Software Engineering Report

Company Information

- **Company Name:** Virginia Tech University (VT)
- **Company Specialization:** University (International Software Engineering Capstone Project)
- **Company Address:** Blacksburg, VA 24061, United States
  Company Website: Virginia Tech
- **Email of the contact person & mentor in the company:** seyam@vt.edu

Title Page

- **Project Title:** Steger Center Mobile Application
- **Names:** Begol & Ayten & Sharanya & Neha & Luca

Table of Contents

1. Project Overview

- The Steger Center, a prominent educational institution, faces several challenges in enhancing the experience of its students and staff. Firstly, students often need quick access to basic information such as the center's location, history, and key facts, which currently requires significant staff time to address manually. Additionally, organizing travel itineraries for students visiting the center is complex, with a need for clear guidance on travel plans, accommodations, and schedules. Managing meal preferences is another issue, as catering to diverse dietary needs is prone to errors and inefficiencies without a streamlined system. Furthermore, the center hosts numerous events that require efficient communication to ensure students are well-informed and engaged, yet the current methods are inadequate. Lastly, collecting feedback from students is essential for continuous improvement, but the process lacks structure and ease of access, leading to low response rates and incomplete data. Addressing these challenges requires a comprehensive solution to streamline information accessibility, travel coordination, meal management, event communication, and feedback collection.

2. Project Plan

Timeline

- **Start Date:** 06-Feb-2024 **End Date:** 18-May-2024

Tasks

Task	End Date	Status
Requirement Gathering	10-Feb-2024	Completed
Requirement Analysis and Research	15-Feb-2024	Completed
Technology & Subsystem Planning	20-Feb-2024	Completed
Project Charter Completion	05-Mar-2024	Completed
Project Estimation & Prioritization	12-Mar-2024	Completed
Design Approval	19-Mar-2024	Completed
Development phase 1	02-April-2024	Completed
Development phase 2	09-April-2024	Completed
Development phase 3	16-April-2024	Completed
Development phase 4	23-April-2024	Completed
Development Completion	06-May-2024	Completed
Testing Completion	15-May-2024	Completed
Project Launch	18-May-2024	Completed

Deliverables

- **Meal Planning System:** By 02-Apr-2024
- **Itinerary Portal:** By 09-Apr-2024
- **Event Hub:** By 16-Apr-2024
- **Feedback System:** By 23-Apr-2024
- **Chatbot Implementation:** By 6-May-2024

Resources

- **Tools & Technologies:**
  - Languages: Java – XML – Python - Flask
  - Database: SQLite
  - Local Tunnel : Ngrok
  - Project Management Tools: Trello
  - version control: GitHub

3. Milestones Achievements

Tasks Completed (From 06-Feb-2024 to 05-Mar-2024)

First Milestone:

- **Team Creation:**
  - Formed a team of 5 members, each with specific roles and responsibilities.
- **Requirement Gathering:**
  - Conducted two meetings with stakeholders to gather requirements.
  - Handled two rounds of requirement changes to ensure alignment with stakeholder expectations.
- **Requirement Analysis and Research:**
  - Conducted detailed research on similar applications to identify best practices.
  - Analyzed requirements to understand the problem space and potential solutions.
- **Technology & Subsystem Planning:**
  - Identified all necessary technologies to be used for the project.
  - Split the project into subsystems and created user stories for each component.
- **Project Charter:**
  - Defined project goals, scope, values, and success measures.

    - Ensured all stakeholders were aligned with the project's objectives

Challenges and Solutions

- **Requirement Changes:**
    - Used agile methodologies to adapt to changing requirements and maintain project alignment with stakeholder needs.
    - Conducted regular reviews to ensure all changes were documented.
- **Technology Alignment:**
    - Conducted in-depth research to ensure technology stack alignment with project goals.
    - Held Scrum meetings to ensure the team was proficient with chosen technologies.

Tasks Completed (From 05-Mar-2024 to 02-Apr-2024)

Second Milestone:

- **Project Estimation & Prioritization:**
    - Utilized a combination of bottom-up and top-down estimation techniques to ensure comprehensive and accurate task estimates.
    - Collaborated with team members to gather input and validate estimates.
- **Design Approval:**
    - Developed comprehensive documentation and visual aids, including diagrams and prototypes, to effectively communicate the system architecture and UI designs.
    - Conducted detailed presentations and walkthroughs, highlighting key design decisions and their rationale.
- **Development Phase 1:**
    - Established a detailed development plan, outlining the next milestones, deliverables, and timelines for coding core functionalities.
    - Assigned specific tasks and responsibilities to team members based on their skills and expertise.

- Got started with the meal planning feature with implementing an interface of it alongside backend functions

Challenges and Solutions

- **Requirement Changes:**
  - Frequent changes in requirements disrupting the development flow were addressed by using agile methodologies to adapt and conducting regular reviews to document all changes.
- **Technology Alignment:**
  - Difficulty in selecting the right technology stack was mitigated by conducting in-depth research and holding Scrum meetings to ensure team proficiency with chosen technologies.
- **Stakeholder Alignment:**
  - Ensuring stakeholder satisfaction and alignment was achieved through regular meetings to discuss progress, gather feedback, and address concerns promptly.

Tasks Completed (From 02-Apr-2024 to 23-Apr-2024)

Third Milestone:

- **Development Phase 2: Itinerary Portal:** (09-April-2024):
  - Designed and implemented forms that allow staff to collect detailed travel data from students, including flight information, arrival and departure dates, local transport arrangements, and accommodation details.
  - Developed backend processes to automatically update travel plans based on student input, reducing manual effort by staff
- **Development Phase 3: Event Hub:** (16-April-2024):
  - Developed features that allow staff and students to create, edit, and manage events. Staff members can define event details, including name, description, date, time, and location.
  - Implemented an interactive calendar that displays upcoming events. The calendar allows users to filter events by the type and view detailed event information.

- **Development Phase 4: Feedback System:** (23-April-2024):
    - Implemented secure data storage solutions to ensure that all feedback is stored confidentially.
    - Added functionality for prospective students to view anonymized feedback from current and past students. This helps prospective students gain insights into the Steger Center's environment and community.
- **Development Completion: Chatbot Implementation**:(06-May-2024):
    - Scraped data from the official Steger Center website to gather relevant information for the chatbot knowledge base. This data includes frequently asked questions, program details, contact information, nearest airports, and other useful content.
    - Chose the DialoGPT-medium model, which is trained on a large corpus of conversational data to handle diverse queries effectively. This model was selected for its ability to understand and respond naturally to user inputs.
    - Developed a Flask-based API to handle chatbot requests. This API serves as the backend, processing user queries and generating responses using the DialoGPT model.
    - Utilized Ngrok to create a secure tunnel, allowing the Flask API to be accessed and tested globally. This setup facilitated real-time testing and interaction with the chatbot from different locations.
    - Designed and implemented a user interface for the chatbot using Java in Android Studio. This interface allows users to interact with the chatbot directly from the mobile application.
    - Connected the Ngrok-generated URL to the Java interface in Android Studio, enabling seamless communication between the mobile app and the Flask API for chatbot functionalities.

Challenges and Solutions

- **Choosing the Programming Languages:**
    - We had to decide on programming languages and tools for the project. We made sure to choose languages and technologies that everyone on the team was already familiar with. This helped us avoid a steep learning curve and allowed us to work more efficiently.
- **API Deployment:**

    - Finding a good way to deploy our Flask APIs was a challenge. We used Ngrok.

Tasks Completed (From 23-Apr-2024 to 18-May-2024)

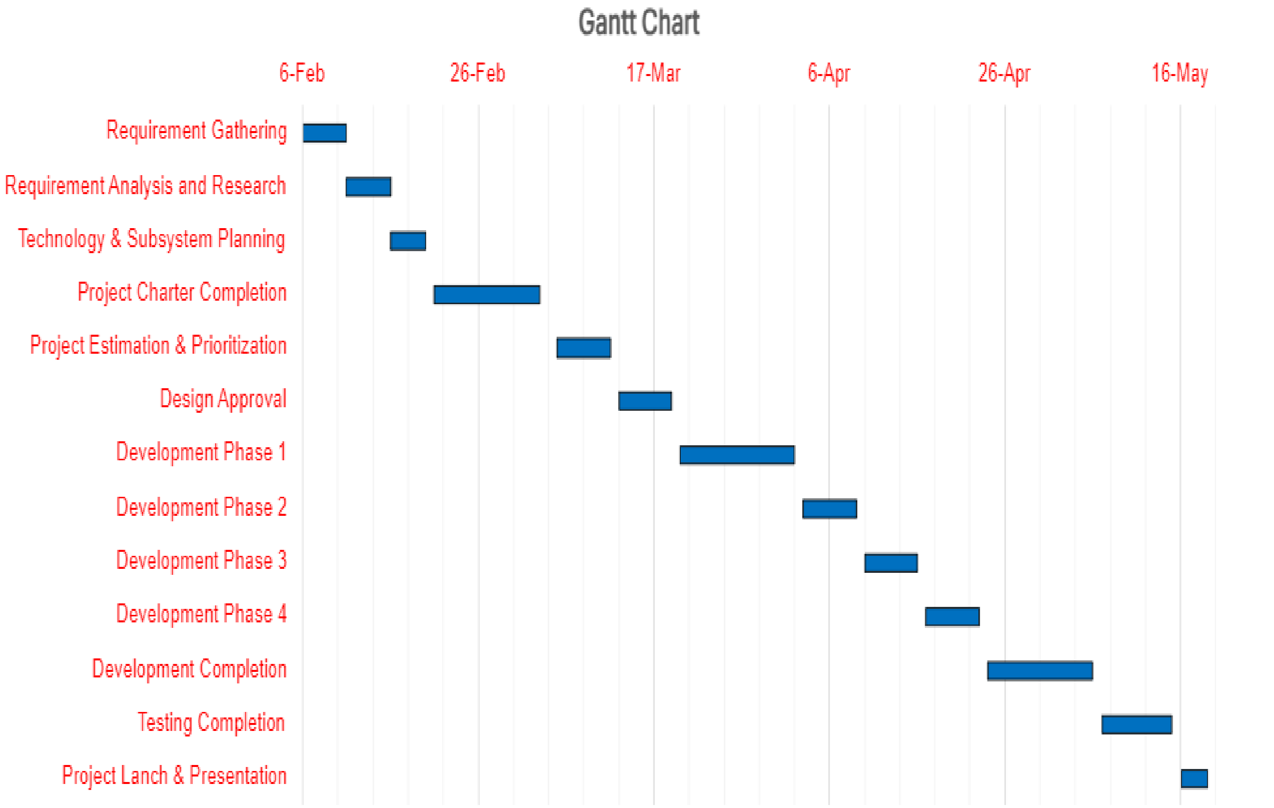
Fourth Milestone:

- **Testing Phase** (06-May-2024 to 15-May-2024):
    - Conducted end-to-end testing for all modules (Meal Planning, Itinerary Portal, Event Hub, Feedback System, Chatbot).
    - Executed user acceptance testing (UAT) with selected students and staff.
    - Identified and resolved bugs related to performance, UI inconsistencies, and data handling.
    - Prepared the system for deployment by finalizing test cases.
    - Build the project to generate the APK file.
    - Deploy the application to an Android device for final testing.
- **Project Launch Preparation** (15-May-2024 to 18-May-2024):
    - Final review and approval from stakeholders.
    - Created user manuals and training materials for staff and students.

Challenges and Solutions

- **Testing Scope Adjustments:**
    - We initially planned to test specific modules, but as we progressed, we realized additional testing was needed. We updated our test cases to cover new scenarios and ensured all features were thoroughly tested. This allowed us to address potential issues early and improve overall reliability.
- **Bug Fixes and Enhancements:**
    - During testing, we discovered new bugs and performance issues. We focused on fixing these bugs and making necessary enhancements based on feedback. Regularly updating and re-testing the application helped us maintain quality and address issues promptly.
- **Final Review and Approval:**
    - Securing final approval from stakeholders required additional revisions and updates. We held detailed review meetings to address stakeholder feedback and made necessary changes. This thorough review process ensured that the final product met all requirements and expectations.

4. Gant Chart



	Project Lanch & Presentation	Testing Completion	Development Completion	Development Phase 4	Development Phase 3	Development Phase 2	Development Phase 1	Design Approval	Project Estimation & Prioritization	Project Charter Completion	Technology & Subsystem Planning	Requirement Analysis and Research	Requirement Gathering
Start Date	16-May	7-May	24-Apr	17-Apr	10-Apr	3-Apr	20-Mar	13-Mar	6-Mar	21-Feb	16-Feb	11-Feb	6-Feb
■ Duration	3	8	12	6	6	6	13	6	6	12	4	5	5

5. Software Goals

The Steger Center Mobile Application is designed to facilitate streamlined communication and management of important logistical processes for students and staff at the Steger Center. The main goal is to reduce manual intervention, allow for efficient event management, and simplify the submission and review of travel and meal forms. By providing a centralized hub for feedback collection and automating workflows, the app ensures that students and staff can interact with the system seamlessly, enhancing their overall experience. Key goals include:

- Simplifying the management of events for staff.
- Allowing students to submit important forms related to meals and travel.
- Enabling staff to view and process submitted forms quickly.
- Facilitating feedback collection from students to improve the center's operations.
- Offering a chatbot feature for quick access to assistance.

6. Functional Requirements & System Services

The functional requirements define what the system must be able to do for both students and staff members:

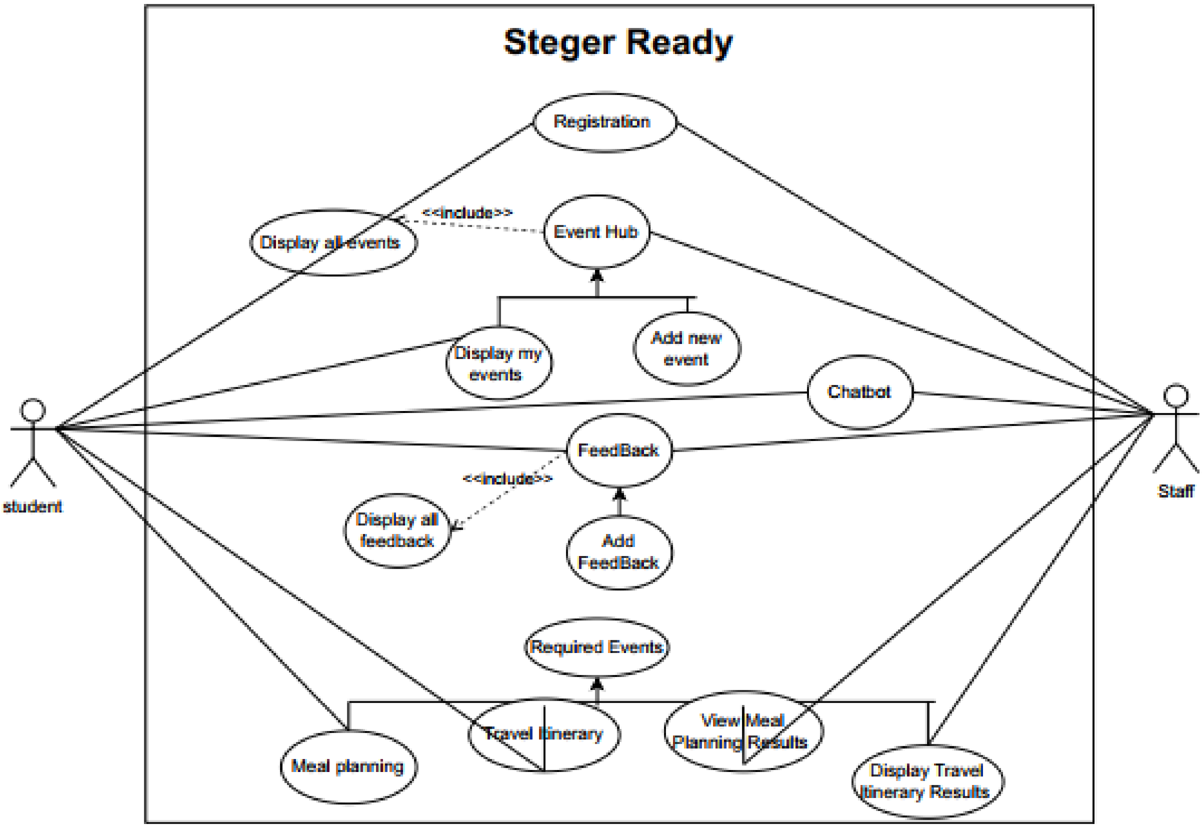
1. **(FR 1)** User Registration & Login
    - **System Services:**
        - 1.1 New users (students and staff) enter their personal details (name, email, role) in a registration form.
        - 1.2 The system validates the provided information and saves it in a database.
        - 1.3 Registered users can log in using their credentials to access the app and its features.
2. **(FR 2)** Event Management (Staff Only)
    - **System Services:**
        - 2.1 Staff members can create new events by providing essential details such as title, description, date, location, and optional image uploads.
        - 2.2 Staff can view a comprehensive list of all upcoming events.

3. **(FR 3)** View Meal & Travel Results (Staff Only)
    - **System Services:**
        - 3.1 Staff can view submitted meal preferences and travel itineraries submitted by students through designated forms.
        - 3.2 The system provides a search functionality, enabling staff to filter submissions by student name for easier management.
        - 3.3 The results are displayed in an organized format, making it easy for staff to review student submissions.
4. **(FR 4)** Meal & Travel Form Submissions (Students)
    - **System Services:**
        - 4.1 Students can fill out pre-defined forms to submit their meal preferences and travel itineraries.
        - 4.2 The system validates the submissions and saves the data in a database.
5. **(FR 5)** Feedback (Students)
    - **System Services:**
        - 5.1 Students can submit feedback regarding their experiences at the Steger Center through a dedicated feedback form.
        - 5.2 The system allows students to view feedback submitted by others, ensuring anonymity to promote honest sharing.
        - 5.3 Staff can access aggregated feedback for insights into student experiences and areas for improvement.
6. **(FR 6)** Chatbot Assistance
    - **System Services:**
        - 6.1 Both students and staff can access a chatbot integrated into the app for real-time assistance.
        - 6.2 The chatbot provides answers to frequently asked questions (FAQs).

7. Subsystem Identifications

Subsystem Name	Classes Included	Function	Interface
Welcome	Signin, SignUp, User	Handles user registration, login, and user profile management.	User input interface and database interaction
Event Hub	Event, EventHub, EventListAdapter, NewEvent, MyEvents, SingleEventDisplay, EventType, EventTypeAdapter Links Adapter	Manages event creation, listing, and display functionalities for staff and students.	User interface for event handling
Forms	MealPlanning, MealResults, Traveltinerary, TravelResults	Facilitates meal preference submissions, travel itineraries.	User input and display interfaces
Feedback	Feedback, FeedbackAdapter, FeedBackData, ViewFedback	Feedback collection and display.	User input and display interfaces
Chatbot	ChatAdapter, Chatbot, ChatMessage	Implements chatbot functionalities to assist users with FAQs and provide instant responses.	User interface and Backend API calls
Home	Home	Represents the main application interface and navigation flow.	Main activity interface

8. use case Diagram

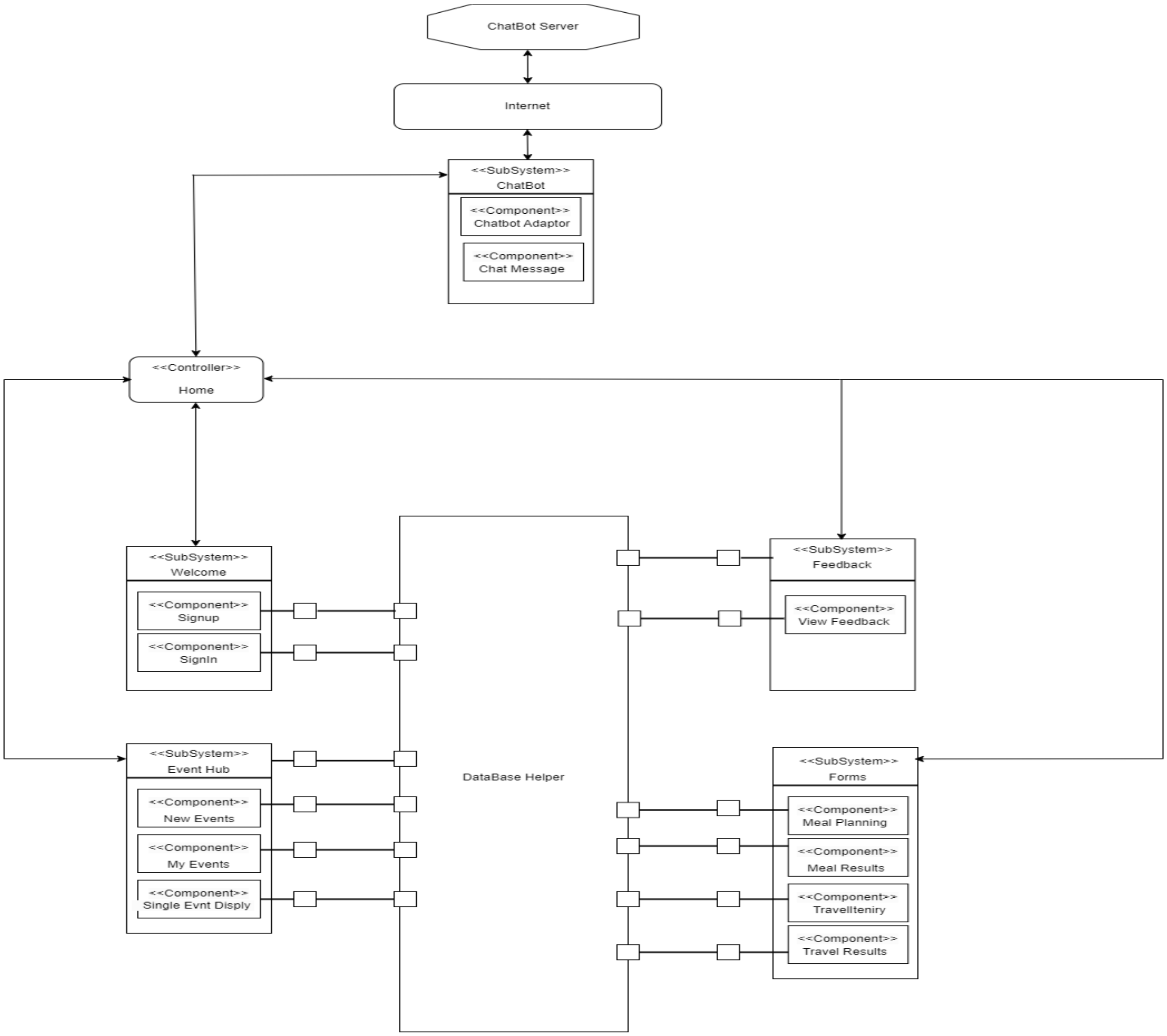


9. Use Case Scenario

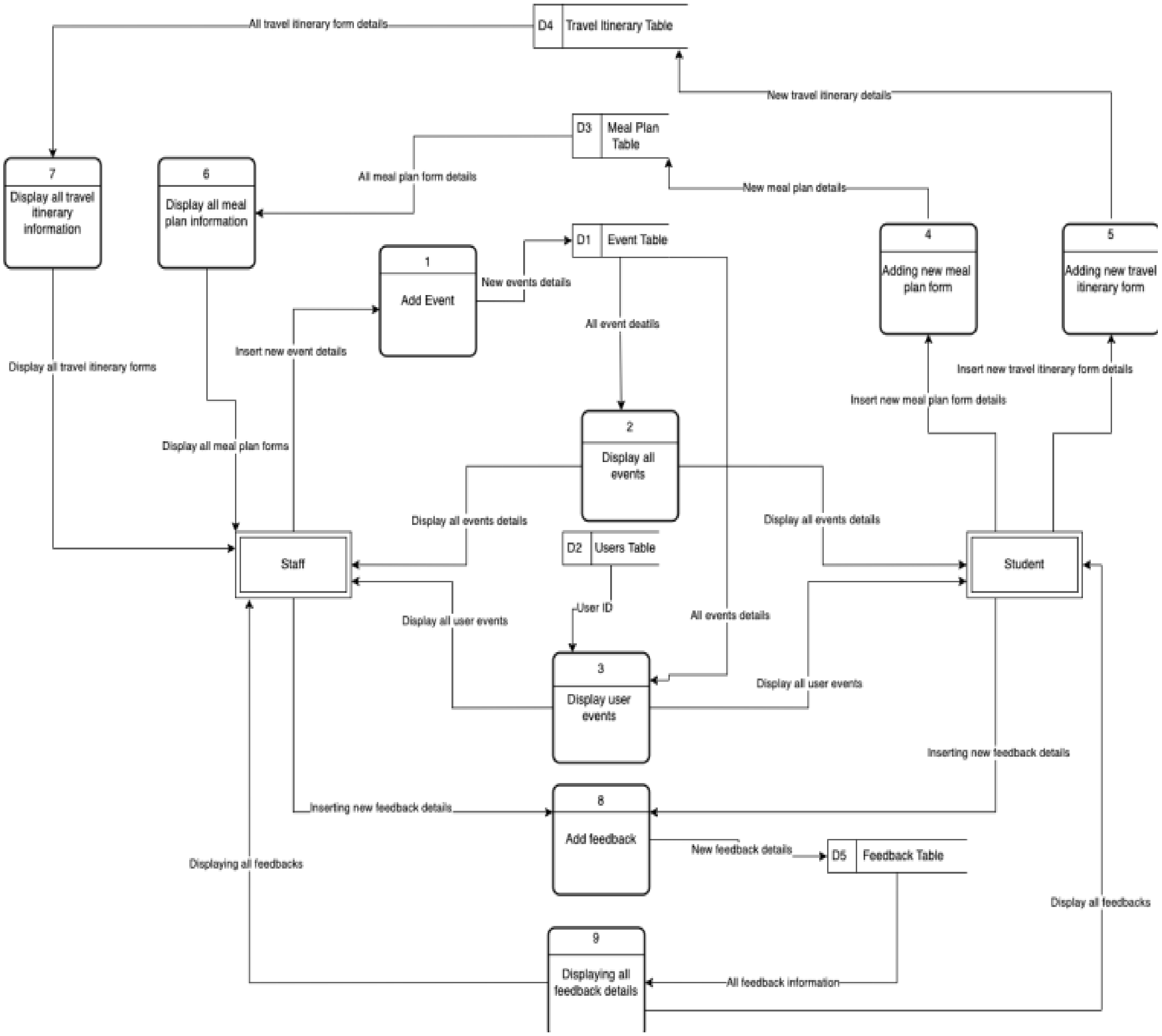
Use Case	Actors	Description	Preconditions	Main Flow	Postconditions	Alternative Flow
Event Management	Student, Staff	Allows users to view events, manage personal events, and staff members can add new events based on their role.	User must be logged in and have proper authorization.	1. User logs in. 2. User navigates to "Event Hub". 3. User views all events or personal events. 4. User can add new events (if authorized).	System displays events and saves new events to the database if added	If the user is unauthorized, an error is displayed when trying to add new events.
Feedback Submission	Student, Staff	Users can submit feedback and view feedback submitted by others.	User must be logged in.	1. User navigates to feedback section. 2. User submits feedback or views others' feedback. 3. Feedback is stored in the database.	Feedback is stored and accessible for reference	If submission fails, user gets an error.
Meal Planning	Student, Staff	Users plan meals and view results to staff.	User must be logged in.	1. User inputs meal details. 2. System processes and displays meal plan results.	Meal plans are displayed and stored correctly.	If meal planning fails, system shows an error.
Chatbot Interaction	Student, Staff	Users interact with a chatbot that	User must be logged in.	1. User logs in. 2. User opens the chatbot.	Chatbot provides FAQ response	If the chatbot cannot resolve

		answers frequently asked questions (FAQ).		3. User types a question. 4. Chatbot responds with the most relevant FAQ.	s based on user queries.	a query, it prompts the user to rephrase or seek help.
Travel Itinerary	Student, Staff	Users plan travel itineraries and view results to staff.	User must be logged in.	1. User enters travel details. 2. System shows travel results based on input to the staff.	Travel plans are displayed and stored in the database.	If travel input fails, system returns an error.

10. Component Diagram



11. Data Flow Diagram


D4
Travel Itinerary Table
All travel itinerary form details
New travel itinerary details
D3
Meal Plan Table
All meal plan form details
New meal plan details
7
Display all travel itinerary information
6
Display all meal plan information
1
Add Event
New events details
D1
Event Table
All event deatils
4
Adding new meal plan form
5
Adding new travel itinerary form
Insert new event details
Display all travel itinerary forms
Insert new travel itinerary form details
Insert new meal plan form details
2
Display all events
Display all meal plan forms
Display all events details
Display all events details
D2
Users Table
Staff
Student
User ID
Display all user events
All events details
3
Display user events
Display all user events
Inserting new feedback details
Inserting new feedback details
8
Add feedback
New feedback details
D5
Feedback Table
Displaying all feedbacks
Display all feedbacks
9
Displaying all feedback details
All feedback information

12. Test Requirements

Feature	Objective	Test Requirements	Expected Outcome
Event Management	Verify users can view and add events.	1. Test login with valid/invalid credentials. 2. Test viewing events. 3. Test adding new events as an authorized user.	1. Successful login. 2. Events are displayed. 3. Authorized users can add new events.
Feedback Submission	Verify feedback submission and retrieval.	1. Test feedback form submission. 2. Test viewing feedback. 3. Ensure feedback is saved and retrieved correctly.	1. Feedback is submitted and stored. 2. Feedback can be viewed. 3. Database updates properly.
Meal Planning	Validate meal planning functionality and results.	1. Test inputting meal details. 2. Test viewing meal plans. 3. Ensure meal plans are stored correctly.	1. Input is accepted. 2. Results are generated. 3. Meal plan data is stored successfully.
Travel Itinerary	Ensure itineraries can be generated and stored.	1. Test inputting travel details. 2. Test generating itineraries. 3. Ensure itineraries are saved in the system.	1. Travel input is accepted. 2. Itinerary is generated. 3. Travel plans are stored successfully.
Chatbot FAQ	Validate FAQ chatbot functionality.	1. Test chatbot responses to common FAQs. 2. Test the chatbot's behavior when it doesn’t recognize a question.	1. Chatbot provides relevant FAQ responses.

13. Test cases

Feature	Objective	Test Case	Expected Outcome
Event Management	Verify users can view and add events.	1. Test login with valid credentials (username: begol_osama, password: pass123).	1. Successful login.
		2. Test login with invalid credentials (username: begol_osama, password: pass231.	2. Incorrect Pass .
		3. Test viewing events (e.g., user clicks on "View Events").	3. Events ("Tech Conference 2024", "Music Fest") are displayed.
		4. Test adding a new event as an authorized user (Event: "AI Workshop").	4. Event is added
Feedback Submission	Verify feedback submission and retrieval.	1. Test submitting feedback (Feedback: "Great event! Would attend again!").	1. Feedback is submitted and stored in the database.
		2. Test viewing feedback (e.g., "Great event!", "Poor management!").	2. Feedback is displayed correctly from the database.
		3. Test system saves and retrieves feedback correctly (submit and then view feedback).	3. Database updates properly, and feedback can be retrieved.