CODEE

Mamdoh M. Zeyad and Ahmed M. Sabbagh

Department of Software Engineering, University of Jeddah

CCSW 429: Senior Project II

Dr. Ahmed Alghamdi

April 19, 2024

Abstract

In recent days, technology has evolved at an increasing pace. Therefore, companies must stay updated with the latest advancements to remain competitive. This requires a strong focus on programming, which has seen an increase in demand. Several websites have emerged worldwide to assist programmers in resolving their coding challenges. Additionally, these platforms serve as a bridge for non-programmers to connect with developers for development, providing an opportunity for developers to earn extra income.

In the current situation, many users have difficulties challenges when expressing their programming problems and identifying a suitable developer to solve them. Moreover, developers seek out the necessary development environment that helps them develop some apps without the need for external resources or tools.

We are dedicated to meeting the programming community's needs by providing essential tools, including an integrated development environment (IDE) and an AI-powered chatbot. We are also implementing a direct chat feature for developers to engage with users and help them resolve their programming challenges efficiently. Furthermore, we will offer consultancy services to connect users with the right developer for their specific needs.

This project report will contain seven main chapters in the first chapter we will cover the pre-production stage thesis consisting of the problem definition, project aims and objectives, project methodology, and project plan, while the second chapter contains the stakeholder's definition, project domain, data loggers, literature review, results, and comparison results and the feasibility study. The third and fourth chapters will describe the requirements, diagrams, architecture of the system, and GUI designs. The fifth chapter will cover the implementation stage consisting of APIs and Plunges, Data Collection Instruments, and the implementation tools used. The sixth chapter will cover the testing stage consisting of testing approaches, data collection, and test cases. The seventh chapter indicates the conclusions and the future work of our project.

Table of Contents

LIST OF FIGURES..... 6

LIST OF TABLES..... 8

CHAPTER 1: INTRODUCTION..... 10

1.1. Introduction: 11

1.2. The Problem Definition 12

1.3. Project Aims and Objectives 13

1.4. Project Methodology 13

1.5. Project Plan 14

1.6. Conclusion 16

CHAPTER 2: LITERATURE REVIEW..... 17

2.1. Introduction 18

2.2. Stakeholders Definition..... 18

2.3. Project Domain 21

2.4. Data Collection 22

2.5. Literature Review 23

2.5.1. Comparison Criteria Definition 26

2.5.2. Comparison Results and the Feasibility Study 27

2.6. Conclusion 28

CHAPTER 3: ANALYSIS PHASE 29

3.1. Introduction 30

3.2. Requirements Gathering 30

3.2.1. Functional Requirements 31

3.2.2. Non-Functional Requirements 33

3.2.3. Hardware Requirements 33

3.3. Actors Definition 33

3.4. UML Diagrams 34

3.4.1. Activity Diagrams 38

3.4.2. Use Case Diagrams 40

3.4.2.1. Use Case Diagrams Specifications 48

3.4.3. Sequence Diagrams 48

3.4.4. Class Diagram 51

3.5. Conclusion 51

CHAPTER 4: DESIGN PHASE 52

4.1. Introduction 53

4.2. System Architecture 53

4.2.1. Architecture Pattern 53

4.3. Deployment Diagram 54

4.4. User Interfaces Design 54

4.5 Conclusion 60

CHAPTER 5: DEVELOPMENT PHASE 61

5.1. Introduction 62

5.2. APIs and Plugins 62

5.3. Data Collection Instruments 63

5.4. Implementation 64

5.4.1. Building the Database 64

5.4.2. Building the System 66

5.4.3. Samples of the Code 66

5.5 Conclusion 68

CHAPTER 6: TESTING PHASE 69

6.1. Introduction 70

6.2. Testing Approach 70

6.3. Data collection (Quantitative, Qualitative) 73

6.4. Test Cases and Result 74

6.4.1. Home Page Testing 74

6.4.2. Signup Page Testing 75

6.4.3. Login Page Testing 76

6.4.4. Reset Password Page Testing 77

6.4.5. Catalog Page Testing 77

6.4.6. Development Area Page Testing 78

6.4.7. Users-Chat Page Testing 78

6.4.8. Chat-Chat Page Testing 79

6.4.9. Verification Page Testing 79

6.4.10. Admin Page Testing 80

6.5 Conclusion 81

CHAPTER 7: CONCLUSIONS 82

7.1. Conclusion 83

7.2. Future Work 83

REFERENCES 84

Table of Figures

Figure 1.1: Waterfall vs Agile 14

Figure 1.2: Project Plan (CH1 & CH2) 15

Figure 1.3: Project Plan (CH3 & CH4) 15

Figure 1.4: Project Plan, Timeline 16

Figure 1.5: Project Plan, Timeline Cont. 16

Figure 2.1: Stakeholders' influence and importance mapping 21

Figure 2.2: StackOverflow 24

Figure 2.3: Khmsat 24

Figure 2.4: Freelancer 25

Figure 2.5: Gitter 26

Figure 3.1: The user (consumer) activity diagram 35

Figure 3.2: The developer (provider) activity diagram. 36

Figure 3.3: The consultant activity diagram 37

Figure 3.4: The admin activity diagram 38

Figure 3.5: Users and consultants use case diagram. 39

Figure 3.6: Developers and admins diagram 40

Figure 3.7: Login sequence diagram 50

Figure 3.8: Chat system sequence diagram 50

Figure 3.9: IDE and AI-Talking bot sequence diagram 51

Figure 3.10: User Management sequence diagram 51

Figure 3.11: Class diagram 52

Figure 4.1: layered pattern 54

Figure 4.2: Deployment diagram 55

Figure 4.3: Home page 56

Figure 4.4: Register page 57

Figure 4.5: Login page 57

Figure 4.6: User page (catalogue) 58

Figure 4.7: Chat page 58

Figure 4.8: Developer page (catalogue) 59

Figure 4.9: Development page (IDE & AI bot). 59

Figure 4.10: Verification page (for developers & consultants) 60

Figure 4.11: Admin page 61

Figure 5.1: All tables 66

Figure 5.2: Users table 66

Figure 5.3: developemnt_skills table 66

Figure 5.4: Messages table 67

Figure 6.1: CFG for approve function 71

Figure 6.2: CFG for insert function 72

Table of Tables

Table 2.1: Stakeholders definition 18

Table 2.2: Comparison criteria definition 27

Table 3.1: Functional Requirements 30

Table 3.2: Non-Functional Requirements 32

Table 3.3: Hardware Requirements 33

Table 3.4: Actors Definition 33

Table 3.5: Use-Case.1 40

Table 3.6: Use-Case.2 40

Table 3.7: Use-Case.3 41

Table 3.8: Use-Case.4 41

Table 3.9: Use-Case.5 42

Table 3.10: Use-Case.6 43

Table 3.11: Use-Case.7 43

Table 3.12: Use-Case.8 44

Table 3.13: Use-Case.9 44

Table 3.14: Use-Case.10 45

Table 3.15 Use-Case.11 45

Table 3.16: Use-Case.12 46

Table 3.17: Use-Case.13 47

Table 3.18: Use-Case.14 48

Table 6.1: Home page test table 74

Table 6.2: Signup page test table 75

Table 6.3: Login page test table 76

Table 6.4: Reset password page test table 77

Table 6.5: Catalog page test table..... 77

Table 6.6: Development area page test table..... 78

Table 6.7: Users-chat page test table 78

Table 6.8: Chat-chat page test table 79

Table 6.9: Verification page test table 79

Table 6.10: Admin page test table 80

CHAPTER1: INTRODUCTION

1.1. Introduction

In recent days, technology has evolved at an increasing pace. Therefore, companies must keep pace with this development and make and work on competitive technical systems in the market. Correspondingly, programming constitutes an essential factor of these modern technologies, as their demand has increased among people or companies.

Regarding that, many websites distributed around the world help programmers solve their programming issues. Moreover, these websites give users with no programming knowledge to find developers to build their applications, which allows the developers to make additional income from these websites.

One of the most commonly used websites is “StackOverflow”, a public platform built to define programming issues and find other developers to solve them. This process may take some days to find the right solution. In addition, there is another known website called “Freelancer”, this website helps users to choose developers to solve their problems. Moreover, users can create a project allowing the developers to join it and try to build it.

Our site CODEE allows programmers and users a comfortable and unique experience in terms of using programming websites, as it gives an integrated suitable environment for programmers so that it can work all the requirements needed within the site. In addition, the CODEE website provides users with unique services, for example, a consultant who will guide the user to the appropriate programmer who will help him with his programming issue.

This chapter was divided into several sections including the definition of the problem, discussing the project objectives and aims. Moreover, we defined the methodology followed in this project. Lastly, we construct the project plan.

1.2. The Problem Definition

In the current situation, many users face difficulties in expressing their programming issues and finding a suitable developer to solve them. Moreover, developers need to find the required development environment that helps them develop some apps without the need for external resources or tools.

User's difficulties can be divided into two stages:

1- **Finding a suitable developer to solve their programming issues:**

   One of the most significant and prominent difficulties that the user was facing was determining the type of issues he is trying to solve and the definition of a suitable developer to solve their issues. Moreover, we developed a very suitable solution by adding a consultant so that he could guide the user to the appropriate developer so that he could solve his issues.

2- **After finding a suitable developer, how to communicate effectively:**

   Once we have found a suitable developer, we will prepare a chat that allows the user to make a conversation with the developer directly and effectively.

Developer's difficulties can be divided into three stages:

1- **Finding a suitable program for his skills and he can solve it easily:**

   Instead of trying to solve programming issues not suitable for the developer's skills and wasting the developer's time on them, consultants will guide every programming issue to a developer who has the required skills to solve it.

2- **Finding a development environment to help him in development:**

   Programmers are always looking for new development environments to code easily with the help of new technologies like AI-talking bots. Thus, all the similar websites do not

provide these features. Accordingly, they need to download external tools to use them for small programming issues that do not take much time or resources.

3- **Enable him to chat directly with users:**

Developers using similar websites may need to wait for emails to communicate with users and solve their issues. Alternatively, they can chat directly on our website and share the required files if needed.

1.3. Project Aims and Objectives

Through this project, we aim for among the most common sites, not only in the Middle East but also on a global scale, we aim to serve the programmers' needs by furnishing them with all the essential tools throughout the project, including the website's integrated development environment (IDE) and an AI-talking bot. In addition, we will implement a chat that enables a developer to chat directly with users on the site. Therefore, we will design a chat system to facilitate effective discussions with users to solve their programming issues. We are also integrating consultant services to provide assistance and guide users toward the most appropriate developer for their specific issues.

1.4. Project Methodology

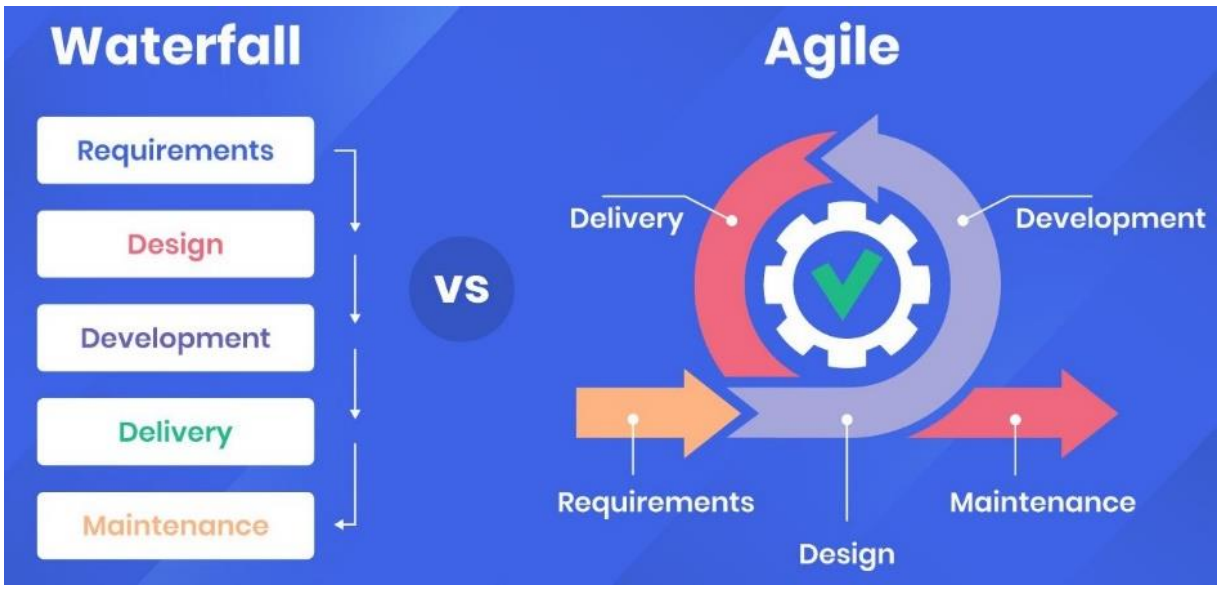
Utilizing a methodology in projects is critical for effective organization and planning. Additionally, various methodologies are available, with the prominent ones being the Waterfall model and the Agile methodology.

Accordingly, in implementing the CODEE project we will use hybrid methodologies. Firstly, we will use the Waterfall methodology, because this methodology provides a clear structure of the project and early determination of goals. Secondly, in the implementation phase, we will choose Agile - Extreme Programming (XP) because we will be programming as a pair. Therefore, Agile methodology will give us flexibility in programming.

- Figure 1.1: This figure shows the Waterfall model and Agile model.

Figure 1.1

Waterfall vs Agile.



1.5. Project Plan

Planning is an essential phase for the success of any project. Planning is the responsibility of the project manager to create a helpful plan for the team and guide them. One of the most commonly used project planning tools is the MS Project tool. This tool will help you create your plan and manage the resources wisely.

Therefore, we used MS Project as a planning tool. The figures below show the project plan that we created using MS Project:

- Figure 1.2: This figure shows the tasks for the initiating phase, planning phase, and problem-understanding phase.
- Figure 1.3: This figure shows the tasks for the analysis phase and design phase.
- Figure 1.4: This figure shows the timeline for initiating phase, planning phase, and problem-understanding phase.
- Figure 1.5: This figure shows the timeline analysis phase and design phase.

Figure 1.2

Project Plan (CH1 & CH2).

GANTT CHART

	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1		**CODEE PROJECT**	**73 days**	**Tue 8/29/23**	**Thu 12/7/23**		**Ahmed Sabbagh,Dr.Ahmed Alghamdi,Mamdoh Zeyad**
2		**Initiating**	**8 days**	**Tue 8/29/23**	**Thu 9/7/23**		**Ahmed Sabbagh,Dr.Ahmed Alghamdi,Mamdoh Zeyad**
3		Team Formation	2 days	Wed 8/30/23	Thu 8/31/23		Mamdoh Zeyad,Ahmed Sabbagh
4		Propose the Idea	0 days	Fri 9/1/23	Fri 9/1/23	3	Ahmed Sabbagh,Dr.Ahmed Alghamdi,Mamdoh Zeyad
5		Review the Idea	4 days	Fri 9/1/23	Wed 9/6/23	4	Dr.Ahmed Alghamdi
6		Supervisor Meeting	0 days	Thu 9/7/23	Thu 9/7/23	5	Ahmed Sabbagh,Dr.Ahmed Alghamdi,Mamdoh Zeyad
7		**Chapter 1: Planning**	**4 days**	**Tue 9/12/23**	**Mon 9/18/23**	**2**	**Ahmed Sabbagh,Dr.Ahmed Alghamdi,Mamdoh Zeyad**
8		1.1 Introduction	1 day	Tue 9/12/23	Tue 9/12/23		Mamdoh Zeyad
9		1.2 The Problem Definition	1 day	Wed 9/13/23	Wed 9/13/23		Mamdoh Zeyad
10		1.3 Project Aims and Objectives	1 day	Thu 9/14/23	Thu 9/14/23		Ahmed Sabbagh
11		1.4 Project Methodology	1 day	Fri 9/15/23	Fri 9/15/23		Ahmed Sabbagh
12		1.5 Project Plan	1 day	Fri 9/15/23	Fri 9/15/23		Mamdoh Zeyad
13		1.6 Conclusion	1 day	Sat 9/16/23	Sat 9/16/23		Ahmed Sabbagh,Mamdoh Zeyad
14		Supervisor Meeting	0 days	Mon 9/18/23	Mon 9/18/23	13	Ahmed Sabbagh,Dr.Ahmed Alghamdi,Mamdoh Zeyad
15		**Chapter 2: Literature Review**	**9 days**	**Mon 9/18/23**	**Fri 9/29/23**	**7**	**Ahmed Sabbagh,Dr.Ahmed Alghamdi,Mamdoh Zeyad**
16		2.1 Introduction	1 day	Mon 9/18/23	Mon 9/18/23		Mamdoh Zeyad
17		2.2 Stakeholders Definition	1 day	Tue 9/19/23	Tue 9/19/23		Mamdoh Zeyad
18		2.3 Project Domain	1 day	Wed 9/20/23	Wed 9/20/23		Ahmed Sabbagh
19		2.4 Interviews With Some Stakeholders	3 days	Thu 9/21/23	Sat 9/23/23		Ahmed Sabbagh,Mamdoh Zeyad
20		2.5 Literature Review	2 days	Sun 9/24/23	Mon 9/25/23		Ahmed Sabbagh,Mamdoh Zeyad
21		2.6 Comparison Criteria Definition	1 day	Tue 9/26/23	Tue 9/26/23		Ahmed Sabbagh
22		2.7 Comparison Results and the Feasibility Study	1 day	Wed 9/27/23	Wed 9/27/23		Mamdoh Zeyad
23		2.8 Conclusion	1 day	Thu 9/28/23	Thu 9/28/23		Ahmed Sabbagh,Mamdoh Zeyad
24		Supervisor Meeting	0 days	Fri 9/29/23	Fri 9/29/23	23	Ahmed Sabbagh,Dr.Ahmed Alghamdi,Mamdoh Zeyad

Figure 1.3

Project Plan (CH3 & CH4).

GANTT CHART

25		**Chapter 3: Analysis Phase**	**14 days**	**Mon 10/2/23**	**Thu 10/19/23**	**15**	**Ahmed Sabbagh,Dr.Ahmed Alghamdi,Mamdoh Zeyad**
26		3.1 Introduction	1 day	Mon 10/2/23	Mon 10/2/23		Mamdoh Zeyad
27		3.2 Requirements Gathering	1 day	Wed 10/18/2	Wed 10/18/2	35	Mamdoh Zeyad,Ahmed Sabbagh
28		3.3 Functional Requirements	1 day	Thu 10/19/23	Thu 10/19/23	27	Mamdoh Zeyad
29		3.4 Non-Functional Requirements	1 day	Thu 10/5/23	Thu 10/5/23		Mamdoh Zeyad
30		3.5 Hardware Requirements	1 day	Fri 10/6/23	Fri 10/6/23		Ahmed Sabbagh
31		3.6 Actors Definition	1 day	Sat 10/7/23	Sat 10/7/23		Ahmed Sabbagh
32		3.7 Use Case Diagrams	2 days	Mon 10/9/23	Tue 10/10/23		Ahmed Sabbagh
33		3.8 Sequence Diagrams	2 days	Wed 10/11/2	Thu 10/12/23		Mamdoh Zeyad
34		3.9 Activity Diagrams	2 days	Fri 10/13/23	Mon 10/16/2		Mamdoh Zeyad
35		3.10 Conclusion	1 day	Tue 10/17/23	Tue 10/17/23		Ahmed Sabbagh
36		Supervisor Meeting	0 days	Wed 10/18/2	Wed 10/18/2	35	Ahmed Sabbagh,Dr.Ahmed Alghamdi,Mamdoh Zeyad
37		**Chapter 4: Design Phase**	**14 days**	**Sun 10/22/23**	**Wed 11/8/23**	**25**	**Ahmed Sabbagh,Dr.Ahmed Alghamdi,Mamdoh Zeyad**
38		4.1 Introduction	1 day	Mon 10/23/2	Mon 10/23/2		Mamdoh Zeyad
39		4.2 Architecture Design	3 days	Tue 10/24/23	Thu 10/26/23		Mamdoh Zeyad,Ahmed Sabbagh
40		4.3 Class Diagram	1 day	Fri 10/27/23	Fri 10/27/23		Ahmed Sabbagh
41		4.4 Interfaces	5 days	Mon 10/30/2	Fri 11/3/23	40	Ahmed Sabbagh,Mamdoh Zeyad
42		4.5 Conclusion	1 day	Mon 11/6/23	Mon 11/6/23	41	Ahmed Sabbagh,Mamdoh Zeyad
43		Supervisor Meeting	0 days	Wed 11/8/23	Wed 11/8/23	42	Ahmed Sabbagh,Dr.Ahmed Alghamdi,Mamdoh Zeyad
44		**Colsing**	**20 days**	**Sun 11/12/23**	**Thu 12/7/23**	**25,7,15,37**	**Ahmed Sabbagh,Dr.Ahmed Alghamdi,Mamdoh Zeyad**
45		Review the Final Report	5 days	Sun 11/12/23	Thu 11/16/23		Dr.Ahmed Alghamdi
46		Supervisor Meeting	0 days	Thu 11/16/23	Thu 11/16/23		Mamdoh Zeyad,Ahmed Sabbagh,Dr.Ahmed Alghamdi
47		Create the Final Presentation	3 days	Thu 11/16/23	Mon 11/20/2	46	Ahmed Sabbagh,Mamdoh Zeyad
48		Supervisor Meeting	0 days	Mon 11/20/2	Mon 11/20/2		Ahmed Sabbagh,Mamdoh Zeyad,Dr.Ahmed Alghamdi
49		Final Presentation	0 days	Thu 12/7/23	Thu 12/7/23	48	Ahmed Sabbagh,Mamdoh Zeyad,Dr.Ahmed Alghamdi

Figure 1.4

Project Plan, Timeline.

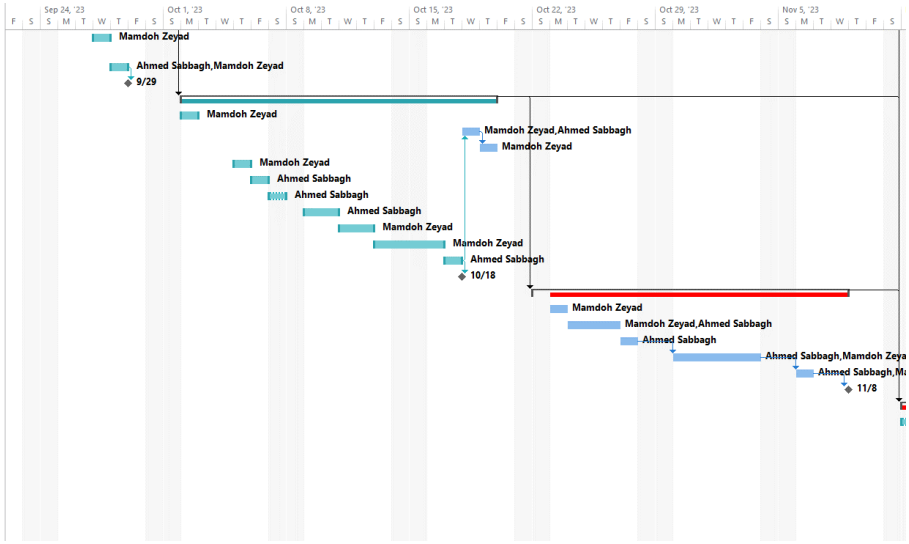
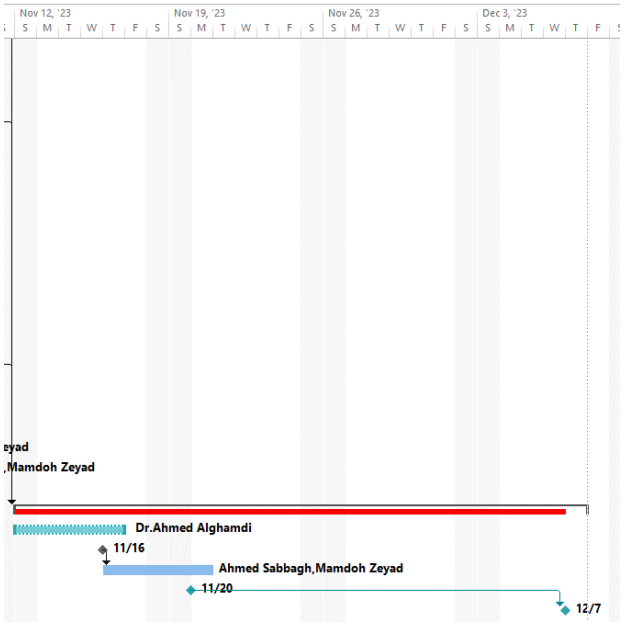


Figure 1.5

Project Plan, Timeline Cont.



1.6. Conclusion

In this chapter, we presented the main motivations that prompted us to propose this project, as well as the main problem facing the users and developers. In addition, also discussed our project aims and objectives as well as the methodology and project plan to start our project. Finally, we have a clear vision and motivation to move to the next phase which is the problem understanding phase.

CHAPTER2: LITERATURE REVIEW

2.1. Introduction

In this chapter, we will discuss the stakeholders of the project and show the project domain. Moreover, we will highlight some competitors who have similar and close ideas to our project to compare them with our idea. We will summarize the comparison and create our feasibility study to put our project on the right track.

2.2. Stakeholders Definition

Table 2.1

Stakeholders' definition.

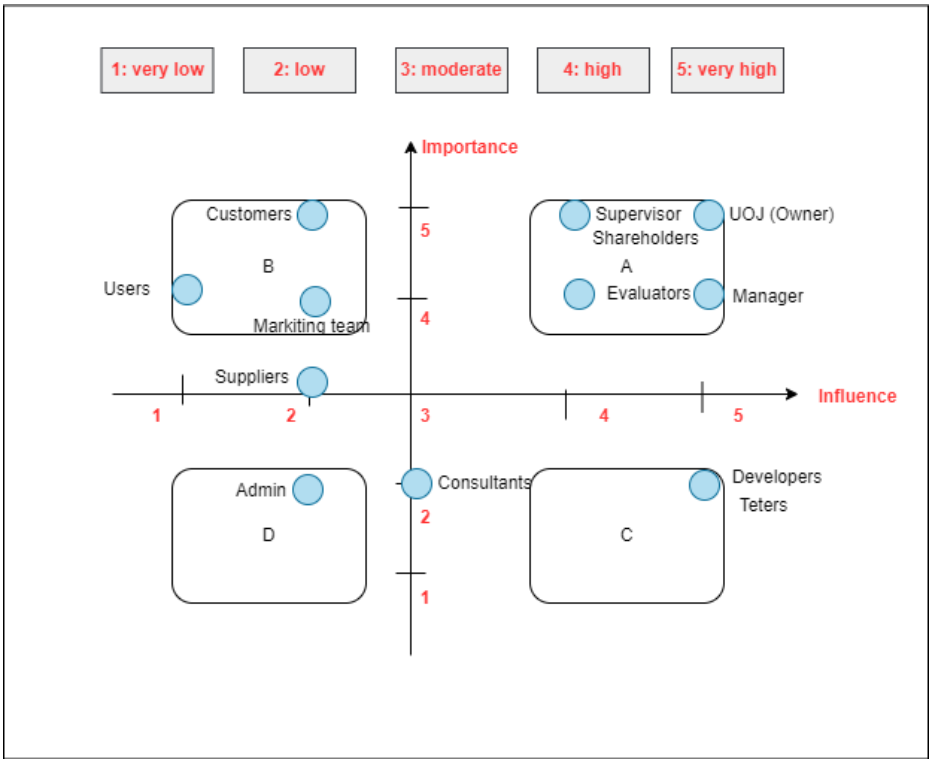
Stakeholder	Importance	Influence	Description
UOJ (Owner)	5	5	Holds an important position in resolving the project, as they are the owners, and they have a financial and strategic investment in the project. In terms of impact, the university has a significant effect on the project. They have the authority to make decisions, set priorities, and determine the project's eventual success.
Dr. Ahmed (Supervisor)	5	4	Holds an important role in overseeing the project and its progress and ensuring that the project is in line with organizational objectives and standards. In terms of impact, influence provides guidance, advice, and guidance to team members, which can affect the course of the project. However, his decision-making powers may be somewhat limited compared to the owner.
Evaluators	4	4	Evaluators provide feedback on the completeness, accuracy, and testability of the requirements in the software system. In terms of impact, evaluators play a critical role by importance and influence by ensuring that the requirements for a software system are complete, testable, and feasible.

Shareholders	5	4	Shareholders are ownership important and influential stakeholders in the project because they are part or all the owners of the company. In addition, Shareholders provide funding for the software development project, which means they have an important role and influence within the project.
Mamdoh (Project Manager)	4	5	He has a critical role in the project including scheduling, resource allocation, and team allocation. Hence, the success of the project is directly linked to Mamdoh's ability to manage these elements effectively. In terms of impact, influence has the authority to make decisions regarding resource allocation, team appointments, and project priorities.
Mamdoh, Ahmed (Developers)	2	5	Has influence over the specific tasks and functionalities he works on, as well as the technical aspects of the project. Possible “project killers “are able to halt the project and jeopardize its success, it is one of the most important elements until the project is successful.
Testers	2	5	Testers are responsible for testing software to ensure that it meets requirements and works as expected. in addition, Testers have a strong influence because they have the responsibility to find defects and problems in the software and report on them.
Suppliers	3	2	Suppliers provide the goods and services that are needed to develop the software system. This can include hardware, software, and other services.

Marketing team	4	2	The marketing team plays an important role by providing insights into the needs and wants of the target market; Therefore, the marketing team has a significant importance on the software system.
Users (consumers, providers)	4	1	Users are the people who interact with a software system. They may be consumers or providers in the system, The user cannot influence the project, but it is very important to ensure pleased with the project by adding all the services he needs.
Customers	5	2	Customers are the people or organizations that pay for a software system. They may be the same people as the users, or they may be different. Customers cannot influence the project, but it is very important to ensure pleased with the project.
Consultants	2	3	Consultants can provide expertise and guidance to users in terms of presenting their specific problem and directing it to the appropriate developer, the consultant has a specific role in influencing the project, so he gives each person his specific role.
Admin	2	2	The admin is the person who is responsible for the day-to-day management of a project. Admin typically has a wide range of responsibilities, for example, providing technical support to users, monitoring system performance and security, etc. Their interests should not be neglected completely.

Figure 2.1

Stakeholders' influence and importance mapping.



2.3. Project Domain

This project operates to support the field of programming services for users and developers which they are users also. The main objectives of the project include making the site unique by providing all services used for the user and the developer to have a suitable environment that helps him solve problems. In addition, one of our goals is to be among the most common sites, not only in the Middle East but also on a global scale.

The project aims to comprehensively the needs of the developer by providing basic services that help the developer solve small problems through the availability of an integrated development environment (IDE) within the site. Moreover, an AI-talking bot will be added to provide programming assistance to developers.

Furthermore, a chat system in a suitable and effective form will be implemented that allows the user to chat with the developer or consultant directly.

2.4. Data Collection

To fully understand our project (CODEE) from another perspective, we conducted many interviews with two main stakeholders' the user and developer as will be discussed in the following:

Interviews with users: We interviewed several users to understand the issues they currently face in the sites with similar ideas to our project, and the views of the majority were the same, they agreed on several points, which I will clarify in the following topics:

- **The initial consensus** agreed upon is that they face difficulties while determining the type of issues assigned to their project. At this point, we have worked on our project (CODEE) by adding a consultant, which will contribute to solving these issues that many people face.
- **The second consensus**, was about the problems they faced in communicating with the developer to solve the problem, so the majority were facing problems due to the delay in responding due to the way of communication by e-mail for example, and our solution was within our project (CODEE), which is to add chat in an organized and effective manner to the conversation between the user and the developer so that it is appropriate for everyone.

Interviews with developers: We interviewed several developers to understand the problems they currently face in the sites that use similar ideas to our project, and the views of the majority were similar, and they agreed on several points, which I will clarify in the following topics:

- **The initial consensus**, agreed upon is that they face difficulties when users come to them and they have issues it is not from the field of the developer to solve, we have worked on this within our project (CODEE) by adding a consultant, which will contribute to solving these issues that many people face.

- **The second consensus**, was about the problems developers faced in communicating with the user to solve the problem, so the majority were facing problems due to the delay in responding due to the way of communication by e-mail, for example, and our solution was within our project (CODEE), which is to add chat in an organized and effective manner to the conversation between the user and the developer so that it is appropriate for everyone.
- **The third consensus**, agreement pertained to Many developers who agreed that the current site environment does not support sufficient services for the developer so that he can solve issues, for example, many sites do not provide IDE, and this is an issue facing the developer because the basic services are not provided by the sites.

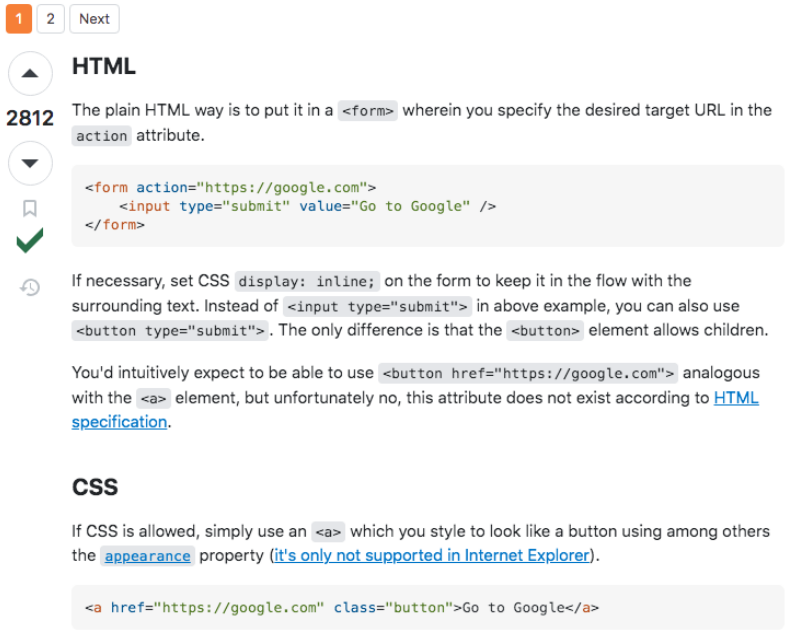
2.5. Literature Review

In terms of improving the user and developer experience, we searched many websites that provide the exact idea of our project, which can be categorized into four main categories including:

1. **websites that include solving programming issues services without any direct chat.** On these kinds of websites, users often have programming experience. Thus, they pose their programming problems while waiting for other programmers to solve these programming issues. Moreover, finding the right solution will take some time. One of the most commonly used websites in this category is StackOverflow. We noticed there is no chatting system, no IDE, no AI-Chatbot, and no guarantee that the programmer has sufficient experience to solve your problem. Figure 2.2 shows a snippet from the website.

Figure 2.2

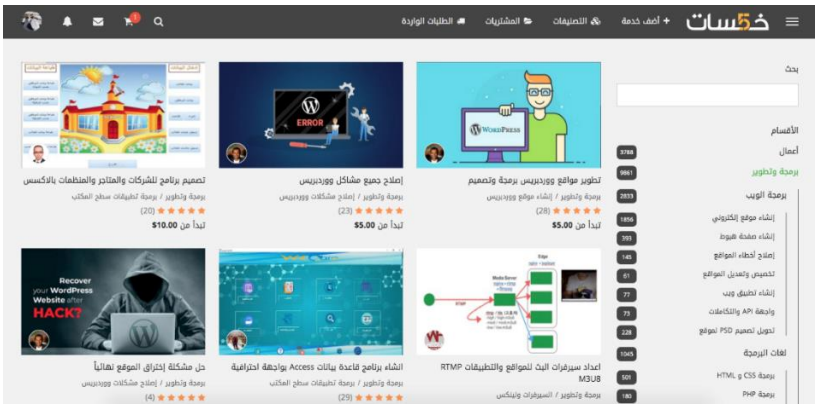
StackOverflow.



2. **Websites that include a guarantee and effective developers without chatting system or consultants.** Moreover, programmers do not have any features. Khamsat is an example of this category. The working process of the website is as follows: The user will choose one of the available programmers, each programmer has a programming category, and you cannot chat with the programmer directly. In addition, programmers do not have any features on their side. Figure 2.3 shows a snippet from the website.

Figure 2.3

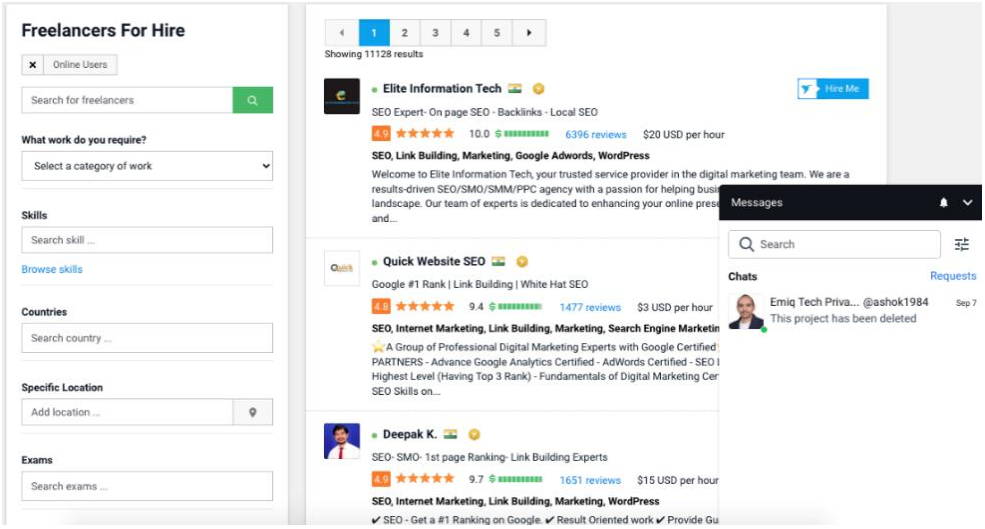
Khmsat.



3. **Websites that include solving programming issues services, and users can chat with the developers directly.** Freelancer is a good example of this category. The working process for this website is as follows: as a user, you can chat with any available developer and choose him if you wish, users can post a project and let the developers join it and decide which one to choose after you chat with him directly. In addition, the developer can chat with any user before joining any project to work on it. Seemingly, developers do not have any other features on the website like IDE or AI bot that will help them to develop small projects directly on the website. Figures 2.4 shows a snippet from the website.

Figure 2.4

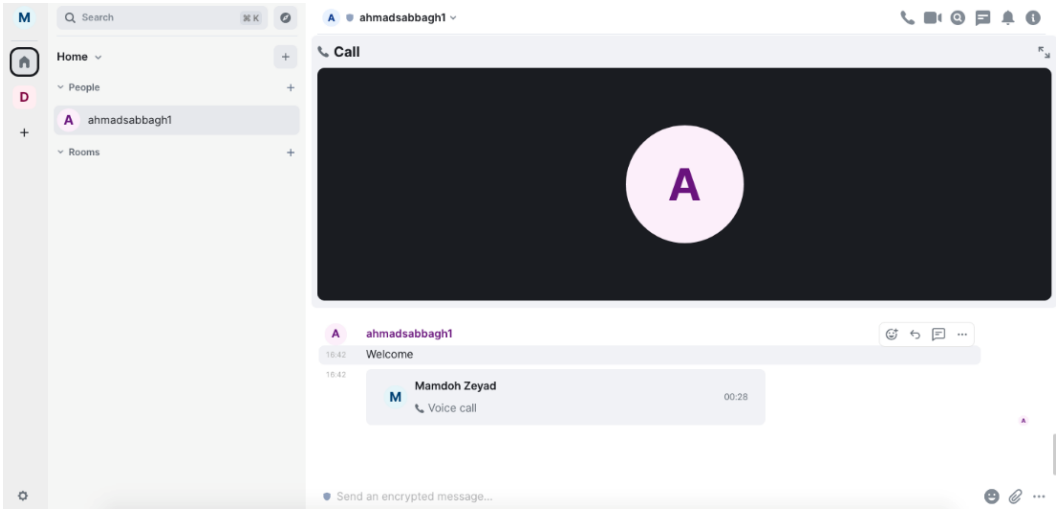
Freelancer.



4. **Websites that include only a chatting system without any other features.** Gitter is an example of this category, this website only provides a chat system to communicate with users or developers or any kind of communication. Figure 2.5 shows a snippet from the Gitter website.

Figure 2.5

Gitter.



2.5.1. Comparison Criteria Definition

After an in-depth analysis of the working process on other websites that we discussed previously in this chapter, we found that the services that we are trying to provide are provided by these sites in parts, but we did not find a site that provides all these services at the same time. Therefore, in comparing our project with other sites, we relied on important main points as follows:

- Effective mechanism to solve programming issues.
- Direct chatting system on the website.
- The consultant will guide the users to choose the programmers wisely.
- Development environment for the developer (IDE, AI-talking bot).

After visiting all the sites, we discussed earlier and using them according to each process, in the following Table 2.1, we will summarize the differences between our project and these sites according to the points mentioned before.

Table 2.2

Comparison criteria definition.

Site name	Solve the programming issues effectively	Chatting availability	Consultant availability	IDE & AI availability
StackOverflow	Yes	No	No	No
Khamsat	Yes	No	No	No
Freelancer	Yes	Yes	No	No
Gitter	No	Yes	No	No
CODEE	Yes	Yes	Yes	Yes

2.5.2. Comparison Results and The Feasibility Study

After the analysis mentioned before, we decided that our project would avoid all the flows found in other sites we covered in our study. Moreover, our goal is to succeed in serving all the users' and developers' needs to help them with their programming issues. Furthermore, our proposed project will help the user to be guided by the consultant to choose the right developer for his programming needs and keep the user tracking his issues until it is solved. On the other hand, all the mentioned sites in our study only focus on the users; as programmers, this made us more excited to propose this idea to help the developers gain an effective development environment.

As we have noted from our competitive sites attempting to provide the minor functionalities that will help users or developers solve their programming issues without looking to other services seriously, but this is not enough in some cases; not all the users have the required experience to choose the developers individually, and developers will waste their time on projects not curious on them. Thus, our project will guide the user and save the developer's valuable time.

2.6. Conclusion

In this chapter, we reviewed the stakeholders of the project and showed the project domain and the working area. Moreover, we studied the current competitors who have similar and close ideas, and we compared our idea with these ideas and summarized the most important points that the project provides over the rest of the competitors. Now we have the full vision to begin collecting user needs, detailing them, analyzing them, and starting the next phase entitled “Analysis Phase.”

CHAPTER3: ANALISYS PHASE

3.1. Introduction

In this chapter, we will discuss how we will gather the requirements. Moreover, we will describe all the functional requirements, non-functional requirements, and hardware requirements. We will define the actors of the system. Lastly, we will present the UML diagrams proposed for our system (Activity Diagrams, Use Case Diagrams, Sequence Diagrams).

3.2. Requirements Gathering

Requirements gathering is a long process that requires patience and powerful analysis skills. Regarding that, our proposed method for requirements gathering is observations and some interviews that we conducted previously which showed us the required functions for supporting our proposed ideas.

3.2.1. Functional Requirements

Table 3.1

Functional Requirements

#	Description	Actors
1	Users must be able to register for an account on the website.	All Users
2	Users must be able to log in to their account by using their email and password.	All Users
3	Users must be able to reset their password.	All Users
4	The user (consumer) must be able to view a list of all developers and consultants with their related information (name, category, and chat icon).	All Users Except Admin
5	The user (consumer) must be able to choose any developer or consultant to initiate a chat session with them.	All Users Except Admin
6	The users must be able to send and receive text messages, and files.	All Users Except Admin

7	The users must be able to be notified when they receive new messages.	All Users Except Admin
8	The developer must be able to use the IDE to write, edit, debug, and test code.	Developer
9	The developer must be able to use an AI-talking bot that can help him in solving programming issues.	Developer
10	The developer or consultant must request (Developer and Consultant Verification) to authenticate this user is a certified developer or consultant.	Developer & Consultant
11	The admin must be able to view a list of all users in the system	Admin
12	The admin must be able to add, edit, and delete users to the system	Admin
13	The admin must be able to view a list of all messages from all users related to any issue in the system.	Admin
14	The admin must be able to verify the requests of developers and consultants.	Admin

3.2.2. Non-Functional Requirements

Gathering non-functional requirements requires important quality attributes to be considered and gathering requirements depends on these quality attributes. We will gather the requirements regarding the following quality attributes:

- Performance: how fast does the system need to react to requests?
- Scalability: how will the system react to heavy load?
- Availability: how the system will be available to users?
- Security: how the system will protect the users from unauthorized access?
- Usability: how the system be effective, efficient, and easy to use?
- Maintainability: how easy modifying the system regarding fixing bugs, environmental adaptation, and adding new features?
- Portability: how the system can accommodate different platforms?

Table 3.2

Non-Functional Requirements

#	Description	Quality Attribute
1	The system must be able to respond to user requests under 150ms.	Performance
2	The system must be able to handle 2000 concurrent users.	Performance
3	The system must be able to handle a 100x increase in load without sacrificing performance.	Scalability
4	The system must be able to scale vertically by adding more resources to existing servers.	Scalability
5	The system must be available 24/7 of the time.	Availability
6	The system must be able to recover from failures within 10 minutes.	Availability
7	The system must be protected from unauthorized access, use, disclosure, disruption, modification, or destruction.	Security
8	The system must use strong encryption to protect sensitive data.	Security
9	The system must be designed to be resistant to common attack vectors, such as SQL injection, cross-site scripting, and denial-of-service attacks.	Security
10	The system must be easy to learn, efficient to use, and memorable to use.	Usability
11	The system must be friendly with standard language.	Usability
12	The system must be easy to modify and add new features without introducing new errors.	Maintainability
13	The system must be well-documented.	Maintainability
14	The system must be easy to deploy and update.	Maintainability
15	The system must have a process for backing up and restoring data.	Maintainability
16	The system must be able to be deployed on multiple platforms without modification.	Portability
17	The system must use standard programming languages and technologies.	Portability

3.2.3. Hardware Requirements

Using web-based applications does not require high-performance hardware components, because it depends on the internet connection speed instead of the hardware components.

Table 3.3

Hardware Requirements

Hardware Component	Minimal Requirements
Processor	Intel I3 Gen 5 or above or the equivalent from AMD.
RAM	4 GB.
Disk Capacity	256 GB.

3.3. Actors Definition

Identifying the actors is a critical step that will help you design the required UML diagrams for the system. In our proposed system, we have four main actors identified as follows:

Table 3.4

Actors Definition

Actor	Definition
User (Consumer)	Requesting help for solving his/her programming issues.
Developer (Provider)	Providing help related to the consumer’s programming issues.
Consultant	Providing the help for the user to choose the right developer.
Admin	Managing users of the system by monitoring their issues and fixing them.

3.4. UML Diagrams

UML Diagrams are the best way to represent the system to others, especially those who don't have any technical knowledge. We will present the activity diagrams, use case diagrams, sequence diagrams, and class diagram.

3.4.1. Activity Diagrams

Activity diagrams will help you to gain knowledge about the sequence of activities for each actor during the use of the system. The following are our proposed activity diagrams:

- Figure 3.1: The user (consumer) activity diagram.
- Figure 3.2: The developer (provider) activity diagram.
- Figure 3.3: The consultant activity diagram.
- Figure 3.4: The admin activity diagram.

Figure 3.1

The user (consumer) activity diagram.

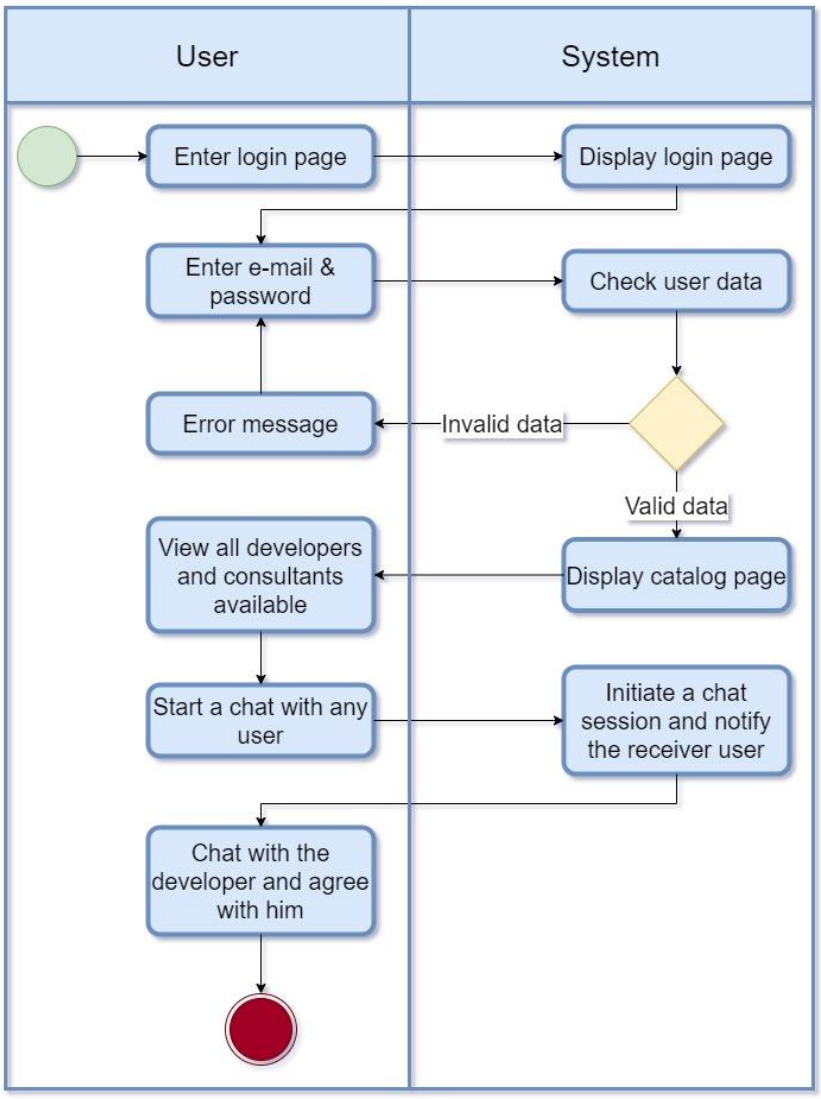


Figure 3.2

The developer (provider) activity diagram.

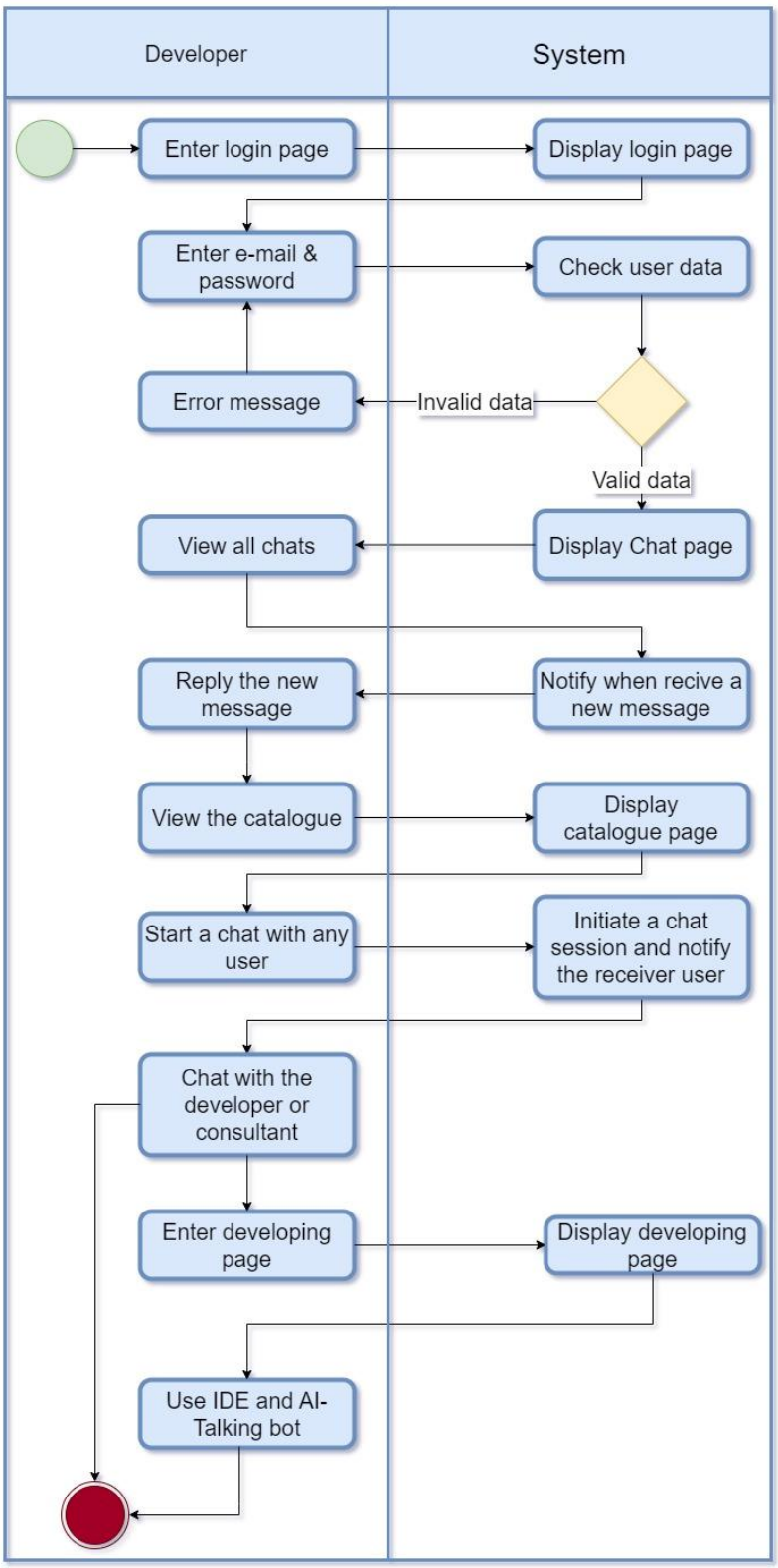


Figure 3.3

The consultant activity diagram.

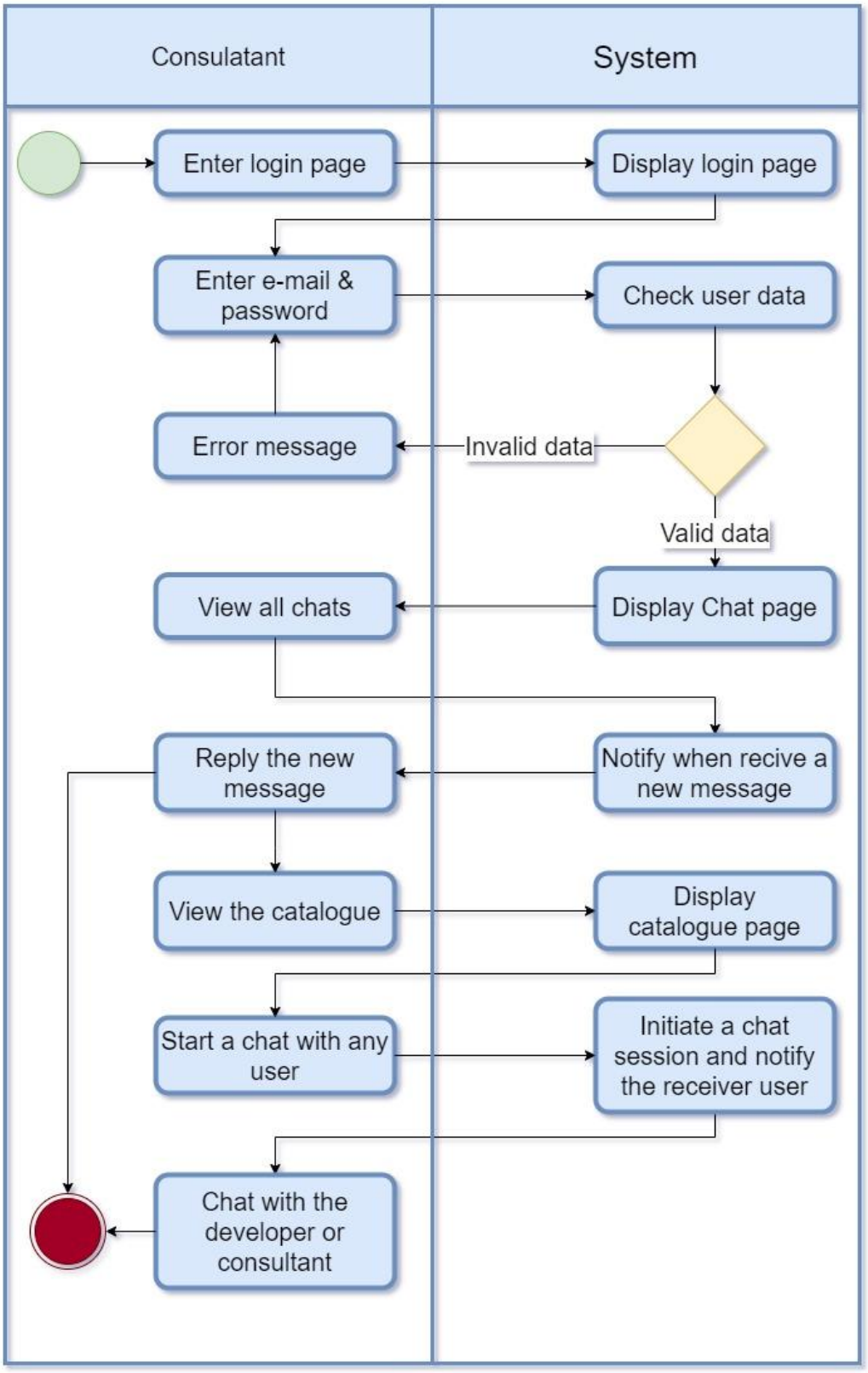
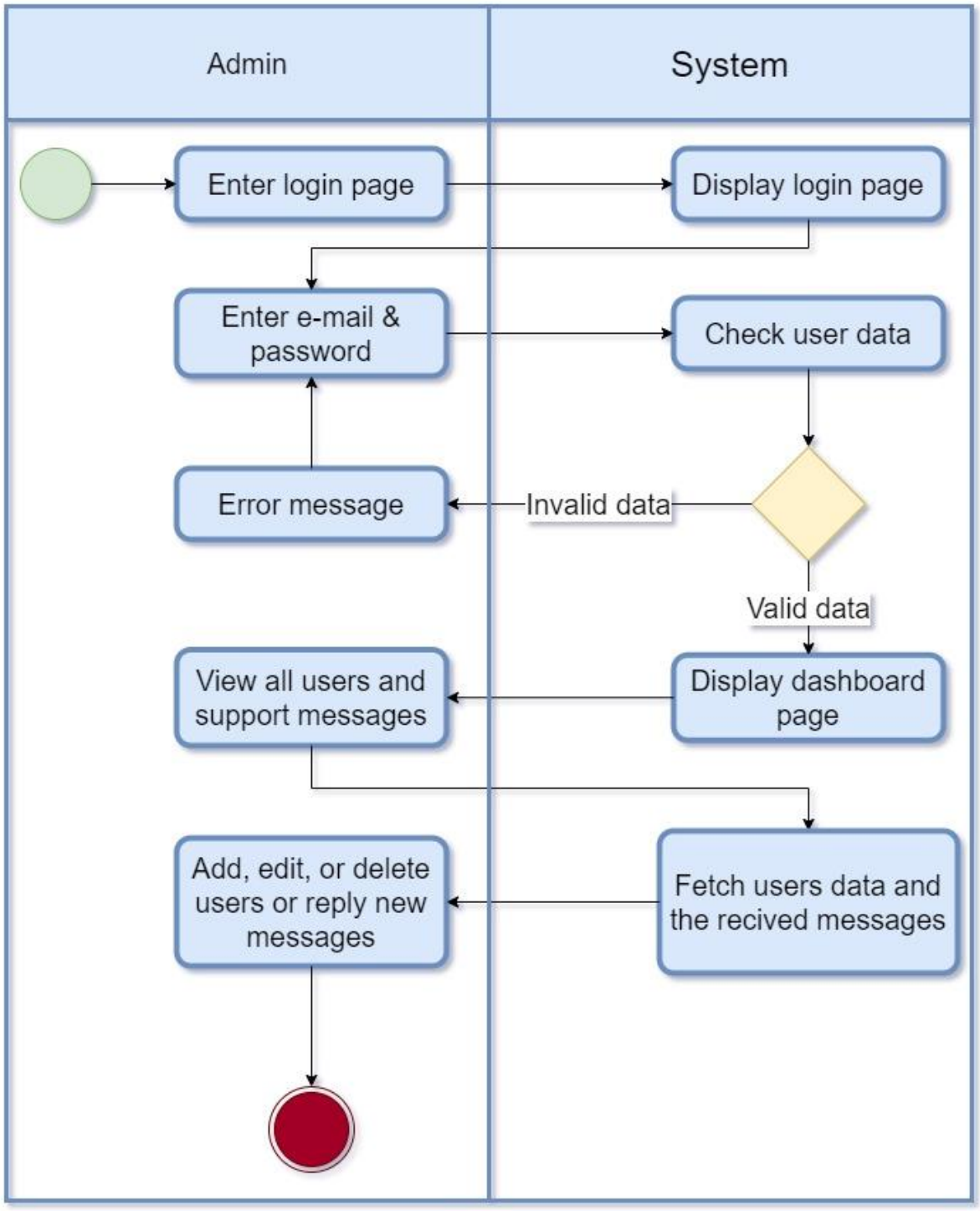


Figure 3.4

The admin activity diagram.



3.4.2. Use Case Diagrams

Use case diagrams are considered behavior diagrams in UML, they are used to represent the functionality of the system using actors and use cases. "Actors" are people or entities operating under defined roles within the system. These actors perform the set of functionalities represented by the use case diagrams. The following are our proposed use case diagrams:

- Figure 3.5: Users and consultants use case diagram.
- Figure 3.6: Developers and admins diagram.

Figure 3.5

Users and consultants use case diagram.

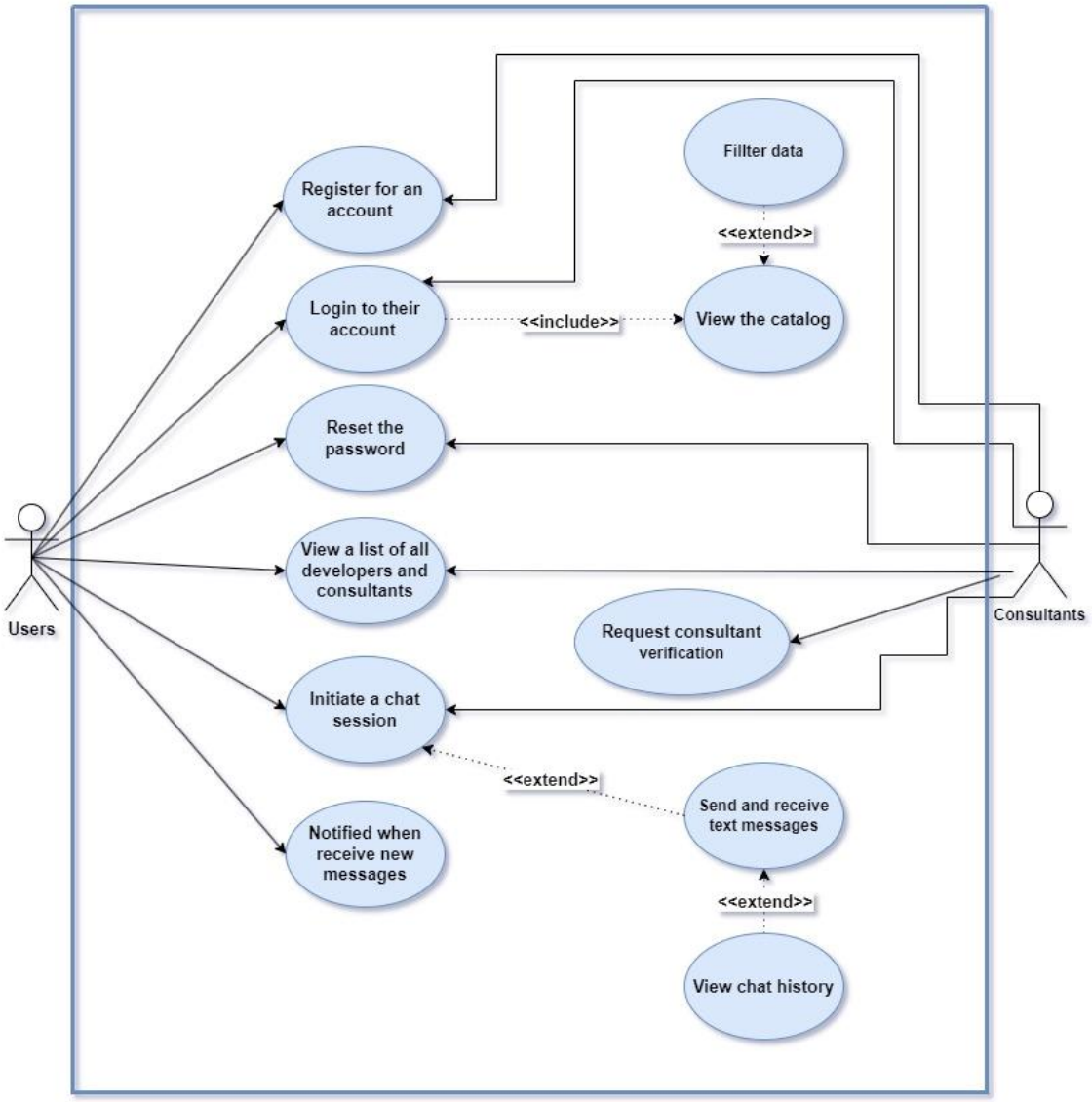
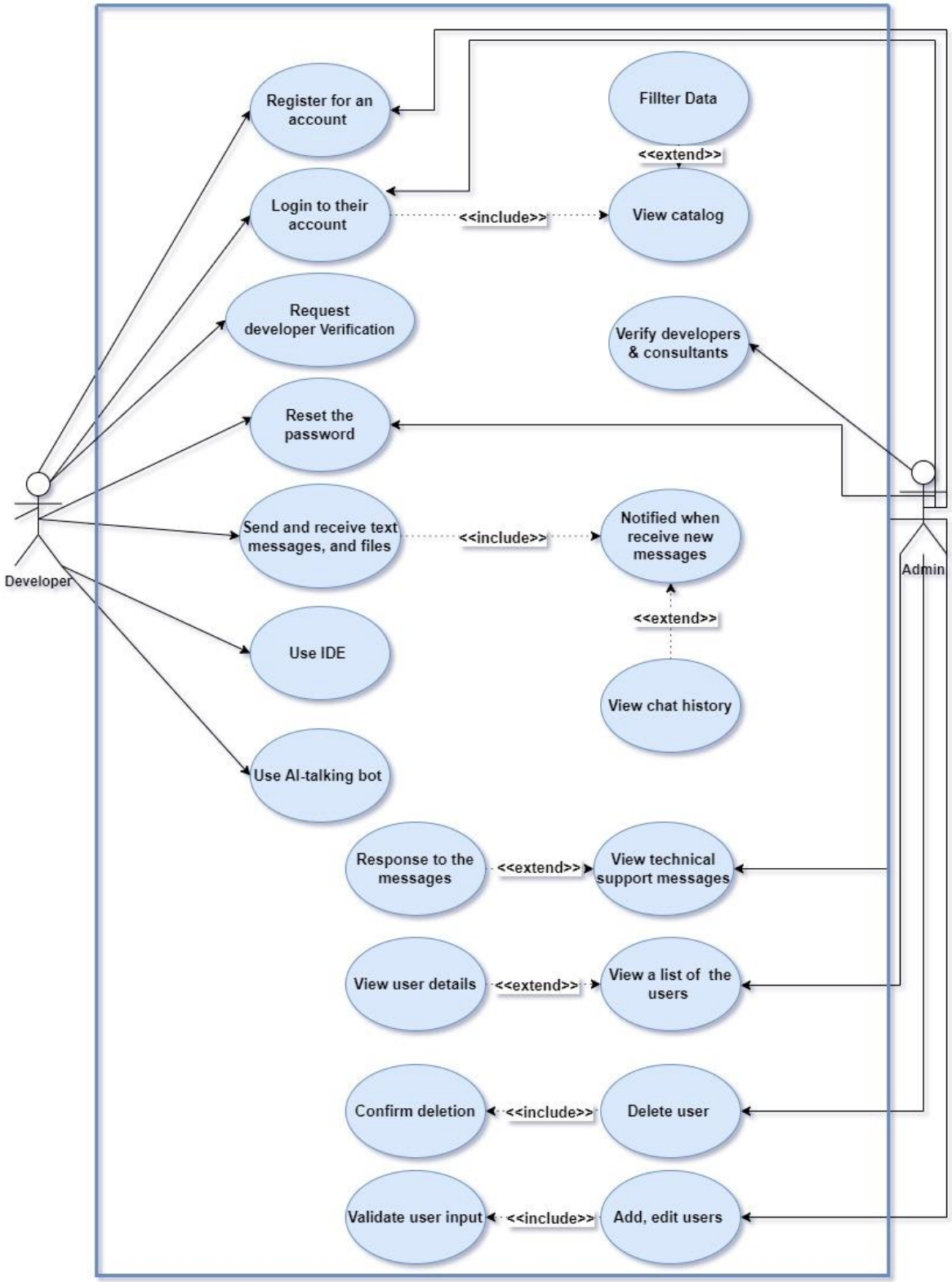


Figure 3.6

Developers and admins diagram.



3.4.2.1. Use Case Diagrams Specifications

Table 3.5

Use-Case.1

Use case name	Register for an account.	UC NO. 1
Scenario	register for an account is the first page of the system user will go through this page to the login page when entering valid personal information.	
Triggering Event	The user wants to create an account for a website or service.	
Brief description	The user moves to the registration page and enters their personal information, The user then clicks the Submit button to create their account.	
Actors	**All system actors.**	
Related use case	**Related to all use cases.**	
Preconditions	- The user is not already logged in. - The user has a valid email address. - The user has created a strong password. - The user agrees to the terms of service.	
Postconditions	- A new account is created for the user.	
Flow of activities	- The user visits the registration page and enters their personal information. - The user agrees to the terms of service. - The user clicks the "Register" button. - The system validates the user's email address and password. - The system creates a new account for the user.	
Exception Condition	- The user enters an invalid email address. - The user enters a weak password. - The user does not agree to the terms of service.	

Table 3.6

Use-Case.2

Use case name	Log in to their account.	UC NO. 2
Scenario	User login is goes through this page to specific pages when entering a valid username and password system will detect the user type and show the user permission depending on his type.	
Triggering Event	The user wants to access their account on a website.	
Brief description	The user moves to the login page and enters their email address and password. The user then clicks the Login button to access their account.	
Actors	All system actors.	

Related use case	Related to all use cases.
Preconditions	- The user is not already logged in. - The user has a valid username and password.
Postconditions	- The user has access to all of the functionality of the system that is available to logged-in users.
Flow of activities	- The user visits the login page and enters their username and password. - The user clicks the "Login" button. - The system validates the user's username and password.
Exception Condition	- username and password empty values. - wrong username or password.

Table 3.7

Use-Case.3

Use case name	Reset the password.	UC NO.	3
Scenario	This page exists if the user wants to change the password.		
Triggering Event	The user has forgotten their password.		
Brief description	The user reset their password by following a link sent to their email address.		
Actors	All system actors.		
Related use case	Related to all use cases.		
Preconditions	The user has a valid email address associated with his account.		
Postconditions	The user's password has been reset.		
Flow of activities	- The user visits the reset password page and enters their email address. - The user clicks the "Reset password" button. - The system sends a password reset link to the user's email address.		
Exception Condition	The user enters an invalid email address.		

Table 3.8

Use-case.4

Use case name	View a list of all developers and consultants.	UC NO.	4
Scenario	After completing the login process, a page will appear for the user so that the available developers and consultants appear to him and choose the appropriate one for him.		
Triggering Event	The user wants to view a list of all developers and consultants in the system.		
Brief description	The user moves to the Developers and Consultants page and views a list of all developers and consultants in the system.		

Actors	Users (consumers) and consultants.
Related use case	-
Preconditions	The user is logged in to the system.
Postconditions	The user is given a list of all developers and consultants available in the system.
Flow of activities	- The user moves to the Developers and Consultants page. - The system displays a list of all available developers and consultants in the system. - The user can view the details of each developer and consultant by clicking on their name.
Exception Condition	The user is not logged in to the system.

Table 3.9

Use-Case.5

Use case name	Initiate a chat session	UC NO. 5
Scenario	This page will appear after the user chooses the appropriate developer who will be able to create a chat session that will enable him to be chatting with the developer.	
Triggering Event	The user wants to initiate a chat session with a developer.	
Brief description	initiate a chat session with a developer.	
Actors	Users (consumers).	
Related use case	-	
Preconditions	- The user is logged in to the system. - The chat feature is enabled.	
Postconditions	The user is connected to a developer and can begin chatting.	
Flow of activities	- The user clicks the chat button. - The system displays a chat window. - The user clicks the "Start chat" button.	
Exception Condition	The user is not logged in to the system.	

Table 3.10

Use-Case.6

Use case name	Send and receive text messages. UC NO. 6
Scenario	This page will appear after the user initiates a chat session through which he will be able to send and receive text messages and files.
Triggering Event	The user wants to send or receive a text message.
Brief description	The user initiates a chat session after the user chooses the appropriate developer they want to message and enters their message. The user then clicks the "Send" button. In addition, the user can view the messages they have sent and received.
Actors	Users.
Related use case	Initiate a chat session.
Preconditions	The user's is connected to the cellular network or Wi-Fi.
Postconditions	- The message is sent to the appropriate developer. - The user can view the messages they have sent and received.
Flow of activities	- The user initiates a chat session. - The user selects the appropriate developer. - The user enters their message. - The user clicks the "Send" button. - The user can view the messages they have sent and received in the messaging app.
Exception Condition	The user is not connected to the cellular network or Wi-Fi.

Table 3.11

Use-Case.7

Use case name	Notified when receive new messages. UC NO. 7
Scenario	This page will appear after the messages are sent to the user. The system will notify the user of the being of new message.
Triggering Event	The user receives a new message.
Brief description	The system sends a notification to the user informing them that they have received a new message.
Actors	users
Related use case	Send and receive text messages
Preconditions	The user's device is connected to the cellular network or Wi-Fi.
Postconditions	The user is notified that they have received a new message.
Flow of activities	- The user receives a new message. - The system generates a notification. - The system sends the notification to the user's device.

Exception Condition	The user's device is not connected to the cellular network or Wi-Fi.

Table 3.12

Use-Case.8

Use case name	Use IDE.	UC NO. 8
Scenario	This page will appear to the developer after the user sends the specific issues to the developer, which will allow him to write, debug, and test code, Inside the website.	
Triggering Event	The developer wants to write, edit, debug, and test code.	
Brief description	The developer opens the IDE and begins writing code. in addition, the developer can also use the IDE to debug and test the code.	
Actors	Developer.	
Related use case	-	
Preconditions	The developer has sign in inside the website.	
Postconditions	The developer has written, debugged, and tested the code.	
Flow of activities	- The developer selects the specific issues. - The developer opens the IDE. - The developer selects the project they want to work on. - The developer begins writing code.	
Exception Condition	The IDE crashes.	

Table 3.13

Use-Case.9

Use case name	Use AI-talking bot.	UC NO. 9
Scenario	This page will appear to the developer so that the developer can deal with the AI-talking bot in order to give it some assistance for specific issues.	
Triggering Event	The developer wants to use an AI-talking bot to help them in solution the issues	
Brief description	The developer opens the AI-talking bot and asks questions about their code and debugging tasks. The bot provides the developer with answers and assistance.	
Actors	Developer	
Related use case	-	
Preconditions	The developer is connected to the internet.	
Postconditions	The developer has used the AI-talking bot to help them with their work.	

Flow of activities	- The developer opens the AI-talking bot. - The developer enters a question to the bot. - The bot processes the developer's input and generates a response.
Exception Condition	- The developer is not connected to the internet. - The AI-talking bot is unable to understand the developer's input.

Table 3.14

Use-Case.10

Use case name	View technical support messages.	UC NO.	10
Scenario	This page will appear to the admin so that he can view technical support messages for the user.		
Triggering Event	The admin wants to view technical support messages.		
Brief description	The admin moves to the Technical Support Messages page and views a list of all technical support messages.in addition, the admin can also filter the messages by status.		
Actors	Admin		
Related use case	-		
Preconditions	The admin is logged in to the system.		
Postconditions	The admin is presented with a list of all technical support messages.		
Flow of activities	- The admin moves to the Technical Support Messages page. - The system displays a list of all technical support messages. - The admin can filter the messages by status. - The admin can view the details of each technical support message by clicking on the message's subject line.		
Exception Condition	The user is not logged in to the system.		

Table 3.15

Use-Case.11

Use case name	Add, edit, and delete users.	UC NO.	11
Scenario	This page will appear when the admin wants to add, edits, or deletes a new user within the system		
Triggering Event	The admin wants to add, edit, or delete a user.		
Brief description	The admin moves to the User Management page and adds, edits, or deletes users.		
Actors	Admin.		

Related use case	View a list of the users.
Preconditions	- The admin is logged in to the system. - The admin has permission to add, edit, and delete users.
Postconditions	The admin has added, edited, or deleted a user.
Flow of activities	To add a user: o The admin moves to the User Management page. o The admin clicks the Add User button. o The admin enters the user information. o The admin clicks the Submit button. o The system creates the user account. To edit a user: o The admin moves to the User Management page. o The admin finds the user they want to edit and clicks on their name. o The admin edits the user's information. o The admin clicks the Submit button. To delete a user: o The admin moves to the User Management page. o The admin finds the user they want to delete and clicks on their name. o The admin clicks the Delete User button. o The system confirms the order. o The admin clicks the Delete button. o The system deletes the user account.
Exception Condition	- The admin is not logged in to the system. - The admin does not have permission to add, edit, and delete users. - The user's information is invalid.

Table 3.16

Use-Case.12

Use case name	View a list of the users.	UC NO. 12
Scenario	This page will appear when the admin wants to view a list of all users.	
Triggering Event	The admin wants to view a list of all users.	
Brief description	The admin moves to the User Management page and views a list of all users. The admin can also filter the users by status.	
Actors	Admin.	
Related use case	Add, edit and delete users for the admin.	
Preconditions	- The admin is logged in to the system. - The admin has permission to view users.	
Postconditions	The admin is presented with a list of all users, filtered by the status.	
Flow of activities	- The admin moves to the User Management page.	

	- The system displays a list of all users. - The admin can filter the users by status. - The admin can view the details of each user by clicking on the user's name.
Exception Condition	- The admin is not logged in to the system. - The admin does not have permission to view users.

Table 3.17

Use-Case.13

Use case name	Request developer and consultant verification. UC NO. 13
Scenario	This page will appear when the developer and consultant want to send a Request for verification of their account.
Triggering Event	The developer and consultant want to verification of their account.
Brief description	The Developer and Consultant move to the Verification page and fill out a form with their experience such as a portfolio of work or a copy of their resume.
Actors	Developer and consultant.
Related use case	-
Preconditions	- The developer or consultant has an account. - The developer and consultant have met the requirements for his personal information.
Postconditions	The developer or consultant has submitted a request for verification of their account.
Flow of activities	- The Developer and Consultant move to the Verification page. - Submit the documentation, such as a portfolio of work or a copy of their resume. - The user clicks the Submit button. - The system receives the user's submission. - The system reviews the user's submission. - If the user's submission is approved, the system sends the user a notification informing them that their request has been approved and that they are now a verified developer or consultant. - If the user's submission is denied, the system sends the user a notification informing them that their request has been denied and will stay as a user.
Exception Condition	- The Developer or Consultant doesn't have an account. - The Developer or Consultant doesn't submit all the documentation.

Table 3.18

Use-Case.14

Use case name	Verify developers and consultant. UC NO. 14
Scenario	This page will appear when the admin verifies the developer and consultant request.
Triggering Event	The admin wants to review and verify requests for developer and consultant verification.
Brief description	The admin moves to the Developer and Consultant Verification page and reviews the submitted requests. The admin then decides whether to approve or deny each request.
Actors	Admin.
Related use case	Request developer and consultant verification
Preconditions	The admin is logged in to the system.
Postconditions	The admin has reviewed and verified developer and consultant requests.
Flow of activities	- The admin views the developers and consultants' verification requests through the dashboard. - The admin selects a request to review. - The admin reviews the documentation they have submitted. - The admin decides whether to approve or deny the request.
Exception Condition	The admin is not logged in to the system.

3.4.3. Sequence Diagrams

A sequence diagram shows the sequential flow of messages between objects during an interaction. The following are our proposed sequence diagrams:

- Figure 3.7: Login sequence diagram.
- Figure 3.8: Chat system sequence diagram.
- Figure 3.9: IDE and AI-Talking bot sequence diagram.
- Figure 3.11: User Management sequence diagram.

Figure 3.7

Login sequence diagram.

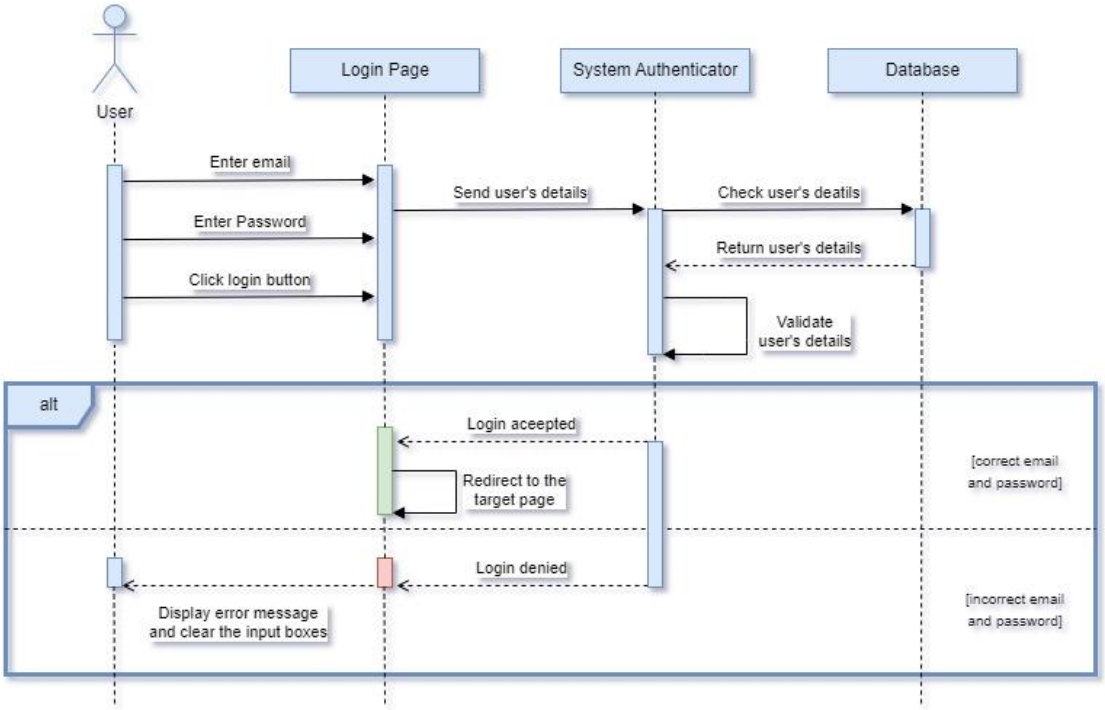


Figure 3.8

Chat system sequence diagram.

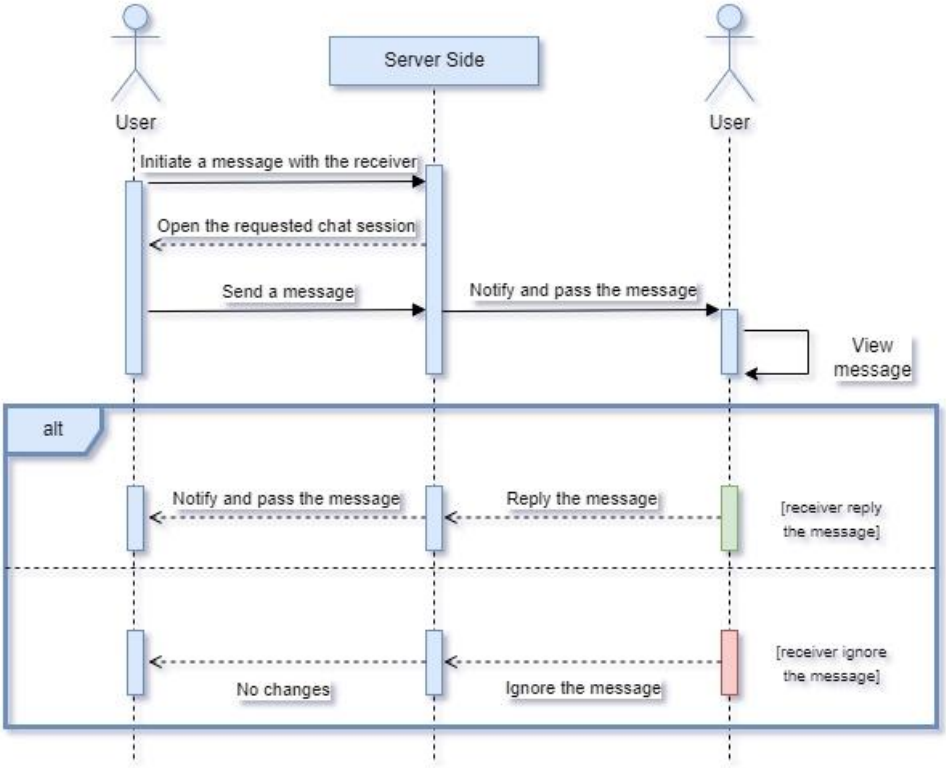


Figure 3.9

IDE and AI-Talking bot sequence diagram.

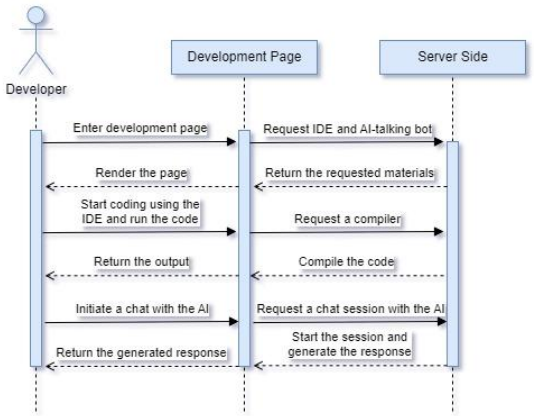
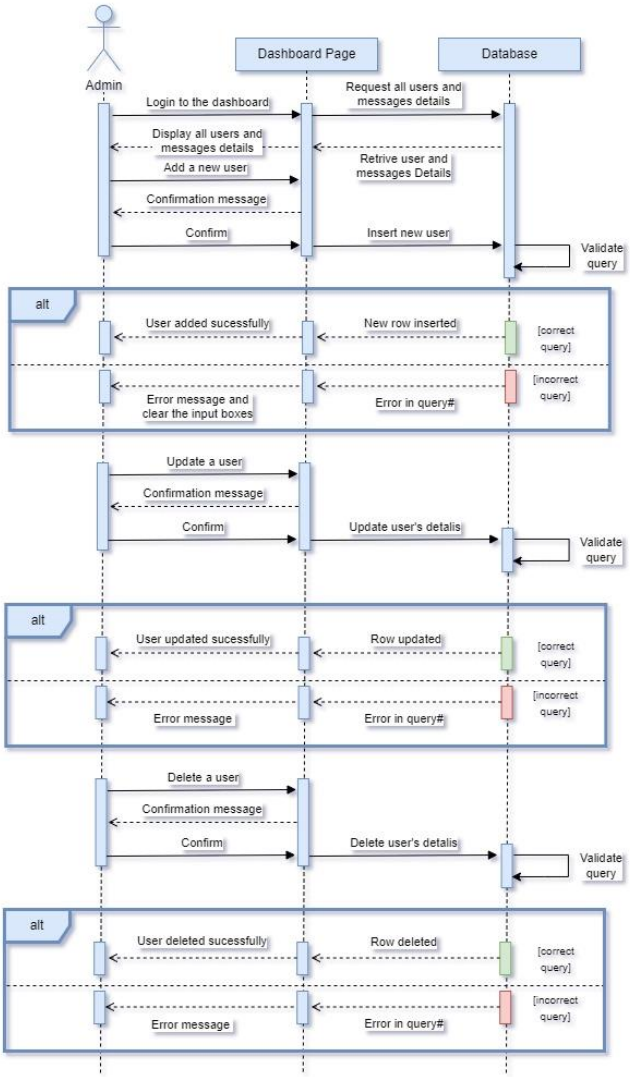


Figure 3.10

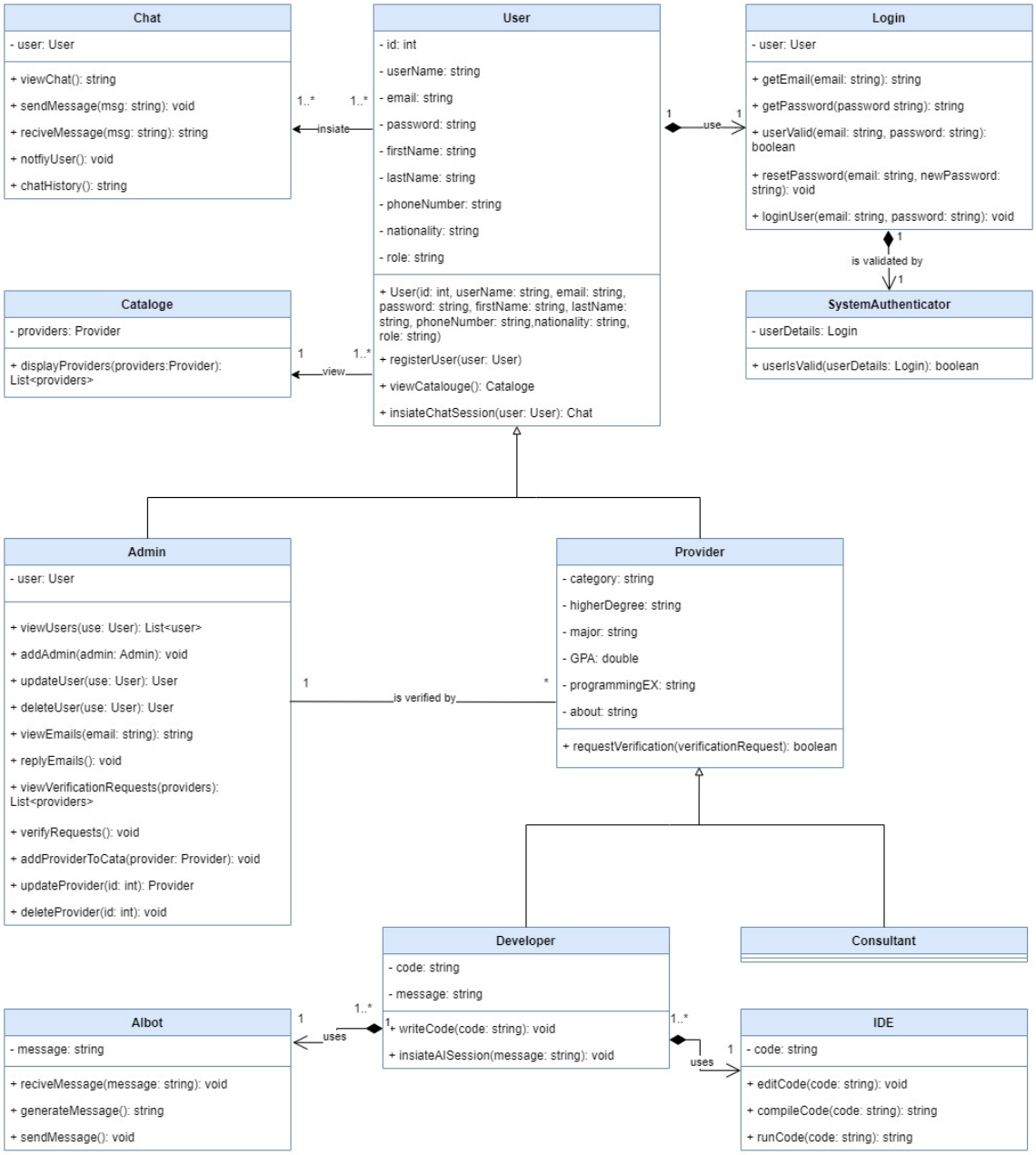
User Management sequence diagram.



3.4.4. Class Diagram

Figure 3.11

Class diagram.



3.5. Conclusion

In this chapter, we reviewed the required functions for our proposed system and identified and explained the influential parties in this system. Moreover, we explained the system's structure through appropriate diagrams.

CHAPTER4: DESIGN PHASE

4.1. Introduction

In this chapter, we will discuss the architecture of the system and related patterns. Moreover, we will discuss the class diagram, entity diagram, database schema, deployment diagram, and the proposed user interfaces.

4.2. System Architecture

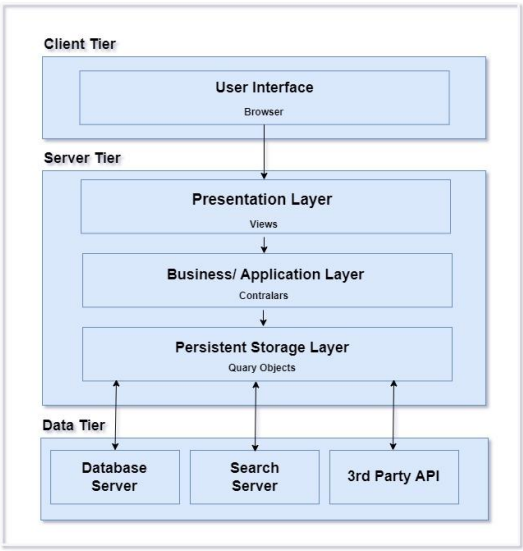
"Codee" is a web-based application that has multiple business domains including chat, payment, IDE, and AI-bot. Thus, we decided to divide our system into layers to enhance understanding and maintainability.

We discussed using the layered architecture pattern for our proposed system because it organizes an application into horizontal layers, each with a defined responsibility and interacting with other layers through well-defined interfaces. This pattern is a scalable solution and solves the problem of updating the clients as the user interface lives and is compiled on the server, making it a suitable choice for our proposed system.

4.2.1. Architectural Pattern

Figure 4.1

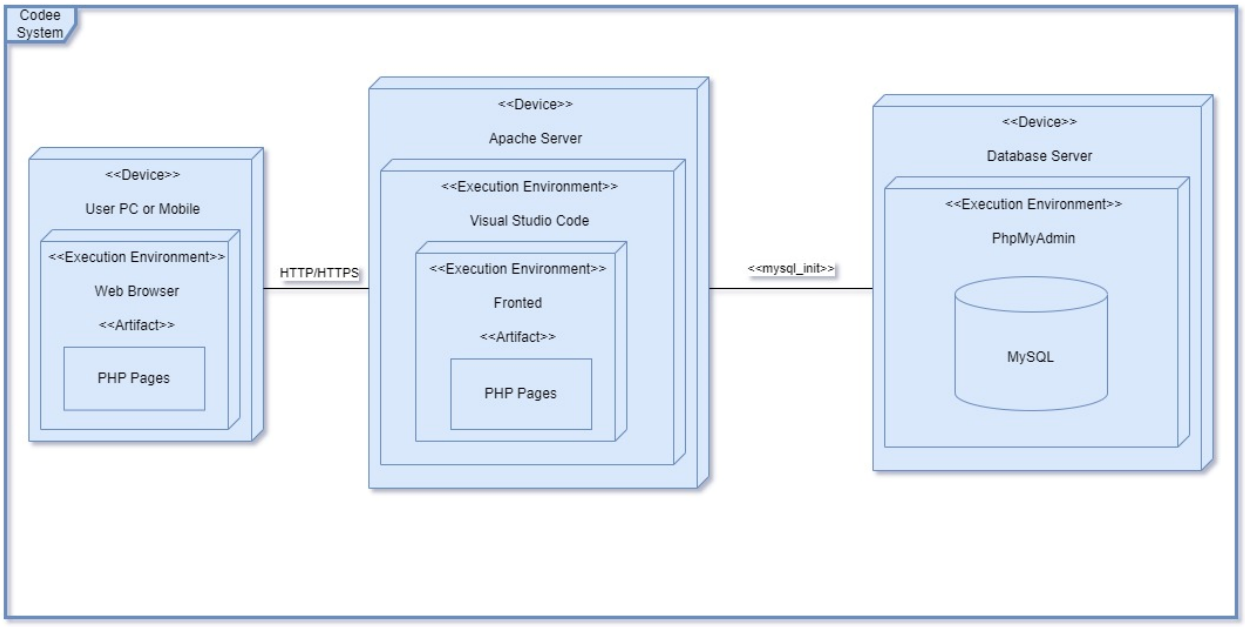
layered pattern.



4.3. Deployment Diagram

Figure 4.2

Deployment diagram.



4.4. User Interfaces Design

Designing the interfaces is preparation for the next stage which is the development phase. Regarding to that we proposed the following interfaces:

- Figure 4.4: Home page.
- Figure 4.5: Register page.
- Figure 4.6: Login page.
- Figure 4.7: User page (catalogue).
- Figure 4.8: Chat page.
- Figure 4.9: Developer page (catalogue).
- Figure 4.10: Development page (IDE & AI bot).
- Figure 4.11: Verification page (for developers & consultants).
- Figure 4.12: Admin page.

Figure 4.3

Home page.

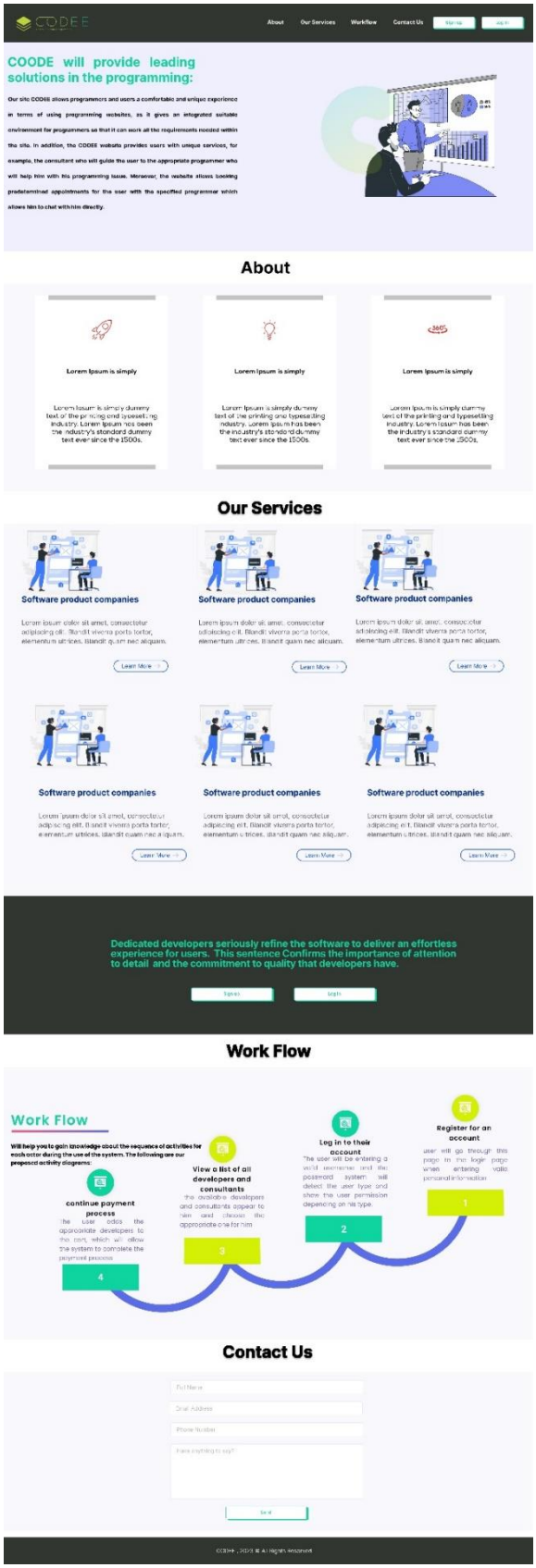
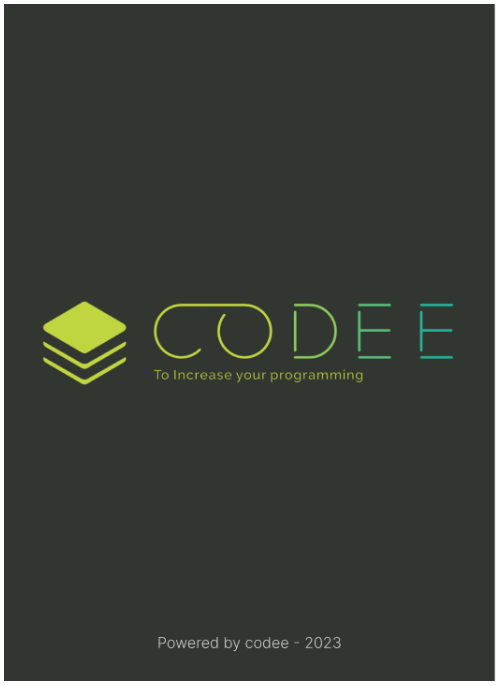


Figure 4.4

Register page.



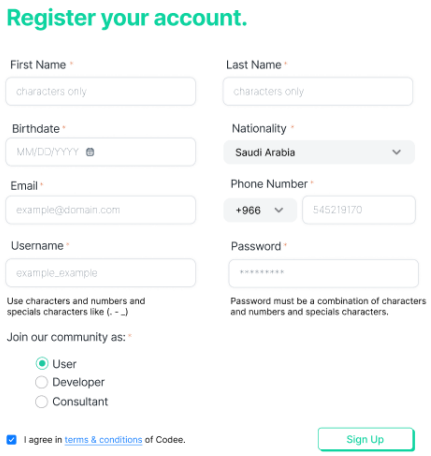
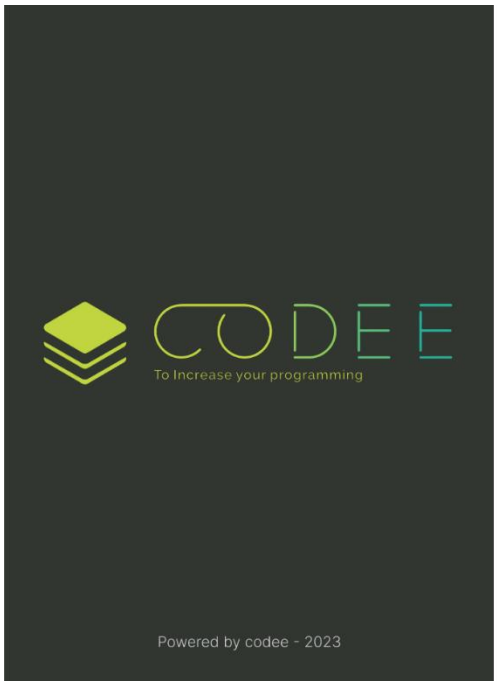


Figure 4.5

Login page.



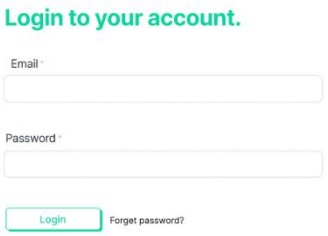


Figure 4.6

User page (catalogue).

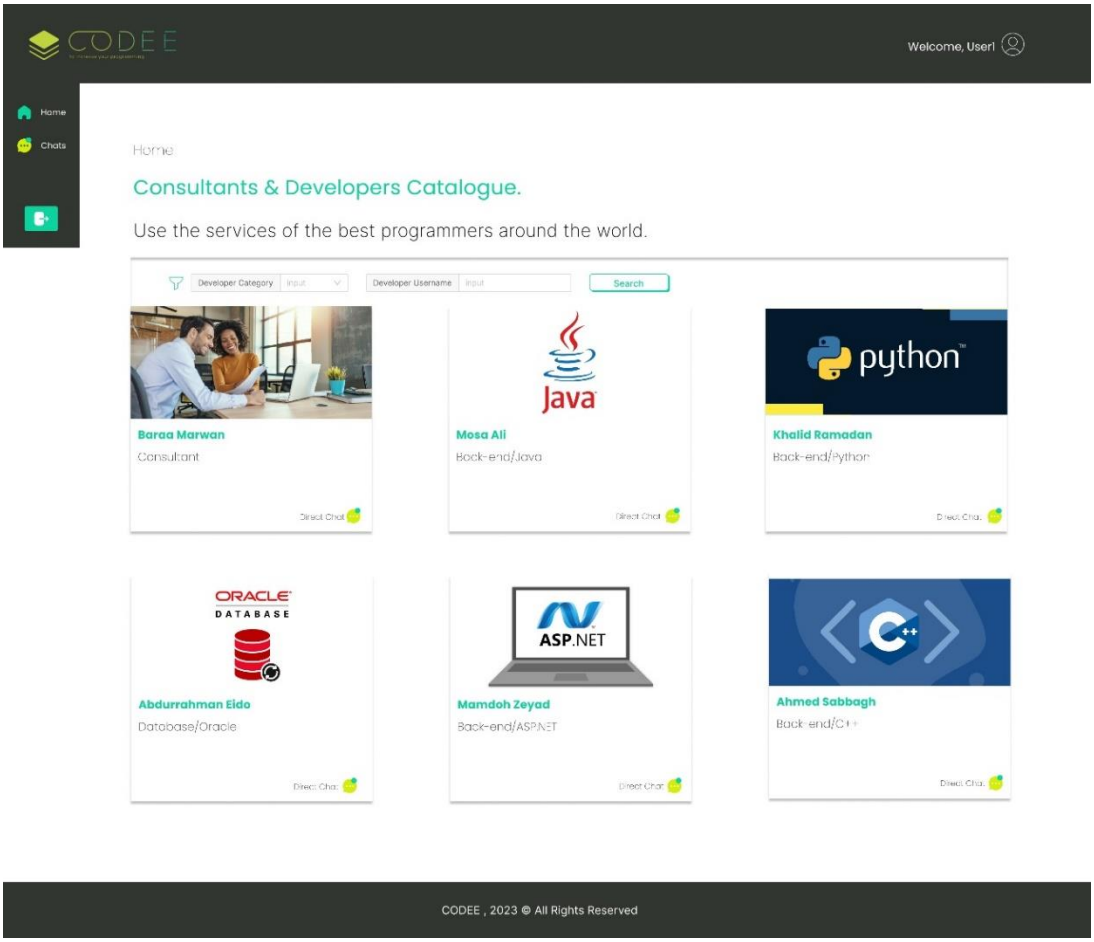


Figure 4.7

Chat page.

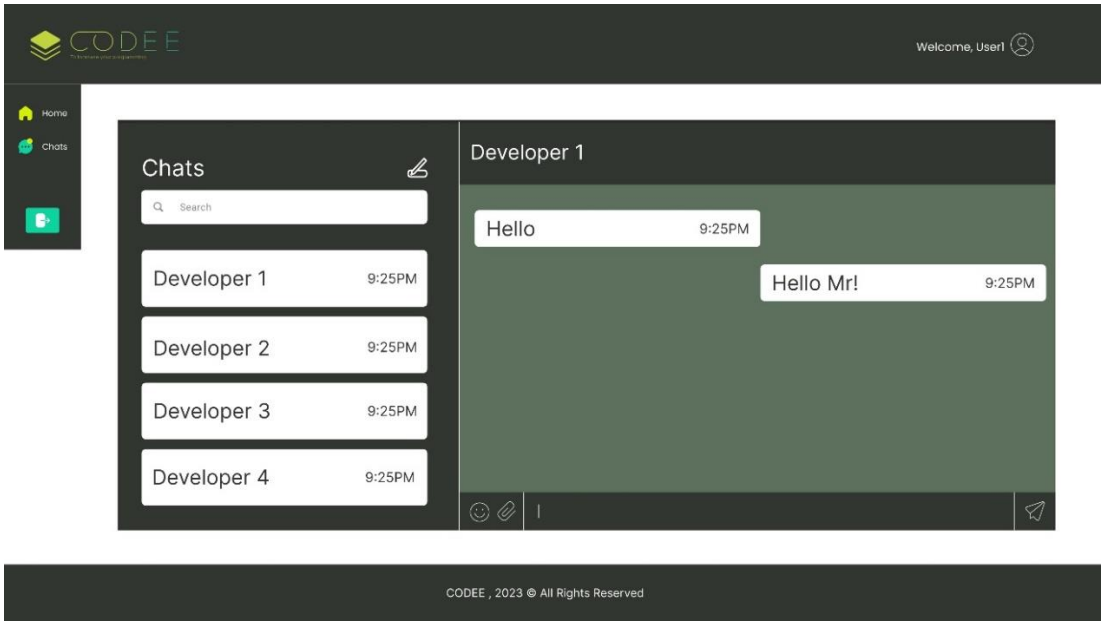


Figure 4.8

Developer page (catalogue).

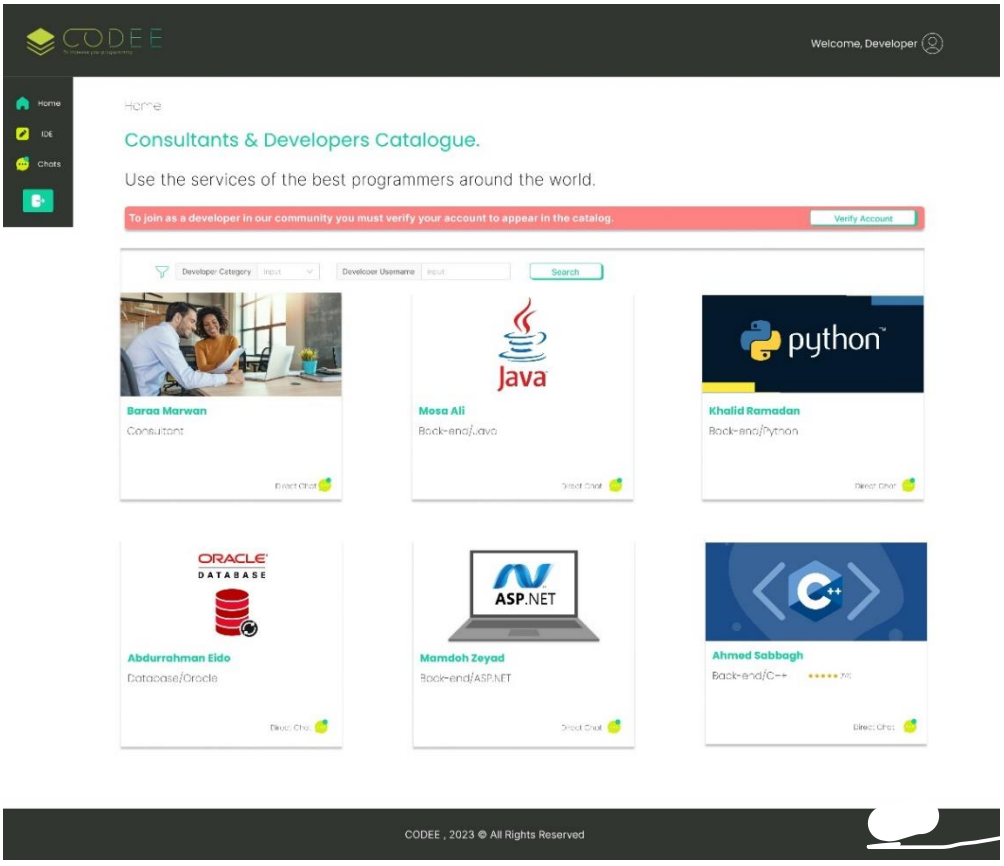


Figure 4.9

Development page (IDE & AI bot).

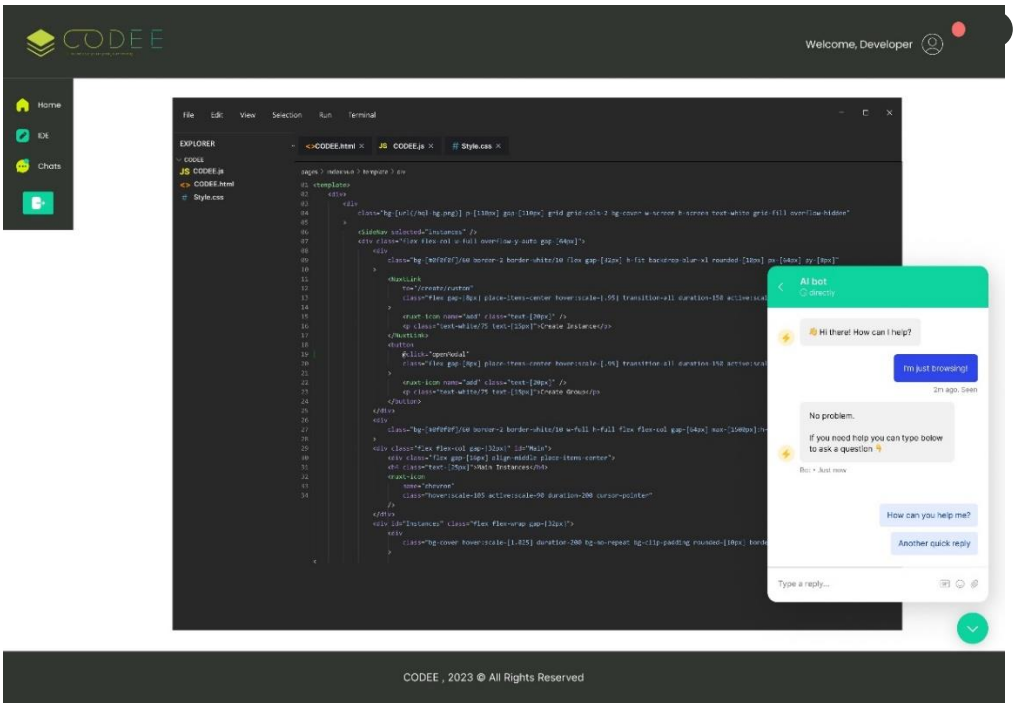


Figure 4.10

Verification page (for developers & consultants).

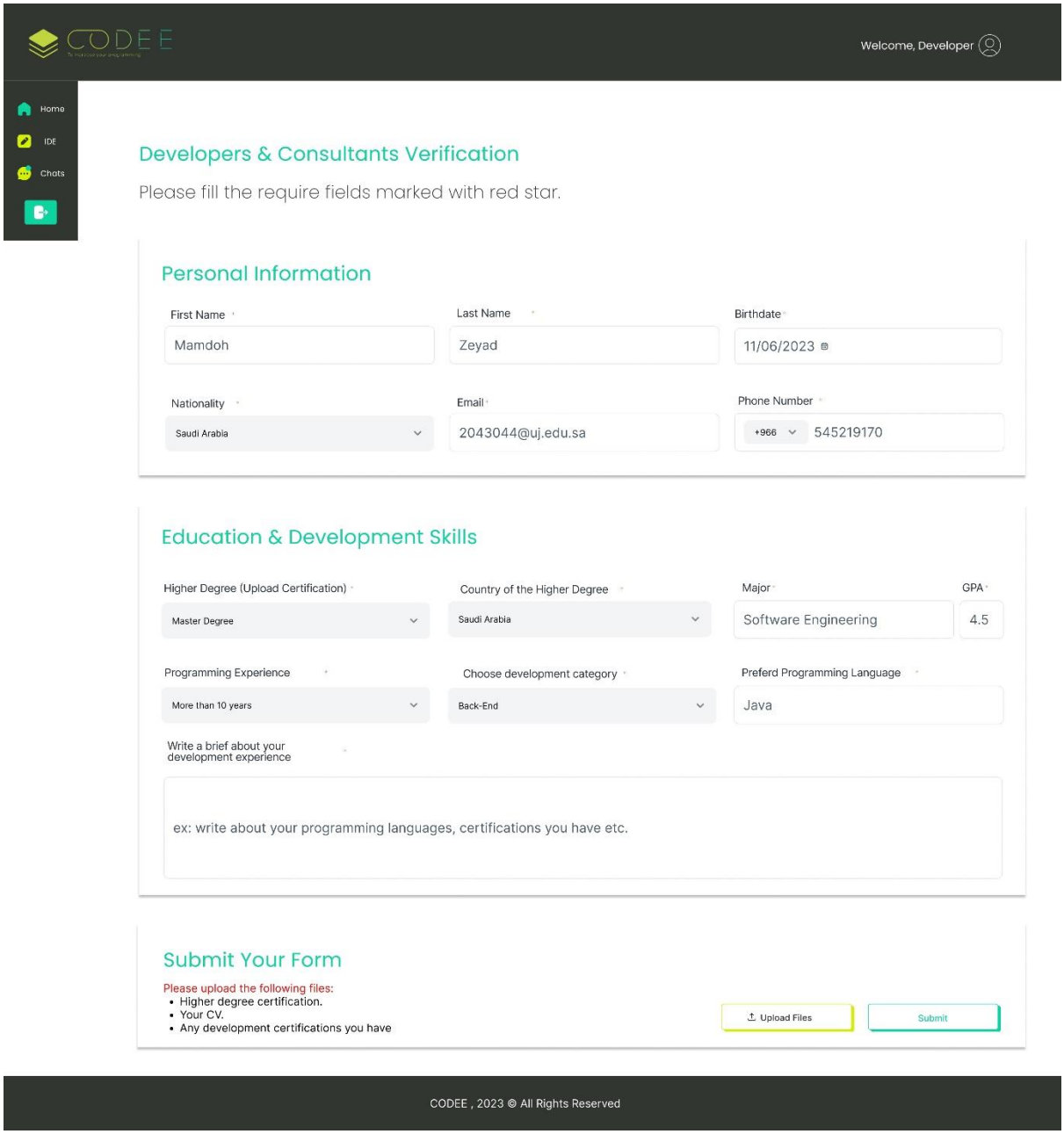
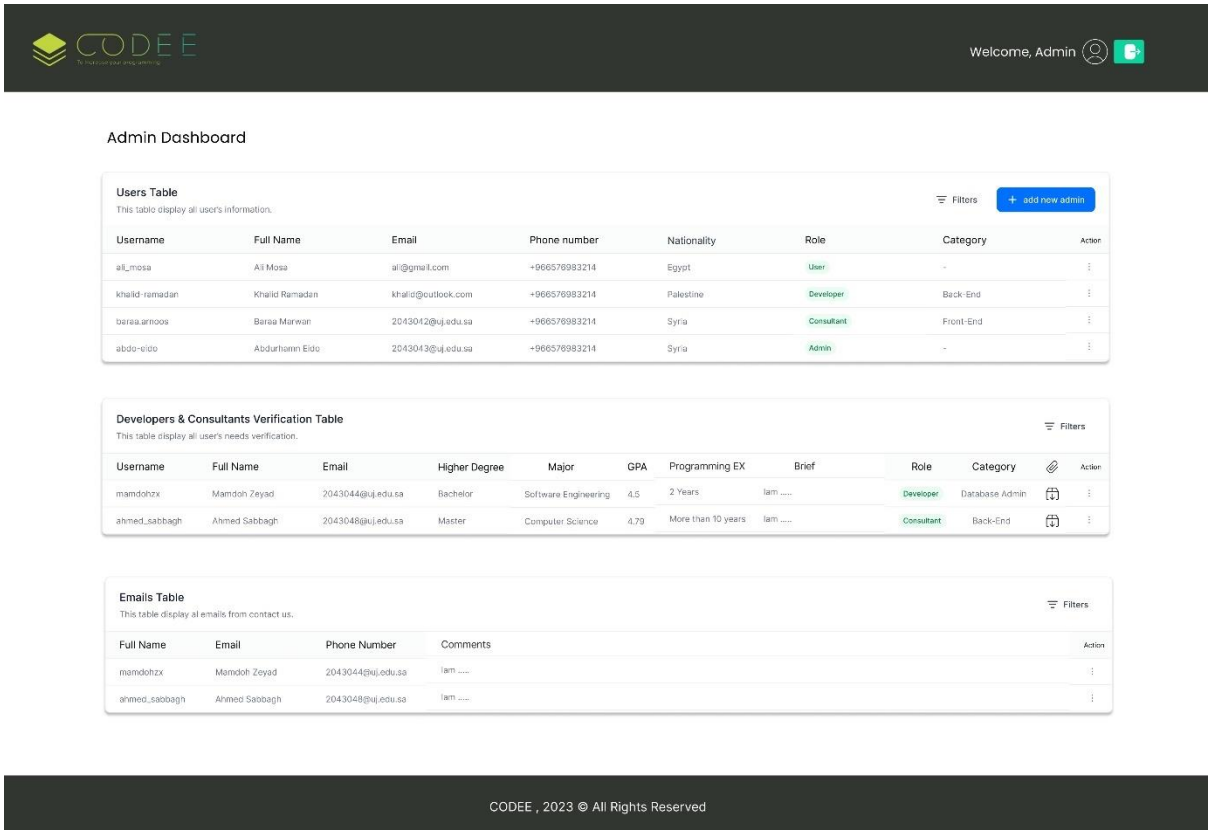


Figure 4.11

Admin page.



4.5. Conclusion

In conclusion, we have laid the foundation stone for the proposed project, and we have a complete vision of the mechanism for construction of this project through knowledge of the main structure of the project and the proposed designs for it. Thus, this will lead us to the implementation phase.

CHAPTER5: IMPLEMENTATION PHASE

5.1. Introduction

In this chapter, we will discuss the development phase of our project using all front-end phases and the main back-end language PHP. Additionally, we will define what we've used in APIs and Data collection instruments and how to complete the project development phase.

5.2. APIs and Plugins

To continue the development of our project and ensure that our codebase meets all the needs of the users and developers, we have provided two APIs within our platform. These APIs actively contribute to enhancing the functionality of the site, making it work more efficiently and conveniently for developers and users:

The first API we have integrated is the OpenAI GPT (ChatGPT) API, a well-known tool that assists developers in solving various issues. This integration allows developers to address any issues they encounter directly within our platform, eliminating the need to visit external websites. It empowers developers to perform all necessary tasks seamlessly within our site.

Additionally, we have included the PHP Mailer API to facilitate email communication. This API is instrumental in sending messages through email, providing a secure method for users to reset their passwords. Additionally, it assists us by sending an email message through the designated email address set by the owner for the new admin inside our web, indicating the assignment and providing details through that communication channel.

By incorporating these APIs, we aim to streamline the development process, offering developers the tools they need within the confines of our platform. This ensures a more efficient and user-friendly experience, eliminating the necessity for developers to navigate external sources for essential functionalities.

5.3. Data Collection Instruments

In the CODEE project, our primary aim is to create a conducive platform that caters to the diverse needs of the programming community, whether they are users or developers. This is achieved by providing essential tools, including an Integrated Development Environment (IDE), an AI- chatbot, and an instant chatting system within our platform.

To implement these features effectively, we conducted interviews with numerous users and developers to gain insights into the challenges they face. Users unanimously expressed difficulties in identifying the type of issues assigned to their projects. In response, we introduced a consultant feature within CODEE to address this common challenge.

The second consensus from user interviews focused on communication issues with developers. Many users reported delays in responses, primarily due to communication via email for example. In response, we implemented an organized and efficient chat system feature within CODEE, facilitating seamless communication between users and developers.

Additionally, interviews with developers revealed a consensus regarding challenges when users approach them with non-developer-related issues. To address this, we incorporated a consultant feature, providing a solution within CODEE.

The second consensus among developers pertained to communication challenges with users, emphasizing delays in responses via email. In response, we introduced an organized chat system feature within CODEE to enhance communication efficiency.

The third consensus highlighted that existing coding platforms often lack sufficient services for developers, such as IDE support. To tackle this issue, we have included IDE services within CODEE, ensuring developers have the necessary tools for issue resolution.

In summary, CODEE is designed based on user and developer feedback, incorporating features such as a consultant, chat system functionality, and IDE support to create a comprehensive and supportive environment for the programming community.

5.4. Implementation

In this section, we will cover the aspects of the implementation process from database implementation to Building the System.

5.4.1. Building the Database

We chose to use local host phpMyAdmin (XAMPP). Using a local host environment has numerous advantages, especially during the development phase of a project. It provides a controlled and isolated environment, enabling you to test and debug your PHP scripts and database queries without affecting the live website or making changes visible to the public. This is particularly useful since it allows you to identify and fix any issues before deploying the changes to the live site.

XAMPP provides a fast and efficient setup for development purposes. It enables us to experiment, make changes, and troubleshoot without any limitations. Moreover, the capability to create and manipulate databases locally ensures data integrity and security throughout the development process. It also allows for easy backups and restores without affecting the live database.

In the context of our project CODEE project, opting for local host PhpMyAdmin (XAMPP) to align with our development goals and requirements. The user-friendly package nature of XAMPP encompasses PHP, MySQL, Apache, and PhpMyAdmin. This integrated package is particularly advantageous for our PHP language-based project, which simplified the installation and configuration steps for us.

We have established three tables within our database:

Figure 5.1:

All tables.

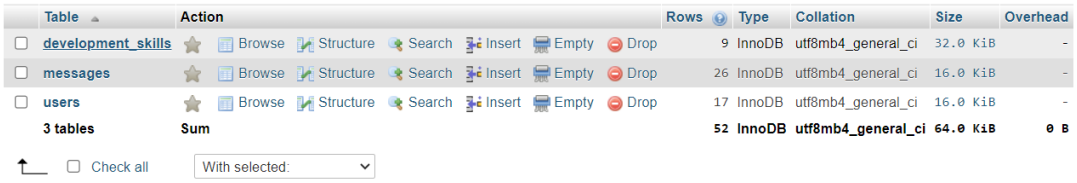
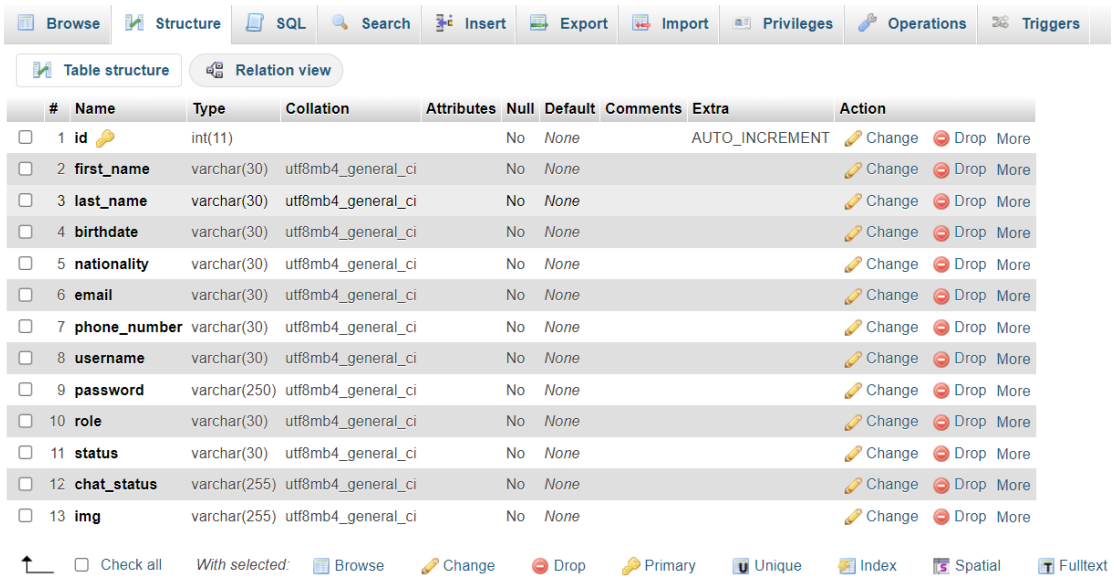


Table	Action	Rows	Type	Collation	Size	Overhead
development_skills	Browse Structure Search Insert Empty Drop	9	InnoDB	utf8mb4_general_ci	32.0 KiB	-
messages	Browse Structure Search Insert Empty Drop	26	InnoDB	utf8mb4_general_ci	16.0 KiB	-
users	Browse Structure Search Insert Empty Drop	17	InnoDB	utf8mb4_general_ci	16.0 KiB	-
3 tables	Sum	52	InnoDB	utf8mb4_general_ci	64.0 KiB	0 B

Check all With selected:

Figure 5.2:

Users table.



Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

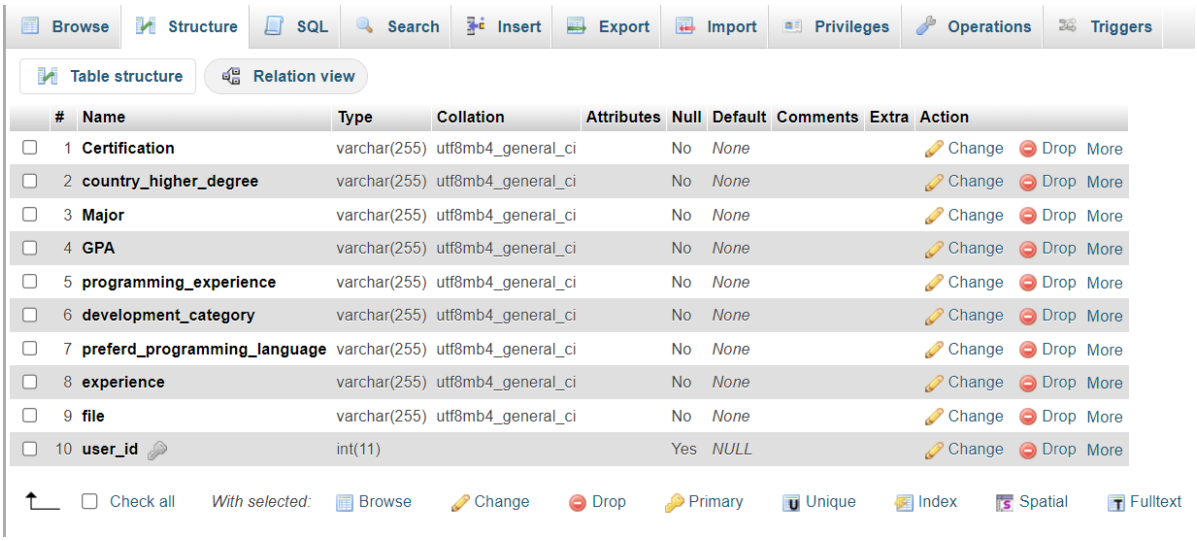
Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	first_name	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
3	last_name	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
4	birthdate	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
5	nationality	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
6	email	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
7	phone_number	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
8	username	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
9	password	varchar(250)	utf8mb4_general_ci		No	None			Change Drop More
10	role	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
11	status	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
12	chat_status	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
13	img	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

Figure 5.3:

development_skills table.



Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

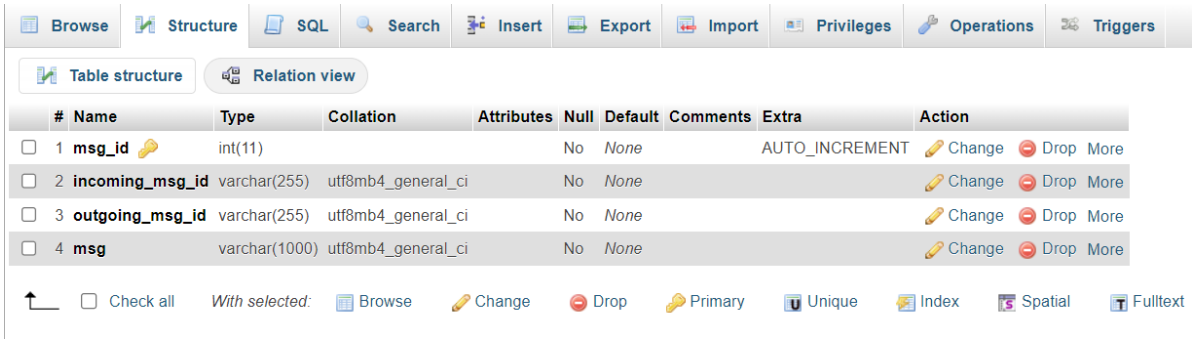
Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Certification	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
2	country_higher_degree	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
3	Major	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
4	GPA	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
5	programming_experience	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
6	development_category	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
7	preferd_programming_language	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
8	experience	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
9	file	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
10	user_id	int(11)			Yes	NULL			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

Figure 5.4:

Messages table.



5.4.2. Building the System.

In the CODEE project, we used PHP as our main programming language and Visual Studio Code as our official Integrated Development Environment (IDE). Visual Studio Code is a powerful code editor that comes with developer tools, providing an enriched development experience for PHP-based web applications. Its features boost productivity and streamline the coding process, making it the perfect choice for building and maintaining PHP projects.

5.4.3. Samples of the Code.

We cannot present all the codes because it is large but here some samples for different functions.

Print all the users in the catalog page.

```
<?php
    include("../includes/mysql_inti.php");
    $rowsPerPage = 6;
    $page = isset($_GET['page']) ? intval($_GET['page']) : 1;
    $offset = ($page - 1) * $rowsPerPage;
    $sql = "SELECT users.first_name, users.last_name, users.img, users.username, users.role,
development_skills.development_category, development_skills.preferd_programming_language
        FROM users INNER JOIN development_skills ON users.id = development_skills.user_id
        WHERE users.status = 'Active' AND users.username != '{$_SESSION['username']}' LIMIT $offset, $rowsPerPage";
    $result = mysqli_query($conn, $sql);
    if (!$result) {
      echo "Error executing query: " . mysqli_error($conn);
      exit;
    }
    while ($row = mysqli_fetch_assoc($result)) {
      echo '<div class="col-md-4 mb-4 ">';
      echo '<div class="card card_shadow animate__animated animate__fadeInUp">';
      echo '<div class="card_header card_shadow"><img class="header_img card_shadow"  src="../chat/php/images/' . $row['img'] . '"
alt=""></div>';
      echo '<div class="card-body text-center">';
```

```
        echo '<h2 class="card-title pt-5 header_caption_blueFont">' . $row['first_name'] . ' ' . $row['last_name'] . '(' . $row['username'] .
')' . '</h2>';
        echo '<p class="card-text">' . '<span class="red_star">' . $row['role'] . '</span>'. ' | ' . $row['development_category'] . ' | ' .
$row['preferd_programming_language'] . '</p>';
        echo '<div class="d-flex align-items-center justify-content-end">';
        echo '<a href="../chat/chat.php?username=' . $row['username'] . '"class="chat_link">Direct Chat <i class="fa-solid fa-comment
my_IconeColor2"></i></a>';
        echo '</div>';
        echo '</div>';
        echo '</div>';
        echo '</div>';
    }
    $conn = mysqli_connect($servername, $username, $password, $dbname);
    $countSql = "SELECT COUNT(*) as total FROM users WHERE status = 'Active'";
    $countResult = mysqli_query($conn, $countSql);
    $countRow = mysqli_fetch_assoc($countResult);
    $totalRows = $countRow['total'];
    mysqli_close($conn);
?>
```

Insert-chat function.

```
<?php
  session_start();
  if(isset($_SESSION['username'])){
    include_once "../../includes/mysql_inti.php";
    $outgoing_username = $_SESSION['username'];
    $incoming_username = mysqli_real_escape_string($conn, $_POST['incoming_username']);
    $message = mysqli_real_escape_string($conn, $_POST['message']);
    if(!empty($message)){
      $sql = "INSERT INTO messages (incoming_msg_id, outgoing_msg_id, msg) VALUES
      ('$incoming_username', '$outgoing_username', '$message')";
      if ($conn->query($sql) === TRUE) {
        echo "All good!";
      } else {
        echo "Error: " . $sql . "<br>" . $conn->error;
      }
    }
  }else{
    header("location: ../login.php");
  }
?>
```

Approve developers and consultants accounts.

```
<?php
  include("../../includes/mysql_inti.php");
  $userId = $_POST['user_id'];
  // Fetch user info from the database
  $sql = "UPDATE users set status = 'Active' WHERE id = $userId";
  $result = mysqli_query($conn, $sql);
  if ($conn->query($sql) === TRUE) {
    ?>
      <script type="text/javascript">
      alert('The user activated successfully. You are redirected to the admin dashboard.');
      window.location = "../../views/admin.php";
      </script>
    <?php
  } else {
    echo "Error: " . $sql . "<br>" . $conn->error;
  }
  mysqli_close($conn);
?>
```

5.5. Conclusion

In this chapter, we discussed the development phase of our project. Additionally, we defined what we've used in APIs, Data collection instruments, and how to complete the project development phase.

CHAPTER6: TESTING PHASE

6.1. Introduction

In this chapter, we will discuss the testing phase of our project by conducting test experiments for the functionalities on our site. We will present test cases and results. Additionally, we will talk about the testing approach and data collection, covering both quantitative and qualitative aspects.

6.2. Testing Approach

We employed a comprehensive testing approach to evaluate the performance and behavior of the software system. First, we used the "Black-Box" testing approach to assess the system's operation and reactions to various inputs. Regarding that, we obtained all the test cases that will be mentioned later in this chapter.

Secondly, we employed the "White-Box" testing approach to verify the correctness of the software's internal logic. Regarding that, we performed Control Flow Graph (CFG) software testing to reveal the hidden pathways of a program's logic. Here are the graphs associated with functions we tested in our project:

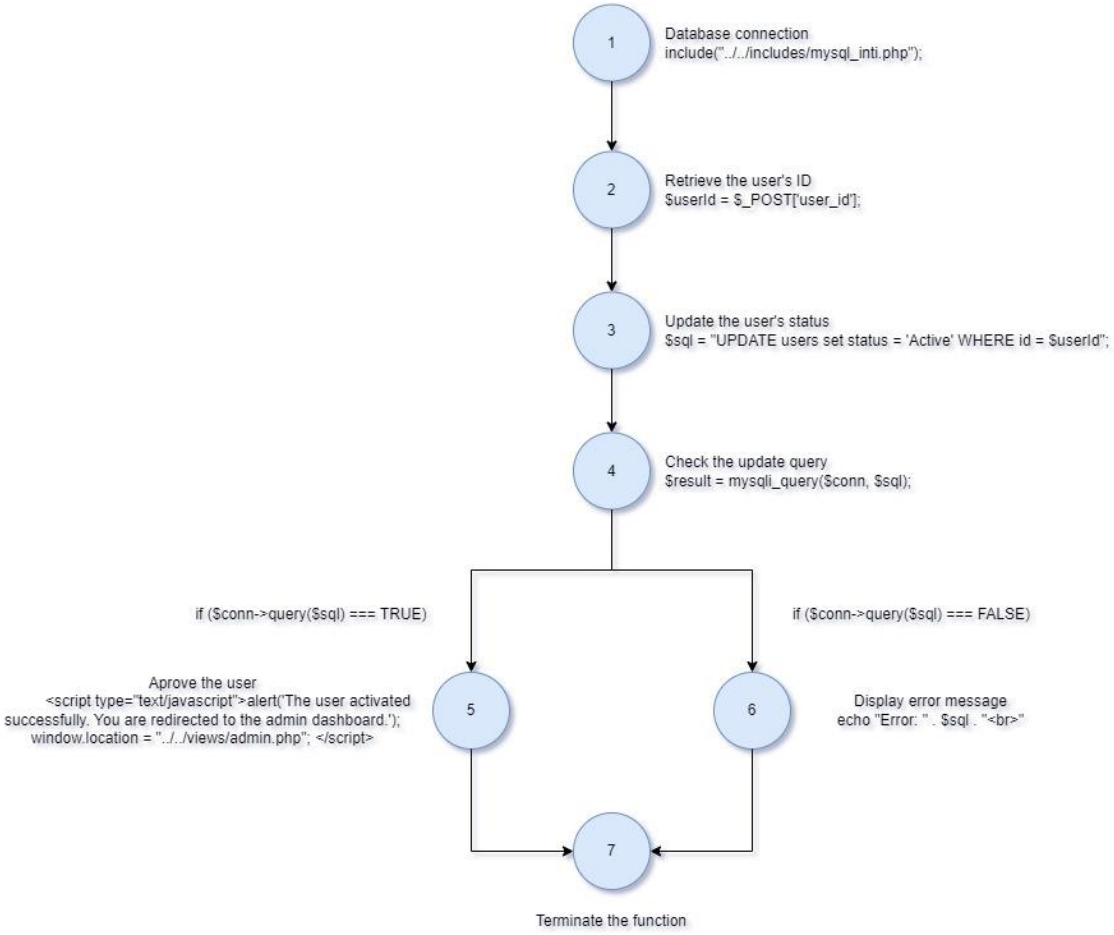
Approve function (code):

```
include("../../includes/mysql_inti.php");
$userId = $_POST['user_id'];
$sql = "UPDATE users set status = 'Active' WHERE id = $userId";
$result = mysqli_query($conn, $sql);
if ($conn->query($sql) === TRUE) {
      ?>
        <script type="text/javascript">
        alert('The user activated successfully. You are redirected to the admin dashboard.');
        window.location = "../../views/admin.php"; </script>
      <?php
} else {
      echo "Error: " . $sql . "<br>" . $conn->error;
}
mysqli_close($conn);
```

Approve function (CFG):

Figure 6.1:

CFG for approve function.



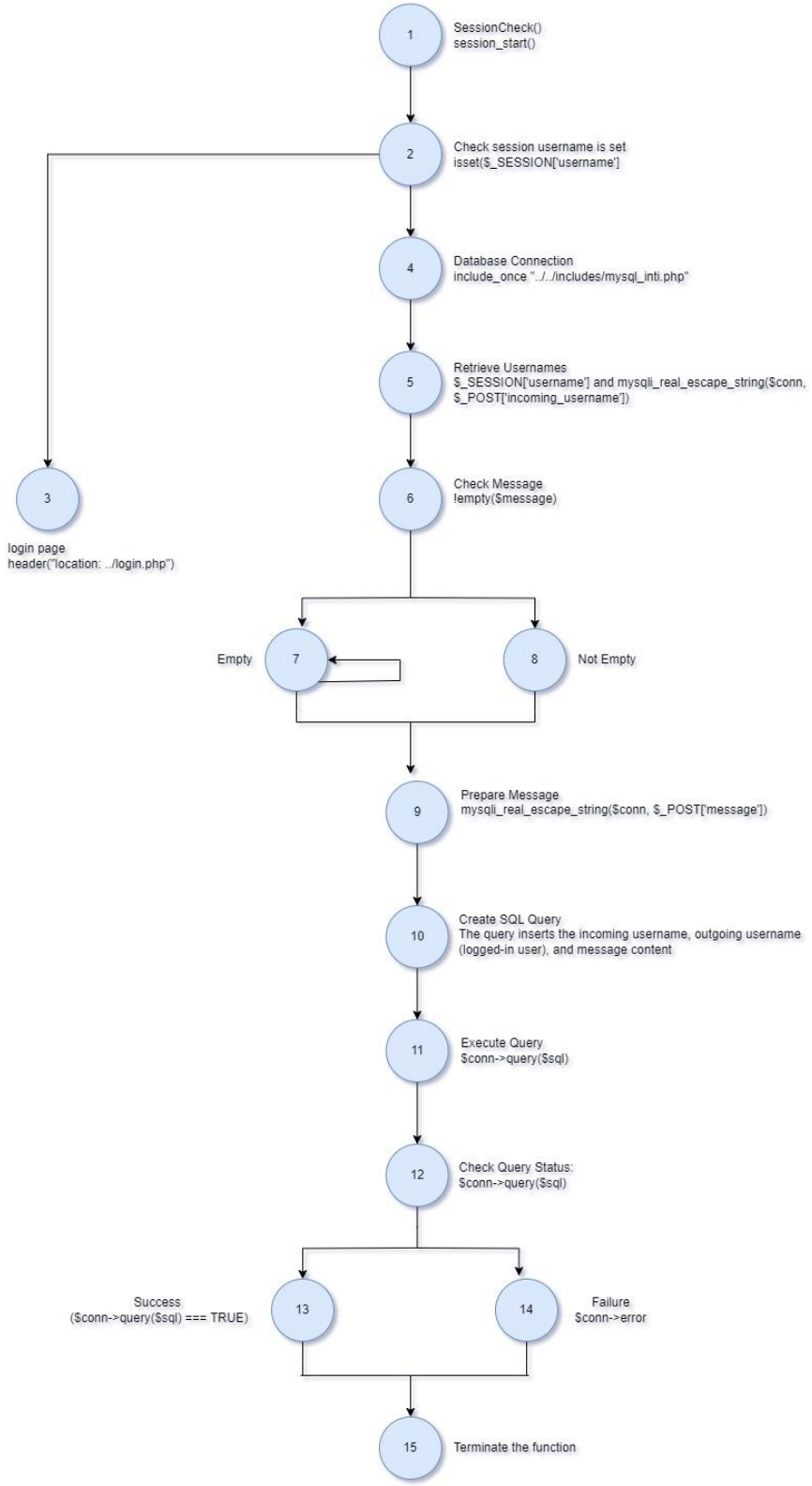
Insert function (Code):

```
session_start();
   if(isset($_SESSION['username'])){
      include_once "../../includes/mysql_inti.php";
      $outgoing_username = $_SESSION['username'];
      $incoming_username = mysqli_real_escape_string($conn, $_POST['incoming_username']);
      $message = mysqli_real_escape_string($conn, $_POST['message']);
      if(!empty($message)){
         $sql = "INSERT INTO messages (incoming_msg_id, outgoing_msg_id, msg) VALUES
         ('$incoming_username', '$outgoing_username', '$message')";
         if ($conn->query($sql) === TRUE) {
            echo "All good!";
         } else {
            echo "Error: " . $sql . "<br>" . $conn->error;
         }
      }
   }else{
      header("location: ../login.php");
   }
```

Insert function (CFG):

Figure 6.2:

CFG for insert function.



6.3. Data Collection (Quantitative, Qualitative)

Through personal interviews with users and developers, we conducted test cases related to various aspects of our project (CODEE), including Sign-up, log-in, catalog page, Chatting User-Developer, and the developers' development area. The test results revealed insights in the following areas:

Sign-up and Log-in:

Both users and developers agreed that the registration and login process on our website is seamless and straightforward. They highlighted that anyone could complete the registration process within a short period without encountering any issues.

Catalog:

Users and developers expressed consensus on the appealing appearance and dynamic functionality of the developers' presentation within the catalog page. The limited display of only six developers on each page, along with their detailed information, was considered significant and beneficial by both parties.

Chatting User-Developer:

Both users and developers confirmed that the communication process within our platform has resolved numerous challenges they faced on other sites, particularly in terms of the smoothness of conversations and the speed of interaction between users and developers.

Developers' Development Area:

Developers praised the development area on our platform, especially the availability of an Integrated Development Environment (IDE). The IDE significantly reduces developers' efforts as it is accessible on our site without the need to navigate elsewhere. Developers appreciated the IDE's

effectiveness in identifying errors across various programming languages. Additionally, the presence of a Chat AI bot played a crucial and effective role in assisting developers with various issues.

The insights gained from interviews and test case conclusions helped identify recurring themes and patterns where both users and developers emphasized the importance of being part of our platform. We express gratitude to all participants who generously dedicated their time to testing our site.

6.4. Test Cases and Results

We will now conduct a test case analysis for all the functionalities present within our website. We will present these test cases and their results, elucidating them through the following tables:

6.4.1. Home Page Testing

Table 6.1

Home page test table.

Test Case No.	Description	Result	Pass/Fail/Other
TC001	The content at the top of the page appears from the bottom to the top when the page is opened or refreshed.	The animation is working correctly and has achieved the intended result.	Pass
TC002	The logo in the middle of the page animates continuously while we are on the page.	The logo positioned in the middle of the page moves up and down continuously.	Pass
TC003	When scrolling the page from top to bottom, the navigation bar remains fixed on the page and its background changes to black.	When scrolling the page, the navigation remains fixed, and its background changes to black.	Pass
TC004	When you click the button on the navigation bar, a menu will appear from the right that contains links to the headings on the page.	A menu appears from the right sidebar, allowing you to navigate to the selected links.	Pass
TC005	When clicking on any link, the page automatically scrolls to the corresponding section.	The transition to the selected link, upon clicking, is executed correctly.	Pass

TC006	If you click on the WhatsApp logo or the contact link, a WhatsApp chat will be opened with Mamdoh.	A new window for WhatsApp has been opened, which enables you to communicate with Mamdoh.	Pass
TC007	If you click on the log-in, the user will be redirected to the log-in page.	The webpage displays a login page, allowing users the option to log in.	Pass
TC008	When scrolling down the page, a "Back to Top" button appears. Upon clicking this button, the page will smoothly scroll back to the top position.	Smooth scrolling to the top of the page.	Pass

6.4.2. Signup Page Testing

Table 6.2

Signup page test table.

Test Case No.	Description	Result	Pass/Fail/Other
TC001	Introducing the complete form with all fields left blank.	Preventing the user from submitting the form.	Pass
TC002	Submitting the form with one of the fields left empty.	Preventing the user from submitting the form.	Pass
TC003	Attempting to enter numbers in the first or last name fields or entering a combination of letters and numbers.	Preventing the user from submitting the form.	Pass
TC004	Attempting to submit an email without the "@" symbol or without ending with a valid domain extension such as .com, .net, etc.	Preventing the user from submitting the form.	Pass
TC005	Attempting to submit an email or a username that is already in use.	Preventing the user from submitting the form.	Pass
TC006	Attempting to enter letters in the phone number field or a combination of numbers and letters.	Preventing the user from submitting the form.	Pass
TC007	Attempting to submit a password that does not meet the specified criteria: "Please choose a password from 8 that contains at least one capital letter and numbers and special characters".	Preventing the user from submitting the form.	Pass
TC008	Rejecting the agreement to the platform's terms and conditions.	Preventing the user from submitting the form.	Pass
TC009	Submitting the request with full compliance with all the conditions.	Confirmation of the completion of the request.	Pass

TC0010	Upon confirming the completion of the request.	Saving the data in the website's database and redirecting the user to the login page.	Pass
TC0011	When clicking on the terms and conditions.	Displaying a model page containing the terms and conditions.	Pass
TC0012	When clicking the "login" button.	Redirecting the user to the login page.	Pass
TC0013	When clicking the animated icon.	Redirecting the user to the home page.	Pass

6.4.3. Login Page Testing

Table 6.3

Login page test table.

Test Case No.	Description	Result	Pass/Fail/Other
TC001	Introducing the complete form with all fields left blank.	Preventing the user from submitting the form.	Pass
TC002	Submitting the form with one of the fields left empty.	Preventing the user from submitting the form.	Pass
TC003	Submitting a valid username but an incorrect password, or vice versa.	Displaying a message indicating that the username or password is incorrect.	Pass
TC004	Entering a valid username and password.	If the user is a "User," "Developer," or "Consultant," they will be redirected to the catalog page. If they are an "Owner" or an "Admin," they will be redirected to the dashboard or the admin page.	Pass
TC005	When selecting the "Forgot Password" link.	Redirecting the user to the reset password page.	Pass
TC006	When clicking the "Sign Up" button.	Redirecting the user to the sign-up page.	Pass
TC007	When clicking the animated icon.	Redirecting the user to the home page.	Pass

6.4.4. Reset Password Page Testing

Table 6.4

Reset password page test table.

Test Case No.	Description	Result	Pass/Fail/Other
TC001	Submitting the form with all fields left blank.	Preventing the user from submitting the form.	Pass
TC002	Submitting the form with an invalid or unavailable email.	Displaying an error indicating that the email is not available or not registered.	Pass
TC003	Submitting the form with a valid and registered email.	Displaying a message indicating that a link has been sent for password recovery.	Pass
TC004	When clicking the "login" button.	Redirecting the user to the login page.	Pass
TC005	When clicking the animated icon.	Redirecting the user to the home page.	Pass

6.4.5. Catalog Page Testing

Table 6.5

Catalog page test table.

Test Case No.	Description	Result	Pass/Fail/Other
TC001	Displaying all the developers and consultants who have verified their accounts for the user.	Displaying information for all developers and consultants.	Pass
TC002	Printing the name and profile picture of the user who logged in at the top of the side bar.	Displaying the name and profile picture.	Pass
TC003	If you click on the "Direct Chat" button for any user, the user will be redirected to the chat page with that specific user.	Redirecting the user to the chat with the selected user upon clicking.	Pass
TC004	Displaying no more than 6 users on the catalog page and utilizing links at the bottom for navigation in case there are more than 6 users.	Printing 6 users and providing pagination links at the bottom of the page.	Pass
TC005	If the logged-in user is a consultant or developer, a message in red will appear prompting them to verify their account.	Displaying a red bar to submit the account verification request.	Pass

TC006	If the request is submitted and not yet approved.	Displaying a yellow bar to indicate that the status is under review.	Pass
TC007	If the request is submitted and approved.	Displaying a green bar to indicate that the status has been approved.	Pass
TC008	If the logged-in user is a developer, the development section will appear in the sidebar for them.	Displaying the development section in the sidebar for the user.	Pass

6.4.6. Development Area Page Testing

Table 6.6

Development area page test table.

Test Case No.	Description	Result	Pass/Fail/Other
TC001	If there is no connection to the ChatGPT IP.	Displaying a red error message indicating that there is an something went wrong.	Pass
TC002	If the storage limit for the ChatGPT IP is exceeded.	Displaying a red error message indicating that there is an something went wrong.	Pass
TC003	If sending any message with an available connection and sufficient space.	The response will be sent through the ChatGPT IP.	Pass
TC004	If the execution file specific to the programming language is not installed and an attempt is made to run the code.	The error message "The system cannot find the path specified" will be displayed.	Pass
TC005	In the case of writing code that contains errors.	Displaying an error message that specifies the error in the code.	Pass
TC006	Writing code without errors and making an attempt to run it.	Displaying the result in the output box.	Pass

6.4.7. Users-Chat Page

Table 6.7

Users-chat page test table.

Test Case No.	Description	Result	Pass/Fail/Other
TC001	Entering the chat page.	Printing the details of the user who logged in, including their status, as well as printing all users, including their usernames and status, excluding admins and owners.	Pass

TC002	If the user is online, there will be a green sign next to their name.	Presence of a green sign next to the user.	Pass
TC003	If attempting to search for a user by their username or the first two or three letters of their name.	Displaying the search results for the full name or all names that contain the first two or three letters entered.	Pass
TC004	If you click on one of the users.	Redirecting the user to the conversation page with the person they clicked on.	Pass

6.4.8. Chat-Chat Page

Table 6.8

Chat-chat page test table.

Test Case No.	Description	Result	Pass/Fail/Other
TC001	Entering the private chat of the selected user.	Printing the details of the selected user and displaying the conversation with a message indicating no messages if it's empty or showing previous messages if there are any.	Pass
TC002	Clicking the back arrow next to the background image.	Returning to the chat page that includes all users.	Pass
TC003	Typing in the writing field and sending the message either by clicking the send button or pressing the enter key.	Sending the message.	Pass

6.4.9. Verification Page Testing

Table 6.9

Verification page test table.

Test Case No.	Description	Result	Pass/Fail/Other
TC001	Displaying user data upon entering the verification page.	Displaying the data without the ability to make edits.	Pass
TC002	Displaying a form containing the required data for account verification.	Displaying the required data for input.	Pass
TC003	Introducing the complete form with all fields left blank.	Preventing the user from submitting the form.	Pass

TC004	Submitting the form with one of the fields left empty.	Preventing the user from submitting the form.	Pass
TC005	If the form is submitted with all the required information without any errors.	Submitting the form and redirecting the user to the login page again.	Pass

6.4.10. Admin Page Testing

Table 6.10

Admin page test table.

Test Case No.	Description	Result	Pass/Fail/Other
TC001	Entering the admin page as an admin or owner.	Printing all available user data in the system.	Pass
TC002	Entering the admin page as an owner.	Displaying a button to create new admins.	Pass
TC003	Click the " Edit User's Information" button.	Displaying a model screen containing the editable information.	Pass
TC004	Introducing the complete edit form with all fields left blank.	Preventing the user from submitting the form.	Pass
TC005	Submitting the edit form with one of the fields left empty.	Preventing the user from submitting the form.	Pass
TC006	Submitting the form with accurate information as required.	Displaying a message indicating that the data has been updated, redirecting the user to the dashboard again, and showing the updated information.	Pass
TC007	Clicking the "Delete User" button.	Displaying a warning message indicating that the user will be deleted.	Pass
TC008	Confirming the deletion of the user.	Displaying a message about deleting the user's data, redirecting to the dashboard, and completely removing the data.	Pass
TC009	Clicking the verification button when the user is already in an inactive state.	Displaying a model screen without data, and when attempting verification, preventing the admin from doing so.	Pass
TC0010	Clicking the verification button when the user is in a pending state.	Displaying a model screen containing verified User data, and when clicking on approve, changing the status to active and redirecting the admin to the dashboard with an updated status.	Pass

6.5. Conclusion

In this chapter, we reviewed the testing phase of our project by conducting test experiments for the functionalities on our site. Additionally, we talked about the testing approach and data collection.

CHAPTER7: CONCLUSIONS

7.1. Conclusion

In conclusion, at CODEE, our goal is to provide a suitable and modern environment for all users and developers by offering a comprehensive set of features and services that users have requested, which are not available on other platforms. We can confidently say that we have successfully achieved the project's goals entirely. We have dedicated our efforts to creating an enjoyable, modern, and user-friendly website for all users. Ultimately, we aspire to be pioneers in this field, not only within our boundaries but globally.

7.2. Future Work

To enhance the website in the future, we believe it's essential to add additional features as follows:

- **Payment System**: implement a payment system within the website, allowing developers to offer their services to users on a paid basis.
- **Rating System**: Introduce a user rating system that enables users to evaluate and provide feedback on the developers. This system can be beneficial for developers by encouraging them to maintain high-quality services.
- **Appointment Booking System**: Integrate a scheduling system that allows users to book appointments with specific developers. This feature would be useful in cases where the developer is busy or unavailable at the current time, enabling users to secure a time slot with the designated developer.

References

Ryan Houma (2020). Agile vs. Waterfall Methodology: Making A Choice. Retrieved from https://apifuse.io/blog/agile-vs-waterfall-methodology/

Stack Overflow (2024, April). https://stackoverflow.com/

Freelancer (2024, April). https://www.freelancer.com/

khamsat (2024, April) https://khamsat.com/

Gitter (2024, April). https://gitter.im/