



# Intelligent Chatbot of Qassim University Student Affairs

by

Muath Aljameeli 391108273  
Abdulmalik Alrebdi 391108073  
Bader Althunayan 391108023

Supervisor:

Dr. Mohammed Alsuhaibani

A project report submitted to Qassim University in partial fulfilment of the requirements for the degree of BSc in Computer Science at Department of Computer Science in College of Computer

Qassim, Saudi Arabia  
2022

## ABSTRACT

Student affairs in universities across the world deal with a huge amount of inquiries from students on a daily basis. Such inquiries include course registration, lecture details and lecturers' timetables. Dealing with such kinds of inquiries in a manual fashion is challenging to comprehend and time-consuming. This can be addressed by developing intelligent chatbot systems which can understand the key inquiry data. Thus, the development of intelligent chatbots to be capable of fulfilling the demands of continuously rising student inquiries becomes of utmost importance. We propose a novel intelligent chatbot system which can deal with Arabic inquiries for Qassim University student affairs.

## CERTIFICATE

It is certified that the project report has been prepared and written under my direct supervision and guidance. The project report is approved for submission for its evaluation.

Dr. Mohammed Alsuhaibani

## DEDICATION

We are bachelor's students at the Department of Computer Science, College of Computer at Qassim University. We would like to dedicate this project to College of Computer at Qassim University's, which provided unwavering support throughout the different phases of this project.

Muath, Abdulmalik and Bader

## ACKNOWLEDGEMENTS

The project team wishes to express their deepest gratitude to everyone who has contributed in any way to making this endeavor a reality. Foremost to almighty Allah, for the sagacity and constant flow of thoughts. We also want to express our sincere gratitude to our supervisor, Dr. Mohammed Alsuhaibani, for his motivation, reassurance, support, and assistance during the project. Likewise, we will thank ourselves for persevering, as well as anyone who believed in our ability to succeed.

Muath, Abdulmalik and Bader



# Contents

<b>List of Tables</b>	<b>8</b>
<b>List of Figures</b>	<b>9</b>
<b>1 Introduction</b>	<b>10</b>
1.1 Introduction . . . . .	10
1.2 Project Overview . . . . .	10
1.3 Project Scope . . . . .	11
1.4 Aim and Objectives . . . . .	11
1.5 Motivation . . . . .	11
1.6 Project Plan and Schedule . . . . .	12
1.7 Outline of the Project . . . . .	13
1.8 Summary . . . . .	13
<b>2 Background and Related Work</b>	<b>15</b>
2.1 Introduction . . . . .	15
2.2 Student Affairs . . . . .	15
2.3 Chatbots . . . . .	16
2.3.1 Rule-Based Chatbots . . . . .	17
2.3.2 AI-Based Chatbots . . . . .	17
2.4 Chatbots in Education . . . . .	18
2.5 Contribution . . . . .	19
2.6 Summary . . . . .	19
<b>3 Problem Analysis</b>	<b>20</b>
3.1 Introduction . . . . .	20
3.2 Problem Specification . . . . .	20
3.3 System Analysis . . . . .	21
3.3.1 Requirement Collection . . . . .	21
3.3.2 Requirement Analysis . . . . .	21
3.4 Requirement Organization . . . . .	23
3.4.1 User Files Processing . . . . .	23
3.4.2 Request Processing Flow and Integration . . . . .	24
3.5 Implementation and Evaluation Plan . . . . .	24

3.6	Technical Tools and Programming Languages . . . . .	25
3.6.1	Programming Languages . . . . .	25
3.7	Technical Tools . . . . .	25
3.8	Summary . . . . .	26
<b>4</b>	<b>System Design</b>	<b>27</b>
4.1	Introduction . . . . .	27
4.2	System Design Specification . . . . .	27
4.2.1	Home Page . . . . .	27
4.2.2	Login/Sign-in Page . . . . .	27
4.2.3	Admin Page . . . . .	27
4.2.4	Operator Page . . . . .	28
4.3	Design Architecture . . . . .	34
4.3.1	User and Web Application Interaction . . . . .	34
4.3.2	Web Application and Chatbot Interaction . . . . .	34
4.4	Summary . . . . .	36
<b>5</b>	<b>System Implementation</b>	<b>37</b>
5.1	Introduction . . . . .	37
5.2	Data . . . . .	37
5.2.1	Gathered data . . . . .	37
5.2.2	Possible Questions . . . . .	38
5.2.3	Chatbot Requests . . . . .	38
5.3	Code Structure . . . . .	40
5.3.1	User Interface (UI) . . . . .	40
5.3.2	Backend Tools and Frameworks . . . . .	40
5.3.3	Processing File . . . . .	41
5.3.4	Academic Schedules . . . . .	41
5.3.5	Chatbot Model . . . . .	42
5.4	Summary . . . . .	42
<b>6</b>	<b>System Testing and Result Discussion</b>	<b>43</b>
6.1	Introduction . . . . .	43
6.2	Testing Description and Test Cases . . . . .	43
6.2.1	Threshold . . . . .	43
6.2.2	Reduction of the Vocabularies . . . . .	43
6.2.3	Error Handling . . . . .	44
6.3	Results and Discussion . . . . .	44
6.3.1	Chatbot Results . . . . .	44
6.3.2	UI Result . . . . .	46
6.3.3	Deployment Result . . . . .	46
6.4	Summary . . . . .	51



<b>7</b>	<b>Conclusion and Future Work</b>	<b>52</b>
7.1	Conclusion . . . . .	52
7.2	Future Work . . . . .	53
7.2.1	Conversation Based . . . . .	53
7.2.2	Expanding the Scope of Questions . . . . .	53
7.2.3	API . . . . .	53
	<b>References</b>	<b>54</b>

# List of Tables

3.1	Keywords for the requirement analysis. . . . .	22
5.1	Sample of the data available provided by the admin. . . . .	38

# List of Figures

1.1	Plan and schedule for the project's phases. . . . .	12
2.1	Chatboat models general classification. . . . .	16
3.1	Workflow of the file processing. . . . .	23
3.2	Workflow of the Input and Output validation process. . . . .	24
4.1	An illustration of the homepage main components. . . . .	29
4.2	An illustration of the login page components. . . . .	30
4.3	An illustration of the register page components. . . . .	31
4.4	An illustration of the profile page components. . . . .	32
4.5	An illustration of the admin page components. . . . .	33
4.6	The process of getting and responding to the question by the proposed chatbot. . . . .	35
5.1	Sample of quesitons. . . . .	39
5.2	Another sample of questions. . . . .	40
6.1	This is the home page's main components. . . . .	45
6.2	Home page's main components. . . . .	46
6.3	Log-In page. . . . .	47
6.4	Register page. . . . .	48
6.5	Profile page. . . . .	49
6.6	Admin page. . . . .	50

# Chapter 1

## Introduction

### 1.1 Introduction

An introduction to the proposed system for this project is provided in this chapter. First, a brief summary and overview of the project is taken into account. The project's scope is then explained in further details. The aim and objectives of the project are next discussed, followed by the motivation for presenting the idea. The timeline for each project phase is then covered in the discussion of the project plan and schedule. The chapter is then ended with a project overview and a succinct summary.

### 1.2 Project Overview

The department or division that provides services and support for student achievement at institutions of higher education to promote student growth and development is known by several names, including student affairs, student assistance, or student services [1]. Educators, practitioners, and professionals in student affairs are all terms used to describe those who work in this area. In higher education institutions, these student affairs professionals support and assist students and promote learning beyond the classroom.

Student affairs deal with several areas including academic advising, admissions, campus activities and enrollment management. As such, every day, student affairs units in universities throughout the world deal with a sizable volume of questions from the students. These inquiries cover topics like course enrollment, lecture information, and lecturer schedules. Manually answering these kinds of queries is time and effort consuming. By developing intelligent chatbot systems that can comprehend the crucial inquiry data, this problem can be solved. Therefore, developing intelligent chatbots that can respond to students' increasingly complex questions becomes of highest relevance.

Chatbots are computer programs that use Artificial Intelligence (AI) to simulate human dialogue [2]. It is intended to serve as the best virtual assistant possible, assisting

users with duties like question-answering. Due to their ability to manage several customers at once and lower customer care costs, chatbots are now more common in corporate settings. However, in order to do a variety of duties, chatbots must be as effective as feasible. In order to solve this issue, we present in this project the development of a chatbot that uses AI techniques to quickly and accurately respond to any Arabic inquiry based on academic data from student affairs at Qassim university. Any department may utilize this chatbot to provide interactive student affairs chatbot to students who are really inquisitive.

## 1.3 Project Scope

The scope of the proposed system will be limited to developing an Arabic web-based chatbot that mimic the work of students affairs offices at Qassim university to answer any inquiry about the course enrollment, course registration, lecturers, timetable, conflict between courses and lecture locations and instructors. The proposed system will utilize AI techniques to fully grasp the inquiries in order to accurately respond.

## 1.4 Aim and Objectives

The aim of the project is to design a web-based intelligent Arabic chatbot that helps students by answering their questions and inquiries which will result in saving students and student affairs units time and effort.

The objectives of the project can be summarized as follows:

- Building an appealing user interface with user experience in mind.
- Building an intelligent model that gives users a human-like Arabic interaction in the Arabic language.
- Utilizing state-of-the-art Machine Learning (ML) techniques to semantically understand student inquiries.
- Providing both guest and registered users services.
- Providing an admin panel to control model files.
- Secure and authenticated user registration.

## 1.5 Motivation

Thousands of students around the world tend to daily have questions and inquiries regarding their academic and non-academic matters which might not be answered accurately or them being waiting for long to obtain the required information. Due to the nature of students' lifestyles during their studying journey, they require a high standard of efficiency and time management. As such, students face difficulty as some questions that are time-sensitive take some time and effort to be answered. Whether the question is to be answered in person, through E-mail or using the university website. Each option

has its weight on the students as each is constrained by time or location, which will result in additional effort from the student.

There are minimal attempts that use cutting edges ML and web development techniques to deal with the Arabic inquiries by students, which motivate us to propose this project. The primary motivation behind our proposing this project is to exploit the AI, ML and web development fields in a specific topic such as chatbot. A contribution to the usage of the Arabic language in the ML and AI community is also one of the key motivations for purposing this project. Moreover, we hope this system can serve as a core system for all student affairs units in the colleges of Qassim university.

## 1.6 Project Plan and Schedule

The project's needed duties have been broken down into many stages. Each stage is divided into a number of weeks and given an approximate time and progress for completion. Figure 1.1, shows the time, tasks and progress towards the project completion.

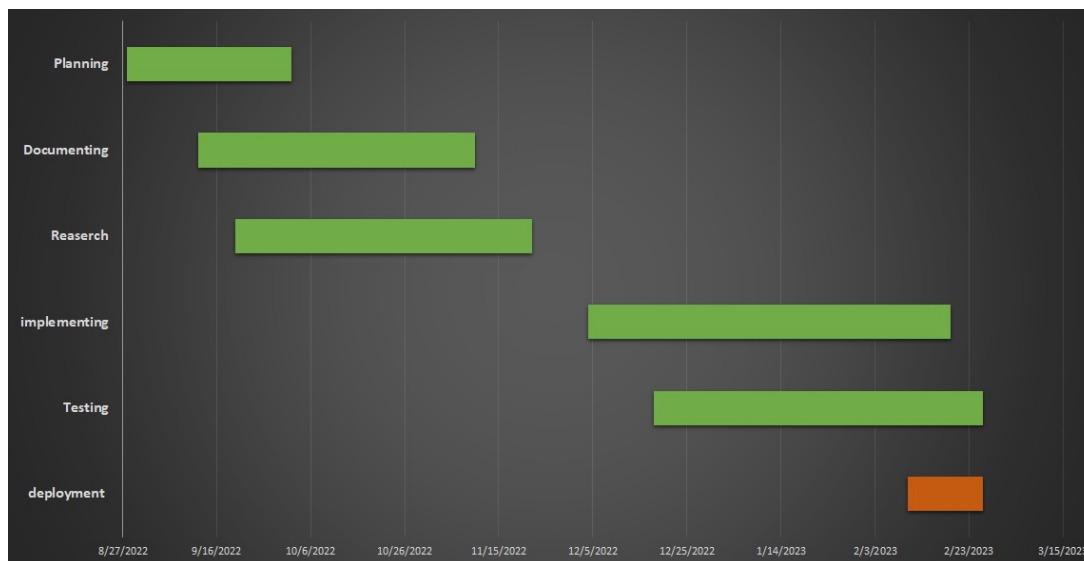


Figure 1.1: Plan and schedule for the project's phases.

The breakdown of each of the tasks shown in the figure above can be summarized as follows:

- **Planning**

Contains the planning of the proposed system and details of the suggested solutions.

- **Documenting**

The writing up of the project report along with a detailed explanation of the proposed solution.

- **Researching**

Finding useful resources to help improve the proposed system.

- **Implementation**

The implementation and technical development of the proposed system.

- **Testing**

Testing will be done while the system is still under development to ensure that development is on the right track and requirements are being met.

- **Deployment**

The last step of building the system is to deploy it to live servers to be ready for launch and mark the end of the system-building phase.

## 1.7 Outline of the Project

The remainder of the report is organised as follows:

- **Chapter 2:**

A literature review of intelligent chatbot systems is illustrated. A such literature review includes a background of technical and automation solutions for student affairs and a comprehensive study of previously proposed intelligent chatbot systems.

- **Chapter 3:**

An extensive analysis of the problem and precise description and specification of the problem with a detailed breakdown of all the system requirements. Listing of technical tools and programming languages.

- **Chapter 4:**

In this chapter detailed system design and specification are discussed. Design architecture that explains the workflow of the different components of the system use case scenarios that will put into perspective the design of the proposed system.

- **Chapter 5:**

This chapter presents an overview of the project implementation process. To begin, a detailed explanation of the various implementation phases will be provided. Then, the data and technical tools necessary for our project will be discussed.

- **Chapter 6:**

This chapter is dedicated to presenting the results and evaluation methods of our project. We will showcase the performance of our model using the chatBot, UI and Deployment results.

- **Chapter 7:**

The concluding chapter summarizes the findings and contributions of the project, highlighting the key results and their implications. It also discusses the limitations and future directions of the study, providing recommendations for further research in the field.

## 1.8 Summary

The chapter provided an overview of the work included in this project, emphasizing the project's scope, purpose, and objectives as well as its rationale, plan, and timetable.

A brief summary and an outline introducing the next chapter follow the chapter's conclusion.



# Chapter 2

## Background and Related Work

### 2.1 Introduction

The proposed system intends to provide an AI and web-based chatbot model for Qassim university student affairs, as was stated in the previous chapter. It was also mentioned that the continuous needs of students in universities have led to a rise in interest in the literature on the use of chatbots. Thus, the chapter will cover the related research to the system that is being proposed. First, the background of student affairs is presented, and then a thorough discussion of related work follows. The contribution of the suggested system is then covered in the chapter. A brief summary then wraps up the chapter.

### 2.2 Student Affairs

Student affairs are the department or division of higher education institutions that provide services and support for students' achievement in order to promote student growth and development. Student affairs practitioners, educators, and professionals are all terms used to describe those who operate in this area [3]. Depending on the size, nature, and location of a university, a student affairs unit (or department) can be different in size and structure. In Saudi Arabian universities, this unit is usually led by the vice-dean of the college, in which the unit itself falls under his/her responsibility.

The duties of student affairs vary from different institutions, universities and colleges, however the responsibilities normally involve activities such as answering students inquiries about timetable, courses registration and lectures details, communicate with the Deanship of Student Affairs to inquire about everything related to students, paying attention to the time of students outside the study halls and investing them in programs that develop the integration of their personalities and provide them with life skills that help them to serve themselves, society and their country, facilitating the university students' life in the universities and providing everything that works to raise their level of

educational attainment in all fields, providing the college departments with all the information they need on the study status of students from the reality of the available records and information, and preparing a report for the dean, and study students' problems and suggest solutions to them.

Keeping pace with the enormous volume of inquiries from students to students affairs for all the above-mentioned activities is now required massive effort, time and process. Therefore, there was a need for more intelligent systems that are able to understand and analyze the students' needs and automatic responses to them.

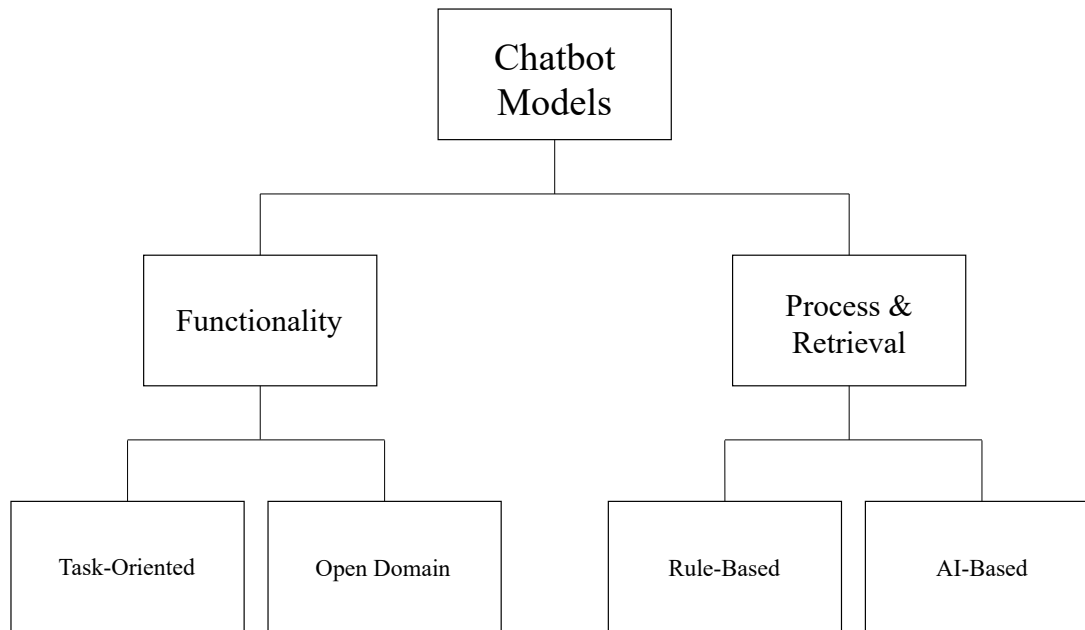


Figure 2.1: Chatboat models general classification.

## 2.3 Chatbots

Human-computer conversation systems using natural language are referred to as chatbots [4]. Alan Turing, who posed the question “*can machines think?*” in 1950, is credited with developing the initial notion of the chatbot [5].

Chatbots mostly fall into two categories in terms of functionality (Figure 2.1). The first is task-oriented, which although not the finest conversational agent, is particularly robust when it comes to carrying out particular duties and managing orders that are specific to a given domain. Their use ranges from reserving a table at a restaurant, purchasing airline tickets, and answering student inquiries, among other things. Open-domain chatbots are the second category. These are the conversational bots that attempt to sound like people. Their goal is to provide human-like reactions so that the other person would think it is

a real human [6]. It is worth mentioning that what we are proposing in this project falls under the first category, task-oriented, for student affairs-related inquiries.

As highlighted in Figure 2.1 too, in terms of process and information retrieval, Chatbots can be broadly classified into two main classes, rule-based and AI-based, which we will review in the next sections.

### 2.3.1 Rule-Based Chatbots

The very earliest implementations of chatbots were rule-based. Although rule-based models are typically smooth to build and deploy and for particular cases give impressive results, their capabilities are sometimes limited since they may face some issues responding to complicated questions. Rule-based chatbots respond to users' questions by searching for patterns that match; as a result, they are likely to give incorrect replies if they encounter a sentence that does not fit any established pattern [7].

As a concrete example, using a mobile chatbot, Miura et al. [8] create and assess a virtual caregiver system. Their main concept is to incorporate a rule-based virtual caregiver system together with the physical, mental, and social surveys into the mobile chat application, as opposed to the traditional health monitoring strategy. The elderly person answers one question every day from the mobile chatbot by pressing an optional button. Similarly, with the help of the Neo4J graph database and morphological analyzer khaiiii (Kakao Hangul Analyzer III), Kim et al. [9] create a rule-based smart tourist chatbot. They employ a morpheme analyzer, a proper noun dictionary that includes the names of tourist destinations, and a general noun dictionary that contains commonly used terms in tourist information searches in their proposed chatbot system to determine the user's purpose. The Neo4J graph database-based tourism knowledge base offers sufficient responses to visitors' inquiries.

### 2.3.2 AI-Based Chatbots

Since Turing, machine learning and improvements in natural language processing (NLP) have helped chatbot technology evolve. The usage of chatbots has also grown in various social networks [10]. In fact, chatbots are currently used in a wide range of industries and applications, including e-commerce, healthcare, entertainment, and education. Consequently, chatbots may provide consumers services as well as help in a variety of domains.

Machine learning, deep learning and NLP-based systems are currently the dominant techniques in developing Chatbots. In fact, Chatbots are nowadays more adaptable in terms of use and upkeep, and increasingly able to replicate human communication thanks to recent advancements in AI, ML and NLP [11].

For example, Shetty et al. [12] built an AI-based healthcare chatbot that can suit all of our needs and benefit the healthcare sector. By setting up appointments and providing them with the appropriate care, this system communicates with patients and helps them find expert doctors. Their system's implementation is based on AI training data and

algorithms. Particularly, simple TF-IDF and Cosine Similarity algorithms are used. In the illustration below, the system first receives input from the user—symptoms—and then analyzes it using the TF-IDF and cosine similarity algorithms. In the same vein, Rui et al. [13] proposed an implementation of a deep learning-to-respond schema in a retrieval-based chat system using a deep neural network architecture that is powered by web data. For many discussion scenarios in an open domain, the suggested model is all-encompassing and consistent. They create many characteristics and aspects with optimization into the deep learning framework, taking into account the influence of diverse data inputs.

The development of new information retrieval algorithms has advanced recently, and it is important to note the machine learning algorithms that are now being utilized as the foundational technology for this sort of model [7]. In these techniques, the Information Retrieval process is improved by a Deep Neural Network that rates questions and answers that correspond with the most recent user's input, as well as those that match with rephrased versions of past conversation, turns. The rating lists for the various reformulations are then combined [13].

Transformers [14], has been one of the most intriguing developments in deep learning. Transformers are linguistic representations that only use the Attention mechanism. By differently valuing the importance of each component of the input data, transformers have replaced Recurrent Neural Network models like long short-term memory (LSTM) as the preferred model for NLP issues. Transformers-based systems like BERT (Bidirectional Encoder Representations from Transformers) [15] are very well-known in the fields of AI and NLP for reporting state-of-the-art results in various tasks. These systems were trained using sizable language datasets like the Wikipedia Corpus and Common Crawl. As such, Daniel et al. [16] unveiled Meena, a generative chatbot model that was trained from beginning to end on 40 billion words that were retrieved and filtered from social media conversations that were in the public domain. They push the boundaries of the end-to-end methodology with Meena in order to demonstrate the viability of a large-scale, low-complexity model for producing high-quality linguistic outputs. The Evolved Transformer serves as the primary architecture in the authors' seq2seq model.

## 2.4 Chatbots in Education

Bhavika et al. [2] provide a proposal for a chatbot that uses Latent Semantic Analysis (LSA) and Artificial Intelligence Markup Language (AIML) trying to efficiently and accurately respond to student questions based on the dataset of FAQs. They use LSA to deliver replies at any moment that would fulfil user satisfaction. Template-based and general questions like welcome/greetings and general queries will be replied to utilizing the AIML. Similarly, to better comprehend the services offered by the university, Sushil et al. [17] have employed DialogFlow, an NLP module to convert the questions raised by students during conversations into structured data. This representative informs students about placement activities. To the best of our knowledge, these are the only attempts of chatbot systems in education.

Although the above-highlighted two previously proposed systems involve dealing with student inquiries in the education environment, they are not specifically designed for student affairs tasks, but mainly for students' placement activities and greetings and welcoming new students rather than dealing with students' academic inquiries in regard to lectures, enrollment, course registration and lecturers information, to name some.

## 2.5 Contribution

Despite the success of the aforementioned recent work in chatbots, there is a lack of effort towards using such efficient techniques with the Arabic language and with student affairs functionalities and students inquiries in particular. Therefore, the contribution of this project is directed towards building a task-oriented, hybrid of rule-based and AI-based and web-based chatbot system for Qassim university student affairs units.

## 2.6 Summary

The related work that is connected to the proposed system was covered in Chapter 2. It starts out by providing background information on chatbot systems and student affairs. The issue in existing systems, which is the primary emphasis on non-Arabic and non-Educational languages, is then provided after a full discussion of a synopsis of the associated work. The significance and contributions of the proposed system to this project are then highlighted. Next, the chapter comes to a close with a quick summary.

# Chapter 3

## Problem Analysis

### 3.1 Introduction

In this chapter, we will define the problem specification and clarify the potential problems that we might face, and then we will go through the system analysis which contains requirement collection, requirement analysis, and technical tools and programming languages.

As such, in this chapter, we will define the problem specification and clarify the potential problems that we might face, and then we will do the system analysis which contains the requirement collection, requirement analysis with its functional requirements, non-functional requirements and technical tools and programming languages.

### 3.2 Problem Specification

Being university students for the last four years we as a team witnessed some of the difficulties that students face in their time as university students. These difficulties range vastly in significance and importance. Each of these difficulties affects the students differently. One of these difficulties is the connection between students and the college students' affairs office. Students their times in the university almost always need to go to the student affairs office at least a few times for inquiries, asking for help or guidance and other different reasons. The student affairs office tried to make this process easier for students by enabling remote connection channels for the office to make public announcements and for the students to send their messages. Although the students' affairs office usually makes a great effort, the process is still time-consuming for the students to wait for a response to their inquiries which might be urgent, and costly as well for the student affairs unit. We recognize this problem and plan to offer an efficient solution. The principle idea is to apply our skills and expertise to build a hybrid of rule and AI-based as well as a web-based chatbot that is intuitive, interactive, intelligent, easily accessible, and available all the time. From the student's perspective, this solution is of great help as it is always available and requires almost no time or effort time to send inquiries and get

the answers. From the viewpoint of the student affairs office, this solution will assist in addressing a multitude of commonly asked questions and inquiries, ultimately conserving a significant amount of time and energy for the office employees.

## 3.3 System Analysis

### 3.3.1 Requirement Collection

Requirement Collection is a crucial aspect of software development. As our project aims to resolve a challenge faced by students, and since our team is comprised of students who are also currently experiencing this issue, we are in a prime position to accurately identify the requirements for the project, leading to an effective solution for the problem at hand.

### 3.3.2 Requirement Analysis

We list below the requirement analysis of the proposed system as bullet points followed by a kind of summary with illustrated keywords ( Table 3.1).

- **Practical**

The system must be easy to use, because the main idea is to design it to be intuitive and does not require a huge amount of time or effort compared with the current classic methods.

- **Availability**

The system must be always available at all times for the users.

- **Compatibility**

The system must work perfectly with any device such as smartphones, personal computers and tablets.

- **Retrieval**

The system should respond to the requests without any failure. If any errors are encountered, the system should communicate the error with the user via chatbot to mimic the human-like interaction.

- **Efficiency**

The system must respond accurately to the user inquiries in a quick manner and with great precision that gives the best result that answers the students' requests.

- **Performance**

While designing the system, high performance should be kept in mind as it will result in availability and ease of user usability.

- **Automation**

The chatbot will be fully automated and there will be no human intervention.

- **Inquiries Data**

The inquiries sent via the chatbot will be kept for further analysis and evaluation purposes such as measuring the model's accuracy as well as analyzing the data to improve the AI integration.

Table 3.1: Keywords for the requirement analysis.

Requirement	Keywords
Practical	easy to use, intuitive
Availability	always available, 24/7
Compatibility	responsive, smartphones, PC, tablets
Retrieval	no failure, requests
Efficiency	accuracy, response time
Performance	high, available, accurate
Automation	automated, no-human
Inquiries Data	analysis, evaluation
Dedicated Answers	personal data, unique
Integration	Symfony Framework
Security	confidentiality, integrity, availability, validation, authentication
User Experience	UX
User Interface	UI

- **Dedicated Answers**

Each user will be having a dedicated answer according to his personal data, such as his timetable and courses already been studied.

- **Integration**

A portion of our project will be developed using the Python programming language, requiring a means for the PHP components of the project to communicate with the Python scripts. A database connection will facilitate this integration and is therefore a crucial requirement for the project. The connection will be established to allow for the seamless transfer of data and files from the users to the models for processing.

- **Security**

Security is an essential requirement to make the system secure and apply the components of information security such as Confidentiality, Integrity and Availability (CIA), input validation and authentication.

- **User Experience (UX)**

UX is concerned with the client's feeling about the style of the system by providing interactable elements and animation, among others.

- **User Interface (UI)**

The UI is part of designing the system and simplifies the usage of this system by providing styling elements such as structure, text, and buttons, to name a few.



## 3.4 Requirement Organization

The project is divided into three main processes and in this section, we will demonstrate these processes step by step aided by flowcharts that help to visualize the workflow of the proposed system.

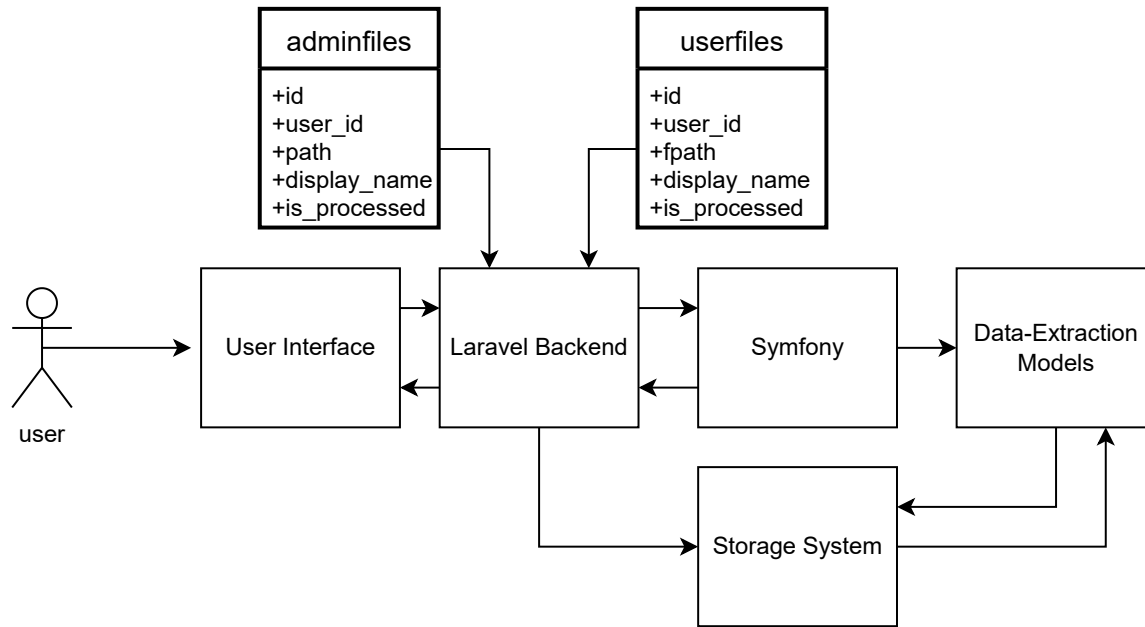


Figure 3.1: Workflow of the file processing.

### 3.4.1 User Files Processing

One of the project's requirements is to allow for the upload of files from users and administrators, as these files are necessary for the proper functioning of the processes. To extract the relevant information from these files, a file processing system is necessary. This system will extract the necessary data into a format that can be easily interpreted by the chatbot model. In order to ensure that the uploaded files are the correct ones, the Laravel PHP framework will be used for validation purposes.

The file processing system will be a subsystem written in Python, and in order to facilitate communication between Laravel and Python, the Symfony framework will be employed. The Symfony framework will allow for the seamless transfer of data between Laravel and Python. In addition, a user files table will be maintained in the database to keep track of all the files stored on the server. The workflow of the process, from user input to file storage, is illustrated in Figure 3.1. Figure 3.1 demonstrates the workflow of the process starting from user input and ending with file storage.

### 3.4.2 Request Processing Flow and Integration

In this section, we will discuss the process of receiving user input (i.e., chat messages) and how the request flows from the user to the chatbot model to generate an appropriate response and propagate it back to the user.

The process begins by receiving the HTTP request and preparing all the necessary data. This involves querying the database to confirm the availability of the latest files and fetching the required information. The chat requests are also recorded at this stage. Once the data preparation is complete, Symfony is used to execute the model and feed it with the prepared data.

After the model finishes processing the request, the record created by Laravel earlier will be updated, and Symfony notifies Laravel that the response is ready to be sent to the user. Finally, Laravel carries out the response to the user. Figure 6.6 illustrates the workflow of the process.

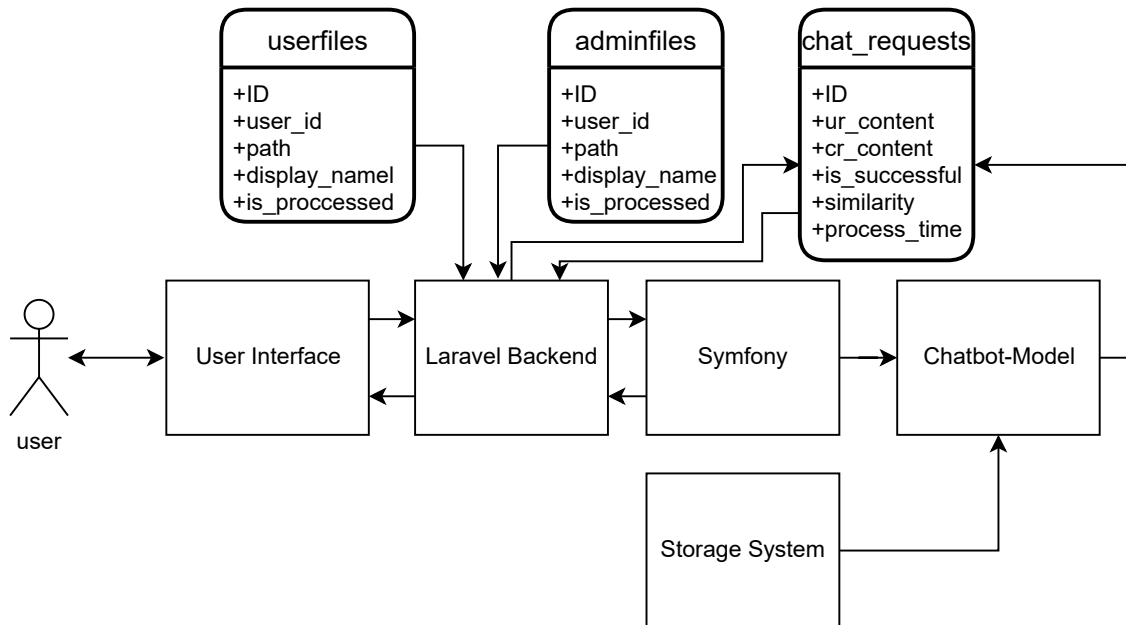


Figure 3.2: Workflow of the Input and Output validation process.

## 3.5 Implementation and Evaluation Plan

The planning for each stage was viewed in Figure 1.1, where each stage will have a specified time, starting from planning to deploying. The planning stage, is about trying to solve identified problems in an optimized and high-quality manner. Documenting stage is about writing the system such as finding similar systems, and services provided by the system, as well as problem specification.

The Research stage is searching and finding helpful resources for implementing the proposed system and for finding optimisation algorithms. The implementation stage focuses on the implementation of the system as a web-based application. The Testing stage is concerned with testing and making sure it works perfectly. Finally, the deployment stage is for deploying the system in a web server and making it globally accessible.

We plan to evaluate our system with real cases of students and student affairs from the College of Computers at Qassim University. The system's ability to respond accurately will be evaluated based on a given actual timetable from student affairs.

## 3.6 Technical Tools and Programming Languages

### 3.6.1 Programming Languages

- **Hyper Text Markup Language (HTML)**

HTML is the foundation of all web page development as it helps developers build the structure of their websites which make essential to our project since it is a web-based application.

- **Cascading Style Sheets (CSS)**

CSS enables developers to improve their UI design and optimize UX to provide an appealing and easy-to-use service to our users we will use CSS.

- **JavaScript**

JavaScript is a scripting language that enables developers to create dynamic web pages opening a great opportunity to be creative. Alongside other languages, JavaScript is also essential for us to make a dynamic web page that enables users to interact with services. Also, many processes need to be done on the browser side which will reduce the load on the server and then send it to the server for further processing or be stored in the database.

- **PHP**

PHP is a popular general-purpose scripting language that is especially suited to web development. we will utilize PHP to handle all the server-side processes such as interacting with the database, file management, interacting with python intelligence models.

- **Python**

Python is one of the most used high-level programming languages in tasks related to ML and NLP and AI in general. Python will be the main programming language to be used for the chatbot model.

## 3.7 Technical Tools

- **Tailwind**

Tailwind CSS is an open-source CSS framework. The main feature of this library is that, unlike other CSS frameworks like Bootstrap, it does not provide a series of predefined classes for elements such as buttons or tables.

- **Laravel**

Laravel is a free, open-source PHP web application framework designed for the development of web applications following the model-view-controller (MVC) architectural pattern. It offers an expressive and elegant syntax that helps developers to create scalable and maintainable applications with ease. With its built-in features like routing, middleware, ORM, templating engine, and more.

- **Bootstrap v5**

Bootstrap is a powerful, straightforward, and well-organised CSS framework that provides a modern UI and UX style. Designed to enable responsive development of websites and provides a collection of syntax for template designs.

- **Pandas**

Pandas is a fast, powerful, flexible, and easy-to-use open-source data analysis and manipulation tool, built on top of the Python programming language, we will use it to read files and manipulate.

- **Tabula-py**

Tabula-py is a simple Python wrapper of tabula-java, which can read tables of PDF. You can read tables from PDF and convert them into pandas' DataFrame. tabula-py also enables you to convert a PDF file into CSV/TSV/JSON file.

- **PyArabic**

Pyarabic Package provides functions to change some properties of Arabic text and numbers to change to English. We will use it to convert the number of each subject from Arabic to English for consistency.

- **Numpy**

Numpy is a library for the Python programming language, adds support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

- **Word2vec and gensim**

Word2Vec is a natural language processing technique used for translating words into vectors that are learned from large amounts of textual data and one of the most popular implementations for Word2vec in python is by using gensim

- **MySQL Connector**

'MySQL.connector' is a Python module used to connect to MySQL databases. It provides a straightforward API for communicating with MySQL servers, including support for executing SQL statements, handling transactions, and retrieving data. The module uses the official MySQL Connector/Python library and supports the latest versions of MySQL. It is lightweight, easy to install, and cross-platform compatible.

## 3.8 Summary

This chapter discussed the requirements of the system by presenting the problem specification, requirement collection, requirement analysis, requirement organization, and implementation and evaluation plan.

# Chapter 4

## System Design

### 4.1 Introduction

In this chapter, we will present the design of the system, including the definition of components, modules, interfaces, and data, in order to meet specific requirements. The implementation phase focuses on bringing the previous planning and design phase to life and turning it into a functioning chatbot project. We will provide a comprehensive explanation of the project's functionality and overall algorithm.

### 4.2 System Design Specification

The website is divided into several pages such as the main, log-in/sign-in, operator, and admin page. Each page will be demonstrated in this section to show the system design specifications.

#### 4.2.1 Home Page

The home page is the entry page of the website, it will include the main content which is chatbot chat and a log-in/sign-in option is available. It will be provided with a modern design and a user-friendly UI/UX.

#### 4.2.2 Login/Sign-in Page

The user may enter a username and password to log in/sign-in.

#### 4.2.3 Admin Page

The admin page, which is only accessible to authorized users with the necessary permissions, will provide the admin with the ability to upload files required by the chatbot module. Additionally, the page will also allow the admin to enter the academic year schedule by completing a form.

#### 4.2.4 Operator Page

The operator page, restricted only to authorized users with the highest level of permissions, will give the operators the capability to manage other admin users. This includes the ability to add new administrators to the list or remove existing administrators from the list

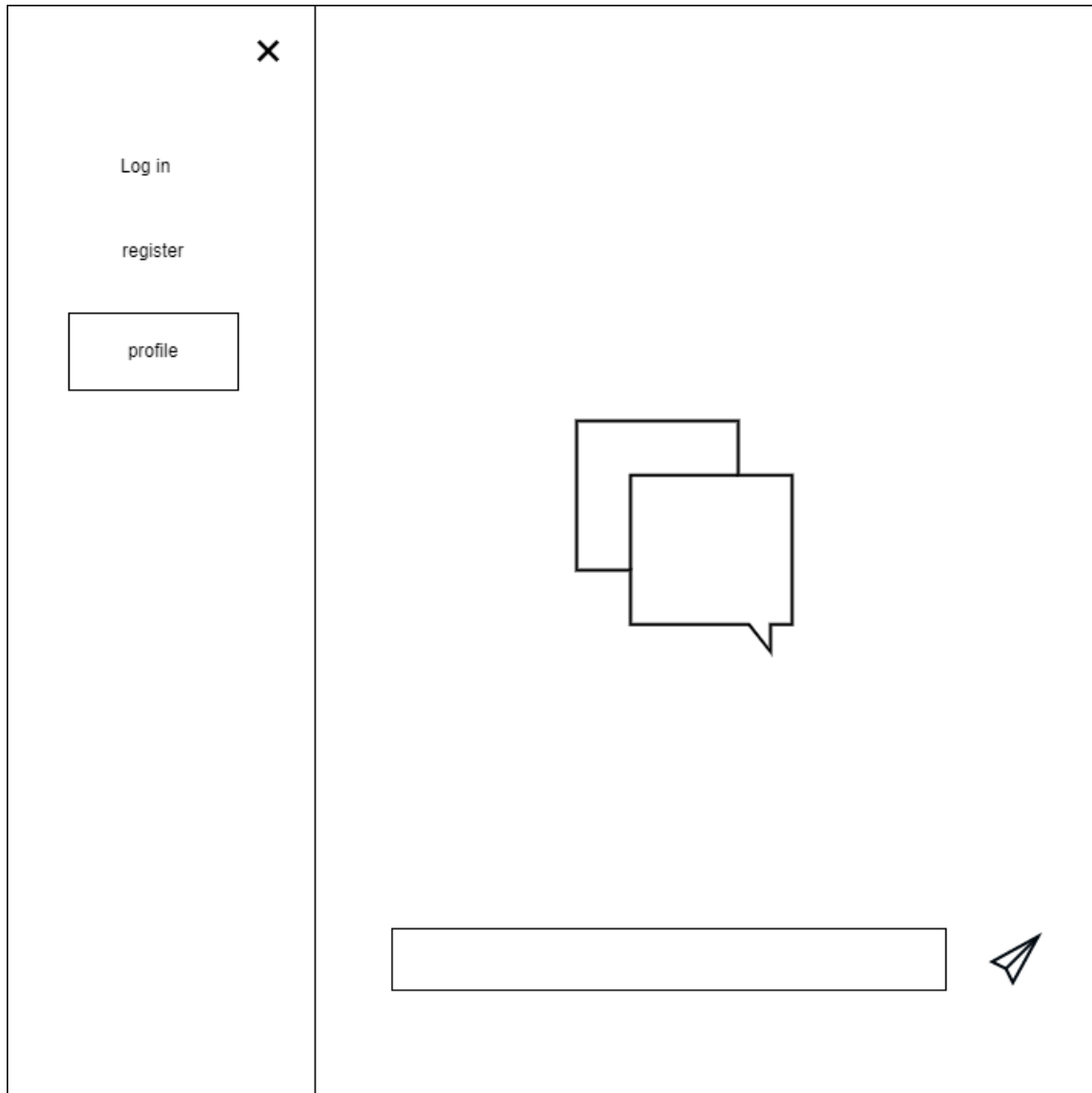


Figure 4.1: An illustration of the homepage main components.

Home

## Log-in

E-mail Address :

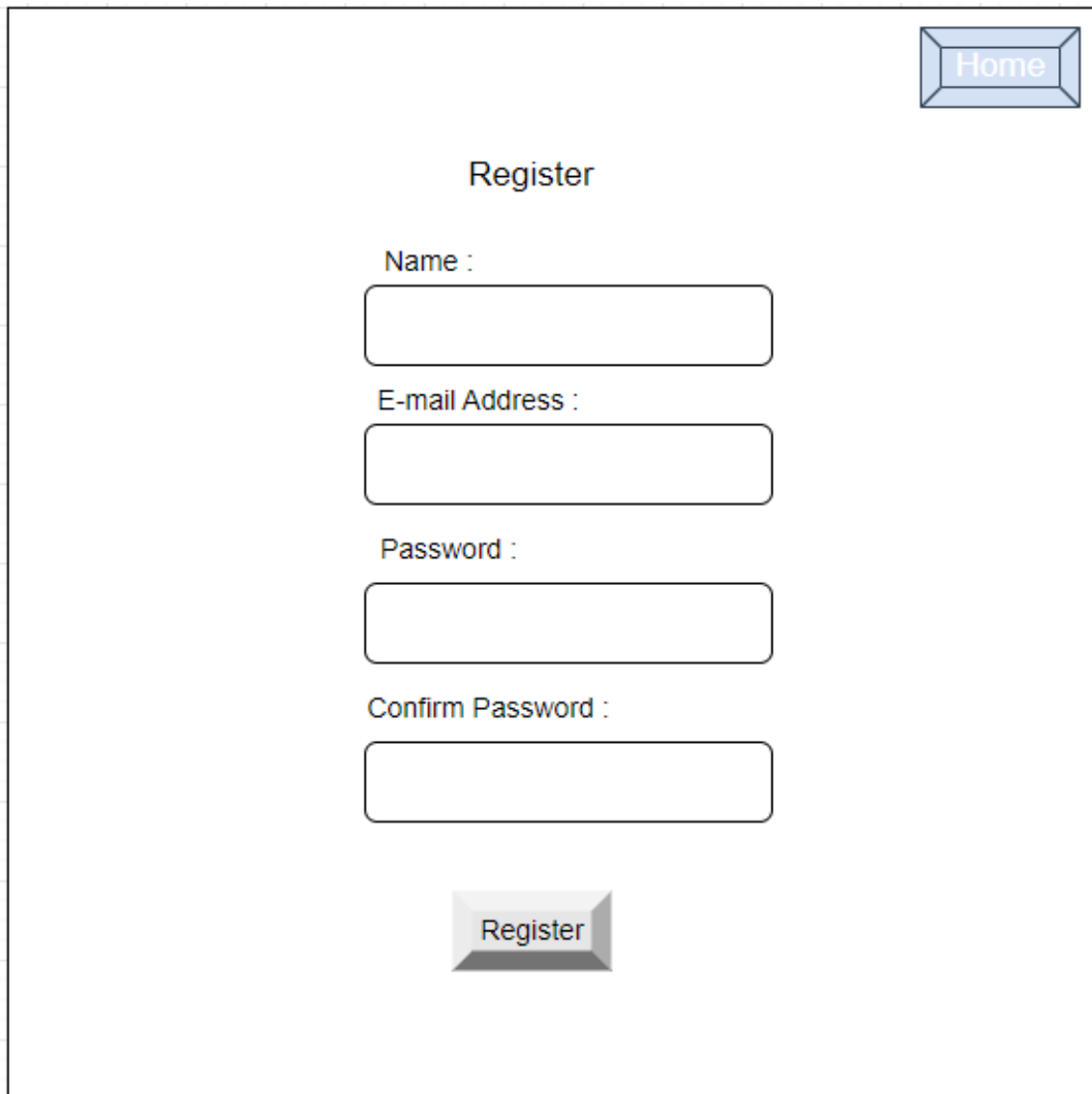
Password :

☐ Remember me

Log-in

Figure 4.2: An illustration of the login page components.





The illustration shows a web page layout for a registration form. In the top right corner, there is a blue button with the text "Home". Centered on the page is the title "Register". Below the title, there are four input fields, each preceded by a label: "Name :", "E-mail Address :", "Password :", and "Confirm Password :". At the bottom center, there is a gray button with the text "Register".

Home

Register

Name :

E-mail Address :

Password :

Confirm Password :

Register

Figure 4.3: An illustration of the register page components.

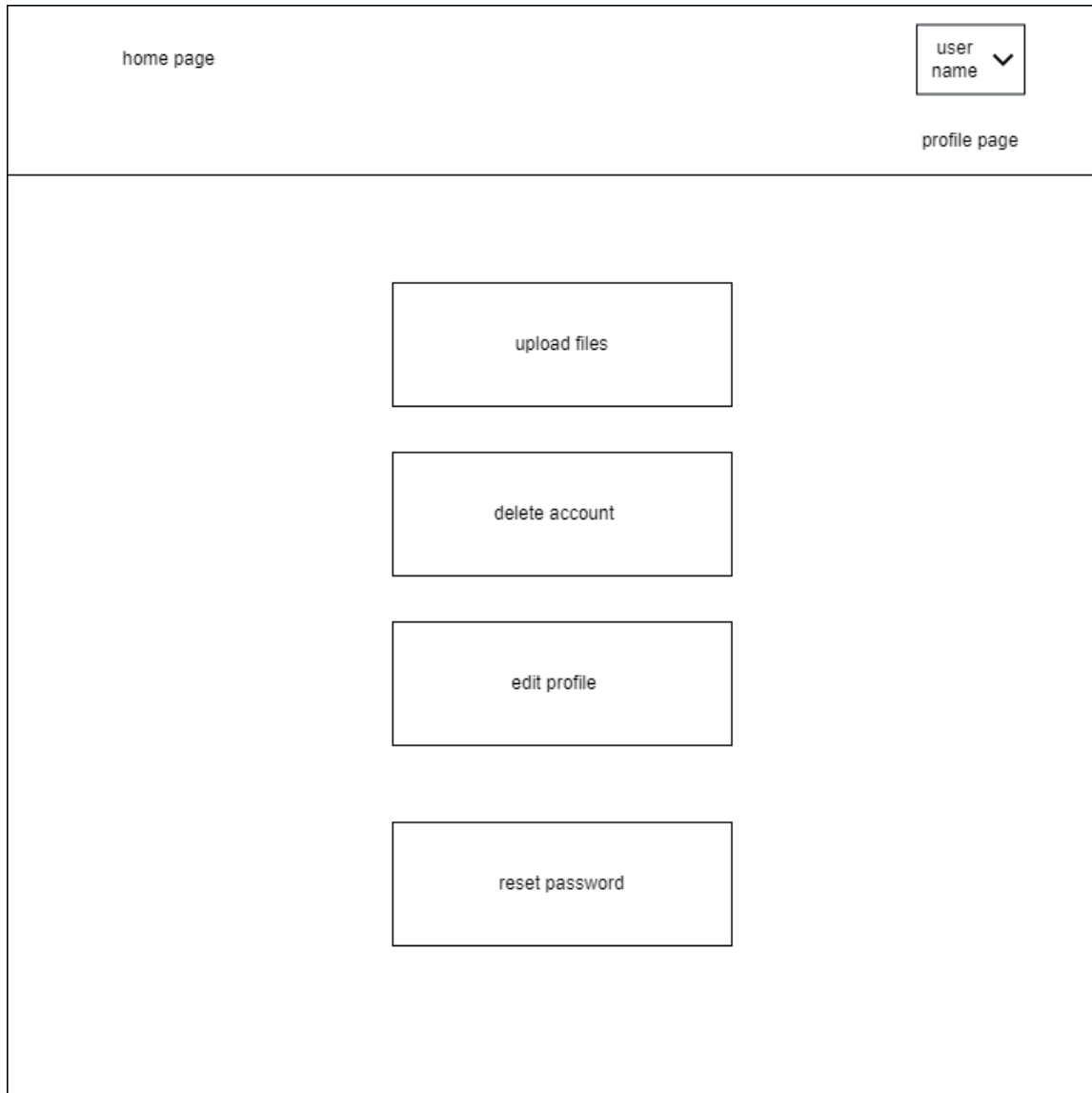


Figure 4.4: An illustration of the profile page components.

home page

user name ▾

admin page

upload college files

the files have been uploaded

academic schedule

academy move:

start

end

mm/dd/yy

mm/dd/yy

●

●

●

submit

Figure 4.5: An illustration of the admin page components.

## 4.3 Design Architecture

Web-based applications generally have many components that work together to allow developers that build such applications to achieve their goals. In what comes next, we demonstrate what components are used and how these components interact with each other.

### 4.3.1 User and Web Application Interaction

The user experience with web-based applications begins with the user interface, where they can interact with the system, make requests, and receive responses. The web server acts as a gatekeeper by verifying the validity of the requests before forwarding them to the Symfony framework. The framework then communicates with the system's modules to process the requests and produce a response. This response is received by the web server through the database and delivered back to the user. The web server also plays an important role in managing uploaded files by users and updating file information in the database, as well as maintaining logs of user activity.

### 4.3.2 Web Application and Chatbot Interaction

- **When a User Send a Message**

We used a pre-trained model from a library called Aravec to convert datasets that include different questions, the user might ask student affairs such as *who is teaching this subject?*, *when is the exam of that specific course?*, so when the user sends any question the system will convert it into a vector (using word2vec) and then will compare it with each question in the dataset we created and choose the highest similarity out of them. Figure 5.2 gives an overall explanation of the whole process.

- **Uploading Files**

For each file that the user will upload, the system will use python for the connection, and this will happen on three different occasions: **First**, the admin will upload a PDF file that has all the information such as the instructor's details, classes and exams. We will use a package that will convert PDF files into CSV files and then clean them. All happen when the administrator uploads a file to the admin page. **Second**, The admin will fill in a form that includes specific term-related dates such as vacation dates, the start of the semester date, and end of the semester. The form will be used to provide information for scheduling purposes for Qassim University. This information will be used as a basis for determining the academic year schedule. **Third**, the user will upload his student's transcript as PDF. And then we will convert PDF files into CSV files. All happen when the user decides that he needs to register on the site to use all the features of the chatbot.

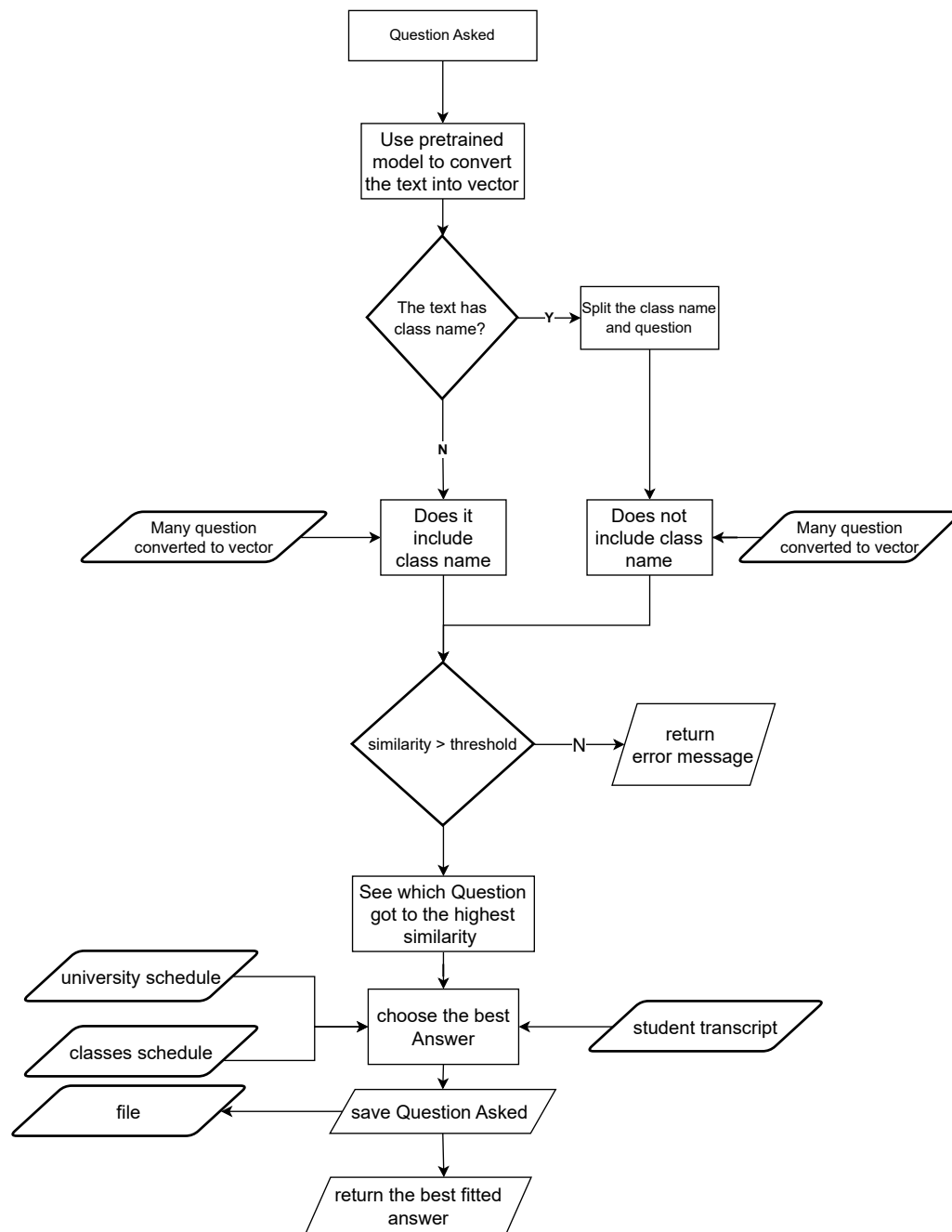


Figure 4.6: The process of getting and responding to the question by the proposed chatbot.

- **Use Case Scenarios**

To demonstrate how the system will be used, we will present two types of user scenarios that would give a clear picture of how the application will look like and how it will be used by all users.

**Admin User Scenario.** Admin users have a crucial role in ensuring that the chatbot provides accurate and up-to-date information to its users. The admin will be able to manage the files utilized by the chatbot modules and monitor their usage. They will also be able to add, update, or delete any necessary files. Additionally, the admin will be tasked with filling in the form for the academic year schedule, which will provide the latest schedules for the chatbot to use in its responses.

**Non-Admin User Scenario.** The chatbot is available for all users to make inquiries and ask questions. Users have the option to log in, which allows the chatbot to identify them and provide personalized answers. When a user makes a request, the chatbot module will try to respond accordingly. However, in certain situations where additional information is required to fulfil the request, the system may prompt the user to log in and upload their academic transcript if it has not been done previously.

## 4.4 Summary

In this chapter, we delved into the design specifications of the proposed application, providing visual representations of the system design through diagrams. A comprehensive analysis of the system architecture was also presented, highlighting the intricacies of each component and how they work together to achieve the desired outcome. To bring the design to life, use case scenarios were demonstrated, providing insight into how the system would work from the user's perspective, and making it easier to understand and visualize the end result. This chapter serves as a blueprint for the implementation of the chatbot application and will guide the development team in creating a seamless and effective system.

# Chapter 5

## System Implementation

### 5.1 Introduction

This chapter will cover the essential components needed to build a successful chatbot, including the data that is required, the different types of data that can be gathered such as the admin file, user file, and admin form, and the possible questions that the chatbot may be asked. Additionally, we will delve into the code structure required for building a chatbot, including the processing file and chatbot model. By the end of this chapter, readers will have a solid understanding of the foundational components required for building the proposed chatbot.

### 5.2 Data

This section covers the technical aspects of the data utilized and gathered by the system. As is the case with all technology systems, particularly intelligent ones, data plays a crucial role in every stage of the system's operation.

#### 5.2.1 Gathered data

We have collected the data in three ways, each of which provided unique insights into the types of questions that users might ask mentioned in section 4.3.2.

- **Admin File**

Admin will upload a file (see 5.1 that will extract most of the questions/answer about classes, and we will have all these possible insights that we can use. In Table 5.1, the \* indicate that these columns will not be used and the rest will not either because the columns do not tell anything or it is not related to students' affair.

- **Admin Form**

Admin will fill out a form to create the dataset that includes the schedule of the semester. The form will have three main columns (I) the name of the event (II) the starting date (III) the ending date.

Table 5.1: Sample of the data available provided by the admin.

Data Type	Description
classNumber	A number for the class indicates the location of subject
subjectName	A short name for the subject such as MATH115
subjectArabic	The Arabic name for the subject
credit hours	Credit hours for the class
practical/theory	Type of the class
*maxStudents	The total number of the max class
*registerdStudents	Number of already register students
days	day of lecture
startingTime	Starting time of the lecture
endTime	Ending time of the lecture
roomNumber	Room of the lecture
InstructorName	Name of the instructor
*ready	Is the subject name ready
ExamDate	Period of the exam

#### • Student File

Students will have the option to upload their student registration to provide a wider kind of questions. This will give us two main columns: subject name and subject name in Arabic. This will give us information for the students to know if he/she can register for a particular subject or not.

### 5.2.2 Possible Questions

When it comes to student affairs, it's crucial to anticipate the range of questions that students may ask. By leveraging the data we've collected, we have created two types of questions, one will be any question that includes a subject name (e.g. MATH115) as Figure 5.2 shown. The other dataset we create is when the question does not include the subject name (see Figure 5.2 as well).

### 5.2.3 Chatbot Requests

Every time a user sends a request to the chatbot model, a new record is added to a table in the Database to track both the user's requests and the model's responses. The data is meticulously constructed to capture as much information as possible from every request.

The data collected includes identifiers for the entry and, if possible, the user (i.e., logged-in users). Additionally, the user's request content is stored in the same table, along with the model's response. If the model successfully produces the intended response, an attribute indicating the response's result, true or false, is set. If the model fails to produce



question	group
اين قاعة	1
وين القاعة	1
اين اجد القاعة	1
مين يدرس	2
مين دكتور	2
من محاضر	2
متى تبدأ محاضره	3
متى محاضره	3
متى اختبار	4
متى موعد اختبار	4
هل استطيع تنزيل	5
هل من الممكن تسجيل	5

Figure 5.1: Sample of quesitons.

a proper response, the system stores the error message if possible, or a null value if not. Therefore, a boolean value that indicates the success result is essential.

Furthermore, the system also tracks two other crucial attributes to monitor the model's performance: similarity percentage and processing time (not response time). These metrics provide valuable information about the accuracy and efficiency of the chatbot model.

Collecting and analyzing this data provides numerous benefits, including identifying areas for improvement in the chatbot model and the system as a whole. With the data being stored in a single table, it's easy to access and analyze, making it a valuable resource for enhancing the system's performance.

متى يبدأ تأجيل الفصل الدراسي	7
متى يفتح التأجيل	7
متى يبدأ الفصل	8
متى الفصل الدراسي	8
متى يفتح الاعتذار	9
الاعتذار متى يفتح	9
متى الاجازات المطولة	10
متى الاجازة الجايه	10
متى يبدأ تغير التخصص	11
متى اقدر اغير التخصص	11
متى تبدأ الاختبارات	12
متى الاختبارات	12

Figure 5.2: Another sample of questions.

## 5.3 Code Structure

Managing and processing is an essential component for any development project and specially when it comes to chatbot in this section we will discuss the technical part in our development from UI, Backend Tools and the chatBot model

### 5.3.1 User Interface (UI)

The website contains six main pages: home, log-in, register, profile, admin, and operator. The home page enables users to communicate with the chatbot and engage in conversations. The log-in/register page allows users to create an account and access features such as uploading files and inquiring about academy records. The admin page provides management with the college schedule by uploading it to the website, as well as defining the academy schedule that includes important dates such as holidays and exam days. The operator page is designed for managing and regulating user rules, such as determining who is an operator, admin, or user.

### 5.3.2 Backend Tools and Frameworks

Laravel offered us a variety of useful backend packages, including the eloquent ORM which enables us to interact with databases in an object-oriented manner. Eloquent provided us with a simple syntax for creating and querying database records, and it also helped us to prevent SQL injection attacks through its built-in sanitization and security

features. Additionally, Laravel's modular architecture allowed us for easy integration of popular tools like Breeze. We used Breeze for user authentication and authorization with features like CSRF protection, throttling of login attempts, and password reset. Breeze also includes optional features like email verification and Two-factor authentication. With the help of Breeze we used bcrypt encryption to keep user passwords secure.

Laratrust, another package for Laravel, provided us with multi-AUTH features for managing access control to different parts of a website. It allowed us the definition of roles and permissions and offers an easy-to-use syntax for checking user permissions. By simplifying the implementation of role-based access control in Laravel applications, Laratrust provides a scalable and flexible solution for managing user permissions. We used Laratrust to construct the 'Admin' and 'Operator' roles which were useful for access control on the website.

In our project, we require integration between PHP and Python operations, which is where Symfony comes in. While we only utilize a single component of this popular PHP web application framework - the Process class - it is a crucial part of our project. The Process class provided us with a secure and reliable way to execute system commands, encapsulating the functionality of the (proc\_\*) family of functions in PHP. With this component, we were able to run our model and pass all of the needed data to it in a secure and predictable manner. Symfony is known for its flexibility, stability, and scalability, and is an excellent choice for web development projects that require complex backend operations.

### 5.3.3 Processing File

For processing the file we used a library called tabula that convert pdf files to CSV files and performs manipulation such as changing the Arabic numbers into English using a library called pyarabic, and data cleaning such as removing rows where with missing values.

- **Admin File**

In the cleaning phase for the admin file some of the rows in the PDF are blank indicates that it is the same as above, so we filled it with the above cell and at the end of the Admin file the days are represented by number for better output we change it to a word representation.

- **User File**

In the cleaning phase for the student file, we moralize the subject name to make it fit with the admin file.

### 5.3.4 Academic Schedules

As previously mentioned, the addition of the academic year schedule is a valuable feature that allows the chatbot model to provide users with information related to dates and times at the university. This feature expands the chatbot's functionality and usefulness.

To ensure that the model is updated with the most current and relevant data, it is the responsibility of the admins to feed and update the model.

To make this process convenient and user-friendly for the admins, a form has been created on the admin page. This form can be filled out at the beginning of the semester or whenever the schedules are updated. The form is quite comprehensive as there are a minimum of 25 dates that can be entered. Since some dates may not apply to every semester, the form includes an option to skip irrelevant entries.

To store the data entered, a popular package called “phpoffice/phpspreadsheet” is used. The entered values are saved into an Excel file, which serves as the source for the chatbot model to retrieve the dates and respond to user inquiries about them. With this system in place, the chatbot model can quickly and accurately respond to user queries about the university’s schedule, providing valuable information in a timely manner.

### 5.3.5 Chatbot Model

The code start with importing the library needed which are (*time, sys, numpy, pandas, re, mysql.connector, and datetime*) all of these libraries will be used latter on. the code then will initialize the value for the input from the user loaded by system value and load the preprocessed data such as the questions, model, year schedule and class schedule. The model will clean and normalize the input then it will perform various tasks such as checking the availability of the user file and admin files, extracting the subject name if available from the user request, checking if the user requests include the subject name or not if so it will compare similarity between the user message and questions that mentioned in fig5.1 and if not it will compare similarity between the user message and questions that mentioned in Figure 5.2, after finding the best similarity it check if it is higher than the threshold or not and responding to the user’s request accordingly, in the end, the responding answer will be saved in a database with similarity, time to execute, and boolean variable indicating if the request was successful or not .

## 5.4 Summary

This section describes the technical aspects of the data used by the system, which plays a crucial role in the system’s operation. The data is collected in three ways, including through an admin file, an admin form, and a student file. Two types of datasets are created based on the data collected, one with subject names and the other without subject names. Every time a user sends a request to the chatbot model, a new record is added to a table in the database to track the user’s requests and the model’s responses. The data collected includes identifiers for the entry and the user, the user’s request content, and the model’s response, along with other crucial attributes like similarity percentage and processing time. Analyzing this data provides valuable information to identify areas for improvement in the chatbot model and the system as a whole.

# Chapter 6

## System Testing and Result Discussion

### 6.1 Introduction

In this chapter, we will represent the results and the evaluation method for the proposed website. In the previous chapter, a comprehensive explanation of the website and model's structure was introduced, it will in this chapter show the results and discuss these obtained results and the website performance

### 6.2 Testing Description and Test Cases

In this section, we will cover the testing phase of the project, which involved verifying the system's functionality and ensuring that it meets the expected requirements. At the beginning of this phase, we conducted tests on local developer environments and collaborated to develop test cases for the system. As we progressed through this phase, we deployed the system onto live servers and continued testing to ensure that the system would work optimally in a public environment.

#### 6.2.1 Threshold

For choosing the best similarity, we started with a minimum of 0.3 using cosine similarity, then we test it by asking many questions and increasing the threshold with taking into consideration having an accurate answer and the practical easy to use bot. after many testing, we reached the end optimum threshold with a 60.

#### 6.2.2 Reduction of the Vocabularies

One of the main requirement analyses mentioned in section 3.3.2 was efficiency, and in testing, we noticed that it was taking almost 3 seconds for the chatbot to respond to

messages in python only. Upon further investigation, we discovered that the pre-trained model we were using (Wikipedia-CBOW) had a large vocabulary of 162,516 words, which was slowing down the chatbot's response time. We decided to reduce the vocabulary using clustering and were able to achieve a significant reduction in response time of almost 50

### 6.2.3 Error Handling

in the testing phase, we notice that 1- some users use requests with a single word such as the only name of the subject, so we decided that we should put a minimum limit to the question asked which will be greater than one word (not containing the class name). 2- another error we notice is if the user enters the wrong course name such as math155 and the subject name does not exist the model will think it is a word and compare it's similarity, so we decided that if the model recognizes any English words that doesn't match any subject name then an error message will be received 3- if the question contains space in the course name such as "CS 181" will not be recognized as subject so this help of RegEx we extract all subjects name from the admin file and removed all letters then we took the unique values from them to have a list of something like ("CS" , "IT" , "COE" , "MATH" ... ) so when user asked any question the model will check if any of these values in the request if so it will delete the space next to it

## 6.3 Results and Discussion

We took a comprehensive approach to the project, starting with the UI, where we utilized various tools to create a minimalist design with a modern look that prioritizes a good user experience. Moving on to the backend, Laravel was the glue that held everything together. It was responsible for managing HTTP requests, handling authentication, authorization, access control, validation, and protection against known attacks like XSS, CSRF, and SQL injection. Additionally, Laravel was the primary component for managing the database and keeping it running smoothly. Laravel, in conjunction with Symfony, facilitated the integration of our Python intelligent model while ensuring security and handling all possible exceptions. The chatbot's brain was developed with the help of Python's powerful tools. Finally, to celebrate the project's success, we deployed the system to AWS servers on a Ubuntu server. We also used AWS's RDS service to manage and preserve our database with weekly backups. You can access the website using the URL <sup>1</sup>.

### 6.3.1 Chatbot Results

After a series of tests, we are confident with the performance of the chatbot, its response time, and its ability to handle multiple conversations simultaneously. The chatbot

---

<sup>1</sup><http://chatcoc.com/>

demonstrated a high level of accuracy in responding to students' inquiries and a fast response with an average response time of **1.5 seconds** and these are examples of how the model responded to questions.

Question	Response	Similliraty	time
متى موعد الاختبار الخاص بمادة arab101	وقت الاختبار في الفترة (9)	0.84	1.851
في اي وقت يفتح تاجيل المواد	يبدأ التأجيل من 26-11-2022 الى 04-12-2022	0.73	1.769
متى تبدأ محاضرة CS 452	شعبه 761 يوم الخميس تبدأ من 11:30 ص الى 12:45 م شعبه 761 يوم الاحد تبدأ من 11:30 ص الى 12:45 م	0.65	1.330
متى تبدأ جميع الاجازات	الاجازات تكون في الايام الاتيه '18-12-2022', '22-02-2023', '16-01-2023', '15-01-2023', '23-02-2023'	0.91	1.505
هل أستطيع تنزيل مادة math 115	نعم يمكنك تنزيل المادة... لا يوجد لها متطلب	0.83	1.415
هل يمكنني تنزيل cs182	لا يمكنك	0.90	1.135
من هو الدكتور الذي يدرس مادة coe122	يدرسها , نوح تولتش	0.66	1.756
اين ممكن اجد قاعة cs221	شعبه (739) في قاعه رقم (101) شعبه (5205) في قاعه رقم (lab-123)	0.74	1.272

Figure 6.1: This is the home page's main components.

### 6.3.2 UI Result

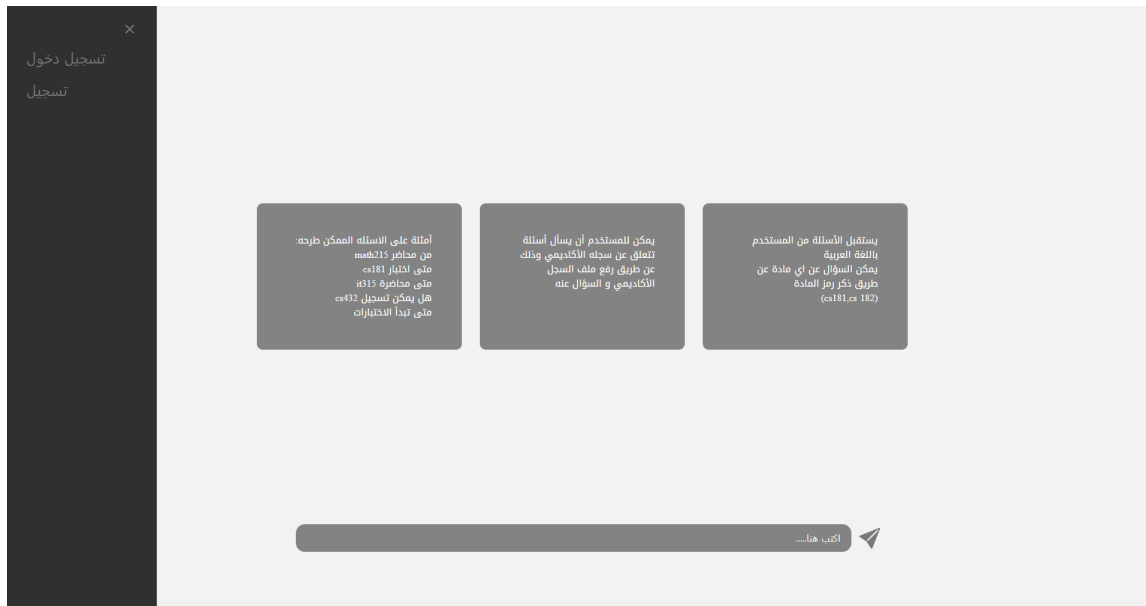


Figure 6.2: Home page's main components.

### 6.3.3 Deployment Result

When it was time to deploy our project on live servers, we opted for AWS services. We started by creating a fresh EC2 instance and then proceeded to install the necessary services. The first step was to install the Laravel base requirements according to the Laravel documentation<sup>2</sup>. In addition to the base requirements of Laravel, we needed some more services installed to ensure that our project runs properly. These included Nginx HTTP Server, PHP-FPM, Composer, MySQL PHP Extension, and pyenv, which was useful for managing versions of python on the server.

After installing the required services, we configured the Nginx HTTP server. To follow best practices, we created a separate user on the server where all the site resources are stored and managed. We also configured permissions and the Nginx server based on the new user.

Next, we created a new RDS instance to separate the Database server on a separate instance and limit access to the RDS to the instance to be only possible to and from the EC2 instance where our site will be running. For MySQL server configuration, we created a separate user with proper privileges.

We then installed all the Python modules needed by our system and finished the deployment process by installing the site itself on the server, which was a relatively easy

<sup>2</sup><https://laravel.com/docs/9.x/deployment#server-requirements>



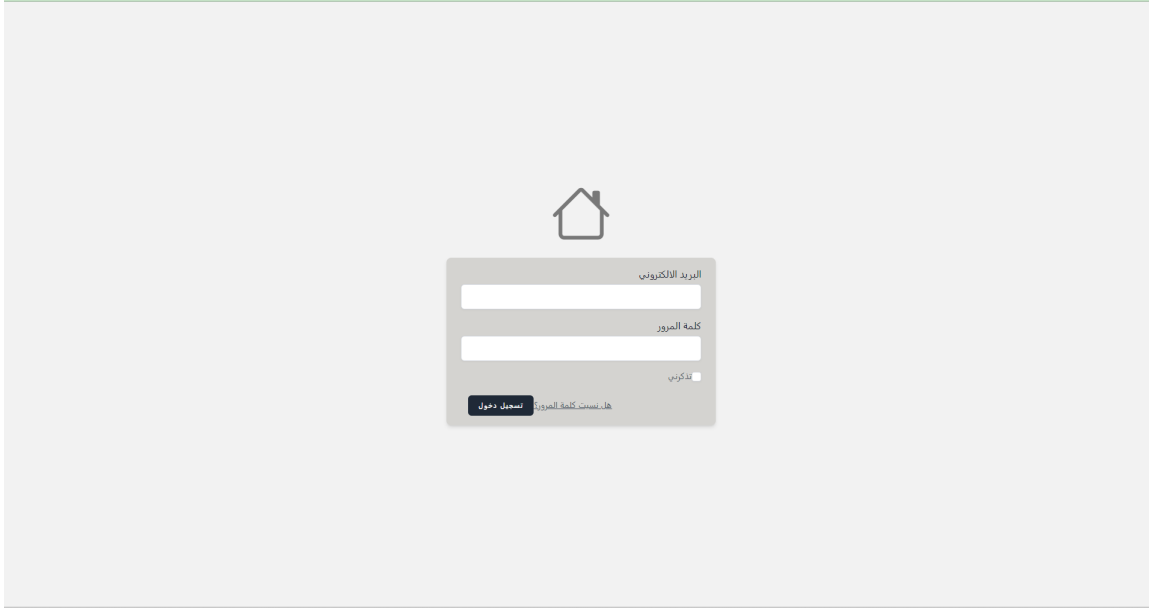



Figure 6.3: Log-In page.

step. Overall, the deployment process was done with great care to ensure the stability, security, and efficiency of the live site.



The image shows a registration form on a light gray background. At the top center is a house icon. Below it is a form with four input fields and a submit button. The labels for the fields are in Arabic: 'الاسم' (Name), 'البريد الإلكتروني' (Email), 'كلمة المرور' (Password), and 'تأكيد كلمة المرور' (Confirm Password). The submit button is labeled 'تسجيل' (Register).

الاسم
<input type="text"/>
البريد الإلكتروني
<input type="text"/>
كلمة المرور
<input type="password"/>
تأكيد كلمة المرور
<input type="password"/>
<input type="button" value="تسجيل"/>

Figure 6.4: Register page.

badr
الصفحة الرئيسية
الملف الشخصي

**ملف المستخدم الشخصي**

يمكن تحميل ملف (Chatbot) الشخصي الخاص بك إلى pdf. لم يتم تحميل ملف

حفظ

**حذف الحساب**

يتم حذف حسابك - سيتم حذف جميع موارده وبياناته نهائيًا. قبل حذف حسابك ، يرجى تنزيل أي بيانات أو معلومات ترغب في الاحتفاظ بها

حذف الحساب

**الملف الشخصي**

قم بتحديث معلومات الملف الشخصي لحسابك وعنوان البريد الإلكتروني.

الاسم

badr

البريد الإلكتروني

badrdrv9@gmail.com

حفظ

يتم حذف حسابك - سيتم حذف جميع موارده وبياناته نهائيًا. قبل حذف حسابك ، يرجى تنزيل أي بيانات أو معلومات ترغب في الاحتفاظ بها

حذف الحساب

**الملف الشخصي**

قم بتحديث معلومات الملف الشخصي لحسابك وعنوان البريد الإلكتروني.

الاسم

badr

البريد الإلكتروني

badrdrv9@gmail.com

حفظ

**تحديث كلمة المرور**

تأكد من أن حسابك يستخدم كلمة مرور طويلة وعشوائية للبقاء آمنة

كلمة المرور الحالية

كلمة المرور الجديدة

Figure 6.5: Profile page.

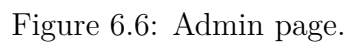


Figure 6.6: Admin page.

## 6.4 Summary

This chapter presents the testing phase and results of the proposed website. The testing phase involved verifying the system's functionality and ensuring that it meets the expected requirements. Test cases were developed, and the system was deployed onto live servers for optimal performance. The chapter discusses the threshold, the reduction of vocabularies number, and error handling. The chatbot showed a high level of accuracy and fast response time, with an average response time of 1.5 seconds.

# Chapter 7

## Conclusion and Future Work

### 7.1 Conclusion

A website's success depends on its UI, which includes visual design, layout, and interactive elements that can impact user engagement and retention. We ensure that the interface we provide is easy to use and has a straightforward layout. ensure smooth and seamless interaction with users.

The user experience (UX) is critical for the success of a website. UX refers to the overall satisfaction and usability of the website, which impacts user engagement and return rate. Our objective is to establish a seamless, smooth, and effortless interaction between the platform and its potential students/users by continuously enhancing the UX through user feedback. Compatibility of a chatbot website with multiple devices is critical for reaching a wider audience and improving user experience. We ensure that our website is compatible with various platforms including PCs, phones, tablets, and other devices.

The practical implementation of a chatbot website requires the development of a functional and user-friendly platform that addresses user needs. We aim to ensure that the website is easy to use and minimizes the amount of time and effort needed.

We were fortunate to be able to deploy our project on live servers that provide uninterrupted services 24/7. even though, we were limited by the lack of an API that could automatically feed the model we managed to minimize the need for human intervention to a minimum.

Keeping logs of requests to the model was an integral part of the project, and we made sure to collect as much data as possible from these requests. In addition, the integration between PHP and Python was successfully achieved with the help of the Symfony framework. Security was a significant requirement for our project, and we were able to accomplish it by utilizing several helpful packages that secure sensitive data, such as user passwords, and protect against CSRF, SQL injection, and XSS.

we made sure that the no failure will happen during the request of the message and by using threshold we are confident with the performance of the model that it will give a dedicated answer for each user and by decreasing the number of vocabulary we occurred efficiency

## 7.2 Future Work

While we are proud of the results achieved in this project, there is always room for improvement. As with any technology, a chatbot is a constantly evolving tool that requires ongoing refinement and optimization to ensure it remains effective and useful for users. In this section, we outline some potential avenues for future work that could build upon the successes of this project and address some of the areas where the chatbot could be improved.

### 7.2.1 Conversation Based

Our chatbot model currently only processes one request from the user and provides a single response. While this approach keeps the model simple, it would be more helpful if the chatbot could remember previous messages and classify them accordingly. For example, instead of providing all classes related to a subject, it would be more useful if the chatbot could identify the specific class the user is asking about. Similarly, identifying the specific holiday the user is asking about would be more helpful than providing information on all holidays.

### 7.2.2 Expanding the Scope of Questions

our chatbot model currently only processes the subject name if written in English this is making it easier to normalize all possible questions so when the user has a request with the name of the subject in English we can easily extract it and then compare similarities if we had more time we can extract the Arabic test.

### 7.2.3 API

As our chatbot expands, one solution to reduce human integration is by utilizing an API that can link directly with the schedules of each student and their corresponding classes. in this way, we would eliminate the need for human intervention in the administrative aspect of our system, allowing a fully automated process.

# References

- [1] E. M. Nuss, “The development of student affairs,” *Student services: A handbook for the profession*, vol. 4, pp. 65–88, 2003.
- [2] B. R. Ranoliya, N. Raghuwanshi, and S. Singh, “Chatbot for university related faqs,” in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, IEEE, 2017, pp. 1525–1530.
- [3] D. K. Deardorff, H. De Wit, J. D. Heyl, and T. Adams, *The SAGE handbook of international higher education*. Sage, 2012.
- [4] J. Jia, “The study of the application of a keywords-based chatbot system on the teaching of foreign languages,” *arXiv e-prints*, cs-0310018, 2003.
- [5] A. M. Turing, “Computing machinery and intelligence (1950),” *The Essential Turing: the Ideas That Gave Birth to the Computer Age*, pp. 433–464, 2012.
- [6] R. Agarwal and M. Wadhwa, “Review of state-of-the-art design techniques for chatbots,” *SN Computer Science*, vol. 1, no. 5, pp. 1–12, 2020.
- [7] S. A. Thorat and V. Jadhav, “A review on implementation issues of rule-based chatbot systems,” in *Proceedings of the International Conference on Innovative Computing & Communications (ICICC)*, 2020.
- [8] C. Miura, S. Chen, S. Saiki, M. Nakamura, and K. Yasuda, “Assisting personalized healthcare of elderly people: Developing a rule-based virtual caregiver system using mobile chatbot,” *Sensors*, vol. 22, no. 10, p. 3829, 2022.
- [9] D.-H. Kim, H.-S. Im, J.-H. Hyeon, and J.-W. Jwa, “Development of the rule-based smart tourism chatbot using neo4j graph database,” *International Journal of Internet, Broadcasting and Communication*, vol. 13, no. 2, pp. 179–186, 2021.
- [10] J. Cahn, “Chatbot: Architecture, design, & development,” *University of Pennsylvania School of Engineering and Applied Science Department of Computer and Information Science*, 2017.
- [11] G. Caldarini, S. Jaf, and K. McGarry, “A literature survey of recent advances in chatbots,” *Information*, vol. 13, no. 1, p. 41, 2022.
- [12] R. Shetty, A. Bhosale, P. Verma, and A. Phalke, “Ai based healthcare chatbot,” 2022.



- [13] R. Yan, Y. Song, and H. Wu, “Learning to respond with deep neural networks for retrieval-based human-computer conversation system,” in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2016, pp. 55–64.
- [14] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [16] D. Adiwardana, M.-T. Luong, D. R. So, *et al.*, “Towards a human-like open-domain chatbot,” *arXiv preprint arXiv:2001.09977*, 2020.
- [17] S. S. Ranavare and R. Kamath, “Artificial intelligence based chatbot for placement activity at college using dialogflow,” *Our Heritage*, vol. 68, no. 30, pp. 4806–4814, 2020.