

Mostqble Chatbot

by

Ghada Khalid Al-wagdani -1912520
Nerdin abdulrahim al-aqeel- 1912112
Suha Aziz al-rajhi-1912005

Computer Science Department

Faculty of Compuing and Information Technology
Jeddah University, SAUDI ARABIA

Supervisor
Dr. Dalal bakheet

(oct 2022)

Abstract

The Chatbot for the University of Jeddah, many students don't know where to go or whom to ask if they had questions related to their college life. This project discusses the development of a chatbot service that can be added to the Mostqbly application to help students answer their inquiries and guide them through a bot without any help from humans. which will make it easier and faster for the students. To this end, artificial intelligence techniques (in particular expert system) will be employed and evaluated. Mostqbly chatbot will analyze the most frequently asked questions to display the sector with the most complaints and questions to improve it

Acknowledgments

we would like to thank the University of Jeddah and the College of Computer and Information Engineering for giving us the opportunity to learn and obtain educational benefit, and special thanks to the evaluation committee for the graduation project and to our supervisor, Dr.Dalal bakheet.

General Introduction

Our project is a chatbot to answer the questions of students and university employees at the University of Jeddah and help them find answers to their inquiries quickly and also help them save effort and time when they want to find an answer

what distinguishes our application from other applications is that our application is smart and uses expert system to answer the users questions

in the next six chapters we will see the structure of mostqibly application from the beginning until the end.

Contents

Abstract	ii
Acknowledgments	iii
General Introduction	1
List of Tables	4
List of Figures	5
1 General Presentation	7
1.1 Introduction	7
1.2 General Context	7
1.3 Problem definition	8
1.4 Aims	8
1.5 Objectives	8
1.6 Proposed Solution	8
1.7 Report Outline	9
1.8 Project plan	10
2 Related work	11
2.1 Introduction	11
2.2 Background	11
2.3 Existing Systems	12
2.4 Comparison	14
2.5 Conclusion	14
3 General Analysis	15
3.1 Introduction	15
3.2 Requirements gathering	15
3.3 Functional requirements	21
3.4 Non-functional requirements	21
3.5 Conclusion	22

4	Sprint 1 Development	23
4.1	Introduction	23
4.2	Use-case diagram	23
4.3	Sprint 1 Static Aspect Design	27
4.4	Sandint 1 Architectural Aspect Design	28
4.5	Sprint 1 Dynamic Aspect Design	29
4.6	Sprint 1 Implementation	30
4.7	Sprint 1 Testing And Evaluation	32
4.8	Conclusion	39
5	Sprint 2 Development	40
5.1	Introduction	40
5.2	Use-case diagram	40
5.3	Static Aspect Design	44
5.4	Sprint 2 Architectural Aspect Design	45
5.5	Sprint 2 Dynamic Aspect Design	46
5.6	Sprint 2 Implementation	48
5.7	Sprint 2 Testing And Evaluation	51
5.8	Conclusion	60
6	Sprint 3 Development	61
6.1	Introduction	61
6.2	Use-case diagram	61
6.3	Sprint 3 Static Aspect Design	65
6.4	Sprint 2 Architectural Aspect Design	67
6.5	Sprint 3 Dynamic Aspect Design	67
6.6	Sprint 3 Implementation	68
6.7	Sprint 3 Testing Evaluation	72
6.8	Conclusion	81
	General Conclusion and Future Work	82
	Bibliography	83

List of Tables

2.1	Comparison table.	14
4.1	Enter chatbot inquires use case table.	24
4.2	select sector use case table.	25
4.3	select questions use case table.	25
4.4	Ask questions use case table.	26
4.5	exit use case table.	26
4.6	integration testing	32
5.1	Enter chatbot inquires use case table.	41
5.2	select sector use case table.	41
5.3	select questions use case table.	42
5.4	Ask questions use case table.	42
5.5	New questions use case table.	42
5.6	exit use case table.	43
5.7	exit use case table.	43
5.8	classes description.	45
5.9	Description	46
5.10	unit Test	51
6.1	Enter chatbot inquires use case table.	62
6.2	select sector use case table.	62
6.3	select questions use case table.	63
6.4	Ask questions use case table.	63
6.5	New questions use case table.	63
6.6	exit use case table.	64
6.7	exit use case table.	64
6.8	classes description.	66
6.9	Description	67
6.10	unit Test	72

List of Figures

1.1	Project plan	10
1.2	Project plan	10
2.1	jazan University	12
2.2	jazan University	12
2.3	jazan University	13
2.4	hayyak account	13
3.1	questionnaire	16
3.2	questionnaire	16
3.3	questionnaire	17
3.4	questionnaire	17
3.5	questionnaire	18
3.6	questionnaire	18
3.7	questionnaire	19
3.8	questionnaire	19
3.9	questionnaire	20
3.10	questionnaire	20
4.1	use-case diagram	24
4.2	Class diagram	27
4.3	ER diagram	28
4.4	Architectural Aspect Design	28
4.5	Sequence diagram	29
4.6	intro frame	30
4.7	sector frame	30
4.8	frequently Questions frame	31
4.9	chatbot frame	31
4.10	beginning frame	33
4.11	beginning frame	34
4.12	option frame	35
4.13	frequently Questions frame	36
4.14	frequently Questions frame	37
4.15	chatbot frame	38

5.1	use-case diagram	41
5.2	Class diagram	44
5.3	Architectural Aspect Design	46
5.4	Sequence diagram	47
5.5	libraries and the chatbot code	48
5.6	server code	48
5.7	frequently asked questions code	49
5.8	call the employee button	49
5.9	call code	50
5.10	database	50
5.11	sector list	52
5.12	Options page	53
5.13	frequent questions page	54
5.14	test application	55
5.15	mostaqbl page	56
5.16	mostaqbl page	57
5.17	mostaqbl page	58
5.18	Ask the employee	59
6.1	use-case diagram	62
6.2	Detailed Class diagram	65
6.3	Architectural Aspect Design	67
6.4	Sequence diagram	68
6.5	trainer code	69
6.6	trainer code	69
6.7	server code	70
6.8	json file	70
6.9	json file	71
6.10	json file	71
6.11	sector list	73
6.12	Options page	74
6.13	frequent questions page	75
6.14	test application	76
6.15	test application	77
6.16	test application	78
6.17	test application	79
6.18	test application	80

Chapter 1

General Presentation

1.1 Introduction

This chapter provides an overview of the proposed feature that we want to add to the Mostqbly application for the University of Jeddah, and this feature is the Mostqbly chatbot, which is a virtual conversation between the student and the bot to help students, by answering their questions and inquiries, using the technique of expert system. Section 1.2 briefly outlines the context of the project, while Section 1.3 identifies the problem behind the proposal for this implementation. The goals and objectives of this project are then defined in Sections 1.4 and 1.5 respectively. Section 1.6 then describes the solution proposed by this project to address the previously identified problem, and Section 1.7 provides an outline for the report. [?].

1.2 General Context

University of Jeddah offers many services to its students, one of these services is the mostaqbly application service which provides students with their study schedules, transcripts, calculating the GPA, the university card and many other services. This project discusses the development of a chatbot service that can be added to Mostqbly to help students answer their inquiries and guide them through a robot without any help from humans. To this end, artificial intelligence techniques (in particular, expert system) will be employed and evaluated. [?].

1.3 Problem definition

Many students face a problem when they search for an answer to their questions or inquiries, there may be no one to help them, or the process of searching for the required answer may be tiring and take a lot of time and effort to walk through the various buildings in the university to answer all the questions. [?].

1.4 Aims

This project aims to develop a chatbot application that can be added in the future to the (Mostqble) app of The University of Jeddah. The chatbot seeks to answer the student's questions quickly and effortlessly.[?].

1.5 Objectives

1. To create a chatbot application that helps students in answering their questions using their phones.
2. To analyze the student's most frequently asked questions to see if there is a problem in a particular sector.
3. To make the chatbot smart enough to answer all students' questions without the need to talk to a human agent.

1.6 Proposed Solution

The chatbot is supposed to be able to help students answer their questions or queries and guide them through a bot without any help from a human unless it's necessary. The main goal is to reduce the time and effort required to search for the answers. The bot will provide all the required information such as the locations of the buildings and will be smart enough to answer any questions related to the university of Jeddah even though it has been asked in a different way rather than the stored questions. Students can do all of this just by select the chatbot icon in the mostqble application by their phones. Then it will lead them to the chat page, which has the options that they can select from it what classification they desire, then the question they want to answer.

1.7 Report Outline

This document presents (Mostqble Chatbot) project phases:

1. Chapter 1 General Presentation:
 - Introduces the background and describes the domain involved in the project.
2. Chapter 2 Related work :
 - Discusses the existing systems and solutions
 - Provides a comparison between the discussion.
3. Chapter 3 General Analysis:
 - Provides a general analysis of the Proposed application.
 - Provides data gathering as well as the functional and nonfunctional requirements.
4. Chapter 4 Sprint 1 Development:
 - Provides a use-case diagram reflecting the basic user functional requirement
 - A presentation of the different hardware components required for the basic functional requirements,
 - Dynamic Aspect Design.
 - Details of the used hardware/software for implementing the basic functional requirements.
 - Description of the different scenarios performed to test the code.
 - Conclusion of the report.

1.8 Project plan

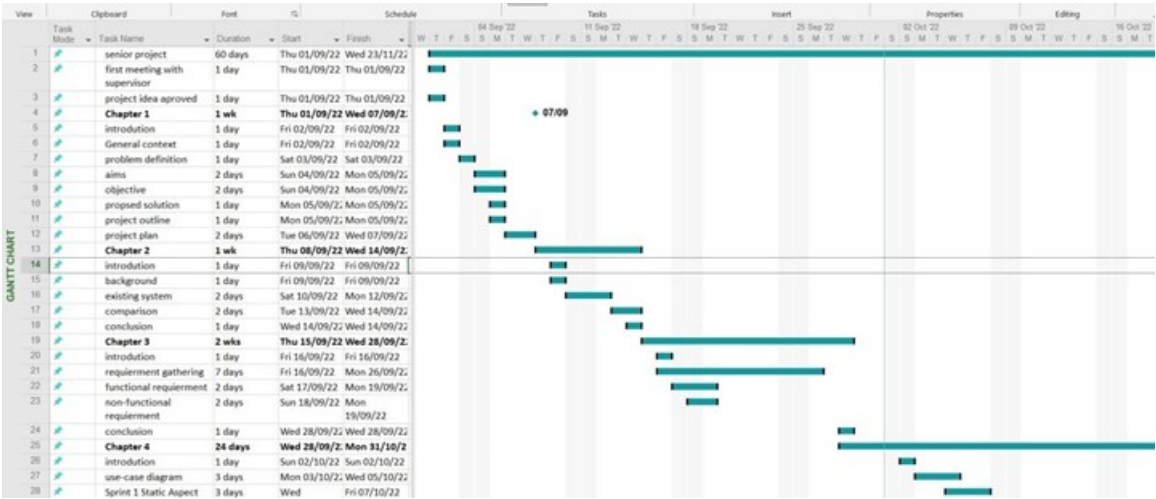


Figure 1.1: Project plan

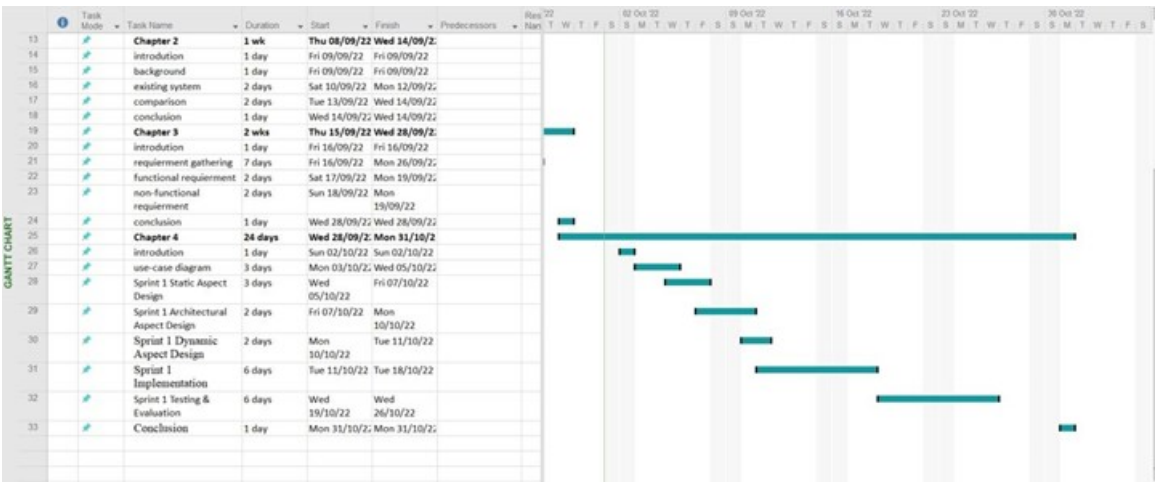


Figure 1.2: Project plan

Chapter 2

Related work

2.1 Introduction

This chapter gives more details about the proposed application. Section 2.2 briefly provides background of the project's domain including chatbot and expert systems technologies. Section 2.3 then identifies existing systems that are trying to solve the same problem as ours. A comparison between the current systems and the suggested application is finally provided in Section 2.4.

2.2 Background

In our project we will use two main technologies chatbot and expert system :

1. Chatbot: is a computer program that simulates and processes human conversation (either written or spoken), allowing humans to interact with digital devices as if they were communicating with a real person. Chatbots can be as simple as rudimentary programs that answer a simple query with a single-line response.[1]
2. An expert system : An expert system is a computer program that uses artificial intelligence (AI) technologies to simulate the judgment and behavior of a human or an organization that has expertise and experience in a particular field.[2]

To develop Mostqble Chatbot we will use this two technologies by creating the chatbot first and using expert system to make the chatbot more intelligent to be able to answer

new questions no matter how different they are from the questions stored in the dataset, and this will helps all fields within the scope of the University of Jeddah which including interaction Students and university staff.

2.3 Existing Systems

1. Jazan University: University of jazan has a feature in their website to help students who have trouble with their schedules, by entering the home page then clicking at the “communicate” icon.[3]



Figure 2.1: jazan University

Then the student selects “communicate contact request” To take them to another page where the students can apply their complaint.



Figure 2.2: jazan University

As you can see in the picture the students can write their complaint or Inquiries about their schedule and press send. then, they receive acceptance or rejection of the complaint.



Figure 2.3: jazan University

2. Screen on campus: Screens at the building of the College of Computing and Information Technology that helps any student or faculty member if they have a question related to the university or thier college in particular. The users of the screen must turn it on and then they can use it by clicking on the chat and start writing their questions.
3. Hayyak: It is a Twitter account set up to respond to questions from Jeddah University students related to admission and registration, and each question is answered by a staff member of the university.[4]

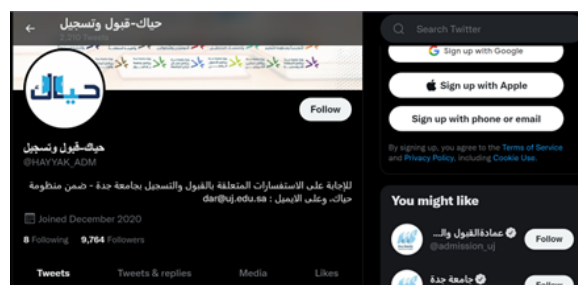


Figure 2.4: hayyak account

2.4 Comparison

	number of target group	how did they present the system?	How many sections are included in the questions?	Accessibility	Responsiveness
Jazan University	university students	university Website	study materials and schedule	-The student can access it by the university website. -students can't get their answer unless it's working hours .	Slow
Screen on campus	university students and faculty members	In the university building	All departments of the university	students can't get their answers unless they come to the university building by themselves	Slow
Hayyak	+10000	chatbot in existing app(twitter)	Admission and Registration Section	Not everyone has Twitter accounts.	Slow
Mostqble Chatbot	university students and faculty members	App	All departments of the university	Anywhere and anytime using the university app.	Quick

Table 2.1: Comparison table.

2.5 Conclusion

This chapter provides the background of Mostqble Chatbot, the techniques that will be used to create it, the scope of our application , and discusses current projects similar to our proposed application. The chapter also presented a comparison between the previous systems and our proposed application. In the next chapter we will continue discussing the details we used in the data gathering process and describing the functional and Non-functional requirements that our application has.

Chapter 3

General Analysis

3.1 Introduction

This chapter provides more details about the general analysis. Section 3.2 The methodology used in the data gathering process, and Data collection results. In Section 3.3 the functional requirements that our application includes is specified , section 3.4 also defines non-functional requirements in our application.

3.2 Requirements gathering

In requirements gathering we did a google form to gather the different opinions from students so we can do our application according to their requirements.

We have received a lot of opinions such as:

- Each service or sector has their own chatbot.
- Adding the camera feature to photograph if there are problems with the university's accessories.
- That the chatbot may not be the only application, but also screens in the colleges.
- Including a feature where students can find their doctor's office number so they can communicate with them.

The most helpful questions were:

- “what is the average time the university takes to respond to your inquiries”
- “how do you search for an answer to your questions”
- “when you get help from the university to answer your questions, how much do you benefit from the university’s answers”

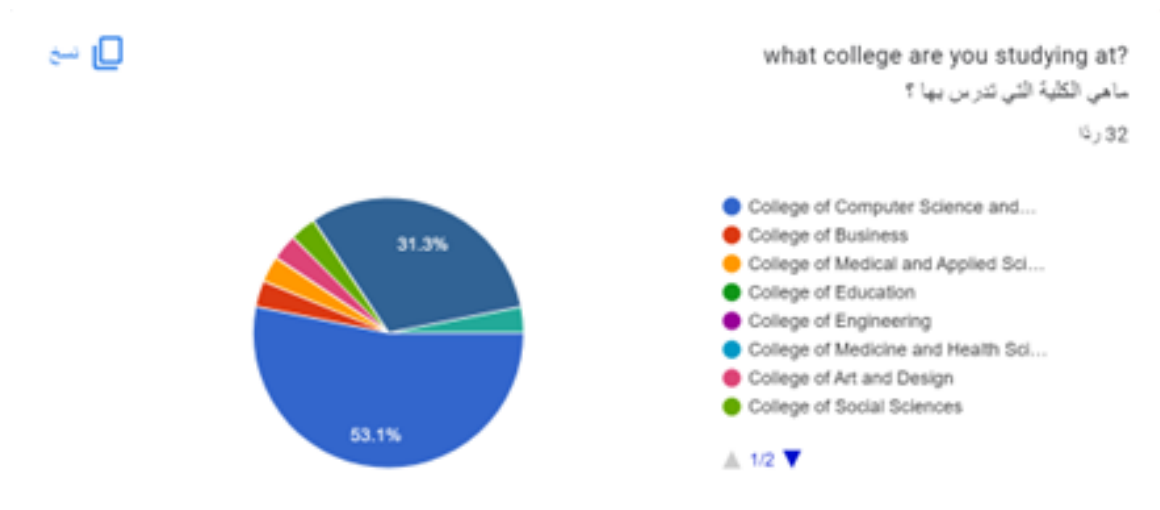


Figure 3.1: questionnaire

The largest percentage of students were in the College of Computer Science and Engineering

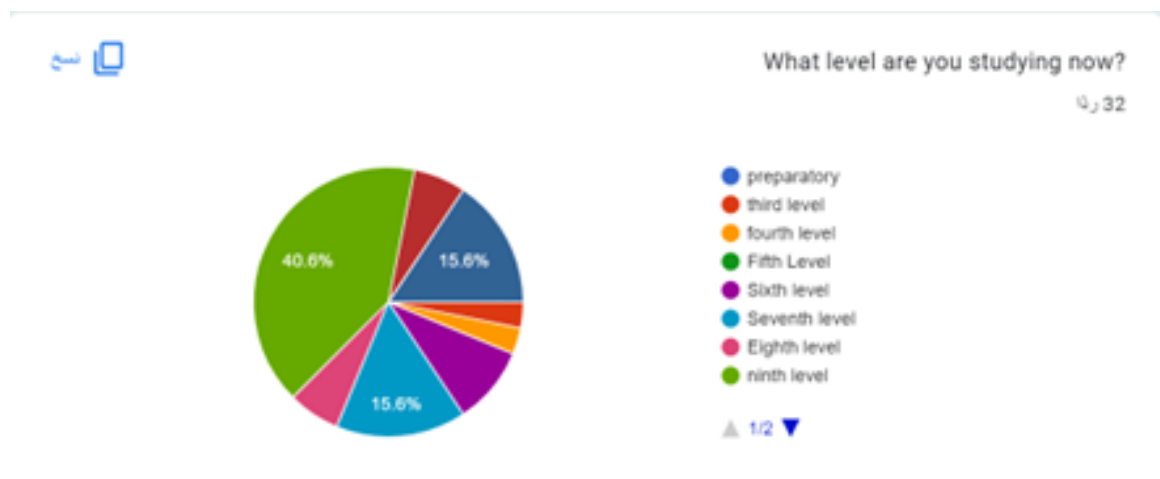


Figure 3.2: questionnaire

The most percentage of students in level nine

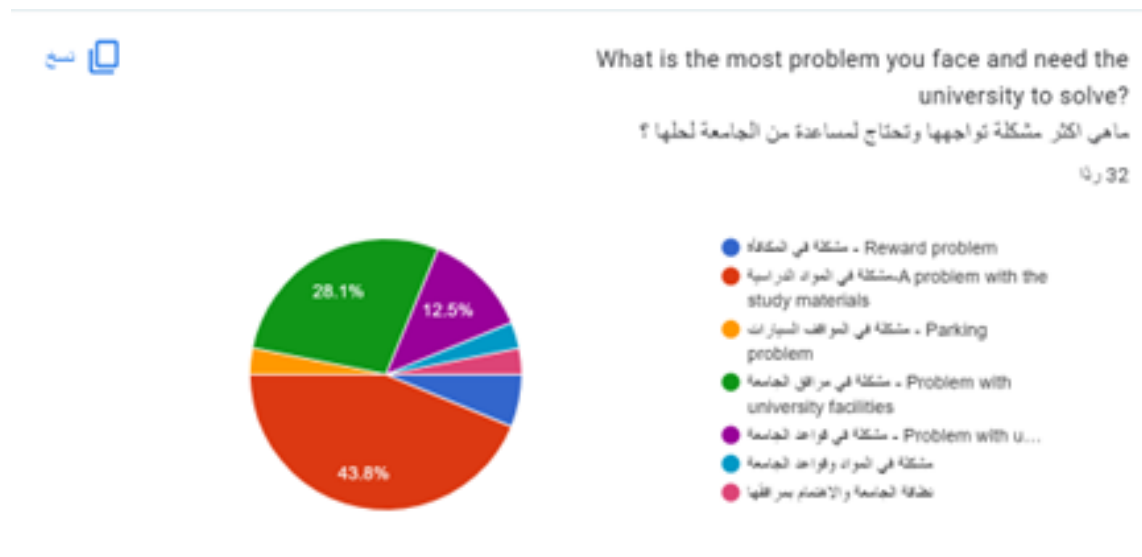


Figure 3.3: questionnaire
The most percentage of students have problem with study materials



Figure 3.4: questionnaire
The Most percentage of students get a response after a day

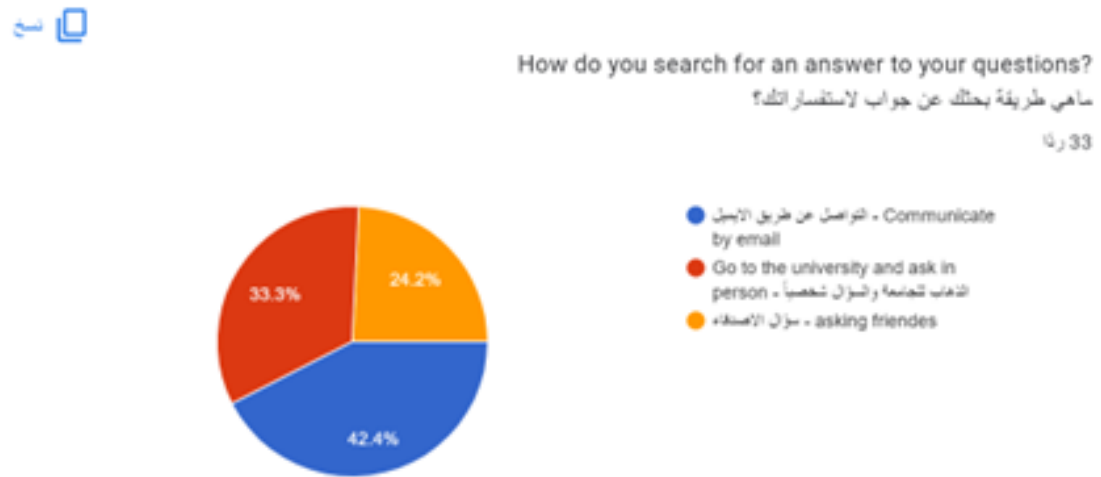


Figure 3.5: questionnaire
The Most percentage of students are communicate by email



Figure 3.6: questionnaire
The largest percentage of students benefit from it

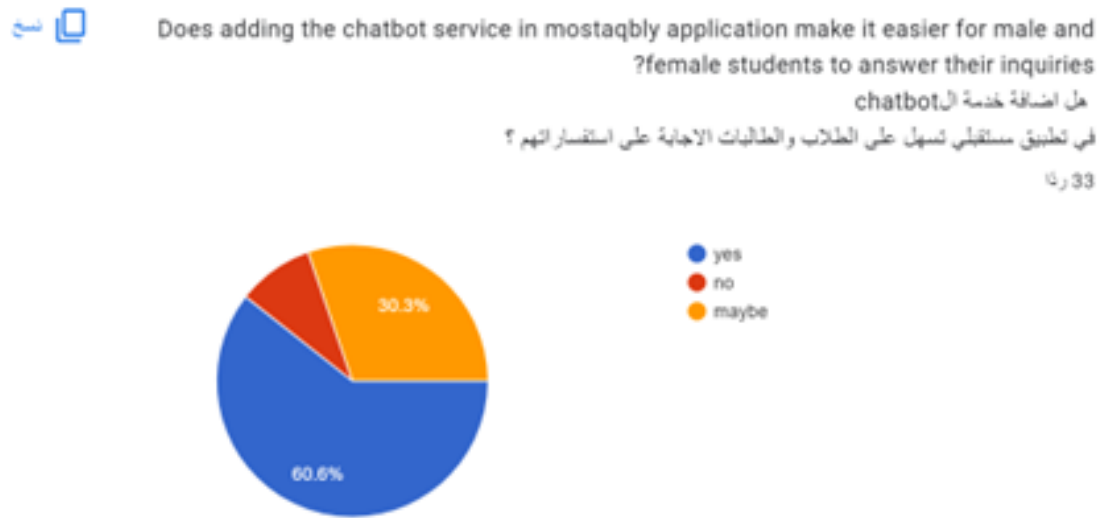


Figure 3.7: questionnaire

The largest percentage of students think that creating a chatbot is a good idea



Figure 3.8: questionnaire

Some of the students' suggestions

السرعه في الاجابه (لا تزيد مدة انتظار الرد الا اكثر من يوم واحد)

وضع الخيارات لمعرفة الاستفسار بأسرع وقت وتسهيل الوصول للخدمة مثل خدمة اراسكس عبر إرسال خيارات وتم الاجابة بالترتيب

ان يكون للخدمة تطبيق او ان تكون الخدمة في الاونديس

اتمنى صراحه انه يكون في تعاون من الجامعة وشخص من العماده جاهز يرد على الاستفسارات ومو يس يقول ما نعرف ما نعرف تنتظر اجابه لان كنا مارج نستفيد من خدمه الشات

خلاله الرد التلقائي للاستفسارات

السرعه في الاستجابه

اقتراح اضافه ارقام عرف الدكتورات

رفع المعاملات بسهوله

Figure 3.9: questionnaire
Some of the students' suggestions

تطويره لتلخص ولحل مشاكل الطلاب بدلا من التواصل على الايميل .. ويكون الرد اسرع

It's going to be very helpful and time saving, and should be self learning so it could learn from the students questions

المواقف و الاستفسارات

سرعه الرد على الاستفسارات

الرد الالى

👍

رفع معاملات او شكوي عن طريقه

يكون الرد في اسرع وقت ممكن

Figure 3.10: questionnaire
Some of the students' suggestions

3.3 Functional requirements

- Create the chatbot interface:
create a simple interface, students can access it by clicking on the icon in mostaqbly application that can lead them to the chatbot immediately (objective1)(Basic-level requirements).
- Create the chatbot application:
The app should have the main classification of the university section (Admission and Registration Section, reward, locker) , and users can select a section of classification , questions ...etc (objective1)(Basic-level requirements).
- Analyzing the student's most frequently asked questions:
We will use a tool for segmenting questions so that we identify the most requested questions from the knowledge base (objective 2)(Intermediate -level requirements).
- When the chatbot couldn't answer the students in a particular question it will allocate the question to a human agent (objective 2)(Intermediate -level requirements).
- Artificial Intelligence Techniques (in particular, expert system), will make the chatbot smart enough to answer any questions related to the university of Jeddah even though it has been asked in a different way rather than the stored questions (objective3)(High-level requirements) .

3.4 Non-functional requirements

- Update: the information will be updated every 10 min.
- Flexibility of the chatbot to serve multiple users at the same time.
- Extensibility: The Extensibility requirements section of the document highlights the need for the system to be scalable and extensible.
- Data access Speed: There needs to be reasonable data access speed on the Client side as well as the staff user side of the application.
- Other questions can be added easily: The system should be able to adapt to future requirements with respect to new question types. If new types of questions become available, then the system should be able to be answered easily.

3.5 Conclusion

This chapter provides the results of the data gathering process and defines the functional and non-functional requirements of the proposed application. In the next chapter we will discuss the development phases, modeling the functional requirements (use-case diagram), other software modeling design, implementation, testing and evaluation.

Chapter 4

Sprint 1 Development

4.1 Introduction

This chapter provides details about the Development phase. Section 4.2 provides the use-case diagram to describe the interactions among the system components. In Section 4.3 Static Aspect Design class diagram (an entity relationship diagram (ERD)) is used to demonstrate structural programming and relationships. Section 4.4 discusses Architectural Aspect Design to identify interaction between its components. Section 4.5 provides Dynamic Aspect Design sequence diagram of chatbot application. Section 4.6 gives Description of the implemented code. Section 4.7 provides Testing And Evaluation Description of the different scenarios performed to test the Sprint1.

4.2 Use-case diagram

Use cases diagram shown in figure 4.1 The use-case is a graphical description of the interactions among the components of a system. The Use-case will clarify what behavior is expected, and the exact methodology for getting there.

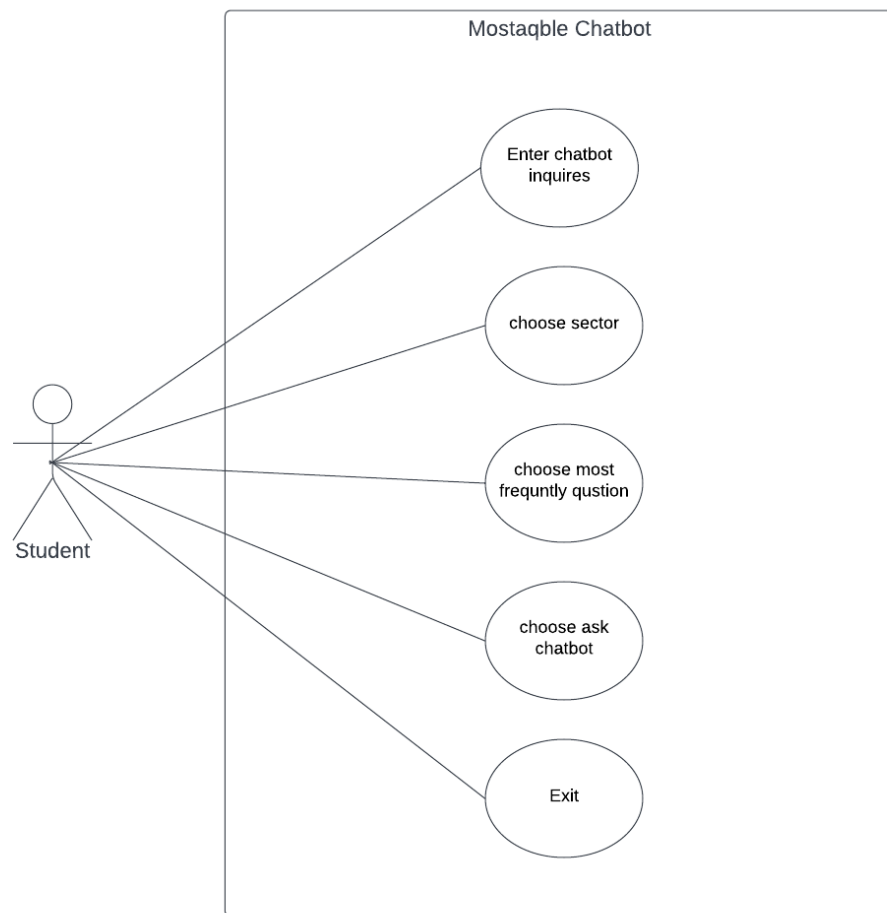


Figure 4.1: use-case diagram

Use case name	Enter chatbot inquires
Use case ID	1
Brief description	Enter to Mostaqble app then click the chatbot inquires icon
Actors	student
Precondition	Download Mostaqble
Flow of activities	<ul style="list-style-type: none"> • The actor downloads a Mostaqble Chatbot add • The actor clicks on chatbot icons

Table 4.1: Enter chatbot inquires use case table.

Use case name	choose sector
Use case ID	2
Brief description	after the user enter the Mostaqble app she/he can select classification to identify the sector of question
Actors	student
Precondition	Have Mostaqble Chatbot
Flow of activities	<ul style="list-style-type: none"> • The actor downloads a Mostaqble Chatbot • The actor clicks on chatbot icons • The actor select the classification of sector

Table 4.2: select sector use case table.

Use case name	choose most frequently question
Use case ID	3
Brief description	select question after selecting classifications of the sector, if the user find the question she/he want
Actors	student
Precondition	select sector
Flow of activities	<ul style="list-style-type: none"> • The actor clicks on chatbot icons • The actor select the classification of sector • The actor select questions from list of most frequently questions

Table 4.3: select questions use case table.

Use case name	choose to ask a different question
Use case ID	4
Brief description	Users can write the question themselves to the chatbot if they did not find it from the list of the most frequently asked questions
Actors	student
Precondition	select classifications for the sector
Flow of activities	<ul style="list-style-type: none"> • The actor clicks on the classification • The actor selects questions from a list of the most frequent questions • when the actor did not find the right question she/he can choose to talk to the chatbot
System respond	<ul style="list-style-type: none"> • The system will display the most frequently asked questions in the selected sector • the system will answer the questions using the expert system

Table 4.4: Ask questions use case table.

Use case name	Exit
Use case ID	5
Brief description	the users can quite and exit whenever she/he wants
Actors	student
Precondition	enter the chatbot
Flow of activities	The actor clicks on exit button
System respond	close the program immediately

Table 4.5: exit use case table.

4.3 Sprint 1 Static Aspect Design

We used class diagram an entity relationship diagram (ERD) to demonstrate structural programming, Detailed Class Diagrams, and the relationships between the tables.

We don't have an obvious attributes in our program that we need to save information in it to put it in a class diagram like the attribute "log-in" more detailed class diagram will be provided in sprint 2.

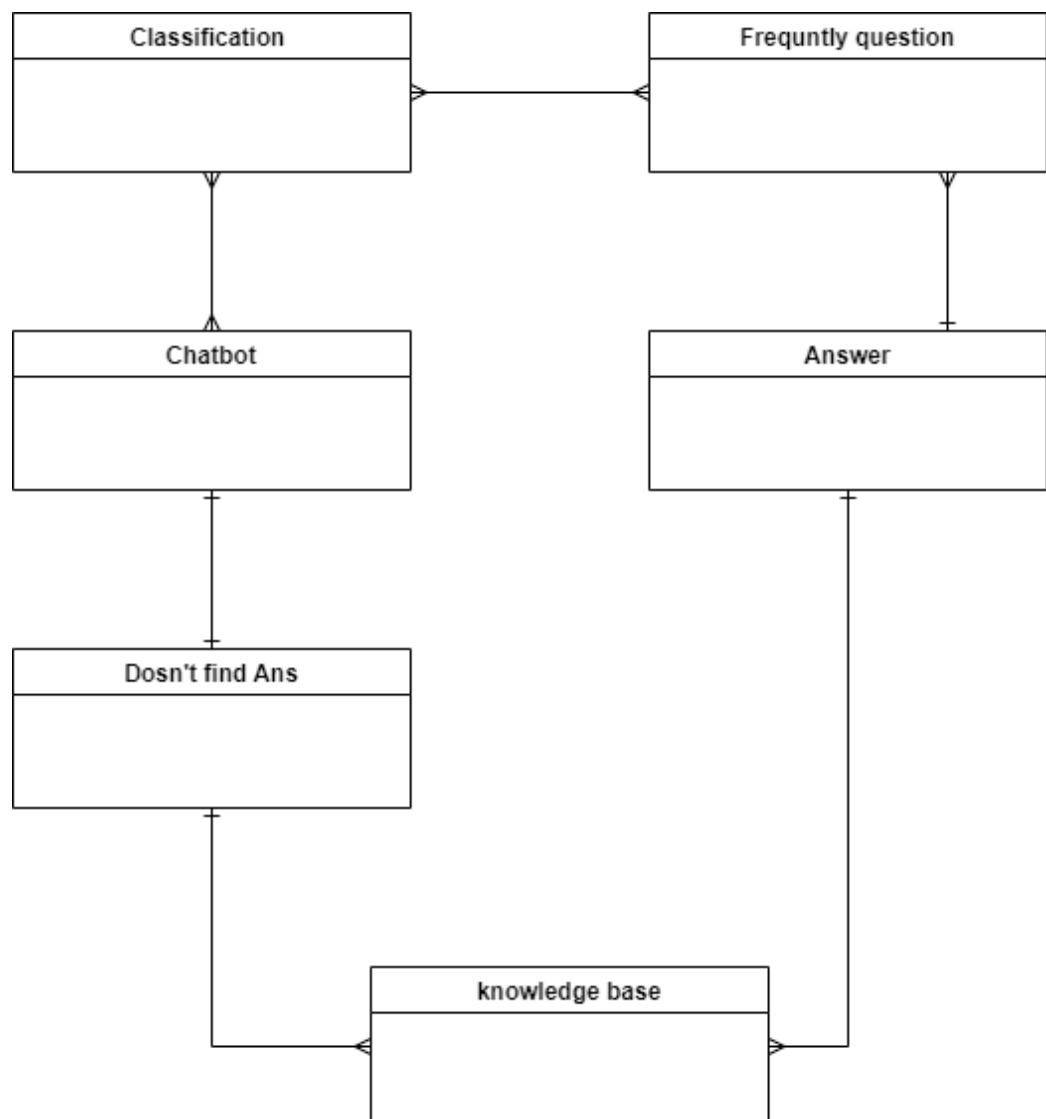


Figure 4.2: Class diagram

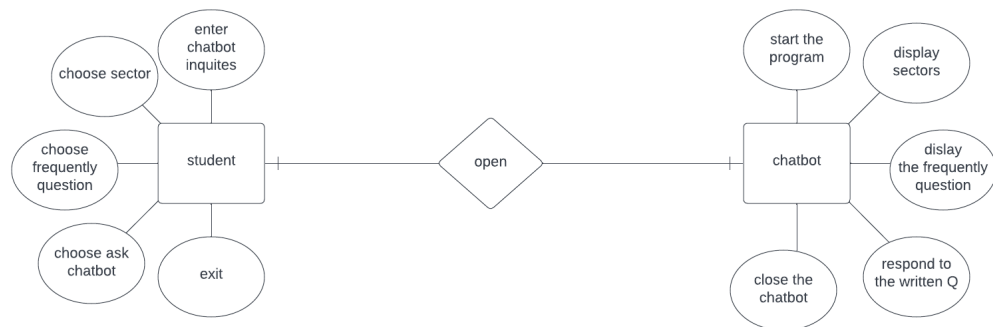


Figure 4.3: ER diagram

4.4 Sandint 1 Architectural Aspect Design

The chatbot application's architecture will define the system's design and structure, as well as the interaction between its components.

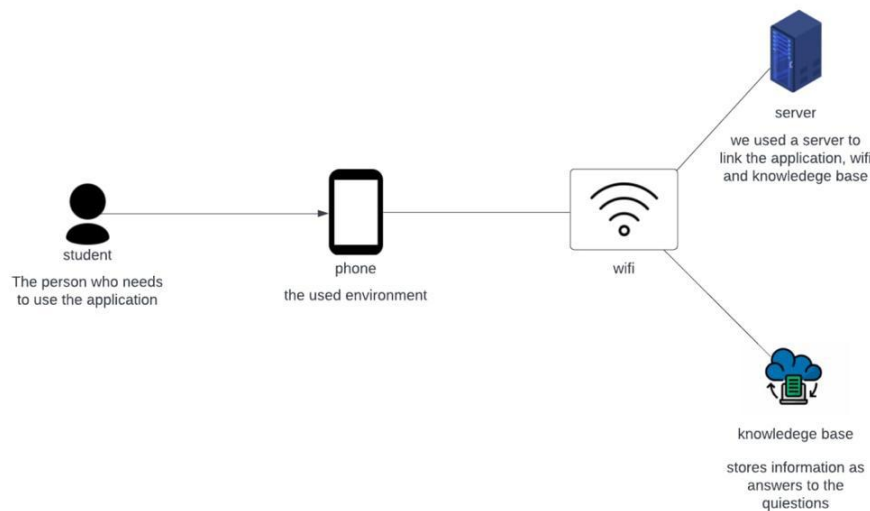


Figure 4.4: Architectural Aspect Design

4.5 Sprint 1 Dynamic Aspect Design

The sequence diagram of chatbot application is shown in figure This graph shows how the chatbot application interacts with users in many situations. Such as, when the user enter to the chatbot. the chatbot will display the sector options, then the user should choose one of sector options to display the most frequently questions and get an answer or choose to ask qusions to the chatbot. In addition, there are many cases that show us how a chatbot application acts in many situations with the user shows in figure 4.5.

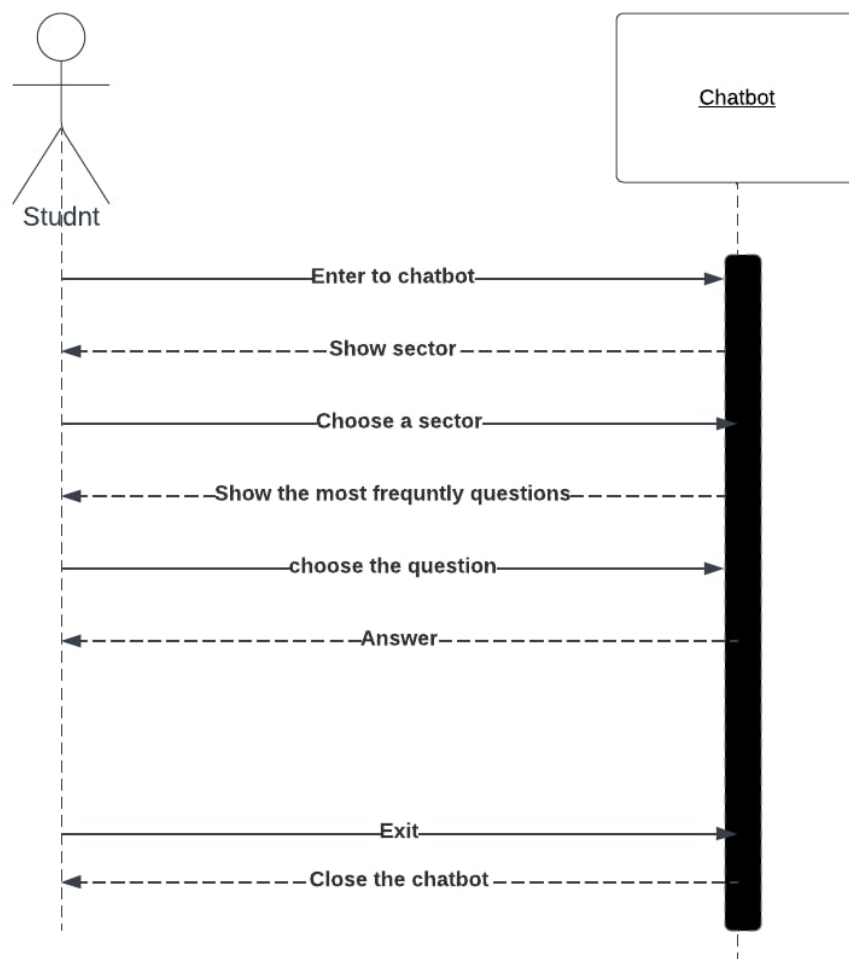


Figure 4.5: Sequence diagram

4.6 Sprint 1 Implementation

In the implementation phase, we have begin with a basic- level requirements which is a creating user interface and the chatbot.

Description of the used software: We used Android Studio and which is an integrated development environment for Google's Android operating system, using XML and Java language for the Interface.

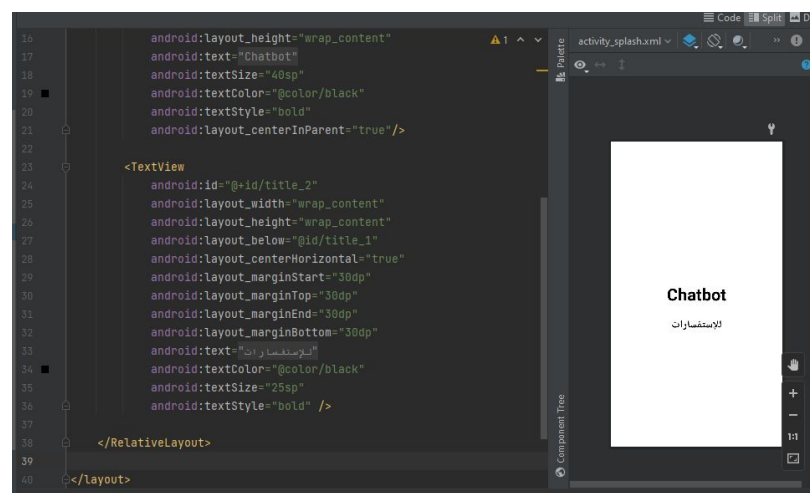


Figure 4.6: intro frame
The first page of our chatbot application

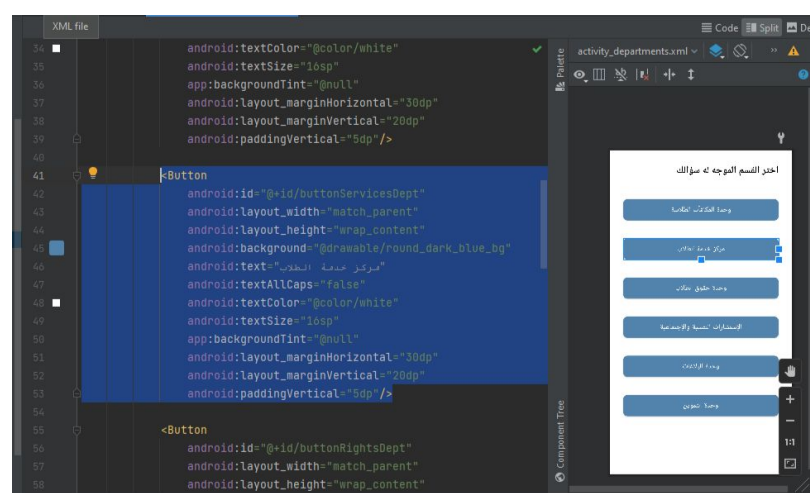


Figure 4.7: sector frame
sector frame that have all sections of classification that chatbots have

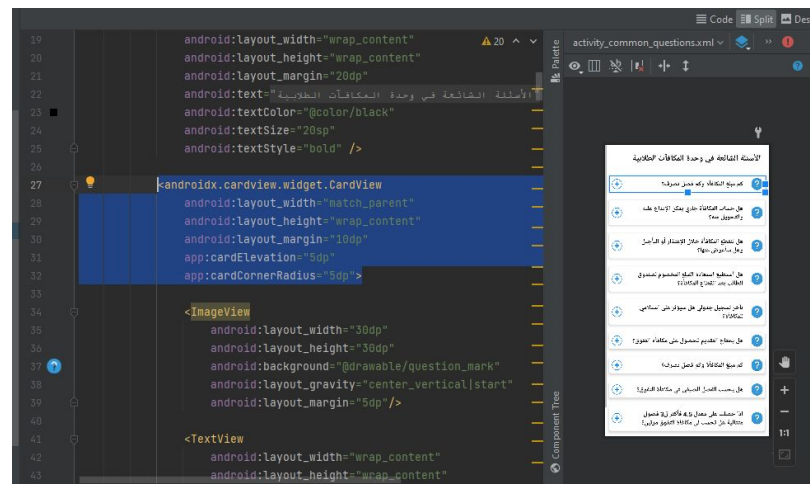


Figure 4.8: frequently Questions frame
frequently questions frame have all frequently questions that asked in the sector which is selected

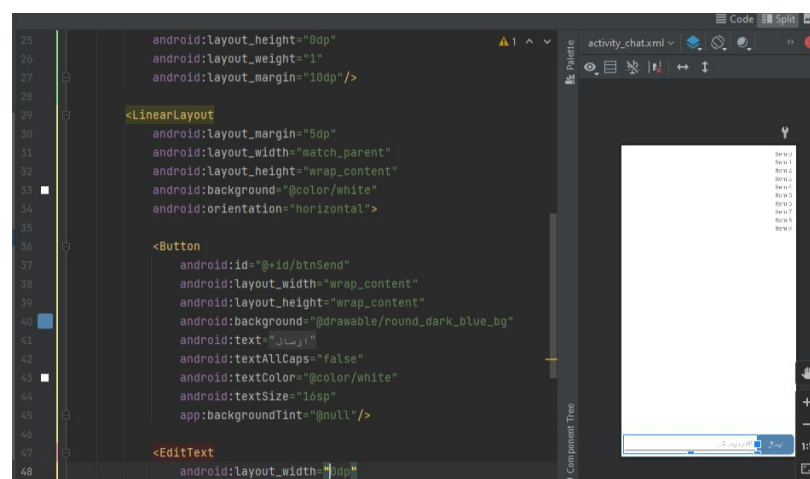


Figure 4.9: chatbot frame
chatbot frame allow the user to ask different questions

4.7 Sprint 1 Testing And Evaluation

Scenarios for unit testing:

For unit testing, it would be difficult to implement it in our software and the presence of an integration testing was sufficient to confirm the test's functionality and reactivity.

Scenarios for Integration Testing:

- When The user enters mostqbly app ↵ view the homepage.
- When the user clicks on the “choose sector” icon ↵ view sectors.
- When the user clicks on the “choose the most frequently asked questions” icon – view the most frequently asked questions.
- When the user clicks on the “ask the chatbot” icon ↵ go to the chatbot page.
- When the user clicks on the “Exit” icon ↵ close the program.

No	Test case	Test case description	Expected result	Result
1	Test the interface	Does the homepage appear when The user enters the mostqbly app?	The homepage will appear when The user enters mostqbly app	PASS
2	Test the functionality of the system	Do the icons display what is it made for?	The icons will display the task depending on the user's choice	PASS
3	Test the functionality of the chatbot	Does the chatbot respond to the user's text?	The chatbot responds to the texts	PASS

Table 4.6: integration testing

we have try using the chatbot in phone .



Chatbot

للإستفسارات



Figure 4.10: beginning frame

beginning frame is the first page that will appear just for two seconds(splash)



Figure 4.11: beginning frame

Second page will appear to the users to allow them to select the required sector for their query



Figure 4.12: option frame

Third page will appear to the users after selecting the sector, and have two options :

1. frequently questions
2. ask chatbot



Figure 4.13: frequently Questions frame
fourth page will appear to the users if he/she select frequently questions to enable them to read this questions with out ask the chatbot



Figure 4.14: frequently Questions frame
fourth page will appear to the users if he/she select frequently questions to allow them to find the answer with out ask the chatbot

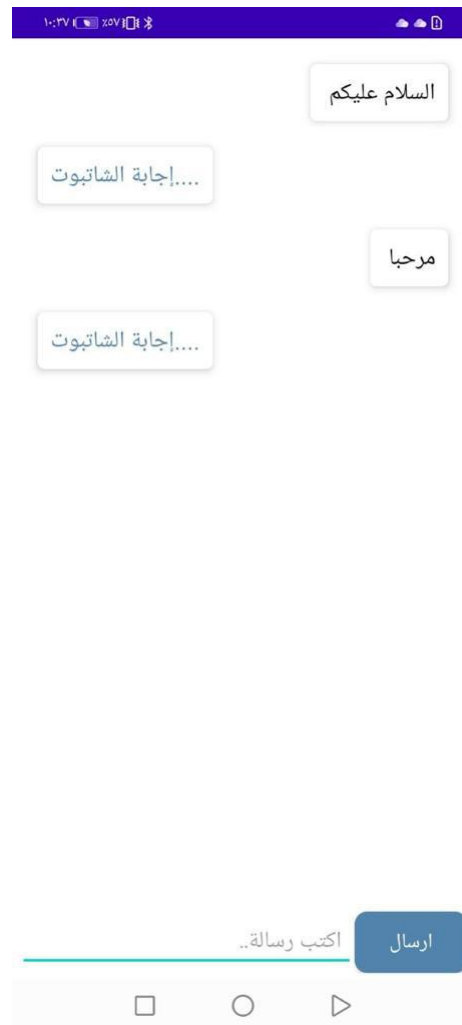


Figure 4.15: chatbot frame

last page will appear to the users if he/she select ask chatbot to allow them enter a conversation with the catbot to ask questions, and the chatbot will answer the questions every time with the same answer in this basic phase.

4.8 Conclusion

This chapter provided the development phase which is the last phase of this semester. it contains only the basic level requirements which is described by a use case diagram, Static Aspect Design, Architectural Aspect Design, Dynamic Aspect Design, Implementation, Testing, and Evaluation. In the next chapters, we will be covering the intermediate-level requirements and the high-level requirements.

Chapter 5

Sprint 2 Development

5.1 Introduction

In This chapter, we will continue to provide details about the second Development phases reflecting the INTERMEDIATE functional requirements. Section 5.2 provides the use-case diagram to describe the interactions among the system components In Section 5.3 Static Aspect Design class diagram (an entity relationship diagram (ERD)) is used to demonstrate structural programming and relationships. Section 5.4 discusses Architectural Aspect Design to identify the interaction between its components. Section 5.5 provides a Dynamic Aspect Design sequence diagram of the chatbot application. Section 5.6 describes the implemented code. Section 5.7 provides a Testing And Evaluation Description of the different scenarios performed to test Sprint 2.

5.2 Use-case diagram

Use cases diagram shown in figure 5.1 The use-case is a graphical description of the interactions among the components of a system. The Use-case will clarify what behavior is expected, and the exact methodology for getting there.

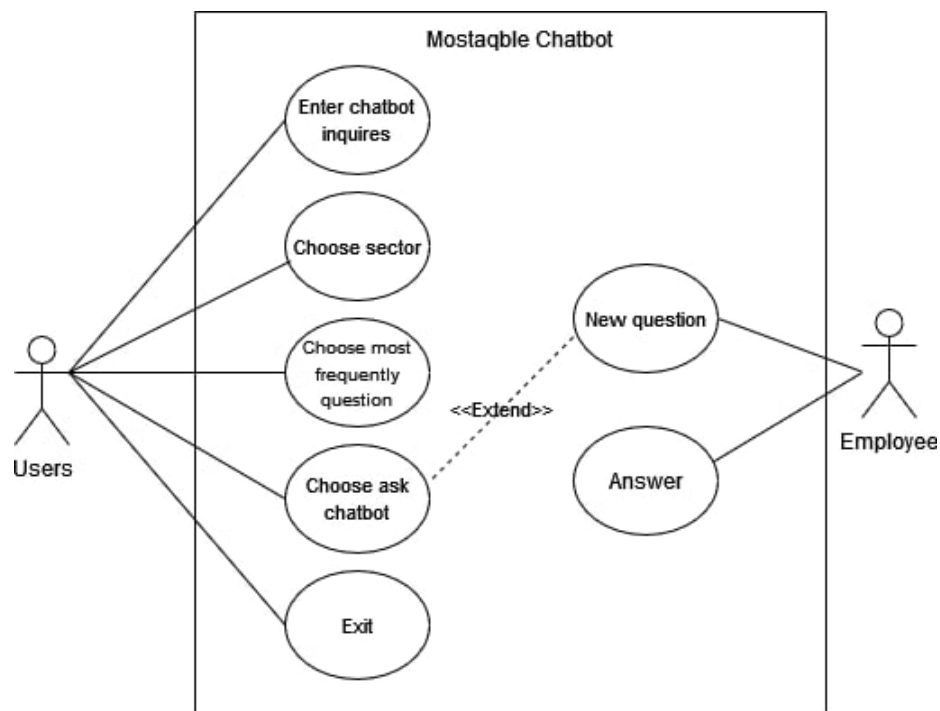


Figure 5.1: use-case diagram

Use case name	Enter chatbot inquires
Use case ID	1
Brief description	Enter to Mostaqble app then click the chatbot inquires icon
Actors	users
Precondition	Download Mostaqble
Flow of activities	<ul style="list-style-type: none"> The actor clicks on chatbot icons

Table 5.1: Enter chatbot inquires use case table.

Use case name	choose sector
Use case ID	2
Brief description	after the user enter the Mostaqble app she/he can select classification to identify the sector of question
Actors	users
Precondition	Have Mostaqble Chatbot
Flow of activities	<ul style="list-style-type: none"> The actor clicks on chatbot icons The actor select the classification of sector

Table 5.2: select sector use case table.

Use case name	choose most frequently question
Use case ID	3
Brief description	select a question after selecting classifications of the sector, if the user finds the question she/he wanted to ask.
Actors	users
Precondition	select sector
Flow of activities	<ul style="list-style-type: none"> • The actor select the classification of sector • The actor select questions from list of most frequently questions

Table 5.3: select questions use case table.

Use case name	choose to ask a different question
Use case ID	4
Brief description	Users can write the question themselves to the chatbot if they did not find it in the list of the most frequently asked questions
Actors	users
Precondition	select classifications for the sector
Flow of activities	<ul style="list-style-type: none"> • The actor clicks on the classification • The actor selects questions from a list of the most frequent questions • when the actor did not find the right question she/he can choose to talk to the chatbot
System respond	<ul style="list-style-type: none"> • the system will answer the questions using the expert system

Table 5.4: Ask questions use case table.

Use case name	New question
Use case ID	5
Brief description	When the user asks a new question to the chatbot .
Actors	users
Precondition	An inference engine interprets and evaluates the facts in the database in order to provide an answer.
Flow of activities	<ul style="list-style-type: none"> • The actor chooses to ask chatbot. • The actor enters his/her new question which was not found in the frequently asked questions.
System respond	The chatbot responds to not found the answer .

Table 5.5: New questions use case table.

Use case name	Answer
Use case ID	6
Brief description	When the user asks the employee a question
Actors	Employee
Precondition	The user clicks on the "Ask the employee" icon
Flow of activities	The user clicks on the "Ask the employee" icon to call the employee
System respond	The employee answer the user then end the call

Table 5.6: exit use case table.

Use case name	Exit
Use case ID	7
Brief description	the users can quite and exit whenever she/he wants
Actors	users
Precondition	enter the chatbot
Flow of activities	The actor clicks on exit button
System respond	close the program immediately

Table 5.7: exit use case table.

5.3 Static Aspect Design

We used a class diagram to demonstrate structural programming, Detailed Class Diagrams, and the relationships between the tables.

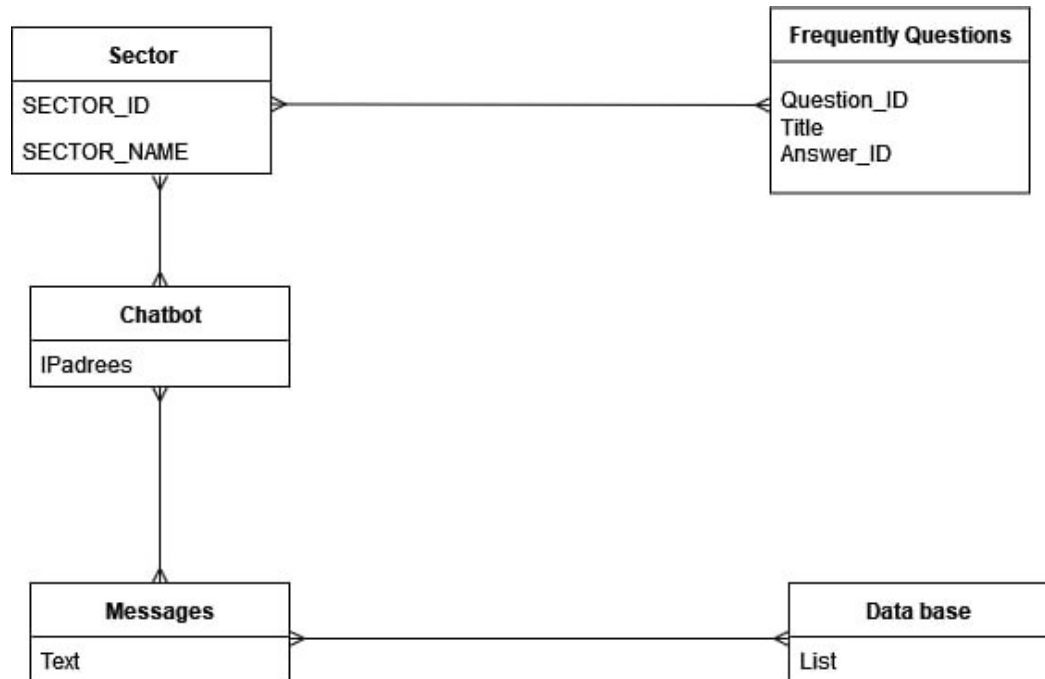


Figure 5.2: Class diagram

Classes	Description	Relations	Relationships Description
Sector	Sector class is about classifying the sectors in a way that makes it easier for the students to find their questions and answers quickly.	The sector has a many-to-many Relationship with Frequently Questions class. Also, has a many-to-many Relationship with the chatbot class.	The user can choose any question or can ask the chatbot more than one question.
Frequently Questions	After selecting the sector, It will appear to the users the most common questions that have been asked by the other students, and the users can choose from them.	Frequently Questions have a many-to-many relationship with sector class.	There may be more than one different Frequent question asked by users that are related to sectors.
Chatbot	When there is another question that the user did not find from the frequent questions, he/she can ask the chatbot by writing the question, and it will be answered.	Chatbot has a many-to-many relationship with Sector. Also, has a many-to-one relationship with Don't find the answer.	In the chatbot class, the user can ask questions from many sectors, and in the chatbot, users can send many messages.
Database	The database is for the new questions that aren't stored.	has a many-to-one Relationship with the answer Messages class.	many questions from the message class could come to the database.
Message	The messages class is about the type of messages that the user writes and its type is "text".	The chatbot received many messages.	The chatbot could receive many messages from the users, but only the new questions go to the database.

Table 5.8: classes description.

5.4 Sprint 2 Architectural Aspect Design

The chatbot application's architecture will define the system's design and structure, as well as the interaction between its components.

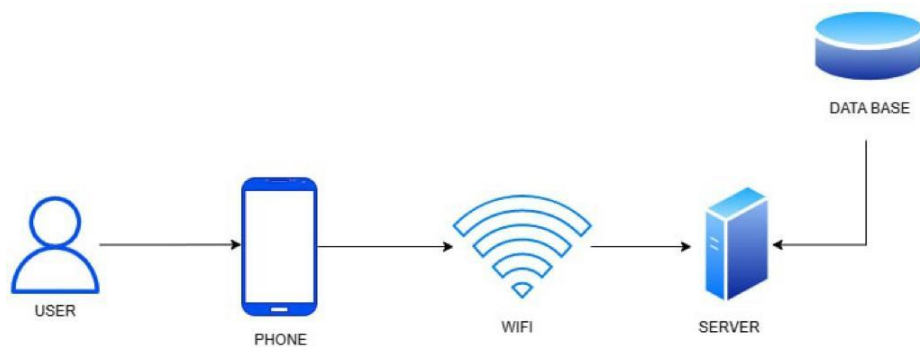


Figure 5.3: Architectural Aspect Design

icon name	Description
user	is the beneficiary of our chatbot, and they are the students and the faculty members
phone	Is the environment where the users can access the chatbot
server	The chatbot code is on a server to connect to the Internet
datasets	A Dataset is a set or collection of data, that store the questions and answers .

Table 5.9: Description

5.5 Sprint 2 Dynamic Aspect Design

The sequence diagram of the chatbot application is shown in the figure. This graph shows how the chatbot application interacts with users in many situations. Such as, when the user enters the chatbot, the chatbot will display the sector options, then the user should choose one of the sector options to display the most frequently questions and get an answer or choose to ask questions to the chatbot. In addition, many cases show us how a chatbot application acts in many situations with the user shown in figure 5.5.

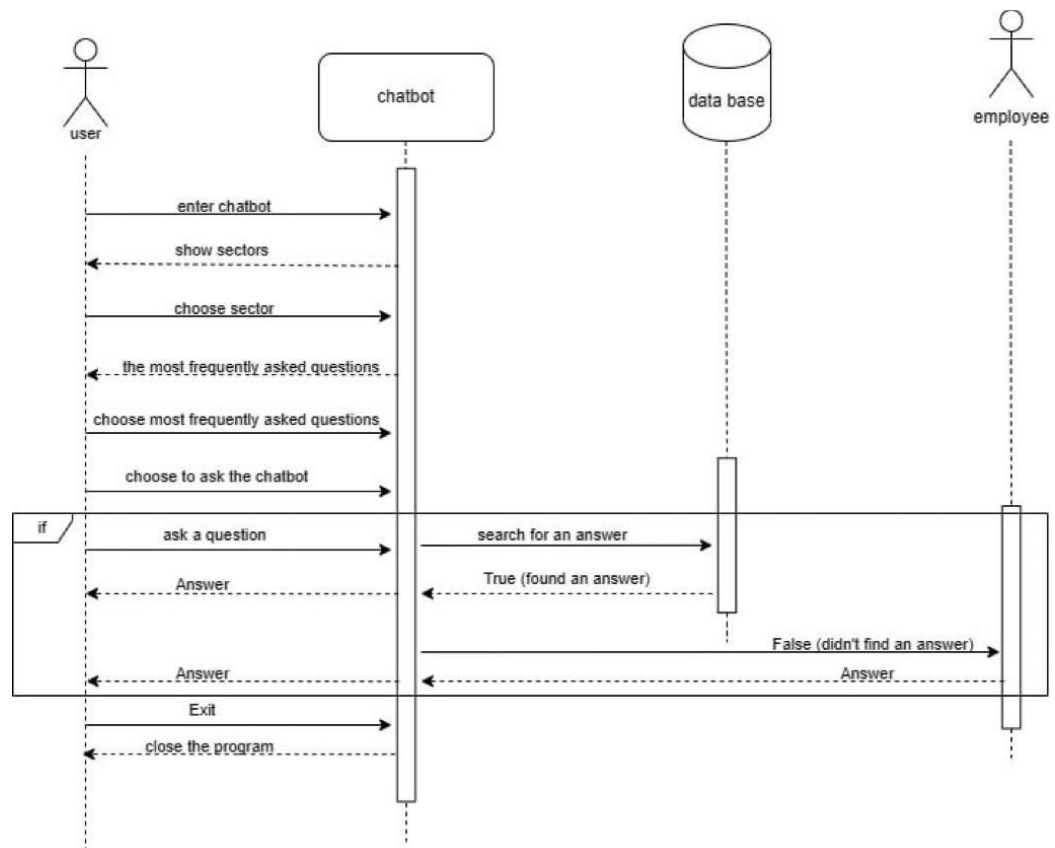


Figure 5.4: Sequence diagram

5.6 Sprint 2 Implementation

In the implementation phase, we created a chatbot to answer the users questions, and also we added a new icon if the user wants to ask an employee.

Description of the used software: We used Android Studio and which is an integrated development environment for Google's Android operating system, using XML and Java language for the Interface.

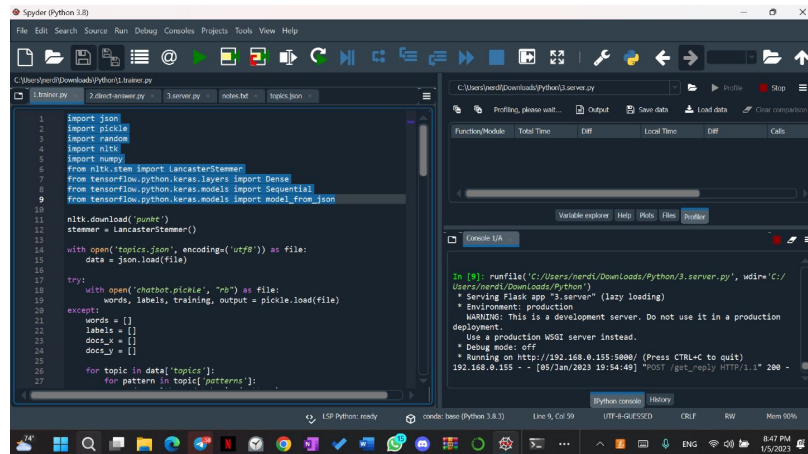


Figure 5.5: libraries and the chatbot code

This figure shows how we trained the chatbot and the libraries we use to save the answers

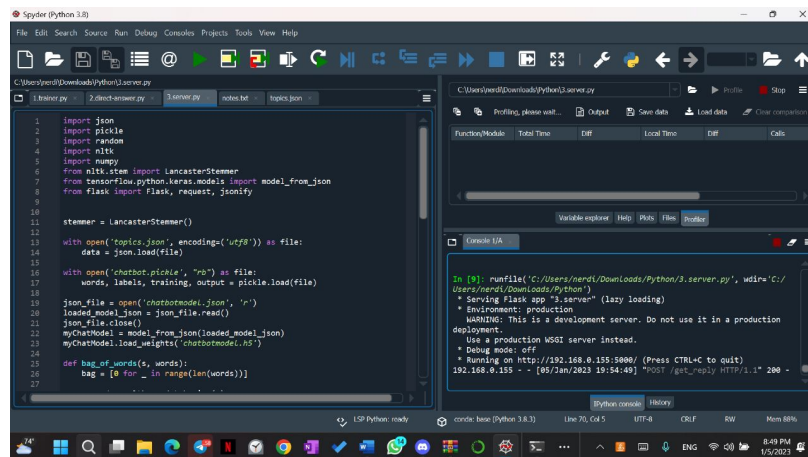


Figure 5.6: server code

This figure shows the server code where it shows the saved answer

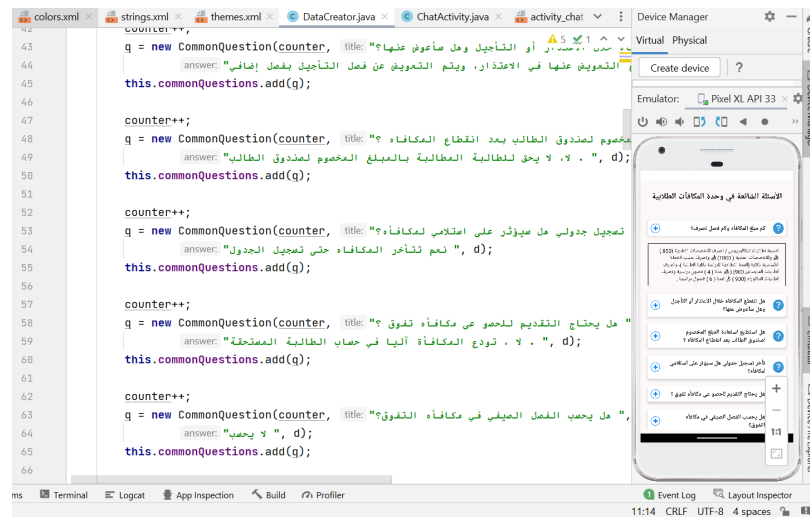


Figure 5.7: frequently asked questions code
This figure shows the frequently asked questions

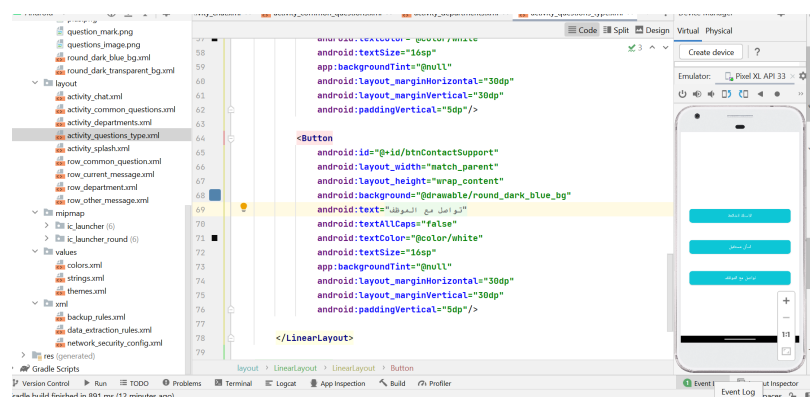


Figure 5.8: call the employee button
This figure shows the new icon "call the employee"

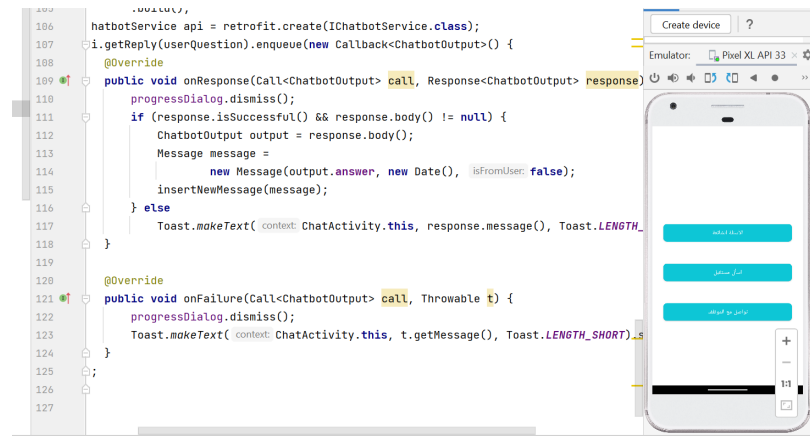


Figure 5.9: call code
This figure shows the "call" code condition

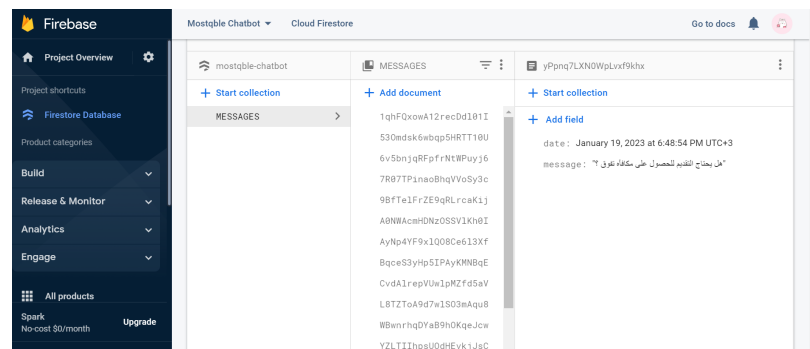


Figure 5.10: database
This figure shows the database we used which's firebase, in the left side, we have collection "messages" that save all questions, and on the medial, there is the "document" and each document has a unique ID, the last side have "field" that stores the date of sending a message and the message

5.7 Sprint 2 Testing And Evaluation

Scenarios for unit testing:

When the user clicks on the “specific question” icon? The question will be answered.
When the user clicks on the “ask the chatbot” icon? go to the chatbot page.

No	Test case	Feature	Description	Step to execute	Test data/Input	Expected results
1	TC-001	Chatbot responses	Test the chatbot when it finds the answers to the questions	The user enters the chat to ask the chatbot	N/A	The chatbot answers the question
2	TC-002	Chatbot responses	Test the chatbot when the answers to the questions are not found	The user enters the chat to ask the chatbot	N/A	The chatbot show ”Did not find the answer please ask the employee”
3	TC-003	Employee assistance	Test the response of the employee when he/she gets a question from the user	The user clicks on the ”ask the employee” icon	N/A	The employee answers the user’s question

Table 5.10: unit Test

Scenarios for Integration Testing:



Figure 5.11: sector list
in this page the user can choose any sector from list

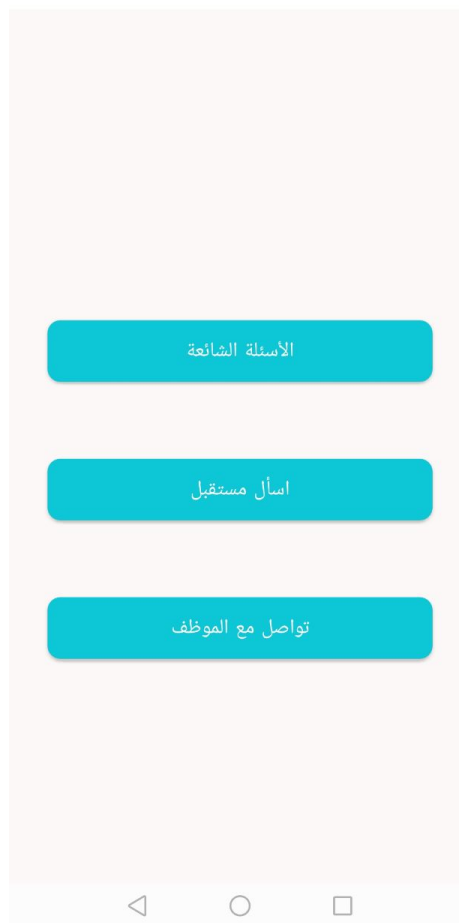


Figure 5.12: Options page
after select the sector this page will appear



Figure 5.13: frequent questions page
when choose "frequent questions" icon this page will appear and the user can select any question from "The most frequent questions page" then the answer will appear to the user.

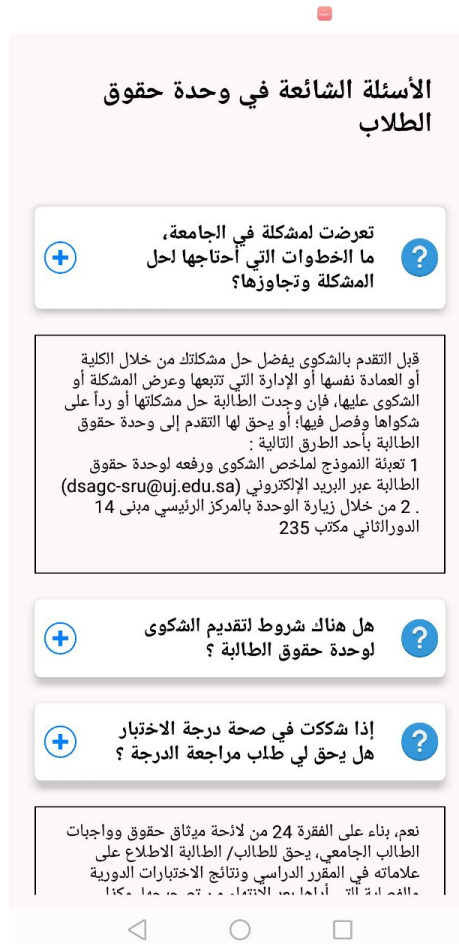


Figure 5.14: test application
The most frequent questions page



Figure 5.15: mostaqbl page
when the user chooses "Ask mostaqbl" this page will appear and can chat with mostaqbl



سعر استأجر اللوكر

سنويا بقيمة 40 ريال

غرامه فقدان مفتاح اللوكر

مبلغ 50 ريال

من الجهة المسؤوله عن المفقودات

مركز خدمة الطالب

هل يمكنني تحليل شخصيتي

نعم يوجد وحدة متخصصة بالمقاييس النفسية
ومنها مقياس mmpi كما يوجد عدة مقاييس
أخرى

اكتب رسالة.. ارسال

Figure 5.16: mostaqbl page



Figure 5.17: mostaqbl page
When no answer is available to the chatbot

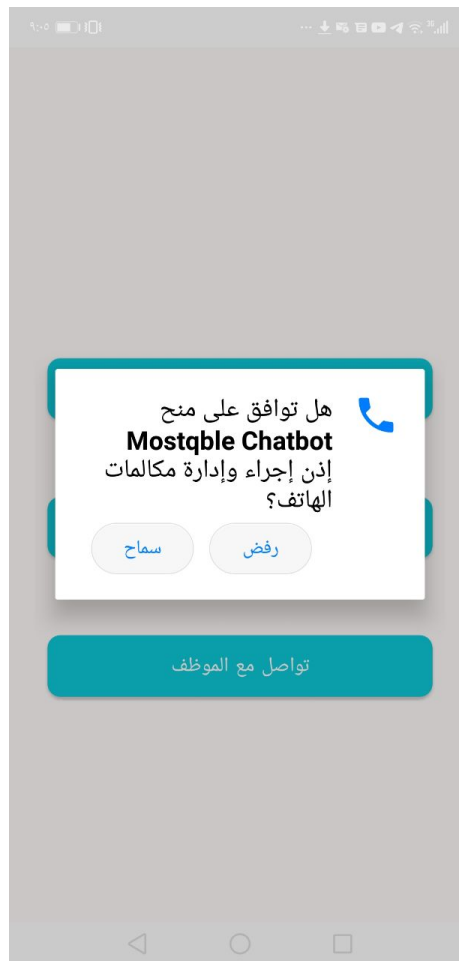


Figure 5.18: Ask the employee
When the user select Ask employee will automatically call them

5.8 Conclusion

This chapter provided the second development phase which is the first phase this semester. it contains the INTERMEDIATE functional requirements level, which is described by a use case diagram, Static Aspect Design, Architectural Aspect Design, Dynamic Aspect Design, Implementation, Testing, and Evaluation. In the next chapters, we will be covering the high-level requirements.

Chapter 6

Sprint 3 Development

6.1 Introduction

In this chapter, we will continue to provide details about the third and final Development phases reflecting the ADVANCED functional requirements. Section 6.2 provides the advanced use-case diagram to describe the interactions among the system components, In Section 6.3 Static Aspect Design advanced class diagram is used to demonstrate structural programming and relationships. Section 6.4 discusses Architectural Aspect Design to identify the interaction between its components. Section 6.5 provides a Dynamic Aspect Design sequence diagram of the chatbot application. Section 6.6 describes the implemented code. Section 6.7 provides a Testing And Evaluation Description of the different scenarios performed to test Sprint 3.

6.2 Use-case diagram

Use cases diagram shown in figure 6.1 The use-case is a graphical depiction of a user's possible interactions with a system. The Use-case will clarify what behavior is expected, and the exact methodology for getting there.

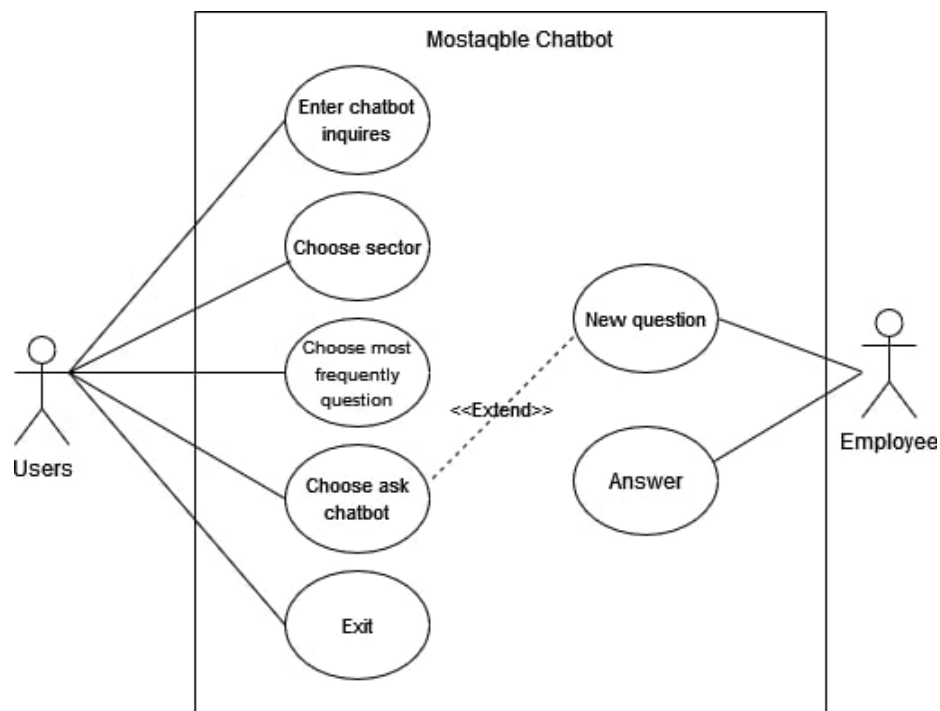


Figure 6.1: use-case diagram

Use case name	Enter chatbot inquires
Use case ID	1
Brief description	Enter to Mostaqble app then click the chatbot inquires icon
Actors	users
Precondition	Download Mostaqble
Flow of activities	<ul style="list-style-type: none"> The actor clicks on chatbot icons

Table 6.1: Enter chatbot inquires use case table.

Use case name	choose sector
Use case ID	2
Brief description	after the user enter the Mostaqble app she/he can select classification to identify the sector of question
Actors	users
Precondition	Have Mostaqble Chatbot
Flow of activities	<ul style="list-style-type: none"> The actor clicks on chatbot icons The actor select the classification of sector

Table 6.2: select sector use case table.

Use case name	choose most frequently question
Use case ID	3
Brief description	select a question after selecting classifications of the sector,the user finds the question she/he wanted to ask.
Actors	users
Precondition	select sector
Flow of activities	<ul style="list-style-type: none"> • The actor select the classification of sector • The actor select questions from list of most frequently questions

Table 6.3: select questions use case table.

Use case name	choose to ask chatbot
Use case ID	4
Brief description	Users can write the question themselves to the chatbot if they did not find it in the list of the most frequently asked questions
Actors	users
Precondition	select ask chatbot icon
Flow of activities	<ul style="list-style-type: none"> • The actor clicks on the classification • when the actor did not find the right question she/he can choose to talk to the chatbot
System respond	<ul style="list-style-type: none"> • the bot will answer the questions

Table 6.4: Ask questions use case table.

Use case name	New question
Use case ID	5
Brief description	When the user asks a new question to the chatbot .
Actors	users
Precondition	The user writes to the chatbot a new question.
Flow of activities	<ul style="list-style-type: none"> • The actor chooses to ask chatbot. • The actor enters his/her new question which was not found in the frequently asked questions.
System respond	The chatbot responds to answer questions and when can't find the answer it well calls an employee.

Table 6.5: New questions use case table.

Use case name	Answer
Use case ID	6
Brief description	When the user asks the employee a question
Actors	Employee
Precondition	The user clicks on the "Ask the employee" icon
Flow of activities	The user clicks on the "Ask the employee" icon to call the employee
System respond	The employee answer the user then end the call

Table 6.6: exit use case table.

Use case name	Exit
Use case ID	7
Brief description	the users can quite and exit whenever she/he wants
Actors	users
Precondition	enter the chatbot
Flow of activities	The actor clicks on exit button
System respond	close the program immediately

Table 6.7: exit use case table.

6.3 Sprint 3 Static Aspect Design

We used a Detailed Class Diagram which is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects, Detailed Class Diagrams, and the relationships between is description in the table below.

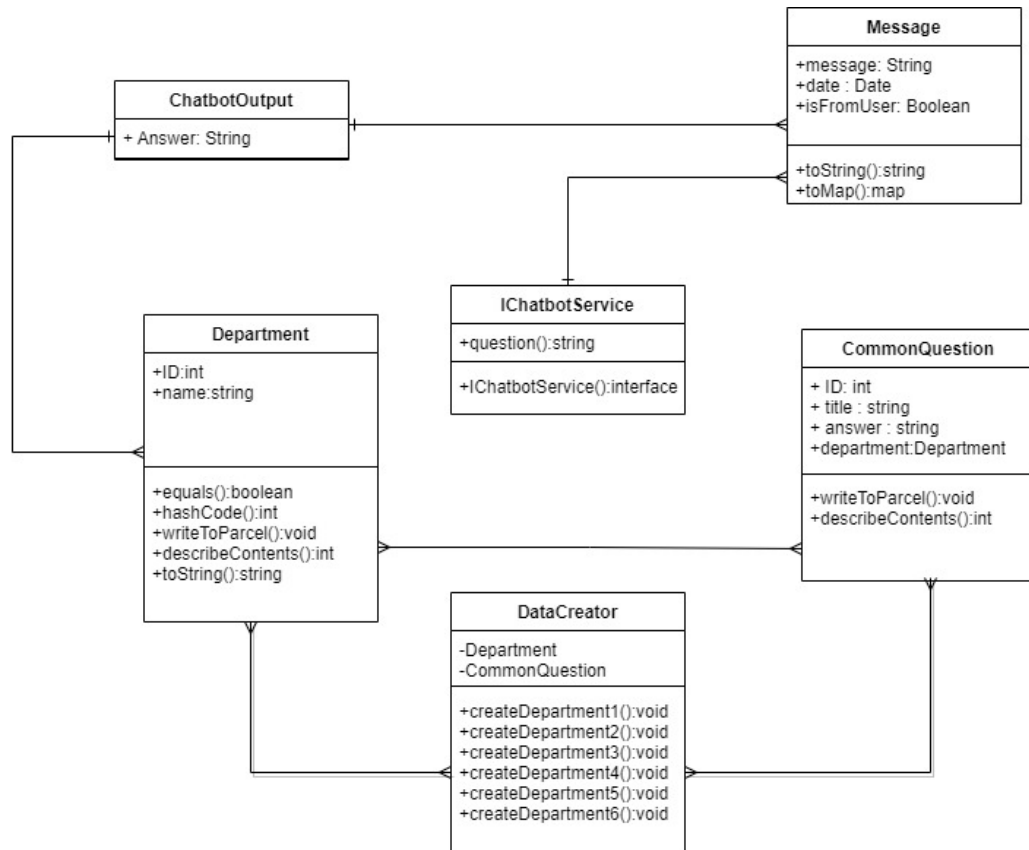


Figure 6.2: Detailed Class diagram

Classes	Description	Relations	Relationships Description
Department	Department class is about classifying the sectors in a way that makes it easier for the students to find their questions and answers quickly.	The department has a many-to-many relationship with DataCreator class, and many-to-many relationship with CommonQuestion class and a many-to-one relationship with ChatbotOutput class	For each department, there is data in it are stored in DataCreator class.
Common Questions	After selecting the sector or Department, It will appear to the users the most common questions that have been asked by the other students, and the users can choose from them.	Common questions have a many-to-manys relationship with DataCreator class, a many-to-many relationship with Department class.	There are many data and questions in the Common questions class saved in DataCreator class.
Chatbot Output	When there is another question that the user did not find from the frequent questions, it is asked to the chatbot by writing the question, and it will be answered.	ChatbotOutput has a one-to-many relationship with Messages and one-to-many relationship with Department class.	In the ChatbotOutput class, users can send many messages.
DataCreator	The list of all data is saved in the DataCreator class.	has a many-to-many Relationship with the department class, And many-to-many relationship with the Common Questions class.	Could come to the database, many questions from the message class.
Message	The messages class is about saving messages and mapping them.	The chatbot received many messages so it is many to one Relationship. Also, has many to one relationship with IChatbotService class.	The chatbot could receive many messages from the users, and these many messages are stored in the Firebase.
IChatbot Service	It is the Firebase, to save the messages and their date.	IChatbotService has one to many relationships with the Messages class.	IChatbotService saves many data coming from the Messages class.

Table 6.8: classes description.

6.4 Sprint 2 Architectural Aspect Design

The chatbot application's architecture will define the system's design and structure, as well as the interaction between its components.

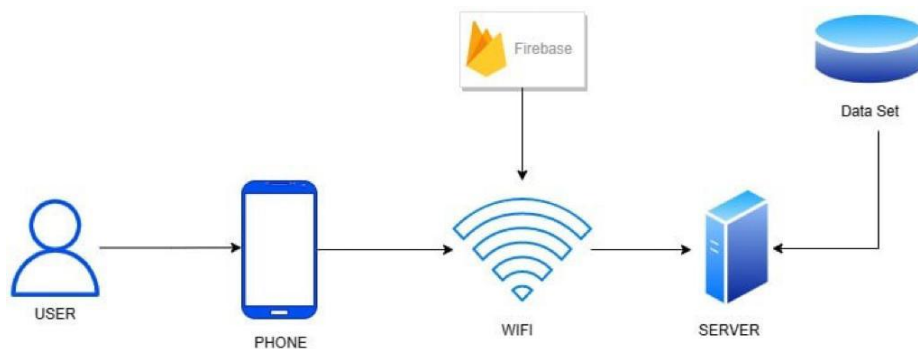


Figure 6.3: Architectural Aspect Design

icon name	Description
user	is the beneficiary of our chatbot, and they are the students and the faculty members
phone	Is the environment where the users can access the chatbot
server	The chatbot code is on a server to connect to the Internet
Dataset	A Dataset is a set or collection of data, that store the questions and answers.
firebase	firebase is a google backed application development software In our project we use it to stores the questions the user writes.

Table 6.9: Description

6.5 Sprint 3 Dynamic Aspect Design

The sequence diagram of the chatbot application is shown in the figure. This graph shows how the chatbot application interacts with users in many situations. Such as, when the user enters the chatbot, the chatbot will display the sector options, then the user should choose one of the sector options to display the most frequently questions and get an answer or choose to ask questions to the chatbot. In addition, many cases show us how a chatbot application acts in many situations with the user shown in figure 6.4.

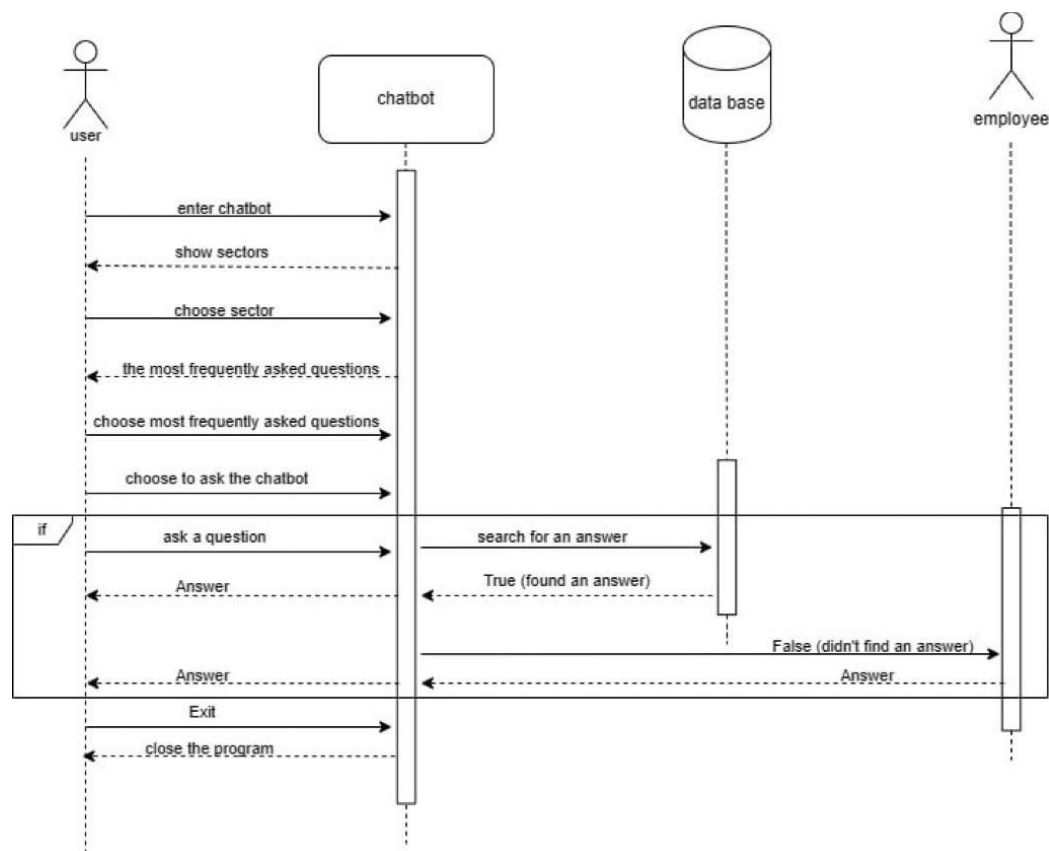


Figure 6.4: Sequence diagram

6.6 Sprint 3 Implementation

In the implementation phase, we make the chatbot smart by using AI techniques (natural language processing) to answer the questions. we used the NLP instead of the expert system because we faced so many problems when we tried to apply it to our app.

Description of the used software: We used Android Studio which is an integrated development environment for Google's Android operating system, using XML and Java language for the Interface. we divided the dataset in the JSON file into categories called tags. The number of classes we have is the number of types or tags in the JSON file. name of the model we used is the sequential model, The type of dataset we used to train and test your model is a JSON file containing statements of questions and answers, We have one dataset to train. The accuracy is 0,96.

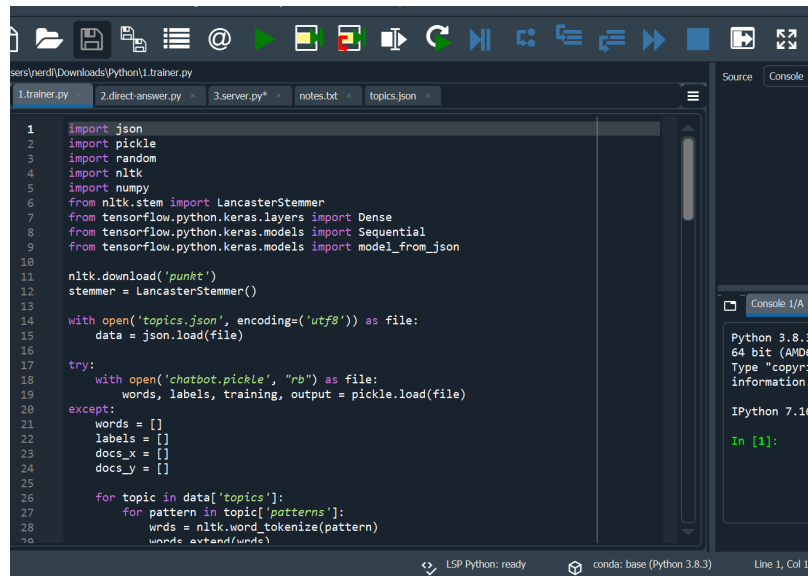


Figure 6.5: trainer code

this code will read from json file and create a model first we have a many librarys that helps such as json to read json file, nltk and LancasterStemmer that handel with word, from tensorflow we take layers and models second in line 14 we use(utf8) to use Arabic word and save the data as json file if the chatbot.pickle not found it will generate four arrays.

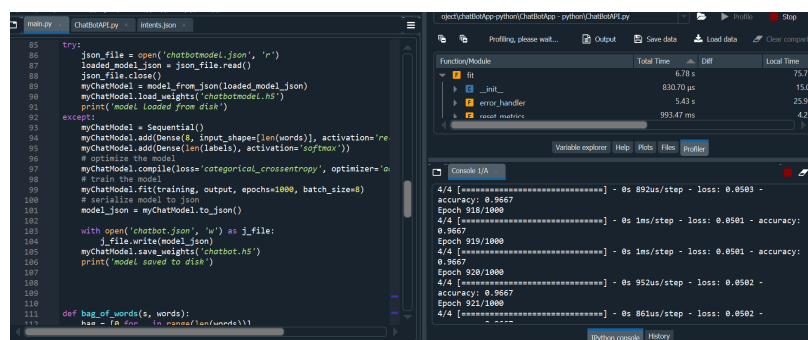


Figure 6.6: trainer code

here we have try to open the model, when the model not found will generated and fit model epochs=1000

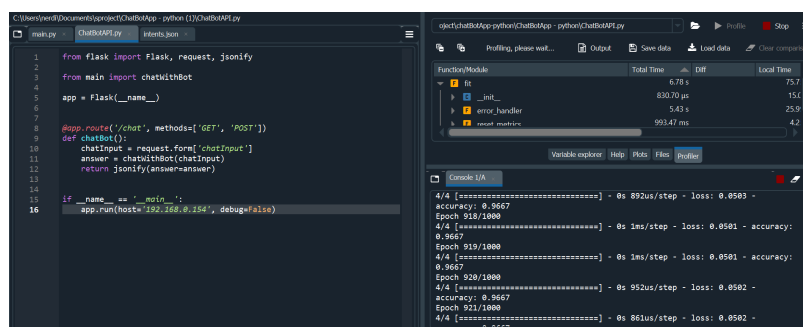


Figure 6.7: server code

This figure shows the server code where it shows the saved answer

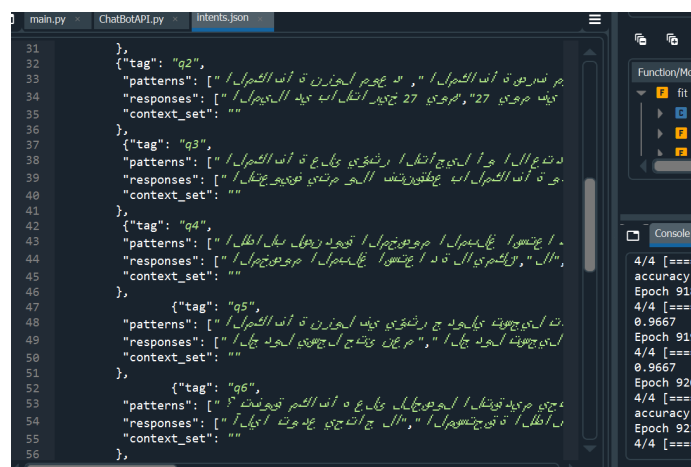



Figure 6.8: json file

This figure shows the json file that like dataset that save questions and train the chatbot

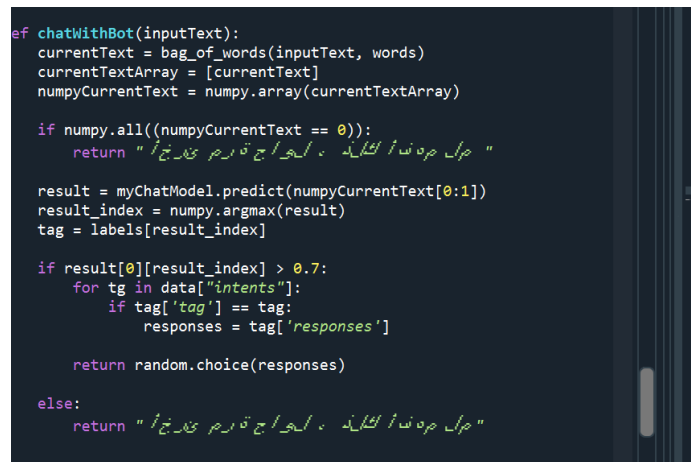


```

101     model_json = myChatModel.to_json()
102
103     with open('chatbot.json', 'w') as j_file:
104         j_file.write(model_json)
105     myChatModel.save_weights('chatbot.h5')
106     print('model saved to disk')
107
108
109
110
111     def bag_of_words(s, words):
112         bag = [0 for _ in range(len(words))]
113
114         s_words = nltk.word_tokenize(s)
115         s_words = [stemmer.stem(word.lower()) for word in s_words]
116
117         for se in s_words:
118             for i, w in enumerate(words):
119                 if w == se:
120                     bag[i] = 1
121
122         return numpy.array(bag)
123
124
125     # def bag_of_words(s, words):

```

Figure 6.9: json file
This figure shows the code



```

def chatWithBot(inputText):
    currentText = bag_of_words(inputText, words)
    currentTextArray = [currentText]
    numpyCurrentText = numpy.array(currentTextArray)

    if numpy.all((numpyCurrentText == 0)):
        return "مرحباً بك، أنا بوت دردش"

    result = myChatModel.predict(numpyCurrentText[0:1])
    result_index = numpy.argmax(result)
    tag = labels[result_index]

    if result[0][result_index] > 0.7:
        for tg in data["intents"]:
            if tg['tag'] == tag:
                responses = tg['responses']

        return random.choice(responses)

    else:
        return "مرحباً بك، أنا بوت دردش"

```

Figure 6.10: json file
This figure shows the code

6.7 Sprint 3 Testing Evaluation

Scenarios for unit testing:

When the user clicks on the “specific question” icon? The question will be answered.
When the user clicks on the “ask the chatbot” icon? go to the chatbot page.

No	Test case	Feature	Description	Step to execute	Test data/Input	Expected results
1	TC-001	Chatbot responses	Test the chatbot when it finds the answers to the questions	The user enters the chat to ask the chatbot	N/A	The chatbot answers the question
2	TC-002	Chatbot responses	Test the chatbot when the answers to the questions are not found	The user enters the chat to ask the chatbot	N/A	The chatbot show ”Did not find the answer please ask the employee”
3	TC-003	Employee assistance	Test the response of the employee when he/she gets a question from the user	The user clicks on the ”ask the employee” icon	N/A	The employee answers the user’s question

Table 6.10: unit Test

Scenarios for Integration Testing:



Figure 6.11: sector list
in this page the user can choose any sector from list

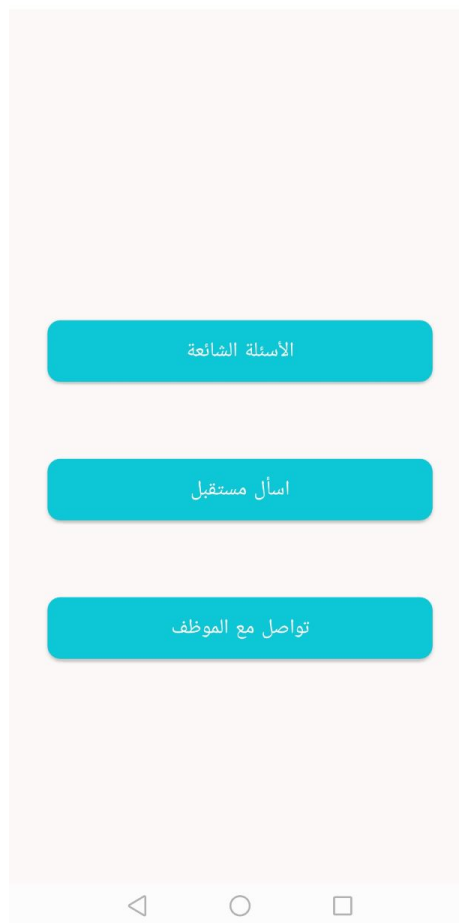


Figure 6.12: Options page
after select the sector this page will appear



Figure 6.13: frequent questions page

when the user choose "frequent questions" icon this page will appear and the user can select any question from "The most frequent questions page" then the answer will appear to the user.



Figure 6.14: test application
The most frequent questions page



Figure 6.15: test application testing the chatbot



Figure 6.16: test application testing the chatbot

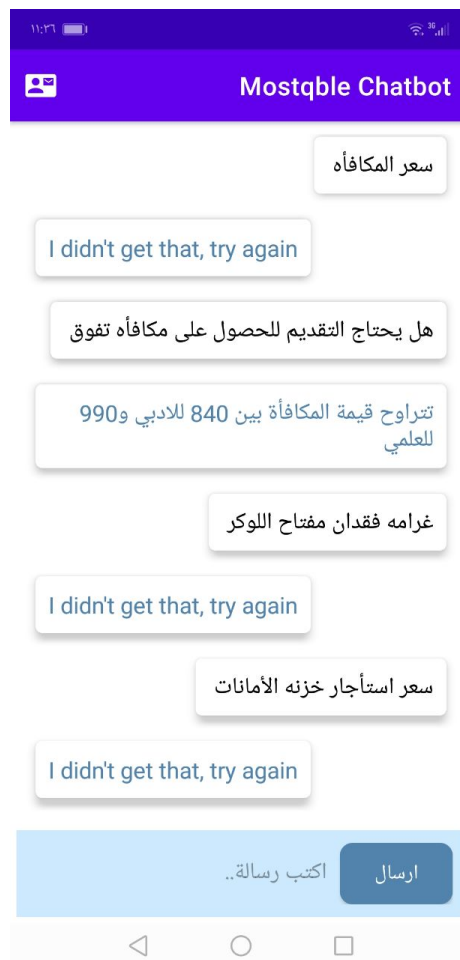


Figure 6.17: test application testing the chatbot

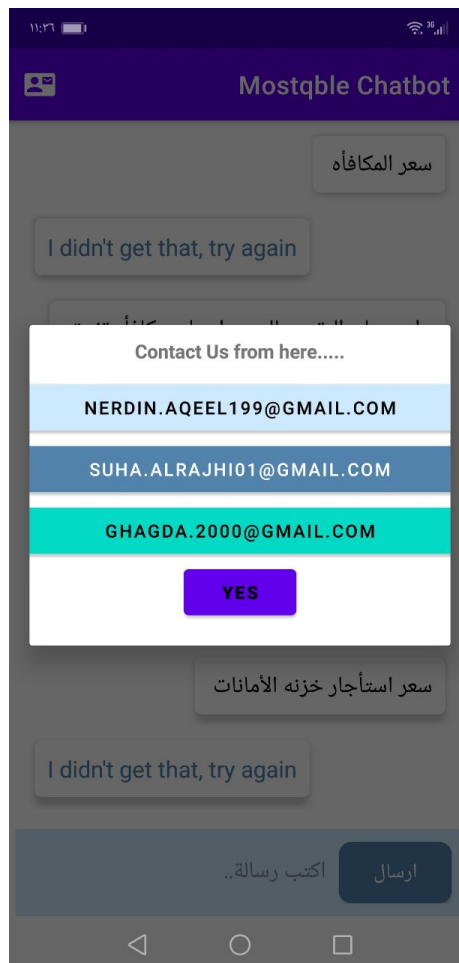


Figure 6.18: test application
new icon for contact us

6.8 Conclusion

This chapter is the final chapter of our senior project, in the first chapter we provided an overview of the Mostqble chatbot, the general context of our chatbot, problem definition, aims, objectives, proposed solution, report outline, and the project plan.

In the second chapter, we provided a background, existing systems, and a comparison between them,

In the third chapter, we provided requirements gathering by a google form to gather the different opinions from students, also we identify the functional and non-functional requirements.

The fourth chapter provided the first development phase which contained a basic use-case diagram, basic class diagram, basic Architectural Aspect Design, basic sequence diagram, Description of the implemented code, and Testing And Evaluation.

In the fifth chapter, we started the second development phase which contained an intermediate use-case diagram, an intermediate class diagram, an intermediate Architectural Aspect Design, an intermediate sequence diagram, a Description of the implemented code, and Testing And Evaluation.

Finally, chapter 6 provides the third development phase. it contains the ADVANCED functional requirements level, which is described by an advanced use case diagram, an advanced Static Aspect Design, an advanced Architectural Aspect Design, and an advanced Dynamic Aspect Design, Implementation, Testing, and Evaluation. We had difficulty applying the expert system and several problems, including the lack of sufficient time to do it and the difficulty applying due to the lack of sufficient resources to learn it, so we made our program through NLP Natural language processing, it is more effective in our project because it has the ability to understand the text from the user and respond.

General Conclusion and Future Work

Finally, an overview of what we've done on our project, we have created an interface for the user to choose from which sector they want to ask about, and then show them several common questions that they can choose from and also show them the answers, The users have the option to talk to the chatbot whenever they want and if they didn't get the answer they want they can click on an icon to communicate with an employee. We have created a server from Google, which is a feature they have, that helped us save the questions that the user asks the chatbot, the messages saved, and the time of the message's arrival is shown. also, we have a Jason file containing the questions and their answers for the chatbot used to answer the user's questions. We hope to add our project to the university application " Mostqble" to benefit from it in the future. In conclusion, we would like to thank our university, the University of Jeddah, for its efforts to educate and guide its students, and a special thanks to our supervisor in the senior project Dr.Dalal bakheet for guiding us in our last year.

Bibliography

- [1] oracle.com. Chatbots. [Online]. Available: <https://www.oracle.com/sa/chatbots/what-is-a-chatbot/>
- [2] techtarget. expert system. [Online]. Available: <https://www.techtarget.com/searchenterpriseai/definition/expert-system>
- [3] J. University. Academic system portal. [Online]. Available: <https://edugate.jazanu.edu.sa/jazan/ui/home.faces>
- [4] H. ADM. Hayyak account. [Online]. Available: https://twitter.com/hayyak_adm?s=21&t=AuKkN0Or6kI6U_6PcdOLug