FIELD OF STUDY: COMPUTER SCIENCE

SPECIALTY: Programming with Big Data, Data Science and Artificial Intelligence

Group 3

Members:

- Tinotenda Chidume 43981
- Chikomborero Chikomba 43572
- Lennon Nyamadzawo 43584

Contents

1.Introduction:

Topic:

The Catering Services Booking Application is a web-based platform designed to facilitate the booking and management of events such as weddings and birthday parties. It offers users the ability to view various event packages, make bookings, and contact the service provider through a contact form. The application also includes user authentication and administrative features to manage event packages and inquiries.

Scope:

- Birthday Parties
- Weddings

Functionality :

The website will let the user state the type of event they want from our listed services. It will also have the user pick the time & date (A booking system) After that they can also list the number of people to attend their event. They can also state whether they are using our listed location or if they simply want to outsource us to their preferred location.

Development tools:

- HTML
- CSS
- SQLite
- Flask
- Flask-Login
- Werkzeug
- JavaScript and AJAX

2.1) Project requirements

Functional vs non-functional Requirements

Functional Requirements:

1. Users should be able to register and log in to their accounts securely.
2. Only authenticated users should be able to access certain parts of the application.
3. Users should be able to submit inquiries through a contact form.
4. Users should be able to view different types of packages for weddings and birthdays
5. Users should be able to view detailed information about each package.
6. Users should be able to filter packages by type (e.g., wedding, birthday).
7. Authenticated users should be able to create bookings.
8. User registration should validate unique usernames and emails.
9. Users should receive appropriate error messages for invalid inputs or actions.

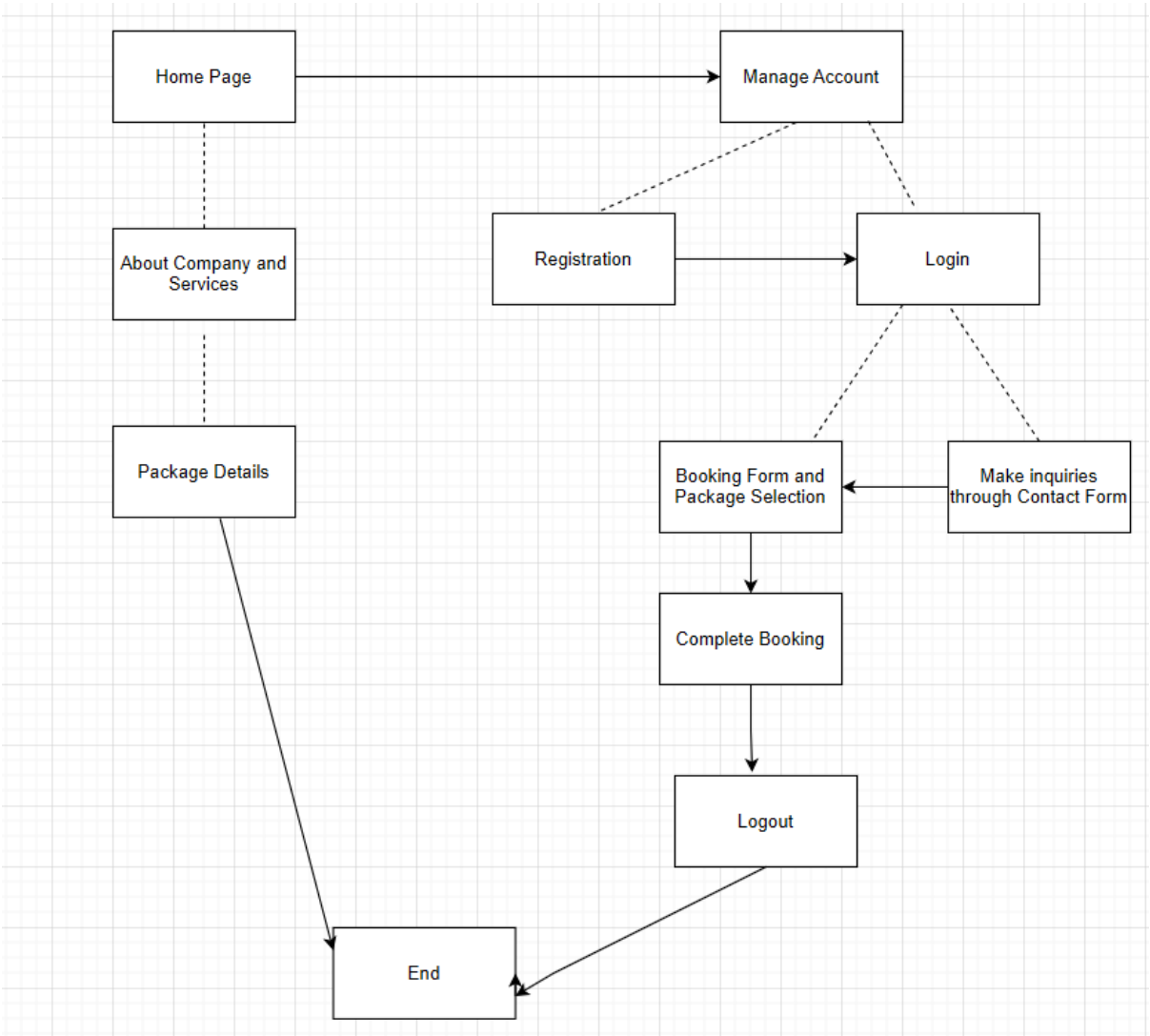
Non-functional Requirements:

1. Ensure secure storage and transmission of sensitive information such as passwords and user data.
2. The application should handle a moderate load of concurrent users.
3. The application should be intuitive and easy to use.
4. The application should be reliable and available for users.
5. Ensure the integrity and consistency of data stored in the database.
6. The application should be compatible with different web browsers and devices.

2.2) UML diagrams

The UML Activity Diagram illustrates the main processes and interactions within the application, detailing the user journey from login to booking and inquiry submission, and finally logout.

2.2.1Activity diagram



2.2.2 Use case diagram

Use Case Descriptions

1. **Register**
    - **Actors:** User
    - **Description:** Allows a user to create a new account by providing a username, email, and password.
    - **Preconditions:** User is not logged in.
    - **Postconditions:** User account is created.

2. **Login**
    - **Actors:** User
    - **Description:** Allows a registered user to log into their account using their username and password.
    - **Preconditions:** User is not logged in.
    - **Postconditions:** User is logged in.

3. **View Pages**
    - **Actors:** User
    - **Description:** Allows users to view various pages such as the Index, About, Contact, Wedding Packages, Birthday Packages, etc.
    - **Preconditions:** None.
    - **Postconditions:** None.

4. **Make Booking**
    - **Actors:** User
    - **Description:** Allows a logged-in user to make a booking by specifying event details like event type, date, time, number of attendees, location, and selecting a package.
    - **Preconditions:** User is logged in.
    - **Postconditions:** Booking is created.

5. **Send Inquiry**
    - **Actors:** User
    - **Description:** Allows a logged-in user to send an inquiry by providing their name, email, and message.
    - **Preconditions:** User is logged in.
    - **Postconditions:** Inquiry is sent.

6. **Logout**
    - **Actors:** User
    - **Description:** Allows a logged-in user to log out of their account, ending their session.
    - **Preconditions:** User is logged in.
    - **Postconditions:** User is logged out.
    -

Actors

- **User:** Interacts with the application to perform various actions such as registering, logging in, viewing pages, making bookings, sending inquiries, and logging out.

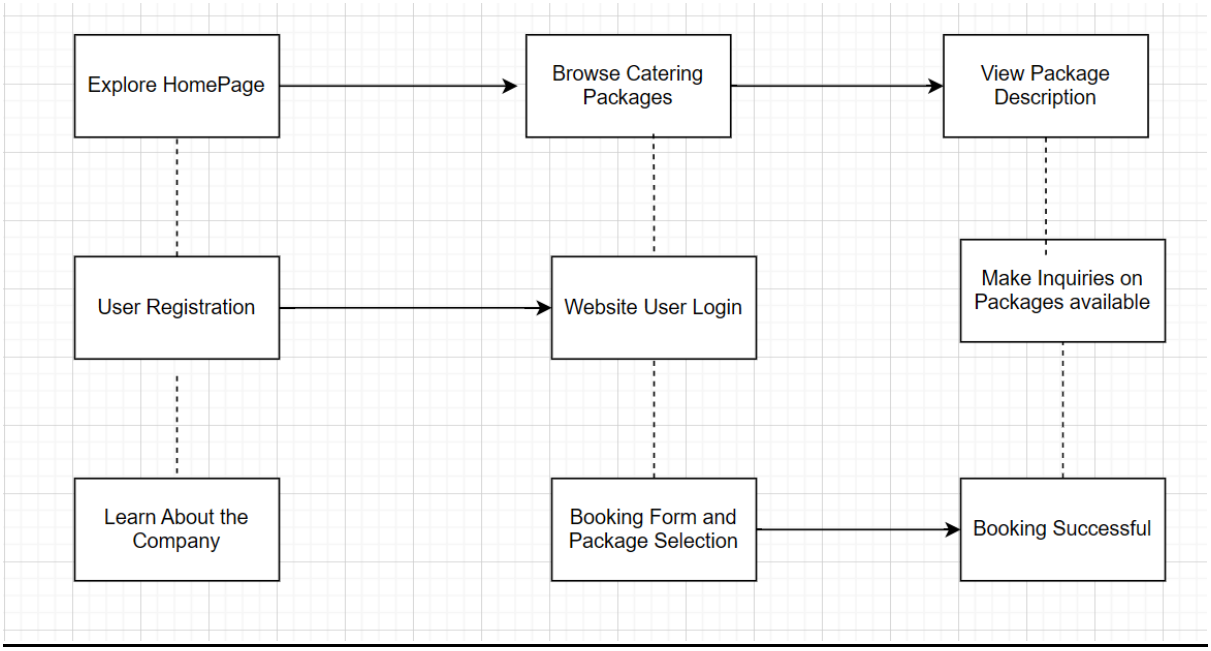
Notes

- **Preconditions:** These are conditions that must be true before the use case can begin.
- **Postconditions:** These are the expected results or states after the use case has been successfully completed.
- This Use Case diagram provides a clear overview of the interactions between actors (users) and the system (web application), including the conditions before and after each use case.
- Each use case represents a distinct user goal and describes the interaction between the user and the system.
- **Description:** Allows a logged-in user to log out of their account, ending their session.

Actors

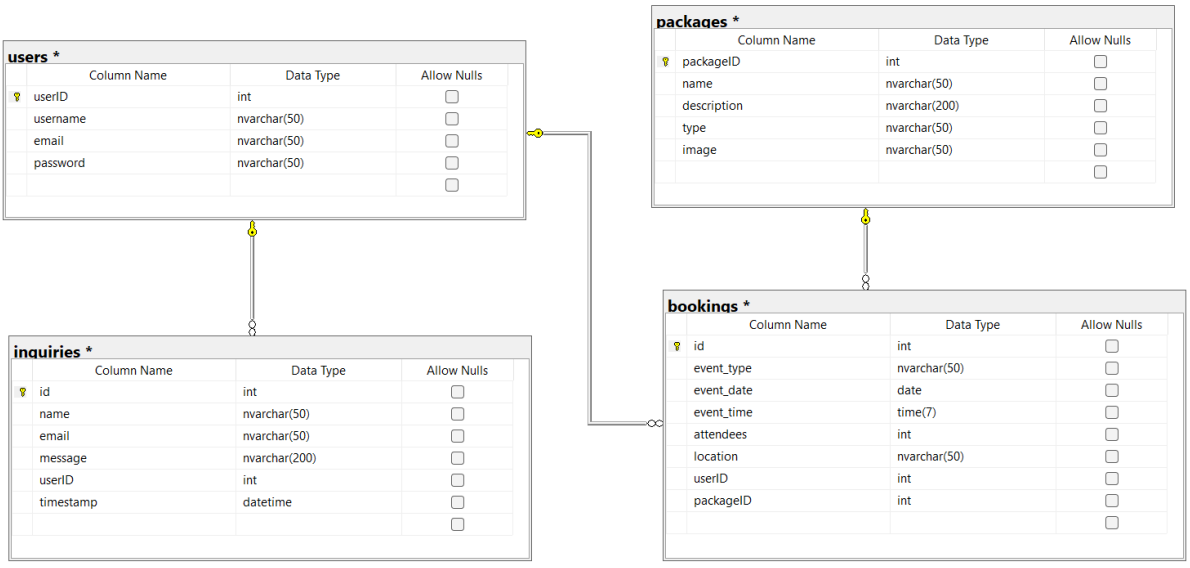
- **User:** Interacts with the application to perform various actions such as registering, logging in, viewing pages, making bookings, sending inquiries, and logging out.

Diagram



2.3) Database Diagram

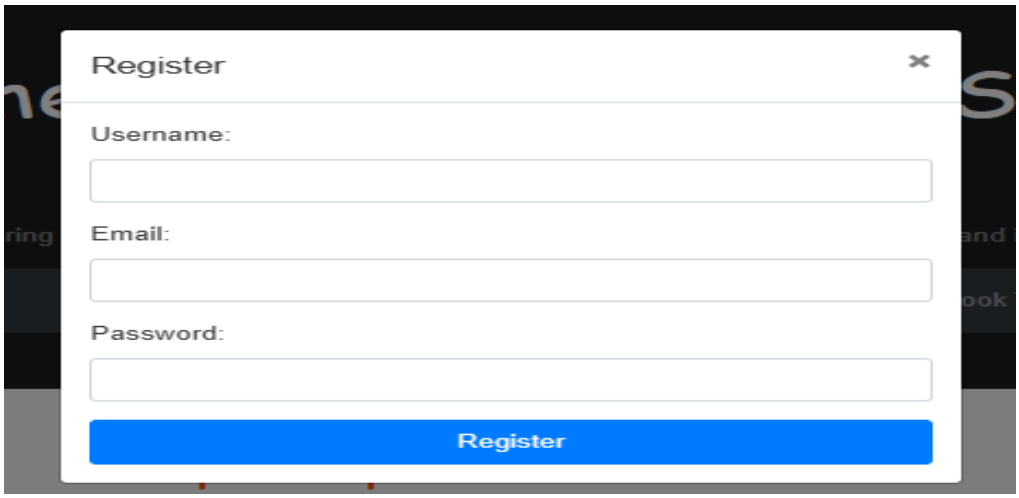
In the case of the Catering Services booking system, having a well-organized and efficient database is crucial for managing a wide range of information related to users, packages, inquiries and bookings. By maintaining a centralized database, there is easy tracking and accessing of important data. Below is the layout of how the database will look like.

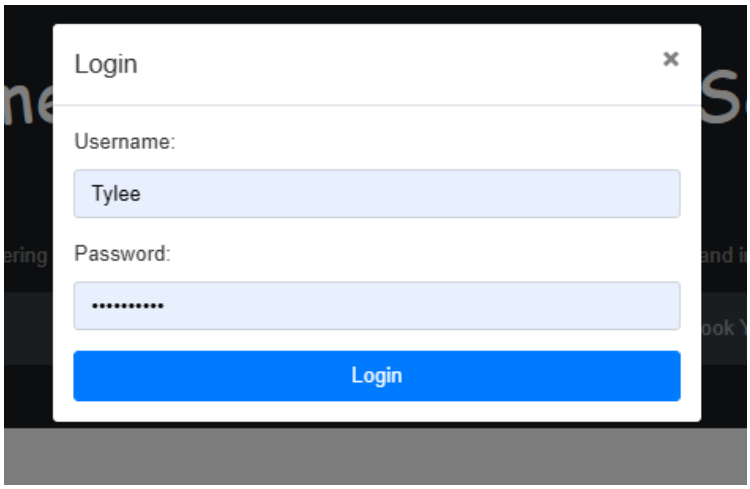


2.4) GUI Project

Home Page:

Registration and Login:





Packages:

Our Birthday Packages

Kids Fun Package

- **Venue:** Indoor Play Center
- **Catering:** Pizza, snacks, and cake
- **Decorations:** Themed decorations (e.g., superheroes, princesses)
- **Entertainment:** Clown or magician, games
- **Party Favors:** Goodie bags with toys and candy
- **Price:** $500

Teen Bash Package

- **Venue:** Private Party Room
- **Catering:** Buffet with snacks and soft drinks
- **Decorations:** Modern decor with a theme of choice
- **Entertainment:** DJ and dance floor, photo booth
- **Party Favors:** Customized items (e.g., T-shirts, hats)
- **Price:** $1,200

Adult Celebration Package

- **Venue:** Banquet Hall
- **Catering:** 3-course meal with wine and cocktails
- **Decorations:** Elegant decor with centerpieces
- **Entertainment:** Live band or DJ, karaoke
- **Party Favors:**
- **Price:** $3,000

Our Wedding Packages

Classic Elegance Package

- **Venue:** Grand Ballroom
- **Price:** $10,000
- **Catering:** 4-course gourmet meal with wine
- **Decorations:** Classic floral arrangements and elegant lighting
- **Music:** Live band for 5 hours
- **Photography:** Full-day coverage

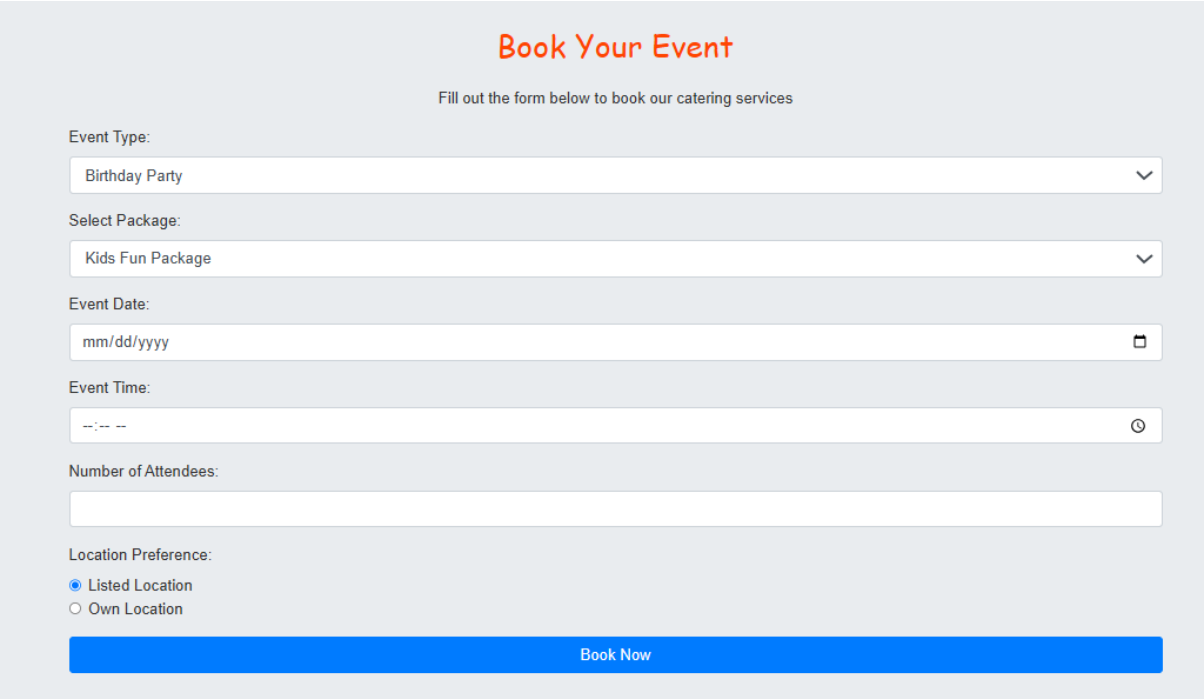
Intimate Ceremony Package

- **Venue:** Private Garden
- **Price:** $5,000
- **Catering:** 3-course meal with champagne toast
- **Decorations:** Minimalist floral decor and candles
- **Music:** Solo musician for ceremony
- **Photography:** 4-hour coverage

Destination Wedding Package

- **Venue:** Beach Resort
- **Price:** $20,000
- **Catering:** Seafood buffet with tropical cocktails
- **Decorations:** Tropical floral arrangements and tiki torches
- **Music:** Steel drum band
- **Photography:** Full-day coverage with sunset photo shoot

Booking:



3.IMPLEMENTATION OF TECHNOLOGY

Tools and Technology to be used:

1. **Flask**:
    - Flask is a lightweight web framework for Python.
    - Used for handling routing, request handling, and rendering templates.

2. **SQLite**:
    - SQLite is a self-contained, serverless, zero-configuration SQL database engine.
    - Used as the database management system (DBMS) for storing user data, bookings, packages, and inquiries.

3. **Flask-Login**:
    - Flask-Login provides user session management for Flask applications.

- Used for handling user authentication and login sessions.

4. **Werkzeug**:
    - Werkzeug is a comprehensive WSGI (Web Server Gateway Interface) utility library for Python.
    - Used for password hashing and verification in the application.
5. **Jinja2**:
    - Jinja2 is a templating engine for Python.
    - Used for rendering HTML templates with dynamic data from the Flask application.

6. **Bootstrap**:
    - Bootstrap is a popular front-end framework for building responsive web designs.
    - Used to style and layout the web pages.

7. **JavaScript and AJAX**:
    - JavaScript is used for client-side scripting, especially for interactive elements and form validation.
    - AJAX (Asynchronous JavaScript and XML) is used to asynchronously communicate with the server to fetch or send data without reloading the entire page.

8. **HTML and CSS**:
    - HTML (HyperText Markup Language) and CSS (Cascading Style Sheets) are used for creating the structure and styling of the web pages.

9. **Python**:
    - Python is the primary programming language used for the server-side logic of the Flask application.
    - Used for implementing backend functionalities such as handling requests, interacting with the database, and processing data.

These tools collectively contribute to the development, testing, and deployment of the Flask application for managing bookings and inquiries. They ensure efficient development, maintainability, and scalability of the application while ensuring security and user experience.

4.Tests

Test Case ID	***Test Scenario***	***Test Steps***	***Expected Outcome***	***Actual Outcome***	***Status***
TC001	User Registration	1.Navigate the Registration page	Registration page displayed	As expected	Passed
		2.Enter valid username,email,and password	User details accepted	Successful registration	
		3.Click the “Register” button	User is registered and redirected to login page	User registration successful	
TC002	User Login	1.Navigate to the login page	Login page displayed displayed	As expected	Passed
		2.Enter valid credentials	User credentials accepted	Successful login	
		Click “login” button	User redirected to homepage	User is successfully logged in	
TC003	Services Browsing	1.Navigate to the Services button	Package categories displayed	As expected	Passed
		2.Select Package Category	Different Packages displayed	As expected	Passed
		3.Browse the Package Options			
TC004	Contact	1.Navigate to Contact	Contact form is displayed	As expected	Passed
		2.User enters name, email and message	User input accepted	As expected	Passed
		3.Click on Send	The inquiry is accepted and save to database	As expected	Passed
TC005	Booking	1.Navigate to Book	Booking Form	As expected	Passed

		your Event	is displayed		
		2.Enter information and select package	User input is accepted	As expected	Passed
		3.Click Book Now	Event is booked successfully	As expected	Passed
TC006	Logout	1.Navigate to Logout			
		2.Click on Logout	Logout successful	As expected	Passed

ABSTRACT

The project document for the catering services booking app encompasses various key components essential for its development and understanding. It begins with Unified Modeling Language (UML) diagrams, providing a visual representation of the app's architecture, including class diagrams, sequence diagrams, and possibly state diagrams to illustrate the flow of events. Database diagrams detail the structure and relationships of the database tables required for users data, available packages, inquiries and successful bookings. Functional requirements outline the app's features, such as user authentication, browsing packages, making bookings and managing accounts, while non-functional requirements specify performance, security, and usability criteria. The graphical user interface (GUI) project presents screenshots of various website pages like Login and Book event. Test cases validate the app's functionalities and ensure its reliability, with both manual and automated testing methods employed. Lastly, the use case diagram provides a high-level overview of user interactions with the app, illustrating various scenarios and user roles. Together, these components form a comprehensive project document guiding the development and implementation of the catering services booking app.

ABSTRAKCYJNY

Dokument projektowy aplikacji do rezerwacji usług cateringowych obejmuje różne kluczowe komponenty niezbędne do jej rozwoju i zrozumienia. Zaczyna się od diagramów Unified Modeling Language (UML), zapewniających wizualną reprezentację architektury aplikacji, w tym diagramy klas, diagramy sekwencji i ewentualnie diagramy stanów w celu zilustrowania przepływu zdarzeń. Diagramy bazy danych szczegółowo opisują strukturę i powiązania tabel bazy danych wymaganych dla danych użytkowników, dostępnych pakietów, zapytań i udanych rezerwacji. Wymagania funkcjonalne opisują funkcje aplikacji, takie jak uwierzytelnianie użytkowników, przeglądanie pakietów, dokonywanie rezerwacji i zarządzanie kontami, natomiast wymagania niefunkcjonalne określają kryteria wydajności, bezpieczeństwa i użyteczności. Projekt graficznego interfejsu użytkownika (GUI) prezentuje zrzuty ekranu różnych stron internetowych, takich jak zdarzenia Login i Book. Przypadki testowe weryfikują funkcjonalność aplikacji i zapewniają jej niezawodność, zarówno przy użyciu ręcznych, jak i automatycznych metod testowania. Na koniec diagram przypadków użycia zapewnia ogólny przegląd interakcji użytkownika z aplikacją, ilustrując różne scenariusze i role użytkowników. Razem te komponenty tworzą kompleksowy dokument projektowy kierujący rozwojem i wdrażaniem aplikacji do rezerwacji usług cateringowych.