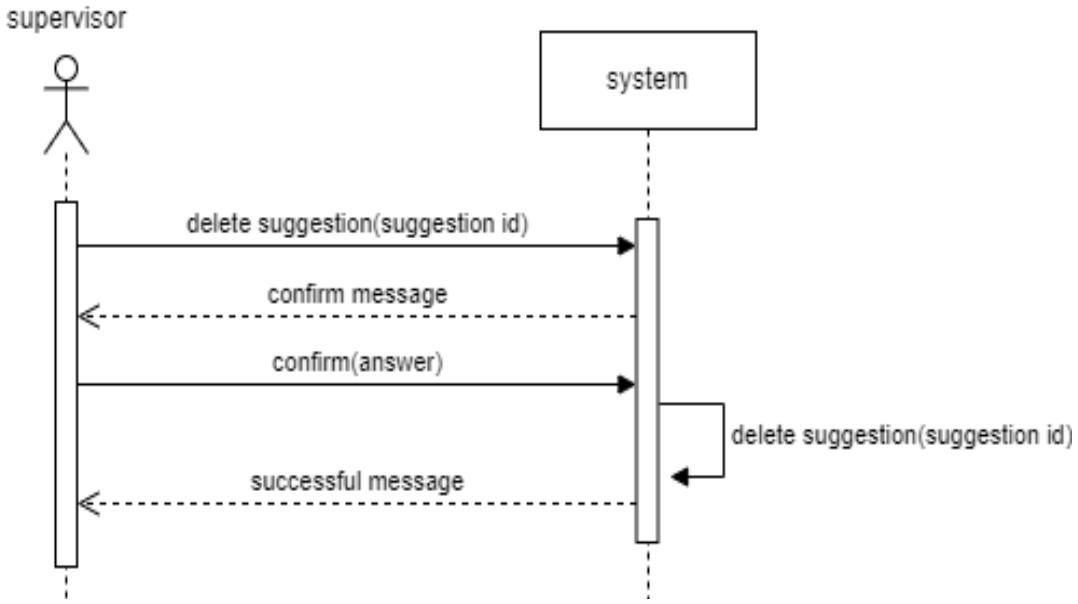


Figure 14 sprint#1 delete suggestion sequence

- Use case – add suggestion:

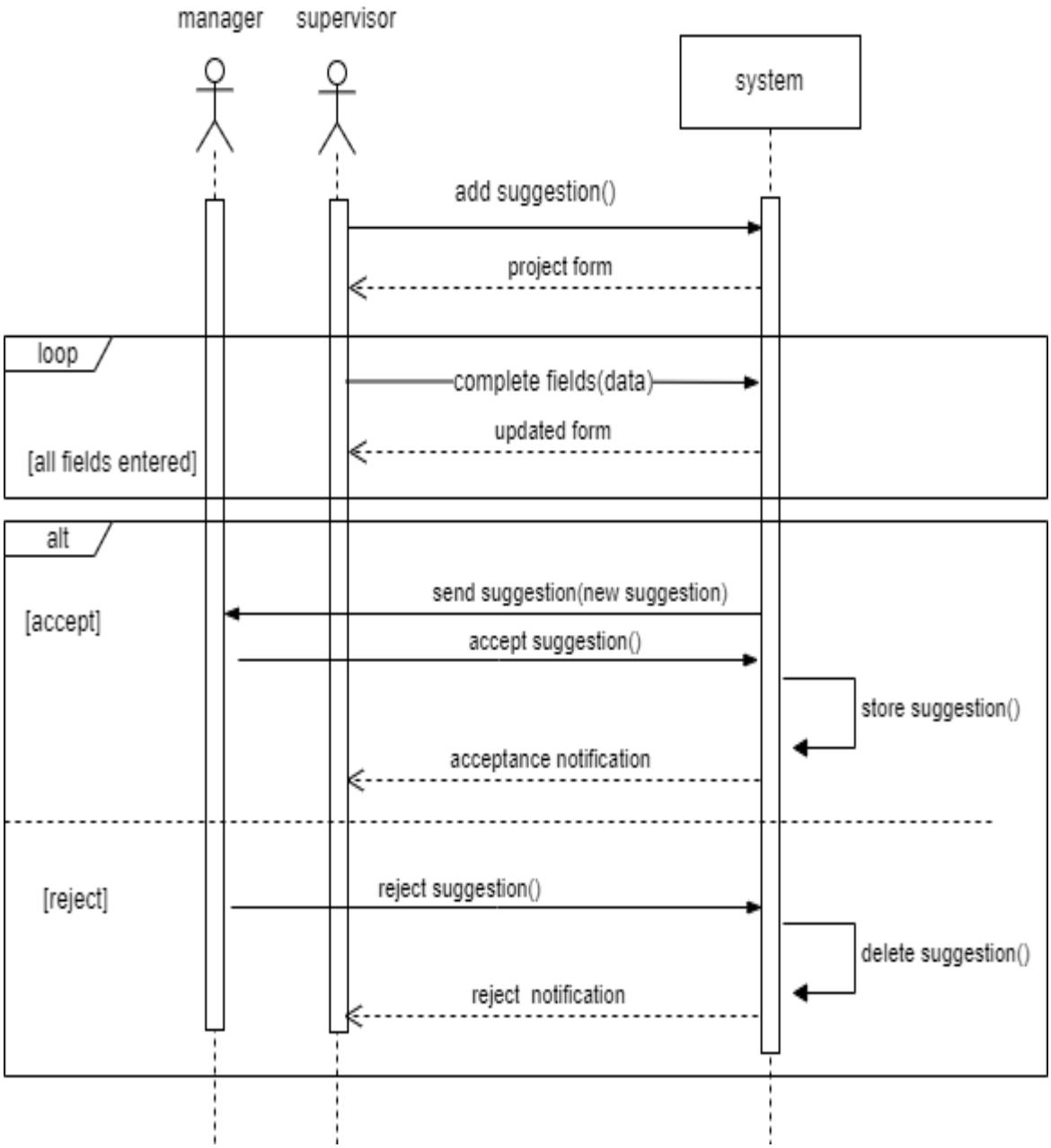


Figure 15 sprint#1 add suggestion sequence

- Use case – change profile photo:

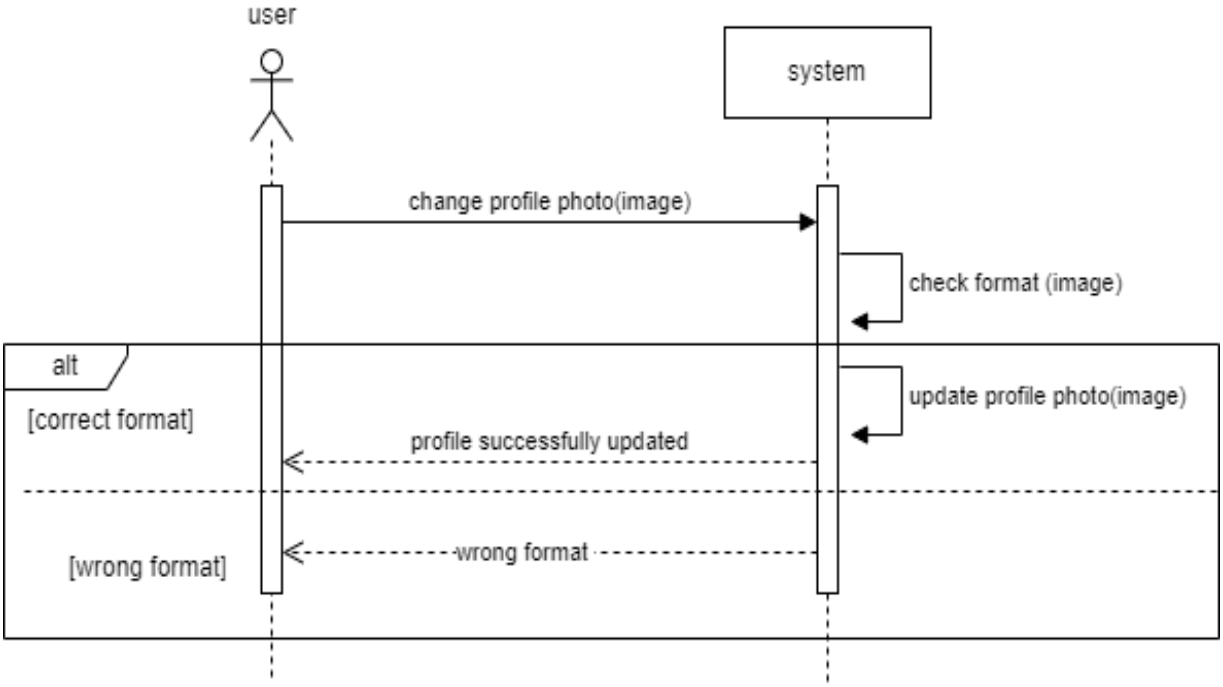


Figure 16 sprint#1 change profile photo sequence

- Class Diagram for analysis phase: In the analysis stage, a class diagram can help you to understand the requirements of your problem domain and to identify its components.

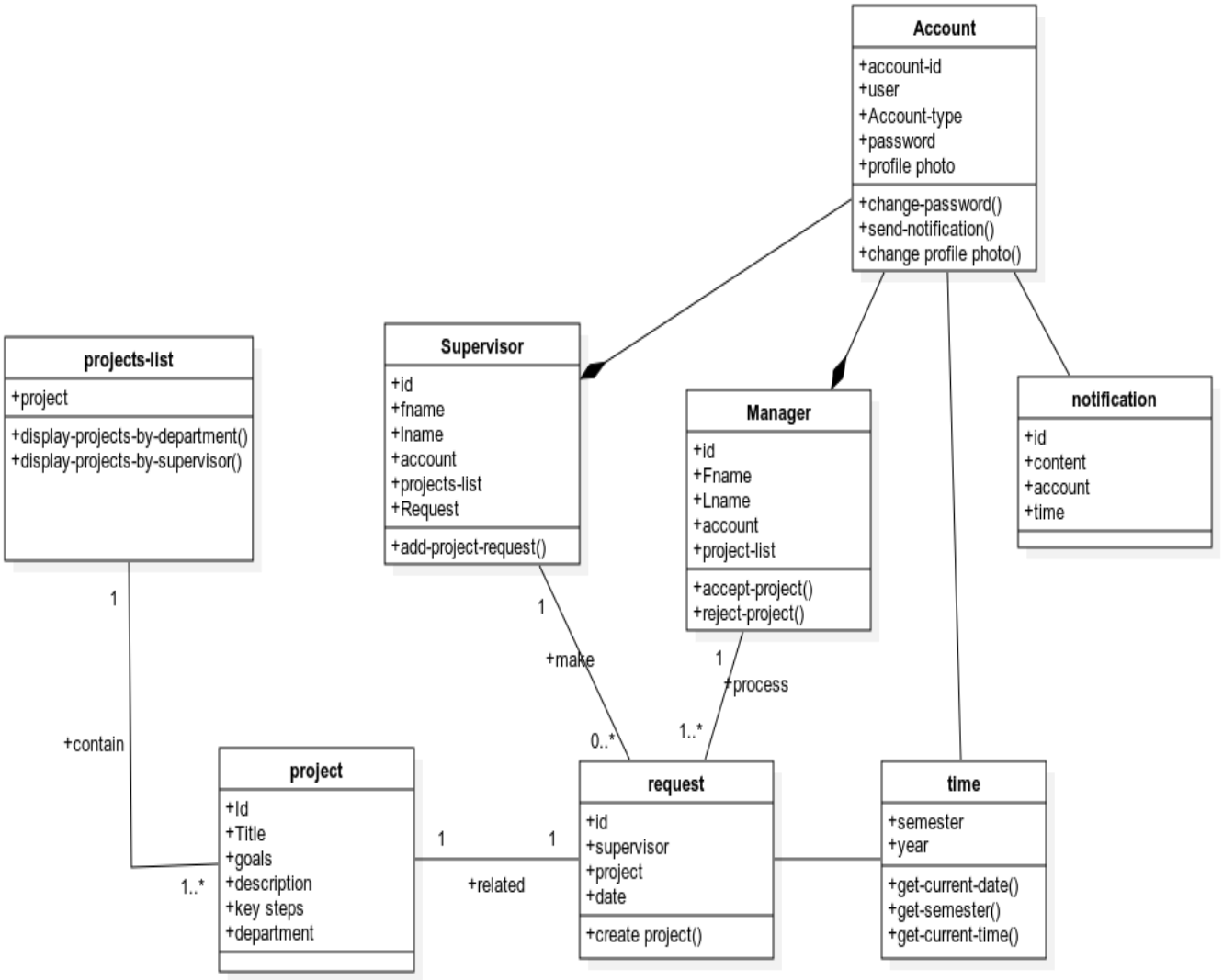


Figure 17 sprint#1 analysis class diagram

4. Initial Test Cases:

Table 9 initial test case

Test case scenario:		Sce-01: Check login functionality.		
Test case id	Test case title	Req-id	Test steps	Expected result
Tc-01	Check results on entering a valid id and password.	Req-01	1. Launch the application on the login page. 2. Enter ID and password. 3. Choose "login".	The login should be successful.
Tc-02	Check results on entering an invalid id, or password.	Req-01	1. Launch the application on the login page. 2. Enter ID and password. 3. Choose "login".	Error message "invalid id or password."
Tc-03	Check results when a user id is empty and the "login" button is pressed.	Req-01	1. Launch the application on the login page. 2. Enter a password. 3. Choose "login".	Error message "a field is missing"

Test case scenario:		Sce-02: Check to add project suggestion functionality.		
Test case id	Test case title	Req-id	Test steps	Expected result
Tc-04	Check results on completing all the project form fields and the "submit" button is pressed.	Req-02	1. Launch the application by the supervisor. 2. Choose to add a project suggestion. 3. Complete the form of the projects.	The suggestion successfully goes to the manager.

Tc-05	Check results by pressing “submit” button with missing fields on the project form.	Req-02	1. Launch the application by the supervisor. 2. Choose to add a project suggestion. 3. Complete the form of the projects.	Error message “Complete the form”.
Tc-06	Check results when entering values that are not strings in the “title”, “description” and “goal” fields.	Req-02	1. Launch the application by the supervisor. 2. Choose to add a project suggestion. 3. Complete the form of the projects.	Error message “Please use characters”.
Tc-07	Check results on choosing to track a user request by pressing “my requests"	Req-04	1. Launch the application by the supervisor. 2. Press “my requests”	Show all requests that the user made.

Test case scenario:		Sce-03: Check delete or edit project suggestion functionality.		
Test case id	Test case title	Req-id	Test steps	Expected result
Tc-08	Check the results on pressing the “confirm delete” button for a suggestion.	Req-03	1. Launch the application by the supervisor. 2. Choose a project from “my suggestion page”. 3. Choose ‘delete project’.	The project must be deleted successfully from the suggestions list and for the manager.
Tc-09	Check the result on pressing “edit button”	Req-03	1. Launch the application by the supervisor.	The project must be edit

	after completing the whole new project form.		2. Choose a suggestion from "my suggestions page". 3. Choose 'edit suggestion'. 4. Complete form fields.	successfully, and the system show "process complete successfully"
Tc-10	Check the result on pressing "edit button" without completing the whole new project form.	Req-03	1. Launch the application by the supervisor. 2. Choose a suggestion from "my suggestions page". 3. Choose 'edit suggestion'. 4. Enter data.	The system will show "please complete the fields".

Test case scenario:		Sce-05: Check to accept or reject suggestions functionality		
Test case id	Test case title	Req-id	Test steps	Expected result
Tc-11	Check results on choosing to open suggestion list by the manager.	Req-05	1. Launch the application by the manager. 2. Choose to open suggestions.	All projects added by the supervisor must be added, and with options to accept or reject.
Tc-12	Check results on pressing the "reject" button for a project suggestion.	Req-06	1. Launch the application by the manager. 2. Choose to open suggestions. 3. Press the "reject" button for a project.	The project must be deleted from the list and the system must inform the supervisor of the result by notification.

Tc-13	Check results on pressing the "accept" button for a project suggestion.	Req-06	1. Launch the application by the manager. 2. Choose to open suggestions. 3. Press the "reject" button for a project.	The project must be added to the accepted suggestions list and the system must inform the project supervisor of the result.
Tc-14	Check result after receiving any response.	Req-7	1. Launch the application by the manager. 2. check the notification page.	The system must send a notification for any user who receive a response or action.

Test case scenario:		Sce-06: Check display project list functionality.		
Test case id	Test case title	Req-id	Test steps	Expected result
Tc-15	Check results by choosing "display suggestion list".	Req-08	1. Launch the application. 2. Choose "display project list".	All accepted suggestions must be displayed in the project list.
Tc-16	Check the result in choosing to filter the list by departments or supervisor.	Req-9	1. Launch the application. 2. Choose "display project list". 3. Choose "filter by departments" or "filter by supervisor".	The list must be sorted by the departments or supervisor and redisplay.

Test case scenario:		Sce-7: Check the change password functionality.		
Test case id	Test case title	Req-id	Test steps	Expected result
Tc-17	Check results on entering the correct old password and a strong new password.	Req-11	1. Launch the application by the supervisor. 2. Choose to “change password”. 3. Enter the old password. 4. Enter a new password	The password must be changed successfully.
Tc-18	Check results on entering an incorrect old password.	Req-11	1. Launch the application by the supervisor. 2. Choose to “change password”. 3. Enter the old password. 4. Enter a new password.	Error message “incorrect old password”.
Tc-19	Check results on entering a new password that is not strong enough	Req-11	1. Launch the application by the supervisor. 2. Choose to “change password”. 3. Enter the old password. 4. Enter a new password.	Error message “New password is not strong enough”

Test case scenario:		Sce-8: Check change profile photo functionality.		
Test case id	Test case title	Req-id	Test steps	Expected result
Tc-20	Check results on uploading a correct format for changing profile pictures.	Req-10	1. Launch the application. 2. Choose to change your profile photo. 3. Upload a photo.	The photo must be changed successfully.
Tc-21	Check results on uploading incorrect format.	Req-10	1. Launch the application. 2. Choose to change the profile photo. 3. Upload a photo.	Error message" uploaded format is not supported".

5. Updating requirements traceability matrix – sprint1:

Req-id	Title	Analysis	Detailed design	App interfaces	coding	Test cases
Req-01	The system must allow users to log in to their accounts with an ID and password.	Sp1an				Tc-01 Tc-02 Tc-03
Req-02	The system must allow a supervisor to add a project suggestion.	Sp1an				Tc-04 Tc-05 Tc-06
Req-03	The system must allow a supervisor to edit or delete suggestion.	Sp1an				Tc-08 Tc-09 Tc-10
Req-04	the system must allow users who request to track their requests state.	Sp1an				Tc-07
Req-05	The system must be able to inform the manager of all projects suggestions	Sp1an				Tc-11
Req-06	The system must be able to inform the manager of all projects suggestions	Sp1an				Tc-12 Tc-13
Req-07	The system must be able to inform a request maker of the response.	Sp1an				Tc-14
Req-08	The system must be able to display the suggestions list for users.	Sp1an				Tc-15
Req-09	The system must be able to display the suggestions	Sp1an				Tc-16

	list filtered by supervisors or department.					
Req-10	The system must allow users to change their profile photo.	Sp1an				Tc-20
Req-11	The system must allow users to change their account password.	Sp1an				Tc-21

Sprint #1 Design:

In this section, we will introduce the detailed design for the components of the first sprint, and database components.

1. Detailed design class diagram:

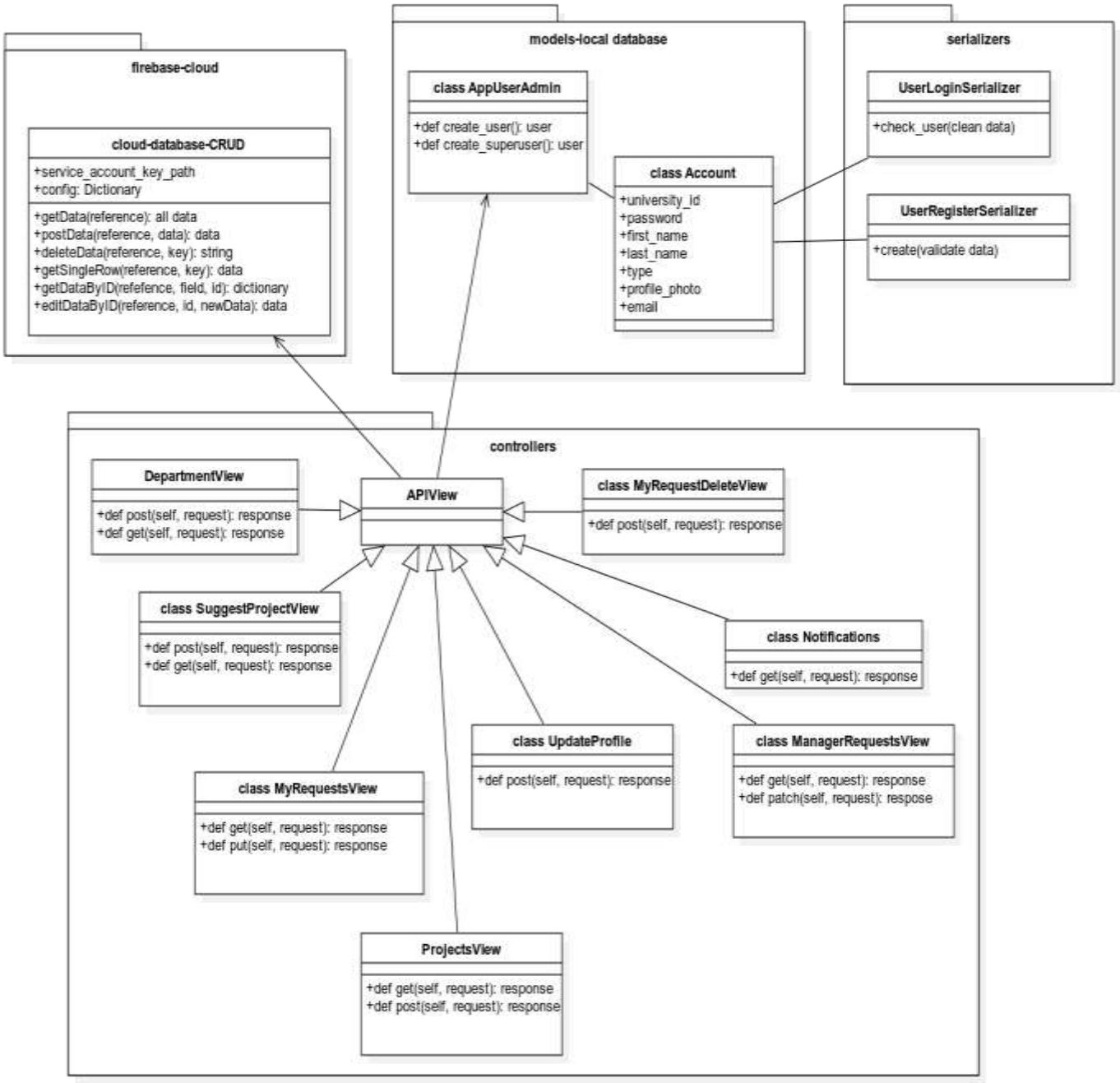


Figure 18 sprint#1 design class diagram

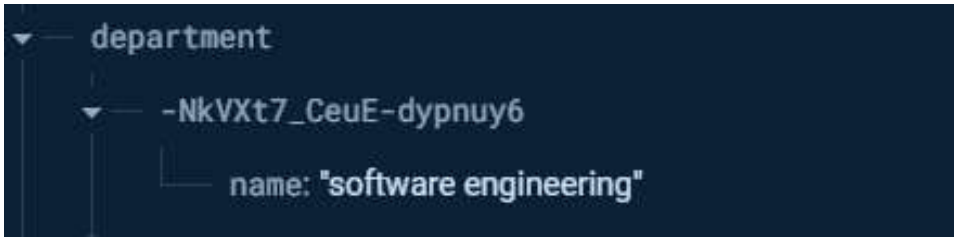
2. Database Design:

A. Firebase Realtime Database: The Realtime Database uses a hierarchical data structure, similar to a tree or a JSON object. The top-level nodes in the database are known as "root" nodes, and each node can have child nodes, forming a nested structure. Each node in the database is identified by a unique key. The data in the database is organized based on these nodes and keys.

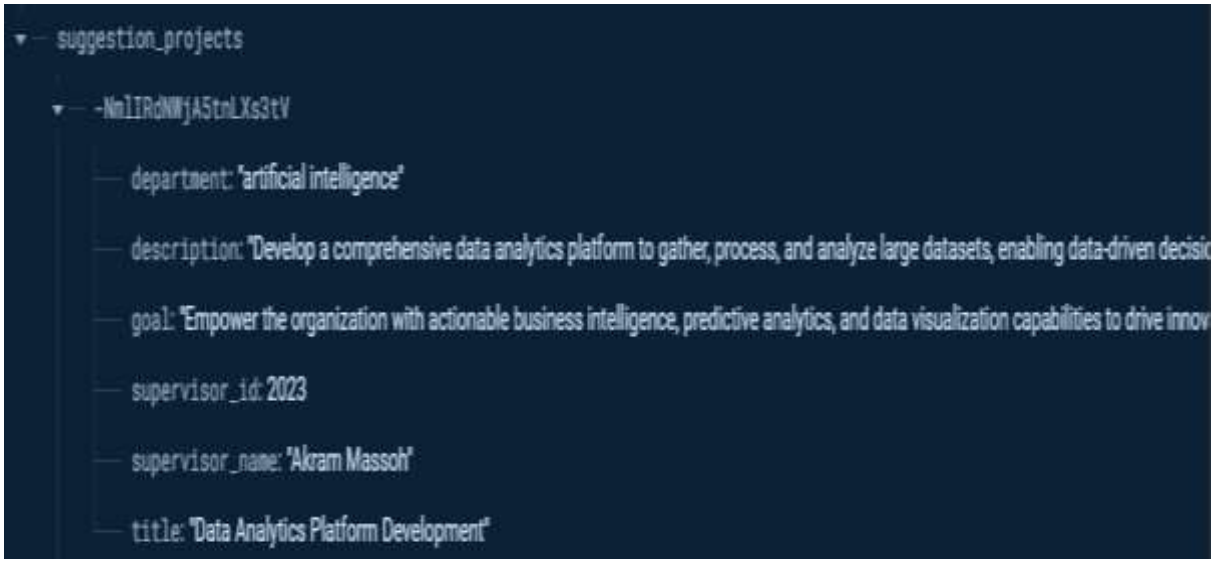
Our project database structure:

Figure 19 sprint#1 database structure

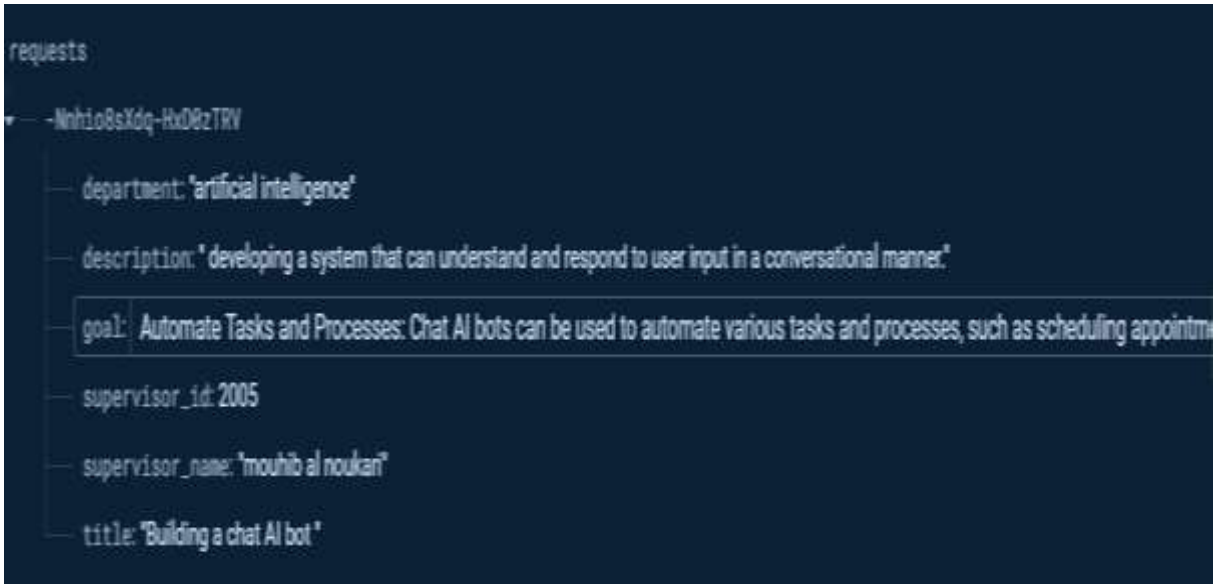
Department reference:



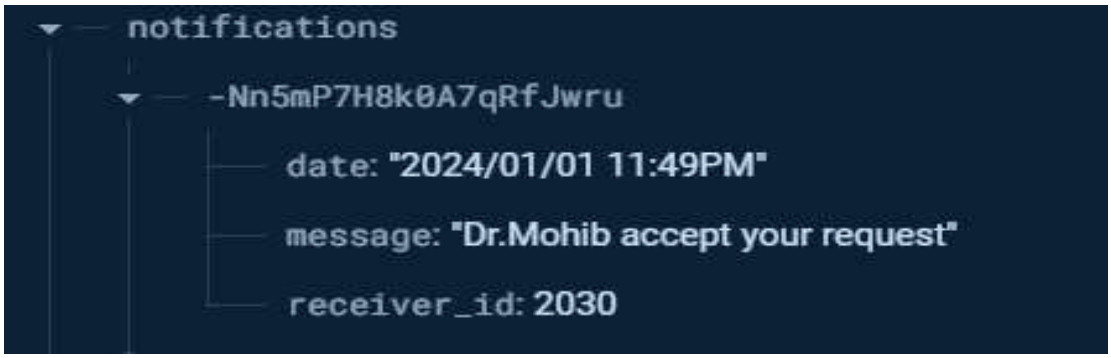
Accepted suggestions reference:



Suggestion request reference:



Notifications reference:



B. For authentication and authorization purposes that firebase Realtime database did not provide, we needed to made a local database containing "account" table for managing accounts and user authentication.

Table 10 account database table design

Account Database Table			
Field name	type	property	The input
University id	Integer Field	PK	user
Password	Char field (255)		user
First name	Char field (255)		user
Last name	Char field (255)		user
type	Char field (255)		user
Profile photo	Image field		user
email	Email field		user

3. Site map:

A sitemap diagram is a visual representation of the structure and organization of a website's pages and content. It illustrates the relationships between different pages, sections, and categories within the website.

Sprint-1 site map:

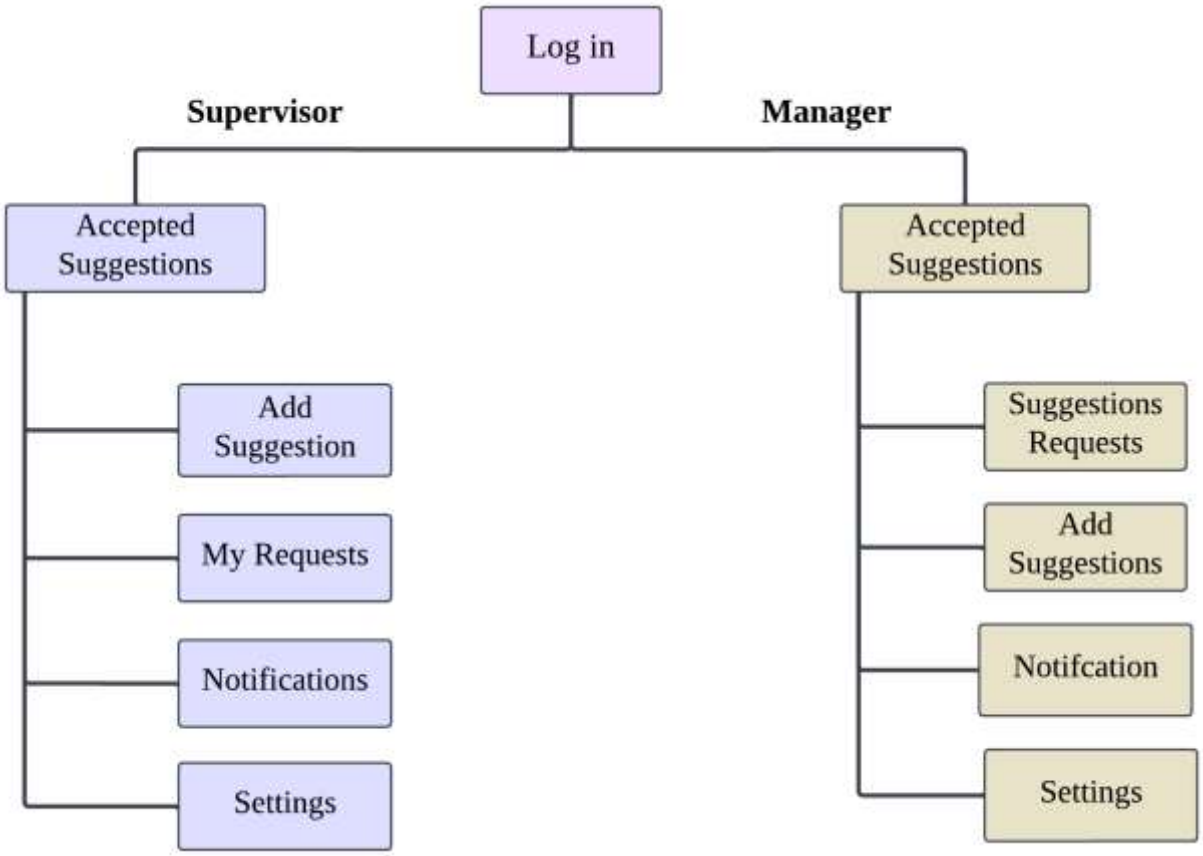


Figure 20 sprint#1 site map

Sprint #1 implementation and testing:

1. Used tools:
    - ❖ Django

      is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source. especially we use DRF: Django REST Framework is a widely-used, full-featured API framework designed for building RESTful APIs with Django. At its core, DRF integrates with Django's core features "models, views, and URLs" making it simple and seamless to create a RESTful API.

    - ❖ React

      React is a popular JavaScript library for building user interfaces. It was created by Facebook and is widely used in web development. React allows developers to build reusable UI components that can efficiently update and render changes to the user interface when the underlying data changes.

      React s primary focus is on building user interfaces, and it excels in creating interactive and dynamic web applications.

❖ Insomnia

Insomnia is an open-source desktop application that takes the pain out of interacting with and designing, debugging, and testing APIs, we use it for test our APIs.

❖ The Realtime Database from Firebase

is a cloud-hosted NoSQL database that allows developers to store and sync data. In the Firebase Realtime Database, data is stored in a JSON format. JSON is a lightweight and widely used data interchange format that represents data as key-value pairs and nested structures. Developers can create, update, and delete data by referencing the path to a specific node and key within the database.

❖ My SQL Database

We used it for local storage for the authentication process, because the Realtime database does not provide an authentication service. MySQL is an open-source relational database management system (RDBMS) that is widely used for storing, managing, and retrieving data. It is one of the most popular and widely adopted databases in the world, known for its reliability, scalability, and ease of use

❖ Visual studio code (VS code):

Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging.

We use it to develop the whole project (frontend, backend).

❖ GitHub:

is a web-based platform that provides a hosting service for version control repositories. It allows developers to collaborate on projects, track changes to code, and manage software development processes "The project repository" .

2. App interfaces

❖ Log in interface:

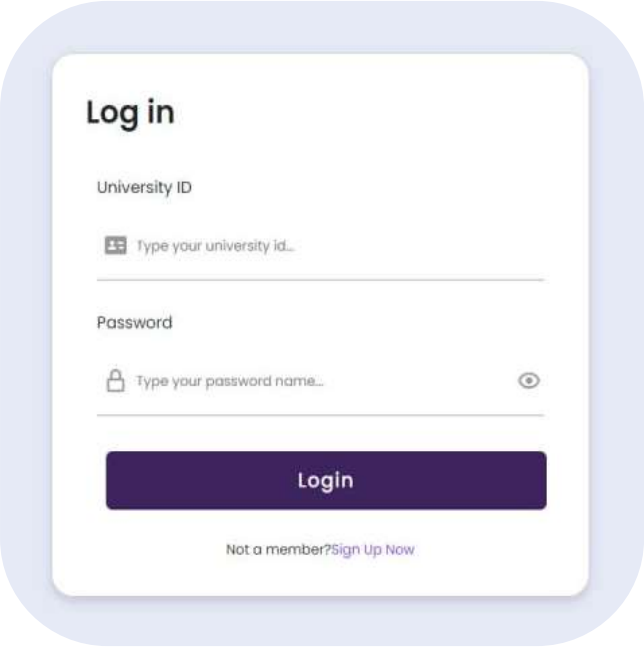


Figure 21 sprint#1 log in interface inrf-01

❖ Main page (accepted suggestions list):

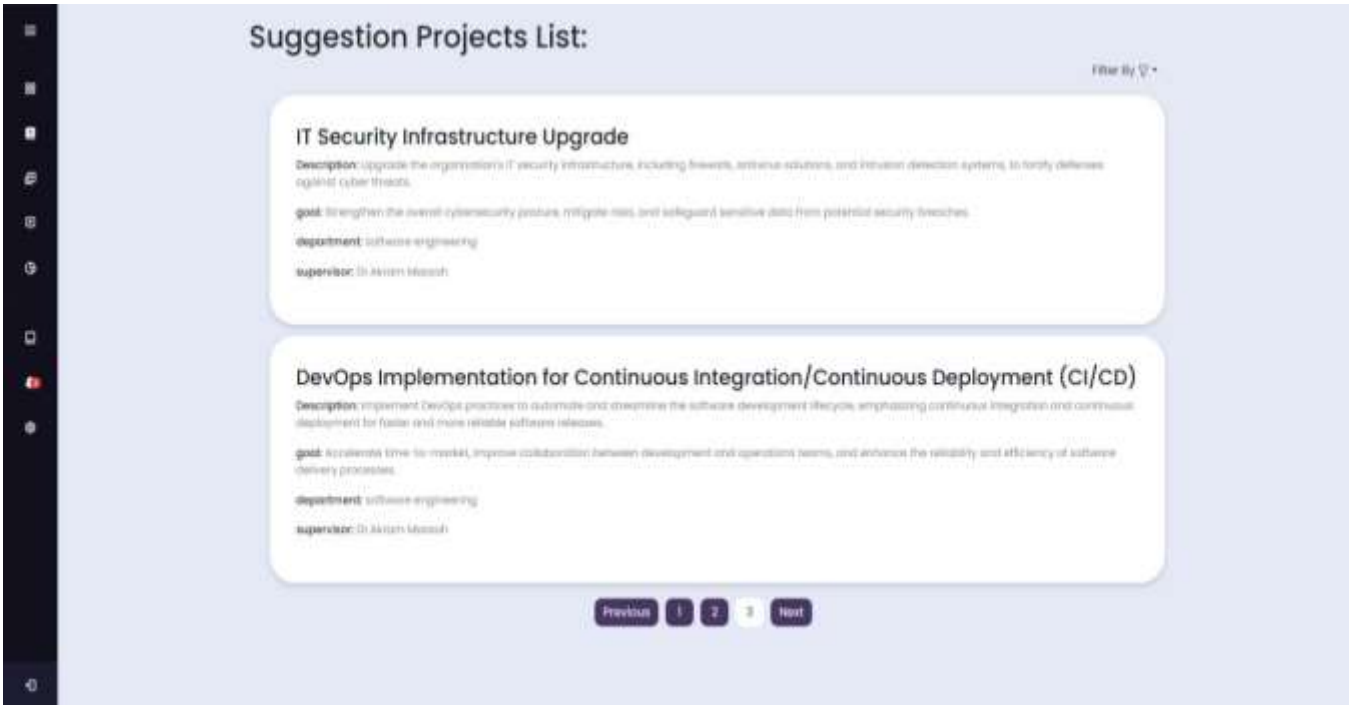


Figure 22 sprint#1 accepted suggestions interface inrf-02

❖ Filtering options:

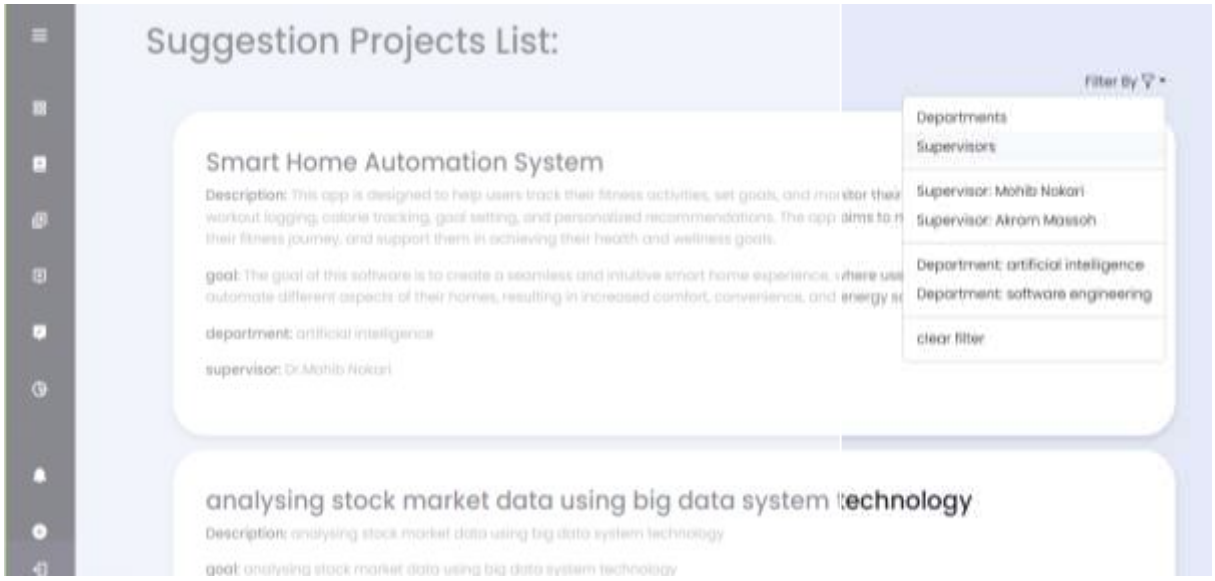


Figure 23 sprint#1 filtering options interface inrf-03

❖ Settings to edit profile photo or password:

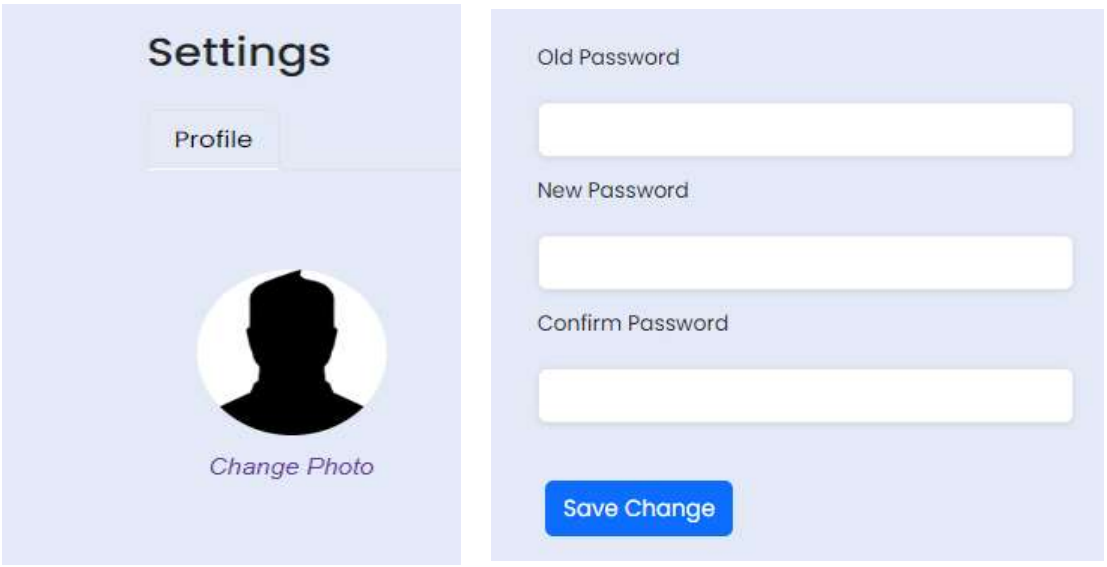


Figure 24 sprint#1 setting interface inrf-04

❖ Add suggestion interface (project form):

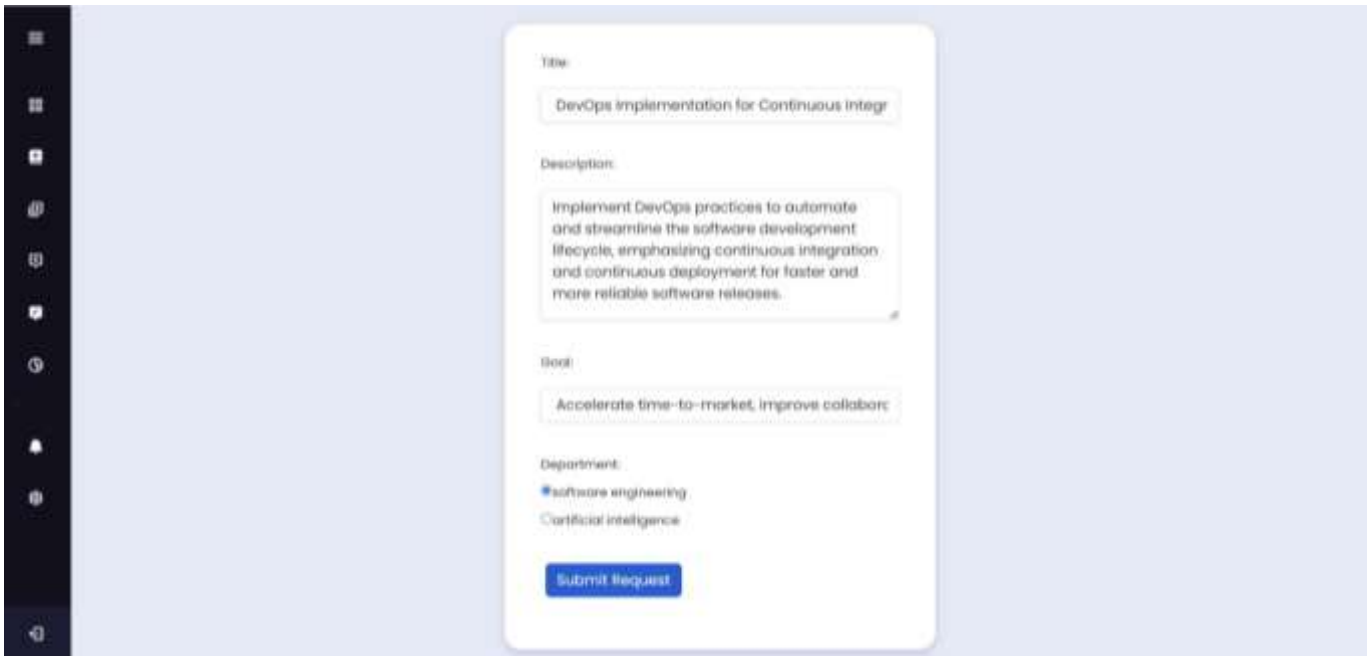


Figure 25 sprint#1 add suggestion interface inrf-05

❖ User requests tracking, editing, and deleting interface(supervisor):

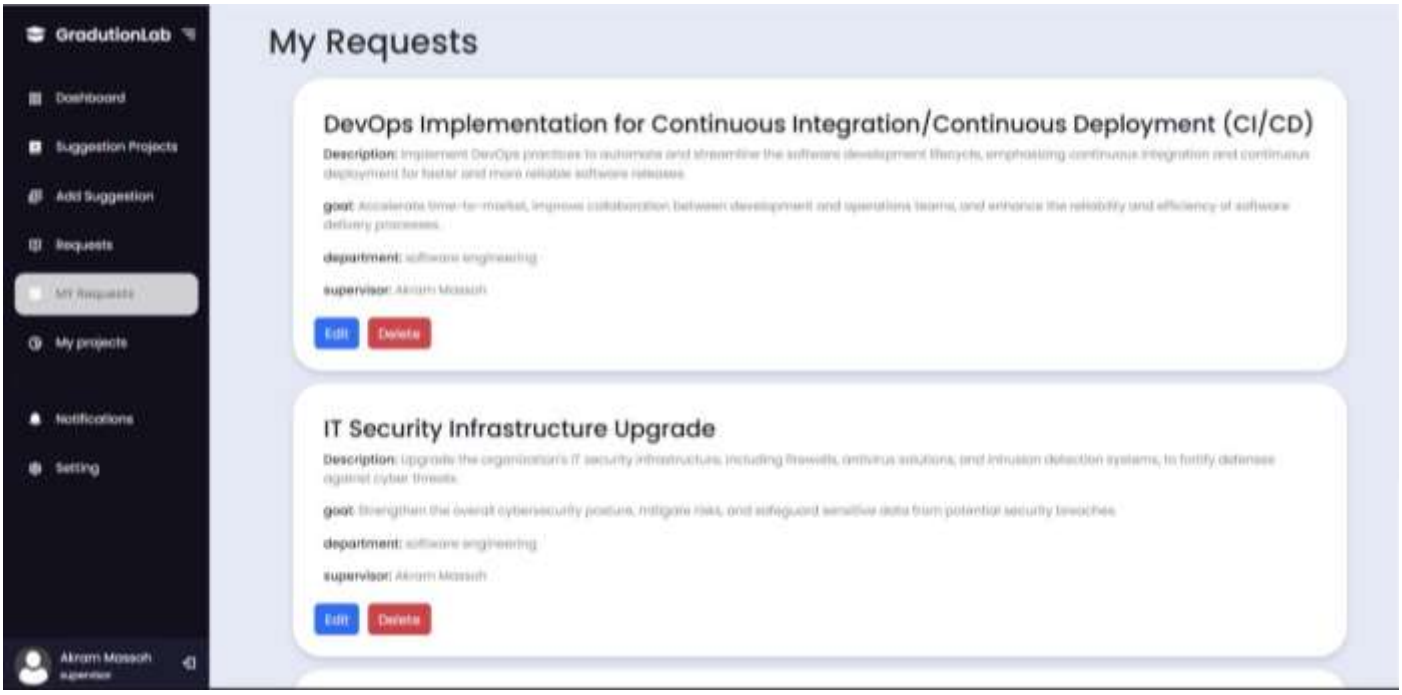


Figure 26 sprint#1 my request interface inrf-06

❖ All pending suggestion requests (manager account):

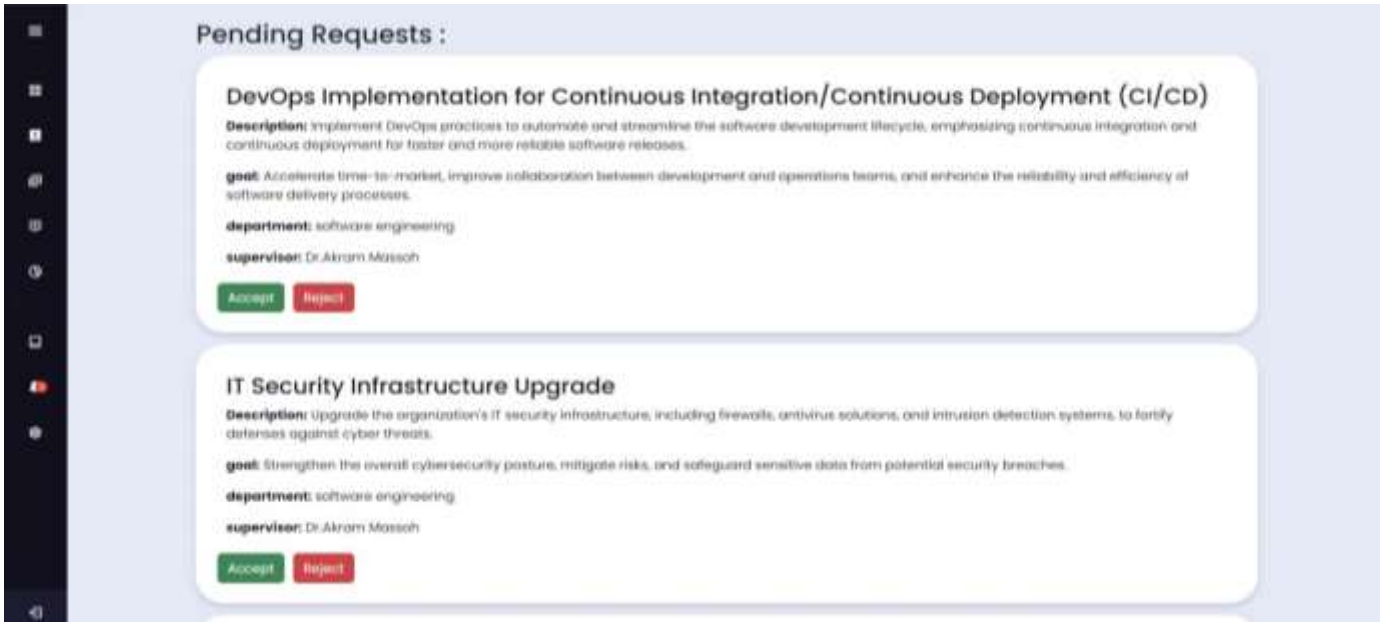


Figure 27 sprint#1 pending suggestions interface inrf-07

❖ Notifications page:

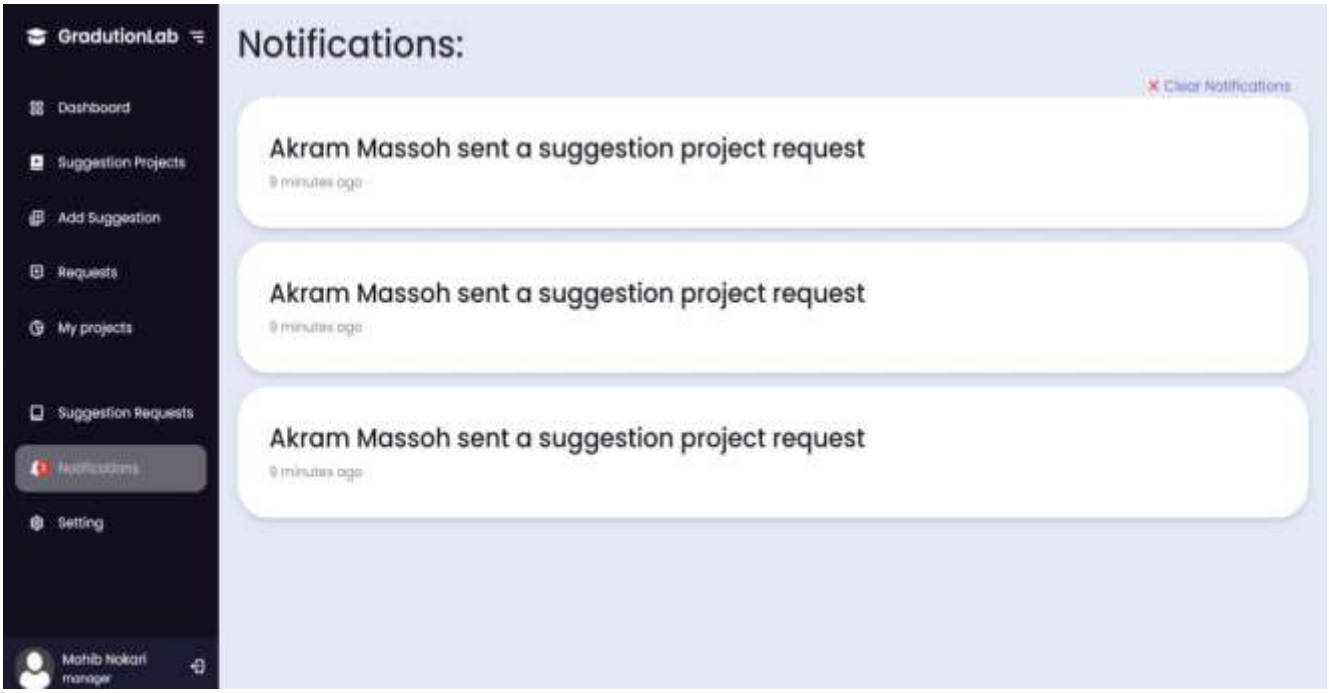


Figure 28 sprint #1 notification interface -inrf-8

3. Test Cases execution:

Table 11 sprint#1 test case execution

TC id	Test case title	Req-id	Tested data	Expected result	Actual result	Pass/ fail
Tc-01	Check results on entering a valid ID and password.	Req-01	Id=4200022 Password= Loki1234	Login successfully.	Login successfully.	Pass
Tc-02	Check results on entering invalid ID, or password.	Req-01	Id=123 Password= Loki0000	Error message "invalid id or password."	Error message "invalid id or password."	Pass
Tc-03	Check results when a user id is empty and the "login" button is pressed.	Req-01	Id= Password=Loki1234	Error message "a field is missing"	Error message "a field is missing"	Pass
Tc-04	Check results on completing all the project form fields and the "submit" button is pressed	Req-02	Title=analysis Coved-19 data. Description: gathering and analysis data. Goal: student will get many skills. Department: AI.	The suggestion successfully goes to the manager, and the system show "process complete successfully".	The suggestion successfully goes to the manager, and the system show "process complete successfully".	Pass

Tc-05	Check results by pressing "submit" button with missing fields on project form.	Req-02	Title=analysis Coved-19 data. Description: Goal: student will get many skills. Department: AI.	Error message "Complete the form".	Error message "complete the form".	Pass
Tc-06	Check results when entering values that are not strings in the "title", "description" and "goal" fields.	Req-02	Title=analysis Coved-19 data. Description: 123. Goal: student will get many skills. Department: AI.	Error message "Please use characters".	Error message "Please use characters".	Pass
Tc-07	Check results on choosing to track a user request by pressing "my requests"	Req-04		Show all requests that the user made.	Show all requests that the user made.	Pass
Tc-08	Check the results on pressing the "confirm delete" button for a suggestion.	Req-03		The project must be deleted successfully from the suggestions list and for the manger.	The project must be deleted successfully from the suggestions list and for the manager.	Pass
Tc-09	Check result on pressing "edit button"	Req-03	Title: analysis store data.	The project must be edit successfully,	The project must be edit successfully, and	pass

	after completing the whole new project form.		Description: gathering and analysis data. Goal: student will get many skills. Department: AI.	and the system show "process complete successfully"	the system show "process complete successfully"	
Tc-10	Check result on pressing "edit button" without completing the whole new project form.	Req-03	Title: analysis store data. Description: gathering and analysis data. Goal: Department: AI.	The system will show "please complete the fields".	The system will show "please complete the fields".	pass
Tc-11	Check results on choosing to open suggestion list by the manager.	Req-05		All projects added by the supervisor must be added, and with options to accept or reject.	All projects added by the supervisor must be added, and with options to accept or reject.	Pass
Tc-12	Check results on pressing the "reject" button for a project suggestion.	Rrq-06		The project must be deleted from the list and the system must inform the supervisor of the result by notification	The project must be deleted from the list and the system must inform the supervisor of the result by notification	Pass
Tc-13	Check results on pressing	Req-06		The project must be added	The project must be added	Pass

	the "accept" button for a project suggestion.			to the accepted suggestions list and the system must inform the project supervisor of the result	to the accepted suggestions list and the system must inform the project supervisor of the result	
Tc-14	Check result after receiving any response.	Req-07		The system must send notification for any user who receive a response.	The system must send notification for any user who receive a response.	Pass.
Tc-15	Check results by choosing "display suggestion list".	Req-08		All accepted suggestions must be displayed in the project list.	All accepted suggestions must be displayed in the project list.	Pass
Tc-16	Check the result in choosing to filter the list by departments or supervisor.	Req-09		The list must be sorted by the departments or supervisor and redisplay.	The list must be sorted by the departments or supervisor and redisplay.	Pass
Tc-17	Check results on entering the correct old password and a strong new password.	Req-11	Old password: loki1234 New password: Loki2002	The password must be changed successfully, and the system show "success".	The password must be changed successfully, and the system show "success".	Pass

Tc-18	Check results on entering an incorrect old password.	Req11	Old password: loki1222 New password: Loki2002	Error message “incorrect old password”.	Error message “incorrect old password”.	Pass
Tc-19	Check results on entering a new password that is not strong enough	Req-10	Old password: loki1222 New password: Loki	Error message “New password is not strong enough”	Error message “New password is not strong enough”	Pass
Tc-20	Check results on uploading the correct format for changing profile pictures.	Req-10	Image.jpg	The photo must be changed successfully.	The photo must be changed successfully.	Pass
Tc-21	Check results on uploading incorrect format.		Image.svg	Error message ” uploaded format is not supported”.	Error message ” uploaded format is not supported”.	Pass

4. Final requirements traceability matrix – sprint 1:

Req-id	Title	Analysis	Detailed design	App interfaces	coding	Test cases
Req-01	The system must allow users to log in to their accounts with an ID and password.	Sp1an	Sp1dds	Inrf-01	Sp1imp	Tc-01 Tc-02 Tc-03
Req-02	The system must allow a supervisor to add a project suggestion.	Sp1an	Sp1dds	Inrf-05	Sp1imp	Tc-04 Tc-05 Tc-06
Req-03	The system must allow a supervisor to edit or delete suggestion.	Sp1an	Sp1dds	Inrf-06	Sp1imp	Tc-08 Tc-09 Tc-10
Req-04	the system must allow users who request to track their requests state.	Sp1an	Sp1dds	Inrf-06	Sp1imp	Tc-07
Req-05	The system must be able to inform the manager of all projects suggestions	Sp1an	Sp1dds	Inrf-07	Sp1imp	Tc-11
Req-06	The system must be able to inform the manager of all projects suggestions	Sp1an	Sp1dds	Inrf-07	Sp1imp	Tc-12 Tc-13
Req-07	The system must be able to inform a request maker of the response.	Sp1an	Sp1dds	Inrf-08	Sp1imp	Tc-14
Req-08	The system must be able to display the suggestions list for users.	Sp1an	Sp1dds	Infr-02	Sp1imp	Tc-15
Req-09	The system must be able to display the suggestions	Sp1an	Sp1dds	Intr-03	Sp1imp	Tc-16

	list filtered by supervisors or department.					
Req-10	The system must allow users to change their profile photo.	Sp1an	Sp1dds	Intr-04	Sp1imp	Tc-20
Req-11	The system must allow users to change their account password.	Sp1an	Sp1dds	Intr-04	Sp1imp	Tc-21

Sprint #2

Sprint #2 Analysis:

In this section, we will introduce the analytical study for the second sprint using the needed UML diagrams.

1. Sprint backlog:

The functional requirement list we will complete for this sprint:

- ✓ Req-01: the system must allow the students to make an account by their university ID (unique account).
- ✓ Req-02: the system must be able to check if a student belongs to the university by comparing some entered data with the student data.
- ✓ Req-03: The system must allow students to request a project.
- ✓ Req-04: The system must be able to check if a student and a team met the project's registration conditions.
- ✓ Req-05: The system must be able to get the acceptance of all team members for a request.
- ✓ Req-06: The system must be able to inform the supervisor about the requests made for his project suggestions.
- ✓ Req-07: The system must allow a student who request to delete his request.

- ✓ Req-08: The system must allow supervisors to either accept or reject a project request.
- ✓ Req-09: The system must inform the employee of the projects that are ready for registration.
- ✓ Req-10: the system must be able to inform the students if their project has been registered.
- ✓ Req-11: The system must be able to display registered project list.
- ✓ Req-12: The system must be able to display registered project list filtered by supervisors or departments.

2. Initial Requirements traceability Matrix – sprint2:

Req -id	Title	Analysis	Detailed design	coding	App user interface	Test cases
Req -01	the system must allow the students to make an account by their university ID (unique account).					
Req -02	the system must be able to check if a student belongs to the university by comparing some entered data with the student data					
Req -03	The system must allow students to request a project.					
Req -04	The system must be able to check if a student and a team met the project's registration conditions					
Req -05	The system must be able to get the acceptance of all team members for a request					
Req -06	The system must be able to inform the supervisor about the requests made for his project suggestions					
Req -07	The system must allow a student who request to delete his request.					

Req -08	The system must allow supervisors to either accept or reject a project request.					
Req -09	The system must inform the employee of the projects that are ready for registration.					
Req -10	the system must be able to inform the students if their project has been registered.					
Req -11	The system must be able to display registered project list.					
Req -12	The system must be able to display registered project list filtered by supervisors or departments.					

3. Requirements Modeling:

- Use Case Diagram:

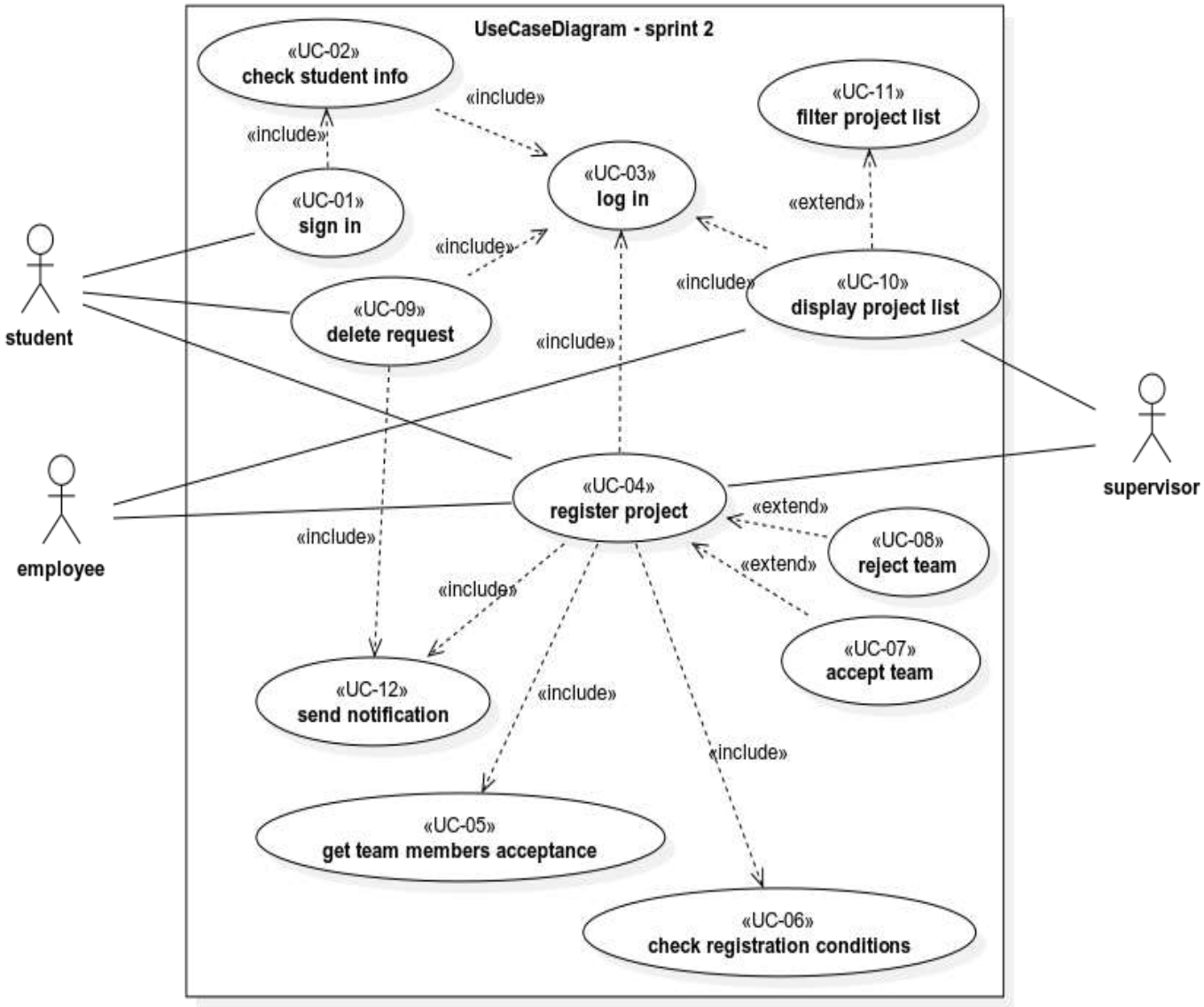


Figure 29 sprint#2 use case

- Use case specification:

Table 12 sprint#2 sign in specification

Use case name:	Sign in
Participating Actors:	initiated by: students
The flow of events:	1. The student enters the website in the sign page. 2. The system shows a form. 3. The student completes the form and chose “create account”. 4. The system check if all fields completed. 5. The system will then compare each field the student enters with the student data it has from the university. 6. If all data are matches the system will add new account and show accept message.
Alternative flows:	First alternative flow A1: start at step 4 in the main flow, there is a missing field: 5. the system will show an error message “there is a missing field”. 6. the user will complete the fields and the flow will return to step 3.
Exception flows:	First exemption flow E1: start at step 5 in the main flow, there is an unmatched data. 6. the system will show an error message “data is not correct”, and the use case will fail.
Entry condition	The system has the student university data.
Exit conditions	The student has an account.

Use case name:	Register a project
Participating Actors:	initiated by: students supervisor, employee
The flow of events:	1. The student chose a suggestion from the suggestions list and chose "apply". 2. The system will show a registration form. 3. The student chose the number of his team member, and then enter their university ID, and then chose "apply". 4. The system will check the registration conditions for all students in this request by the use of the students university data. • The system will check if the students complete more or equal to 100 hours. • The system will check if students complete the necessary courses (application for junior, junior for senior1, senior1 for senior2). • Finally, the system will check if the all the team members hours are close to each other (the difference less than 7 hours). 5. If all these conditions are true the system will send an accept message and sent this request to all other team members to take their acceptance of the project registration request. 6. the other students will receive the request. 7. If all student accepted this request the system will send the request to the supervisor of this project. 8. The system will enable any student to request for other project or any other students to request using their names. 9. The supervisor will receive the request. 10. If the supervisor accepts this request: • The system will inform the employee of the new project that ready to register. • The employee will receive the request and register the project on the university system and chose "complete".

	• The system will send notification to all team member about the acceptance. • The system will add the project in the page "my project" for all team member with their supervisor.
Exception flows:	First exception flow E1: start at step 7 if one of the team members reject the request. 8. the system will delete the request and will not send it the supervisor. 9. the system will send notification to other students to inform them of the reject, and the use case will fail. Second exception flow E2: start at step 10 from the main flow, if the supervisor rejects the request: 11.The system will delete the request from all students. 12.The system will send a notification of the response. 13.the student can request again for another project, and the use case will fail.
Entry condition	The student had logging in
Exit conditions	The students had registered a project.

Table 15 sprint#2 register a project specification

Table 14sprint#2 delete request specification

Use case name:	Delete request
Participating Actors:	initiated by: students
The flow of events:	1. The student chose to delete a request he made for a project. 2. Fist the system will check if all other students accept to send this request the system enable the student from delete this request. 3. If other students did not accept yet the system will ask the student to confirm his decision. 4. The student will confirm his decision. 5. The system will delete the request from the database and from other students pages. 6. The system will send a notification for other students about the updates.
Entry condition	The student had logging in The student had a request.
Exit conditions	The request is deleted.

Table 15 sprint#2 display registered projects list specification

Use case name	Display registered projects list
Participating actors	initiated by all users.
Flow of events	1. The actor selects the "Display registered project List" option from the user interface. 2. The system will display the registered project list.

	3. If the user chooses to filter the list. 4. The system will show two options. 5. If the user chooses “by department”. 6. The system will filter the list and display it. 7. If the user chooses “by supervisor”. 8. The system will filter the list and display it.
Entry conditions:	user log in
Exit conditions:	registered project list displayed

- Activity diagram:

❖ Use case - sign in:

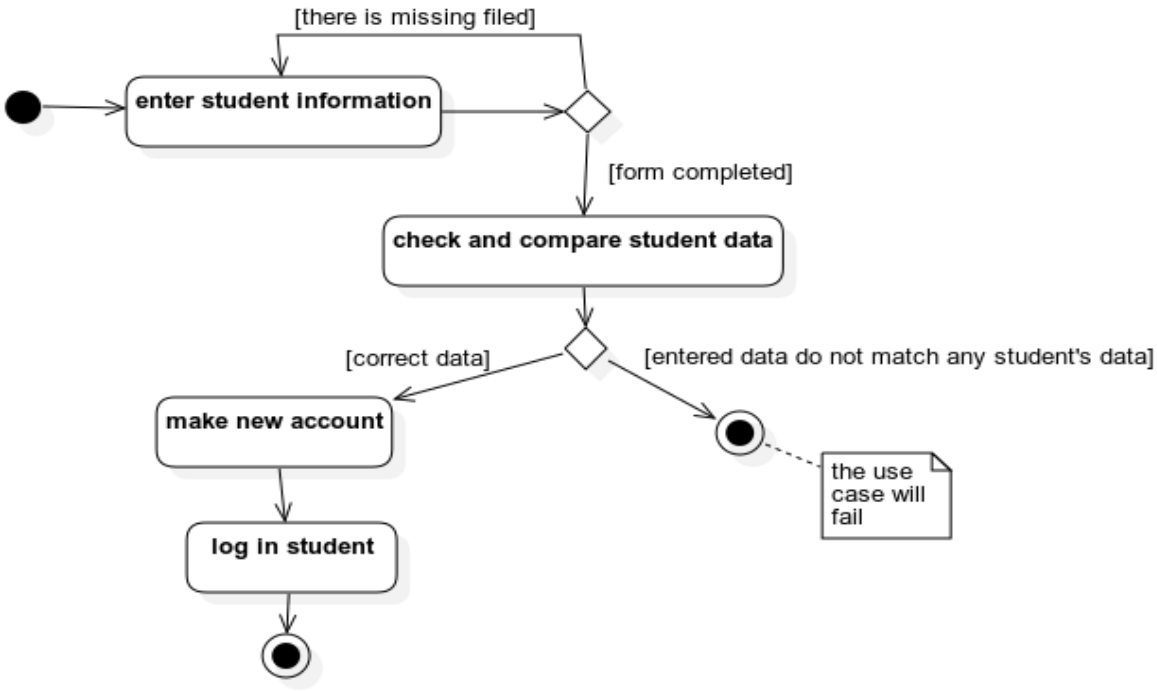


Figure 30 sprint#2 sign in activity

❖ Use case - Register a project:

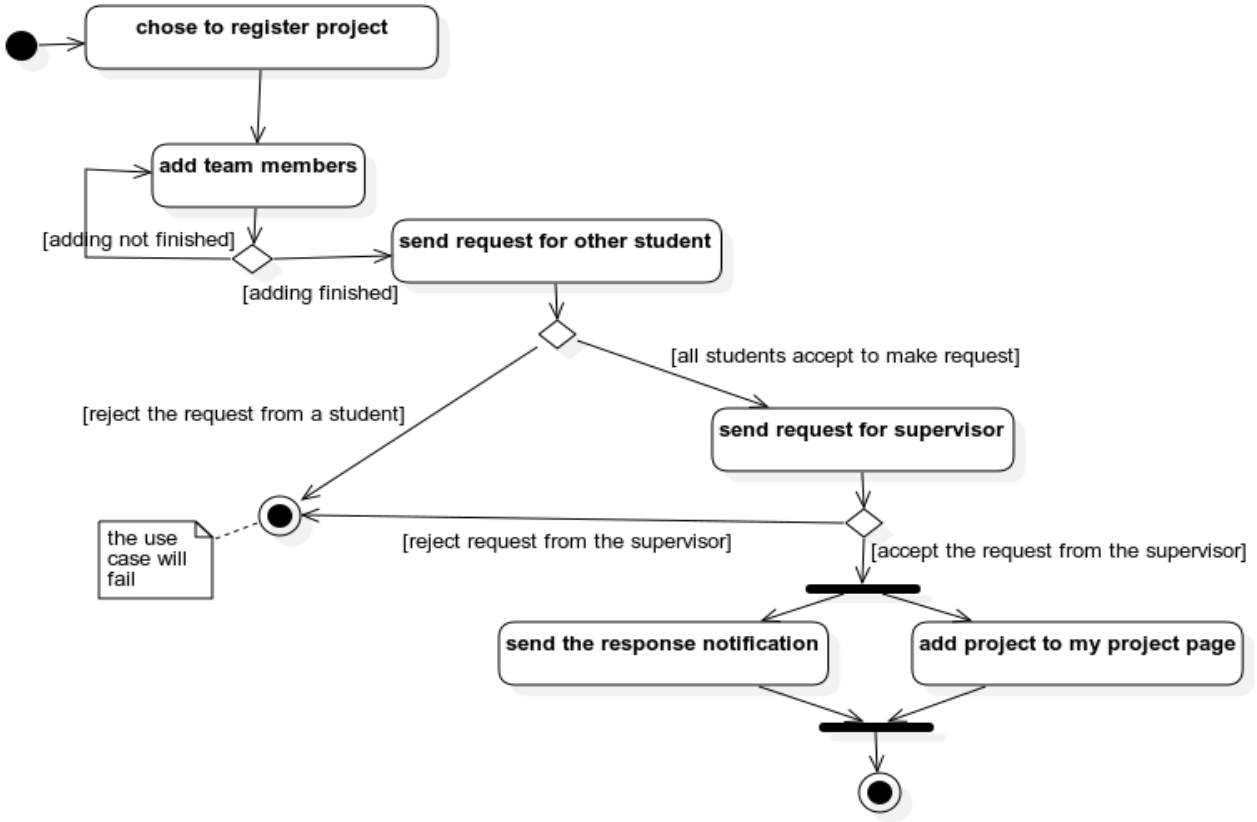


Figure 31 sprint#2 register a project activity

❖ Use case - Delete request:

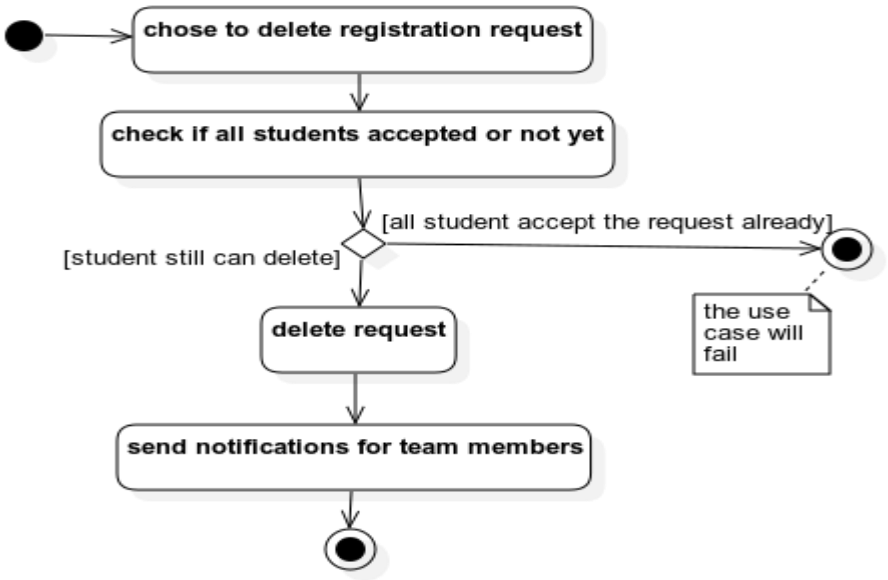


Figure 32 sprint#2 delete request activity

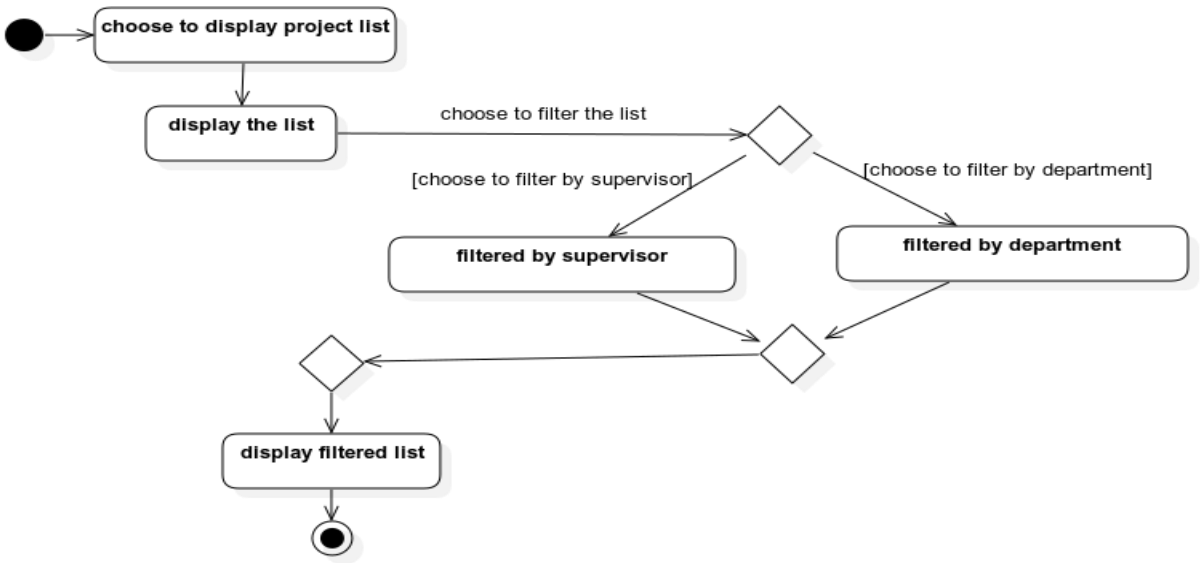


Figure 33 sprint#2 display registered projects activity

❖ Display registered projects list:

- sequence diagram:

❖ use case – delete request:

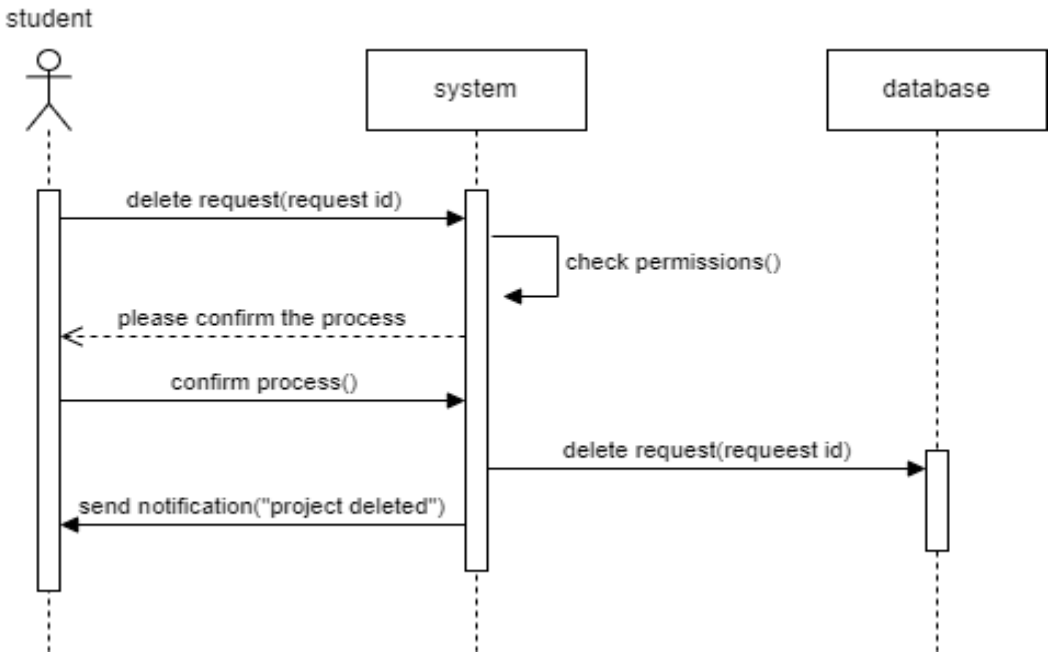


Figure 34 sprint#2 delete request

❖ use case – register project:

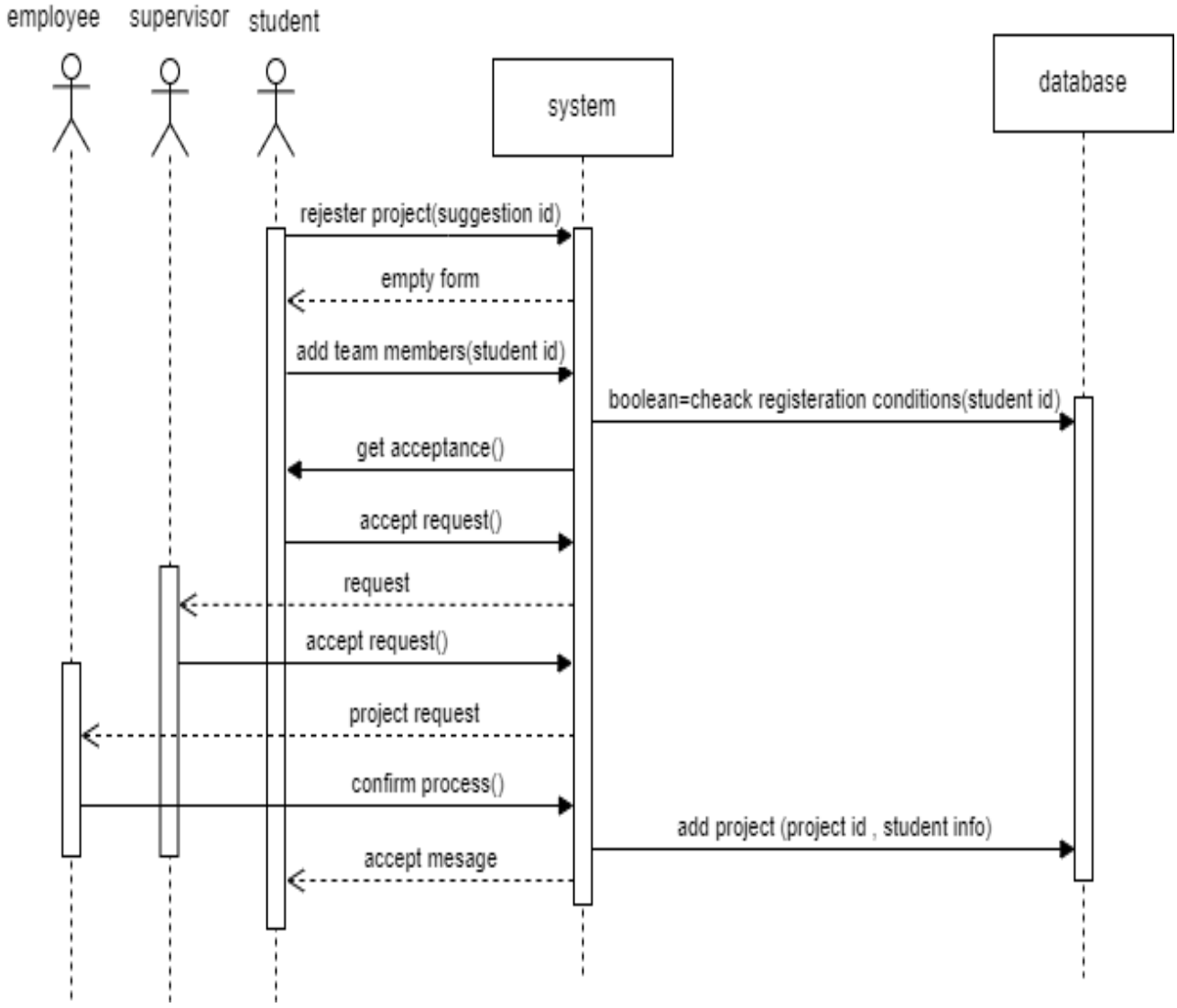


Figure 35 sprint#2 register a project sequence

❖ use case – sign in:

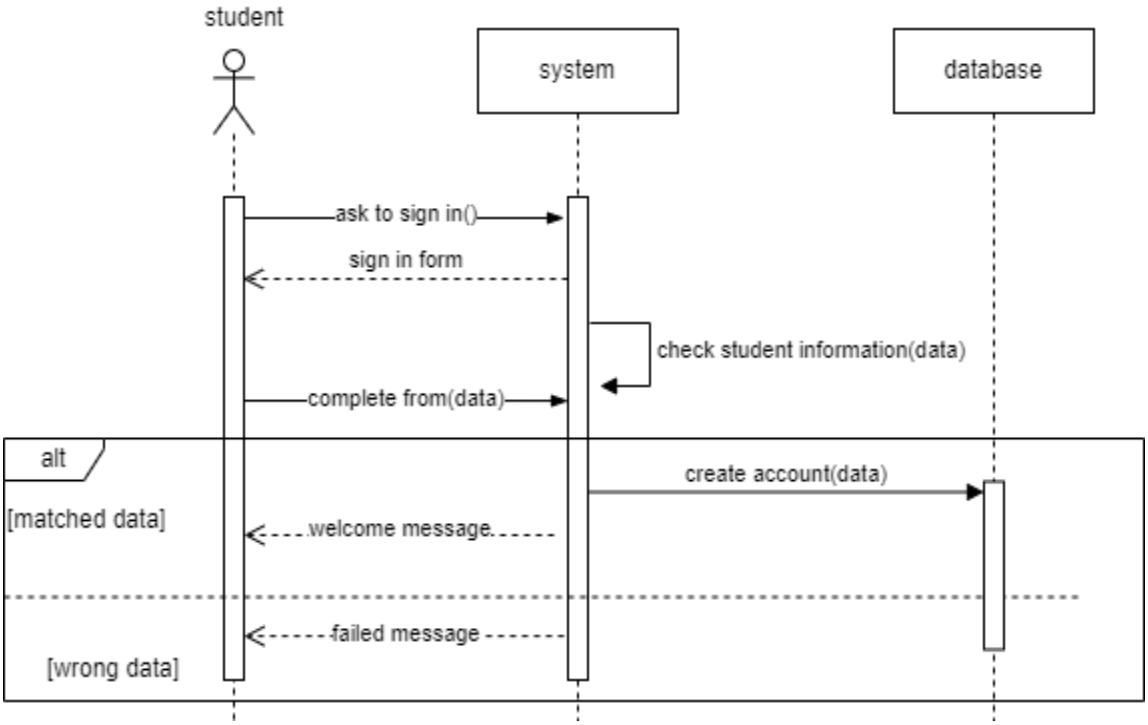


Figure 36 sprint#2 sign in sequence

❖ use case – display registered project list:

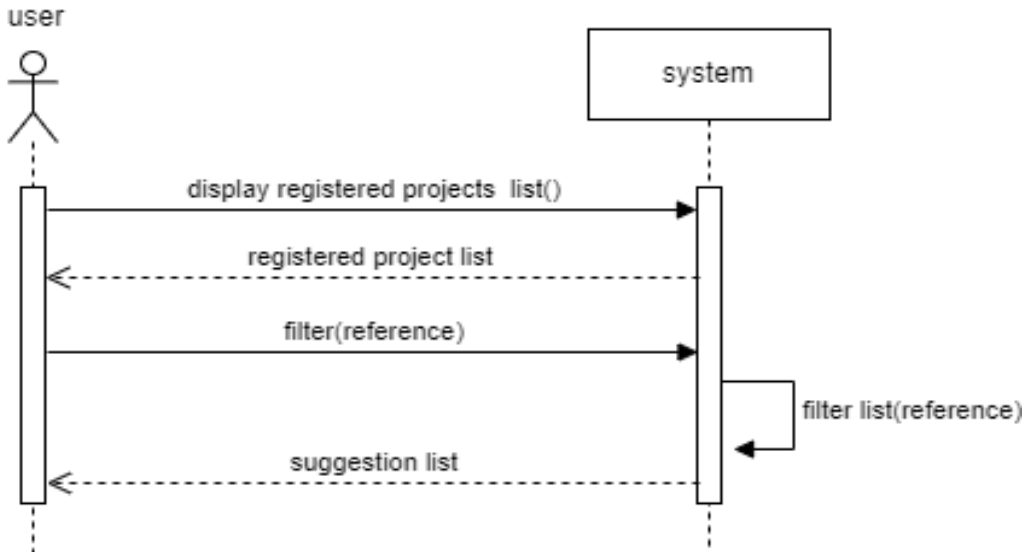


Figure 37 sprint#2 display registered projects sequence

- Class diagram:

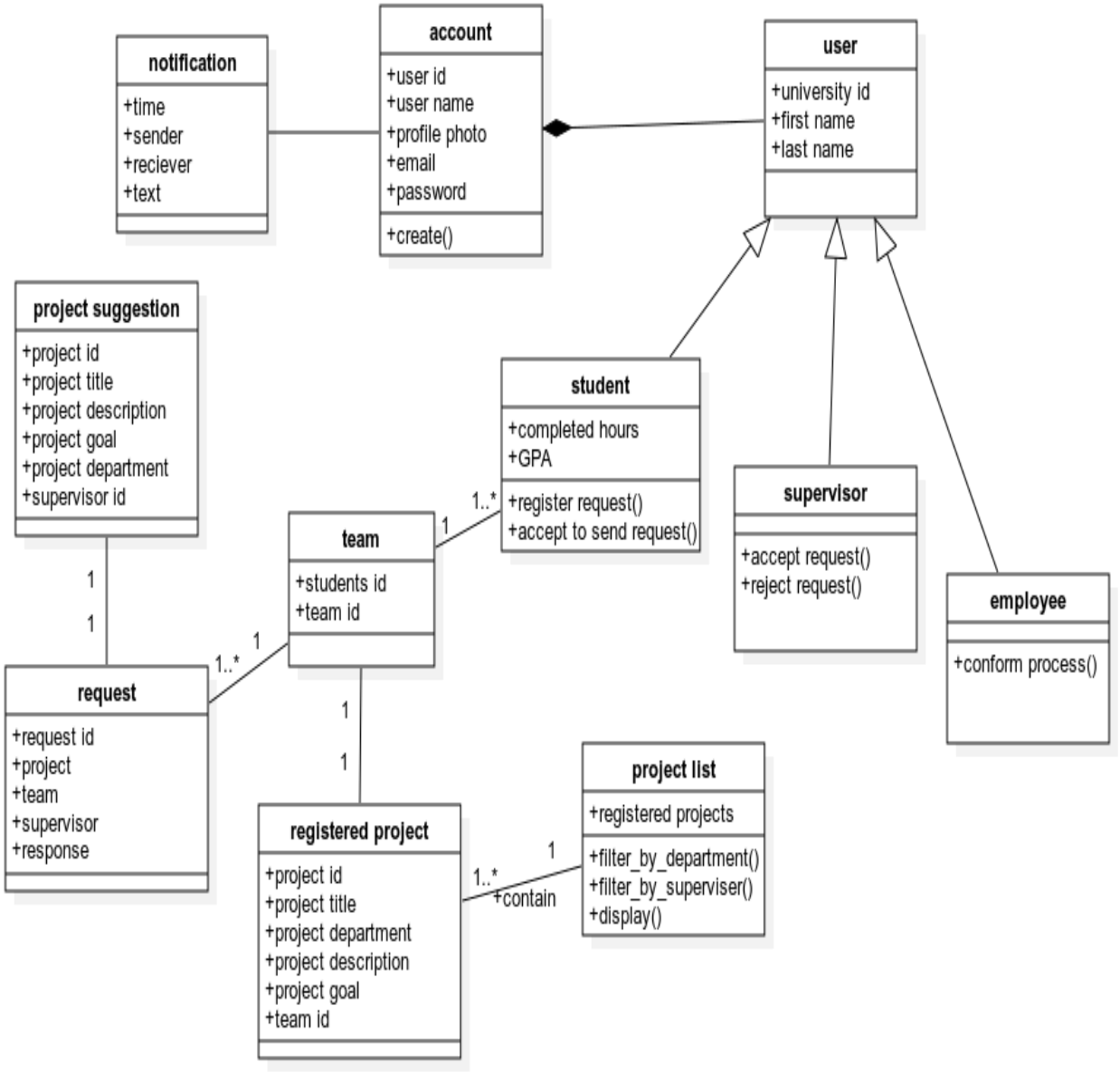


Figure 38 sprint#2 analysis class diagram

4. Initial Test Cases:

Table 16sprint#2 test cases

Test case scenario:		Sce-01: Check Sign in functionality.		
Test case id	Test case title	Req-id	Test steps	Expected result
Tc-01	Check results on entering vailed student data and press "create account".	Req-01 Req-02	1. Launch the application on the sign in page. 2. Complete the student form (ID, first name, last name, completed hours, GPA). 3. Press "create account".	The account must be added successfully and the system show "process competed successfully"
Tc-02	Check results on entering invalid student data (student not existence)	Req-01 Req-02	1. Launch the application on the sign in page. 2. Complete the student form (ID, first name, last name, completed hours, GPA). 3. Press "create account".	Error message "this student dose not existed"
Tc-03	Check results when a field of the student form is empty and the "create account" button is pressed.	Req-01 Req-02	1. Launch the application on the sign in page. 2. Enter some data. 3. Press "create account".	Error message "a field is missing"

Test case scenario:		Sce-02: Check project registration process functionality		
Test case id	Test case title	Req-id	Test steps	Expected result
Tc-04	Check results on applying project registration requests by students whom met the registration conditions.	Req-03 Req-04	1. Launch the application 2. Chose a project suggestion. 3. Press “apply”. 4. Add other students. 5. Press “confirm”	The system must show “the process completed” and send the request to other students to get their approval.
Tc-05	Check results on applying project registration requests by students, when one of them didn’t complete 100 hours.	Req-03 Req-04	1. Launch the application 2. Chose a project suggestion. 3. Press “apply”. 4. Add other students. 5. Press “confirm”	Error message “students don’t meet the conditions less than 100 hours by <student id>”.
Tc-06	Check results on applying project registration requests by students when, one of them didn’t complete “application course”	Req-03 Req-04	1. Launch the application 2. Chose a project suggestion. 3. Press “apply”. 4. Add other students. 5. Press “confirm”	Error message “application is not completed by <student id>”
Tc-07	Check results on applying project registration requests by students, when the difference of completed hours	Req-03 Req-04	1. Launch the application 2. Chose a project suggestion. 3. Press “apply”. 4. Add other students. 5. Press “confirm”	Error message” the difference between your hours more than 7”

	between then more than 7.			
Tc-08	Check result after all student of a team approve to send the request.	Req-05 Req-06	1. Launch the application by student. 2. Open request page. 3. Press “accept button” for the request	The system must send the request to the supervisor of the project, show “process completed”.
Tc-09	Check result when a request maker chose to “delete” request.	Req-07	1. Launch the application by student. 2. Open my-request page. 3. Press “delete button” for the request.	The system must delete the request from all student and send notifications with the updated.
Tc-10	Check result after the supervisor “accept” the request.	Req-08 Req-09 Req-10	1. Launch the application by supervisor. 2. Open “request page”. 3. Press “accept button”.	The system must inform the employee of the new project to register, and send notification to the student “project request accepted”.
Tc-11	Check result after the supervisor “reject” the request.	Req-08 Req-10	1. Launch the application by supervisor. 2. Open “request page”. 3. Press “reject button”.	the system must send notification to the student “project request had rejected”.

Test case scenario:		Sce-03: Check registered project list display functionality.		
Test case id	Test case title	Req-id	Test steps	Expected result
Tc-12	Check results by choosing "display registered projects list".	Req-11	1. Launch the application. 2. Choose "display registered project list".	All registered projects must be displayed in the list.
Tc-13	Check the result in choosing to filter the list by departments or supervisor.	Req-12	4. Launch the application. 5. Choose "display registered project list". 6. Choose "filter by departments" or "filter by supervisor".	The list must be sorted by the departments or supervisor and redisplay.

5. Updating requirements traceability matrix – sprint-2:

Req -id	Title	Analysis	Detailed design	coding	App user interface	Test cases
Req -01	the system must allow the students to make an account by their university ID (unique account).	Sp2an				Tc-01 Tc-02 Tc-03
Req -02	the system must be able to check if a student belongs to the university by comparing some entered data with the student data	Sp2an				Tc-01 Tc-02 Tc-03
Req -03	The system must allow students to request a project.	Sp2an				Tc-01 Tc-02 Tc-03
Req -04	The system must be able to check if a student and a team met the project's registration conditions	Sp2an				Tc-04 Tc-05 Tc-06
Req -05	The system must be able to get the acceptance of all team members for a request	Sp2an				Tc-08
Req -06	The system must be able to inform the supervisor about the requests made for his project suggestions	Sp2an				Tc-08
Req -07	The system must allow a student who request to delete his request.	Sp2an				Tc-09

Req -08	The system must allow supervisors to either accept or reject a project request.	Sp2an				Tc-10 Tc-11
Req -09	The system must inform the employee of the projects that are ready for registration.	Sp2an			Tc-10 Tc-11	
Req -10	the system must be able to inform the students if their project has been registered.	Sp2an			Tc-10	
Req -11	The system must be able to display registered project list.	Sp2an			Tc-12	
Req -12	The system must be able to display registered project list filtered by supervisors or departments.	Sp2an			Tc-13	

Sprint#2 design:

In this section, we will introduce the detailed design for the components of the second sprint, and database components.

1. Detailed class diagram:

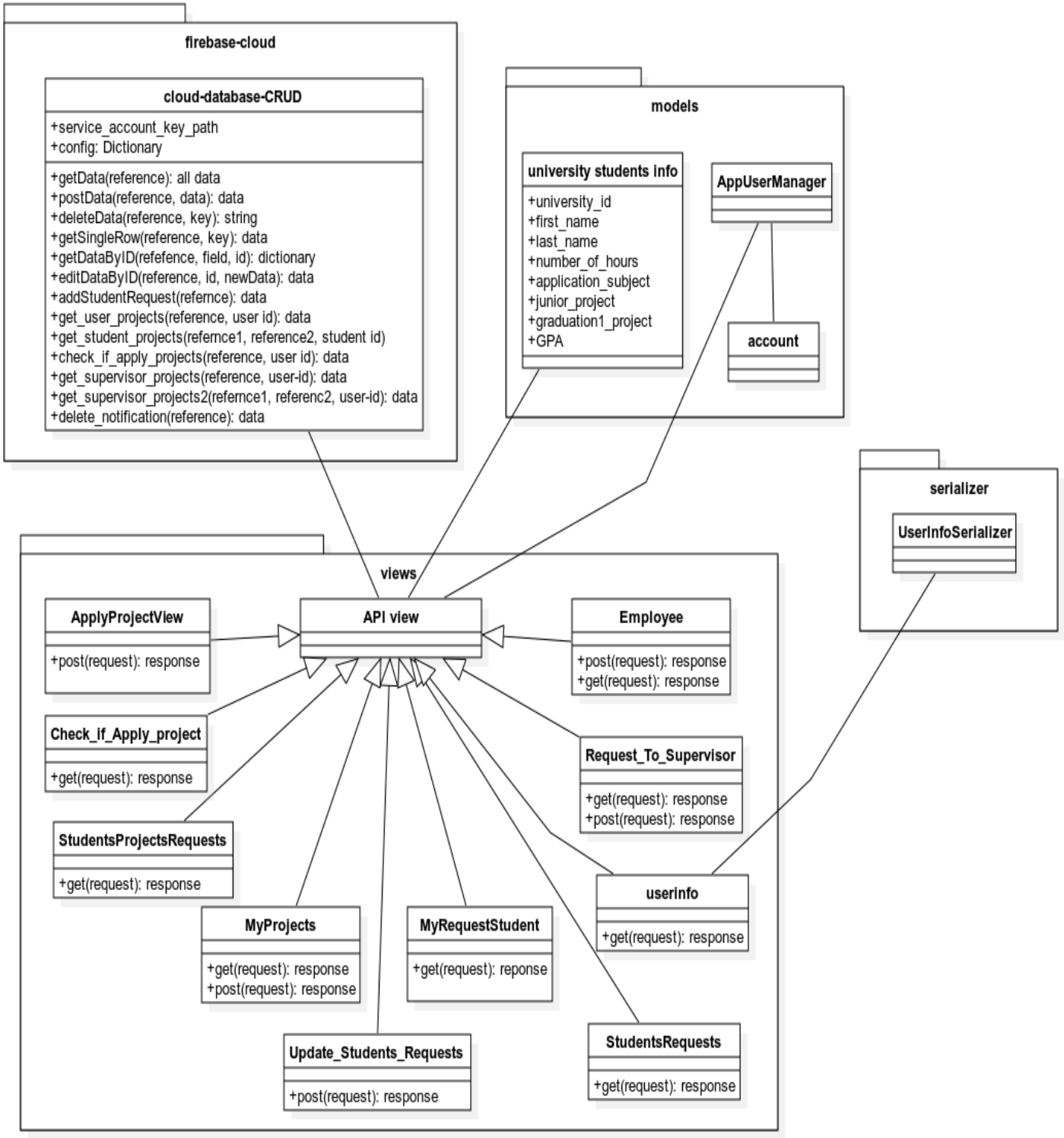


Figure 39 sprint#2 design class diagram

2. Database design:

A. Realtime database (NoSQL):

The updated database structure – json tree:

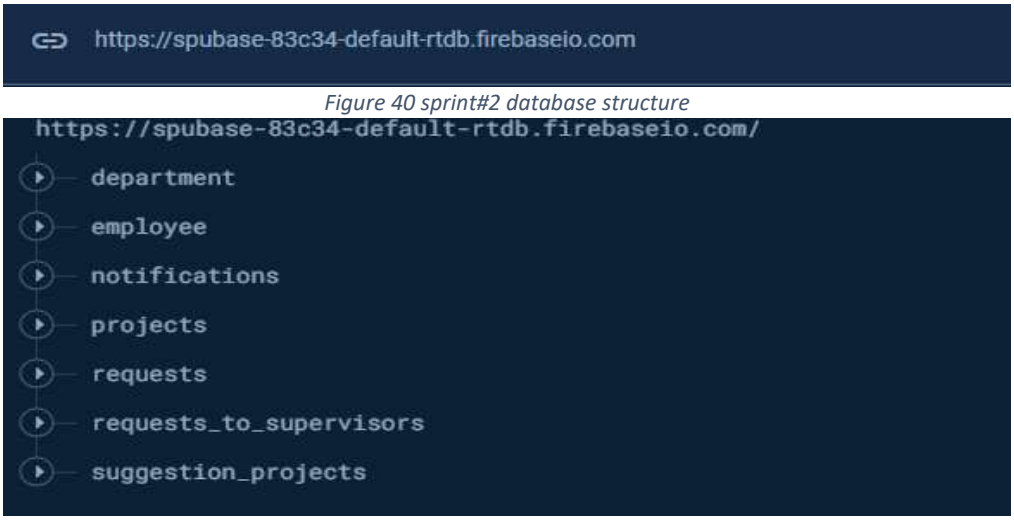
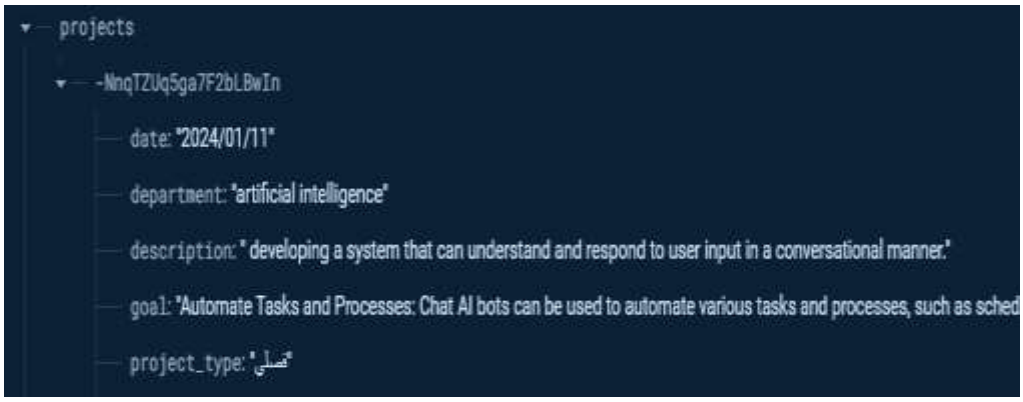
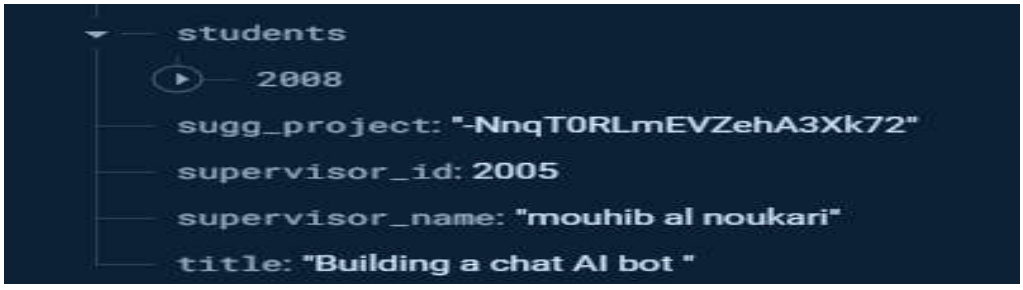


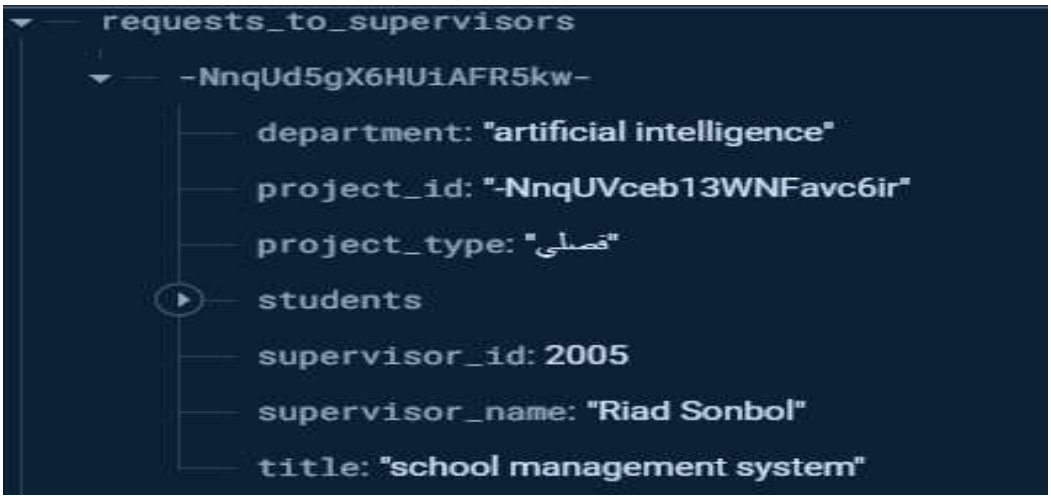
Figure 40 sprint#2 database structure

- Projects reference: store the registered projects.

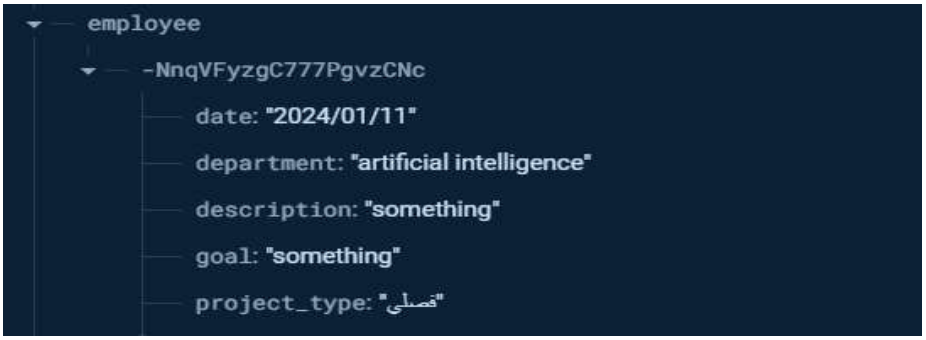


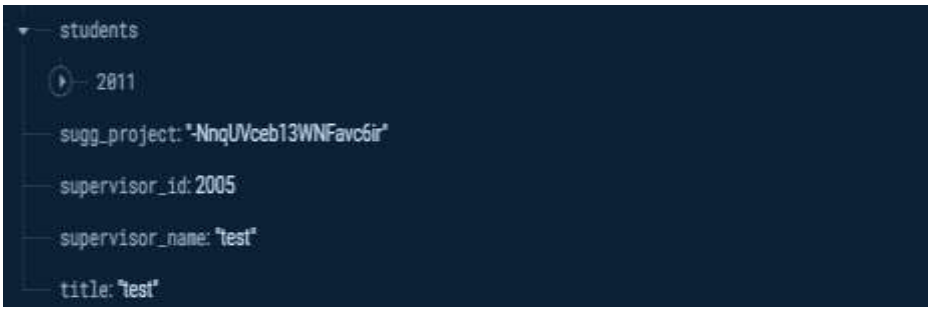


- Request to supervisor reference: store the requests made by students and send to a supervisor for registering a project.



- Employee reference: store the projects that ready to be registered, and sends to the employee.





B. Local database (university student data):

The system needs an updated data for student from the university, first to check if a student belongs to the university, second to check the project registrations conditions.

Table 17 sprint#2 university's students data

Student university data Database Table			
Field name	type	property	The input
University id	Integer Field	PK	admin
First name	Char field (255)		admin
Last name	Char field (255)		admin
Number of hours	Integer Field		admin
application project	Boolean Field		admin
Junior project	Boolean Field		admin
graduation1 project	Boolean Field		
GPA	Float field		

3. Site map update:

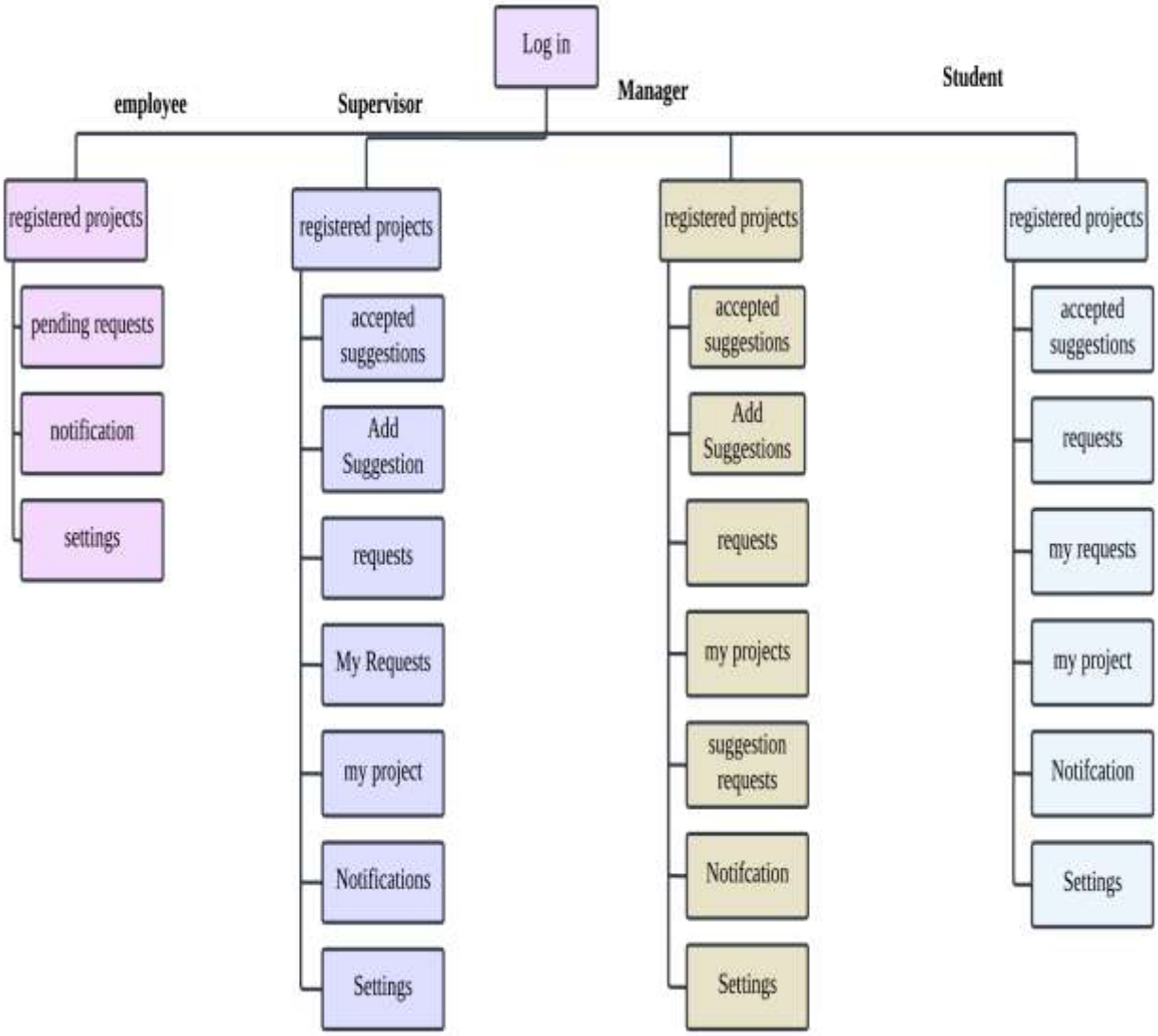


Figure 41sprint#2 updated site map

Sprint#2 implementation and testing

1. App interface:
    - ❖ Sign in interface:

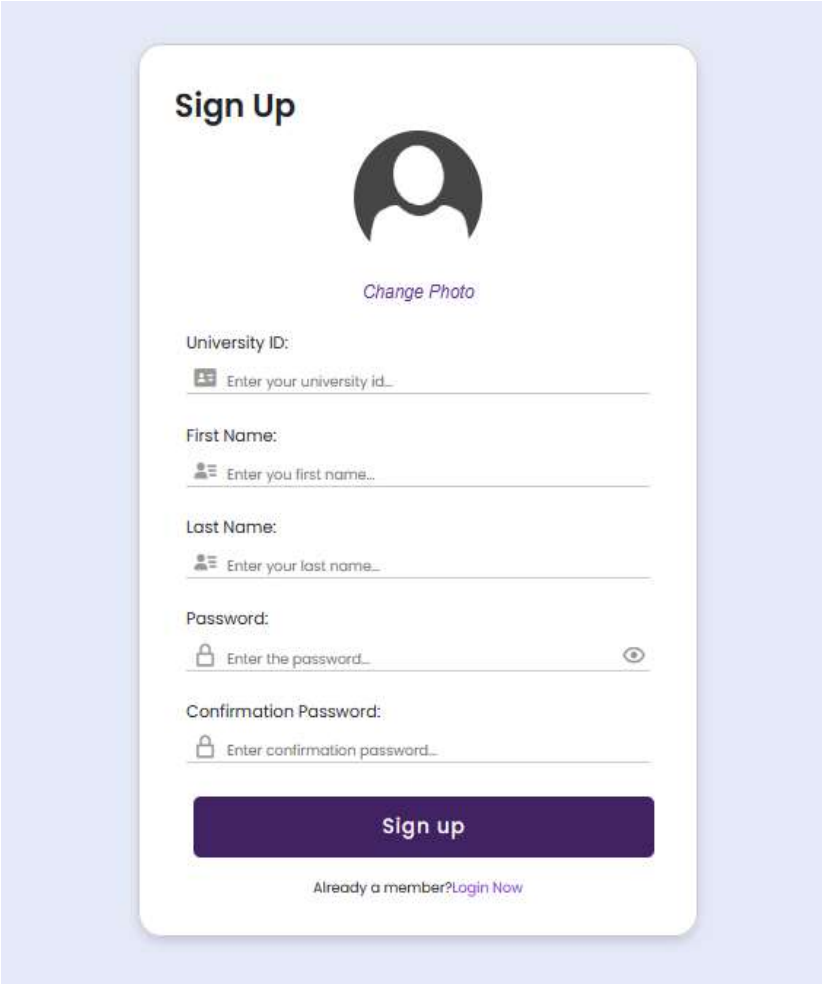


Figure 42 sprint#2 sign in interface inrf-01

❖ Register project (enter team member):

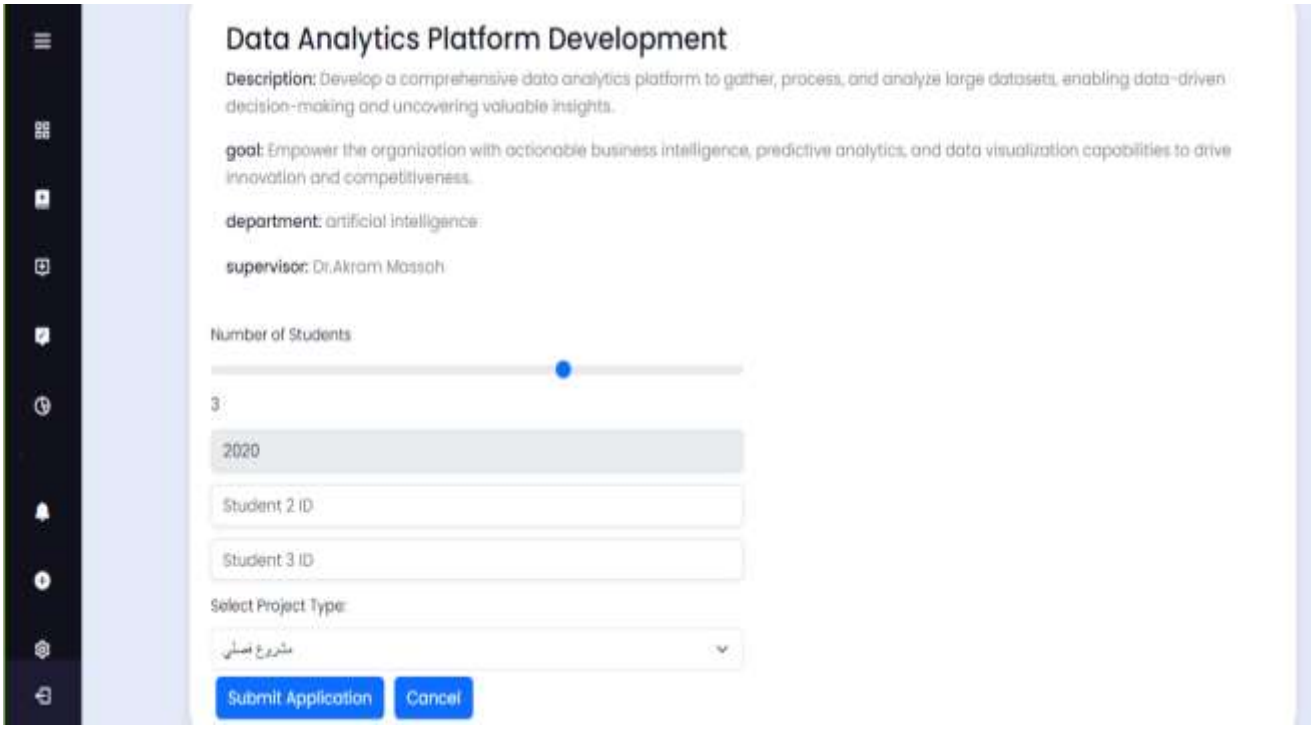


Figure 43 sprint#2 register project interface inrf-02

❖ Register project (request maker interface and delete interface):

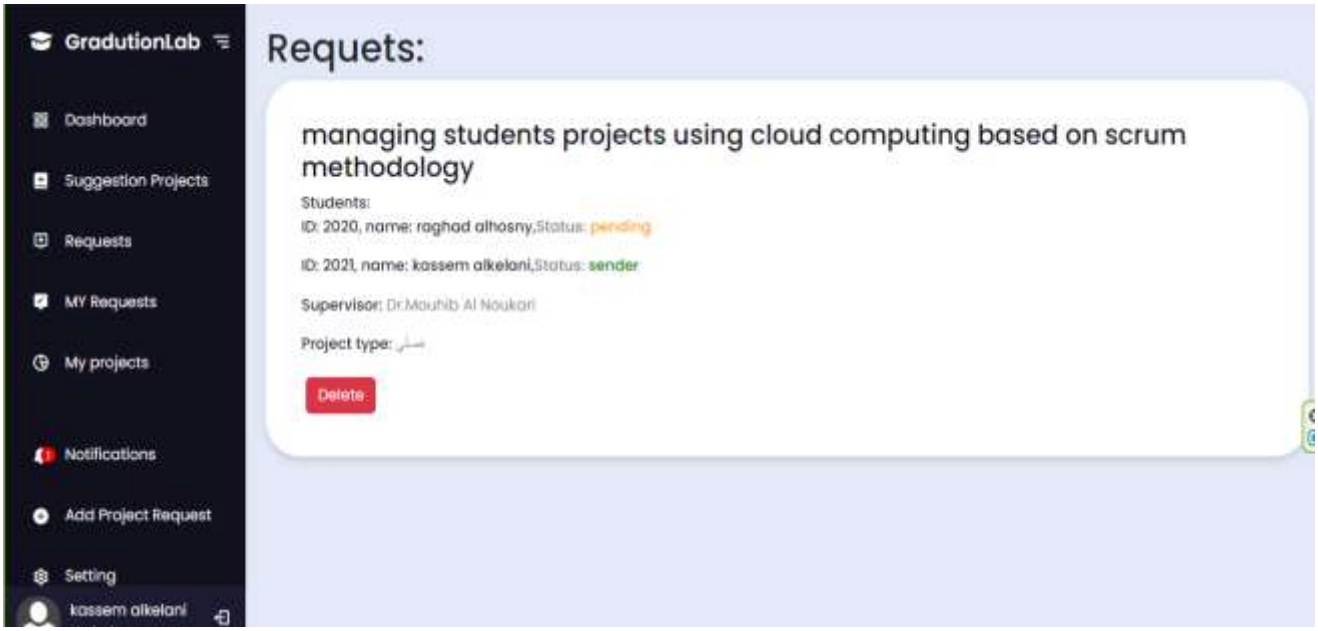


Figure 44 sprint#2 delete request interface inrf-03

❖ Register project (other students take acceptance interface):

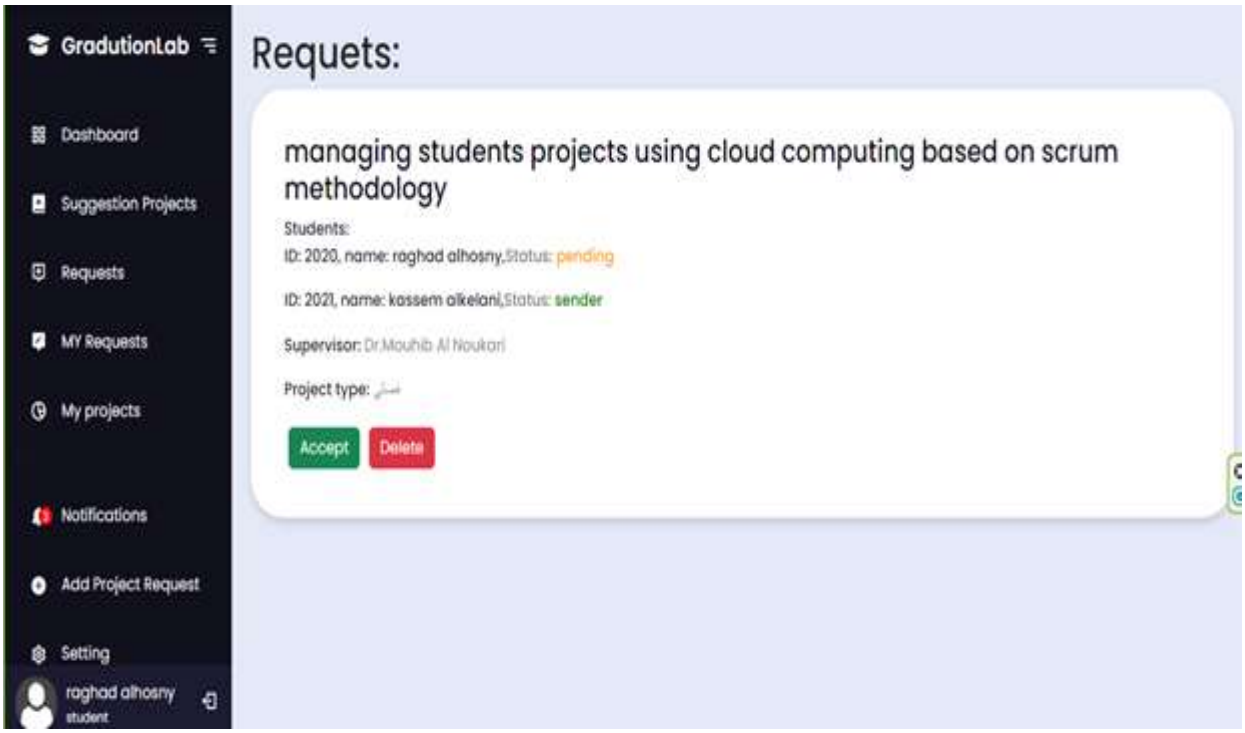


Figure 45 sprint#2 accepted request interface(student) inrf-04

❖ Supervisor interface (after all student accept to send the request):

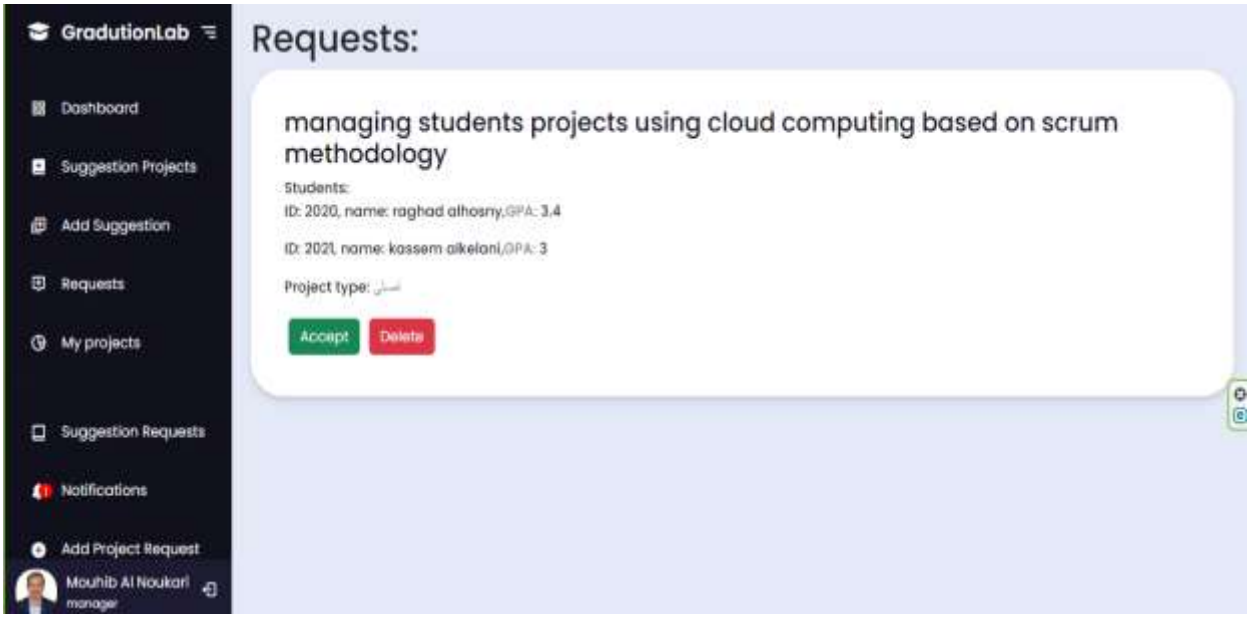


Figure 46 sprint#2 supervisor interface infr-05

❖ Employee notification interface (when the supervisor accepts the request):

Figure 47 sprint#2 notification interface inrf-06

❖ Employee interface after register the project successfully he will confirm the process:

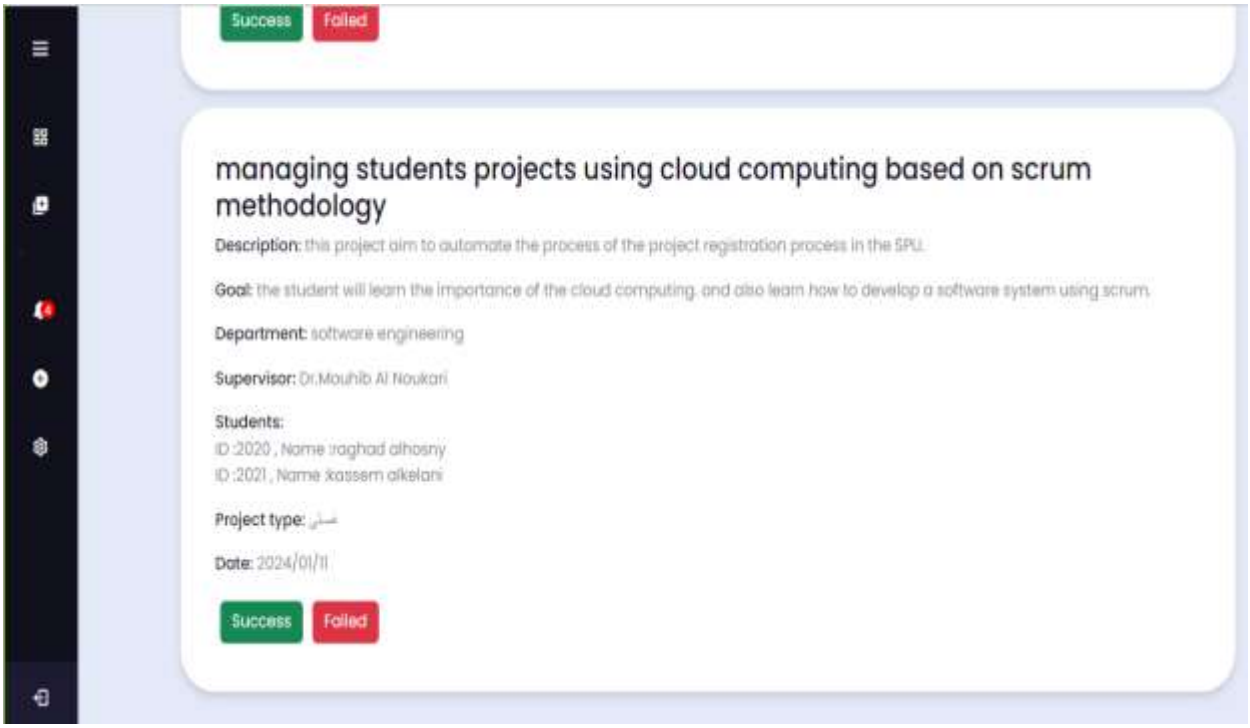


Figure 48 sprint#2 employee interface inrf-07

❖ Registered project list interface:

Figure 49 sprint#2 registered project list interface inrf-08

2. Test Cases execution

Table 18 sprint#2 test case execution

TC id	Test case title	Req-id	Tested data	Expected result	Actual result	Pass/ fail
Tc-01	Check results on entering vailed student data and press "create account".	Req-01 Req-02	University id=4200066 First name=" Raghad" Last name="al-hossny" Number-of-hours=113. Application=true.	The account must be added successfully and the system show "process competed successfully"	The account must be added successfully and the system show "process competed successfully"	Pass
Tc-02	Check results on entering invalid student data (student not existence)	Req-01 Req-02	University id=4200079 First name=" Kassem" Last name="al Kelani" Number-of-hours=113. Application=true.	Error message "this student dose not existed"	Error message "this student dose not existed"	Pass
Tc-03	Check results when a field of the student form is empty and the	Req-01 Req-02	University id=4200066 First name=" Raghad"	Error message "a field is missing"	Error message "a field is missing"	Pass

	"create account" button is pressed.		Last name= Number-of-hours=113. Application=true.			
Tc-04	Check results on applying project registration requests by students whom met the registration conditions.	Req-03 Req-04	Student1 id=4200066 Student2 id=4200079	The system must show "the process completed" and send the request to other students to get their approval.	The system must show "the process completed" and send the request to other students to get their approval.	Pass
Tc-05	Check results on applying project registration requests by students, when one of them didn't complete 100 hours.	Req-03 Req-04	Student1 id=4200066 Student2 id=4200065	Error message "students don't meet the conditions less than 100 hours by <student id>".	Error message "students don't meet the conditions less than 100 hours by <student id>".	Pass
Tc-06	Check results on applying project registration requests by students when, one of them didn't complete "application course"	Req-03 Req-04	Student1 id=4200066 Student2 id=4200064	Error message "application is not completed by <student id>"	Error message "application is not completed by <student id>"	Pass

Tc-07	Check results on applying project registration requests by students, when the difference of completed hours between then more than 7.	Req-03 Req-04	Student1 id=4200066 Student2 id=4200063	Error message" the difference between your hours more than 7"	Error message" the difference between your hours more than 7"	Pass
Tc-08	Check result after all student of a team approve to send the request.	Req-05 Req-06		The system must send the request to the supervisor of the project, show "process completed".	The system must send the request to the supervisor of the project, show "process completed".	Pass
Tc-09	Check result when a request maker chose to "delete" request.	Req-07		The system must delete the request from all student and send notifications with the updated.	The system must delete the request from all student and send notifications with the updated.	pass
Tc-10	Check result after the supervisor "accept" the request.	Req-08 Req-09 Req-10		The system must inform the employee of the new project to register, and send notification to	The system must inform the employee of the new project to register, and send notification to	pass

				the student "project request accepted".	the student "project request accepted".	
Tc-11	Check result after the supervisor "reject" the request.	Req-08 Req-10		the system must send notification to the student "project request had rejected".	the system must send notification to the student "project request had rejected".	Pass
Tc-12	Check results by choosing "display registered projects list".	Req-11		All registered projects must be displayed in the list.	All registered projects must be displayed in the list.	Pass
Tc-13	Check the result in choosing to filter the list by departments or supervisor.	Req-12		The list must be sorted by the departments or supervisor and redisplay.	The list must be sorted by the departments or supervisor and redisplay.	Pass

3. Final requirements traceability matrix – sprint2:

Req -id	Title	Analysis	Detailed design	Coding	App user interfaces	Test cases
Req -01	the system must allow the students to make an account by their university ID (unique account).	Sp2an	Sp2des	Sp2imp	Inrf-01	Tc-01 Tc-02 Tc-03
Req -02	the system must be able to check if a student belongs to the university by comparing some entered data with the student data	Sp2an	Sp2des	Sp2imp	Inrf-01	Tc-01 Tc-02 Tc-03
Req -03	The system must allow students to request a project.	Sp2an	Sp2des	Sp2imp	Inrf-02	Tc-01 Tc-02 Tc-03
Req -04	The system must be able to check if a student and a team met the project's registration conditions	Sp2an	Sp2des	Sp2imp	Inrf-02	Tc-04 Tc-05 Tc-06
Req -05	The system must be able to get the acceptance of all team members for a request	Sp2an	Sp2des	Sp2imp	Inrf-04	Tc-08
Req -06	The system must be able to inform the supervisor about the requests made for his project suggestions	Sp2an	Sp2des	Sp2imp	Inrf-05	Tc-08

Req -07	The system must allow a student who request to delete his request.	Sp2an	Sp2des	Sp2imp	Inrf-03	Tc-09
Req -08	The system must allow supervisors to either accept or reject a project request.	Sp2an	Sp2des	Sp2imp	Inrf-05	Tc-10 Tc-11
Req -09	The system must inform the employee of the projects that are ready for registration.	Sp2an	Sp2des	Sp2imp	Inrf-07	Tc-10 Tc-11
Req -10	the system must be able to inform the students if their project has been registered.	Sp2an	Sp2des	Sp2imp	Inrf-06	Tc-10
Req -11	The system must be able to display registered project list.	Sp2an	Sp2des	Sp2imp	Inrf-08	Tc-12
Req -12	The system must be able to display registered project list filtered by supervisors or departments.	Sp2an	Sp2des	Sp2imp	Inrf-08	Tc-13

Sprint #3

Sprint #3 analysis:

In this section, we will introduce the analytical study for the third sprint using the needed UML diagrams.

1. Sprint backlog:

   The functional requirement list we will complete for this sprint:

   - ✓ Req-01: the system must allow the admin to make an account for the supervisor, manager, and employee by a unique ID and password.
   - ✓ Req-02: The system must allow students to make a new project suggestion and send it to a supervisor they choose.
   - ✓ Req-03: The system must allow the manager to set a supervisor as the head of the evaluation process for a specific department.
   - ✓ Req-04: The system must allow the manager and the head of the evaluation process to set an advertisement.
   - ✓ Req-05: the system must allow the head of the evaluation team and the manager to upload files with an advertisement.
   - ✓ Req-06: The system must log all the events that occur on the system.

The non-functional requirement list we will complete for this sprint:

- ✓ Req-01: The system must be user-friendly
- ✓ Req-02: the system must be secure.

2. Initial Requirements traceability Matrix

Req-id	Title	Analysis	Detailed design	coding	App user interfaces	Test cases
Req-01	The system must allow the admin to make an account for the supervisor, manager, and employee by a unique ID and password.					
Req-02	The system must allow students to make a new project suggestion and send it to a supervisor they choose.					
Req-03	The system must allow the manager to set a supervisor as the head of the evaluation process for a specific department.					
Req-04	The system must allow the manager and the head of					

	the evaluation process to set an advertisement.					
Req-05	The system must allow the head of the evaluation team and the manager to upload files with an advertisement.					
Req-06	The system must log all the events that occur on the system.					

3. Requirements modeling:

- Use case diagram:

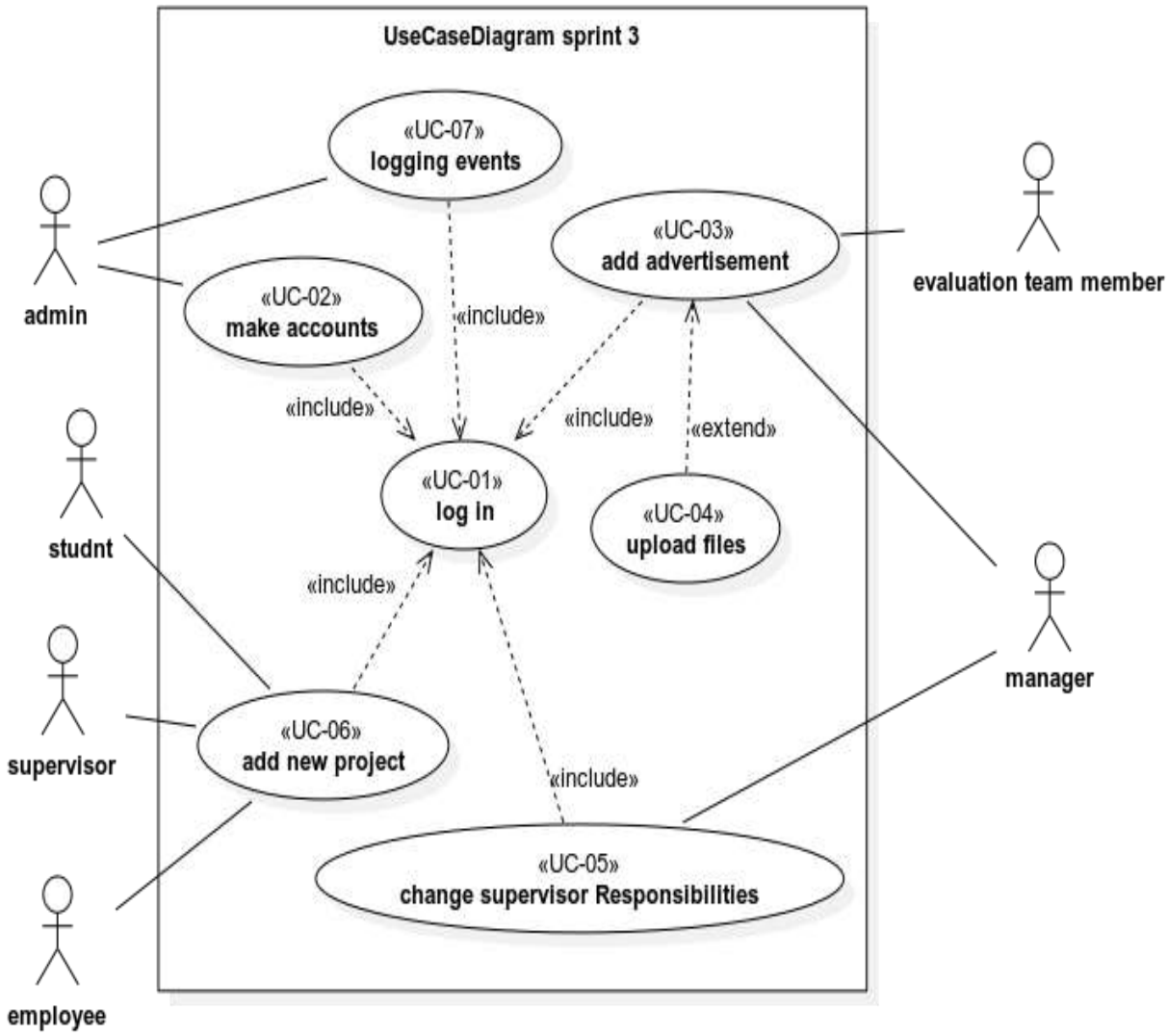


Figure 50 sprint#3 use case

- Use case specification:

Use case name:	Make accounts
Participating Actors:	initiated by: admin
The flow of events:	1. The admin first chose to make new account (add user). 2. The system will show the form of adding account. 3. The admin will complete the form fields. 4. The system will check the entered information and asks the admin to determine the account kind (supervisor, manager, employee) 5. The admin will determine the account type. 6. The system will add the account successfully with the university id the admin choice and the strong password.
Entry condition	The admin had logged in
Exit conditions	New account added.

Use case name:	Change supervisor responsibilities
Participating Actors:	initiated by: manager
The flow of events:	1. The manager chose to set a supervisor as an evaluation team member. 2. The system will show the supervisors list for the manager. 3. The manager will choose a supervisor and press “add”. 4. The system will add new responsibilities to the selected supervisor like adding advertisements to be shown for all users.
Entry condition	The admin had logged in

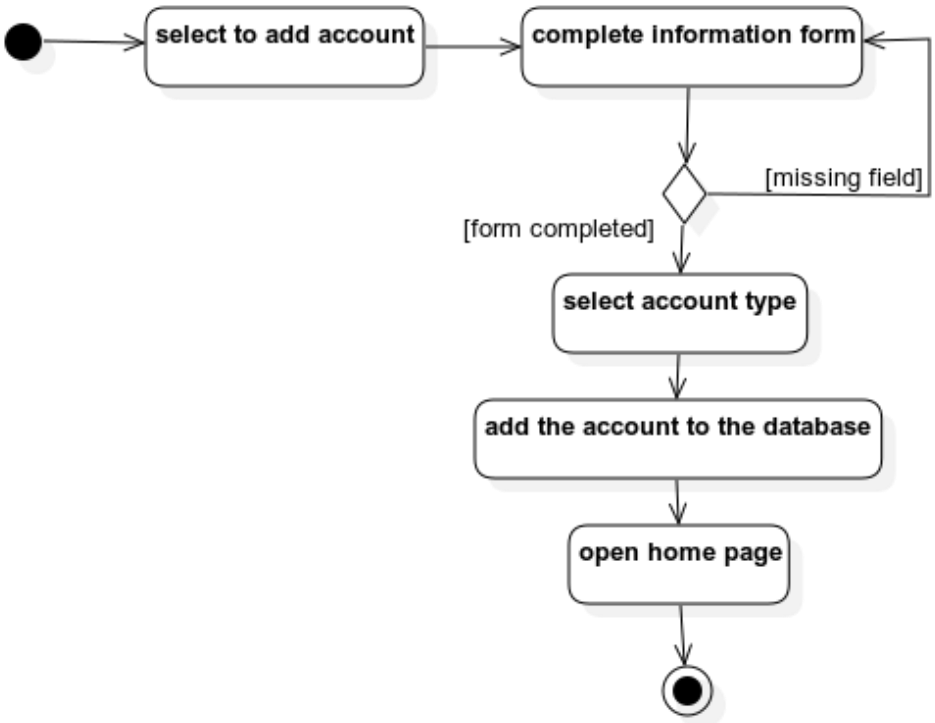
Exit conditions	Supervisor has the evaluation team responsibilities.
Use case name:	Add advertisements
Participating Actors:	initiated by: manager, evaluation team
The flow of events:	1. User choose to add new advertisement. 2. The system will show the form of adding advertisement. 3. The user will enter the title he wants to be shown for the advertisement and access by all users. 4. If the user chooses to "upload file" with the advertisement. 5. The system will ask the user to choose a file from local storage. 6. The user will choose a file. 7. The system will upload it and make it seen for all users.
Entry condition	The user had logged in
Exit conditions	Advertisement shown in the advertisement page for all users.

Use case name:	Add new project
Participating Actors:	initiated by: student participant: supervisor, employee
The flow of events:	1. The student will choose to add new project. 2. The system will show project form. 3. The student will complete the form and adding his team member. 4. The system will check registration conditions and get acceptance from all other students. 5. System will send the request for the supervisor added. 6. The supervisor will accept the request. 7. Employee will confirm the process.

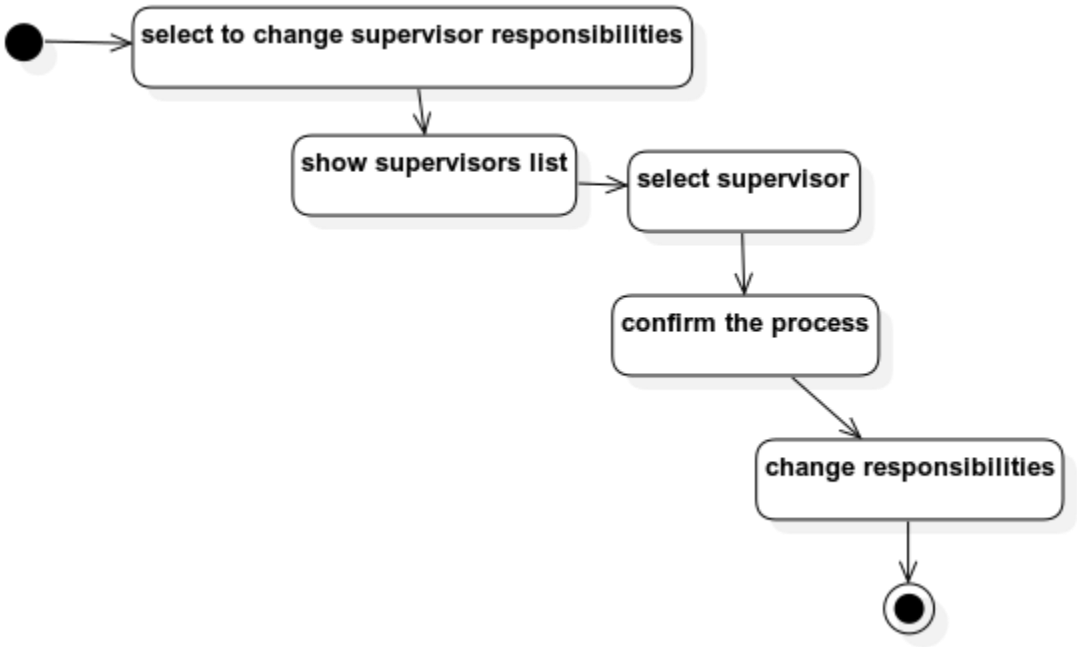
	8. The system will send a notification for all students about the result.
Entry condition	All students logging in
Exit conditions	New project registered.

Use case name:	Logging events
Participating Actors:	initiated by: all users
The flow of events:	1. When any user does an event against the database. 2. The system will register that event with information about it (user that make that event, date, the action that accrue) 3. The system will add event to the "logging" page for the admin account.
Entry condition	The admin had logged in
Exit conditions	All events registered.

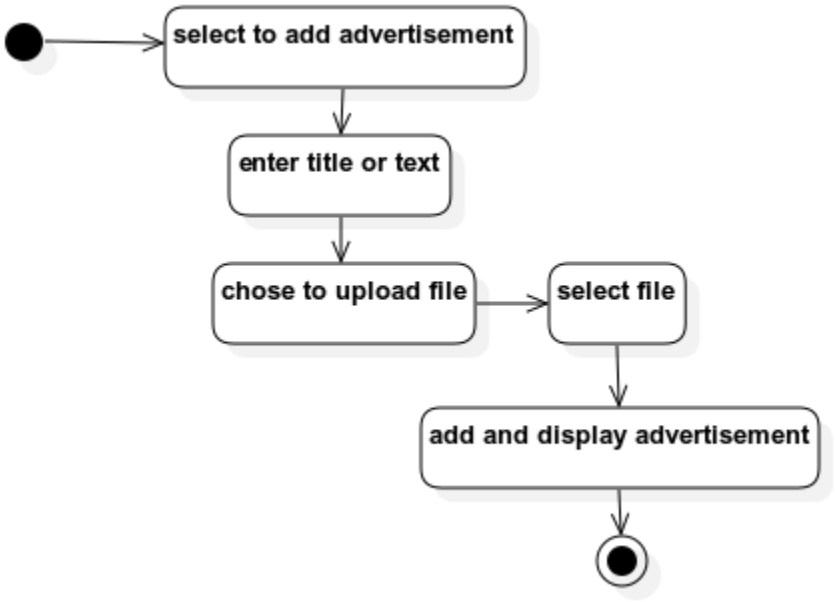
- Activity diagram:
  - Use case - Make accounts



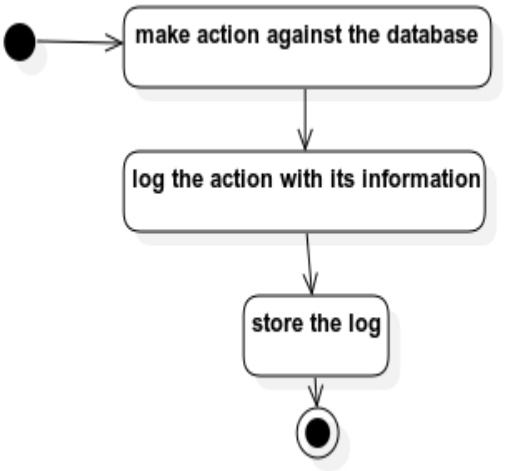
  - Use case - Change supervisor responsibilities



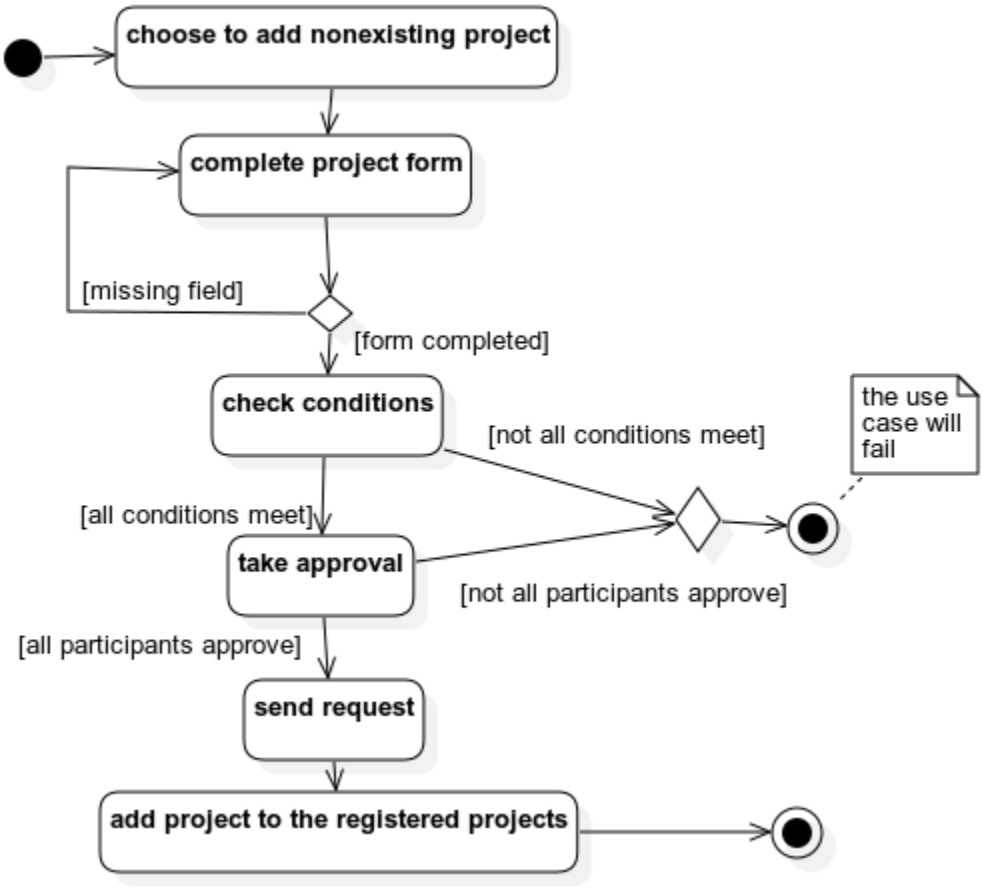
- Use case - Add advertisements



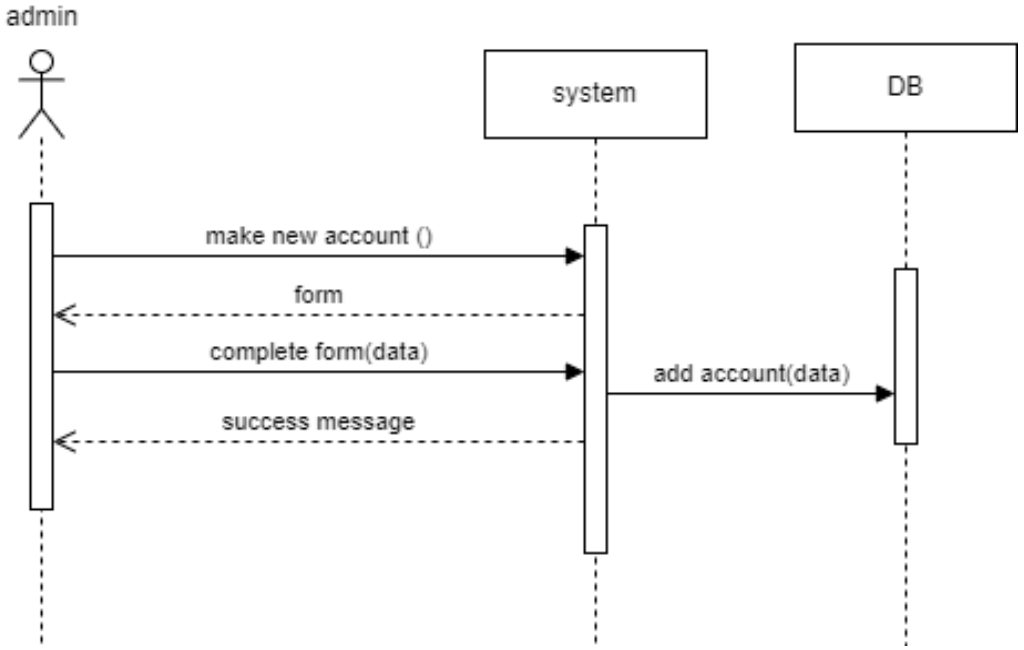
- Use case - Logging events:



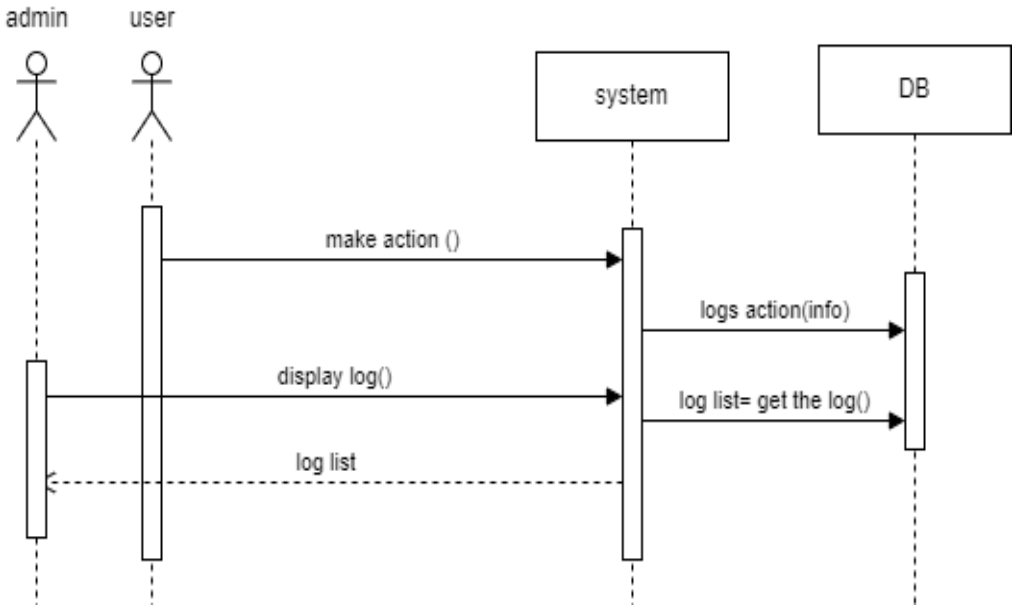
- Use case - Add new project



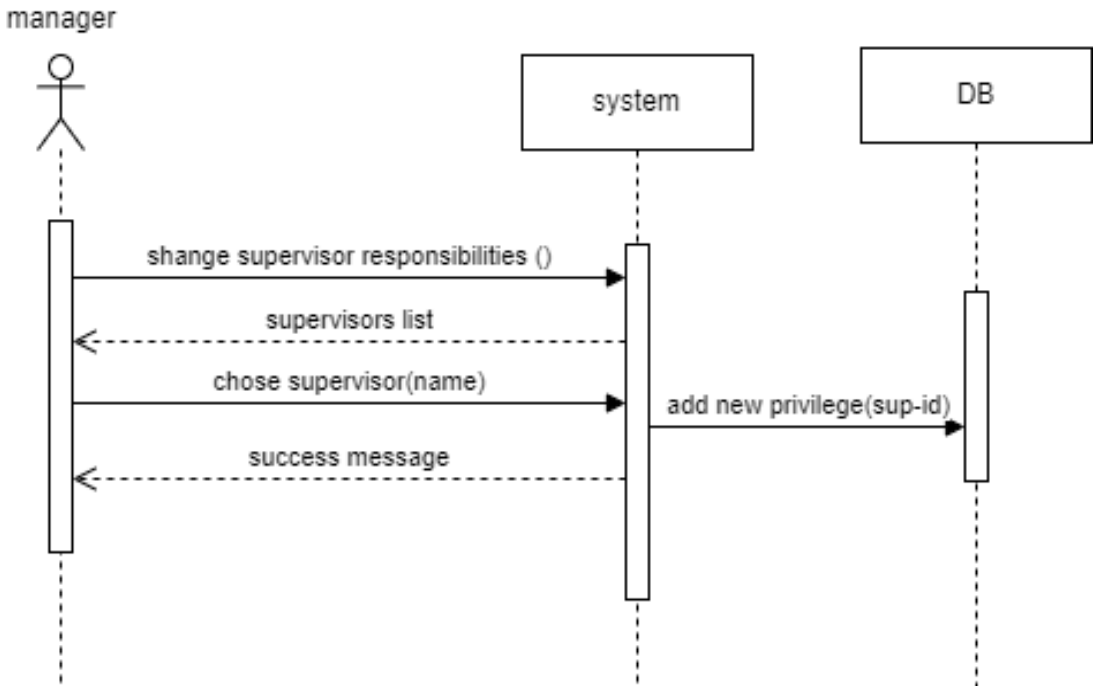
- Sequence diagram:
  - Use case - make accounts:



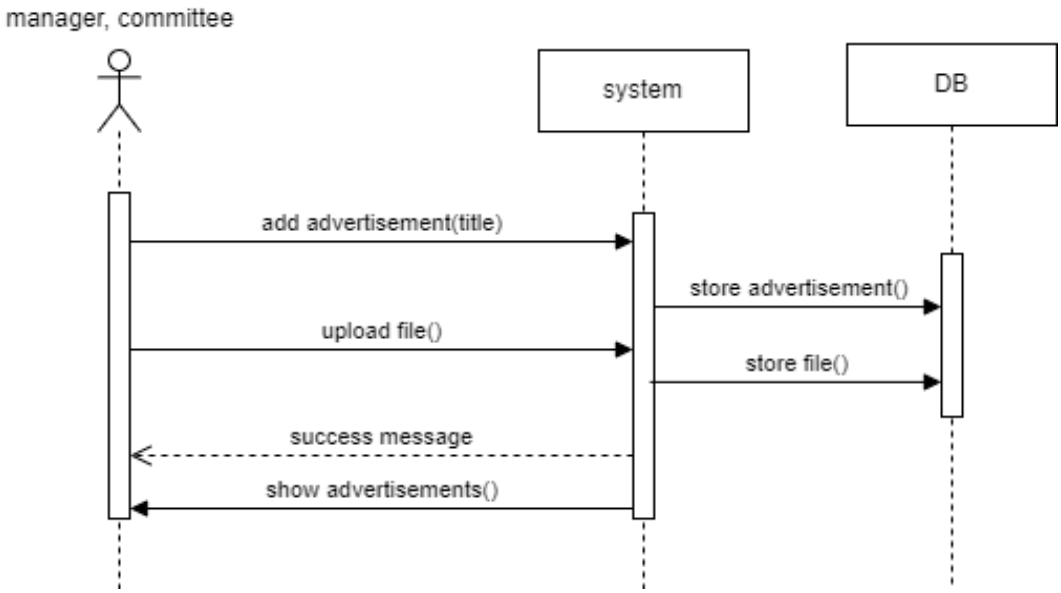
  - Use case – logging events:



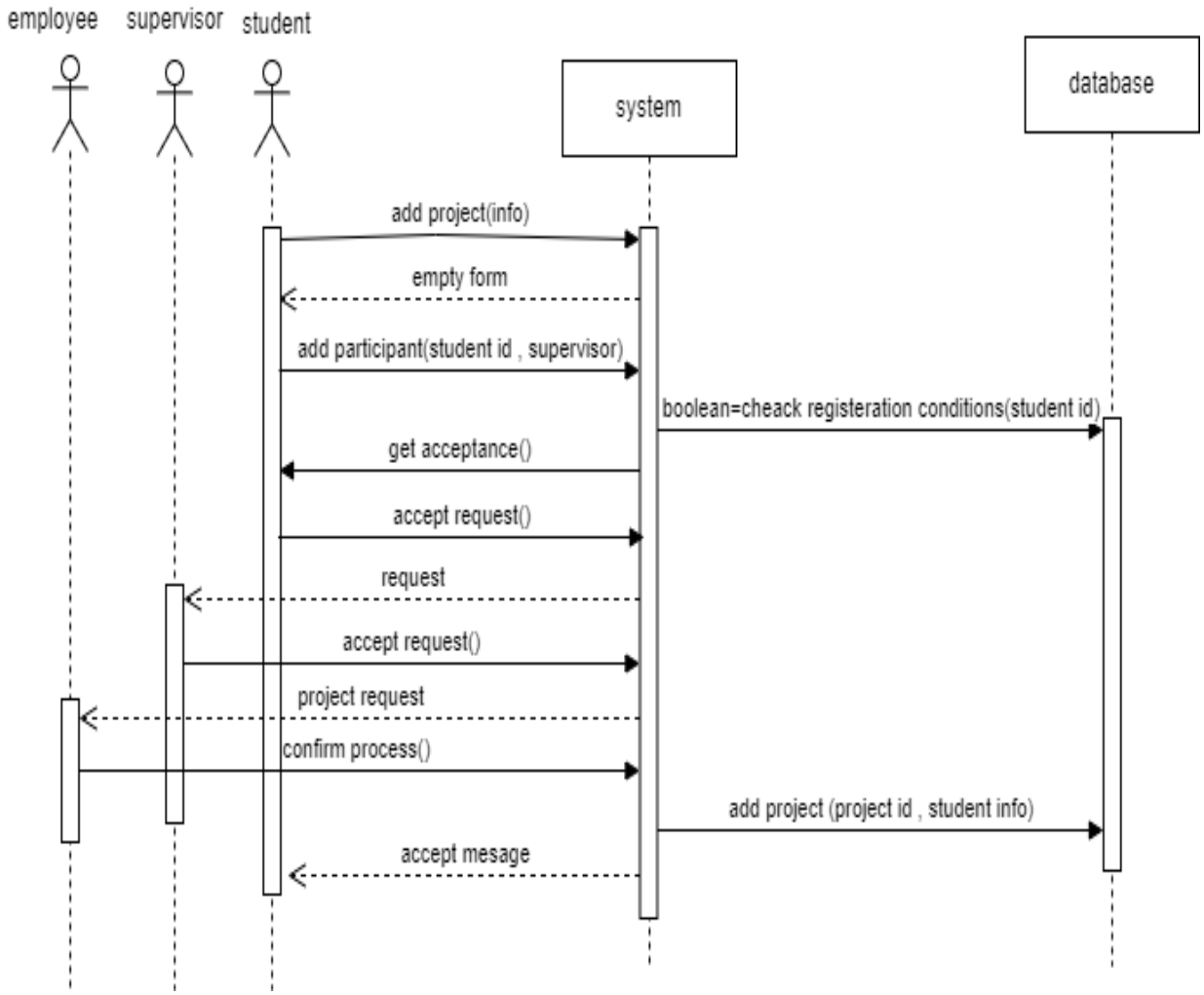
- Use case – change responsibilities:



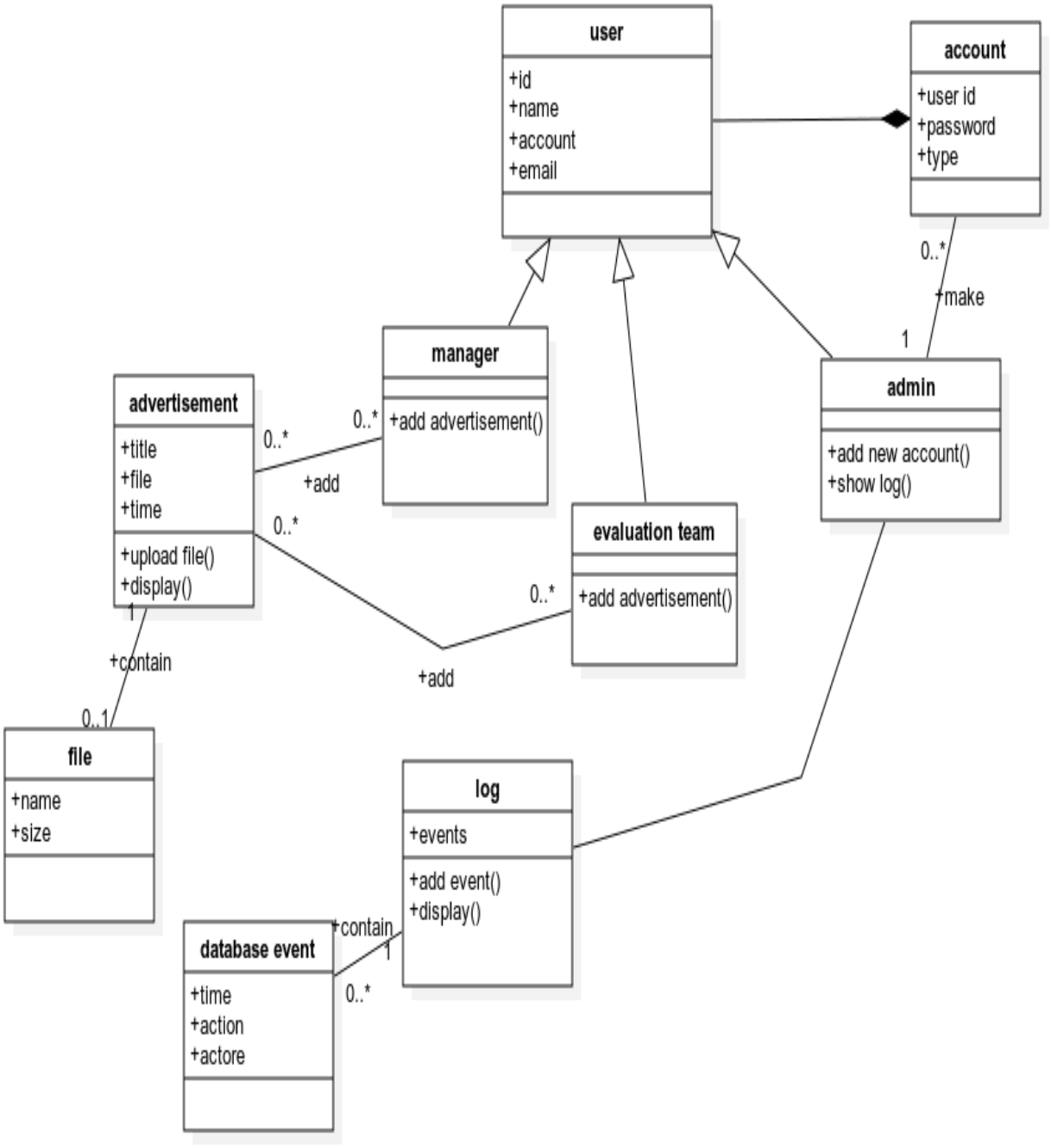
- Use case – add advertisement:



- Use case – add project request:



- Class diagram:



4. Initial Test cases:

Test case scenario:		Sce-01: Check creates new account functionality		
Test case id	Test case title	Req-id	Test steps	Expected result
Tc-01	Check results on choosing a vailed user ID and strong password	Req-01	1. Launch the application by the admin. 2. Choose to create a new account. 3. Choose the type of the account. 4. Enter ID and password. 5. Choose “create”.	Account successfully created.
Tc-02	Check results on choosing an ID that already exists with a strong password	Req-01	1. Launch the application by the admin. 2. Choose to create a new account. 3. Choose the type of the account. 4. Enter ID and password. 5. Choose “create”.	Error message “ID already exists
Tc-03	Check results on choosing a password that is not strong enough.	Req-01	1. Launch the application by the admin. 2. Choose to create a new account. 3. Choose the type of the account. 4. Enter ID and password. 5. Choose “create”.	Error message “password is not strong enough”.

Test case scenario:		Sce-01: Check adding advertisement functionality		
Test case id	Test case title	Req-id	Test steps	Expected result
Tc-04	Check results on choosing to add evaluation team member.	Req-03	1. Launch the application by the manager. 2. Choose to “add committee”. 3. Choose the supervisor from supervisors list. 4. Press “add”.	The selected supervisor had the committee privileges.
Tc-05	Check results on choosing to add new advertisement.	Req-04	1. Launch the application by the manager or committee member. 2. Choose to “add advertisement”. 3. Enter title. 4. Press “add”.	Advertisement must successfully add to the advertisement list to be display for every user.
Tc-06	Check results on choosing to add advertisement with file.	Req-05	1. Launch the application by the manager or committee member. 2. Choose to “add advertisement”. 3. Enter title. 4. Choose to upload file 5. Select file. 6. Press “add”.	Advertisement must successfully add to the advertisement list to be display for every user.

Test case scenario:		Sce-01: Check logging events functionality		
Test case id	Test case title	Req-id	Test steps	Expected result
Tc-07	Check results on making any action against the database (add suggestion,) by any user.	Req-06	1. Launch the application by any user 2. Make an action against the database.	The action must be logged, and add to the database.
Tc-08	Check results on choosing "display logging" by admin	Req-06	1. Launch the application by the admin. 2. Select "display log".	Log must be displayed for the admin each log with its information.

5. Updating requirements traceability matrix:

Table 19RTM SPRINT3

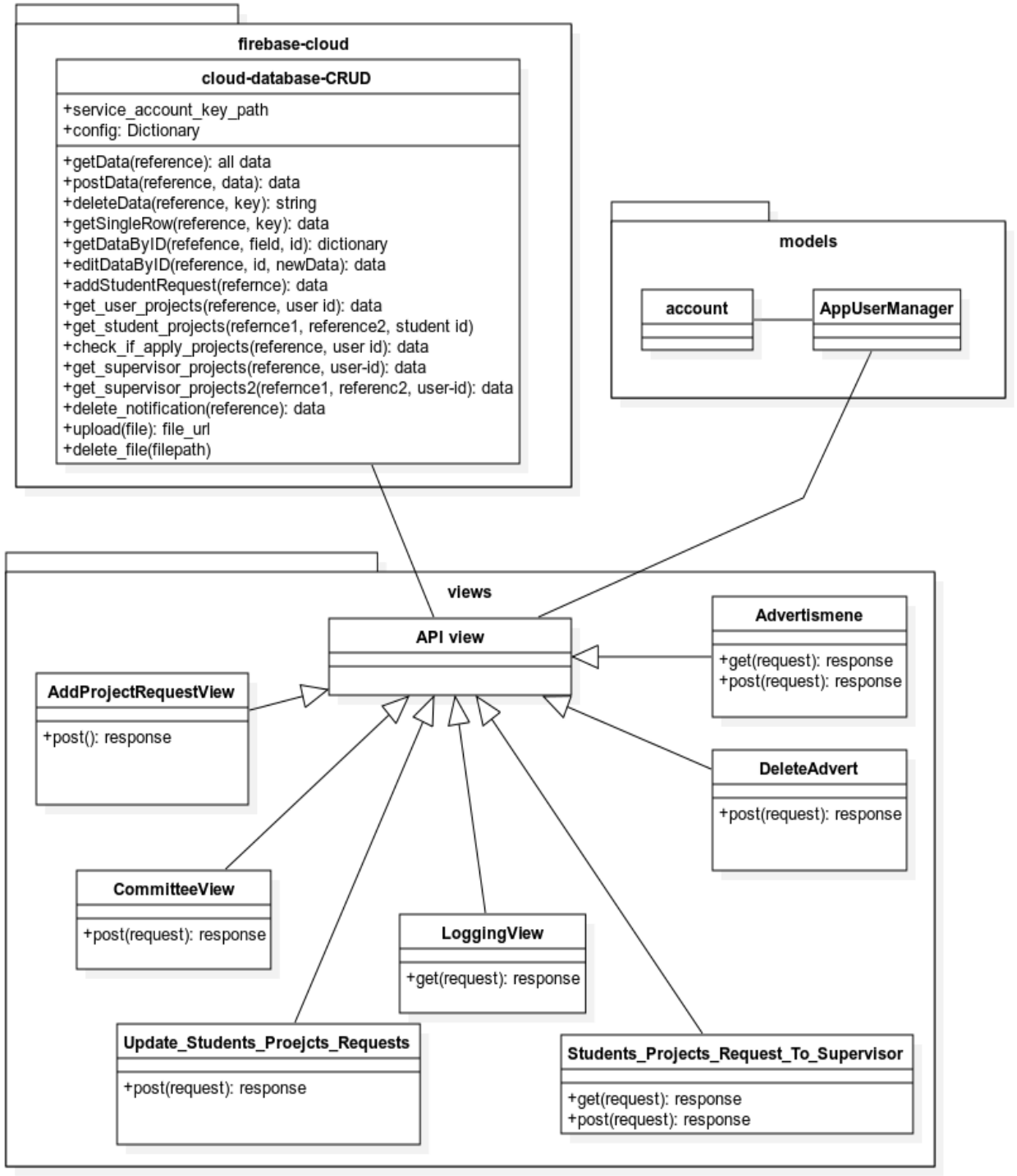
Req-id	Title	Analysis	Detailed design	coding	App user interface	Test cases
Req-01	The system must allow the admin to make an account for the supervisor, manager, and employee by a unique ID and password.	Sp3an				Tc-01 Tc-02 Tc-03
Req-02	The system must allow students to make a new project suggestion and send it to a supervisor they choose.	Sp3an				------
Req-03	The system must allow the manager to set a supervisor as the head of the evaluation process for a specific department.	Sp3an				Tc-04
Req-04	The system must allow the manager and the head of the evaluation process to set an advertisement.	Sp3an				Tc-05
Req-05	The system must allow the head of the evaluation team	Sp3an				Tc-06

	and the manager to upload files with an advertisement.					
Req-06	The system must log all the events that occur on the system.	Sp3an				Tc-07 Tc-08

Sprint#3 design:

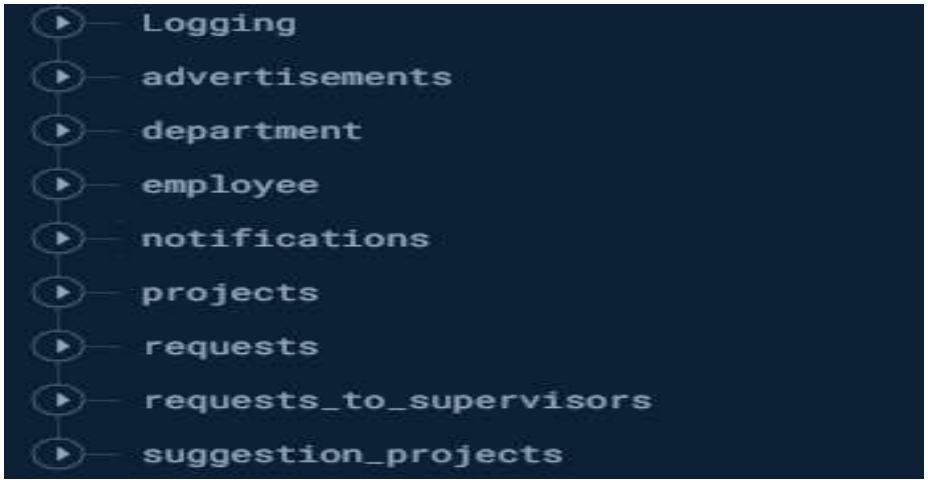
In this section, we will introduce the detailed design for the components of the third sprint, and database components.

1. Detailed design diagram:



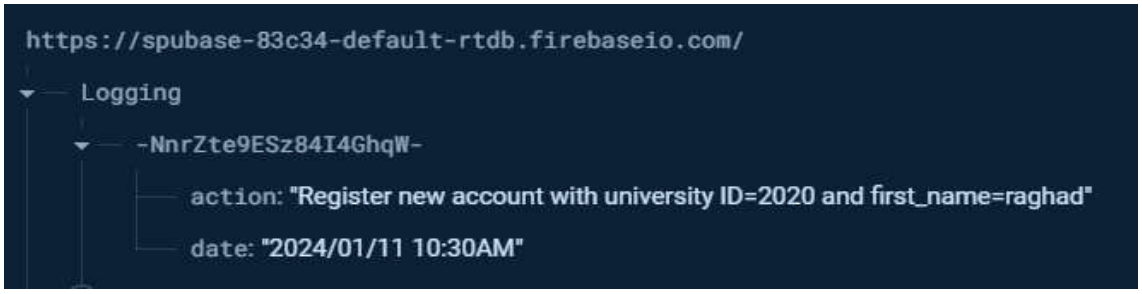
2. Database design:

Realtime database structure:

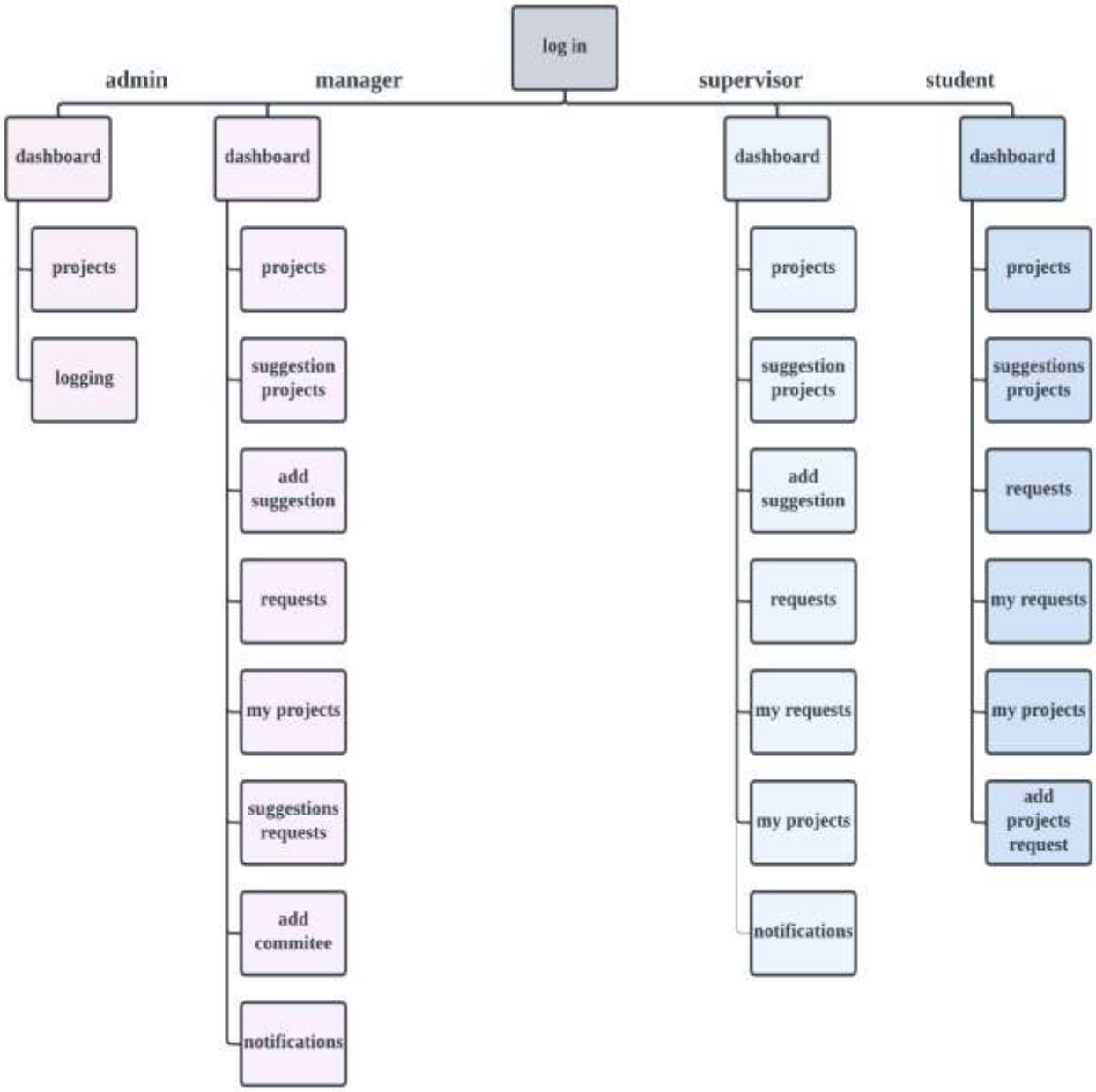


- Advertisements reference:

- Logging reference:



3. Site map:


log in
admin
manager
supervisor
student
dashboard
projects
logging
dashboard
projects
suggestion projects
add suggestion
requests
my projects
suggestions requests
add commitee
notifications
dashboard
projects
suggestion projects
add suggestion
requests
my requests
my projects
notifications
dashboard
projects
suggestions projects
requests
my requests
my projects
add projects request

Sprint#3 implementation and testing

1. App interfaces:

   - Add account in interface (admin account):

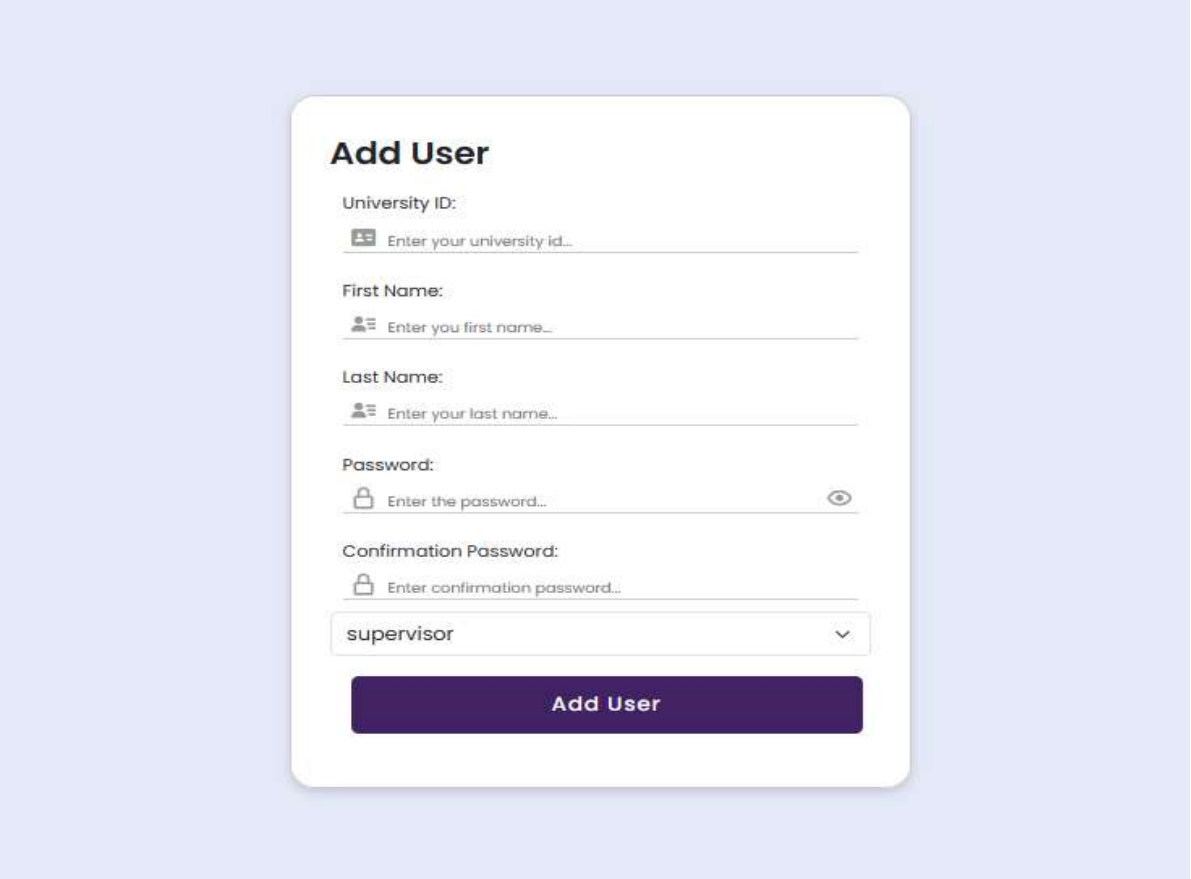


Figure 51 inrf-01 sprint3

- Logging interface (admin account):

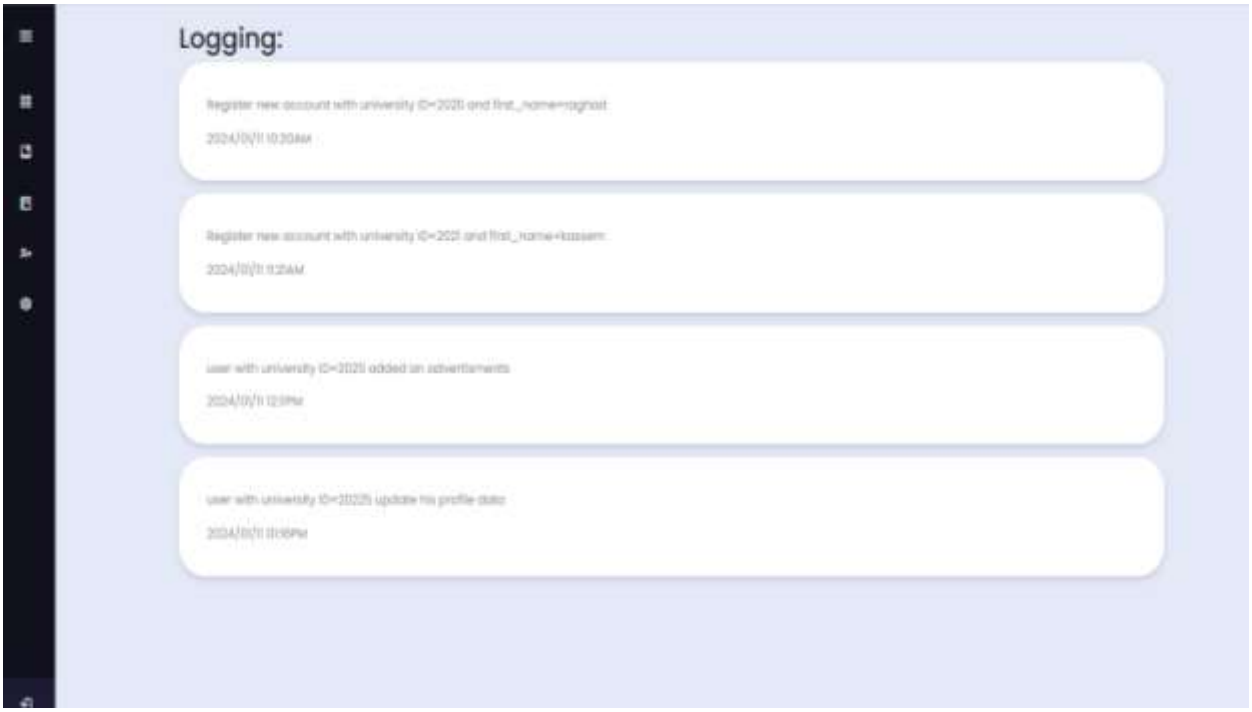


Figure 52 inrf-01 sprint3

- Add evaluation member (manager account):

Figure 53 inrf -04 sprint3

- Add advertisement:

Figure 54 inrf-05 sprint3

- Register new project (student account):

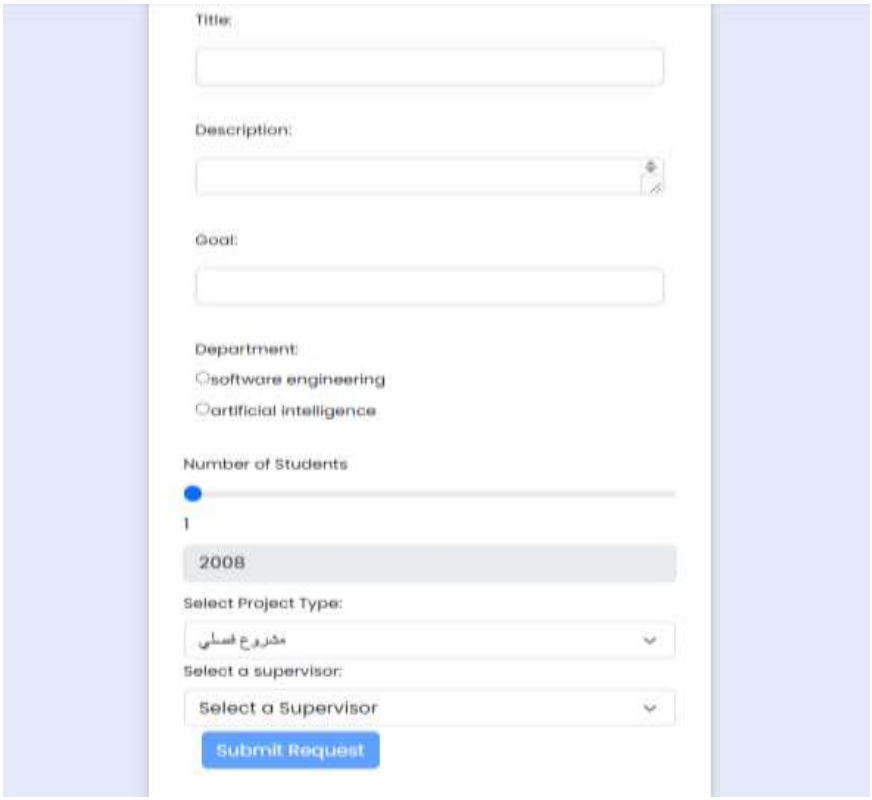


Figure 55 inrf-06 sprint3

2. Test cases execution:

TC id	Test case title	Req-id	Tested data	Expected result	Actual result	Pass/ fail
Tc-01	Check results on choosing a vailed user ID and strong password	Req-01	University id=4200066 First name=" akram" Last name=" masoh" Account-type= supervisor Password=2323@23	Account successfully created.	Account successfully created.	Pass
Tc-02	Check results on choosing an ID that already exists with a strong password	Req-01	University id=4200066 First name=" akram" Last name=" masoh" Account-type= supervisor Password=2323@23	Error message "ID already exists	Error message "ID already exists	Pass
Tc-03	Check results on choosing a password that is not strong	Req-01	University id=4200066 First name=" akram"	Error message "password is not strong enough".	Error message "password is not strong enough".	Pass

	enough. button is pressed.		Last name=" masoh" Account-type= supervisor Password=1234			
Tc-04	Check results on choosing to add evaluation team member.	Req-03		The selected supervisor had the committee privileges.	The selected supervisor had the committee privileges.	Pass
Tc-05	Check results on choosing to add new advertisement.	Req-04		Advertisement must successfully add to the advertisement list to be display for every user.	Advertisement must successfully add to the advertisement list to be display for every user.	Pass
Tc-06	Check results on choosing to add advertisement with file.	Req-05		Advertisement must successfully add to the advertisement list to be display for every user.	Advertisement must successfully add to the advertisement list to be display for every user.	Pass
Tc-07	Check results on making any action against the database	Req-06		The action must be logged, and add to the database.	Log must be displayed for the admin each log with its information.	Pass

	(add suggestion,) by any user.					
Tc-08	Check results on choosing "display logging" by admin	Req-06		Log must be displayed for the admin each log with its information.	Log must be displayed for the admin each log with its information.	Pass

3. Final requirements traceability matrix – sprint3:

Table 20 final RTM Sprint3

Req-id	Title	Analysis	Detailed design	coding	App user interface	Test cases
Req-01	The system must allow the admin to make an account for the supervisor, manager, and employee by a unique ID and password.	Sp3an	Sp3des	Sp2imp	Inrf-01	Tc-01 Tc-02 Tc-03
Req-02	The system must allow students to make a new project suggestion and send it to a supervisor they choose.	Sp3an	Sp3des	Sp2imp	Inrf-05	------
Req-03	The system must allow the manager to set a supervisor as the head of the evaluation process for a specific department.	Sp3an	Sp3des	Sp2imp	Inrf-03	Tc-04
Req-04	The system must allow the manager and the head of the evaluation process to set an advertisement.	Sp3an	Sp3des	Sp2imp	Inrf-04	Tc-05
Req-05	The system must allow the head of the evaluation	Sp3an	Sp3des	Sp2imp	Inrf-04	Tc-06

	team and the manager to upload files with an advertisement.					
Req-06	The system must log all the events that occur on the system.	Sp3an	Sp3des	Sp2imp	Inrf-02	Tc-07 Tc-08

Chapter 4 conclusion

Conclusion:

in the result of out projects, we came with a system aim to help students, supervisors, manger, employee, admin for doing their jobs in more easy and efficient way, using scrum methodology to develop the software and cloud computing storage.

References:

1. https://www.scrum.org/resources/what-scrum-module
2. https://docs.djangoproject.com/en/5.0/
3. https://www.google.com/search?q=react+documentation&oq=react+documentation&gs_lcrp=EgZjaHJvbWUyBggAEEUYOTIGCAEQRRg7MgYIAhBFGDsyBggDEEUYO9IBCDQzMTdqMGo3qAIAsAIA&sourceid=chrome&ie=UTF-8
4. https://docs.python.org/3/