

Department of Information Technology

Faculty of Computing

The Islamia University of Bahawalpur



Software Requirement Specifications

**ML-Based Intelligent WhatsApp Assistant for University FAQs Using
Twilio Integration and Backend Automation**

Submitted by

Muhammad Jamshaid Naseem

F22BINFT1M01014

Submitted to

Sir Abdul Karim Nawaz

Summary

This project develops an ML-based WhatsApp assistant capable of answering university-related queries through natural conversation. The system leverages Twilio's WhatsApp API to receive messages from users. It routes them to a backend server integrated with a machine learning model trained on frequently asked questions (FAQs) about the university. The model intelligently interprets and responds to student inquiries such as admission details, course offerings, or campus facilities. Responses are automatically sent back to users on WhatsApp, providing a smart, real-time communication channel between the university and its students.

Table of Contents

| | |
|-------------------------------------------|-----------|
| 1. Introduction | 4 |
| 1.1. Purpose | 4 |
| 1.2. Scope | 4 |
| 1.3. Product Perspective | 5 |
| 1.4. User Characteristics | 5 |
| 1.5. Proposed Technologies | 5 |
| 2. Requirements | 6 |
| 2.1. Function Requirements | 6 |
| 2.2. Non-Functional Requirements | 8 |
| 3. Use Cases and Flow of Processes | 8 |
| 4. UML Diagram of Workflow | 11 |
| 5. References | 11 |

1. Introduction

This Software Requirements Specification (SRS) document details the functional and non-functional requirements, as well as the use cases, for the WhatsApp University Assistant project. The core objective of the project is to develop a machine learning-based system capable of understanding and responding to university-related queries in natural language. The system leverages Twilio's WhatsApp API to receive messages from users and routes them to a backend server integrated with a trained ML model. The model is designed to interpret frequently asked questions regarding admissions, courses, and campus facilities, providing accurate and context-aware responses. Once implemented, the system will enable students to interact with the university through WhatsApp, offering a real-time, intelligent communication channel for information retrieval and support.

1.1. Purpose

The purpose of this project is to develop a machine learning-based WhatsApp assistant capable of answering university-related queries in real time. The system will be trained on a dataset of frequently asked questions and their corresponding answers, enabling it to interpret student inquiries accurately. Once implemented, the assistant will be integrated with Twilio's WhatsApp API, allowing users to send messages and receive intelligent, context-aware responses instantly. The goal is to provide students with an accessible and efficient platform for obtaining university information, improving communication, and enhancing the overall user experience.

1.2. Scope

The scope of this project involves developing a machine learning-based WhatsApp assistant capable of understanding and responding to university-related queries. The system will not require full-scale natural language understanding from scratch but will leverage preprocessed datasets of FAQs and supervised ML techniques to train a model for accurate question-answering. The assistant will be integrated with Twilio's WhatsApp API to facilitate real-time communication with students. The project encompasses processing incoming messages, interpreting queries using the trained model, and generating context-aware responses. The system aims to provide an intelligent, accessible, and efficient communication channel for students seeking information about admissions, courses, and campus facilities.

1.3. Product Perspective

The developed WhatsApp University Assistant will serve as an intelligent communication module integrated within the university's digital support ecosystem. It will function as a machine learning-driven backend service that receives messages via Twilio's WhatsApp API, processes them using the trained ML model, and generates accurate, context-aware responses. The module will operate seamlessly with the WhatsApp interface, providing students with a real-time, interactive channel for obtaining information about admissions, courses, and campus facilities. Once deployed, the system will enhance the university's communication infrastructure by automating responses to common queries, reducing response time, and improving overall user experience.

1.4. User Characteristics

End-users, such as students or prospective applicants, will interact with the WhatsApp University Assistant by sending messages through the WhatsApp platform to ask questions related to admissions, courses, campus facilities, and other university information. The system is designed to be intuitive and accessible, ensuring that users with minimal technical knowledge can easily communicate with the assistant. Users are expected to have basic familiarity with WhatsApp messaging and an interest in obtaining timely and accurate university-related information. The goal is to provide a convenient, real-time communication tool that enhances user experience and reduces the need for manual support from university staff.

1.5. Proposed Technologies

The proposed project will leverage the following technologies:

1. **ML Models**: Supervised machine learning models for natural language processing, trained on university FAQ datasets to understand and respond to user queries accurately.
2. **Twilio Whatsapp APIs**: To send and receive messages from users through WhatsApp in real-time.
3. **Python**: Primary programming language for backend development, ML model training, and integration with the WhatsApp API, using libraries such as scikit-learn, pandas, and NumPy.
4. **Vectorization (TF-IDF/Embeddings)**: Techniques for converting text queries into numerical features for ML model processing.

2. Requirements

The system will allow students or prospective applicants to send university-related queries via WhatsApp, which will be received through the Twilio WhatsApp API. The backend server will process these messages and pass them to a trained machine learning model that interprets the query and generates an appropriate response. Users will receive replies in real-time, providing clear and accurate information on admissions, courses, campus facilities, or other university-related topics. The system will handle text preprocessing, including tokenization and vectorization, to ensure that queries are correctly understood by the ML model. Additionally, the assistant will be capable of handling multiple users simultaneously and maintain context for better interaction, delivering a seamless and user-friendly experience for students seeking information without needing manual support from university staff.

2.1. Function Requirements

The functional requirements of the project define the essential features and operations that the system must perform to achieve its objectives. These requirements focus on the key functions that will be implemented in the WhatsApp University Assistant system. The system will enable students or users to send queries via WhatsApp, which will then be processed by a backend server and analyzed using a trained machine learning model to generate accurate responses. The responses, covering topics such as admissions, courses, or campus facilities, will be sent back to users in real-time through WhatsApp. The system will handle text preprocessing, including tokenization, cleaning, and vectorization, to ensure the queries are properly understood by the model. Additionally, the system will allow multiple simultaneous interactions, maintain conversational context when needed, and provide accurate, informative replies consistently. Overall, these functional requirements ensure that the assistant operates efficiently, providing students with a seamless, intelligent, and interactive communication experience.

2.1.1. Send Query

- **Id:** FR001
- **Title:** Send Query
- **Description:** The system shall allow users to send text queries related to university information via WhatsApp.

2.1.2. Query Processing

- **Id:** FR002
- **Title:** *Query Processing*
- **Description:** *The system shall preprocess the received query text for analysis by the machine learning model.*

2.1.3. Generate Response

- **Id:** FR003
- **Title:** *Generate Response*
- **Description:** *The system shall classify the user query and generate an appropriate response using the trained ML model.*

2.1.4. Send Response

- **Id:** FR004
- **Title:** *Send Response*
- **Description:** *The system shall send the generated response back to the user through WhatsApp in real-time.*

2.1.5. Maintain Context

- **Id:** FR005
- **Title:** *Maintain Context*
- **Description:** *The system shall maintain conversational context where necessary to provide coherent multi-turn interactions.*

2.1.6. User Interaction

- **Id:** FR006
- **Title:** *User Interaction*
- **Description:** *The system shall provide a seamless interaction experience, allowing multiple users to query simultaneously without delays.*

2.2. Non-Functional Requirements

*Non-functional requirements define the overall qualities and constraints of the system, focusing on how the system performs rather than what it performs. These requirements ensure that the WhatsApp University Assistant is **reliable**, **efficient**, and **user-friendly**. The system should be highly responsive, providing real-time answers to user queries with minimal delay. It must be scalable, capable of handling multiple simultaneous users without performance **degradation**. Accuracy and reliability of the responses are critical, ensuring that the machine learning model provides consistent and correct answers across various types of queries. The system should maintain a high level of usability, enabling users to interact via WhatsApp without needing technical knowledge. Additionally, the application must adhere to **Maintainability** and **Extensibility**, allowing future updates to the model, FAQ dataset, or backend system with ease. Security and privacy of user messages and data must be ensured, complying with relevant data protection standards. Overall, these non-functional requirements guide the design and deployment of the system to provide a **robust**, **efficient**, and **user-friendly** communication experience.*

- **Performance:** Queries should be processed and responded to in real-time (under a few seconds).
- **Scalability:** Must handle multiple simultaneous conversations without lag.
- **Accuracy:** Responses must be correct, relevant, and contextually appropriate.
- **Usability:** Users interact through WhatsApp with minimal effort no technical expertise is required.
- **Maintainability:** Backend code and ML model should be easily updatable.
- **Security & Privacy:** All user messages must be protected; no unauthorized access to data.
- **Reliability:** System should remain operational 24/7 with minimal downtime.

3. Use Cases and Flow of Processes

Use cases describe how users interact with a system to achieve specific goals, translating functional requirements into practical scenarios. In this project, the primary actors are end-users (students or prospective students) who send queries via WhatsApp, and the system, which handles message reception, question processing, response generation, and message delivery. These use cases help the development team understand critical processes and allow stakeholders to visualize the system's operation in real-world scenarios.

3.1. Receive User Query

| | |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID | UC001 |
| Name | Receive User Query |
| Requirement(s) | FR001,FR002 |
| Actor(s) | User, System |
| Precondition | <i>The user has access to WhatsApp and has added the university assistant number.</i> |
| Postcondition | <i>The user's message is successfully received and logged by the system.</i> |
| Basic Flow | <ol style="list-style-type: none"><i>The user sends a message/question via WhatsApp.</i><i>The Twilio API receives the message.</i><i>The backend server logs the message for processing.</i><i>The system confirms receipt internally and prepares the message for analysis.</i> |

3.2. Process and Classify Query

| | |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID | UC002 |
| Name | Process and Classify Query |
| Requirement(s) | FR002,FR003 |
| Actor(s) | System |
| Precondition | <i>The user query has been received by the system.</i> |
| Postcondition | <i>The query is processed, and an appropriate response is generated.</i> |
| Basic Flow | <ol style="list-style-type: none"><i>The system parses the message to extract intent and keywords.</i><i>The machine learning model or FAQ mapping identifies the most relevant answer.</i><i>The system constructs a response, optionally including links or additional information.</i><i>If the query is unclear, the system generates a default "clarification" response.</i> |

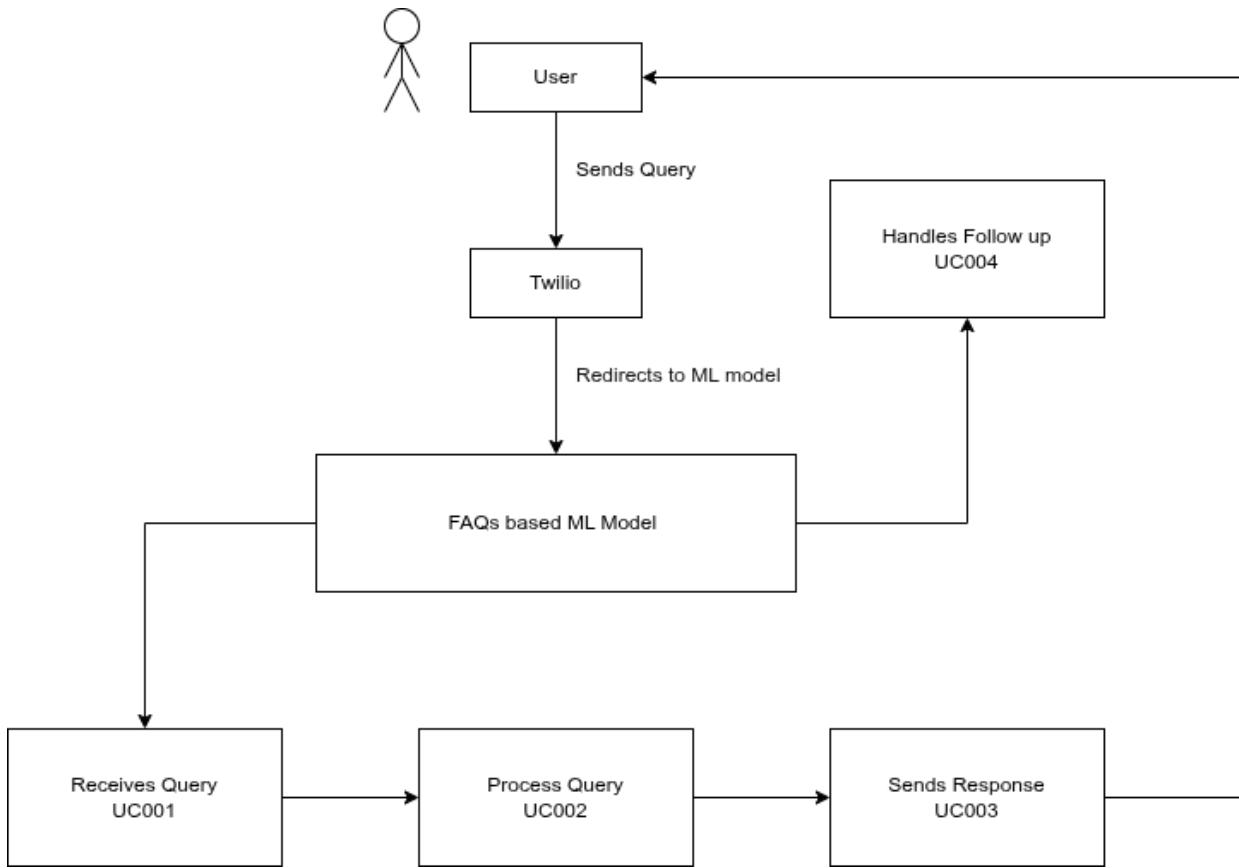
3.3. Send Response to User

| | |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID | UC003 |
| Name | Send Response to User |
| Requirement(s) | FR003, FR004 |
| Actor(s) | System |
| Precondition | <i>The system has generated a response to the user's query.</i> |
| Postcondition | <i>The user receives the response via WhatsApp.</i> |
| Basic Flow | <ol style="list-style-type: none">1. <i>The system sends the generated response through Twilio API.</i>2. <i>The user receives the response in their WhatsApp chat.</i>3. <i>The system logs the sent message for tracking and analytics.</i> |

3.4. Handle Follow-Up Queries

| | |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID | UC004 |
| Name | Handle Follow-Up Queries |
| Requirement(s) | FR001,FR002,FR003, FR004 |
| Actor(s) | User, System |
| Precondition | <i>The user has received a response and may send additional queries.</i> |
| Postcondition | <i>Multiple queries from the same user are handled efficiently in sequence.</i> |
| Basic Flow | <ol style="list-style-type: none">1. <i>The user sends another message to the assistant.</i>2. <i>The system repeats the receive → process → response flow.</i>3. <i>The conversation context may be maintained to improve response relevance.</i> |

4. UML Diagram of Workflow



5. References

1. V. Twilio, "WhatsApp API Documentation," Twilio Web Tool.
<https://www.twilio.com/docs/whatsapp>
2. M. Jamshaid Naseem, "FAQ Chatbot using Python and Machine Learning," GitHub Repository.
<https://github.com/Jamshaid2004/ML-Based-Intelligent-WhatsApp-Assistant-for-University-FAQs>
3. TensorFlow, "Text Classification with Keras," TensorFlow Documentation.
https://www.tensorflow.org/tutorials/keras/text_classification
4. Python Software Foundation, "Python Official Documentation," Python Web Tool.
<https://docs.python.org/3/>