technologies

MDPI

Article

SQUbot: Enhancing Student Support Through a Personalized Chatbot System

Zia Nadir [1,*], Hassan M. Al Lawati [1], Rayees A. Mohammed [1], Muna Al Subhi [1] and Abdulnasir Hossen [1,2,*]

[1] Department of Electrical and Computer Engineering (ECE), Sultan Qaboos University (SQU), Muscat 123, Oman; hlawati@squ.edu.om (H.M.A.L.); rayees80@gmail.com (R.A.M.); muna9525aa@gmail.com (M.A.S.)

[2] UNESCO Chair on AI, Communications and Information Research Center (CIRC)-Sultan Qaboos University (SQU), Muscat 123, Oman

* Correspondence: nadir@squ.edu.om (Z.N.); abhossen@squ.edu.om (A.H.)

Abstract

Educational institutions commonly receive numerous student requests regarding various services. Given the large population of students in a college, it becomes extremely overwhelming for the staff to address the inquiries of all the students while dealing with multiple administrative tasks at the same time. Furthermore, students often make multiple visits to the university's administration, make multiple calls, or write emails about their concerns, which makes it difficult to respond to their queries promptly. AI-powered chatbots can act as virtual assistants that promptly help students in addressing their simple and complex queries. Most of the research work has focused on chatbots supporting the English language, and significant improvement is needed for implementing chatbots in the Arabic language. Existing studies supporting the Arabic language have either employed rule-based models or built custom deep learning models for chatbots. Rule-based models lack understanding of diverse contexts, whereas custom-built deep learning models, besides needing huge datasets for effective training, are difficult to integrate with other platforms. In this work, we leverage the services offered by IBM Watson to develop a chatbot that assists university students in both English and Arabic. IBM Watson employs natural language understanding and deep learning techniques to build a robust dialog and offers a more scalable, integrable, and customizable solution for enterprises. The chatbot not only provides information about the university's general services but also customizes its response based on the individual needs of the students. The chatbot has been deployed at Sultan Qaboos University (SQU), Oman, and tested by the university's staff and students. User testing shows that the chatbot achieves promising results. This first bilingual AI chatbot at SQU supports English and Arabic and offers secure, personalized services via OTP and student email verification. SQUbot delivers both general and individualized academic support. Pilot testing showed 84.9% intent recognition accuracy. Most unidentified queries were due to dialectal variation or out-of-scope inputs, which were addressed through fallback prompts and dataset refinement.

Keywords: chatbot; IBM Watson; intents; entities; dialog

check for updates

Academic Editors: José Fernández-Cerero, Marta Montenegro-Rueda and Hongying Meng

Received: 25 June 2025
Revised: 9 August 2025
Accepted: 2 September 2025
Published: 15 September 2025

Citation: Nadir, Z.; Al Lawati, H.M.; Mohammed, R.A.; Al Subhi, M.; Hossen, A. SQUbot: Enhancing Student Support Through a Personalized Chatbot System. *Technologies* **2025**, *13*, 416. https://doi.org/10.3390/technologies13090416

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Educational institutions often receive a large volume of queries from students concerning different services such as course registration, courses being offered, admissions, advising, and others. Chatbots are currently applied to a variety of different fields and

applications, spanning from education to e-commerce, encompassing healthcare and entertainment [1]. Students need timely responses to their queries, particularly during peak periods such as registration periods, exam times, and so on. Addressing such a large number of inquiries manually can be extremely time-consuming and exhausting, given the limited number of human resources. University staff and administrators who are already engaged in managing various administrative and strategic tasks may find it quite overwhelming and strenuous to deal with the diverse inquiries of a large student population promptly. In addition, students sometimes need to visit the college admin to ask about their concerns, make phone calls, or write emails, which require their time and effort. Artificial intelligence can redefine the way students' queries are handled by automating the process of answering users' queries. AI-powered chatbots have long been used in various domains, such as healthcare, tech-related fields, the educational sector, and others, to support users' concerns in a timely and effective manner. In the field of education, different colleges and universities employ intelligent conversational agents that can interact with users in natural language. Martha, a chatbot employed by George Washington University, helps students connect to the university's wireless network, register devices, and get help with other inquiries by allowing the students to interact via text in natural language (https://www.bmc.com/blogs/introducing-martha-leveraging-cognitive-automation-to-create-an-exceptional-student-experience/, accessed on 1 September 2025). Georgia State University designed an AI-powered Chatbot known as Pounce (https://news.gsu.edu/2022/03/21/classroom-chatbot-improves-student-performance-study-says/, accessed on 1 September 2025) to facilitate students with the registration process. The chatbot employed by California State University (https://www.csusb.edu/admissions/apply-csusb, accessed on 1 September 2025) assists students by providing information about career advice centers, financial aid, courses, and more. So, in a nutshell, a chatbot is needed for the following:

1. Newly admitted students are not aware of the academic regulations;
2. Some students feel shy about visiting advisors;
3. Some students have language issues and can't express their requirements directly to advisors;
4. Sometimes students need an immediate answer that advisors cannot provide due to their schedules and commitments.

Chatbots are broadly classified into rule-based and learning-based approaches. Rule-based methods use a set of predefined rules to respond to queries. These types of chatbots work well when the scope of discussion is limited to specific topics [2]. Chatbots only identify keywords from an input sentence and match them with the rules defined to create responses [3]. Learning-based approaches typically employ deep learning approaches and are further categorized as retrieval-based and generative approaches. Retrieval-based models are capable of learning the context of the conversation. These types of chatbots respond by selecting the most appropriate response from a set of predefined responses using some heuristics [4,5]. On the other hand, generative models generate responses by creating new sentences based on past and current experiences [6]. Despite advances in deep learning that allow chatbots to deliver human-like responses, significant improvement is still needed to support complex behavior [7,8]. Various research articles report on the development of academic chatbots that support the English language [9–12], while there is limited work on the Arabic language. Existing studies use pattern matching [13] and keyword matching approaches [14] to assist Arabic-speaking students with their college inquiries. Both of these approaches rely on a set of predefined patterns and keywords that fail to cater to the morphological richness of the Arabic language and understand the context, semantics, and diverse nuances of the language. In Ref. [15], the authors use

deep learning models available in the RASA framework to develop a chatbot to provide generic information to the students, such as about the admission process, financial aid, and more. However, this chatbot serves only the generic inquiries of the students and does not accommodate students' personalized inquiries. In Ref. [16], deep neural networks are used to encode user intents in the conversation, allowing the users to inquire about their academic and college services more effectively. However, building an efficacious conversational model programmatically entails specific challenges in terms of needing a large amount of annotated data and extensive experimentation. Fine-tuning complex algorithms could be computationally expensive and time-consuming. Scaling such models to handle large amounts of traffic and integrating these models into existing platforms may also be quite challenging. Acquiring the services of established platforms such as IBM Watson, Google, or Amazon can overcome these challenges by offering more practical and cost-effective solutions.

Recent advances in conversational AI have enabled educational institutions to automate student services using intelligent chatbots. However, developing a bilingual chatbot that can serve users in both English and Arabic remains a challenge, particularly due to the complexities of Arabic Natural Language Processing (NLP). This study presents the design and deployment of a bilingual AI-based chatbot for Sultan Qaboos University (SQU), aimed at assisting students with common administrative and academic queries.

To frame this work as a research investigation, we address the following questions.

- How can a bilingual AI chatbot be designed to support Arabic and English language queries effectively in a university setting?
- What are the technical and linguistic challenges of implementing Arabic NLP using IBM Watson, and how can they be addressed?
- How effective is the chatbot in responding to real-world student queries, and what are the limitations of its current implementation?

By exploring these questions, our study aims to contribute to the development of intelligent multilingual student support systems in higher education. Accordingly, we employ IBM Watson Assistant to design and develop a chatbot for educational institutes supporting both English and Arabic. Unlike custom-built AI models, IBM Watson offers more customizable, scalable, and easily integrable solutions leveraging advanced natural language understanding (NLU) and deep learning techniques to ensure intelligent and robust conversations. The chatbot has been deployed for Sultan Qaboos University (SQU), Oman, and is hence named SQUbot. The bot can effectively chat about the simplest to the most complex queries. Not only can the bot converse about the general services provided by the university, but it can also serve the personalized needs of individuals. The services of the bot, available 24/7, can provide timely responses to users, saving time and effort for both university staff and students. The dataset required for the bot comes from two different sources. The data needed to address general services requested by the users is extracted from the university's website. In contrast, the data required to serve the personalized student requirements is fetched from the university's data center. This work contributes to the following:

- Development of a robust chatbot using IBM Watson Assistant supporting both English and Arabic languages;
- Capability of understanding a wide range of simple and complex queries;
- Ability to customize responses based on the individual needs of the students;
- Development of an API to connect the IBM Watson cloud to the university's data center to maintain users' data security and privacy.

2. Literature Review

ELIZA, the first chatbot in the history of computers, was created by a German scientist in 1966 [17]. It uses a pattern-matching approach for conversations and is unable to learn the context. Another chatbot called PARRY was created in 1971 and is considered to have a better control structure than ELIZA [2]. Verma et al. [18] developed an Artificial Linguistic Internet Computer Entity (ALICE) utilizing NLP and pattern-matching algorithms. ALICE lacks an understanding of the context of conversations and hence fails to respond like humans. Siri, a virtual assistant created by Apple, uses natural language to interact with the users and respond to their voice queries [18]. Siri is capable of understanding many languages. However, support for navigational instructions is available only in the English language. IBM's Watson assistant employs NLP and machine learning algorithms to provide answers to the users' questions [19]. Watson can also extract insights from previous conversations.

The development of conversational agents by educational institutes has gained attention as a means of supporting administrative resources and providing prompt assistance to students by addressing their academic and personal inquiries. Chatbots can be developed by building Artificial Intelligence and Natural Language Processing (NLP) models programmatically or by utilizing conversational agents such as Google Assistant, Microsoft's Cortana, XiaoIce, Amazon's Alexa, Apple's Siri, IBM Watson, and others [20]. Unlike the custom-built models, the established platforms offer more scalable, robust, and cost-effective solutions for enterprises.

There are various types of chatbots, as well as different platforms and frameworks. Some are presented in the section below. Table 1 summarizes the limitations of the existing studies on chatbots supporting the Arabic language.

Table 1. Type of Chatbot.

Chatbot Type	Rule-Based	AI and ML	Hybrid Model
Definition	Provide a set of predefined options. The user has to select the appropriate option. Great for smaller numbers and straightforward queries. Guided by a decision tree. Can provide personalized responses based on user identification.	Use natural conversation. Use NLP technology to understand the user query and serve with an appropriate response. Use ML to remember user chats and learn from them to improve and grow over time. Can provide personalized responses based on user identification.	Use the simplicity of a rule-based chatbot. Use the sophistication of an AI and ML-based chatbot. Cater to customers based on the situation and query. Integrate with back-end systems to provide personalized transactions.
Types	Menu-based chatbot and linguistic-based chatbot.	Keyword recognition-based chatbots and machine learning chatbots	-
Example	Healthcare bot for appointments, booking bot to book rooms and services, and travel bot for flight bookings.	Retail support bot, banking bot, telecom bot and orders, deliveries, and logistics bot	Education-course bot, automotive lead generation bot, social media marketing bot

There are different methods of developing a virtual assistant, and each has advantages and disadvantages. Table 2 summarizes the overall platforms and the frameworks. Figure 1 presents a simple comparison that was conducted to help us determine which method is most suitable for our requirements in creating the research deliverables.

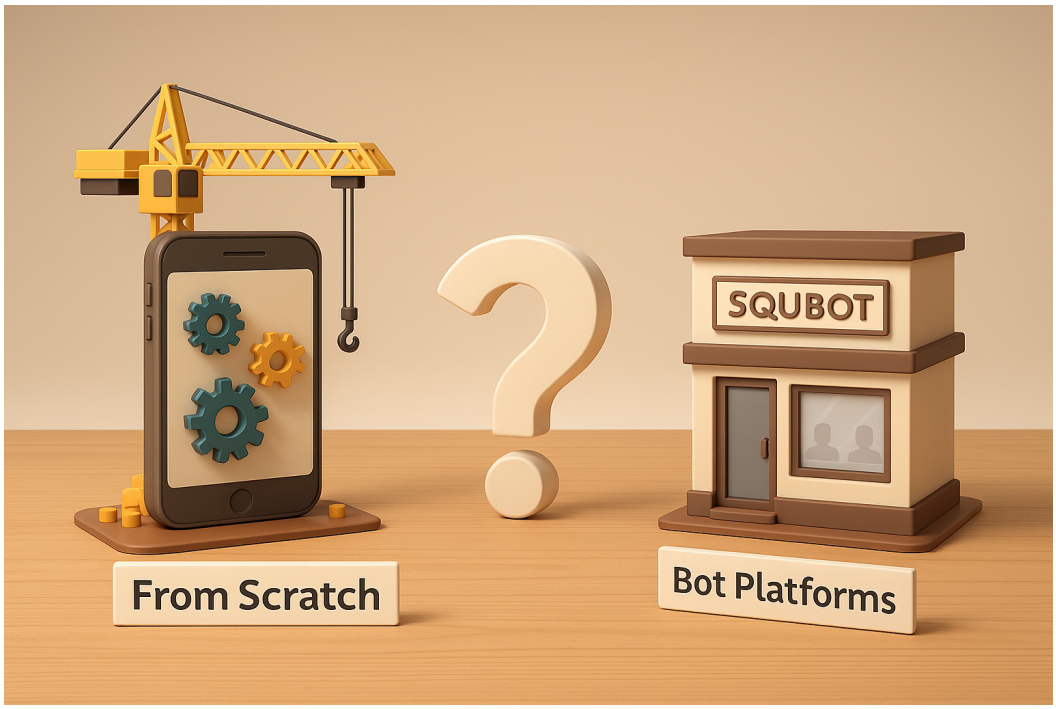

Figure 1. Platforms and frameworks.

Table 2. Platforms and frameworks.

Frameworks	From Scratch	Ready to Use	Hybrid
Features	- Difficult - Requires programming - More time - SDKs and libraries - Free and open source - Need hosting - Full Control	- Easy to use - No programming - Less time - Online interface - Paid - Hosting-ready - Restricted	Scratch/Readymade
Examples	- NLTK [Python] - Botkit [Node.js-SDK] - Chatterbot [Python] - RASA [Python] - Botpress [Type-Script]	Mobile Monkey Botsify Chatfuel Manychat Flowxo	- OneReach.ai - Microsoft Bot - IBM Watson - Google Dialogflow - Amazon Lex - Wit.ai

Most of the studies in the domain of academic conversational agents support the English language. Reference [9] developed a chatbot using IBM Watson, which assists the students by answering their questions about the facts regarding a college or university before enrolling. Reference [11] reports the development of a chatbot using IBM Watson, which is capable of addressing users' queries regarding exam stress. Reference [10] used Amazon Lex to develop a chatbot by recognizing intents in the user queries and returning a response from the database, consisting of student questions and answers. The authors reported that the chatbot successfully handled students' course-related queries, particularly focusing on addressing assignment extension requests. Reference [12] presented an AI-based conversational bot to answer students' queries regarding the admission process. The bot is built using neural networks and is capable of handling simple to complex queries in the English language. Reference [21] used DialogGlow to implement a chatbot for placement activity. The proposed model requires structured data handling for each of the predefined dialogs and is unable to handle semantic queries. Table 3 summarizes the limitations of the existing studies on chatbots supporting Arabic.

Table 3. Limitations of the existing Arabic chatbots.

Papers	Language	Technique	Limitations
[13]	Arabic	Pattern Matching	- Relies on limited rule set - Does not cater context and semantics - Addresses only general services
[14]	Arabic	Keyword Matching	- Relies on predefined keywords - Does not cater context and semantics
[15]	Arabic	RASA framework	- Addresses only general services
[16]	English + Arabic	Deep neural network	- Resource intensive - Less scalable - Addresses only general services
[22]	Arabic	Keyword matching + String distance comparison algorithm	- Partially relies on keywords - Addresses only general services

Some studies have also focused on developing academic chatbots supporting the Arabic language. Reference [13] used a rule-based, pattern-matching approach to develop a chatbot for Applied Science University in Jordan. The bot acts as an information point advisor supporting native Arabic students. However, pattern-matching technique-based chatbots have certain drawbacks, as there are numerous ways to construct a user's utterance in natural language. Given the morphological richness of a language, it is not feasible to cover all possible cases of sentence construction. Reference [14] developed an Arabic language chatbot using a keyword-matching approach to match the users' queries against the queries stored in the database. If the incoming query matches any of the queries in the database, the relevant response is returned to the user based on the query type, which could either be static or non-static.

The static query refers to a generic question that has the same answer for all the users, while the non-static query requires a response according to the user's personal needs. However, the keyword-matching approach relies on predefined keywords and fails to understand the context, semantics, and linguistic nuances of a language. Reference [22] developed an Arabic chatbot supporting the Jordanian dialect. They used a combination of a keyword-matching algorithm and a string distance comparison algorithm to retrieve the results from the database relevant to the user's query. The chatbot supports administrators by reducing their workload and helps regular users/visitors by promptly responding to their queries about the university. Reference [15] employed deep learning models available within the RASA framework to develop an AI-powered chatbot. They incorporated a customized NLP pipeline for the Arabic language for classifying intents, extracting entities, and receiving responses. The bot efficiently addresses inquiries regarding generic university services, such as the admission process, student services, and financial aid. Reference [16] developed a bilingual chatbot supporting both English and Arabic languages. They employed a deep neural network to encode user intents, enabling the bot to converse about various topics, including but not limited to registration, summer courses, and semester exams. However, the bot is capable of addressing only general academic or college inquiries, ignoring requests related to the personal needs of the student.

This work aimed to address issues such as dialect variation, limited training, and lack of secure personalization while building on prior efforts. To provide a clearer context, we have structured this review thematically and employed varied sentence constructions and comparative analysis to present a more cohesive overview of related contributions.

3. Technical Implementation Details

3.1. Training Data Configuration

- Our chatbot was configured with over 90 distinct intents, designed to cover both general inquiries and personalized services relevant to university life.
- Each intent was trained using 15–25 sample sentences, created carefully in both English and Arabic, to reflect a wide range of natural student phrasing, including the usual local jargon used within the university.
- The overall training data consisted of the following:
  - Around 2500 manually extracted entries from the SQU website for general services.
  - 45 intents tailored specifically for Arabic, incorporating cultural and linguistic nuances relevant to the region.
  - For personalized services, in addition to intents and entities, the chatbot relied on real-time integration with the university's internal systems through a secure API. API Payload was prepared by extracting relevant data from the conversation through entities.

3.2. Watson Assistant Pipeline Configuration

- We used IBM Watson's Natural Language Understanding (NLU) engine as the core of the assistant.
- Intent recognition was based on a confidence-scoring model, with a custom threshold of 0.65 (higher than the default of 0.2) to strike a balance between precision and fallback handling.
- Over 15 custom entities were defined (e.g., @empname, @coursecode, @degreetype), each populated with relevant synonyms and variants to enhance recognition accuracy.
- The dialog flow was structured using over 200 nodes, combining condition-based branching, fallback prompts, and multilingual support, including button-based disambiguation where appropriate.

3.3. Challenges in Arabic NLP

Supporting Arabic introduced unique complexities, which we encountered and addressed during development:

- Morphological richness: Arabic's structure required careful synonym expansion and multiple phrasing examples to ensure proper intent recognition.
- Dialectal variation: Although we focused on Modern Standard Arabic (MSA) for consistency, we also included standard Gulf dialect expressions to make the chatbot feel more natural and relatable to local users.
- Code-switching behavior: Many student queries included a mix of Arabic and English. Handling such input required manual tuning and diverse training samples, as out-of-the-box support was limited.
- Training effort: We found that Arabic intents required more example variations than their English counterparts to achieve comparable performance, which was an important finding for future multilingual bot development.
- Mixed Arabic–English Queries: The system is designed for the user to select either Arabic or English at the beginning of the conversation, with separate intent and entity flows for each language. Mixed Arabic–English queries within a single input are not explicitly supported; such inputs typically trigger a clarification prompt or fallback. During pilot testing, a general observed behaviour was that a mix of Arabic and English was mainly the use of domain-specific jargon (such as course codes, GPA, etc.) from the English language, while conversation was basically in Arabic. For

future work, we can incorporate a lightweight language identification step and add code-switched utterances for standard terms to improve mixed-input handling.

4. Conversational AI: Selecting a Platform

Selecting a platform for developing an AI chatbot proved to be a challenging and time-consuming task in our research. Given the numerous options available in the market, finalizing one proved challenging, even with clear requirements.

4.1. Key Factors

The following were our main requirements for the selection:

- Features

In addition to the basic features for developing an AI chatbot, the platform must have advanced features that can be utilized as the research progresses. For example, the platform should have an API endpoint for the chatbot to integrate with other systems, like a mobile app. Other essential aspects are customization levels and flexibility provided in the platform.

- API Integration

The platform must have a flexible and secure way of integrating and using APIs, adhering to industry standards and best practices.

- Localization

The platform must support localization for the Arabic language. It must support Arabic NLP at least at the basic levels for developing chatbots in Arabic.

- Cost and Efforts

With a small team and a limited budget for our research, we sought a platform that would enable us to achieve more with less input.

- Data Privacy

The platform must be compatible with the university data privacy requirements, i.e., no student or employee information must be saved on the cloud storage.

4.2. Selection Process

The following were our main requirements for the selection:

- One-to-One Demo Sessions

We contacted popular vendors providing conversational AI products to set up a demonstration of their products. With this approach, we could get familiar with product features in a short time. Most of these vendors provide their products through subscription. Some of the products with which we had one-to-one demo sessions are Botsify, Engatti, Kommunicate, ManyChat, and Flow XO.

- Trial Implementation

We implemented a trial chatbot using some of the popular frameworks and platforms. This helped us gain a better understanding of the features available in them that meet our requirements. We made trial implementations using a variety of products, including BotKit, IBM Watson, Google DialogFlow, and Botsify.

- Documentation Study and Community Survey

We studied the documentation to familiarize ourselves with the product and even surveyed the community to gather users'/developers' feedback. Some of the products were Chatterbot, RASA, Microsoft Bot Framework, and Chatfuel.

4.3. Our Understanding

The following are some of the key points from our evaluation process:

- All of them are usable to create a basic to advanced chatbot and automate the customer service agent for an organization.
- Each one has its way of implementation, including concepts and design, due to which some of them are a little hard to follow.
- Each has its pros and cons. As a result, if a good feature is present in one, then it is missing from the other. For example, if ease of use is better in one, then it has low customization, and vice versa.
- API integration is available in all, but the flexibility level is different in each one of them.
- Some of them even have templates for educational institutions, which can be customized according to requirements.

4.4. Final Selection

Out of the three classifications we made previously (i.e., Ready-to-use, Hybrid, From Scratch), we concluded that the hybrid cloud-based platforms are better suited to our requirements. We short-listed the following platforms:

- Google DialogFlow
- IBM Watson

The characteristics of both are as follows:

- More flexible API integration with university systems.
- Cloud-based platforms with a variety of products and services for future growth.
- Follow industry standards and best practices.
- Have a subscription-based pricing model.

5. Design, Implementation, and Proposed System Architecture

The design and implementation of the SQUbot system involved two major components: the AI chatbot built using IBM Watson Assistant, and a set of secure API services connected to the university's backend systems, as illustrated in Figure 2. This section provides a structured overview of the system's architecture, data sources, core components, and integration methodology.

- AI Chatbot (IBM Watson).
- API Project.

The Figure 2 shows the project design.

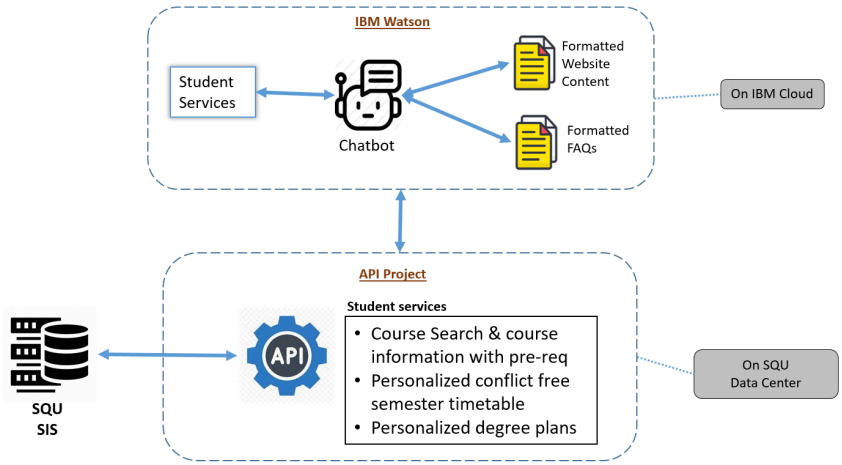

Figure 2. Project design.

5.1. System Overview

The chatbot interface begins by allowing the user to choose a preferred language: English or Arabic. Next, the user identifies their category (e.g., prospective student, current student, or visitor). Based on the selected user type and query topic, the system forwards the request to the IBM Watson Assistant, which processes it using intent classification and entity recognition. Depending on the request, the chatbot may retrieve static information from its internal knowledge base or dynamic data via secure API calls to the university's servers.

The bot carries out the conversation about several topics, including but not limited to providing contact information of employees, providing the academic calendar, providing information about admissions, suggesting courses to the students, providing students with their timetable, and providing students with their degree plan and degree audit report. The widespread accessibility that IBM Watson provides enables SQUbot to support multiple users concurrently. The architecture of the proposed chatbot (SQUbot) is illustrated in Figure 3.

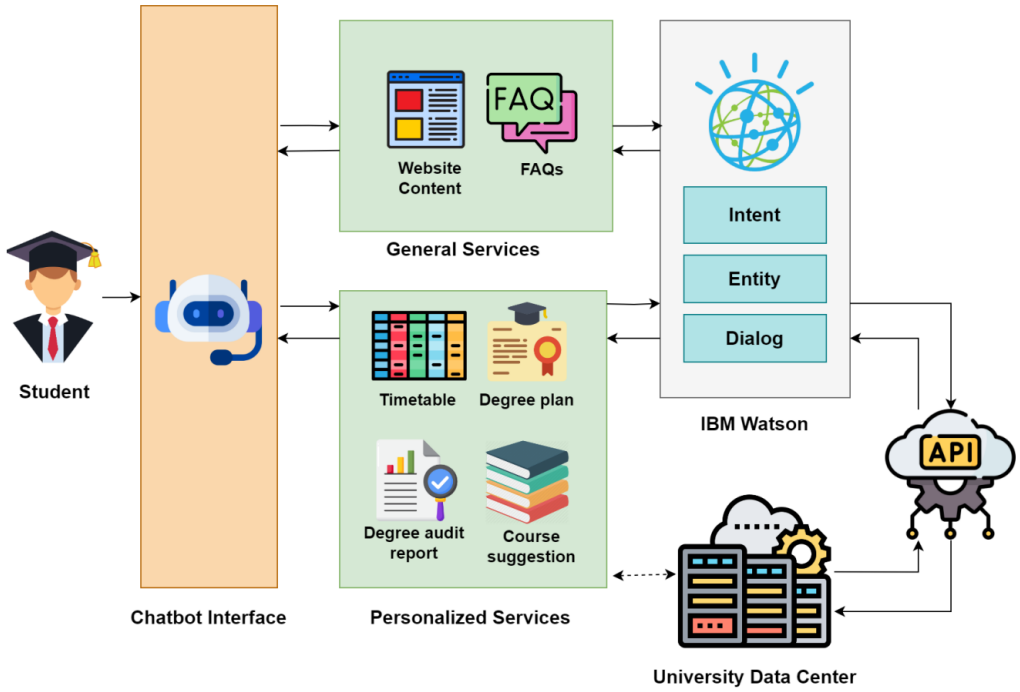

Figure 3. Architecture of the proposed SQUbot.

5.2. Data Sources

To implement the SQUbot and provide the information requested by the users, we utilized two different data sources.

- General Services: Information regarding university policies, calendars, contact directories, and course offerings was manually extracted from the SQU website and structured into training data for IBM Watson.
- Personalized Services: For personalized queries (e.g., retrieving student timetables, degree plans, or audit reports), data is accessed via a custom API layer that connects IBM Watson with secure university databases. No sensitive data is stored in the cloud, ensuring compliance with privacy standards.

One is to provide general information requested by the user about the university, while the other is to provide personalized services. The data comprising general information about the university was manually extracted from the SQU website and formatted into an appropriate structure based on IBM Watson requirements, and then fed to the chatbot.

The AI chatbot then applies NLP and ML algorithms to learn from that data. An online tool was used to extract information from the SQU website manually. The information extracted from the website is shown in Figure 4. To provide personalized services, an API has been developed that connects the SQUbot to the university's database and returns the requested information.

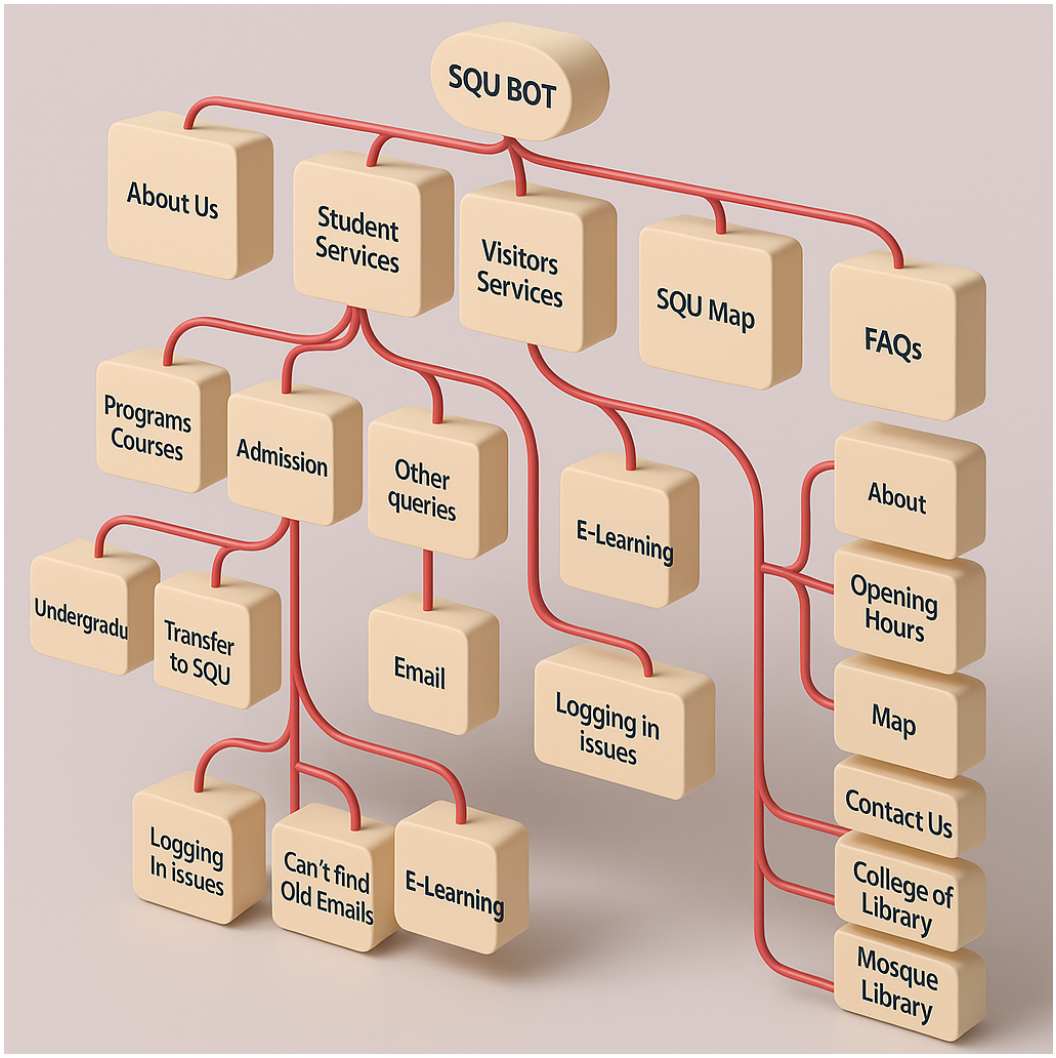

Figure 4. Sample flowchart from the SQU website.

5.3. Intents

In IBM Watson, the intent represents the goal or purpose conveyed in a user's input. Intents enable IBM Watson to understand what the user wants to communicate. By identifying the intent expressed in a user's input, the Watson Assistant can determine the relevant response or action to address it. In Watson, the intents are represented by a prefix symbol, "#". The skill for identifying intent is trained by providing multiple examples of user input and indicating which predefined intent corresponds best to each input.

Suppose a user requests the SQUbot to provide the contact information of an employee x. The query raised by the user reflects his desire to obtain the contact information of an employee, which is mapped to the intent #RetrieveEmployeeContact in SQUbot. For training IBM Watson, we provided different examples of user inputs requesting an employee's contact details. For instance, the user might use other phrases to convey the same purpose, such as, "get me contact details of employee x", "please show me the contact information of employee x", "How can I contact employee x?" etc. Providing multiple

phrases for the same intent enables Watson to understand what the user wants to accomplish. Similarly, the intents related to requesting details about admission to undergraduate and postgraduate programs, transferring to the university, and switching between university colleges are all mapped to the intent, "#getAdmission". This way, we created about 90 different intents to handle a wide range of user queries regarding several topics.

If the user's input is unclear or does not match any of the created intents, the bot suggests that the user connect to an agent. The bot also provides suggestions about the topics that the user can converse about and asks for clarification of the request if the user's input is not understood. In this case, the user can select a desired topic from the buttons, and the bot will respond accordingly. Table 4 summarizes the sample intents and entities used in SQUbot.

Table 4. Sample intents and entities used in SQUbot.

Intent ID	Example Utterances	Related Entity	Purpose
#GetAdmissionDetails	"How can I apply?", "Undergrad admission info"	@program_type	Admission info by type
#RetrieveEmployeeContact	"Show contact for Dr. X"	@emp_name	Get employee contact details
#ShowTimetable	"Show my timetable", "What are my classes today?"	@student_id	Retrieve student schedule
#CourseDetails	"Give details of CS101"	@course_code	Provide course description

5.4. Entities

In IBM Watson, an entity is a component used to extract certain vital pieces of information from the user input. Entities represent the key details required by IBM Watson to process the user's input effectively and provide a precise and accurate response to a wide range of user queries. Such pieces of information include names, locations, dates, or any other relevant information. In IBM Watson, the entities are prefixed with the symbol '@'.

For example, in SQUbot, when a user requests an employee's contact information, the "#RetrieveEmployeeContact" intent is triggered. The response provided by the assistant should indicate the contact of the specific employee the user is referring to. Hence, the *@emp_name* entity is created, which can be used to extract information about the name of the employee the user is requesting the contact information for. Each entity can have multiple values. In the user's input example, "provide the contact information of employee x", "x" is the name of the employee that can be extracted from the *@emp_name* entity. Here, "x" is one of the values of the entity *@emp_name*. We created a range of entities to structure the dialog.

This chatbot was trained on a dataset of approximately 1200 utterances covering over 90 distinct intents. Each intent represents a specific user goal (e.g., #RetrieveEmployeeContact, #GetAdmissionDetails). To improve language understanding, multiple variants of user queries were provided for each intent in both Arabic and English. Entities were defined to extract key parameters from the user's input (e.g., @empname, @coursecode), enabling personalized responses.

5.5. Dialog Management

Dialog flows were structured using IBM Watson's visual dialog editor. Each dialog node maps to a specific intent and follows conditional logic to determine the appropriate response. The system supports both text- and button-based responses to help guide the

user in refining or clarifying their requests. If the chatbot is unable to match an intent confidently, it triggers a fallback dialog and provides users with suggested topics or prompts for clarification. In IBM Watson, dialog refers to the flow of conversation to appropriately respond to the user's request based on the intents and entities identified in the user's request. For example, in response to the user's query "provide the contact information of employee x", the SQUbot shows the contact details of the employee "x". We built the dialog structure for each of the intents and entities. The dialog flow was designed to be easy to interact with. For example, if the user requests details about university admission, the chatbot provides button replies to enable the user to refine their query. The button replies to the aforementioned user's query, including admission to undergraduate programs, admission to postgraduate programs, and transfer to the university.

Besides getting the employee contact details and admission details, we built the dialog flow to handle several user requests. The services provided by the chatbot are broadly categorized into general services and personalized student services. Here, we discuss a few services under these categories.

5.5.1. General Services

Figure 5 shows the screenshot of the SQU Virtual Assistant. Using this, some services can be provided to any user using the SQUbot (e.g., students, employees, guest users). The data required to provide these services is retrieved from the university's website. Some of these services that the chatbot can address via conversation are as follows:

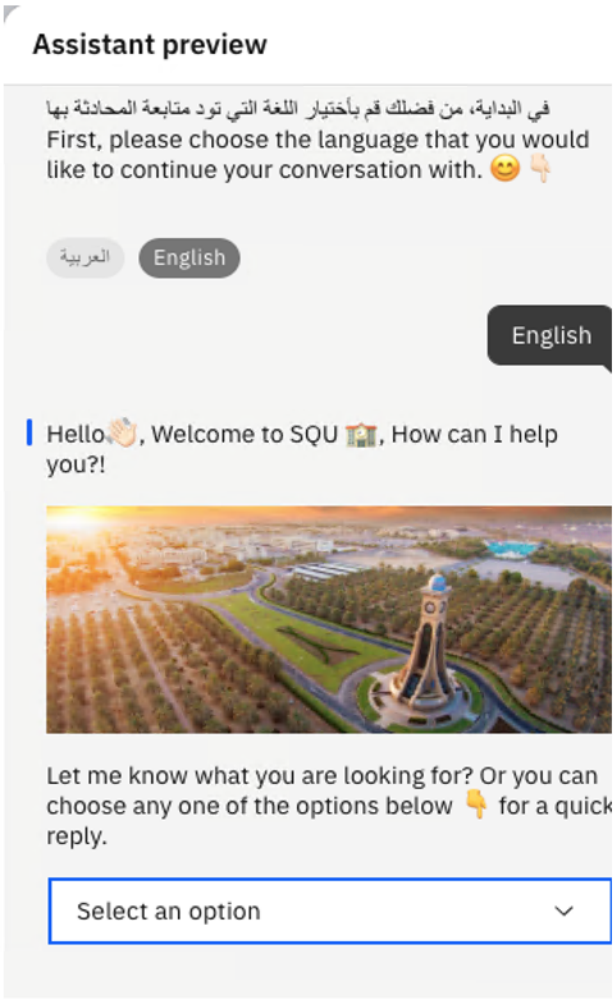

Figure 5. SQU virtual assistant [Arabic and than its english translation is provided respectively].

- Search a Course: The SQUbot allows users to search for relevant courses by typing a general phrase with the search keyword within it. The bot will connect to the API to run the search process and provide the user with a list of courses matching the keyword in the user's query. For example, a user may request the programming-related courses offered by the University, and the bot will provide a list of all the courses related to programming along with their course codes. This feature saves students valuable time and effort by eliminating the need to browse or search through course catalogs manually.
- Get the course details: Users can quickly access course information anytime by using the SQUbot without having to make manual efforts, such as searching the course details via course catalogs or contacting the administrators, and waiting for their responses. Similarly, making this feature available 24/7 via a chatbot reduces the efforts of the administrators, allowing them to focus on more strategic tasks. The students can request the chatbot to provide them with the course details of a particular course by mentioning the course code, and the chatbot will return the course details, including title, prerequisites of the course, a brief course description, credit hours of the course, and the college offering the course.
- Get the academic calendar: Users can use the SQUbot to get the academic calendar as well by typing a simple query, such as "show the university's academic calendar," and the chatbot will return the calendar of the entire year.

5.5.2. Personalized Services

Personalized services are tailored to an individual's needs. The data required for these services is retrieved from the university's data center. To provide these services, an API endpoint is developed for each intent and integrated with the chatbot. These endpoints accept requests from the chatbot with the required parameters and then send responses back to the chatbot by fetching the relevant details from the university's database. The chatbot then processes the response and presents it to the user. Some of these services are detailed below:

- Student verification: The SQUbot is capable of tailoring the search results according to the needs of the individual students. To provide personalized services to the students, the bot first verifies the student. Student verification is crucial for maintaining security and protecting students' privacy by ensuring that only authorized students can access their information. Student verification is a prerequisite for using students' personalized services using SQUbot. The bot uses the One Time Password (OTP) method to verify a student before they use any of the personalized services. If the user is an existing student, the chatbot first asks the student to enter their student ID and then calls an API endpoint to send an OTP SMS to the student's mobile number. Students have to enter the correct OTP to proceed. Once verified, students can use personalized services without needing to verify again. Figure 6 shows how the chatbot verifies the student ID for providing personalized services. Also, the following figures demonstrate the process by which the chatbot checks the student's ID to offer tailored services. After getting verified, the student can quickly access the following services:
- Show student timetable: Their timetable is saved anytime using the chatbot, which saves them the time of manually looking for the timeline. The student can type a simple phrase like 'show me my timetable'. The chatbot will connect to the university's data center through the API, look for the requested student ID, and fetch the timetable for the matching ID.
- Show student's degree plan: The student can also check out their degree plan. A degree plan consists of course categories and a list of course credits in each category

that a student has to acquire during their academic period to complete the degree. It is an essential piece of information that is sometimes very hard to get. Through the chatbot, a student can ask with a simple phrase like 'show my degree plan'. The chatbot will then automatically identify the student and make an appropriate API call to retrieve the required information, which it will then present to the student. Figure 7 shows a snapshot of the basic degree plan. If students would like to get a more detailed plan, it is also provided.

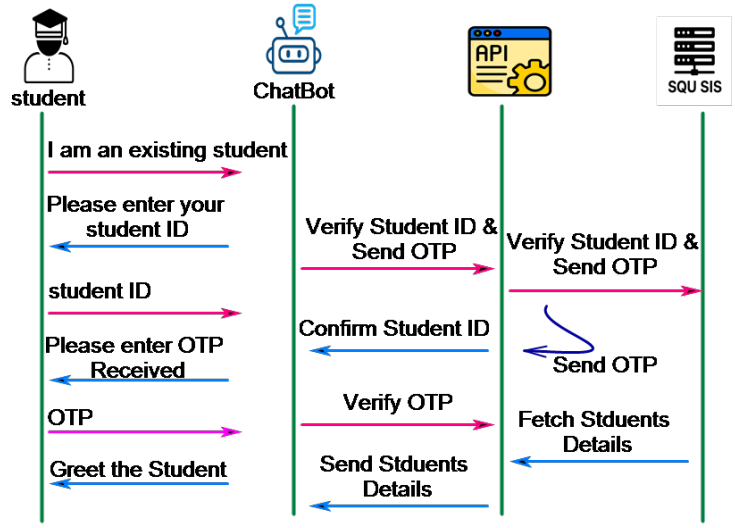

Figure 6. Student verification.

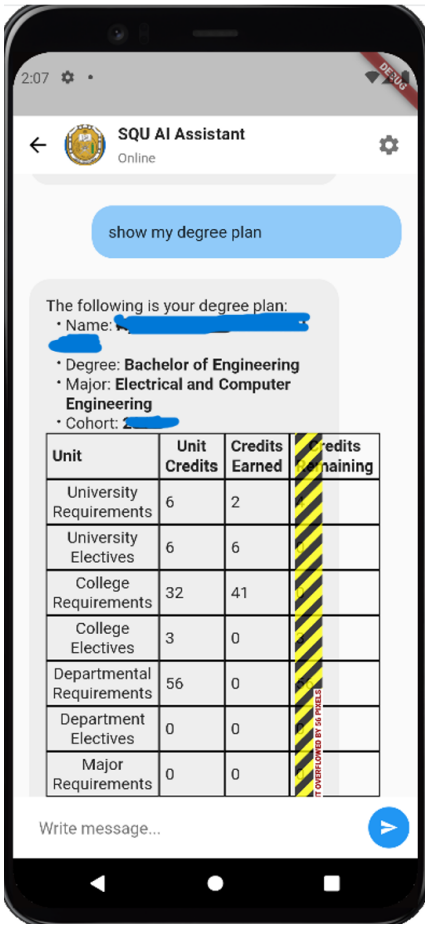

Figure 7. Students degree plan.

Figures 8–10 show different screens for different requests:

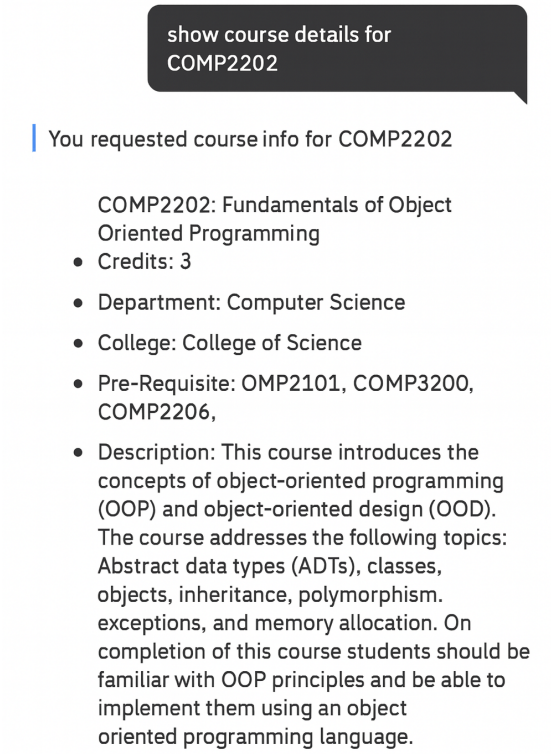

Figure 8. Course details.

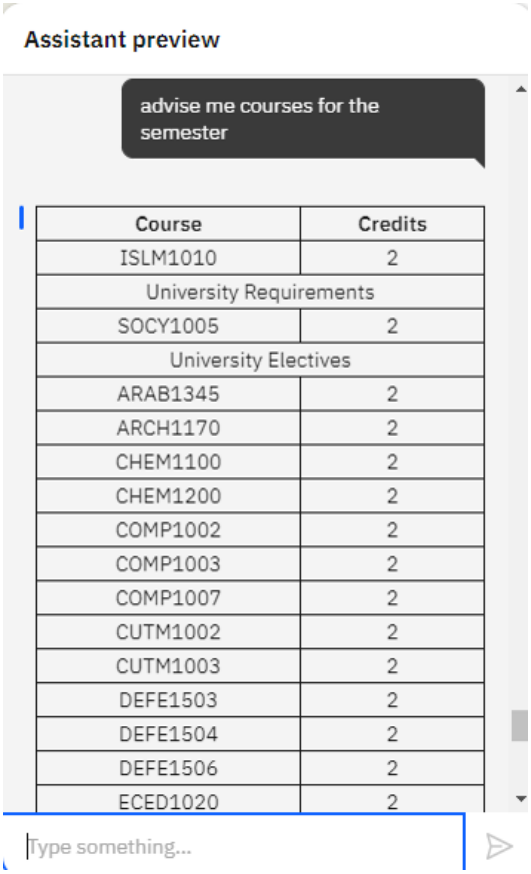

Figure 9. Advice for the courses.

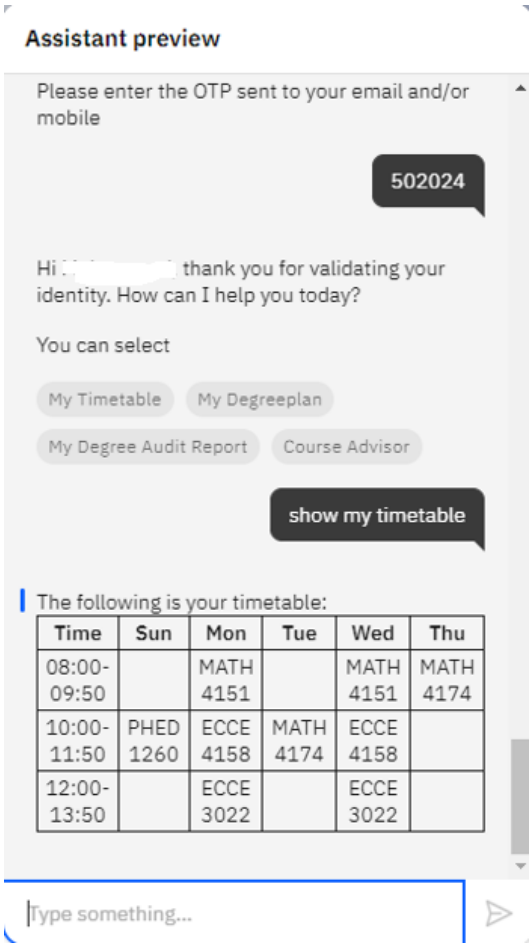

Figure 10. Student's timetable.

- Show student degree audit report: A degree audit report is a critical academic advising tool that systematically evaluates a student's educational progress toward completing their degree requirements. It functions by mapping the student's completed coursework against the prescribed curriculum for their major or program of study. The report typically breaks down credits earned by category—such as general education, core major courses, electives, and capstone requirements—and indicates which requirements have been fulfilled, which are in progress, and which remain outstanding. This structured breakdown enables students to make informed decisions when planning their course schedules for upcoming semesters. Additionally, by identifying unmet prerequisites or credit deficiencies early, the degree audit helps prevent delays in graduation and facilitates more productive advising sessions.
- Suggest courses for the next semester: The course recommendation feature can help students quickly discover relevant courses they can take without having to search through extensive catalogs, saving time and effort manually. The SQUbot can advise on possible courses a student can register for in the coming semester based on various factors, including the student's degree plan, completed courses, and courses to be offered in the coming semester.

5.6. API Integration for Personalization

A custom API layer was developed to enable secure communication between the chatbot and the university's backend systems. For sensitive tasks like retrieving a student's timetable or degree audit, the chatbot first authenticates the user using an OTP-based verification process through SMS and/or through the student's SQU email. After verification,

the chatbot calls the appropriate API with the student's ID and retrieves the required data in real time.

5.7. Platform Evaluation and Selection

An extensive evaluation was conducted to select a suitable chatbot platform. The criteria included support for Arabic NLP, API flexibility, ease of integration, customization capabilities, and compliance with data privacy requirements. After trial implementations with various platforms (e.g., Dialogflow, Botsify), IBM Watson was selected due to its strong NLU capabilities, Arabic language support, and enterprise-level integration features. Table 5 summarizes the platform evaluation.

Table 5. Platform evaluation summary.

Platform	Arabic NLP Support	API Flexibility	Ease of Use	Decision
Dialogflow	Partial	High	Easy	No
Botsify	Limited	Moderate	Very easy	No
IBM Watson	Good	High	Moderate	Yes

5.8. User Testing Protocol

The system was evaluated through a pilot deployment conducted over two weeks with a group of undergraduate students from the Department of Electrical and Computer Engineering. Participants interacted with the chatbot across five distinct sessions, where they were asked to retrieve both general information (e.g., academic calendar, employee contacts) and personalized data (e.g., degree plan, course recommendations). During this period, over 300 user messages were collected.

The chatbot successfully identified user intent in 84.9% of cases. An analysis of the remaining 15.1% of unidentified queries revealed that most errors stemmed from dialectal variations, vague phrasing, or topics outside the chatbot's current domain. To mitigate these issues, we refined the training dataset and implemented fallback prompts to encourage query clarification. These results provide insight into the system's real-world performance and highlight key areas for improvement, particularly in expanding Arabic language coverage and increasing input robustness.

This structured design ensures that the system is scalable, secure, and adaptable to future enhancements.

6. Results and Discussion

The efficacy of SQUbot was assessed in a real-world scenario. An online web page was designed and integrated with the IBM Watson chatbot via a web API integration channel. For ease of use, the web page was designed to be responsive and mobile-friendly. The URL of the SQU virtual assistant was shared with the university's students and faculty members, inviting them to interact with the virtual assistant. The ECE Department staff and students were selected for the initial pilot study. The purpose of user testing was to assess the chatbot's ability to understand users' inquiries and provide accurate information. Figure 5 shows the screenshot of the SQU Virtual Assistant.

6.1. Evaluation Setup and Data Collection

The dataset required for the bot comes from two different sources. The data needed to address general services requested by users is extracted from the university's website. In contrast, the data necessary to serve personalized student requirements is fetched from the university's data center. The chatbot was deployed for over two weeks in a controlled

setting within the Department of Electrical and Computer Engineering. Participants were undergraduate students with varied technical backgrounds. The test involved five interactive sessions with a mix of general and personalized queries. No prior training was given to users to simulate realistic usage scenarios. The chatbot was accessible through a simple web interface.

6.2. Quantitative Results

A total of 318 user messages were exchanged during the evaluation period. The system was able to correctly identify the user's intent in 270 cases, yielding an intent recognition accuracy of 84.9%. Among these successful cases, 163 involved general information queries and 107 were personalized queries (e.g., degree plans, timetables). The remaining 48 queries failed to match any existing intent with sufficient confidence. As shown in Table 6, the chatbot demonstrated strong overall performance.

Table 6. Evaluation summary.

Metric	Value
Total messages	318
Correctly matched intents	270 (84.9%)
General queries	163
Personalized queries	107
Unrecognized queries	48 (15.1%)

6.3. Error Analysis

The 48 unrecognized queries were further categorized based on their source of failure:

- Dialectal variation (22 queries): Users used informal or colloquial Arabic phrases not included in the training set.
- Ambiguity (15 queries): Vague or incomplete questions led to low-confidence predictions.
- Out-of-scope (11 queries): The query topic was outside the current chatbot capabilities (e.g., visa status, cafeteria menu).

This analysis informed a revision of the training dataset and expansion of fallback strategies. Future improvements could involve integrating transformer-based Arabic embeddings (e.g., AraBERT) to handle language variability better.

6.4. User Feedback and Engagement Patterns

Users generally found the interface intuitive and appreciated bilingual support. The average session lasted 7 min, and personalized services were accessed in over 40% of sessions. Most users preferred Arabic input, highlighting the importance of Arabic NLP performance.

6.5. Discussion and Implications

The chatbot performed reliably in handling a diverse set of queries, demonstrating the viability of using a hybrid architecture combining IBM Watson with a custom API layer. The high accuracy in general queries validates the intent classification and dialog design. However, challenges with Arabic dialects suggest a need for richer linguistic datasets and the use of more advanced NLP models. The evaluation also confirms that even a rule-based fallback strategy helps preserve user trust when intent recognition fails. The API design was effective in securely delivering personalized content with OTP-based access. This evaluation provides a baseline for future iterations of the system and lays the foundation for large-scale deployment across more university services.

We provided a short description of the research on the webpage, mentioning the features available. Furthermore, we presented a set of guidelines on how to use the chatbot, including some examples. The users were asked to inquire about specific topics, like admission information, employees' contact details, and personalized course recommendations. Figure 11 shows the distribution of user queries for general services and customized services. In total, 60.8% of the users queried the chatbot about the university's general services, while 39.2% of the users requested personalized services.

This study introduced SQUbot, the first bilingual AI chatbot deployed at Sultan Qaboos University (SQU) that supports both English and Arabic and incorporates OTP-based student verification for delivering secure, personalized academic services. Unlike existing Arabic-language chatbots that are typically limited to general or static responses, SQUbot leverages IBM Watson Assistant and integrates with SQU's internal data systems through secure APIs. This enables it to provide tailored information such as student timetables, degree plans, audit reports, and course recommendations based on real-time data. During the pilot deployment, involving students and staff from the Department of Electrical and Computer Engineering, the chatbot handled over 300 messages across five user sessions. It achieved an intent recognition accuracy of 84.9%, demonstrating strong performance in understanding and responding to diverse student queries. A detailed error analysis of the 15.1% of unidentified queries revealed that these were primarily due to dialectal variations, ambiguous phrasing, or out-of-scope topics, such as "my major" or "circuit," which were not represented in the training data. To address these issues, fallback mechanisms and guided prompts were implemented, and the training dataset was iteratively expanded to improve intent coverage. These results highlight both the system's effectiveness and the challenges of deploying Arabic NLP solutions in multilingual, real-world academic environments.

Figure 11 shows the results of the user testing process by indicating the percentage of the user queries that were correctly identified and unidentified by the chatbot. It is evident from the figure that the SQUbot achieved promising results by correctly identifying 84.9% of the queries.

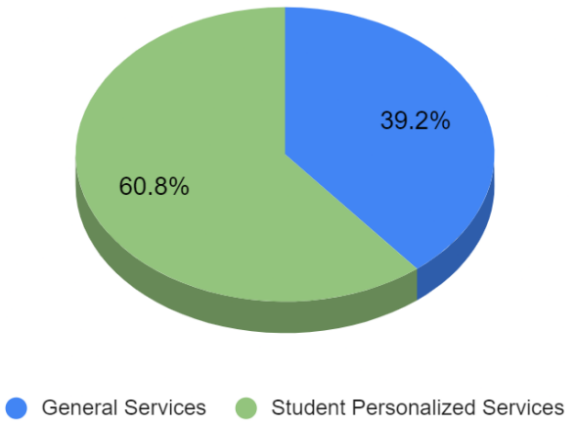

Figure 11. Distribution of user query topics.

Figure 12 shows the percentage of queries based on whether the chatbot identified it or not. From the figure, it is clear that the chatbot determined a good majority of the user queries. There could be multiple reasons for the high success ratio:

- The test was performed on a relatively small user base.
- Proper guidelines were provided about how to use the chatbot.
- Options were provided in the chatbot to guide users.

However, some of the queries that the chatbot could not answer are about the following:

- My major;
- Patients;
- Circuit;
- My specialization;
- Registration of spring 22.

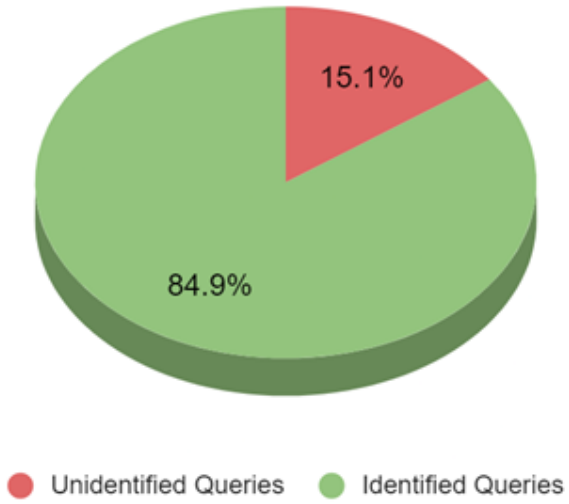

Figure 12. Unidentified and identified queries.

6.6. Novel Features and Design Contributions

Novel Aspects:

- First bilingual (English–Arabic) educational chatbot implementation at Sultan Qaboos University (SQU) using IBM Watson Assistant with cloud-based NLP capabilities and support for a dual-language dialog.
- OTP-based student verification system integrated into the chatbot to securely enable personalized academic services, such as timetable access and course advising, while preserving data privacy by processing sensitive information locally within the university's data center.
- Hybrid system architecture that connects cloud-based conversational AI with on-premise REST APIs, ensuring regulatory compliance and real-time data access without exposing personal student data to the cloud.
- Comprehensive personalized services:
  - Student timetable retrieval;
  - Degree plan consultation;
  - Degree audit reports;
  - Automated course recommendations based on academic history and course availability.

Technical Novelty: This work addresses known limitations in existing Arabic educational chatbot literature, where prior systems relied heavily on rule-based [13] and keyword-matching [14] approaches. Our system extends these capabilities by using a hybrid AI-driven architecture capable of context-aware responses and student-specific interactions.

6.7. Error Analysis of Unidentified Queries:

During the two-week pilot testing, approximately 84.9% of user queries were correctly recognized by the chatbot. The remaining 15.1% of unidentified queries were categorized as follows:

Categories of Unidentified Queries:

- Out-of-scope academic terms: e.g., "Patients", "Circuit"—queries related to disciplines or domains not covered in the training data.
- Ambiguous personal references: e.g., "My major", "What is my specialization"—these queries imply contextual personalization, but lack sufficient entity resolution or training coverage.
- Temporal queries: e.g., "When is the registration for Spring 22 semester?"—such queries require access to frequently updated academic calendars not included in the initial chatbot version.

6.8. Contributing Factors to High Success Rate:

The chatbot's relatively strong performance can be attributed to:

- Controlled user base (students from a specific college and department).
- Provision of usage guidelines through the pilot web portal and a live demo session.
- Guided conversation flows and topic suggestions within the chatbot interface.

These observations have been integrated into the Discussion section of the revised manuscript. We plan to extend the system with more dynamic data sources and structured logging for deeper error diagnostics in future iterations.

7. Conclusions

This study presented the design, development, and pilot deployment of SQUbot, a bilingual Arabic–English chatbot for university services using IBM Watson Assistant integrated with secure APIs. The system was able to provide both general and personalized responses with an overall intent recognition accuracy of 84.9%. These findings highlight the feasibility of deploying AI-driven academic assistants in multilingual institutional environments.

From a technical perspective, the architecture demonstrated that a hybrid cloud–local integration with OTP-based authentication can enable secure access to backend services without storing sensitive user data in the cloud. The structured intent–entity design and dialog management flow proved effective for handling a variety of query types. Furthermore, the analytics layer enabled real-time monitoring and post-session evaluation for system improvement.

However, the system currently faces several limitations. The use of rule-based fallback mechanisms restricts recovery options in ambiguous scenarios. Dialectal and informal Arabic expressions present recognition challenges, and the training data remains limited in domain coverage. Additionally, the evaluation was conducted on a relatively small scale and for a limited duration, which may not fully capture long-term usage trends.

Future extensions of this work include enhancing the NLP backend with deep learning models, such as AraBERT or LLM-based frameworks to improve Arabic understanding and disambiguation. The system can also benefit from the integration of reinforcement learning for dynamic intent adaptation based on user feedback. Expanding the scope of supported services (e.g., financial aid, library search, faculty directories) and deploying SQUbot across mobile and voice-enabled interfaces will be explored too. A large-scale longitudinal deployment with broader user demographics is planned to validate system scalability and performance in real-world settings.

The current fallback mechanism is completely rule-based, offering rephrase suggestions or directing users to alternate help, without adaptive learning. Unrecognised queries are reviewed manually and, if relevant, added to the training data; however, the present implementation does not include reinforcement learning or an automated feedback loop.

For future iterations, we can implement structured logging of fallback utterances and a user feedback prompt (e.g., "Was this helpful?") to prioritise training updates. Furthermore, a hybrid approach can be implemented by using reinforcement learning techniques, enabling the system to self-improve its recognition accuracy over time.

This work provides a foundation for academic chatbot systems that support Arabic NLP and paves the way for intelligent, accessible, and scalable digital services in higher education. The findings in [23] also suggest that, with proper training and support, individuals with disabilities can gain new skills and become citizen developers within their organizations. This conclusion suggests a new research direction for future updates to our designed chatbots.

Author Contributions: Conceptualization, A.H., R.A.M., Z.N. and H.M.A.L.; Methodology, Z.N. and R.A.M.; Software, M.A.S., H.M.A.L. and A.H.; Validation, R.A.M., Z.N., H.M.A.L. and A.H.; Formal Analysis, A.H.; Investigation, A.H.; Resources, Z.N.; Data Curation, H.M.A.L., R.A.M. and M.A.S.; Writing—original draft preparation, Z.N.; Writing—review and editing, H.M.A.L. and A.H.; Visualization, A.H.; Supervision, Z.N.; Project administration, Z.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by an Omantel Grant No. (EG/SQU-OT/21/02) and CIRC-SQU Grant No: (IG/DVC/CIRC/23/01), Sultanate of Oman, under the research project "Artificial Intelligence in Education System at Sultan Qaboos University: Implementation Plan and Challenges" and "Plan for Implementing AI in the Education System at SQU and Challenges".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this study were fully anonymized prior to access and analysis. No personally identifiable information (such as names, student IDs, or contact details) was included in the dataset. The study received formal authorization from the Assistan Dean of the Student Affairs Department (the co-author), who granted access to the anonymized student data for research purposes. Given that the data were de-identified and used in aggregate, individual informed consent from students was not required under our institution's ethical guidelines. However, all procedures adhered to applicable data protection and research ethics standards. Supporting data are not available at this time.

Acknowledgments: This work is a testament to the university's commitment to advancing its knowledge and fostering academic excellence. We would like to thank the SQU administration for their support, which was instrumental in the successful completion. Also, we would like to acknowledge the significant contribution of Ramsha Saeed, who provided valuable insights for the research. Her dedication and expertise greatly enhanced the quality of this work.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Caldarini, G.; Jaf, S.; McGarry, K. A Literature Survey of Recent Advances in Chatbots. *Information* **2022**, *13*, 41. [CrossRef]
2. Adamopoulou, E.; Moussiades, L. Chatbots: History, technology, and applications. *Mach. Learn. Appl.* **2020**, *2*, 100006. [CrossRef]
3. Anutariya, C.; Chawmungkrung, H.; Jearanaiwongkul, W.; Racharak, T. ChatBlock: A Block-Based Chatbot Framework for Supporting Young Learners and the Classroom Authoring for Teachers. *Technologies* **2025**, *13*, 1. [CrossRef]
4. Luo, B.; Lau, R.Y.; Li, C.; Si, Y.W. A critical review of state-of-the-art chatbot designs and applications. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2022**, *12*, e1434. [CrossRef]
5. Palasundram, K.; Sharef, N.M.; Nasharuddin, N.; Kasmiran, K.; Azman, A. Sequence to sequence model performance for education chatbot. *Int. J. Emerg. Technol. Learn. (iJET)* **2019**, *14*, 56–68. [CrossRef]
6. Hussain, S.; Ameri Sianaki, O.; Ababneh, N. A survey on conversational agents/chatbots classification and design techniques. In *Web, Artificial Intelligence and Network Applications, Proceedings of the Workshops of the 33rd International Conference on Advanced Information Networking and Applications (WAINA-2019) 33, Matsue, Japan, 27–29 March 2019;* Springer: Cham, Switzerland, 2019; pp. 946–956.

7. Vaira, L.; Bochicchio, M.A.; Conte, M.; Casaluci, F.M.; Melpignano, A. MamaBot: A System based on ML and NLP for supporting Women and Families during Pregnancy. In Proceedings of the 22nd International Database Engineering & Applications Symposium, Villa San Giovanni, Italy, 18–20 June 2018; pp. 273–277.
8. Zhang, S.; Dinan, E.; Urbanek, J.; Szlam, A.; Kiela, D.; Weston, J. Personalizing dialogue agents: I have a dog, do you have pets too? *arXiv* **2018**, arXiv:1801.07243. [CrossRef]
9. Anumala, R.R.; Chintalapudi, S.L.; Yalamati, S. Execution of College Enquiry Chatbot using IBM virtual Assistant. In Proceedings of the 2022 International Conference on Computing, Communication and Power Technology (IC3P), Visakhapatnam, India, 7–8 January 2022; pp. 242–245.
10. Fleming, M.; Riveros, P.; Reidsema, C.; Achilles, N. Streamlining student course requests using chatbots. In Proceedings of the 29th Australasian Association for Engineering Education Conference, Hamilton, New Zealand, 9–12 December 2018; Engineers Australia: Hamilton, New Zealand, 2018; pp. 207–211.
11. Ralston, K.; Chen, Y.; Isah, H.; Zulkernine, F. A voice interactive multilingual student support system using IBM Watson. In Proceedings of the 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), Boca Raton, FL, USA, 16–19 December 2019; pp. 1924–1929.
12. Attigeri, G.; Agrawal, A.; Kolekar, S. Advanced NLP models for Technical University Information Chatbots: Development and Comparative Analysis. *IEEE Access* **2024**, *12*, 29633–29647. [CrossRef]
13. Hijjawi, M.; Bandar, Z.; Crockett, K.; Mclean, D. ArabChat: An arabic conversational agent. In Proceedings of the 2014 6th International Conference on Computer Science and Information Technology (CSIT), Amman, Jordan, 26–27 March 2014; pp. 227–237.
14. Sweidan, S.Z.; Laban, S.S.A.; Alnaimat, N.A.; Darabkh, K.A. SEG-COVID: A student electronic guide within COVID-19 pandemic. In Proceedings of the 2021 9th International Conference on Information and Education Technology (ICIET), Okayama, Japan, 27–29 March 2021; pp. 139–144.
15. Alabbas, A.; Alomar, K. Tayseer: A Novel AI-Powered Arabic Chatbot Framework for Technical and Vocational Student Helpdesk Services and Enhancing Student Interactions. *Appl. Sci.* **2024**, *14*, 2547. [CrossRef]
16. Krishnam, N.P.; Bora, A.; Swathi, R.R.; Gehlot, A.; Chandraprakash, V.; Raghu, T. AI-Driven Bilingual Talkbot for Academic Counselling. In Proceedings of the 2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 12–13 May 2023; pp. 1986–1992.
17. Sharma, V.; Goyal, M.; Malik, D. An intelligent behaviour shown by chatbot system. *Int. J. New Technol. Res.* **2017**, *3*, 263312.
18. Verma, S.; Sahni, L.; Sharma, M. Comparative analysis of chatbots. In Proceedings of the International Conference on Innovative Computing & Communications (ICICC), Delhi, India, 21–23 February 2020.
19. Qaffas, A.A. Improvement of Chatbots semantics using wit. ai and word sequence kernel: Education Chatbot as a case study. *Int. J. Mod. Educ. Comput. Sci.* **2019**, *11*, 16. [CrossRef]
20. Shum, H.Y.; He, X.d.; Li, D. From Eliza to XiaoIce: Challenges and opportunities with social chatbots. *Front. Inf. Technol. Electron. Eng.* **2018**, *19*, 10–26. [CrossRef]
21. Ranavare, S.S.; Kamath, R. Artificial intelligence based chatbot for placement activity at college using dialogflow. *Our Herit.* **2020**, *68*, 4806–4814.
22. Al-Madi, N.A.; Maria, K.A.; Al-Madi, M.A.; Alia, M.A.; Maria, E.A. An intelligent Arabic chatbot system proposed framework. In Proceedings of the 2021 International Conference on Information Technology (ICIT), Amman, Jordan, 14–15 July 2021; pp. 592–597.
23. Hamideh Kerdar, S.; Kirchhoff, B.M.; Adolph, L.; Bächler, L. A Study on Chatbot Development Using No-Code Platforms by People with Disabilities for Their Peers at a Sheltered Workshop. *Technologies* **2025**, *13*, 146. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.