A a cover page including your project title, your names, etc.

```
# Functional Specification for DCU CampusBot



# Table of Contents

1. [Introduction](#1-introduction)
- [1.1 Overview](#11-overview)
- [1.2 Business Context](#12-business-context)
- [1.3 Glossary](#13-glossary)
2. [General Description](#2-general-description)
   - [2.1 Product / System Functions](#21-product--system-functions)
   - [2.1.1 System Interfaces](#211-system-interfaces)
   - [2.1.2 Frontend Interfaces](#212-frontend-interfaces)
   - [2.2 User Characteristics and
Objectives](#22-user-characteristics-and-objectives)
   - [2.2.1 Guiding Principles for UI Design](#221-guiding-principles-for-ui-design)
   - [2.3 Operational Scenarios](#23-operational-scenarios)
   - [2.4 Constraints](#24-constraints)
   - [2.5 Assumptions and Dependencies](#25-assumptions-and-dependencies)
3. [Functional Requirements](#3-functional-requirements)
   - [3.1 Responding to Campus-Related
Queries](#31-responding-to-campus-related-queries)
   - [3.2 Event Notifications](#32-event-notifications)
   - [3.3 User Interaction](#33-user-interaction)
   - [3.4 Study Room Information](#34-study-room-information)
   - [3.5 Canteen Details](#35-canteen-details)
   - [3.6 Website Navigation](#36-website-navigation)
   - [3.7 Prompted Conversations](#37-prompted-conversations)
   - [3.8 Accessibility Considerations](#38-accessibility-considerations)
4. [System Architecture](#4-system-architecture)
   - [4.1 Architectural Framework](#41-architectural-framework)
   - [4.2 User Interface Module](#42-user-interface-module)
   - [4.3 Chat Logic Module](#43-chat-logic-module)
   - [4.4 Data Management Module](#44-data-management-module)
   - [4.5 Integration Module (Future Scope)](#45-integration-module-future-scope)
   - [4.6 Reused and Third-Party Components](#46-reused-and-third-party-components)
   - [4.7 System Scalability and
Maintenance](#47-system-scalability-and-maintenance)
5. [High Level Design](#5-high-level-design)
   - [5.1 High level System Architecture
Overview](#51-high-level-system-architecture-overview)
   - [5.2 Deployment Model](#52-deployment-model)
```

1. Introduction:

1.1 Overview:

The DCU CampusBot project is envisioned as an interactive web application with a chatbot interface, designed to provide real-time campus-related information to the students and faculty of Dublin City University.

The website will serve as a centralised platform for real-time updates on campus-related information, catering specifically to the needs of first-year students and exchange students who are less familiar with campus life.

The DCU CampusBot will be hosted on a full-stack website and subsequently integrated into a Twitter chat bot. Initially, the primary focus will be on crafting a user-friendly web interface. This phased strategy aims to first establish a robust website, with a subsequent plan to extend functionality to Twitter for broader communication within the Dublin City University community.

1.2 Business Context:

We envision the DCU CampusBot becoming an indispensable tool for students, lecturers and really all people that interact with the DCU campuses.

As first year students, we often found ourselves wishing for the existence of a tool like this, to keep us updated, replacing the need to log into various platforms to get needed campus related information and keeping us from being generally confused at times.

The main aim of our project is to fulfil the overarching goal of improving communication and the learning experience for students and faculty of Dublin City University (DCU). The phased approachensuresatargeteddeployment,seamlesslyintegratingthewebsiteandTwitterbot functionalities, with a specific focus on addressing the needs of newcomers to campus.

Getting all the university and campus related information and updates we need from one place by asking direct questions to a robot online in this era of great developments in the

area of artificial intelligence is a tool people most especially students will definitely be open to exploring and being a part of.

We believe that our target market for the DCU CampusBot is diverse and will welcome and greatly benefit this tool, primarily consisting of students and faculty within the Dublin City Universitycommunity.Thiscommunityischaracterisedbyitstech-savvynature,particularly in the proficient use of social media platforms, with Twitter being a prominent channel of interaction. This is why we have proposed the possibility of deployment via the Twitter platform

1.3 Glossary:

- NaturalLanguageProcessing(NLP):
  - Definition:Afieldofartificialintelligencethatfocusesontheinteraction between computers and human language, enabling machines to understand, interpret, and generate human-like text.
- Tweepy:
  - Definition:Anopen-sourcePythonlibrarythatsimplifiestheinteraction with the Twitter API, facilitating the development of Twitter-related applications.
- GraphicalUserInterface(GUI):
  - Definition:Thevisualinterfaceofasoftwareapplication,including buttons, menus, and other graphical elements that users interact with.
- Full-stackwebdevelopment:
  - Definition:Thecompleteprocessofdevelopingboththefront-end (user interface) and back-end (server-side) of a website or web application.
  -
- API(ApplicationProgrammingInterface):
  - Definition:Asetofrulesthatallowsonesoftwareapplicationto interact with another, facilitating the exchange of data and functionality.

Consider whether there are further concepts which are unusual in your project, and should be defined here.

2. General Description

2.1 Product / System Functions

A Comprehensive overview of the primary functions that the DCU CampusBot will under take

1. **Respondingtostudentscampusrelatedqueries:givingaccurateresponses to users when information is enquired. This is the main priority of our project development.**

2. **Study Room Information: Users will have the capability to access detailed information about available study spaces on campus. This feature aims to assist students in finding suitable locations for their academic endeavours.**

3. **Canteen Details: The bot will provide users with information about canteen offerings, including menus, special deals, and operating hours. This feature aims to streamline the dining experience for students and faculty.**

4. **Event Notifications: Users will receive timely notifications about upcoming events, ensuring they are aware of and can participate in various activities happening on campus.**

5. **User Interaction: The design of the DCU CampusBot prioritises intuitive and easy to understand interactions. Whether through the dedicated website or Twitter platform, the bot aims to provide a centralised and straightforward interface for users to engage with and extract relevant campus-related information.**

As I advised when we met. This list mixes up general objectives - good usablity - point 5 - do not use the expression “user friendly”, it doesn’t have a clear meaning, and specific functions Canteen Details.

I wouldn’t described |real-time Updates” as a primary function of the system - this is a feature of the primary functions to ensure that they are up to date.

2.1.1 System Interfaces.

1. UniversityDatabase:(timetablewebsite,dcuwebsite)

- Functionality:Retrievestudentandfacultyinformation,coursedetails,andcampus event schedules.
- InterfaceDescription:**UtilisesasecureAPIendpointprovidedbytheuniversity's database system. Requires authentication credentials for access.

2. Twitter API (Planned for Future Integration):

- **Functionality:** Post updates, respond to user queries, and retrieve real-time informationfromTwitter.

- **Interface Description:** Integrates with the Twitter API using Tweepy library. Requires API keys for authentication and supports standard Twitter API endpoints for posting and retrieving data.

2.1.2 Frontend Interfaces:

1. Graphical User Interface (GUI):

- **Logical Characteristics:** The primary interface is a visually intuitive GUI featuring an easy-to-navigate design, vibrant graphics, and interactive elements.

- **Optimization:** Tailored for first-year students, the GUI prioritises simplicity and clarity in presenting campus-related information.

2. **Chat Interface (Website):**

- **Logical Characteristics:** The website features a chat interface where users can engage in real-time conversations with the DCU CampusBot. Predefined prompts in little rectangle boxes guide users for easy navigation and to prevent potential errors.

- **Optimization:** This interactive chat functionality allows users to inquire about various topics, such as room availability, events, and other campus-related information. Users can either use predefined prompts or input their questions directly into the chat box, creating a dynamic and conversational user experience.

3.**TwitterCommandInterface(PlannedforFutureIntegration-TwitterBot):**

- **Logical Characteristics:** In the future Twitter bot implementation, a twitter command interface may be introduced for users familiar with this interaction style.

- **Optimization:** While not the primary interface, the command line option aims to provide an alternative for users comfortable with this mode. Users will be able to interact by mentioning the bot (@DCUcampusbot) and entering commands in their tweets, such as "@DCUcampusbot What events are on today?"

2.2 User Characteristics and Objectives

This section looks good.

UserCharacteristics:

- ***DiverseEducationalBackgrounds:*** **Encompassesawiderangeof educational levels, including first-year students new to the campus environment and faculty members with varying levels of experience.**
- ***Tech-SavvyCommunity:*** **Exhibitsfamiliarityandproficiencyinutilising social media platforms, emphasising the possibility of Twitter as a key channel for communication.**

Objectives:

- ***EnhancingCampusExperience:*** **Theprimaryobjectiveistoenhance the overall campus experience by providing efficient and convenient access to relevant information.**
- ***SeamlessIntegrationwithTwitter:*** **Giventhecommunity'sfamiliarity with social media, particularly Twitter, the design prioritises seamless integration with the platform, optimising for user-friendly interactions.**

Efficiency:

- ***Objective:*** **Streamlineuser interactionstoefficientlydeliver information.**
- ***Implementation:*** **Implement quickresponsetimesfor user queriesand prioritise relevant information in responses.**

Learnability:

- ***Objective:*** **Facilitateeaseof learningfor userswithvaryinglevelsof technological expertise.**
- ***Implementation:*** **Provideclear prompts, intuitivenavigation, and tooltips to assist users in learning the system quickly.**

Memorability:

- ***Objective:*** **Reducetheneedfor userstomemorisecomplex commands or interactions.**
- ***Implementation:*** **Utiliseconsistent designpatternsandprovideeasily accessible help features to reduce memory load.**

Error Rate:

- *Objective:* Minimiseerrorsinuser interactionstoenhanceoverall satisfaction.
- *Implementation:* Implement error preventionmechanisms, offer clear error messages, and provide avenues for users to correct mistakes.

Satisfaction:

- *Objective:* Ensureuser satisfactionwiththeoverall experienceof using DCU CampusBot.
- *Implementation:* Gather user feedbackthroughsurveys, actively address user concerns, and continuously iterate on the system based on user input.

2.2.1 Guiding Principles for UI Design:

Again this section looks good.

The design of the DCU CampusBot's user interface (UI) will be guided by principles that enhance usability and user satisfaction.

SimpleandNaturalDialogue:

- *Implementation:* Craft intuitiveandstraightforwarddialogueoptions within the chat interface, making interactions with the bot feel natural and user-friendly.

Consistency:

- *Implementation:* Maintainaconsistent designlanguageand interaction flow across the website and potential Twitter interface to reduce user confusion.

ReducedUser'sMemoryLoad:

- *Implementation:* Minimisetheneedfor userstoremember complex commands or details by providing clear prompts, visual cues, and easy navigation.

SpeakingtheUser'sLanguage:

- *Implementation:* Uselanguageandterminologyfamiliar totheDCU community, aligning with their expectations and communication style.

ShortcutsforFrequentUse:

- *Implementation:* Implement shortcutsandquickcommandsfor frequently used functions, allowing users to navigate the system efficiently.

2.3 Operational Scenarios

Another nice section.

Real-TimeUpdates:

- UseCase:Afirst-yearstudentwantstoreceivereal-timeupdatesabout campus events. They access the DCU CampusBot through the chat interface, inquire about upcoming events, by saying *“Are there any events going on in DCU today? my friends and I overheard People talking about some events!”* and receive a reply saying *“Yes there's an Exam stress management talk going on that could be very useful, it's on in HG22 at 4pm and the talk is by professor Stephen Blott”*. They are given instant information with details on locations, timings, and event descriptions.

Twitter Interaction:

- UseCase:Afacultymember,accustomedtousingTwitter,mentions @DCUcampusbot in a tweet saying “ *@DCUcampusbot any new exciting news concerning DCU in the past few days i’m a little out of the loop*“The DCUCampusBot, integrated with the Twitter API and the web app, responds promptly with something like*“Not too much has been going on but recently DCU launched a new quiet space informed by an Autism-Friendly University environment 2 days ago!That's great news for the University, isn't it?”*.

Student sample scenarios and queries:

Sample Scenario: A student has just gotten to campus. They interact with the chat interface, saying, 'I haven't eaten lunch yet; I'm hungry. What's on offer today at the canteen?' The DCU CampusBot processes the request, retrieves relevant data from the university database, and provides the student with relevant information regarding options.

The student has now met their friends at the canteen and wonders how long until they have to go. His friends say they're done for the day. The student isn't sure when he's done, but he knows his course code, and he asks the campus bot, "When is my next class?" It replies, "You have 3 hours till your next class."
They now decide they want to do something in the meantime. The student now asks the campus bot, "What events are going on today?" They get a response saying, "Dealing with peer pressure."

They decide not to go to that and want to find somewhere to hang out for a few hours and maybe do some work. The student then asks the campus bot what Computer lab is free for two hours (there isn't class scheduled). The bot replies with "LG25 is free for the next two hours."

Sample Queries

Query 1 - Lunch Options:

- UserInput:"Ihaven'teatenlunchyet;I'mhungry.What'sonoffertodayatthe canteen?"
- SystemResponse:TheDCUCampusBotprocessestherequest,retrieves relevant data from the university database, and provides the student with options, including menus, special deals, and operating hours “Roast turkey and mash is the meal deal today!”.

Query 2 - Class Schedule:

- UserInput:"Whenismynextclass?"
- SystemResponse:TheDCUCampusBotchecksthestudent'scoursecode, accesses the schedule from the university database, and replies, "You have 1 hour till your next class."

Query 3 - Events and Hangout:

- UserInput:"Whateventsaregoingontoday?"
- SystemResponse:TheDCUCampusBotsuggestsatalk,"Dealingwithpeer pressure." The student decides not to attend and asks, "What lab is free for two hours?" The bot replies, "LG25 is free for the next two hours."

2.4 Constraints:

Are there further constraints on how you can develop the system?

Twitter API Limitations: Adherence to Twitter API limitations is crucial for real-time updates and user interactions, guiding the development process.

Platform Compatibility: Ensuring compatibility across platforms is a constraint, particularly for the full-stack website, necessitating consideration of operating system requirements.

Learning Challenges : Collecting and cleaning data from various sources and bringing them all together to one database for our project. Addressing potential challenges associated with implementing Natural Language Processing (NLP) is a constraint influencing the design and development of conversational interfaces.

Data Security and Privacy: Paramount importance is given to data security and privacy considerations throughout the development lifecycle, guiding decisions on data handling and storage. Particularly, if granted access to the university's database, we acknowledge the need for meticulous handling of sensitive data. Committing to utilizing information solely for the defined purposes of the DCU CampusBot's functionality.

2.5 Assumptions and Dependencies:

Platform Availability: The assumption is that the designated hardware for the software product, especially the web server, will have the required operating system (e.g., Linux or Windows) available.For instance, if the web server lacks compatibility with the assumed operating system, it may lead to deployment issues and hinder the proper functioning of the application. Ensuring that the chosen platform aligns with the anticipated operating system is essential for a smooth and effective deployment of the DCU CampusBot, influencing decisions related to server setup and configuration.

Twitter API Stability: Stable and consistent performance from the Twitter API is assumed, as integration with Twitter is vital for real-time updates and user interactions.

Data Privacy Compliance: Dependencies on data privacy regulations are acknowledged, and changes may require adjustments to ensure ongoing compliance and guiding decisions on data handling and storage. For example if there are updates or changes in data privacy laws, we must be ready to adapt the system to remain compliant.

External Services Reliability: Dependencies on external services, such as databases and APIs, are considered. Changes in reliability or availability may impact the outlinedfunctionality.ForexampletheDCUCampusBotreliesontheuniversity's timetable database to provide accurate and up-to-date class schedules. If there are changes or interruptions in the availability of this external service, it could impact the bot's ability to retrieve real-time class information for students. Error handling and fallback options to ensure that the bot can still offer relevant information or notify users of potential delays in class schedule updates are vital.

User Engagement Patterns: Assumptions about user engagement patterns influence feature priorities and system behaviour, guiding design choices.
E.g Assumption: Users are more likely to engage with the DCU CampusBot during peak times like 9am - 4pm.
Influence on Features: prioritize certain types of queries (‘What class is next if im in com sci 3?’), or provide tailored notifications to accommodate expected user behavior patterns.

Add more detail to elaborate on the points identified above with more specific detail.

3. Functional Requirements:

See my note above regarding system functions and different types of functions.

3.1 Responding to Campus-Related Queries

Description:

The system will respond to students' campus-related queries by providing accurate and relevant information. Utilising our developed chat interface, the system will interpret questions posed by users and fetch corresponding answers from the integrated databases or external APIs.

Criticality:

Critical - The ability to respond accurately to user inquiries represents the primary value proposition of the DCU CampusBot, directly impacting its effectiveness and the users' reliance on the system.

TechnicalIssues:

- Design and implementation of a responsive chat interface capable of parsing user input.
- Integration of Natural Language Processing (NLP) techniques to understand the context and intent behind user queries.
- Development of a knowledge base that includes frequently asked questions and their answers.
- Ensuring the system can access real-time data from campus databases to provide the most current information.

Dependencies with Other Requirements:

- Dependency on the Database Access component for retrieving accurate information from campus databases.

- Dependency on the Chat Logic module to process and understand natural language queries.
- Dependency on User Interface for presenting information in a clear and accessible manner.
- Dependency on External APIs for extending the knowledge base with off-campus data sources, if required.

3.2 Event Notifications

- **Description:**
  - **Thesystemwillnotifyusersaboutupcomingeventsoncampus, sending event details and reminders through the website and Twitter.**
- **Criticality:**
  - **Critical-Eventnotificationsarecrucialforkeepingusersinformed and engaged.**
- **TechnicalIssues:**
  - **Integrationwitheventdatabasesforretrievingandbroadcastingevent information.**
  - **Implementationofaschedulingalgorithmfortimelyevent notifications.**
- **Dependencies with Other Requirements:**
  - **DependenciesonCommunicationInterfacesforsendingevent notifications.**

3.3 User Interaction

- **Description:**
  - **Thesystemwillfacilitateuser-friendlyinteractionsonboththe website and Twitter platforms, allowing users, especially first-year students, to ask questions, seek information, and engage in conversations with the CampusBot.**
- **Criticality:**
  - **Critical-Userinteractionisafundamentalaspectofthesystem, impacting user experience and satisfaction.**
- **TechnicalIssues:**
  - **Designandimplementationofchatinterfacesonthewebsiteforuser interactions.**

- Integrationofnaturallanguageprocessing(NLP)forunderstanding and responding to user queries if time allows.
- Dependencies with Other Requirements:
  - DependenciesonWebsiteInterfacefordesigningandimplementing user-friendly chat features.
  - DependenciesonSoftwareInterfacesforintegratingNLPcapabilities.

3.4 Study Room Information

- Description:
  - Thesystemwillprovideinformationabouttheavailabilityandlocation of study rooms on the campus through the website and Twitter.
- Criticality:
  - Important-Whilenotascriticalasreal-timeupdates,studyroom information contributes to a comprehensive campus experience.
- TechnicalIssues:
  - Integrationwithcampusdatabasestoretrieveandupdatestudyroom availability.
  - Developmentofanintuitiveinterfacefordisplayingstudyroom information.
- Dependencies with Other Requirements:
  - DependenciesonSystemInterfacesforaccessingandupdatingstudy room data.

3.5 Canteen Details

- Description:
  - Thesystemwillofferdetailsaboutthecanteenofferings,including menus, operating hours, and special events, accessible through both the website and Twitter.
- Criticality:
  - Important-Providingcanteendetailsenhancestheoveralluser experience.
- TechnicalIssues:
  - Integrationwithcanteendatabasesforup-to-dateinformation.
  - Designandimplementationofauser-friendlyinterfacefor canteen-related information.
- Dependencies with Other Requirements:

3.6 Website Navigation

- Description:
  - Thewebsitewillfeatureanintuitivenavigationsystem,allowingusers to easily explore and access different sections, including news, events, study spaces, and canteen details.
- Criticality:
  - Important-Intuitivewebsitenavigationisvitalforapositiveuser experience.
- TechnicalIssues:
  - Designandimplementationofauser-friendlymenuandnavigation bar.
  - Integrationwithfront-endtechnologies(HTML,CSS,Angular.js)for seamless navigation.
- Dependencies with Other Requirements:
  - DependenciesonWebsiteInterfacefordesigningandimplementing navigation features.
  - DependenciesonSoftwareInterfacesforinteractionwithfront-end technologies.

3.7 Prompted Conversations

- Description:
  - ThewebsitewillfeaturepromptedconversationswiththeCampusBot, providing users with predefined options for easy navigation and interaction, minimising potential errors and enhancing user experience.
- Criticality:
  - Important-Promptedconversationscontributetoamore user-friendly and error-resistant interaction model.
- TechnicalIssues:
  - Designandimplementationofpromptedconversationinterfaces.
  - IntegrationwithNLPforunderstandingandrespondingtouser prompts.
- Dependencies with Other Requirements:

- **Dependencies on Website Interface for designing and implementing prompted conversation features.**

3.8 Accessibility Considerations

- **Description:**
  - **ThewebsitewilladheretoWebContentAccessibilityGuidelines (WCAG), ensuring accessibility for individuals with disabilities.**
- **Criticality:**
  - **Critical-Accessibilityisessentialforprovidinganinclusiveuser experience.**
- **TechnicalIssues:**
  - **Implementationoffeaturessuchasalternativetextforimagesand keyboard navigation support.**
  - **TestingandvalidationagainstWCAGguidelines.**
- **Dependencies with Other Requirements:**
  - **DependenciesonConstraintsforadheringtoaccessibilityconstraints.**

4. System Architecture

Add an architecture diagram which can be referenced in the descriptions below.

This section will detail an overview of the DCU CampusBot with regards to the different modules and technologies we will be using to bring our project to fruition and how they are architecturally inter linked.

The architecture is primarily focused on building a robust web application with potential for future integration with Twitter if time allows. Below is a high-level overview of the anticipated system architecture, emphasising the distribution of functions across system modules and highlighting reused or third-party components.

4.1 Architectural Framework:

The architecture of the DCU CampusBot will be based on a client-server model. The server will host the application logic and manage interactions with the database, while the client will present the user interface for interactions with the bot.

4.2 User Interface Module:

The front end of the application will be developed using Angular.js, HTML, CSS, and JavaScript to deliver a responsive and engaging user experience. It will feature a chat window where users can input their queries and receive responses from the bot.

4.3 Chat Logic Module:

The backend, which will be implemented using Python Django, will process user inputs, leverage Natural Language Processing (NLP) techniques for interpretation, and produce corresponding responses. This module will also be responsible for managing user sessions to ensure a coherent conversation experience.

4.4 Data Management Module:

A PostgreSQL database will be employed to store and manage the information the bot will provide. This module will be crucial in ensuring data integrity and security, adhering to best practices in database management.

4.5 Integration Module (Future Scope):

Although the primary focus remains the web application, the architecture will include an allowance for a potential future integration with Twitter through the Twitter API. This module will handle the bot's activities on Twitter, such as sending tweets, engaging with users, and managing interactions on the platform.

4.6 Reused and Third-Party Components:

- Angular.js:Thisthird-partyframeworkwillbeusedfordevelopingthefront-end interface.
- PythonDjango:Selectedforthebackendlogicforitsefficiencyandtheease of writing reusable code.
- PostgreSQL:Chosenforitsperformanceasarelationaldatabase management system.
- NLPLibraries:Possibleinclusionofthird-partylibrariessuchasNLTKor spaCy for natural language processing capabilities.
- TwitterAPI(tweepy):inclusionoffunctionalityontwitteriftimeallowsus

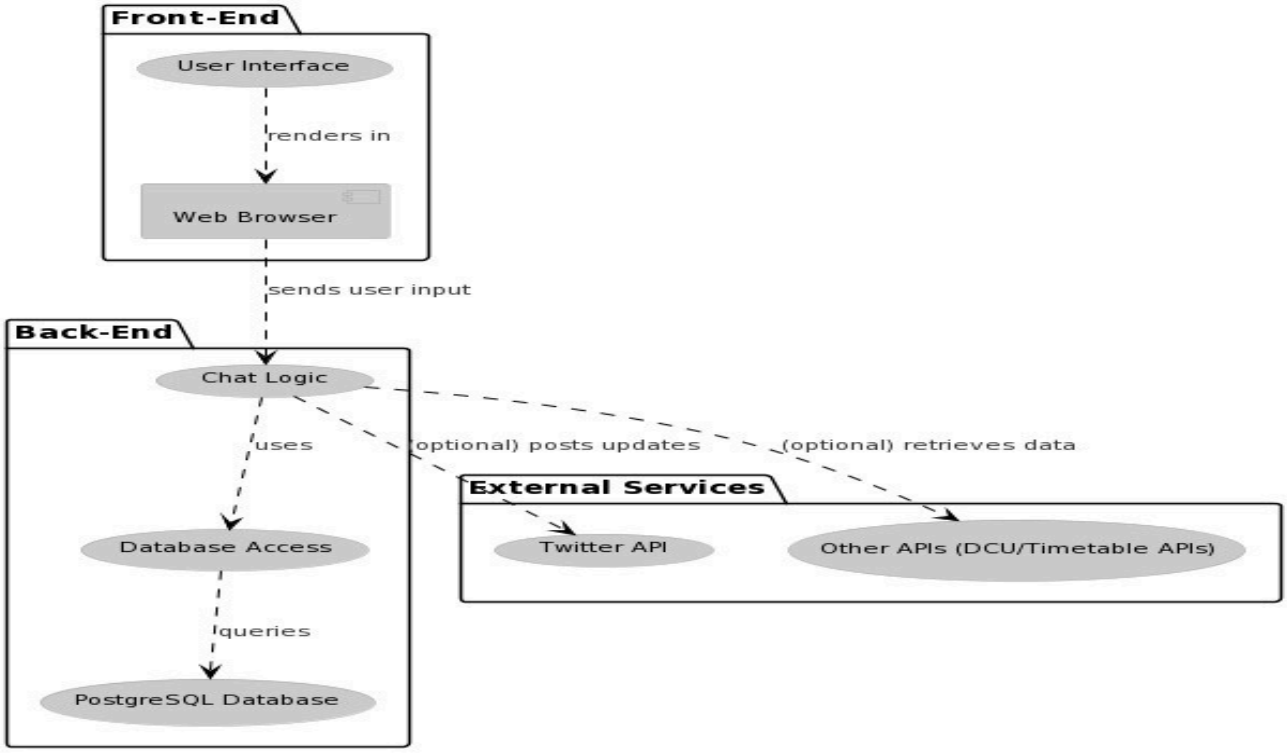
4.7 System Scalability and Maintenance:

The system will be designed to accommodate scalability, enabling the addition of new modules and functionalities as the project grows. Its modular nature will also simplify maintenance and facilitate future updates.

5. High Level Design

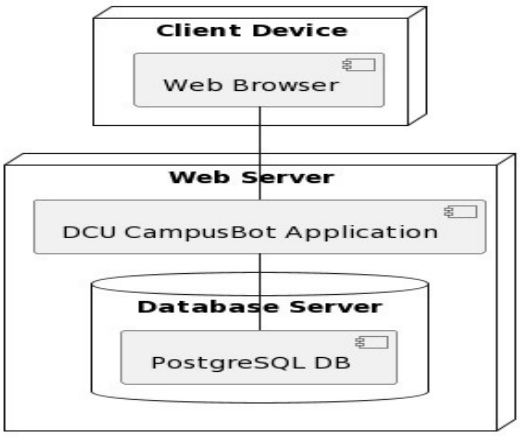
5.1 High level **System Architecture Overview**

The component diagram provides a high-level overview of the DCU CampusBot's architecture. It highlights the main components of the system, including the User Interface, Chat Logic, Database Access, PostgreSQL Database, and their interactions with external services such as Twitter API and DCU/Timetable APIs.



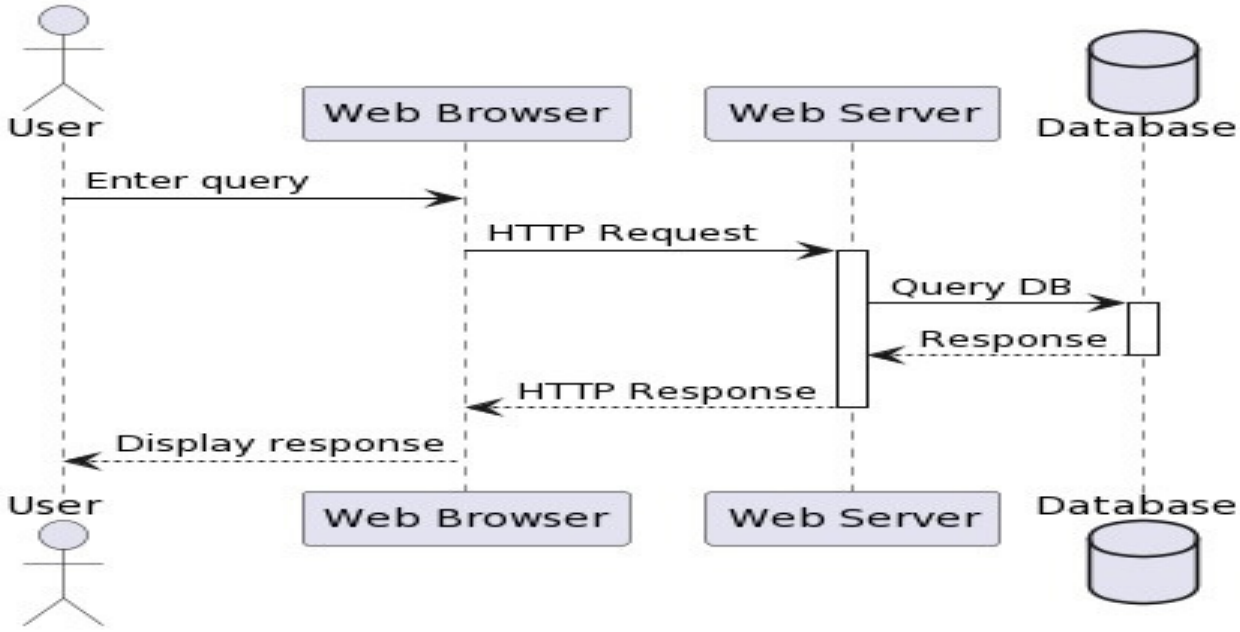
5.2 Deployment Model

The deployment diagram depicts the physical setup of the DCU CampusBot system, showing how the web application is deployed on the client device and web server, and how the PostgreSQL database is set up on the database server.



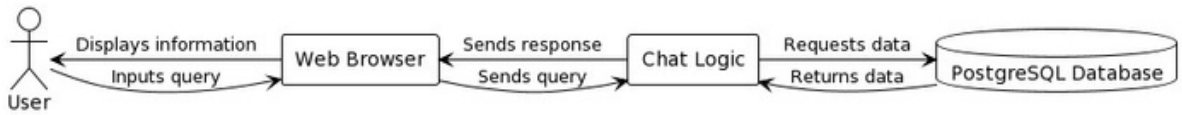
5.3 User Interaction Flow

This sequence diagram illustrates the steps a user takes to interact with the DCU CampusBot through a web browser. It starts with the user entering a query and follows the request-response cycle between the user, web browser, web server, and database.



Consider adding a Data flow diagram

Data Flow Diagram



6. Preliminary Schedule

Front-End Development:

Description: This task involves designing the graphical user interface (GUI) of the CampusBot. It includes creating wireframes, selecting colour schemes, and implementing the chat interface using front-end technologies.

- UserInterfaceCreation:Designandimplementthechatinterface.
- UserExperienceTesting:Conductusabilitytestsandrefinetheinterface.
- Dependencies:High-leveldesigncompletion

Deliverables: A fully functional and styled front-end application.

Back-End Development:

Description: This phase encompasses developing the core logic that powers the chatbot, including developing capabilities of the chatbot using prompt/key words and exploring NLP capabilities to interpret and respond to user queries.

- ChatLogicImplementation:Developthechatlogicusingpromptwordsand exploring NLP capabilities.
- DatabaseIntegration:SetupPostgreSQLdatabaseandensuredataflowis correct.
- Dependencies:High-leveldesigncompletion

Deliverables: A backend capable of processing and responding to user inputs. A fully integrated database with the backend.

Testing and Deployment:

- UnitTesting:Testindividualcomponentsforfunctionality.Thistaskinvolves writing and executing test cases for each individual component of the system

to ensure they function correctly in isolation. All throughout the development, main focus on

- IntegrationTesting:Ensurethatallcomponentsworktogetherasintended. Testing the interaction between different components of the system to ensure they work together as intended. All throughout the development, main focus on
- UserAcceptanceTesting:Validatethesystemwithend-users.Realuserswill test the system to validate the functionality and ensure it meets the requirements.

- Deployment via Twitter: extending the functionality of our web app to be able to be used on the twitter platform

- Dependencies:Front-endandback-enddevelopmentcompletion,possible deployment via twitter

Deliverables: Test cases, reports, and a list of issues to be resolved. Integration test report and system ready for user acceptance testing. User feedback report and final adjustments based on feedback.

Project Submission:

The final phase involves compiling all the documentation, code, and reports, and preparing the final project for submission.