



Cairo University
Faculty of Statistical Studies and Research
Department of Computer Sciences

Graduation Projects Recommendations Website

Project Documentation Report
For
CS 599 evaluation

Diploma project
2021 - 2023

Submitted by
Ahmed Mohamed Soliman Radwan
Ahmed Elsayed Mouhamed Elkhamissi
Mahmoud Ali Mohammed Abdullah
Ahmed Mouhamed Abdelkader Hamed
Mahmoud Mostafa Ahmed Hegab
Khaled Mouhamed Fathallah Mohamed
Ahmed Abdelmowla Ibrahim Elsayed

Supervised by
Dr. Moustafa Ezzat

Abstract

Annually, thousands of students of final years in computer science faculties search for graduation projects ideas, thus makes them unleash their imaginations in making projects that may not suit their knowledge, accordingly, makes them in the end face two choices, either a project that they do not know how to implement, or they get lost in forests of ideas.

So, we developed a website that will help computer science student to find the most suitable project depending on their knowledge.

To get a project idea, just create an account, then select your mastered technology, and the system will recommend best 5 projects that fit your knowledge.

This report is an attempt to present the system's working mechanism, supported by system analysis and design artifacts that preceded the implementation process.

Team members

Name	Task
Ahmed Mohamed Soliman Radwan	Database design (ERD & Mapping) System design (Sequence diagram) Database implementation
Ahmed Elsayed Mouhamed Elkhamissi	System analysis (SRS) System design (System architecture, Use case description & diagram) Acceptance testing (test cases design & executing)
Mahmoud Ali Mohammed Abdullah	Database design (ERD & Mapping) System analysis (Traceability matrix) Database implementation
Ahmed Mouhamed Abdelkader Hamed	System analysis (Actor Description) System design (Component Diagram) AI engine design and implementation
Mahmoud Mostafa Ahmed Hegab	System analysis (User Stories, Work backlog & Project duration) System design (Activity Diagram & Class Diagram) and front-end implementation
Khaled Mouhamed Fathallah Mohamed	Team leader and front-end implementation
Ahmed Abdelmowla Ibrahim Elsayed	Backend implementation

Acknowledgement

First of all, we thank Allah for granting us success in completing this work.

In acknowledgment of our esteemed professors at the Faculty of Graduate Studies for Statistical Research for helping us reach our final semester, we especially thank Dr. Mustafa Ezzat for his great advice and continuous support for us to accomplish this work.

We also owe thanks to all the faculty members and the supporting staff in the Department of Computer Science, who did not spare us advice or guidance, whether in our academic curricula or in our practical projects.

In conclusion, we thank everyone who created the conditions for us to implement this project easily and smoothly, and we thank everyone who placed obstacles in front of us. Without the challenges, success would not have been of any value.

Table of Contents

Abstract	2
Team members	3
Acknowledgement	4
Table of Contents	5 - 6
Table of Figures	7
Table of Tables	8
Chapter 1: Introduction	9 - 14
1.1 Introduction	10
1.2 Motivation	10
1.3 Problem Definition	11
1.4 Scope	11
1.5 Aims and Objectives	11
1.6 project Stakeholders	12
1.6.1 Computer science final-year students	12
1.7 Development phases and methodology	12 - 13
1.7.1 System Development Life Cycle	12
1.7.2 Development methodology	13
1.7.4 Agile	13
1.7.5 Scrum	13
1.8 Design and Development tools	14
1.8.1 UML Diagrams tool	14
1.8.2 Development Tools	14
Chapter 2: System Analysis	15 – 20
2.1 Requirements Analysis	15 - 17
2.1.1 Functional Requirements	15 - 16
2.1.2 Non-Functional Requirements	17
2.2 User story	18
2.3 Work backlog	19
2.4 Project duration	19
2.5 Actor description	20
Chapter 3: SYSTEM DESIGN	21 - 43
3.1 Use case diagram	22
3.2 Use case description	23 - 37
3.3 Traceability Matrix	38
3.4 Activity diagram	39
3.5 Sequence diagram	40
3.6 Component diagram	41
3.7 Class diagram	42
3.8 Entity relationship diagram (ERD)	43
3.9 Mapping	43

Chapter 4: System Implementation and Testing Plan	44 - 76
4.1 Interfaces	45 - 49
4.2 AI Engine	50 - 59
4.3 Acceptance testing	60 - 76
4.3.1 Test plan	60
4.3.2 Test cases.....	61 - 75
4.3.3 Bug report	76
Chapter 5: Conclusion and Future Work	77 - 79
5.1 Conclusion	78
5.2 Future work	79
References	80

Table of Figures

Fig 1.1: Scrum sprints	13
Fig 3.1: Use case diagram	22
Fig 3.2: Activity diagram	39
Fig 3.3: Sequence diagram	40
Fig 3.4: Component diagram	41
Fig 3.5: Class diagram	42
Fig 3.6: Entity relationship diagram (ERD)	43
Fig 3.7: Mapping	43
Fig 4.1 Home page	45
Fig 4.2 Sign up page	45
Fig 4.3 Sign in page	46
Fig 4.4 Dashboard page	46
Fig 4.5 Skills selection & Projects recommendations page	47
Fig 4.6 Learning resources recommendations page	47
Fig 4.7 Create team page	48
Fig 4.8 Create task page	48
Fig 4.9 User data page	49

Table of Tables

Table 2.1: Functional Requirement	16 - 17
Table 2.2: Non-Functional Requirement	17
Table 2.3: User story	18
Table 2.4: Work backlog	19
Table 2.5: Actor description	20
Table 3.1: Use case 01	23
Table 3.2: Use case 02	24
Table 3.3: Use case 03	25
Table 3.4: Use case 04	25
Table 3.5: Use case 05	26
Table 3.6: Use case 06	27
Table 3.7: Use case 07	28
Table 3.8: Use case 08	29
Table 3.9: Use case 09	30
Table 3.10: Use case 10	30
Table 3.11: Use case 11	31
Table 3.12: Use case 12	31
Table 3.13: Use case 13	32
Table 3.14: Use case 14	33
Table 3.15: Use case 15	34
Table 3.16: Use case 16	35
Table 3.17: Use case 17	35
Table 3.18: Use case 18	36
Table 3.19: Use case 19	37
Table 3.20: Use case 20	37
Table 3.21: Traceability Matrix	38

Chapter 1: Introduction

1.1 Introduction

Over the last ten years, we have witnessed an increasing demand for studying computer science, given that job opportunities in software development are the most in demand in the world.

However, theoretical studies do not produce a graduate who is able to meet the requirements of the labor market. Hence the importance of the idea of graduation projects to measure the extent to which students are able to apply their academic studies in a practical way.

Our system is a website supported by a database, to provide more than 100 projects ideas, implementable by a combination of more than 100 programming language, frameworks, libraries, and query language.

Our system consists of a start page and various page for the rest system functions, and a database system to store all relevant data, in addition to an AI engine that recommends the projects.

The main concept of the system is constructing a service that allows computer science final years students to get the fittest graduation project.

System also includes many interfaces, which show variant functions, like recommend a learning resource for a specific topic in computer science in case the student want to learn a tool before use it in a project.

In addition to the above, the system supports the principle of teamwork through the function of forming a team that can work on a specific project.

The main purpose of our system is helping students in choose suitable project, to saves time and effort in searching for projects without clear constraints.

1.2 Motivation

The main motive for us to do this project was what we faced while choosing our graduation project. At the beginning, we had chosen a project that did not fit our capabilities, and when we re-searched and brainstormed to choose new projects, the results were similar to the previous one.

So, we wanted our colleagues not to go through what we went through.

1.3 Problem Definition

Computer science students can easily obtain Graduation projects ideas with help of many websites, which are interested in the fields of software development, but these projects ideas don't satisfy the tools, that the students can work with-it.

So, we develop a system to recommend graduation projects based on the tools, which students can work with-it.

1.4 Scope

Many websites provide a lot of graduation projects ideas, but these ideas don't consider what the students tools that can work with it.

Our system handles this problem by asking the user first what the tool that he mastered then based on this tool the system recommends a project.

Our system is based on “Stack overflow” survey, which investigate the most active application ideas and the tools are used in them.

If the system recommends a project with many tools but one of them doesn't master by the student, the student can also ask the system to provide a learning resource for this unmastered tool.

1.5 Aims and Objectives

- Recommending 5 projects ideas, that fit the mastered tools in no time “figuratively”.
- Provides a brief for each recommended project.
- Supporting team-work concept.
- Supporting Learn and work concept.

1.6 project Stakeholders

Stakeholder is a beneficiary who wants to take advantage of our system.

1.6.1 Computer science final-year students

Computer science final-year students are the main stakeholder in our project.

- They will get the most perfect project idea.
- They will learn for free one of the software development tools.
- They will learn how to work in a group.

1.7 Development phases and methodology

We developed our system by a parallel process, we initiate the development process by analysis the system characteristics, functions, and circumstances then we design the main UML diagrams, then we start implementing the most important functions. After every function we apply component test for this function then moving to the next function.

Finally, when we finished the whole system, we applied the acceptance testing to determine that we did the right system, and we did it right.

1.7.1 System Development Life Cycle

Systems development life cycle is a conceptual model used in project management that describes the stages involved in an information system development project, it generally consists of 4 phases.

System initiation: defining the project through determine (business scope, goals, and feasibility).

System analysis: defining the system requirements and its priorities.

System design: constructing graphical models for the system requirements.

System implementation: construction the source code, testing the system, maintain the system, and deployment.

1.7.2 Development methodology

The development methodology consists of multiple sequential phases in a project lifecycle, with some phases being optional, and the order of phases being dependent on the project being undertaken.

We choose agile methodology with scrum framework, due to the nature of our system, which is based on working in a iterative phased manner.

1.7.4 Agile

Agile is the ability to create and respond to change, and applying these concepts in software development refers a set of frameworks and practices to help figure out the right things to do given your particular context.

1.7.5 Scrum

Scrum is a lightweight framework based on Agile methodology, that helps people, teams and organizations generate value through adaptive solutions for complex problems.

Scrum works in short timeframes called sprints. that happen one right after the other, to maintain a steady project cadence.

Each sprint begins with a (sprint planning) and ends with a review of the work completed (sprint review) and an additional look back at the way in which the team worked together (sprint retrospective).

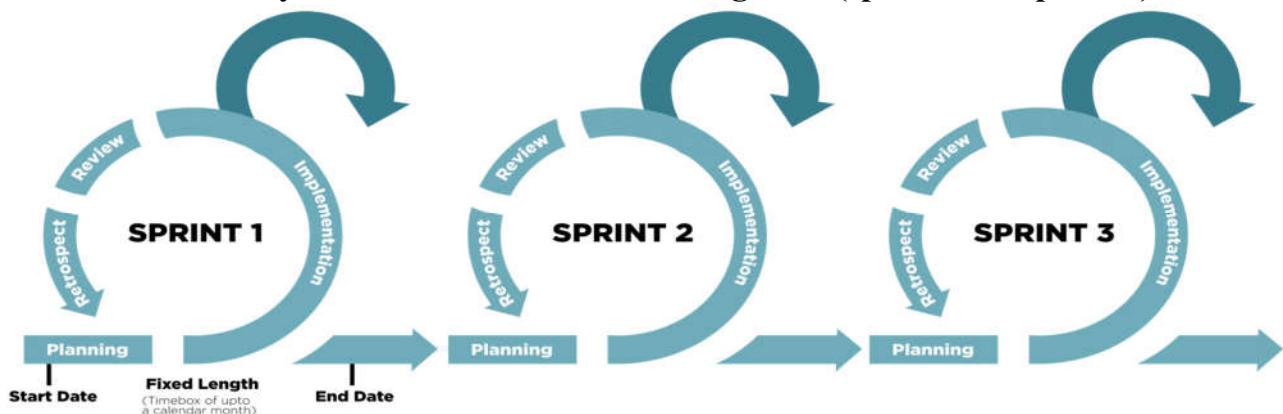


Fig 1.1: Scrum sprints

1.8 Design and Development tools

1.8.1 UML Diagrams tool

UML stands for Unified Modeling Language, is a way to visually represent the architecture, design, and implementation of complex software systems.

We used <https://www.drawio.com/> to design our UML diagrams & Mapping.

We used erdplus.com to design ERD.

1.8.2 Development Tools

- **Front-End**
 - **Angular with TypeScript**
 - **UI template design has been adopted from an open source UI templates website and is attempted to modification.**
- **Back-End**
 - **Asp.net CORE WEB API**
 - **Sql server**
- **AI Engine implemented by Python**
- **Code Editor (Visual Studio Code)**

Chapter 2: System Analysis

2.1 Requirements Analysis

2.1.1 Functional Requirements

We have elicited the functional requirements by two methods.

1- Interview: We imagined ourselves as a client and business analysis “stakeholders”, the interview is done as an open interview, in which there is no pre-defined agenda.

2- We have completed the functional requirements gathering using scenarios, that helped to describe clearer the interaction of the requirements.

Table 2.1: Functional Requirement

Functional Requirements		
Req ID	Requirement	Priority
Fr 001	The user shall create an account	5
Fr 002	The user shall sign in to the system using their account	5
Fr 003	The registered and unregistered users shall search for projects	3
Fr 004	The system should show a brief for the projects	4
Fr 005	The registered users shall choose their tools / technologies / frameworks / libraries / interests	5
Fr 006	The registered user shall be able to edit its chosen tools, technologies, frameworks, libraries.	2
Fr 007	The system shall recommend projects depending on preferences	5
Fr 008	The system shall show the whole details for the chosen project	4
Fr 009	The registered users shall create a team	5
Fr 010	The registered individual user shall create a team	5
Fr 011	The system shall make the registered user who create a team a team leader	3
Fr 012	The team leader shall choose "Accept" a project	3
Fr 013	The system shall show a control dashboard for the chosen project	5
Fr 014	The team leader shall choose a team name and tag	2
Fr 015	The team leader shall add the other registered users to the team	4
Fr 016	The system should contribute the project tasks on the team members depending on the members preferences	3
Fr 017	The system shall recommend Learning resources to the team for the missing technology / tool	4
Fr 018	The system shall choose a team member that will learn a technology / tool	4
Fr 019	The system shall recommend learning resources for the user without a team, depending on their tools & interests	5
Fr 020	The system shall recommend learned user to a team, that need their tools / tech.	4
Fr 021	The system should test the users in the topics, that they have learned.	3
Fr 022	The system should deactivate the unlearned technology / tool tasks	1
Fr 023	The team leader shall determine start / end date of the projects	2
Fr 024	The team leader shall delay start / end date of the projects	2
Fr 025	The team Leader should be able remove a member from team.	3
Fr 026	The team Leader shall be able to deactivate the chosen project.	2
Fr 027	The team member should be able to disjoin his team.	3

2.1.1 Functional Requirements (Cont.)

Table 2.1: Functional Requirement (Cont.)

Fr 028	The whole project dashboard shall be shown for all team members	2
Fr 029	System shall recommend a new team member, able to work by a specific technology / tool at the project's predetermined dates, when a team member disjoin the team.	3
Fr 030	The team and project name shall appear for all members	1
Fr 031	The system should save the last data in case of system crash	3
Fr 032	The whole project dashboard shall be temporarily deactivated	3
Fr 033	The specific task dashboard shall be temporarily deactivated when a team member quit the project	3

2.1.2 Non-Functional Requirements

Table 2.2: Non-Functional Requirement

Nonfunctional requirements		
I- Product requirements		
Req ID	Requirment	Priority
NFR 001	The search operation shall be fast.	5
NFR 002	The system shall be able to manage the increasing workload in registration operations.	5
NFR 003	The system shall be able to manage the increasing workload in search operation.	5
NFR 004	The system shall be able to manage the increasing workload in projects recommendation operation.	5
NFR 005	The system shall be able to protect students' data.	4
NFR 006	The system failure probabilities shall be minimal.	4
NFR 007	The system critical failures shall be low.	4
NFR 008	The user shall understand how the system working easily.	3
NFR 009	The user shall handle the system functions easily.	3

II- Organizational requirements		
Req ID	Requirment	Priority
NFR 010	The system shall be able to work on different browsers.	5
NFR 011	The system shall be able to work on windows, macOS and Linux Operating system.	5
NFR 012	The system functions shall be maintainable	4
NFR 013	The system should be upgradable	3

III- External requirements		
Req ID	Requirment	Priority
NFR 013	The system shall support non-plagiarism	5
NFR 015	The system should be able to work on projects potential misuse.	3

2.2 User story

Table 2.3: User story

Identifier	User Story	Size
ST-1	As a unregister Student, I can view projects briefs.	5pt
ST-2	As a unregister Student, I can register and enter all information about me.	2pt
ST-3	As a register Student, I can choose tools, technologies, frameworks, libraries that's know it.	3pt
ST-4	As a register Student, I can see recommended projects depend on my knowledge.	6pt
ST-5	As a register Student, I can search for all projects.	3pt
ST-6	As a register Student, I can see description for any project.	2pt
ST-7	As a register Student, I can choose project.	2pt
ST-8	As a register Student, I can see all details about the chosen project.	3pt
ST-9	As a register Student, I can see recommendation for learning resources.	4pt
ST-10	As a register Student, I can create team as a team leader.	3pt
ST-11	As a register Student (Team Leader), I can see dashboard to mange team.	4pt
ST-12	As a register Student (Team Leader), I can add students to team.	2pt
ST-13	As a register Student (Team Leader), I can see recommendations for students.	2pt
ST-14	As a register Student (Team Leader), I can remove student from team.	2pt
ST-15	As a register Student (Team Leader), I can distribute tasks for all team members.	4pt
ST-16	As a register Student (Team Leader), I can determine start and end date for project.	2pt
ST-17	As a register Student (Team Leader), I can deactivate chosen project.	2pt
ST-18	As a register Student, I can use chatbot to chat with any member on team.	6pt
ST-19	As a register Student, I can see recommendation teams to join.	2pt
ST-20	As a register Student, I can accept invitation to join team.	2pt
ST-21	As a register Student, I can disjoin team.	2pt
ST-22	As a register Student, I can see my tasks in project dashboard.	4pt
ST-23	As a register Student, I can edit tools, technologies, frameworks, libraries.	2pt

2.3 Work backlog

Table 2.4: Work backlog

Work Item	User Story	Iteration No.	Estimate Work duration
1)	ST-1 View project	Iteration 1	6pt (5 days)
2)	ST-2 Registration	Iteration 1	2pt (1 day)
3)	ST-3 Select Skills (tools, technologies, frameworks, libraries)	Iteration 1	3pt (2 days)
4)	ST-4 projects	Iteration 1	6pt (5 days)
5)	ST-6 project description	Iteration 1	2pt (1 day)
6)	ST-5 Search	Iteration 2	3pt (2 days)
7)	ST-7 select project	Iteration 2	2pt (1 day)
8)	ST-9 Learning Resources	Iteration 2	4pt (3 days)
9)	ST-8 Project details	Iteration 2	3pt (2 days)
10)	ST-10 create team	Iteration 2	3pt (2 days)
11)	ST-11 dashboard	Iteration 2	4pt (3 days)
12)	ST-13 student recommendation	Iteration 3	2pt (1 day)
13)	ST-16 estimate project duration	Iteration 3	2pt (1 day)
14)	ST-12 add student	Iteration 3	2pt (1 day)
15)	ST-14 remove student	Iteration 3	2pt (1 day)
16)	ST-17 deactivate project	Iteration 3	2pt (1 day)
17)	ST-15 distribute tasks	Iteration 3	4pt (3 days)
18)	ST-20 accept invitation	Iteration 3	2pt (1 day)
19)	ST-21 disjoin team	Iteration 3	2pt (1 day)
20)	ST-18 Chatbot	Iteration 4	6pt (5 days)
21)	ST-19 team recommendation	Iteration 4	2pt (1 day)
22)	ST-22 view tasks	Iteration 4	4pt (3 days)
23)	ST-23 Edit skills	Iteration 4	2pt (1 day)

2.4 Project duration

Time estimate

2 point = 1 day

3 point = 2 days

4 point = 3 days

5 point = 4 days

6 point = 5 days

Total point = 70 point

Project duration = 70/2 = 35 days

2.5 Actor description

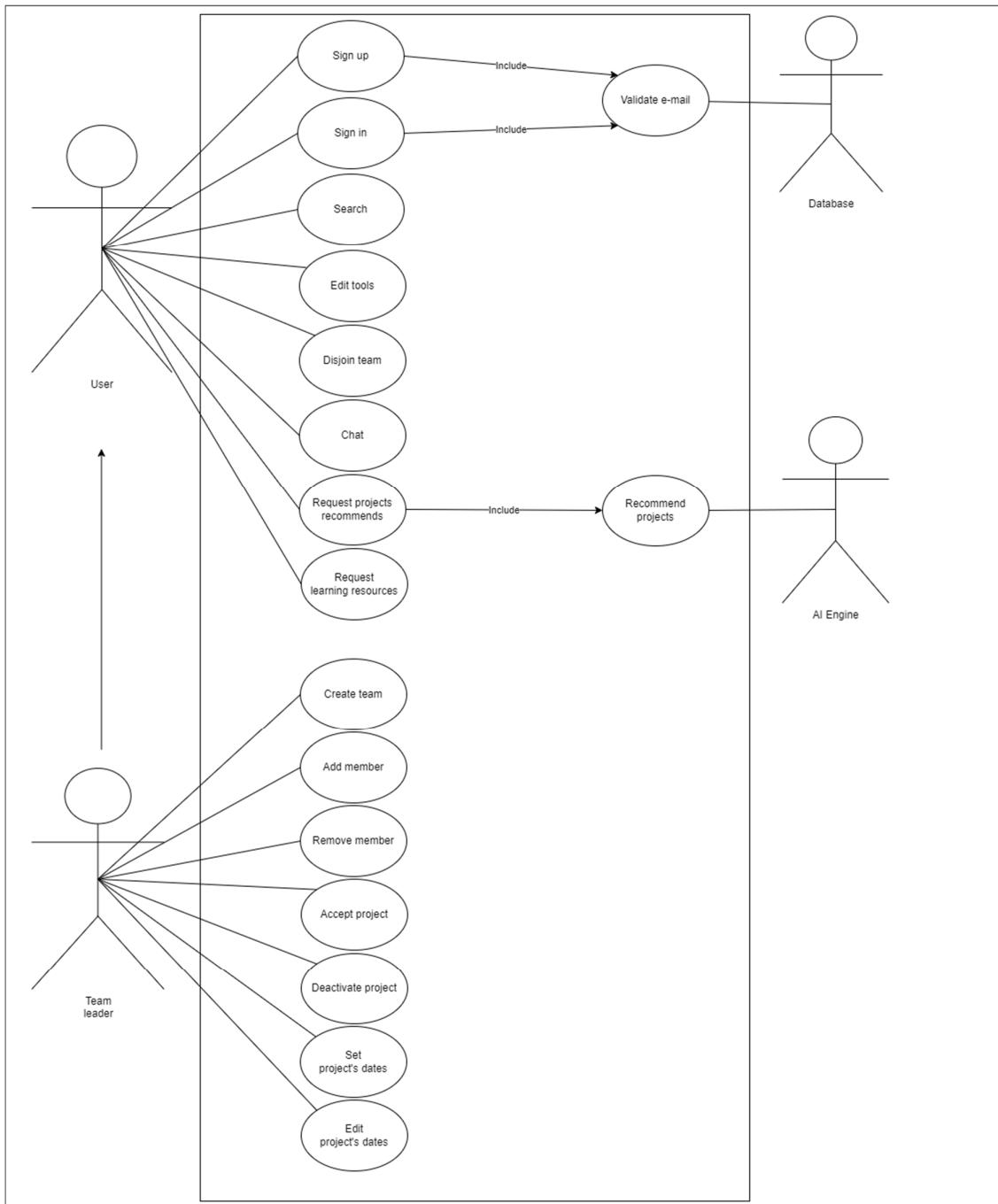
Table 2.5: Actor description

Actor	Actor's Goal	Use Case Name
User	Create a new account on the website and choosing preferences.	Create a new account – sign up
User	Sign in using his / her account	Sign in
User	Edit his / her tools / technologies / frameworks / libraries / interests.	Edit tools / technologies / frameworks / libraries / interests
User	Searching for projects by its names and view its briefs.	Search for projects
User	Accept a specific project.	Accept a project and show its details “DASHBOARD”.
User	User must be registered, and all his / her data and preferences are existing in the data base.	Create a team
User	Disjoin a team.	Disjoin a team.
User	Project’s dashboard temporarily deactivated when the team leader quit the project	Deactivate dashboard
User	Team members communicate through chatbot.	Chatbot
Team leader	Show the project’s details.	Accept a project and show its details “DASHBOARD”
Team leader	Add other registered users to the team	Add other registered users to the team
Team leader	System provides Learning resources to the team for the missing technology / tool, or a new team member can work with the team’s missed tool.	Recommending Learning resources / new team member
Team member	System tests the team member that was recommended to learn a specific tool.	Test the users in the topics, that they have learned.
Team leader	Determine start / end date of the projects	Determine start / end date of the project
Team leader	Edit start / end date of the projects	Delay start / end date of the projects
Team leader	Remove a member from team.	Remove a member from team.
Team leader	Deactivate a chosen project.	Deactivate a chosen project.
Team Leader	Recommend a new team member, able to work by a specific technology/tool at the project’s predetermined dates, when a team member disjoin the team.	Recommend a new member for a team is lifted by another user
Registered user without a team	System provides Learning resources to the team for the missing technology / tool, or a new team member can work with the team’s missed tool.	Recommending Learning resources / new team member
Database	Validate e-mail not registering	Create a new account – sign up
Database	Validate e-mail and password are existed	Sign in
System	System contributes the project’s tasks on the team members depending on the members preferences	Contribute project’s tasks
System	Deactivate the unlearned technology/tool tasks until the team member learn it	Deactivate the unlearned technology / tool tasks
AI Engine	Recommend projects depending on preferences.	Recommending projects

Chapter 3: SYSTEM DESIGN

3.1 Use case diagram

Fig 3.1: Use case diagram



3.2 Use case description

Table 3.1: Use case 01

Use Case ID:	UC 001
Use Case Name:	Create a new account – sign up
Related requirements:	Fr 001, Fr 005
Actor:	1. User 2. Database
Actor's Goal:	Create a new account on the website and choosing preferences.
Preconditions:	None
Postconditions:	The user's data be stored in the database
Main success scenario Flow	
→ 1. The user clicks the “SIGN UP” button. ← 2. The system redirects the unregistered user to the sign-up page. → 3. The unregistered user fills his/her name, ID, mobile, number, e-mail, and password. ← 4. The database checks if the e-mail is already registered or not. ← 5. The system redirects the unregistered user to the tools page. → 6. The unregistered user fills his/her tools, technologies, libraries, frameworks, and interests. → 7. The unregistered user clicks the “CREATE” button. ← 8. The system shows a message “Congratulation... Your account is created successfully”	
Alternate scenario Flow (1)	
← 4.1 The database checks if the e-mail is already registered or not, and finds the E-mail is already registered, and stops the sign process. ← 1. The system shows message “E-mail is already registered, press ok enter another one” → 2. The unregistered user clicks ok button. ← 3. The system redirects the unregistered user to the sign page “to change only the e-mail and keep the rest data”. → 4. The unregistered user fills his/her name. ← 5. The database checks if the e-mail is already registered or not. ← 6. The system redirects the unregistered user to the tools page. → 7. The unregistered user fills his/her tools, technologies, libraries, frameworks, and interests. → 8. The unregistered user clicks the “CREATE” button. ← 9. The system shows a message “Congratulation... Your account is created successfully”	

Table 3.2: Use case 02

Use Case ID:	UC 002			
Use Case Name:	Sign in			
Related requirements:	Fr 002	Related Use Case: UC 001		
Actor:	1. User 2. Database			
Actor's Goal:	Sign in using his / her account			
Preconditions:	User must exist in database			
Postconditions:	User is being singed in			
Main success scenario Flow				
<ul style="list-style-type: none"> → 1. The user clicks the “SIGN IN” button. ← 2. The system opens a sign in window to enter his / her e-mail and password. → 3. The registered user fills his/her e-mail and password. ← 4. The database checks if the e-mail and password are matched and exist in database. ← 5. The system redirects the registered user to his/her profile page. 				
Alternate scenario Flow (1)				
<ul style="list-style-type: none"> ← 4.1 E-mail and password are not matched, and the system sign in process. ← 1. The system shows message “E-mail and password not matched.” ← 2. The system opens sign in window to enter e-mail and password again. → 3. The user fills his/her e-mail and password. ← 4. The database checks if the e-mail and password are matched and exist in database. ← 5. The system redirects the registered user to his/her profile page. 				
Alternate scenario Flow (2)				
<ul style="list-style-type: none"> ← 4.2 E-mail and password are not matched, and the system sign in process. ← 1. The system shows message “E-mail and password not matched.” ← 2. The system opens sign in window to enter e-mail and password again. → 3. The user fills his/her e-mail and password. ← 4. The database checks if the e-mail and password are matched and exist in database. ← 5. The system finds that the e-mail and password not matched for 3 times. ← 6. The system deactivates the account. 				
Alternate scenario Flow (3)				
<ul style="list-style-type: none"> ← 4.3 E-mail is not existed in the database, and the system sign in process. ← 1. The system shows message “This e-mail is not existed, Do you want to sign up.” ← 2. The system shows “OK” and “CANCEL” buttons. → 3. The unregistered user clicks on “OK” button. ↔ 4. Repeat “UC 001” scenarios. 				
Alternate scenario Flow (4)				
<ul style="list-style-type: none"> ← 4.4 E-mail is not existed in the database, and the system sign in process. ← 1. The system shows message “This e-mail is not existed, Do you want to sign up.” ← 2. The system shows “OK” and “CANCEL” buttons. → 3. The user clicks on “CANCEL” button. ← 4. The system redirects the unregistered user to the homepage. 				

Table 3.3: Use case 03

Use Case ID:	UC 003			
Use Case Name:	Edit tools / technologies / frameworks / libraries / interests			
Related requirements:	Fr 006	Related Use Case: UC 002		
Actor:	1. User			
Actor's Goal:	Edit his / her tools / technologies / frameworks / libraries / interests.			
Preconditions:	The user must be registered and exist in the database.			
Postconditions:	The choices and preferences are being edited in the database.			
Main success scenario Flow				
<ul style="list-style-type: none"> ↔ 1. Repeat use case “UC 002” scenarios. → 2. The user clicks “PREFERENCES” button. ↔ 3. The system redirects the registered user to his / her preferences page. → 4. The user use clicks “EDIT” button. → 5. The user edits his / her tools / technologies / frameworks / libraries / interests. ↔ 6. The system shows a “SAVE” and “CANCEL” buttons. → 7. The user clicks on “SAVE” button. ↔ 8. The system shows a message “Your preferences are edited.”. 				
Alternate scenario Flow (1)				
<ul style="list-style-type: none"> → 7.1 The user clicks on “CANCEL” button. ↔ 1. The system shows message “Your preferences are not edited.” 				

Table 3.4: Use case 04

Use Case ID:	UC 004			
Use Case Name:	Search for projects			
Related requirements:	Fr 003, Fr 004			
Actor:	1. User			
Actor's Goal:	Searching for projects by its names and view its briefs.			
Preconditions:	Projects and its briefs are existed in database.			
Postconditions:	The users find the related results to their search titles.			
Main success scenario Flow				
<ul style="list-style-type: none"> → 1. The user clicks on the “Search bar”. → 2. The user types a search title on the “Search bar”. → 3. The user clicks “SEARCH” button. ↔ 4. The system searches for the projects that relate to search title in the existed projects. ↔ 5. The system shows the results sorted as related to search title. → 6. The user clicks on a specific search result. ↔ 7. The system shows a brief of the project, contains a simple discription and the tools used in it. 				
Alternate scenario Flow (1)				
<ul style="list-style-type: none"> ↔ 5.1 The system cannot find any results that relate to the search title. ↔ 1. The system shows message “We are sorry, there is no projects relate to your search title.” ↔ 2. The system shows “TRY AGAIN” and “CANCEL” buttons. → 3. The user clicks on the “TRY AGAIN”. ↔ 4. The system redirects the user to the homepage to try search again. 				
Alternate scenario Flow (1)				
<ul style="list-style-type: none"> ↔ 4. The system cannot find any results that relate to the search title. ↔ 1. The system shows message “We are sorry, there is no projects relate to your search title.” ↔ 2. The system shows “TRY AGAIN” and “CANCEL” buttons. → 3. The user clicks on the “CANCEL”. ↔ 4. The system redirects the user to the homepage. 				

Table 3.5: Use case 05

Use Case ID:	UC 005					
Use Case Name:	Recommending projects					
Related requirements:	Fr 007, Fr 020	Related Use Case:	UC 002			
Actor:	1. User 2. AI Engine					
Actor's Goal:	1. Recommend projects depending on preferences. 2. Recommend a team member to a team depending on preferences.					
Preconditions:	1. Projects are existed in database. 2. User must be registered, and all his / her data and preferences are existing in the data base. 3. Team is created.					
Postconditions:	1. System shows the projects, that recommended depend on preferences. 2. System Recommend a team member to a team depending on preferences.					
Main success scenario Flow						
↔ 1. Repeat use case UC 002 scenarios. → 2. The user clicks on “RECOMMEND A PROJECT” button. ← 3. The AI Engine recommends projects depending on the preferences.						
Alternate scenario Flow (1)						
← 3.1 The AI Engine cannot recommend projects depending on the preferences. ← 1. The system shows a message “There are no projects matching your preferences, but you can join a team with your tools.” → 2. The system shows “RECOMMEND ME TO TEAM” and “CANCEL” buttons. → 3. The user clicks on “RECOMMEND ME TO TEAM” button. ← 4. The system recommends the registered user to a team.						
Alternate scenario Flow (2)						
← 3.2 The AI Engine cannot recommend projects depending on the preferences. ← 1. The system shows a message “There are no projects matching your preferences, but you can join a team with your tools.” → 2. The system shows “RECOMMEND ME TO TEAM” and “CANCEL” buttons. → 3. The user clicks on “RECOMMEND ME TO TEAM” button. ← 4. The AI Engine cannot recommend the registered user to a team. ← 5. System shows a message “Sorry, there is no team matching your preferences, try again later.”						
Alternate scenario Flow (3)						
← 3.3 The AI Engine cannot recommend projects depending on the preferences. ← 1. The system shows a message “There are no projects matching your preferences, but you can join a team with your tools.” → 2. The system shows “RECOMMEND ME TO TEAM” and “CANCEL” buttons. → 3. The user clicks on “CANCEL” button. ← 4. The system redirects the registered user to homepage.						

Table 3.6: Use case 06

Use Case ID:	UC 006					
Use Case Name:	Accept a project and show its details “DASHBOARD”.					
Related requirements:	Fr 008, Fr 012, Fr 013, Fr 028, Fr 030	Related Use Case:	UC 002, UC 005			
Actor:	1. Individual user as a team leader 2. Team leader					
Actor's Goal:	1. Accept a specific project. 2. Show the project's details.					
Preconditions:	1. Projects are existed in database. 2. User must be registered, and all his / her data and preferences are existing in the data base. 3. Team is created. 4. System recommends projects.					
Postconditions:	1. Registered user accept a specific project 2. System shows the project's details “Dashboard” for all members.					
Main success scenario Flow						
<ul style="list-style-type: none"> ↔ 1. Repeat use case UC 002 scenarios. ↔ 2. Repeat use case UC 005 scenarios. → 3. The Individual user as a team leader / a team leader click on a specific project. ← 4. The system shows a brief about the project. ← 5. The system shows “ACCEPT” and “DECLINE” buttons. → 6. The Individual user as a team leader / a team leader click on “ACCEPT” button. ← 7. The system shows a message “Congratulation, you chose “the project name, you will be redirected to the project dashboard”. ← 8. The system redirects the registered user to the project dashboard. 						
Alternate scenario Flow (1)						
<ul style="list-style-type: none"> → 5.1 The Individual user as a team leader / a team leader click on “DECLINE” button. ← 1. The system redirects the Individual user as a team leader / a team leader click on to the recommended projects page. 						

Table 3.7: Use case 07

Use Case ID:	UC 007			
Use Case Name:	Create a team			
Related requirements:	Fr 009, Fr 010, Fr 011, Fr 014	Related Use Case: UC 002		
Actor:	1. User 2. System			
Actor's Goal:	Create a team			
Preconditions:	1. User must be registered, and all his / her data and preferences are existing in the data base.			
Postconditions:	1. Team is created and exist in the database. 2. The registered user, who created the team become the team leader.			
Main success scenario Flow				
<p>↔ 1. Repeat use case UC 002 scenarios.</p> <p>→ 2. The user clicks “CREATE TEAM” button.</p> <p>← 3. The system redirects the registered user to team creation page, to fill the team data.</p> <p>→ 4. The user chooses team name and tag.</p> <p>← 5. The system creates an ID for the team.</p> <p>← 6. The system shows “CREATE” and “DECLINE” buttons on the end of the page.</p> <p>→ 7. The user clicks “CREATE” button.</p> <p>← 8. The system shows a message “Congratulation, you have created a team successfully.</p> <p>← 9. The system makes the registered user who create the team a team leader.</p>				
Alternate scenario Flow (1)				
<p>→ 7.1 The user clicks “DECLINE” button.</p> <p>← 1. The system redirects the registered user to homepage.</p>				

Table 3.8: Use case 08

Use Case ID:	UC 008					
Use Case Name:	Add other registered users to the team					
Related requirements:	Fr 015	Related Use Case:	UC 002			
Actor:	1. Team leader 2. Database					
Actor's Goal:	Add other registered users to the team					
Preconditions:	1. User must be registered, and all his / her data and preferences are existing in the data base. 2. Team is created and exist in the database. 3. Another user must be registered.					
Postconditions:	Team have more than one member.					
Main success scenario Flow						
↔ 1. Repeat use case UC 002 scenarios. → 2. The team leader clicks “TEAM” button. ← 3. The system redirects the registered user to the team page. → 4. The team leader clicks “ADD TEAM MEMBER” button. ← 5. The system opens a window, in it the team leader types the registered user e-mail that he wants to add him to the team. → 6. The team leader types the registered user e-mail. ← 7. The system shows in the same window “ADD USER” and “DECLINE” buttons. → 8. The team leader clicks “ADD USER” button. ← 9. The database checks if the e-mail that the team leader has added is already exist. ← 10. The database finds the e-mail that the team leader has added is already exist. ← 11. The system shows a message “We have sent an adding request to the user e-mail”.						
Alternate scenario Flow (1)						
→ 8.1 The team leader clicks “DECLINE” button. ← 1. The system redirects the registered user to team page.						
Alternate scenario Flow (2)						
← 10.2 The database finds the e-mail that the team leader has added is not exist. ← 1. The system shows a message “The e-mail you have entered is not exist, try another e-mail.” ← 2. The system shows in “OK” and “CANCEL” button. → 3. The team leader clicks “OK” button. ↔ 4. REPEAT use case UC 008 main scenario.						
Alternate scenario Flow (3)						
← 10.3 The database finds the e-mail that the team leader has added is not exist. ← 1. The system shows a message “The e-mail you have entered is not exist, try another e-mail.” ← 2. The system shows in “OK” and “CANCEL” button. → 3. The team leader clicks “CANCEL” button. ← 4. The system redirects the registered user to team page.						

Table 3.9: Use case 09

Use Case ID:	UC 009			
Use Case Name:	Contribute project's tasks			
Related requirements:	Fr 016	Related Use Case:		
Actor:	1. System			
Actor's Goal:	System contributes the project's tasks on the team members depending on the members preferences			
Preconditions:	1. User must be registered, and all his / her data and preferences are existing in the data base. 2. Team is created and exist in the database. 3. Team leader has chosen a project.			
Postconditions:	Every member in the team knows his/her task, and that task exist in database.			
Main success scenario Flow				
← 1. System compares team members preferences with project requirements. ← 2. The system contributes the project's tasks depending on team members preferences.				

Table 3.10: Use case 10

Use Case ID:	UC 010			
Use Case Name:	Recommending Learning resources / new team member			
Related requirements:	Fr 017, Fr 019, Fr 018, Fr 020	Related Use Case: UC 002		
Actor:	1. Team leader 2. Registered user without a team			
Actor's Goal:	System provides Learning resources to the team for the missing technology / tool, or a new team member can work with the team's missed tool.			
Preconditions:	1. User must be registered, and all his / her data and preferences are existing in the data base. 2. Team is created and exist in the database. 3. Team leader has chosen a project. 4. The recommended new member hasn't joined a team.			
Postconditions:	Team gets the learning resources for the tool/technology, that they missed them in the project requirements, or a new team member can work with the team's missed tool.			
Main success scenario Flow				
→ 1. Repeat use case UC 002 scenarios. → 2. Team leader clicks "PROJECTS" button on the homepage. ← 3. System redirects team member to "Project Dashboard" page. ← 4. System shows a message "Your team missed a tool name, that required in this project, Do you want to learning resources for this tool name, or recommend a new team member know this tool?" ← 5. System shows "LEARNING RESOURCES" and "NEW MEMBER" buttons. → 6. Team leader clicks "LEARNING RESOURCES" button. ← 7. AI Engine shows the recommended learning resources for a specific tool. ← 8. System chooses a team member that will learn the missing tool.				
Alternate scenario Flow (1)				
→ 7.1 Team leader clicks "NEW MEMBER" buttons. ← 1. System compares the registered users who haven't a team, and their preferences met the missed tool in the chosen project. ← 2. System shows a list of the recommended new team members. ← 3. System shows beside every new recommended team member "SEND JOIN REQUEST" button. → 4. Team leader clicks "SEND JOIN REQUEST" button. ← 5. System sends the join request to the new recommended team member.				

Table 3.11: Use case 11

Use Case ID:	UC 011			
Use Case Name:	Test the users in the topics, that they have learned.			
Related requirements:	Fr 021	Related Use Case: UC 002		
Actor:	1. System 2. Team member			
Actor's Goal:	System tests the team member that was recommended to learn a specific tool.			
Preconditions:	1. User must be registered, and all his / her data and preferences are existing in the data base. 2. Team is created and exist in the database. 3. Team leader has chosen a project. 4. System recommends learning resources for a missing tool / technology. 5. Team member has chosen to learn a specific tool.			
Postconditions:	Team member should pass the test and the learned tool will be add to his / her preferences in the database.			
Main success scenario Flow				
↔ 1. Repeat use case UC 002 scenarios. → 2. Team member clicks “PROJECTS” button on the homepage. ← 3. System redirects team member to “Project Dashboard” page. ← 4. System shows “FINISH COURSE” button in the dashboard. → 5. Team member clicks “FINISH COURSE” button. ← 6. System redirects the team member to the test page. ← 7. System validates the team member answers. ← 8. System finds the team member passed the test. ← 9. System adds the new learned tool the team member preferences in the database.				
Alternate scenario Flow (1)				
← 10.1 System finds the team member passed the test. ← 1. System shows a message “You didn’t pass the test; you have to review the course again.” ← 2. System shows “OK” button. → 3. Team member clicks “OK” button. ← 4. System redirects the team member to the course page.				

Table 3.12: Use case 12

Use Case ID:	UC 012			
Use Case Name:	Deactivate the unlearned technology / tool tasks			
Related requirements:	Fr 022	Related Use Case:		
Actor:	1. System			
Actor's Goal:	Deactivate the unlearned technology/tool tasks until the team member learn it			
Preconditions:	1. User must be registered, and all his / her data and preferences are existing in the data base. 2. Team is created and exist in the database. 3. Team leader has chosen a project. 4. System recommend learning resources for a missing tool / technology.			
Postconditions:	The related task of the missed tool / technology deactivated from dashboard.			
Main success scenario Flow				
← 1. System deactivates the unlearned technology/tool tasks.				

Table 3.13: Use case 13

Use Case ID:	UC 013					
Use Case Name:	Determine start / end date of the project					
Related requirements:	Fr 023	Related Use Case:	UC 002			
Actor:	1. Team leader					
Actor's Goal:	Determine start / end date of the projects					
Preconditions:	1. User must be registered, and all his / her data and preferences are existing in the data base. 2. Team is created and exist in the database. 3. Team leader has chosen a project.					
Postconditions:	1. Team leader sets the required dates and these dates stored in the database. 2. The defined dates must be shown in the project dashboard.					
Main success scenario Flow						
↔ 1. Repeat use case UC 002 scenarios. → 2. Team leader clicks “PROJECTS” button on the homepage. ← 3. System redirects team leader to “Project Dashboard” page. → 4. Team leader clicks “SET PROJECT DATES” button. ← 5. System shows new window contains “Start date” “End date” empty fields and “SET” “CANCEL” buttons. → 6. Team leader set the required dates. → 7. Team leader clicks “SET” buttons. ← 8. System redirects the team leader to the project dashboard.						
Alternate scenario Flow (1)						
→ 7.1 Team leader clicks “CANCEL” buttons. ← 1. System redirects the team leader to the project dashboard.						

Table 3.14: Use case 14

Use Case ID:	UC 014			
Use Case Name:	Delay start / end date of the projects			
Related requirements:	Fr 024	Related Use Case: UC 002		
Actor:	1. Team leader			
Actor's Goal:	Determine start / end date of the projects			
Preconditions:	1. User must be registered, and all his / her data and preferences are existing in the data base. 2. Team is created and exist in the database. 3. Team leader has chosen a project. 4. Start / end date of the projects must be set.			
Postconditions:	1. Team leader edit the end / start dates and these dates updated in database. 2. The edited dates must be updated in the project dashboard.			
Main success scenario Flow				
↔ 1. Repeat use case UC 002 scenarios. → 2. Team leader clicks “PROJECTS” button on the homepage. ← 3. System redirects team leader to “Project Dashboard” page. → 4. Team leader clicks “Edit PROJECT DATES” button. ← 5. System shows new window contains the previous start / end date in the “Start date” “End date” fields and “SET” “CANCEL” buttons. → 6. Team leader edits the start / end dates. → 7. Team leader clicks “SET” buttons. ← 8. System redirects the team leader to the project dashboard.				
Alternate scenario Flow (1)				
→ 7.1 Team leader clicks “CANCEL” buttons. ← 1. System redirects the team leader to the project dashboard.				

Table 3.15: Use case 15

Use Case ID:	UC 015					
Use Case Name:	Remove a member from team.					
Related requirements:	Fr 025	Related Use Case:	UC 002			
Actor:	1. Team leader					
Actor's Goal:	Remove a member from team.					
Preconditions:	1. User must be registered, and all his / her data and preferences are existing in the data base. 2. Team is created and exist in the database. 3. Team leader has chosen a project. 4. A user must be a member in a team.					
Postconditions:	1. A specific member is removed from a team. 2. The new team members state be updated in database.					
Main success scenario Flow						
↔ 1. Repeat use case UC 002 scenarios. → 2. Team leader clicks “PROJECTS” button on the homepage. ← 3. System redirects team leader to “Project Dashboard” page. → 4. Team leader clicks “REMOVE MEMBER” button. ← 5. System shows new window contains a list of the team member names and “X” button beside each name. → 6. Team leader clicks “X” button beside a specific name. ← 7. System shows a new window contains a message “Are you sure to remove member name” and “REMOVE” “CANCEL” buttons. → 8. Team leader clicks “REMOVE” buttons. ← 9. System redirects the team leader to the project dashboard.						
Alternate scenario Flow (1)						
→ 7.1 Team leader clicks “CANCEL” buttons. ← 1. System redirects the team leader to the project dashboard.						

Table 3.16: Use case 16

Use Case ID:	UC 016					
Use Case Name:	Deactivate a chosen project.					
Related requirements:	Fr 026	Related Use Case:	UC 002			
Actor:	1. Team leader					
Actor's Goal:	Deactivate a chosen project.					
Preconditions:	1. User must be registered, and all his / her data and preferences are existing in the data base. 2. Team is created and exist in the database. 3. Team leader has chosen a project.					
Postconditions:	1. The chosen project dashboard must be deactivated. 2. Only team member can access the project dashboard.					
Main success scenario Flow						
↔ 1. Repeat use case UC 002 scenarios. → 2. Team leader clicks “PROJECTS” button on the homepage. ← 3. System redirects team leader to “Project Dashboard” page. → 4. Team leader clicks “DEAVTIVATE PROJECT” button. ← 5. System shows a new window contains a message “Are you sure to deactivate the project” and “DEACTIVATE” “CANCEL” buttons. → 6. Team leader clicks “DEACTIVATE” buttons. ← 7. System redirects the team leader to the homepage.						
Alternate scenario Flow (1)						
→ 7.1 Team leader clicks “CANCEL” buttons. ← 1. System redirects the team leader to the project dashboard.						

Table 3.17: Use case 17

Use Case ID:	UC 017					
Use Case Name:	Disjoin a team.					
Related requirements:	Fr 027, Fr 033	Related Use Case:	UC 002			
Actor:	1. User					
Actor's Goal:	Disjoin a team.					
Preconditions:	1. User must be registered, and all his / her data and preferences are existing in the data base. 2. Team is created and exist in the database. 3. The recommended user must be a member in a team.					
Postconditions:	1. A specific member is disjoined a team. 2. The new team members state be updated in database.					
Main success scenario Flow						
↔ 1. Repeat use case UC 002 scenarios. → 2. User clicks “MY TEAM” button on the homepage. ← 3. System redirects team leader to “TEAM” page. → 4. User clicks “disjoin team” button. ← 5. System shows a new window contains a message “Are you sure to disjoin member name” and “DISJOIN” “CANCEL” buttons. → 6. User clicks “DISJOIN” buttons. ← 7. System deactivates the lifted member tasks.						
Alternate scenario Flow (1)						
→ 7.1 User clicks “CANCEL” buttons. ← 1. System redirects the team member to home page.						

Table 3.18: Use case 18

Use Case ID:	UC 018					
Use Case Name:	Recommend a new member for a team is lifted by another user					
Related requirements:	Fr 029	Related Use Case:	UC 002			
Actor:	1. Team Leader 2. User					
Actor's Goal:	Recommend a new team member, able to work by a specific technology/tool at the project's predetermined dates, when a team member disjoins the team.					
Preconditions:	1. User must be registered, and all his / her data and preferences are existing in the data base. 2. Team is created and exists in the database. 3. The new member hasn't joined a team.					
Postconditions:	1. A new member joins a team. 2. The new team members state be updated in database.					
Main success scenario Flow						
<p>↔ 1. Repeat use case UC 002 scenarios.</p> <p>→ 2. Team leader clicks “Project” button on the homepage. ← 3. System redirects team leader to “Project” dashboard. ← 4. System shows “RECOMMEND” “CANCEL” buttons. → 5. Team leader clicks “RECOMMEND” button. ← 6. System compares the registered users who haven’t a team, and their preferences met the task requirements of the disjoined member. ← 7. System shows a list of the recommended new team members. ← 8. System shows beside every new recommended team member “SEND JOIN REQUEST” button. → 9. Team leader clicks “SEND JOIN REQUEST” button. ← 10. System sends the join request to the new recommended team member. → 11. The chosen registered user receives Join Request to a team. ← 12. System shows a new window contains a message “You have recommended to work with team name” and “ACCEPT” “CANCEL” buttons. → 13. The chosen registered user clicks “ACCEPT” button.</p>						
Alternate scenario Flow (1)						
<p>→ 5.1 Team leader clicks “CANCEL” buttons. ← 1. System redirects the team leader to project dashboard.</p>						
Alternate scenario Flow (2)						
<p>→ 13.2 The chosen registered user clicks “CANCEL” button. ← 1. System shows a new window contains a message “Member refused your request, Do you want to recommend a new team member” and “RECOMMEND” “CANCEL” buttons. → 2. Team leader clicks “RECOMMEND” button. ← 3. System compares the registered users who haven’t a team, and their preferences met the task requirements of the disjoined member. ← 4. System shows a list of the recommended new team members. ← 5. System shows beside every new recommended team member “SEND JOIN REQUEST” button. → 6. Team leader clicks “SEND JOIN REQUEST” button. ← 7. System sends the join request to the new recommended team member. → 8. The chosen registered user receives Join Request to a team. ← 9. System shows a new window contains a message “You have recommended to work with team name” and “ACCEPT” “CANCEL” buttons. → 10. The chosen registered user clicks “ACCEPT” button.</p>						

Table 3.19: Use case 19

Use Case ID:	UC 019			
Use Case Name:	Deactivate dashboard			
Related requirements:	Fr 032	Related Use Case:		
Actor:	1. User			
Actor's Goal:	Project's dashboard temporarily deactivated when the team leader quit the project			
Preconditions:	1. Team is created and exist in the database. 2. A project has been chosen 3. Project is activated. 3. Team leader has quitted the project.			
Postconditions:	1. Deactivate project's dashboard. 2. Choose a new team leader.			
Main success scenario Flow				
← 1. System shows a new window contains a message “Team leader have quitted the project, The dashboard is temporarily deactivated, until choose a new team leader” and “New TEAM LEADER” “CANCEL” buttons. → 2. User clicks “New TEAM LEADER” button. ← 3. System shows a list of the team members' names. → 4. Each User selects a new team member by clicks a team member name. ← 5. System counts the votes and make the member with the highest votes a team leader.				
Alternate scenario Flow (1)				
→ 2.1 User clicks “CANCEL” buttons. ← 1. System redirects the team member to project dashboard.				

Table 3.20: Use case 20

Use Case ID:	UC 020			
Use Case Name:	Chatbot			
Related requirements:	Fr 034, Fr 035, Fr 036			
Actor:	1. User			
Actor's Goal:	Team members communicate through chatbot.			
Preconditions:	1. User must be registered, and all his / her data and preferences are existing in the data base. 2. Team is created and exist in the database. 3. Project has been chosen			
Postconditions:	1. Chat register has been created and exit in database. 2. Conversation history is kept. 3. The chat conversation is shown for all team members.			
Main success scenario Flow				
→ 1. Repeat use case UC 002 scenarios. → 2. User clicks “Project” button on the homepage. ← 3. System redirects team leader to “Project” dashboard. → 4. User clicks “Chat” tap on Project dashboard. ← 5. System shows a new window contains a “text typing field”, “SEND” button and “chat history window”. → 6. User types a text in the “text typing field”. → 7. User clicks “SEND” button. ← 8. System shows the sent message in the chat history with his her “name / e-mail / ID”				

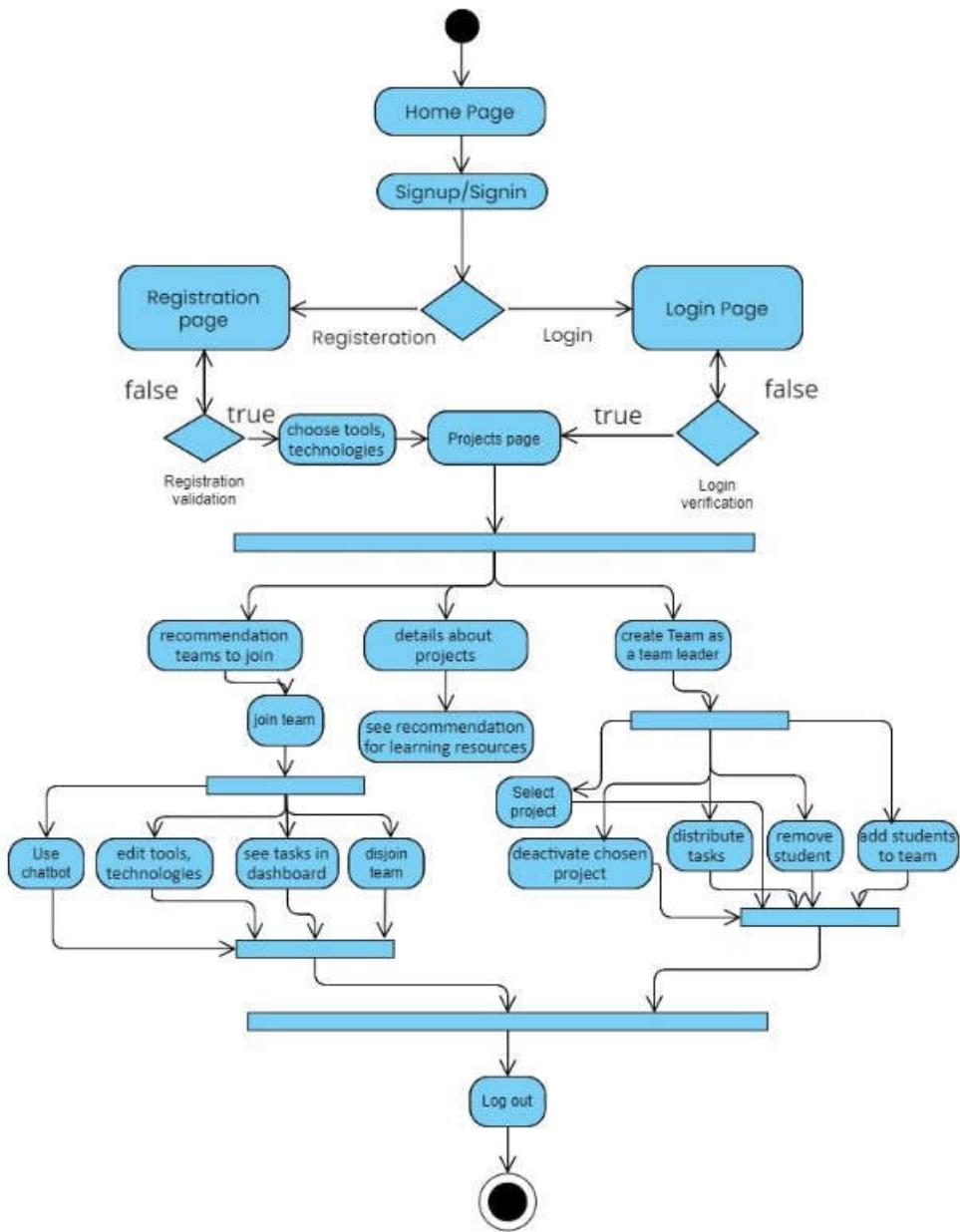
3.3 Traceability Matrix

Table 3.21: Traceability Matrix

REQ.	PW	UC - 1	UC - 2	UC - 3	UC - 4	UC - 5	UC - 6	UC - 7	UC - 8	UC - 9	UC - 10	UC - 11	UC - 12	UC - 13	UC - 14	UC - 15	UC - 16	UC - 17	UC - 18	UC - 19	UC - 20
REQ.1	5	X																			
REQ.2	5		X																		
REQ.3	3			X																	
REQ.4	4				X																
REQ.5	5	X																			
REQ.6	2					X															
REQ.7	5						X														
REQ.8	4							X													
REQ.9	5								X												
REQ.10	5									X											
REQ.11	3									X											
REQ.12	3										X										
REQ.13	5										X										
REQ.14	2										X										
REQ.15	4											X									
REQ.16	3											X									
REQ.17	3												X								
REQ.18	4												X								
REQ.19	5												X								
REQ.20	4												X								
REQ.21	3													X							
REQ.22	1													X							
REQ.23	2														X						
REQ.24	2															X					
REQ.25	3																X				
REQ.26	2																	X			
REQ.27	3																		X		
REQ.28	2																			X	
REQ.29	3																				X
REQ.30	1																				X
REQ.31	3																				X
REQ.32	3																				X
REQ.33	3																				X
REQ.34	3																				X
REQ.35	3																				X
REQ.36	3																				X
Max PW	5	5	2	4	5	5	4	3	5	3	1	2	2	3							
Total PW	10	5	2	7	9	15	4	3	16	3	1	2	2	3	2	6	3	3	9	9	9

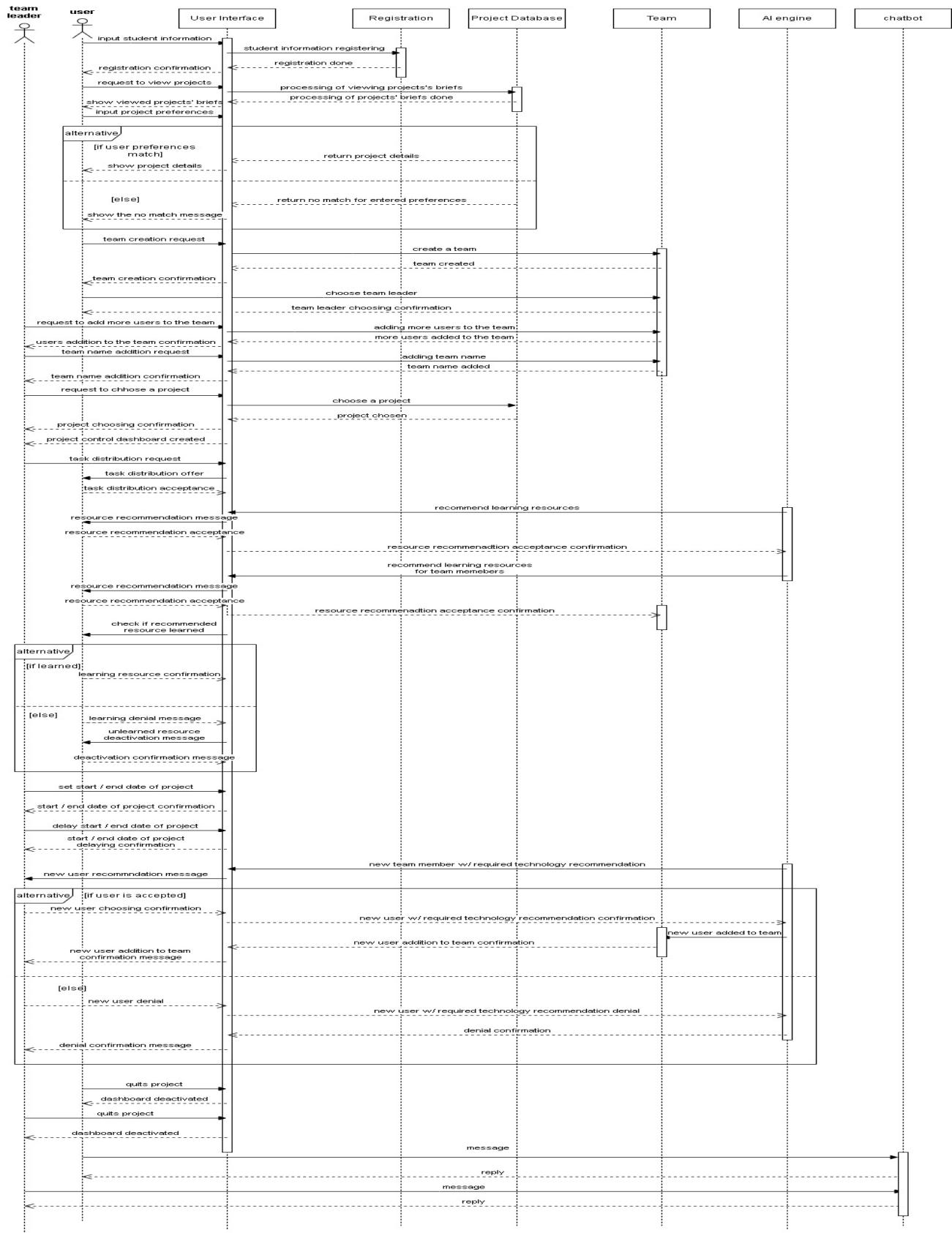
3.4 Activity diagram

Fig 3.2: Activity diagram



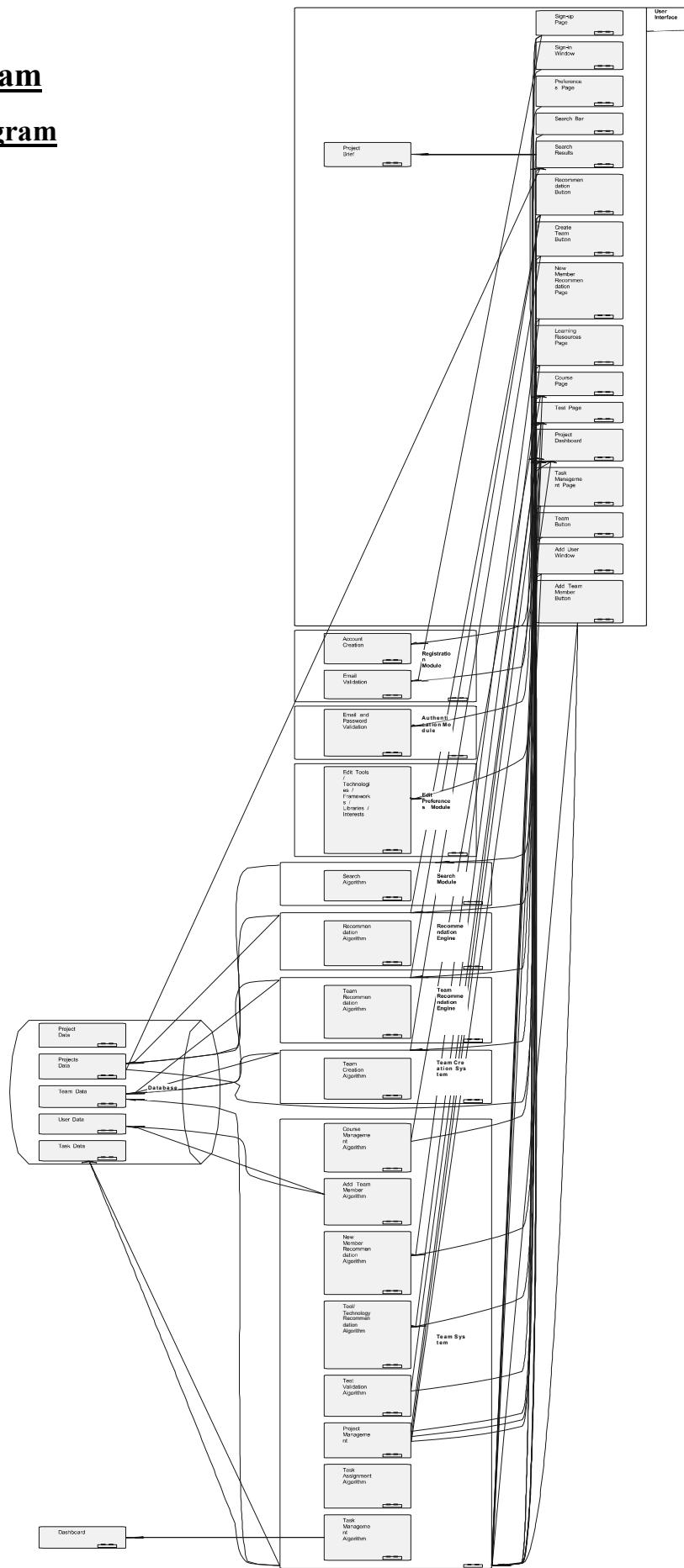
3.5 Sequence diagram

Fig 3.3: Sequence diagram



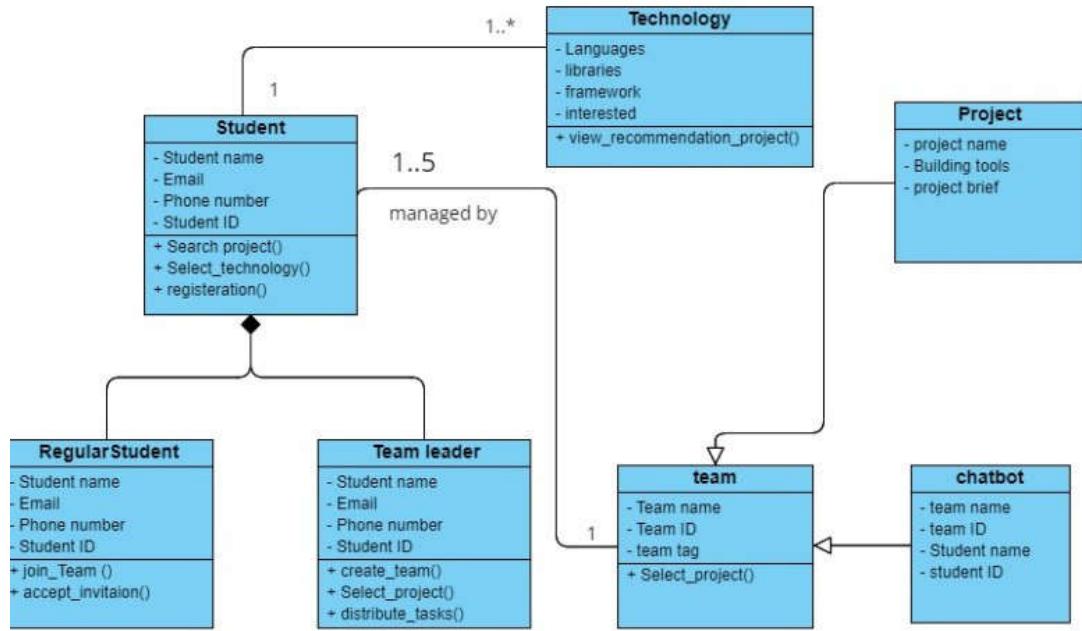
3.6 Component diagram

Fig 3.4: Component diagram



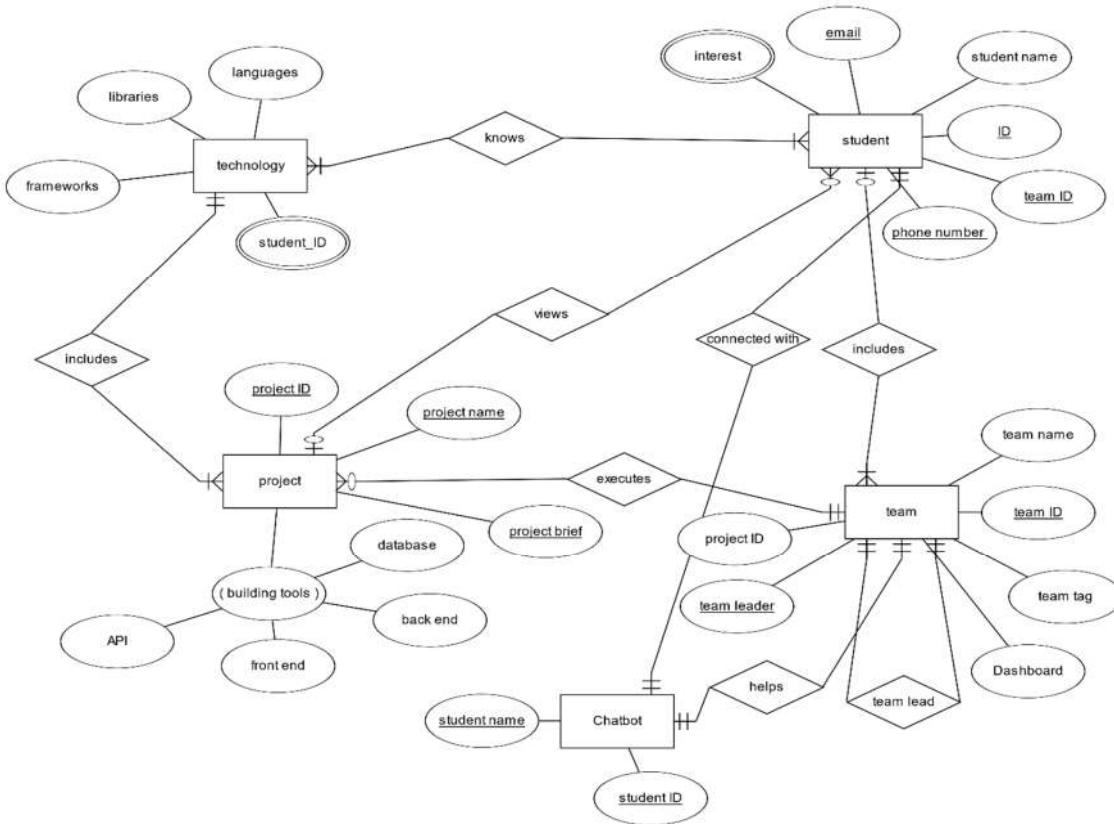
3.7 Class diagram

Fig 3.5: Class diagram



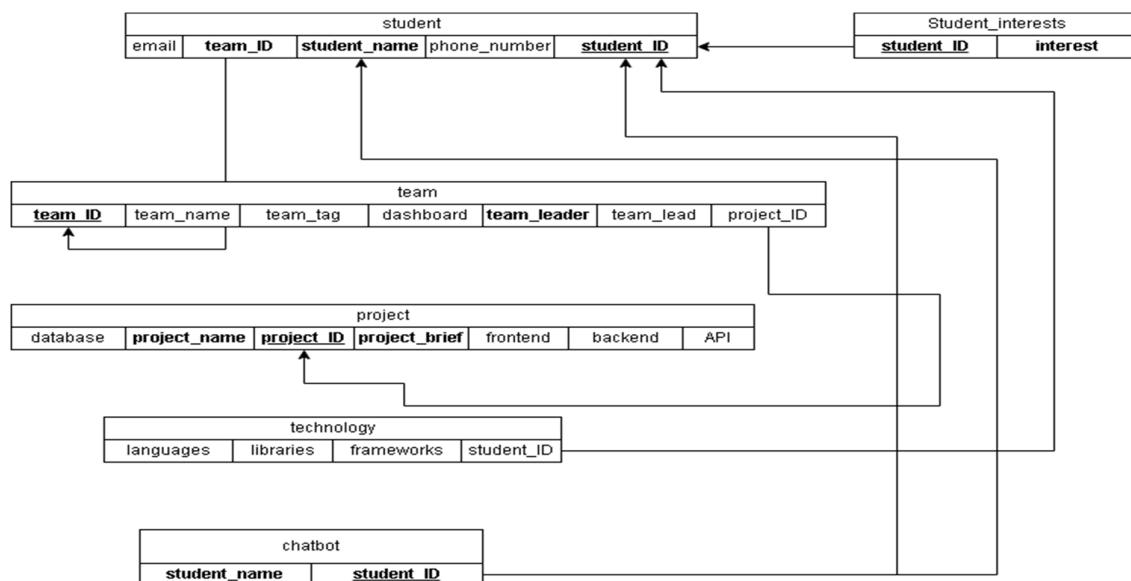
3.8 Entity relationship diagram (ERD)

Fig 3.6: Entity relationship diagram (ERD)



3.9 Mapping

Fig 3.7: Mapping



**Chapter 4: System Implementation
And
Testing Plan**

4.1 Interfaces

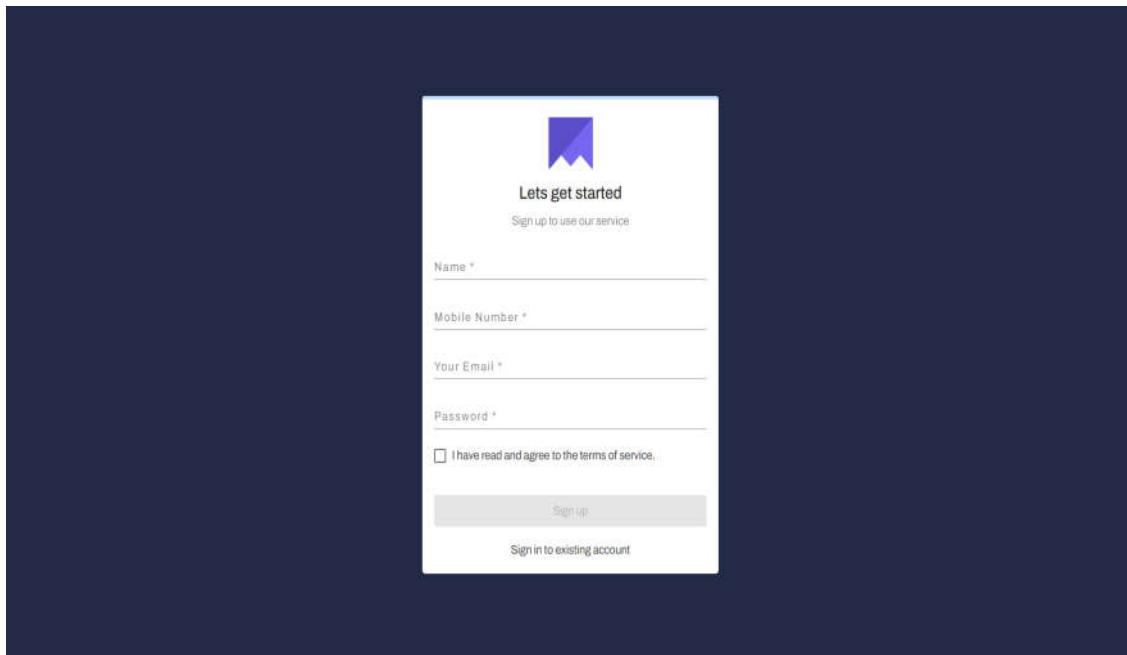
4.1.1 Home page

Fig 4.1 Home page



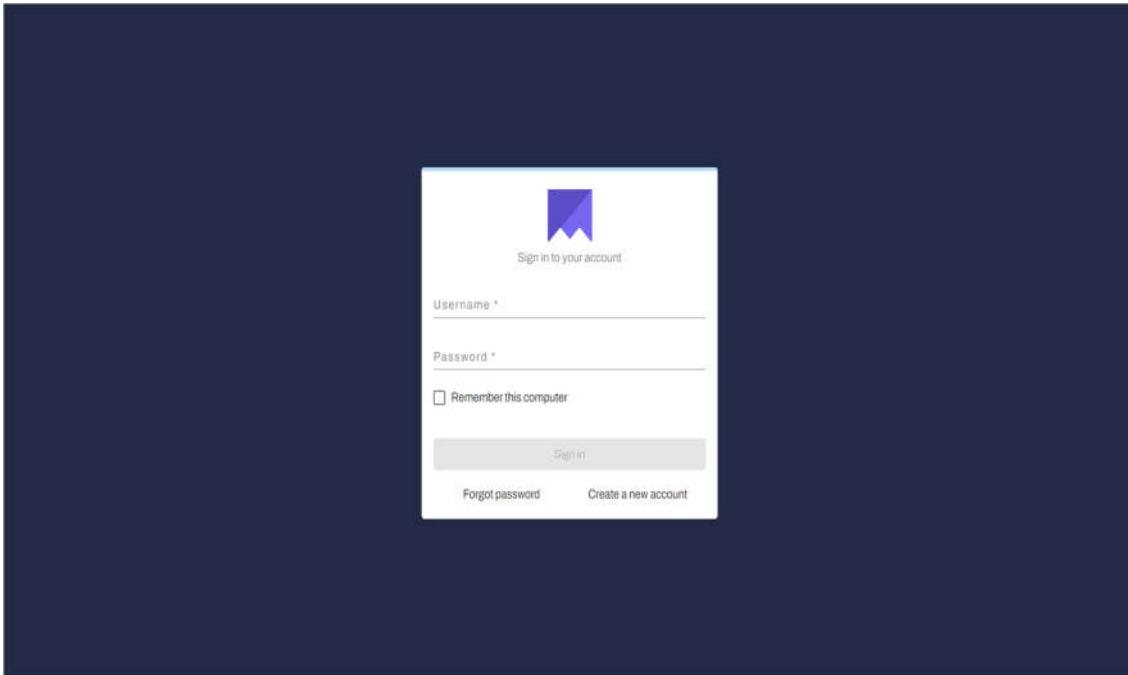
4.1.2 Sign up page

Fig 4.2 Sign up page



4.1.3 Sign in page

Fig 4.3 Sign in page



4.1.4 Dashboard page

Fig 4.4 Dashboard page

A screenshot of a dashboard page for the "FGSSR Project". The left sidebar has a dark blue background with white icons and text: "Welcome Admin", "Dashboard" (highlighted in blue), "Communications", "Projects", "Chat", "Team Member", "Skills", "Tasks", "Sources", "Team Request", "Forms", and "User Data". The main area has a light gray header with "Dashboard / Analytics". Below it is a grid of six colored boxes: "Add Project" (light blue), "Add Memebr" (light green), "Add Task" (pink), "Learning Resources" (red), "Create Team" (blue), and "Add Skills" (purple). At the bottom, there are input fields for "Team Name" and "Project Name", and a section for "Ongoing Tasks". A blue circular button with a gear icon is in the bottom right corner.

4.1.5 Skills selection & Projects recommendations page

Fig 4.5 Skills selection & Projects recommendations page

The screenshot shows a web application interface for 'FGSSR Project'. On the left, a dark sidebar menu lists various project management tasks: Dashboard, Communications, Projects, Chat, Team Member, Skills (which is selected and highlighted in blue), Tasks, Sources, Team Request, Forms, User Data, and Create Team. The main content area has a light orange header 'Skills' and a sub-header 'Add Your Skills'. Below this is a search bar containing 'HTML x CSS x JavaScript x PHP or Python x MySQL or PostgreSQL x'. A large, empty text input field follows. At the bottom right of the main area is a 'Submit' button. Below the main area, under 'Recommendations Project', is a list of five projects with their similarity scores:

- 1. Project Online Therapy Platform Development (Similarity score: 0.9949)
- 2. Project E-commerce Website Development (Similarity score: 0.9948)
- 3. Project Social Media App Development (Similarity score: 0.9948)
- 4. Project Online Education Platform Development (Similarity score: 0.9948)
- 5. Project Restaurant Management System (Similarity score: 0.9948)

4.1.6 Learning resources recommendations page

Fig 4.6 Learning resources recommendations page

The screenshot shows the 'Sources For Learning' section of the FGSSR Project. The sidebar menu is identical to Fig 4.5. The main content area has a light orange header 'Sources For Learning' and a sub-header 'Choose your skill to see learning resources'. A dropdown menu is open, showing 'C++' as the selected skill. To the right of the dropdown is a 'Submit' button. Below this, under 'Learning Sources', are three recommended learning resources:

- Name of Channel: Apna College
Title: C Language Tutorial for Beginners (with Notes & Practice Questions)
URL: <https://www.youtube.com/watch?v=irqbMNz2Bo>
Views: 16908307
- Name of Channel: freeCodeCamp.org
Title: C++ Tutorial for Beginners - Full Course
URL: <https://www.youtube.com/watch?v=vLnPwxZdW4Y>
Views: 11213587
- Name of Channel: CodeWithHarry
Title: Introduction to C++, Installing VS Code, g++ & more | C++ Tutorials for Beginners #1

4.1.7 Create team page

Fig 4.7 Create team page

The screenshot shows the 'Create Team' page of the FGSSR Project application. The left sidebar has a dark blue background with white icons and text. It includes 'Dashboard', 'Communications', 'Projects', 'Chat', 'Team Member', 'Skills', 'Tasks', 'Sources', 'Team Request', 'Forms', 'User Data', and a highlighted 'Create Team' button. The main area has a light gray background with a header 'Forms / Create Team'. Below it is a 'Create Team' section with four numbered steps: 1. Fill out our Team Name, 2. Fill out our Team Tag, 3. Fill out Team Description, and 4. Done. Step 1 is active, showing a 'Team Name *' input field and a 'Next' button. A progress bar at the bottom of the steps indicates the current step.

4.1.8 Create task page

Fig 4.8 Create task page

The screenshot shows the 'Team Tasks' page of the FGSSR Project application. The left sidebar has a dark blue background with white icons and text. It includes 'Dashboard', 'Communications', 'Projects', 'Chat', 'Team Member', 'Skills', a highlighted 'Tasks' button, 'Sources', 'Team Request', 'Forms', 'User Data', and 'Create Team'. The main area has a light gray background with a header 'Team Tasks'. Below it is a 'Team Tasks' section with two input fields: 'Enter Task' and 'Enter Description', followed by an 'Add Task' button.

4.1.9 User data page

Fig 4.9 User data page

The screenshot shows a user interface for a project management application named "FGSSR Project". On the left, there is a dark sidebar with various navigation options: Dashboard, Communications, Projects, Chat, Team Member, Skills, Tasks, Sources, Team Request, Forms, User Data (which is highlighted in blue), and Create Team. The main content area is titled "Form validation" and contains the following fields:

- Username (Min Length: 4, Max Length: 10): Admin
- Password *
- Email *: Admin@123.com
- Confirm Password
- Your Website (http://mrafi.com)
- Gender*: Male (radio button selected)
- Female (radio button)
- I have read and agree to the terms of service. (checkbox)
- Your job
- Profile picture:
Choose File (No file chosen)
Select

A blue circular icon with a gear symbol is located in the bottom right corner of the main content area.

4.2 AI Engine

WHAT IS RECOMMENDATION SYSTEM

- A recommendation system is a type of information filtering system that suggests relevant items to users based on their preferences, behavior, or past interactions. The goal of a recommendation system is to provide personalized and relevant recommendations to users, thereby helping them discover new products, services, or content that align with their interests and needs.
 - Overall, recommendation systems play a vital role in providing personalized experiences, improving user engagement, and driving business success in various industries.
-

WHAT ARE TYPES OF REC SYSTEM

- **Recommendation systems** are important because they help users discover new and relevant items that they might not have found on their own
- **Content-based filtering:** systems recommend items that are similar to items that the user has rated or expressed interest in in the past. For example, if a user has rated a movie as 5 stars, a content-based filtering system might recommend other movies that have similar ratings or reviews.
- **Collaborative Filtering:** systems recommend items that other users who have similar interests to the user have rated highly. For example, if a user has rated a book as 5 stars, a collaborative filtering system might recommend other books that have been rated highly by other users who have also rated the same book highly.

WHICH TYPE OF RECOMMENDATION SYSTEM

- **Traditional recommendation systems** like SVD and matrix factorization work by decomposing the user-item interaction matrix into a lower-dimensional space. This is done by finding latent factors that represent the user's preferences and the item's features. The recommendations are then made by predicting the user's rating for an item based on the latent factors.
 - **Deep learning recommendation systems** use neural networks to learn the relationship between users and items. Neural networks are able to learn more complex relationships than traditional methods, and they can be used to handle large datasets with sparse data.
 -
-

WHAT ARE TYPES OF REC SYSTEMS

- Here are some of the most common types of recommendation systems:
- **User-based collaborative filtering:** This is the most common type of collaborative filtering. It works by finding other users who have similar interests to the current user and then recommending items that those users have rated highly.
- **Item-based collaborative filtering:** This type of collaborative filtering works by finding items that are similar to items that the current user has rated highly and then recommending those items.
- **Hybrid recommendation systems:** These systems combine content-based and collaborative filtering techniques. They can be more accurate than either type of system on its own.

WHICH TYPE OF RECOMMENDATION SYSTEM

Feature	Traditional	Deep Learning
Method	Decomposition	Neural networks
Data requirements	Dense data	Sparse data
Accuracy	Good	Excellent
Complexity	Simple	Complex
Training time	Fast	Slow

WHY DEEP LEARNING MODEL

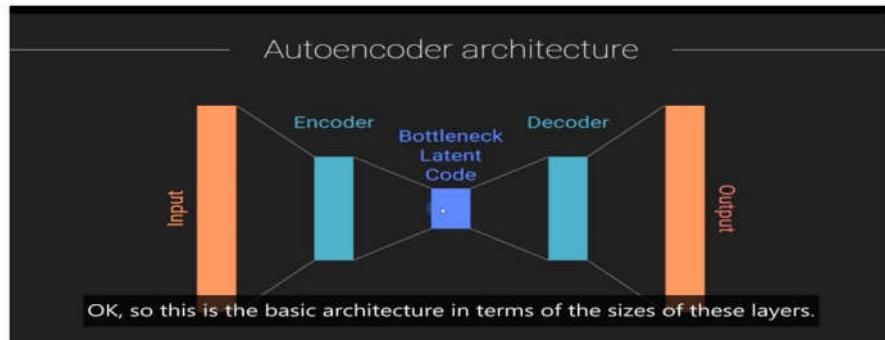
- **Better accuracy:** Deep learning recommendation systems can achieve better accuracy than traditional methods, especially for large datasets with sparse data.
- **More flexibility:** Deep learning recommendation systems are more flexible than traditional methods and can be used to handle a wider variety of tasks, such as personalized ranking and product recommendation.
- **More scalable:** Deep learning recommendation systems can be scaled to handle large datasets and high traffic.

TYPES OF DEEP LEARNING MODELS

- **Deep autoencoder:** A deep autoencoder is a neural network that learns to reconstruct its input data. This can be used to build recommendation systems by encoding user and item features into a latent space and then decoding those features to generate recommendations.
 - **Recurrent neural network:** A recurrent neural network (RNN) is a type of neural network that is well-suited for processing data that has a temporal structure, such as sequences of words. RNNs can be used to build recommendation systems by processing the user and item features as sequences of words.
 - **Convolutional neural network:** A convolutional neural network (CNN) is a type of neural network that is well-suited for processing data that has a grid-like structure, such as images. CNNs can be used to build recommendation systems by processing the user and item features as images.
-

WHY DEEP AUTOENCODERS

- **Non-linear Mapping:** Deep autoencoders can capture complex non-linear relationships between users and items.
 - **Feature Learning:** Autoencoders are unsupervised learning models that can learn meaningful latent representations or features from the input data. In the context of recommendation systems, this means that deep autoencoders can automatically discover relevant user and item features without explicit feature engineering.
 - **Collaborative Filtering:** Deep autoencoders can effectively perform collaborative filtering, which is a common approach in recommendation systems. By learning from user-item interaction data, deep autoencoders can generate accurate recommendations by capturing the underlying collaborative patterns and preferences of users.
-



WHAT IS DEEP AUTOENCODER

- **A deep autoencoder** for recommendation systems is like a smart assistant that learns from the preferences and behaviors of users to make personalized recommendations. It works by understanding the underlying patterns and relationships between users and items in a dataset.
- **Here's how it works:** The deep autoencoder takes the information about the skills and projects as input. It has several hidden layers that learn to capture the important features and patterns in the data. These hidden layers act like a filter that processes and transforms the input information, gradually learning to represent it in a more meaningful way.
- **The magic** happens in the hidden layers of the autoencoder. They learn to extract the most relevant and distinguishing features of the users and items.

WHAT IS AUTOENCODER

- deep autoencoder structure consists of an encoder that compresses the input data into a lower-dimensional representation, and a decoder that reconstructs the original input from this compressed representation. The hidden layers within the encoder and decoder learn to extract relevant features and transform the data, ultimately enabling the autoencoder to capture the essential patterns and make accurate recommendations.

PROJECT DATASET

- We have utilized a professionally curated dataset extracted from the Stack Overflow Developer Survey. This dataset serves as a valuable resource for training our recommendation system
- The dataset containing information from 85,000 programmers, encompassing their skills and projects. This extensive dataset forms the backbone of our recommendation system, allowing us to leverage a vast pool of knowledge and experiences.
- By analyzing the skills and projects of these 85,000 programmers, our recommendation system can identify meaningful patterns and connections. It can then offer personalized suggestions to programmers based on their individual skills, interests, and project requirements. This large-scale dataset enables us to deliver precise and tailored recommendations, catering to the diverse needs and preferences of our users.

HOW IT WORKS

- **The model** takes an input of dimensions `input_dim`, which represents the features or attributes of the programmers and projects in our dataset. The input is passed through a series of dense layers, starting with an encoder layer.
 - **The encoder** layer is responsible for reducing the dimensionality of the input data and extracting its essential features. It is implemented using a dense layer, which applies the Rectified Linear Unit (ReLU) activation function.
 - After the **encoder** layer, we have a series of additional dense layers, with each layer having a number of units defined by `hidden_units`. These layers progressively learn more abstract representations of the input data, capturing higher-level features and patterns.
-

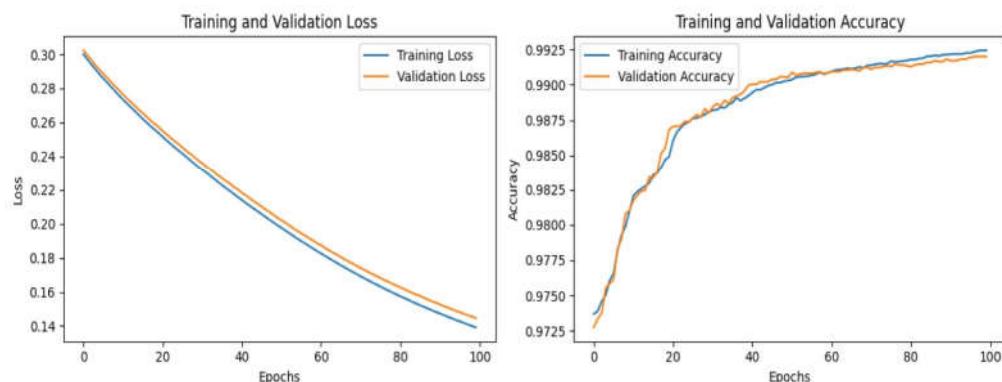
HOW IT WORKS

- Once the encoded representation is obtained, we move to the decoder part of the model. The decoder aims to reconstruct the original input from the compressed representation. It consists of dense layers with decreasing units, mirroring the structure of the encoder layers. This process allows the model to reconstruct the input data by gradually expanding the dimensions.
 - The final layer of the model is a dense layer with `output_dim` units, using the softmax activation function. This layer generates the output predictions, which correspond to the recommended projects for the programmers. The softmax activation ensures that the output represents a probability distribution over the available projects.
-

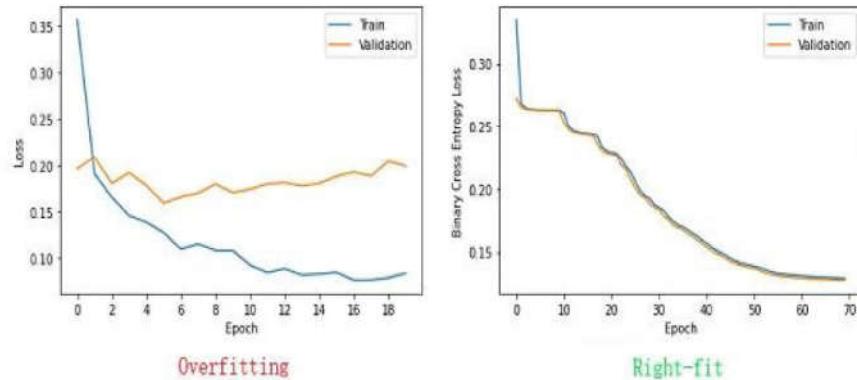
WHAT IS OVERFITTING

- It occurs when a model becomes too specialized in learning the training data and fails to generalize well to new, unseen data. In other words, the model "memorizes" the training examples instead of learning the underlying patterns and relationships.
 - When a model overfits, it performs exceptionally well on the training data, but its performance deteriorates when applied to new data. This can lead to poor generalization and inaccurate recommendations.
-

DETECT OVERFITTING



FOR CLARIFYING



EVALUATION THE MODEL

Metric	Category	Value	Interpretation	Performance
Test accuracy	Classification	97.07%	The model correctly predicts the target class with 97.07% accuracy.	Excellent
Precision	Classification	90.68%	Out of all the predicted positive instances, 90.68% are actually positive.	Good
Recall	Classification	93.50%	The model identifies 93.50% of the actual positive instances.	Good
F1 score	Classification	91.57%	The harmonic mean of precision and recall is 91.57%.	Good

EVALUATION THE MODEL

Metric	Category	Value	Interpretation	Performance
Accuracy (Avg. for all folds)	Classification	99.6807 (+- 0.0331)	On average, the model achieves an accuracy of 99.6807 with a small standard deviation of 0.0331.	Excellent
Loss	Error	0.0582	The average loss value (a measure of model error) is 0.0582.	Excellent
Hit Rate	Recommendation	99.00%	The model achieves a hit rate of 99.00%, indicating a high proportion of correctly recommended items.	Excellent
Mean Absolute Error (MAE)	Error	0.3167	The average absolute difference between the predicted and actual values is 0.3167.	Good

EVALUATION THE MODEL

Metric	Category	Value	Interpretation	Performance
Mean Average Precision (MAP)	Recommendation	0.0164	The average precision across all users is 0.0164.	Bad
Root Mean Squared Error (RMSE)	Error	3.9239	The square root of the average squared difference between the predicted and actual values is 3.9239.	Good
Novelty	Recommendation	97.43%	97.43% of the recommended items are previously unseen by the users.	Excellent
Diversity Score	Recommendation	0.91	The diversity score, measuring the variety of recommended items, is 0.91.	Excellent

4.3 Acceptance testing

4.3.1 Test plan

Scope:
All implemented use cases.
Test level:
Acceptance test (Black box)
Test types:
Functional test
Confirmation test
Regression test
Exit criteria:
All implemented test cases been executed.
Testing documents:
Test case
Bug report
Environment:
Windows 11
Chrome browser
Test references:
Use case description

4.3.2 Test cases

Test Case ID	TC 001	Test Case Description	Create a new account – sign up
Test result	Pass		
Related Requirement	Fr 001, Fr 005	Related use case	UC 001
Expected result	Signning up after fill e-mail, password, name, and mobile number & Sign in		
Actual result			
Prerequisites		Test Data	
1- Desktop Operating system		Name: Ahmed, Mobile number: 01005509148	
2- Web browser		E-mail: ahmed_elkhamissi@yahoo.com	
3-		Password: Fpor156*	
		Steps	
1- Open website			
2- Fill name, mobile number, e-mail, and password			
3- Click sign up			

Test Case ID	TC 002	Test Case Description	Sign up with registered e-mail
Test result	Pass		
Related Requirement	Fr 001, Fr 005	Related use case	UC 001
Expected result	Cannot sign up and show "E-mail is already registered" message.		
Actual result	Cannot sign up without show "E-mail is already registered" message.		
Prerequisites		Test Data	
1- Registered e-mail		Name: Ahmed, Mobile number: 01005509148	
2- Desktop Operating system		E-mail: ahmed_elkhamissi@yahoo.com	
3- Web browser		Password: Fpor156*	
		Steps	
1- Open website			
2- Fill name, mobile number, registerd e-mail, and password			
3- Click sign up			

Test Case ID	TC 003	Test Case Description	Sign up with invalid format e-mail
Test result	Pass		
Related Requirement	Fr 001, Fr 005	Related use case	UC 001
Expected result	Cannot sign up and show "Invalid e-mail" message.		
Actual result	Cannot sign up and show "Invalid e-mail" message.		
Prerequisites		Test Data	
1- Registered e-mail		Name: Ahmed, Mobile number: 01005509148	
2- Desktop Operating system		E-mail: ahmed_elkhamissi	
3- Web browser		Password: Fpor156*	
		Steps	
1- Open website			
2- Fill name, mobile number, invalid format e-mail, and password			
3- Click sign up			

Test Case ID	TC 004	Test Case Description	Sign up without e-mail		
Test result	Pass				
Related Requirement	Fr 001, Fr 005	Related use case	UC 001		
Expected result	Cannot sign up and show "E-mail is required" message.				
Actual result	Cannot sign up and show "E-mail is required" message.				
Prerequisites		Test Data			
1- Desktop Operating system		Name: Ahmed, Mobile number: 01005509148			
2- Web browser		E-mail:			
3-		Password: Fpor156*			
Steps					
1- Open website					
2- Fill name, mobile number, and password					
3- Click sign up					

Test Case ID	TC 005	Test Case Description	Sign up without name		
Test result	Pass				
Related Requirement	Fr 001, Fr 005	Related use case	UC 001		
Expected result	Cannot sign up and show "Name is required" message.				
Actual result	Cannot sign up without show "Name is required" message.				
Prerequisites		Test Data			
1- Desktop Operating system		Name: , Mobile number: 01005509148			
2- Web browser		E-mail: ahmed_elkhamissi@yahoo.com			
3-		Password: Fpor156*			
Steps					
1- Open website					
2- Fill mobile number, e-mail, and password					
3- Click sign up					

Test Case ID	TC 006	Test Case Description	Sign up without mobile number		
Test result	Pass				
Related Requirement	Fr 001, Fr 005	Related use case	UC 001		
Expected result	Cannot sign up and show "Mobile number is required" message.				
Actual result	Cannot sign up and show "Mobile number is required" message.				
Prerequisites		Test Data			
1- Desktop Operating system		Name: Ahmed, Mobile number:			
2- Web browser		E-mail: ahmed_elkhamissi@yahoo.com			
3-		Password: Fpor156*			
Steps					
1- Open website					
2- Fill name, e-mail, and password					
3- Click sign up					

Test Case ID	TC 007	Test Case Description	Sign up with invalid mobile number		
Test result	Pass				
Related Requirement	Fr 001, Fr 005	Related use case	UC 001		
Expected result	Cannot sign up and show "Min length 11 digits" message.				
Actual result	Cannot sign up and show "Min length 11 digits" message.				
Prerequisites		Test Data			
1- Desktop Operating system		Name: Ahmed, Mobile number: 5509148			
2- Web browser		E-mail: ahmed_elkhamissi@yahoo.com			
3-		Password: Fpor156*			
Steps					
1- Open website					
2- Fill name, invalid format mobile number, e-mail, and password					
3- Click sign up					

Test Case ID	TC 008	Test Case Description	Sign up with invalid password		
Test result	Postponed				
Related Requirement	Fr 001, Fr 005	Related use case	UC 001		
Expected result	Cannot sign up and show "Password is invalid" message.				
Actual result					
Prerequisites		Test Data			
1- Desktop Operating system		Name: Ahmed, Mobile number: 01005509148			
2- Web browser		E-mail: ahmed_elkhamissi@yahoo.com			
3-		Password: 1234			
Steps					
1- Open website					
2- Fill name, mobile number, e-mail, and invalid format password					
3- Click sign up					

Test Case ID	TC 008	Test Case Description	Sign up without password		
Test result	Pass				
Related Requirement	Fr 001, Fr 005	Related use case	UC 001		
Expected result	Cannot sign up and show "Password is required" message.				
Actual result	Cannot sign up and show "Password is required" message.				
Prerequisites		Test Data			
1- Desktop Operating system		Name: Ahmed, Mobile number: 01005509148			
2- Web browser		E-mail: ahmed_elkhamissi@yahoo.com			
3-		Password:			
Steps					
1- Open website					
2- Fill name, mobile number, and e-mail					
3- Click sign up					

Test Case ID	TC 009	Test Case Description	Sign in
Test result	Pass		
Related Requirement	Fr 002	Related use case	UC 002
Expected result	Signning in after fill e-mail / username and password.		
Actual result			
Prerequisites		Test Data	
1- Registered account		username: Ahmed	
2- Desktop Operating system		Password: Fpor156*	
3- Web browser			
Steps			
1- Open website			
2- Fill username and password			
3- Click sign In			

Test Case ID	TC 010	Test Case Description	Sign in without unregistered username
Test result	Pass		
Related Requirement	Fr 002	Related use case	UC 002
Expected result	Cannot sign in and show failure message.		
Actual result	Cannot sign in and show failure message.		
Prerequisites		Test Data	
1- Unregistered account		username: unreguser	
2- Desktop Operating system		Password: 12345Un#	
3- Web browser			
Steps			
1- Open website			
2- Fill username and password			
3- Click sign In			

Test Case ID	TC 011	Test Case Description	Sign in without username
Test result	Pass		
Related Requirement	Fr 002	Related use case	UC 002
Expected result	Cannot sign in and show "username is required".		
Actual result	Cannot sign in without show "username is required".		
Prerequisites		Test Data	
1- Desktop Operating system		Username:	
2- Web browser		Password: Fpor156*	
3-			
Steps			
1- Open website			
2- Fill password			
3- Click sign in			

Test Case ID	TC 012	Test Case Description	Sign in with wrong password		
Test result	Fail				
Related Requirement	Fr 002	Related use case	UC 002		
Expected result	Cannot sign in and show "Password not exist".				
Actual result	Signed in				
Prerequisites		Test Data			
1- Registered account		Username: Ahmed			
2- Desktop Operating system		Password: Fpor156			
3- Web browser					
Steps					
1- Open website					
2- Fill username and password					
3- Click sign In					

Test Case ID	CTC 001	Test Case Description	Sign in with wrong password		
Test result	Pass	Related Test Case	TC 012		
Related Requirement	Fr 002	Related use case	UC 002		
Expected result	Cannot sign in and show "Password not exist".				
Actual result	Signed in				
Prerequisites		Test Data			
1- Registered account		Username: Ahmed			
2- Desktop Operating system		Password: Fpor156			
3- Web browser					
Steps					
1- Open website					
2- Fill username and password					
3- Click sign In					

Test Case ID	TC 013	Test Case Description	Sign in without password		
Test result	Pass				
Related Requirement	Fr 002	Related use case	UC 002		
Expected result	Cannot sign in and show "Password is required".				
Actual result	Cannot sign in without show "Password is required".				
Prerequisites		Test Data			
1- Registered account		Username: Ahmed			
2- Desktop Operating system		Password:			
3- Web browser					
Steps					
1- Open website					
2- Fill username					
3- Click sign In					

Test Case ID	TC 012	Test Case Description	Edit tools
Test result	Postponed		
Related Requirement	Fr 006	Related use case	UC 003
Expected result	Edit tools and show "Tools edited" message.		
Actual result			
Prerequisites		Test Data	
1- Registered account		E-mail: ahmed_elkhamissi@yahoo.com	
2- Desktop Operating system		Password: Fpor156*	
3- Web browser			
Steps			
1- Open website			
2- Fill e-mail and password			
3- Click sign In			
4- Go to tools page			
5- Click edit tool			
6- Edit tool and click submit			

Test Case ID	TC 013	Test Case Description	Search for projects
Test result	Postponed		
Related Requirement	Fr 003, Fr 004	Related use case	UC 004
Expected result	Show related results to search entire.		
Actual result			
Prerequisites		Test Data	
1- Desktop Operating system		Chatbot	
2- Web browser			
3-			
Steps			
1- Open website			
2- Type "Chatbot" in search bar			
3- Click search			

Test Case ID	TC 014	Test Case Description	Search for projects with unmatched entire
Test result	Postponed		
Related Requirement	Fr 003, Fr 004	Related use case	UC 004
Expected result	No results shown to search entire and show "We are sorry, there is no projects relate to your search title". and "TRY AGAIN" and "CANCEL" buttons.		
Actual result			
Prerequisites		Test Data	
1- Desktop Operating system		Satellite	
2- Web browser			
3-			
Steps			
1- Open website			
2- Type "Chatbot" in search bar			
3- Click search			

Test Case ID	TC 015	Test Case Description	Re-search
Test result	Postponed		
Related Requirement	Fr 003, Fr 004	Related use case	UC 004
Expected result	Back to search bar to rewrite a new search entire.		
Actual result			
Prerequisites		Test Data	
1- Desktop Operating system		Satellite	
2- Web browser			
3-			
Steps			
1- Open website			
2- Type "Chatbot" in search bar			
3- Click search			
4- Click try again			

Test Case ID	TC 016	Test Case Description	Recommend projects
Test result	Pass		
Related Requirement	Fr 007, Fr 020	Related use case	UC 005
Expected result	Show 5 projects, that recommended depend on preferences.		
Actual result	Show 5 projects, that recommended depend on preferences.		
Prerequisites		Test Data	
1- Registered account		Username: Ahmed	
2- Desktop Operating system		Password: Fpor156*	
3- Web browser		Skills: HTML, CSS, JavaScript, PHP, Python, MySQL	
Steps			
1- Open website			
2- Sign In with registeres username and password			
3- Select skills button			
4- Fill HTML, CSS, JavaScript, PHP, Python, MySQL in tools page			
5- Click Recommend projects			

Test Case ID	TC 017	Test Case Description	Recommend projects
Test result	Fail		
Related Requirement	Fr 007, Fr 020	Related use case	UC 005
Expected result	Show projects, that recommended depend on preferences.		
Actual result	Show projects, that didn't fit preferences.		
Prerequisites		Test Data	
1- Registered account		Username: Ahmed	
2- Desktop Operating system		Password: Fpor156*	
3- Web browser		Skills: React Native, Flutter, Angular, Vue.js, Node.js	
Steps			
1- Open website			
2- Sign In with registeres username and password			
3- Select skills button			
4- Fill React Native, Flutter, Angular, Vue.js, Node.js			
5- Click Recommend projects			

Test Case ID	CTC 002	Test Case Description	Recommend projects		
Test result	Pass	Related Test Case	TC 017		
Related Requirement	Fr 007, Fr 020	Related use case	UC 005		
Expected result	Show projects, that recommended depend on preferences.				
Actual result	Show projects, that recommended depend on preferences.				
Prerequisites		Test Data			
1- Registered account		Username: Ahmed			
2- Desktop Operating system		Password: Fpor156*			
3- Web browser		Skills: React Native, Flutter, Angular, Vue.js, Node.js			
Steps					
1- Open website					
2- Sign In with registeres username and password					
3- Select skills button					
4- Fill React Native, Flutter, Angular, Vue.js, Node.js					
5- Click Recommend projects					

Test Case ID	TC 018	Test Case Description	Recommend projects		
Test result	Fail				
Related Requirement	Fr 007, Fr 020	Related use case	UC 005		
Expected result	Show projects, that recommended depend on preferences.				
Actual result	Show projects, that didn't fit preferences.				
Prerequisites		Test Data			
1- Registered account		Username: Ahmed			
2- Desktop Operating system		Password: Fpor156*			
3- Web browser		Python, Natural Language Processing, Machine Learning, Deep Learning, Cloud Computing			
Steps					
1- Open website					
2- Sign In with registeres e-mail and password					
3- Fill Python, Natural Language Processing, Machine Learning, Deep Learning, Cloud Computing in tools page					
4- Click Recommend projects					

Test Case ID	CTC 003	Test Case Description	Recommend projects		
Test result	Pass	Related Test Case	TC 018		
Related Requirement	Fr 007, Fr 020	Related use case	UC 005		
Expected result	Show projects, that recommended depend on preferences.				
Actual result	Show projects, that recommended depend on preferences.				
Prerequisites		Test Data			
1- Registered account		Username: Ahmed			
2- Desktop Operating system		Password: Fpor156*			
3- Web browser		Python, Natural Language Processing, Machine Learning,			
Steps					
1- Open website					
2- Sign In with registeres e-mail and password					
3- Fill Python, Natural Language Processing, Machine Learning, Deep Learning, Cloud Computing in tools					
4- Click Recommend projects					

Test Case ID	TC 019	Test Case Description	Recommend projects with unlisted tool		
Test result	Fail				
Related Requirement	Fr 007, Fr 020	Related use case	UC 005		
Expected result	shows a message "There are no projects matching your preferences, but you can join a team with your tools."				
Actual result					
Prerequisites		Test Data			
1- Registered account		Username: Ahmed			
2- Desktop Operating system		Password: Fpor156*			
3- Web browser		C#			
Steps					
1- Open website					
2- Sign In with registeres e-mail and password					
3- Fill C# in tools page					
4- Click Recommend projects					

Test Case ID	TC 020	Test Case Description	Accept project		
Test result	Postponed				
Related Requirement	Fr 008, 012, 013, 028, 030	Related use case	UC 006, UC 009		
Expected result	Redirect to "DASHBOARD" to show the project's details and Tasks.				
Actual result					
Prerequisites		Test Data			
1- Registered account		E-mail: ahmed_elkhamissi@yahoo.com			
2- Desktop Operating system		Password: Fpor156*			
3- Web browser		HTML, CSS, JavaScript, PHP, Python, MySQL			
Steps					
1- Open website					
2- Sign In with registeres e-mail and password					
3- Fill HTML, CSS, JavaScript, PHP, MySQL in tools page					
4- Click Recommend projects					
5- Click "Accept project"					

Test Case ID	TC 021	Test Case Description	Create team		
Test result	Pass				
Related Requirement	Fr 009, Fr 010, Fr 011, Fr 014	Related use case	UC 007		
Expected result	Create team with generated id, and show "Congratulation, you have created a team successfully" message				
Actual result					
Prerequisites		Test Data			
1- Registered account		E-mail: ahmed_elkhamissi@yahoo.com			
2- Desktop Operating system		Password: Fpor156*			
3- Web browser					
Steps					
1- Open website					
2- Sign In with registeres e-mail and password					
3- Click "CREATE TEAM"					
4- Fill team name and choose team tag					
5- Click create					

Test Case ID	TC 022	Test Case Description	Add registered users to the team		
Test result	Postponed				
Related Requirement	Fr 015	Related use case	UC 008		
Expected result	Add user and show "We have sent adding request to the user e-mail" message.				
Actual result					
Prerequisites		Test Data			
1- Registered account		E-mail: ahmed_elkhamissi@yahoo.com			
2- Desktop Operating system		Password: Fpor156*			
3- Web browser		Registered e-mail: any temp e-mail			
4- Created team					
Steps					
1- Open website					
2- Sign In with registeres e-mail and password					
3- Click "Team"					
4- Click "ADD TEAM MEMBER"					
5- Type a registered user e-mail					
6- Click "ADD USER"					

Test Case ID	TC 023	Test Case Description	Add unregistered users to the team		
Test result	Postponed				
Related Requirement	Fr 015	Related use case	UC 008		
Expected result	Cannot add user and show "The e-mail you have entered is not exist, try another e-mail." message.				
Actual result					
Prerequisites		Test Data			
1- Registered account		E-mail: ahmed_elkhamissi@yahoo.com			
2- Desktop Operating system		Password: Fpor156*			
3- Web browser		Unregistered e-mail: any temp e-mail			
Steps					
1- Open website					
2- Sign In with registeres e-mail and password					
3- Click "Team"					
4- Click "ADD TEAM MEMBER"					
5- Type an unregistered user e-mail					
6- Click "ADD USER"					

Test Case ID	TC 024	Test Case Description	Recommend learning resources
Test result	Pass		
Related Requirement	Fr 017, Fr 018, Fr 019	Related use case	UC 010
Expected result	Show learning resources for the project's missed tool.		
Actual result			
Prerequisites		Test Data	
1- Registered account		E-mail: ahmed_elkhamissi@yahoo.com	
2- Desktop Operating system		Password: Fpor156*	
3- Web browser			
3- Internet connection			
Steps			
1- Open website			
2- Sign In with registeres e-mail and password			
3- Click "Sources".			
4- Fill skill name.			
5- Click submit.			

Test Case ID	TC 025	Test Case Description	Recommend new member
Test result	Postponed		
Related Requirement	Fr 020	Related use case	UC 010
Expected result	Recommend new member can work with the project's missed tool.		
Actual result			
Prerequisites		Test Data	
1- Registered account		E-mail: ahmed_elkhamissi@yahoo.com	
2- Desktop Operating system		Password: Fpor156*	
3- Web browser			
4- Team created			
5- Project choosen			
Steps			
1- Open website			
2- Sign In with registeres e-mail and password			
3- Click "Projects" to redirect to dashboard.			
4- Click "NEW MEMBER"			
5- Click "SEND JOIN REQUEST"			

Test Case ID	TC 026	Test Case Description	Test the user in a learned tool
Test result	Postponed		
Related Requirement	Fr 021	Related use case	UC 011
Expected result	Show the exam and answer it.		
Actual result			
Prerequisites		Test Data	
1- Registered account		E-mail: ahmed_elkhamissi@yahoo.com	
2- Desktop Operating system		Password: Fpor156*	
3- Web browser			
4- Team created			
5- Learned user			
Steps			
1- Open website			
2- Sign In with registeres e-mail and password			
3- Click "Projects" to redirect to dashboard.			
4- Click "FINISH COURSE"			
5- Answer the test question and submit result			

Test Case ID	TC 027	Test Case Description	Deactivate the project missed tool's tasks
Test result	Postponed		
Related Requirement	Fr 022	Related use case	UC 012
Expected result	Show the project missed tool's tasks in deactivated view		
Actual result			
Prerequisites		Test Data	
1- Registered account		E-mail: ahmed_elkhamissi@yahoo.com	
2- Desktop Operating system		Password: Fpor156*	
3- Web browser			
4- Project choosen			
Steps			
1- Open website			
2- Sign In with registeres e-mail and password			
3- Click "Projects" to redirect to dashboard.			

Test Case ID	TC 028	Test Case Description	Determine start / end date of the project
Test result	Postponed		
Related Requirement	Fr 023	Related use case	UC 013
Expected result	Start / end dates shown in Dashboard		
Actual result			
Prerequisites		Test Data	
1- Registered account		E-mail: ahmed_elkhamissi@yahoo.com	
2- Desktop Operating system		Password: Fpor156*	
3- Web browser			
4- Project choosen			
Steps			
1- Open website			
2- Sign In with registeres e-mail and password			
3- Click "Projects" to redirect to dashboard.			
4- Click "SET PROJECT DATES"			
5- Set start / end dates			
6- Click "SET"			

Test Case ID	TC 029	Test Case Description	Delay start / end date of the project
Test result	Postponed		
Related Requirement	Fr 024	Related use case	UC 014
Expected result	Start / end dates shown in Dashboard as edited		
Actual result			
Prerequisites		Test Data	
1- Registered account		E-mail:	ahmed_elkhamissi@yahoo.com
2- Desktop Operating system		Password:	Fpor156*
3- Web browser			
4- Project choosen			
Steps			
1- Open website			
2- Sign In with registeres e-mail and password			
3- Click "Projects" to redirect to dashboard.			
4- Click "EDIT PROJECT DATES"			
5- Edit start / end dates			
6- Click "SET"			

Test Case ID	TC 030	Test Case Description	Remove a member from team.
Test result	Postponed		
Related Requirement	Fr 025	Related use case	UC 015
Expected result	The removed member is no longer shown in dashboard		
Actual result			
Prerequisites		Test Data	
1- Registered account		E-mail:	ahmed_elkhamissi@yahoo.com
2- Desktop Operating system		Password:	Fpor156*
3- Web browser			
4- Team created			
Steps			
1- Open website			
2- Sign In with registeres e-mail and password			
3- Click "Projects" to redirect to dashboard.			
4- Click "REMOVE MEMBER"			
5- Click "X"			
6- Click "REMOVE"			

Test Case ID	TC 031	Test Case Description	Deactivate a chosen project.
Test result	Postponed		
Related Requirement	Fr 026	Related use case	UC 016
Expected result	Dashbaord are shown in deactivated view		
Actual result			
Prerequisites		Test Data	
1- Registered account		E-mail:	ahmed_elkhamissi@yahoo.com
2- Desktop Operating system		Password:	Fpor156*
3- Web browser			
4- Project choosen			
Steps			
1- Open website			
2- Sign In with registeres e-mail and password			
3- Click "Projects" to redirect to dashboard.			
4- Click "DEAVTIVATE PROJECT"			
5- Click "DEACTIVATE"			

Test Case ID	TC 032	Test Case Description	Disjoin a team.
Test result	Postponed		
Related Requirement	Fr 027, Fr 032, Fr 033	Related use case	UC 017, UC 019
Expected result	The left member cannot access the team page and the dashboarded, in case of the team leader was the lefeted member, dashboared will be deactivated.		
Actual result			
Prerequisites		Test Data	
1- Registered account		E-mail:	ahmed_elkhamissi@yahoo.com
2- Desktop Operating system		Password:	Fpor156*
3- Web browser			
4-Team created			
Steps			
1- Open website			
2- Sign In with registeres e-mail and password			
3- Click "MY TEAM".			
4- Click "DISJOIN TEAM"			
5- Click "DISJOIN"			

Test Case ID	TC 033	Test Case Description	Recommend a new member for a team is lifted by another user
Test result	Postponed		
Related Requirement	Fr 029	Related use case	UC 018
Expected result	Recommend a new team member, able to work by a specific technology/tool at the project's predetermined dates, when a team member disjoin the team and send adding request to the new member.		
Actual result			
Prerequisites		Test Data	
1- Registered account		E-mail: ahmed_elkhamissi@yahoo.com	
2- Desktop Operating system		Password: Fpor156*	
3- Web browser			
4- Project chosen			
Steps			
1- Open website			
2- Sign In with registeres e-mail and password			
3- Click "Projects" to redirect to dashboard.			
4- Click "RECOMMEND" in "Member name has left the team, Do you want to recommend a new team member" message.			
5- Click "SEND JOIN REQUEST"			

Test Case ID	TC 032	Test Case Description	Request acceptance
Test result	Postponed		
Related Requirement		Related use case	UC 008, UC 018
Expected result	User us shown in the team list		
Actual result			
Prerequisites		Test Data	
1- Registered account		E-mail: ahmed_elkhamissi@yahoo.com	
2- Desktop Operating system		Password: Fpor156*	
3- Web browser		Any regidtered temp e-mail	
4- Team created			
Steps			
1- Open website			
2- Sign In with registeres e-mail and password			
3- Click "Notification" to show the requests			
4- Click specific request			
5- Click "APPROVE"			

Test Case ID	TC 033	Test Case Description	Send messages through Chatbot
Test result	Postponed		
Related Requirement	Fr 034, Fr 035, Fr 036	Related use case	UC 020
Expected result	Sent message will be shown in the Chatbot.		
Actual result			
Prerequisites		Test Data	
1- Registered account		E-mail: ahmed_elkhamissi@yahoo.com	
2- Desktop Operating system		Password: Fpor156*	
3- Web browser		Any regidtered temp e-mail	
4- Project chosen			
5- Team created			
Steps			
1- Open website			
2- Sign In with registeres e-mail and password			
3- Click "Projects" to redirect to dashboard.			
4- Click "CHATBOT"			
5- Type a text message			
6- Click "SEND"			

4.3.2 Bug report

Test Case ID	TC 012	Test Case Description	Sign in with wrong password		
Test result	Fail	Bug ID	Bg 001		
Related Requirement	Fr 002	Related use case	UC 002		
Expected result	Cannot sign in and show "Password not exist".				
Actual result	Signed in				
Status	Open				
Environment		Test Data			
1- Windows 11		Username: Ahmed			
2- Chrome browser		Password: 1			
Steps					
1- Open website					
2- Fill username and password					
3- Click sign In					

Test Case ID	TC 017	Test Case Description	Recommend projects		
Test result	Fail	Bug ID	Bg 002		
Related Requirement	Fr 007, Fr 020	Related use case	UC 005		
Expected result	Show projects, that recommended depend on preferences.				
Actual result	Show projects, that didn't fit preferences.				
Status	Closed				
Environment		Test Data			
1- Windows 11		Username: Ahmed			
2- Chrome browser		Password: Fpor156*			
3-		Skills: React Native, Flutter, Angular, Vue.js, Node.js			
Steps					
1- Open website					
2- Sign In with registeres username and password					
3- Select skills button					
4- Fill React Native, Flutter, Angular, Vue.js, Node.js					
5- Click Recommend projects					

Test Case ID	TC 018	Test Case Description	Recommend projects		
Test result	Fail	Bug ID	Bg 003		
Related Requirement	Fr 007, Fr 020	Related use case	UC 005		
Expected result	Show projects, that recommended depend on preferences.				
Actual result	Show projects, that didn't fit preferences.				
Status	Closed				
Environment		Test Data			
1- Windows 11		Username: Ahmed			
2- Chrome browser		Password: Fpor156*			
3-		D3. js , Plotly, Pandas			
Steps					
1- Open website					
2- Sign In with registeres e-mail and password					
3- Fill Python, Natural Language Processing, Machine Learning, Deep Learning, Cloud Computing in tools page					
4- Click Recommend projects					

Chapter 5: Conclusion and Future Work

5.1 Conclusion

The ultimate goal of information technology is to facilitate human live and enable them to perform their tasks better, and since the destiny of software development specialists is to solve problems permanently, we find that helping final-year students of computer science to find a graduation project suitable for their abilities makes them more rational in taking their decision, so the right decision starts from defining the problem and identifying its constraints.

Our system covers most software complexity levels to suit different levels of students that want to get a project idea.

We worked in Our system not just provide an idea, but also to support value like teamwork and learning and work, believing that these values increase the add values of graduates' skills, according to the labor market requirements.

We deliberately developed this project in a way that makes the user experience very easy, All the user has to do is create a new account and log in to the website and enter his/her mastered skills, and then the website will recommends a number of projects for him/her, and if the user is not able to work with a specific skill, the website will support him/her with free learning resources to learn more than 122 tools that contribute to software development.

5.2 Future work

Although perfection is an incomprehensible goal, and although we are satisfied with what we have done in this project, we intend to make modifications and improvements to the current version of the project, including the following:

Implement a function that helps the student by explain a task to the student in case he finds it difficult to implement it.

Implement a function that evaluating the student's work on a specific task.

Implement a function that tests the extent to which the student retrieves what he learned from one of the courses recommended by the system to learn a specific skill.

Implement tools that will be used in software documentation, systems analysis, and drawing UML diagrams.

Linking the project to Stack Overflow surveys to update its data source by making it always added to it the latest tools and the latest projects.

References

Books

Ian Sommerville, “SOFTWARE ENGINEERING”, Ninth Edition, 2010.

Ramez elmasry, Shamkant B. Navathe, "Fundamentals of Database Systems",7th edition, Pearson, 2016.

Certified Tester Foundation Level (CTFL) Syllabus, Version 2018 v3.1.1 , 2021

Paper

Oleksii Kuchaiev, Boris Ginsburg, “Training Deep AutoEncoders for Collaborative Filtering”, NVIDIA, <https://arxiv.org/pdf/1708.01715v3.pdf>

URL

<https://www.techtarget.com/searchsoftwarequality/definition/systems-development-life-cycle>

<https://www.ibm.com/docs/fr/bps/8.10.1?topic=development-methodology>

<https://www.agilealliance.org/agile101/>

<https://scrumguides.org/scrum-guide.html#the-sprint>

<https://www.scrumalliance.org/about-scrum#!section5>

<https://www.lucidchart.com/blog/types-of-UML-diagrams>

<https://archive.org/download/stackexchange>

<https://bigquery.cloud.google.com/dataset/bigquery-public-data:stackoverflow>

<https://bigquery.cloud.google.com/dataset/bigquery-public-data:stackoverflow>