ESCI upf.
School of International Studies

System Design Specification (SDS)

'PetS' Application

Software engineering
Bachelor degree in Bioinformatics (BDBI)
(2nd Year 3rd Term, 2023)

Prepared by:
Irene Porta
Alba Mas
Paula Vela

Under the supervision of:
Daniel Soto Álvarez

Revision History

Name	Date	Reason For Changes	Version
Irene/Alba/Paula	17-04-2023	Creation of the document	0.1
Irene/Alba/Paula	03-05-2023	Clean the contents and 1. Introduction, 2. Overall description (2.1, 2.2, 2.3)	0.2
Irene/Alba/Paula	19-05-2023	Use case list and system requirements	0.3
Irene/Alba/Paula	24-05-2023	System requirements	0.4
Irene/Alba/Paula	02-06-2023	Huge changes in the table of contents and the design of the document	1.0
Irene/Alba/Paula	09-06-2023	Add diagrams	1.0

Table of Contents

1. Introduction

1.1 Purpose

The purpose of *"PetS"* mobile app is to provide a platform that enables pet owners to search for pet sitters based on their location, availability, and pet sitting experience, and request appointments with them. Pet sitters can accept or reject appointments, communicate with pet owners, and manage their schedules and payments.

1.2 Intended Audience

This document is intended to be read by the developers working on creating the system for the client, the project managers, the marketing staff and the testers.

Through this document there is detailed information about the application and its development.
It explains the purpose and main features of the system, external interface requirements and other non-functional requirements.

1.3 Scope

To put the proposed system in perspective, the result of the project will be a mobile application available both on iOS and Android platforms that allows pet owners (users) to find pet sitters (users) and vice versa. The system will include features such as user registration, location based search and booking of pet sitters services, real-time messaging, secure payment integration, rating and review system, and other relevant functionalities.

The main benefit that this application will provide is speed up the search process of pet owners and pet sitters. Other specific benefits according to the type of user are:

As a pet **owner**:

- Being able to travel meanwhile your pets will be well attended
- Make sure the tasks to do are completed according to previous personalized owner specifications
- Make sure your home and pets will be secure

As a pet **sitter**:

- Help you find a job near to your location and being able to communicate easily with the owners in case anything happens through the chat.
- Having information about the pet you will be sitting so that you can be prepared and aware of its necessities.
- Payment through the app in a fast and secure way.

1.4 References / Links

Github:

- HTTPS: https://github.com/IrenePorta/PetSitters.git
- GitHub CLI: gh repo clone IrenePorta/PetSitters

Trello: https://trello.com/b/NoWlRT8l/project-management-petsitters

PlantUML: https://plantuml.com/

1.5 Overview

The SRS document provides a comprehensive overview of the PetS mobile app. This document covers the following key aspects:

- **Introduction:** provides an overview of the purpose, intended audience, and scope of the application.
- **Product functions:** describes the main functions of the app, including user registration, search for pet sitters, and appointment management, messaging, payments, rating and review system, and user profile management.
- **Constraints:** outlines the general constraints and limitations that need to be considered during the development and implementation of the app, such as platform compatibility, data security, performance, and legal compliance.
- **User Stories:** presents a series of user stories that illustrate specific interactions and workflows between the app's actors (pet owners and pet sitters) and the system.
- **System Requirements:** includes both functional and non-functional requirements. Specifies each non-functional requirement associated with each use case, including performance, usability, security, reliability, and scalability.
- **Design Constraints:** specifies any specific design constraints that need to be followed during the app's development, such as design principles, branding, and user interface guidelines.
- **Appendices:** includes any additional information that may be relevant to the understanding of the app.

The SRS is organized in a structured manner to provide clarity and ease of navigation. The document begins with an introduction, followed by sections on product function, external interface requirements, user stories, system requirements, design constraints, and appendices.

Each section is divided into subsections, providing detailed information on specific topics. Additionally, the SRS may include diagrams or tables for a better understanding.

2. General Description

2.1 Product Perspective

This is a new application designed to be used by pet owners and sitters.
Is a standalone software application that runs on mobile devices, specifically designed for pet owners and pet sitters to connect and facilitate pet sitting services. The app operates independently from other systems, but it relies on external services such as: payment gateways, messaging services, and location services to provide its functionality.

2.2 Product Functions

The product functions of this mobile application are:

- **User registration:** allows users to create accounts as pet owners or pet sitters, providing personal information.
- **Search for pet sitters:** enables pet owners to search for pet sitters based on different criteria.
- **View pet sitters profile:** allows pet owners to view detailed profiles of selected pet sitters.
- **Request appointment:** enables pet owners to request appointments with selected pet sitters
- **Manage appointments:** allows both pet owners and pet sitters to manage their respective appointments within the app.
- **In-App messaging:** facilitates real-time communication between pet owners and pet sitters.

- **Make payment:** enables pet owners to make secure payments within the app for the services provided by pet sitters.
- **Rate and review:** Allows pet owners to rate and provide feedback on pet sitters based on their pet sitting experience.
- **User profile management:** allows both pet owners and pet sitters to manage their user profiles within the app.
- **Select appointment days:** enables pet owners to select desired days for pet sitting services using a calendar view.
- **Confirm appointment:** allows pet owners to review and confirm selected appointment details, finalizing the booking process and notifying the pet sitter.
- **Update appointment:** permits pet owners to update selected appointment dates in the calendar view if scheduling needs change.
- **Cancel appointment:** allows pet owners to cancel previously booked appointments if necessary.

These product functions provide the main capabilities of the application, facilitating connection between pet owners and pet sitters.

2.3 User Characteristics

There will be three types of users for this system. The first type will be the pet owners that are the customers who need to find reliable and trustworthy pet sitters to take care of their pets while they are away.The second type of user will be the pet sitter that are the service providers who offer their pet sitting services to pet owners and need to manage their schedules and appointments. As these two types of users have to interact by the chat of the app in order to reach an agreement we need a third type of user, the administrator. The administrative user will utilize the reports the user produces and penalize, if it is convenient, to a user.

2.4 General Constraints

The general constraint for PetS mobile application include:

- **Platform compatibility:** the app needs to be compatible with both Android and iOS systems to reach a wide user base.
- **Internet connectivity:** the app requires internet connection for full functionality. Users should have access to an internet connection to search for pet sitters, communicate, make payments…
- **User privacy and data security:** the app should implement appropriate measures to safeguard user information.
- **Integration with external services:** the app may need to integrate with external services for features such as secure payment processing, geolocation services, and messaging functionality.
- **Performance and scalability:** the app should be designed and optimized for efficient performance, with smooth navigation, quick response times, and minimal loading times. It should also be scalable to handle higher data and transaction volumes and accommodate a growing user base.
- **User interface and usability:** the user interface should be intuitive, making it easy for users to navigate, search for information, and perform desired actions.
- **Legal and regulatory compliance:** the app should comply with relevant legal and regulatory requirements, such as data protection laws and payment processing regulations.
- **Device compatibility:** the app should be compatible with a wide range of devices, screen sizes, and resolutions. It should adapt to different device configurations and orientations.
- **Availability and reliability:** the app should strive for high availability and reliability, minimizing downtime and ensuring that users can access the app without interruption.

- **Localization and internationalization:** The app may need to support multiple languages, currencies, and time zones to cater to a global user base. It should allow for localization and internationalization features to provide a personalized experience for users in different regions.

2.5 Assumptions and Dependencies

It is assumed that users have access to compatible smartphones with internet connectivity and that they will provide accurate information during the registration process. The application will depend on external services for features such as secure payment processing, geolocation services, and messaging functionality.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The user interface is a user-friendly and intuitive mobile application interface.
It should support different screen sizes and orientations, and be compatible with both iOS and Android.
The user interface includes: a registration screen for users to log in, a search screen to find pet sitters, a profile screen to view pet sitter details and reviews, a booking screen to manage appointments, and a chat screen for real-time communication.
User Interfaces should adhere to the accessibility and usability standards.

3.1.2 Hardware Interfaces

The PetS mobile app is designed to run on smartphone devices with sufficient processor speed and memory for better operation.
Mobile devices require internet connectivity capabilities (Wi-Fi/4G/5G) and GPS for location-based services.
The app should be compatible with different screen resolutions and sizes.
No specific hardware interface other than the one present on the smartphone is required to interact with the app.

3.1.3 Software Interfaces

The PetS mobile app will interact with iOS and Android operating systems requiring compatibility with the latest or some previous versions of the operating systems.
It will also interface with the server-side application for data storage and retrieval, requiring an API for interaction.
External software interfaces include: payment gateways for in-app transactions, messaging services for real-time communication, and location services for location-based search functionality.

3.1.4 Communications Interfaces

The PetS app will use standard internet protocols (HTTP/HTTPS) for communication with the server-side application. For real-time communication, it might utilize WebSockets or similar technologies. Additionally, for location services, it will use GPS communication protocols. For payments, secure communication protocols will be used to ensure safe transactions. Internet connectivity is a requirement for the app to function properly.

3.2 Use Cases

3.2.1 Use Case List

- UC-01 User registration
- UC-02 Search for Pet Sitter Profile
- UC-03 View Pet Sitter Profile
- UC-04 Request Appointment
- UC-05 Accept/Reject request
- UC-06 Manage appointments
- UC-07 In-App messages
- UC-08 Make Payment
- UC-09 Rate and Review
- UC-10 User Profile Management
- UC-11 Select Appointment days
- UC-12 Confirm Appointment
- UC-13 Update Appointment
- UC-14 Cancel Appointment

3.2.2 Use Case Table

Identifier	Use case	Actors	Description
UC-01	User registration	User (Pet Owner, Pet Sitter)	During the registration process, the user is prompted to specify whether they are registering as a pet owner or a pet sitter. This use case enables users, whether they are pet owners or pet sitters, to register for an account in the app by providing different types of information such as personal information, contact details depending on which user they are.
UC-02	Search for Pet Sitters	Pet owner	Allows pet owners to search for pet sitters based on various criteria such as location, availability, services offered, ratings, and pet-specific requirements. Pet owners can view a list of matching pet sitters along with their profiles, ratings, and reviews. The search functionality helps pet owners find suitable pet sitters for their specific needs.
UC-03	View Pet Sitter Profile	Pet Owner	Enables pet owners to view the detailed profile of a selected pet sitter. Pet owners can access information about the sitter's experience, service offering, sitter's experience, ratings and reviews. Viewing pet sitter profiles helps pet owners users make informed decisions when selecting a suitable setter for their pet.
UC-04	Request Appointment	Pet Owner	Enables pet owners to make an appointment with a particular pet sitter. The preferred date, time, duration, and any additional needs or instructions for the pet sitting service can be specified by the pet owner. The request is sent to the chosen pet sitter for review and confirmation.
UC-05	Accept/Reject Request	Pet Sitter	Enables pet sitters to accept or reject appointment requests based on their availability and preferences.
UC-06	Manage Appointments	Pet Owner, Pet Sitter	Allows both pet owners and pet sitters to manage their respective appointments within the app. Can view their upcoming appointments, modify appointment details, reschedule appointments if needed, and cancel appointments when necessary. Managing appointments provides flexibility and control for pet owners in their scheduling and coordination with pet sitters.
UC-07	In-App Messaging	Pet Owner, Pet Sitter	Facilitates real-time communication between pet owners and pet sitters within the app. Users can exchange messages to discuss appointment details, pet care instructions, and any other relevant information. In-app messaging provides a convenient and secure channel for communication.

UC-08	Make Payment	Pet Owner	Enables pet owners to make secure payments within the app for the services provided by pet sitters. After the completion of pet sitting services, pet owners can initiate payments, view payment details, and choose from various payment options. This use case ensures a seamless and convenient payment process for pet owners.
UC-09	Rate and Review	Pet Owner	Allows pet owners to provide ratings and write reviews for pet sitters based on their pet sitting experience. Pet owners can rate various aspects of the service, such as the quality of care, communication, reliability, and overall satisfaction. The ratings and reviews help other pet owners make informed decisions when selecting a pet sitter.
UC-10	User Profile Management	Pet Owner, Pet Sitter	Allows both pet owners and pet sitters to manage their user profiles within the app. Users can update their personal information, contact details, profile pictures, service offerings, pet information, and other relevant details. User profile management ensures that user information is up to date and accurately represents their preferences and capabilities.
UC-11	Select Appointment Days	Pet Owner	Enables pet owners to select the desired days in a calendar view when they require pet sitting services. Pet owners can navigate through the calendar, choose specific dates or date ranges, and mark their availability for pet sitting. Selecting appointment days helps pet owners communicate their scheduling preferences to pet sitters.
UC-12	Confirm Appointment	Pet Owner	Allows pet owners to confirm the selected appointment days for pet sitting services, finalizing the booking process. Pet owners review the chosen dates, verify the details, and provide confirmation to the pet sitter. Confirming the appointment ensures that both parties are aware of the scheduled pet sitting arrangement.
UC-13	Update Appointment	Pet Owner	Permits pet owners to update the selected appointment days in the calendar view, modifying the previously booked pet sitting services. They can change the dates or time, or make adjustments to the pet sitting requirements. Updating appointments allows pet owners to accommodate any changes or unexpected circumstances.
UC-14	Cancel Appointment	Pet Owner	Allows pet owners to cancel a previously booked appointment for pet sitting services. Pet owners can navigate to their scheduled appointments, select the desired appointment, and cancel it if necessary. Canceling appointments provides flexibility and allows pet owners to adjust their plans or seek alternative arrangements.

3.3 System Requirements

3.3.1 System Requirements List

Functional requirements	**Non-functional requirements**
F-01: User registration	NF-US: Usability
F-02: Search for Pet Sitters	NF-PE: Performance
F-03: View Pet Sitter Profile	NF-SE: Security
F-04: Request Appointment	NF-REL: Reliability
F-05: Accept/Reject Request	NF-VA: Validation
F-06: Manage Appointments	NF-NO: Notifications
F-07: In-App Messaging	NF-TRA: Tracking
F-08: Make Payment	NF-AU: Auditing
F-09: Rate and Review	NF-AV: Availability
F-10: User Profile Management	NF-SY: Synchronization
F-11: Select Appointment Days	NF-AD: Adaptability
F-12: Confirm Appointment	NF-RES: Responsiveness
F-13:Update Appointment	
F-14: Cancel Appointment	

3.4 Functional Requirements

3.4.1 Functional Requirements Tables

F-01 User Registration Version: v1.0
The user access to a registration page and enters their personal information and contact details such as:

- Name
- e-mail
- Phone number
- Address
- Gender
- Age

Later one the user has to select their role: Pet Owner or Pet Sitter. Depending on the selected role the user will have to give the given Role-Specific Information.
The user will have to create and confirm their password to complete the registration.
After the registration they have to do an account verification by login for the first time to validate the information. If everything is valid and correct the account will be created.

Comments:
If the person selects the role wrongly they can go back to select a different role.
The password should follow the following requirements:

- 8 to 10 characters
- A combination of uppercase letters, lowercase letters, numbers, and symbols.

If the information is invalid the user will return to the registration page where they have to enter their personal information to correct any errors.
Some Role-Specific information are:

Pet Owner:

- The user provides additional details about their pet(s), including the pet's name, breed, age, and any specific care instructions.

Pet Sitter:

- The user provides details about their experience, qualifications, availability, and services they offer as a pet sitter.

Relationships: F-02

F-02 Search for Pet Sitters v1.0
Pet owners can search for available pet sitters based on different criteria:

- Location
- Availability
- Services offered
- Pet-specific requirements

The system presents a list of potential pet sitters that meet the selected criteria, along with their profiles, ratings, and reviews. Pet owners can then make an informed decision on which pet sitter to choose.

Comments:
Pet owners can adjust the search criteria and run the search again if they don't find a suitable pet sitter.

Relationships: F-01, F-03

F-03 View Pet Sitter Profile v1.0
Pet owners can view detailed profiles of pet sitters which include:
- Experience
- Qualifications
- Service offerings
- Availability
- Ratings and reviews from previous customers

This use case provides the necessary information to the pet owners to make an informed decision when choosing a pet sitter.

Comments:
Pet owners can navigate back to the list of pet sitters if they want to check another profile.

Relationships: F-02, F-04

F-04 Request Appointment v1.0
Pet owners can request an appointment with a pet sitter by specifying the:
- Preferred date
- Preferred time
- Duration of the service

Any specific requirements for the pet sitting service will be specified.
The request is then sent to the pet sitter for review and confirmation.

Comments:
The appointment is not final until the pet sitter has accepted the request. Pet owners can cancel or modify the request before it is accepted.

Relationships: F-03, F-05, F-06

F-05 Accept/Reject Request v1.0
Pet sitters receive appointment requests from pet owners and have the ability to accept or reject these requests. Upon receiving a request, the pet sitter reviews the details of the request, including the:
- Preferred date and time
- Duration of the service
- Any specific requirements

The pet sitter can then decide to accept or reject the request based on their availability and the details of the request.

Comments:
Pet sitters should respond to requests within a certain timeframe to maintain a reliable service. If a request is rejected, the pet sitter could optionally provide a reason or suggest alternative dates or times.

Relationships: F-04, F-06

F-06 Manage Appointments v1.0
Pet owners can manage their appointments within the app. They can:
- View their upcoming appointments
- Modify appointment details
- Reschedule appointments
- Cancel appointments

Provides flexibility and control over pet owners schedules.

Comments:
The ability to modify an appointment may be subject to the pet sitter's availability and policy on changes or cancellations.

Relationships: F-04, F-05, F-07, F-08

F-07 In-App Messaging v1.0
Pet owners and pet sitters can communicate with each other through in-app messaging. They can exchange messages to:
- Discuss appointment details
- Provide pet care instructions
- Share any other relevant information

Comments:
The messaging service should ensure privacy and security. Users' contact information should not be disclosed through this service.

The messaging functionality should support real-time messaging between pet owners and pet sitters, allowing for quick and efficient communication.

The messaging system should support the exchange of multimedia content such as photos, videos, or documents to facilitate communication related to pet care.

The messaging system should maintain a history of messages exchanged between pet owners and pet sitters, allowing users to refer back to previous conversations if needed.

Relationships: F-05, F-06, F-08

F-08 Make Payment v1.0
Upon completion of the pet sitting service, pet owners can make payments within the app. They can:
- Initiate payments
- View payment details
- Choose from various payment options

Comments:
The payment process should be secure.

The system should support multiple payment options, such as credit cards, debit cards, digital wallets, or other popular payment methods, to accommodate user preferences.

Relationships: F-06, F-07

F-09 Rate and Review v1.0
Enable pet owners to rate and review pet sitters.
The rating and review system should allow pet owners to provide feedback on pet sitters, ensuring a transparent and reliable evaluation process.
The system should accurately capture and display ratings and reviews provided by pet owners, preventing manipulation or fraudulent activities. It should include moderation mechanisms to prevent inappropriate or offensive content from being published.

Relationships: -

F-10 User Profile Management v1.0
Allow users to manage their profiles.
The system should ensure the integrity of user profile data, preventing unauthorized access, modification, or deletion.
The profile management system should validate input data to ensure the accuracy and consistency of the provided information.

Relationships: -

F-11 Select Appointment Days v1.0
Pet owners can select desired days for pet sitting services using a calendar view. They can navigate through the calendar, choose specific dates or date ranges, and mark their availability for pet sitting.
Before the dates are actualized in the calendar, the user has to confirm them.

Comments:
The user can change as many times as needed the selected days of the calendar.

The selection of appointment days helps communicate scheduling preferences to pet sitters and allows for easier planning and coordination.

Relationships: -

F-12 Confirm Appointment v1.0
Pet owners can confirm selected appointment days, thereby finalizing the booking process. This includes:
- Review the chosen dates
- Verify the details
- Send a confirmation to the pet sitter

Later on, the system records the user's confirmation, updates the appointment status and the user receives a confirmation message of the appointment.

Comments:
Confirming the appointment ensures that both parties are aware of the scheduled pet sitting arrangement.

Relationships: F-11, F-06

F-13 Update Appointment v1.0
Pet owners have the ability to update an existing appointment. This can include changing the date or time, adjusting the duration of the service, or updating any specific requirements for the pet sitting service. Upon updating, a notification should be sent to the pet sitter informing them of the changes.

Comments:
The ability to update an appointment may be subject to the pet sitter's availability and policy on changes. These policies should be clearly communicated to the pet owner.

Relationships: F-04, F-05, F-12

F-14 Cancel Appointment v1.0
Pet owners have the ability to cancel an appointment. This action should send a notification to the pet sitter, informing them of the cancellation.

Comments:
Cancellation policies may apply, including potential fees for last-minute cancellations. These policies should be clearly communicated to the pet owner.

The system should provide confirmation prompts or notifications to pet owners before finalizing the cancellation, ensuring they are aware of the cancellation consequences and any applicable fees or restrictions.

Relationships: F-05, F-12

3.5 Non-Functional Requirements

3.5.1 Non-Functional Requirements Tables

NF-US: Usability v1.0
The system should be user-friendly, intuitive and easy to understand, with clear instructions

NF-PE: Performance v1.0
The system should deliver fast and responsive performance across various functionalities, including registration, search, profile page loading, appointment management, and calendar functionality. It should handle high volumes of data, requests, appointments, and selectable dates efficiently, ensuring quick response times and minimal delays.

NF-SE: Security v1.0
The system should ensure the privacy and security of user interactions, including messaging and payments transactions.

NF-REL: Reliability v1.0
The system should maintain a high level of reliability across the appointment request, payment, and confirmation processes, minimizing the risk of errors, failures, or inconsistencies and ensuring the accurate and timely delivery of requests, processing payments and confirmation of bookings.

NF-VA: Validation v1.0
The system should implement data validation mechanisms throughout the request form, profile management, and appointment update processes to ensure the accuracy, completeness, and consistency of user-provided information, preventing errors and inconsistencies.

NF-NO: Notifications v1.0
The system should send notifications to both pet owners and pet sitters, informing them about appointment requests, confirmations, changes, cancellations, new messages, and updates.

NF-TRA: Tracking v1.0
The system should track the status of the appointment request, allowing pet owners to know whether the request is pending, accepted, or rejected.

NF-AU: Auditing v1.0
The appointment request system should log and store relevant information about the request, including the date, time, and additional requirements.

NF-AV: Availability v1.0
The profile information should be available and up to date, reflecting the current status, experience, and service offerings of the pet sitter. The system should be accessible and available to users (pet sitters and pet owners) at all times, allowing them to manage appointments as needed.

NF-SY: Synchronization v1.0
The appointment management system should synchronize appointment updates across multiple devices and platforms to ensure consistent information for pet owners.

NF-AD: Adaptability v1.0
The profile page should be responsive and adapt to different screen sizes and devices to ensure a consistent user experience.

NF-RES: Responsiveness v1.0
The system should provide a fast and responsive experience when updating appointments, ensuring that changes are reflected promptly.

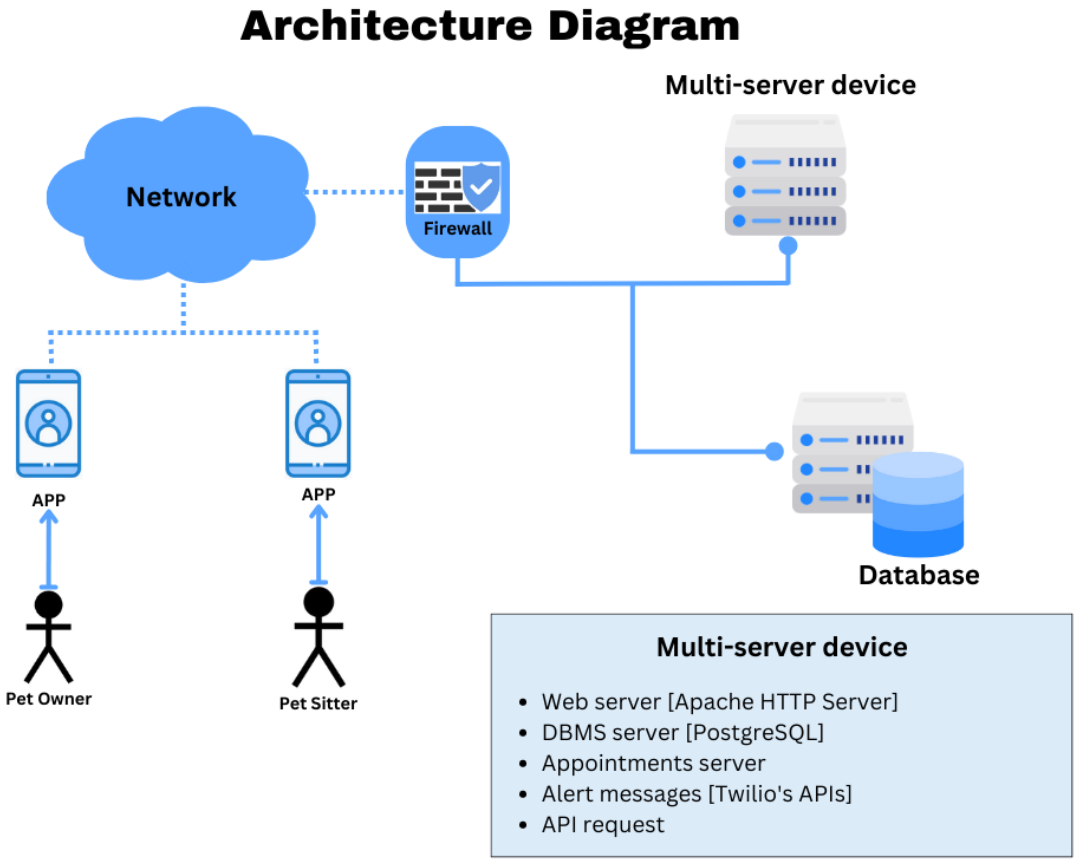
3.6 Design Constraints

"PetS" application includes the following design constraints:

- Standards enforcement: The system needs to adhere to specific design principles
- Hardware limitations: The system should consider the limitations of the hardware such as storage limit and network connectivity. The system should be compatible with various devices, support different time zones and multiple languages for global use.
- Error management: The system should have a mechanism to handle errors and exceptions and provide error messages to the users.
- Security: The system should incorporate security measures to protect the personal information of the users.

4. Analysis Models

4.1 Architecture Diagram



Repository file path:

- PNG: Documents/Architecture_Diagram.png

Description:

This is a distributed client-server architecture that consists of two types of users: Pet Owners and Pet Sitters. Both of these users interact with the system via a mobile app installed on their smartphones.

Mobile App: The mobile application is the primary interface for the system. It provides all the necessary features and services such as searching for a pet sitter (for pet owners), accepting or rejecting appointments (for pet sitters), managing profiles, and more. The app is designed to handle requests from two different types of users, Pet Owners and Pet Sitters, each with distinct roles and functionalities.

Network: The mobile app connects to the internet to communicate with the back-end servers. This network connection allows for the exchange of data between the app and the servers.

Firewall: A security system that monitors and controls incoming and outgoing network traffic based on predetermined security rules. It establishes a barrier between the secure internal network (where the servers are located) and the potentially less secure external network (where the mobile app operates).

Multi-server: This is where the core of the processing takes place (back-end tasks). It consists of several servers, each with its own specific task:

Web Server *(e.g., Apache HTTP Server):* Handles HTTP requests from the mobile application and returns HTTP responses, usually with data to be used by the mobile application.

DBMS Server *(e.g., PostgreSQL):* Handles requests to read or write data from/to the database. The data could include user profiles, appointments, reviews, etc.

Appointments Server: Specifically manages the creation, update, and cancellation of appointments. It may have complex logic to handle scheduling conflicts, rescheduling, and other appointment-related tasks.

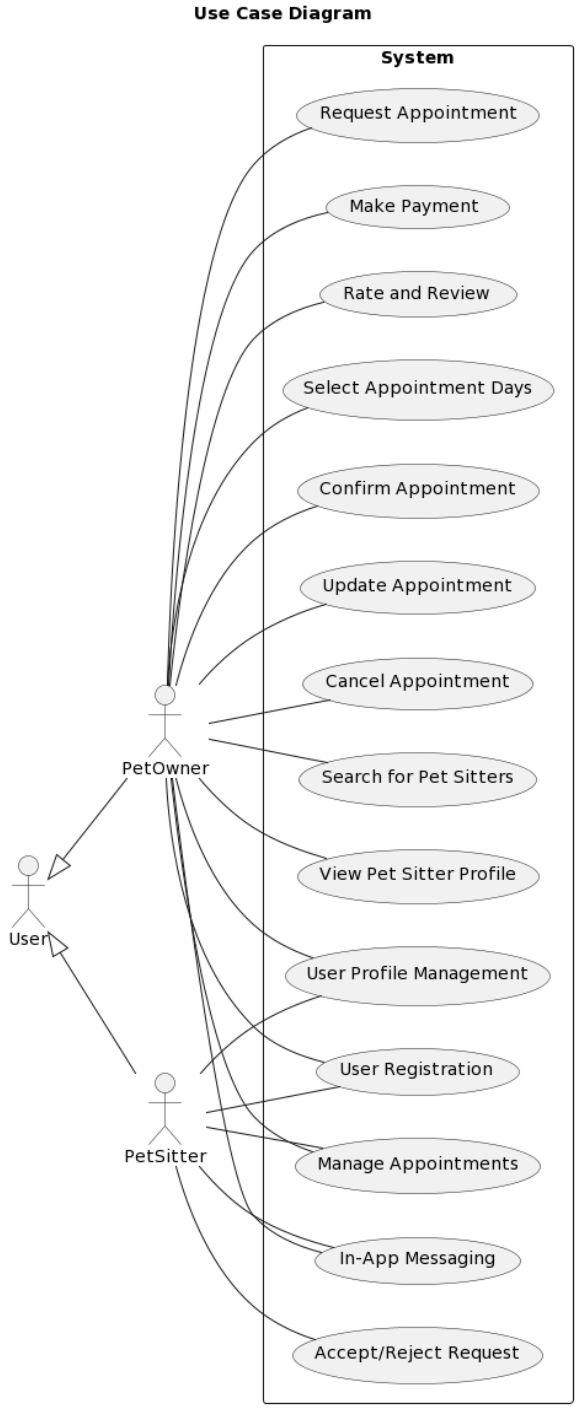
Alert Messages System *(e.g., Twilio's APIs):* Handles the sending of alert messages to users. These could be reminders about upcoming appointments, notifications about changes to appointments, or other important alerts.

API Request Server: Handles other API requests that may not fall into the categories above. These could be requests to third-party services, requests for data not stored in the database, etc.

Database Server: This is a specialized server where the database management system (DBMS) software runs and where the data is physically stored. In this architecture, PostgreSQL has been mentioned as the DBMS of choice. This would include user profiles, pets' details, appointment schedules, reviews, and more.

The architecture is designed to be scalable and efficient, allowing for the system to handle a large number of users and data while providing a smooth and responsive user experience. It's also designed to be secure, with a firewall protecting the servers from potentially malicious traffic.

4.2 Use Case Diagram



Description:

This diagram represents the use case diagram of the PetS application. The diagram shows two main actors: “User” and its two sub-actors “PetSitter” and “PetOwner”. This implies that the “User” can take the role of a “PetOwner” or a “PetSitter” within the application.

The “User” actor is connected to multiple use cases representing the actions or functionalities available to the user.

The “PetOwner” actor is associated with the following use cases: User Registration, Search for Pet Sitters, View Pet Sitter Profile, Request Appointment, Manage Appointments, In-App messaging, Make Payment, Rate and Review, User Profile Managements, Select Appointment Days, Confirm Appointment, Update Appointment and Cancel Appointment.

The “PetSitter” actor includes: User Registration, Manage Appointments, Accept/Reject Request, In-App Messaging, and User Profile Management.

The use cases are organized within the “System” rectangle, indicating that they are part of the application’s functionalities.

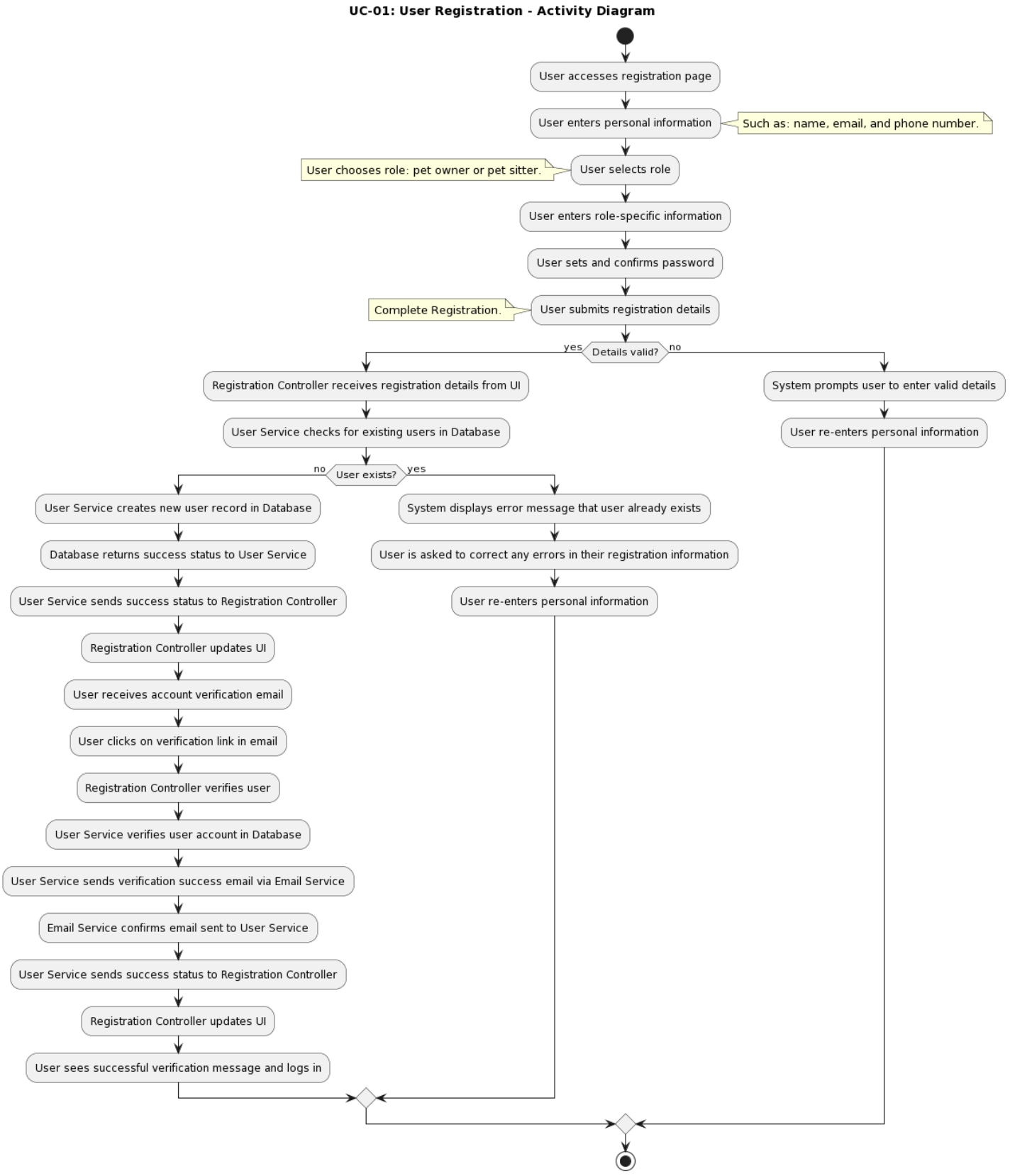
Overall, the use case diagram provides a high-level overview of the interactions between the actors (users) and the system in the "PetS" application. It illustrates the various functionalities available to pet owners and pet sitters, showcasing the key actions they can perform within the system.

Repository file path:

- PNG: Documents/PlantUML/UseCaseDiagram.png
- PUML: Documents/PlantUML/UseCaseDiagram.puml

4.3 Activity Diagrams

4.3.1 UC-01: User registration



Repository file path:

- PNG: Documents/PlantUML/UC-01_ACTIVITY_Diagram.png
- PUML: Documents/PlantUML/UC-01_ACTIVITY_Diagram.puml

Description:

The diagram represents the user registration process for the application.

The process starts when the user accesses the registration page and provides its personal information selecting their role in the app, Pet owner or Pet sitter. In each case it will introduce different information corresponding to their role. In order to submit the registration details they have to create a password.

After that the system checks if the entered details are valid. If not, the system will ask to correct the errors and the user has to re-enter their personal information.

When the details are valid, the Registration Controller receives this information from the User Interface and the User Service checks in the database if this user already exists.

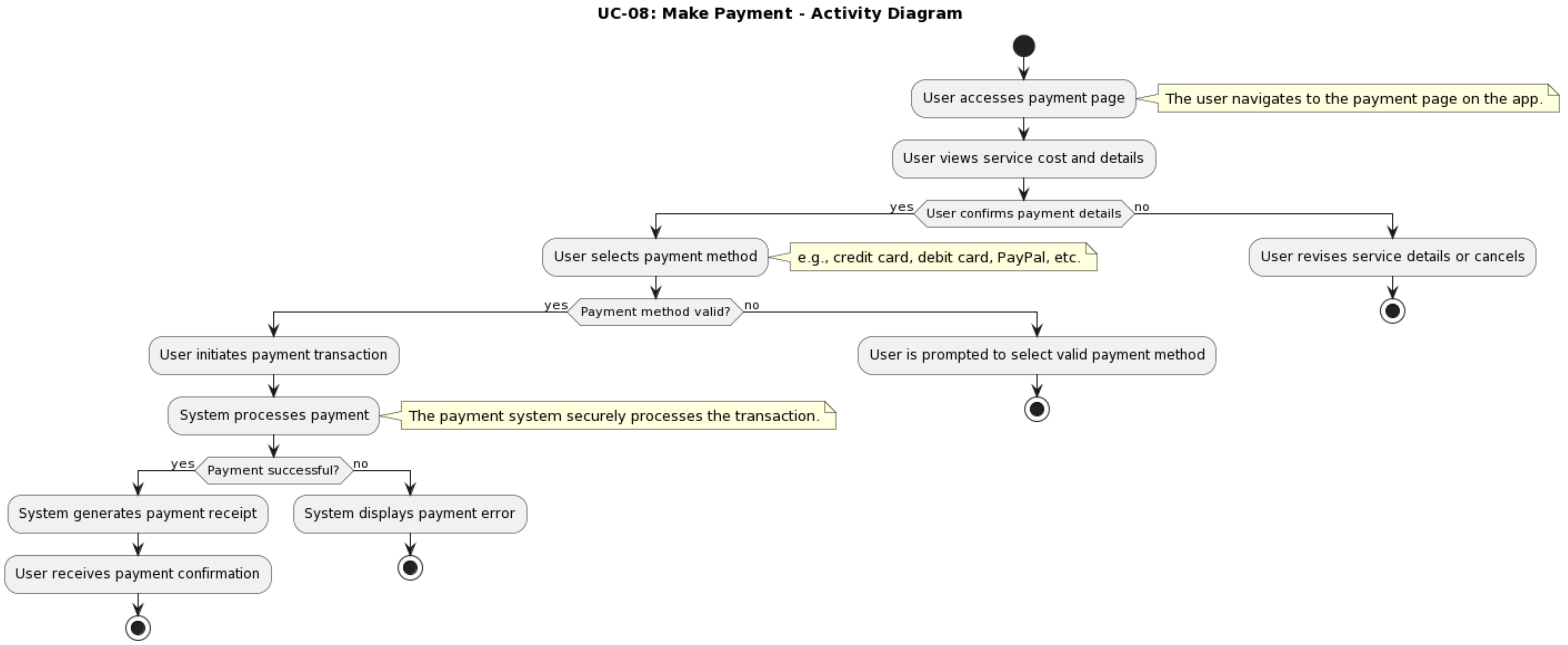
If a user with the same information exists, the system sends an error message to the user, asking them to correct any errors in their registration information and the user has to re-enter their personal information.

If no user exists, the User Service creates a new user record in the Database, which then returns a success status to the User Service. The User Service sends this new status to the Registration Controller which updates the User Interface.

The user then receives an account verification email. The user clicks the verification link in this email, and the Registration Controller verifies the user. The User Service verifies the user account in the Database, then sends a verification success email to the user via the Email Service.

The Email Service confirms that the email has been sent to the User Service, which then sends a success status to the Registration Controller. The Registration Controller updates the User Interface, and the user sees a successful verification message and logs in.

4.3.2 UC-08: Make Payment



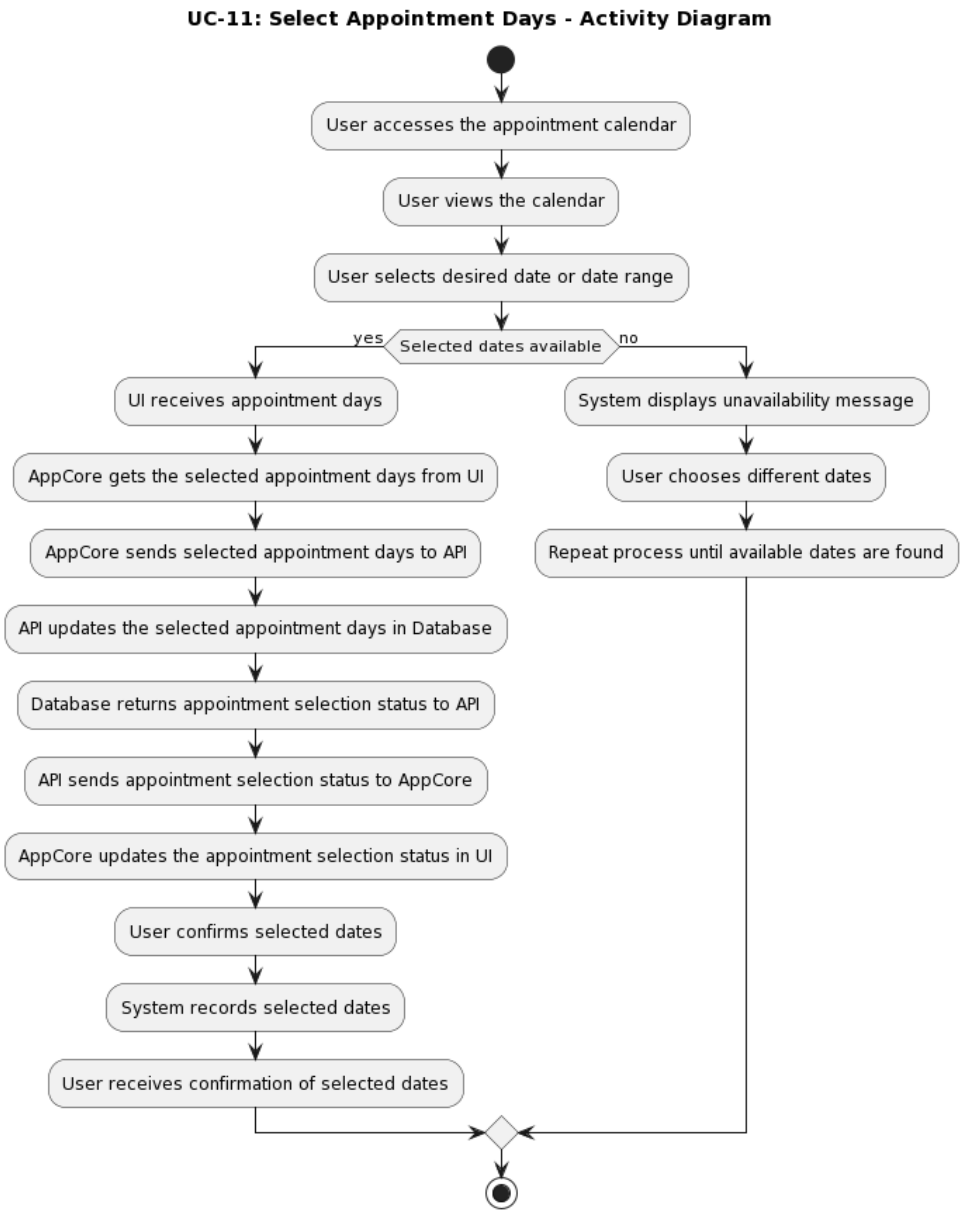
Repository file path:

- PNG: Documents/PlantUML/UC-08_ACTIVITY_Diagram.png
- PUML: Documents/PlantUML/UC-08_ACTIVITY_Diagram.puml

Description:

The diagram shows the process of making a payment within an application.
The process starts when the user accesses the payment page in the application.
The user then views the cost and details of the service they are purchasing.
The next step depends on whether the user confirms the payment details. If the user doesn't confirm or cancel the process, then it stops.
After the user confirms the purchase they have to select the payment method such as credit card, debit card, and PayPal.
Then the payment method selected has to be validated, if the payment it's not valid the user will be asked to select another one and the process stops.
If the payment method is valid, the user initiates the payment transaction, and the system processes the payment. If the payment transaction is successful, the system generates a payment receipt, and the user receives a confirmation of the payment, ending the process. However, if the payment isn't successful, the system displays a payment error, and the process stops.

4.3.3 UC-11: Select Appointment Days



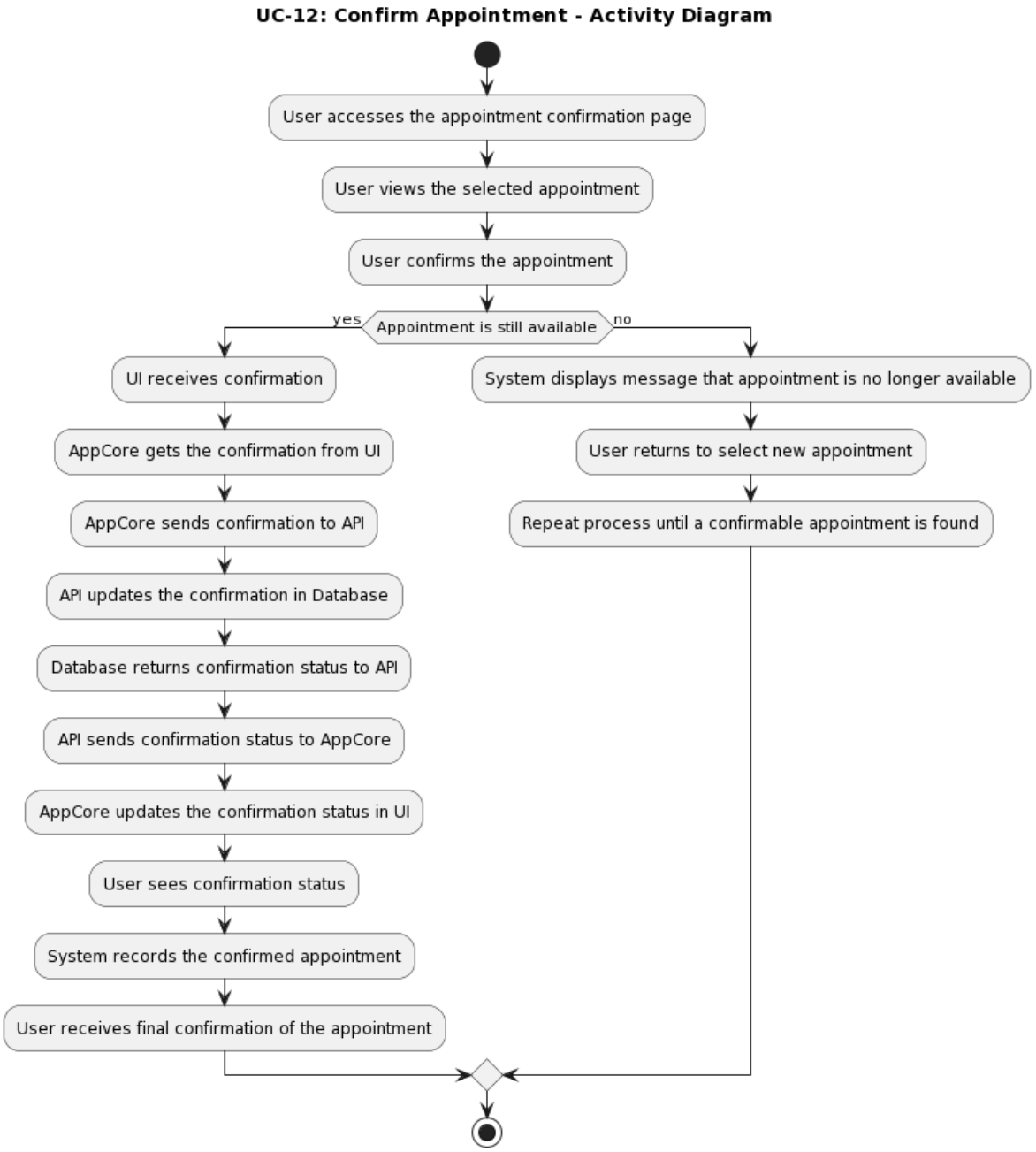
Repository file path:

- PNG: Documents/PlantUML/UC-11_ACTIVITY_Diagram.png
- PUML: Documents/PlantUML/UC-11_ACTIVITY_Diagram.puml

Description:

The diagram represents the process of selecting the available appointment dates.
The process starts when the user accesses the appointment calendar page in the application.
Then the user has to select the dates he is interested in, if there's availability, the system goes through the process of updating and confirming the appointment selection. If the selected dates are not available, the system displays an unavailability message and the user chooses different dates. The process then repeats until available dates are found.

4.3.4 UC-12: Confirm Appointment



Repository file path:

- PNG: Documents/PlantUML/UC-12_ACTIVITY_Diagram.png
- PUML: Documents/PlantUML/UC-12_ACTIVITY_Diagram.puml

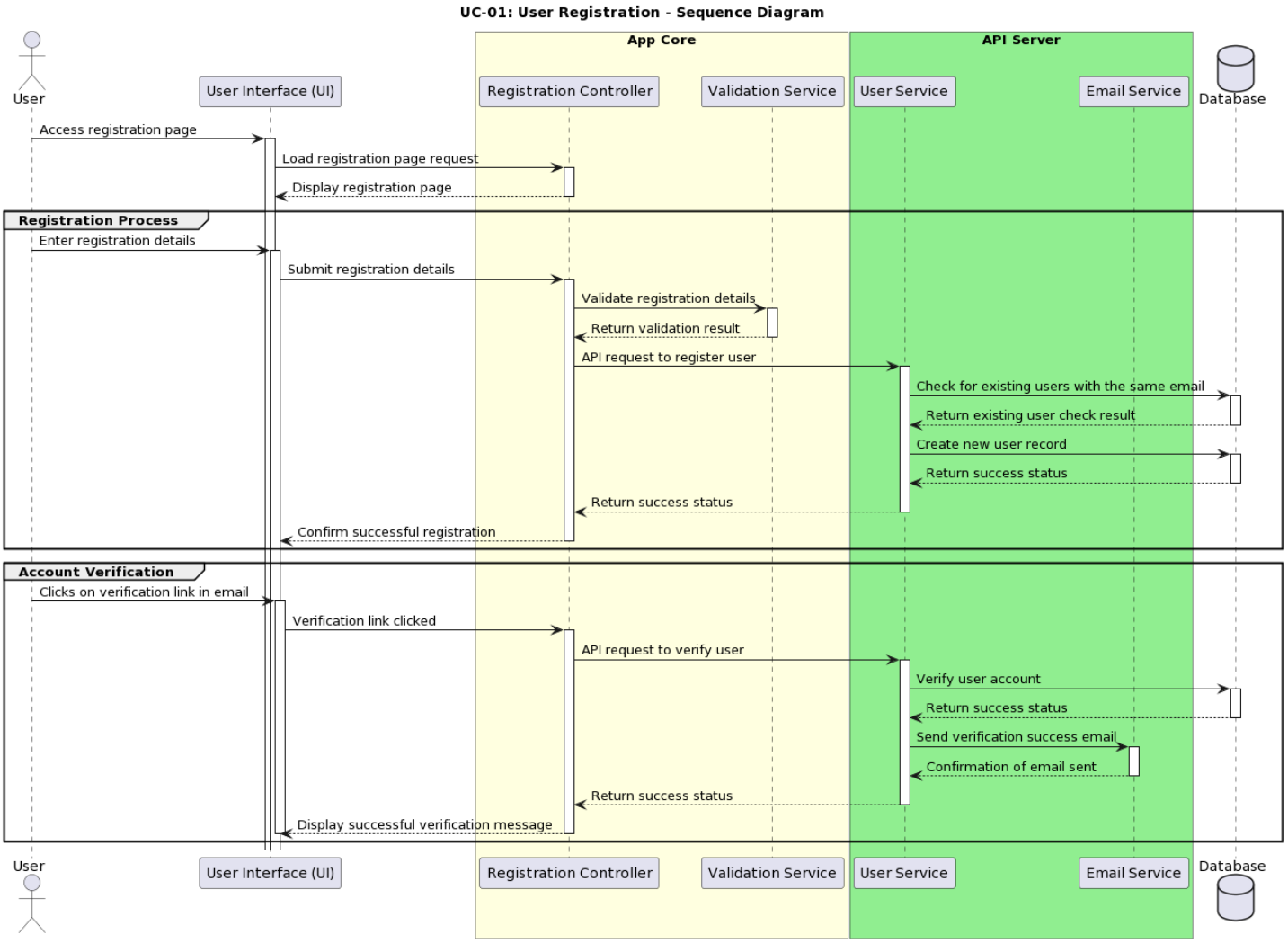
Description:

This diagram shows the confirmation of the previously selected appointment dates.

In this activity diagram, the user navigates to the appointment confirmation page and views the selected appointment. The user then confirms the appointment, at which point the system checks whether the appointment is still available. If it is, the system goes through the process of recording and confirming the appointment. If the appointment is no longer available, the system displays a message to that effect and the user is returned to the appointment selection process. The process repeats until the user is able to confirm an available appointment.

4.4 Sequence Diagrams

4.4.1 UC-01: User Registration



Repository file path:

- PNG: Documents/PlantUML/UC-01_SEQUENCE_Diagram.png
- PUML: Documents/PlantUML/UC-01_SEQUENCE_Diagram.puml

Description:

This sequence diagram represents the process of user registration and account verification.

The process starts when the user accesses the registration page. The User Interface (UI) receives this command and requests the Registration Controller to load the registration page. Once the page is loaded, the UI displays it to the user.

After that, the user enters their registration details, they arrive at the UI and then are sent to the Registration Controller. The Registration Controller sends these details to the Validation Service, which checks if the information provided is valid. The Validation Service sends the validation result back to the Registration Controller.

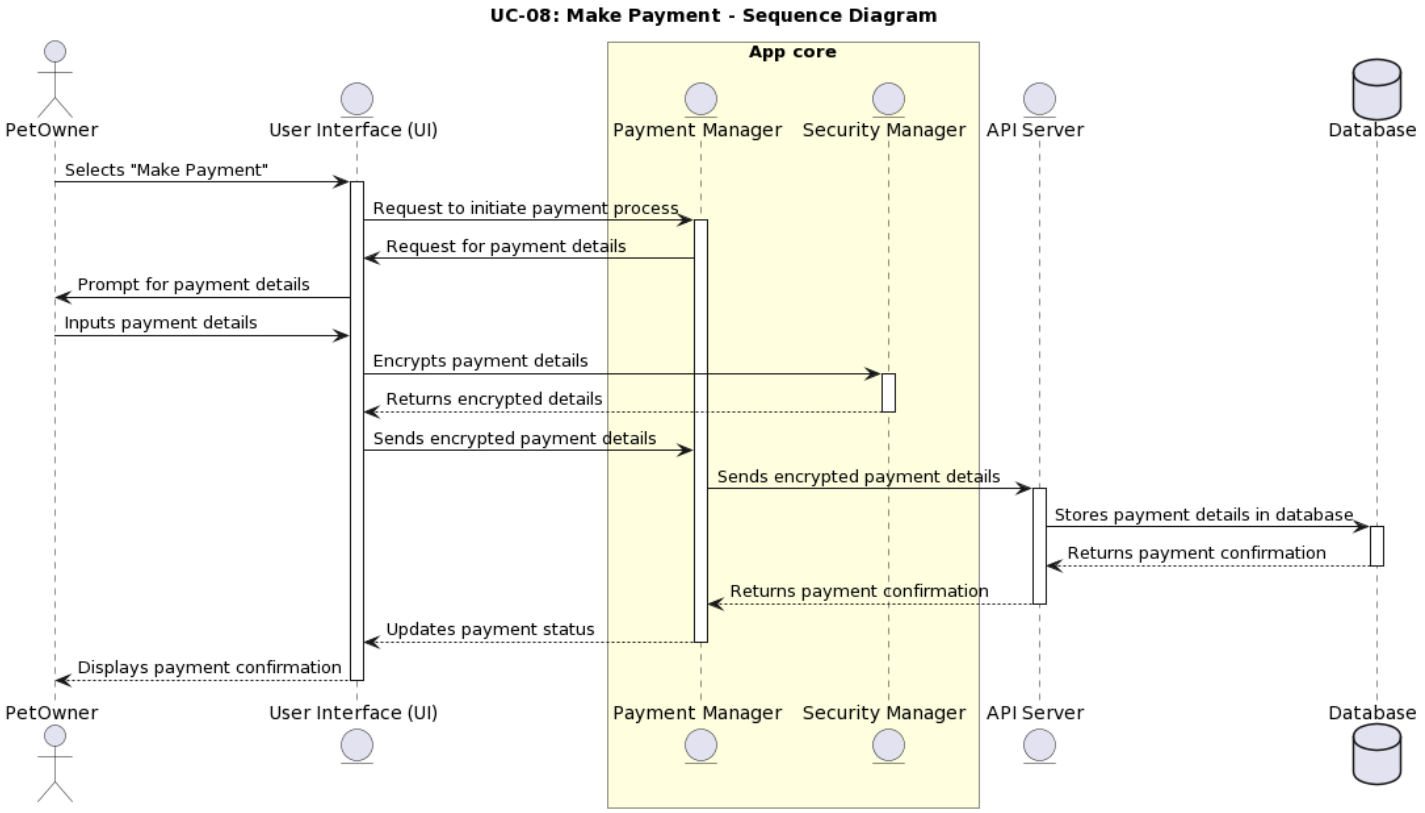
Once the details are validated, the Registration Controller makes an API request to the User Service to register the user. The User Service then checks in the database for any existing users with the same email. If there are no

existing users, it creates a new user record in the database, which in turn, returns a success status back to the User Service. This success status is passed back to the Registration Controller and then to the UI, which confirms successful registration to the user.

For account verification, the user receives an email with a verification link and clicks on it. The UI captures this event and notifies the Registration Controller. The Registration Controller makes another API request to the User Service to verify the user. The User Service verifies the user account in the database, which returns a success status back to the User Service.

Next, the User Service communicates with the Email Service to send a verification success email. The Email Service confirms the email has been sent and passes this confirmation back to the User Service. The User Service, in turn, sends a success status to the Registration Controller. The Registration Controller updates the UI to display a successful verification message to the user, marking the end of the process.

4.4.2 UC-08: Make Payment



Repository file path:

- PNG: Documents/PlantUML/UC-08_SEQUENCE_Diagram.png
- PUML: Documents/PlantUML/UC-08_SEQUENCE_Diagram.puml

Description:

This diagram shows the interactions between the components involved in the “Make Payment” use case.
It begins with the actor "PetOwner" interacting with the user interface (UI) component.
The “UI” component is activated when the PetOwner selects “Make Payment”. It asks AppCore for payment information.

The "App core" box represents the core components of the application, including the "Payment Manager" and "Security Manager."

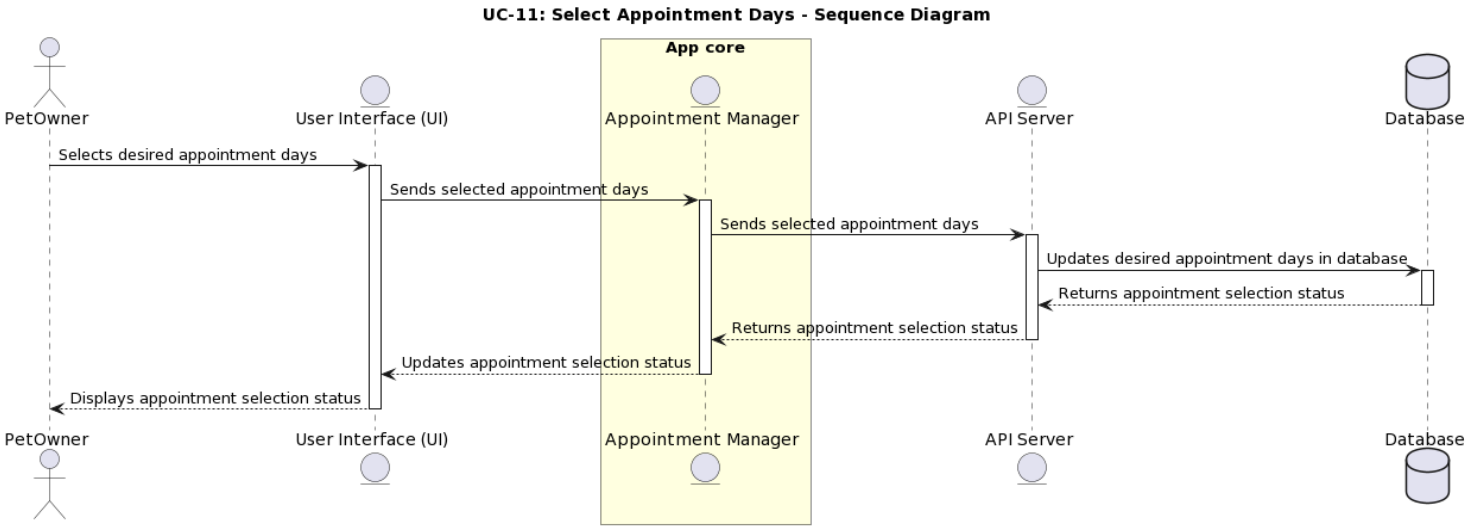
The UI prompts the PetOwner for payment details, and the PetOwner provides the necessary information.
To encrypt the payment information, the UI then communicates with the "Security Manager.", which it is turned on, and it sends the UI the encrypted data.

The AppCore receives the encrypted payment information sent by the UI.
Then the AppCore interacts with the API Server to send the encrypted payment details, this makes the API Server to be activated.

The API Server communicates with the Database to store the payment details. The Database is activated, and the confirmation of the payment is returned to the API Server. The API Server returns the payment confirmation to the AppCore and the AppCore updates the payment status and returns the updated information to the UI.

Finally, the UI displays the payment confirmation to the PetOwner.

4.4.3 UC-11: Select Appointment Days



Repository file path:

- PNG: Documents/PlantUML/UC-11_SEQUENCE_Diagram.png
- PUML: Documents/PlantUML/UC-11_SEQUENCE_Diagram.puml

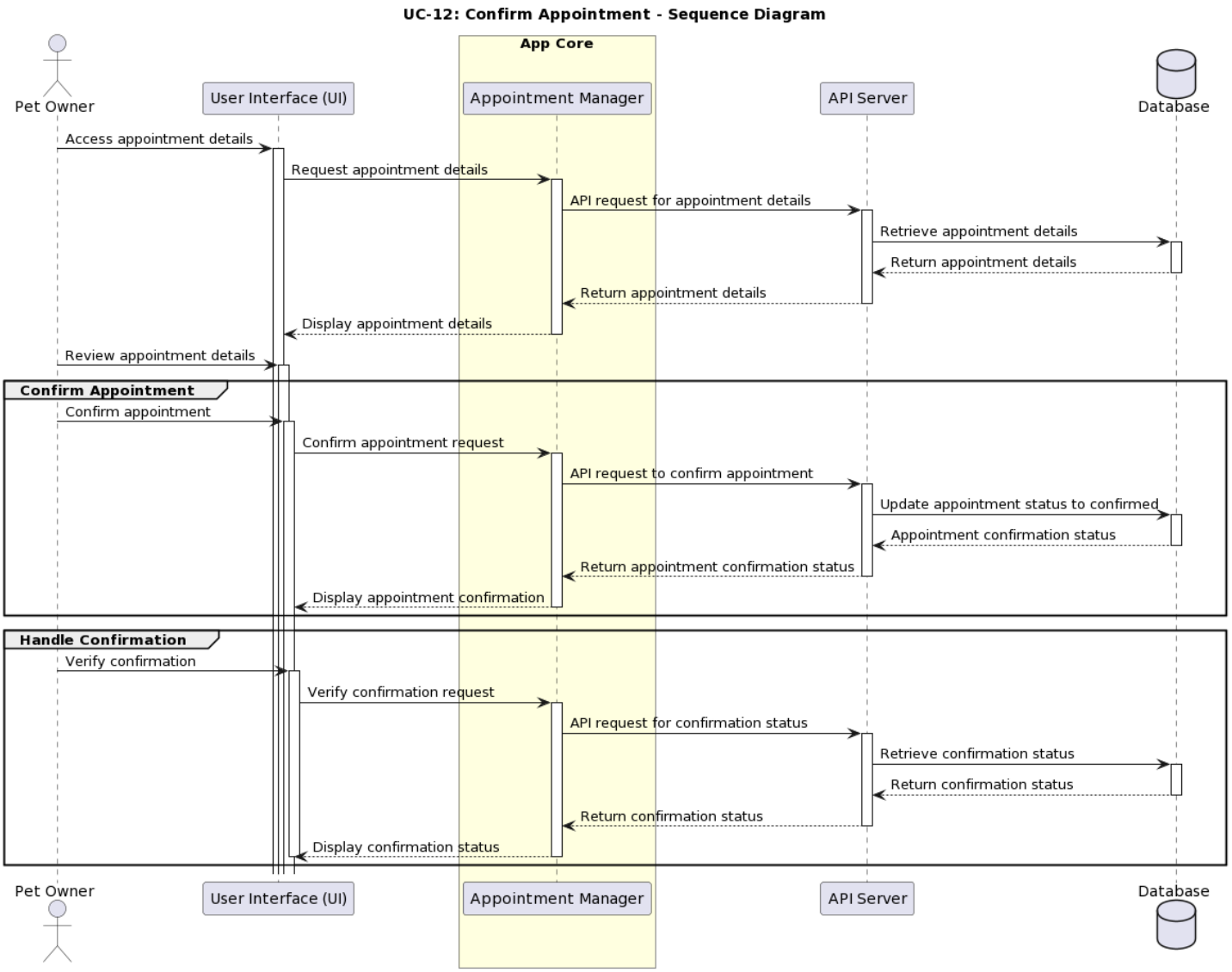
Description:

This diagram represents the simplified sequence of steps for the "Select Appointment Days" use case (UC-11). The main entities involved in this process are the PetOwner, User Interface (UI), the Appointment Manager in the App Core, the API Server, and the Database.

The PetOwner starts by interacting with the UI to select the desired appointment days. The UI communicates this selection to the App Core, where the Appointment Manager takes control. The Appointment Manager sends these selections to the API Server, which updates the database with the new appointment dates.

The Appointment Manager in the App Core is notified of a successful update by the database, which also notifies the API Server of the operation's status. UI, letting the PetOwner know if the appointment selection was successful or not.

4.4.4 UC-12: Confirm Appointment



Repository file path:

- PNG: Documents/PlantUML/UC-12_SEQUENCE_Diagram.png
- PUML: Documents/PlantUML/UC-12_SEQUENCE_Diagram.puml

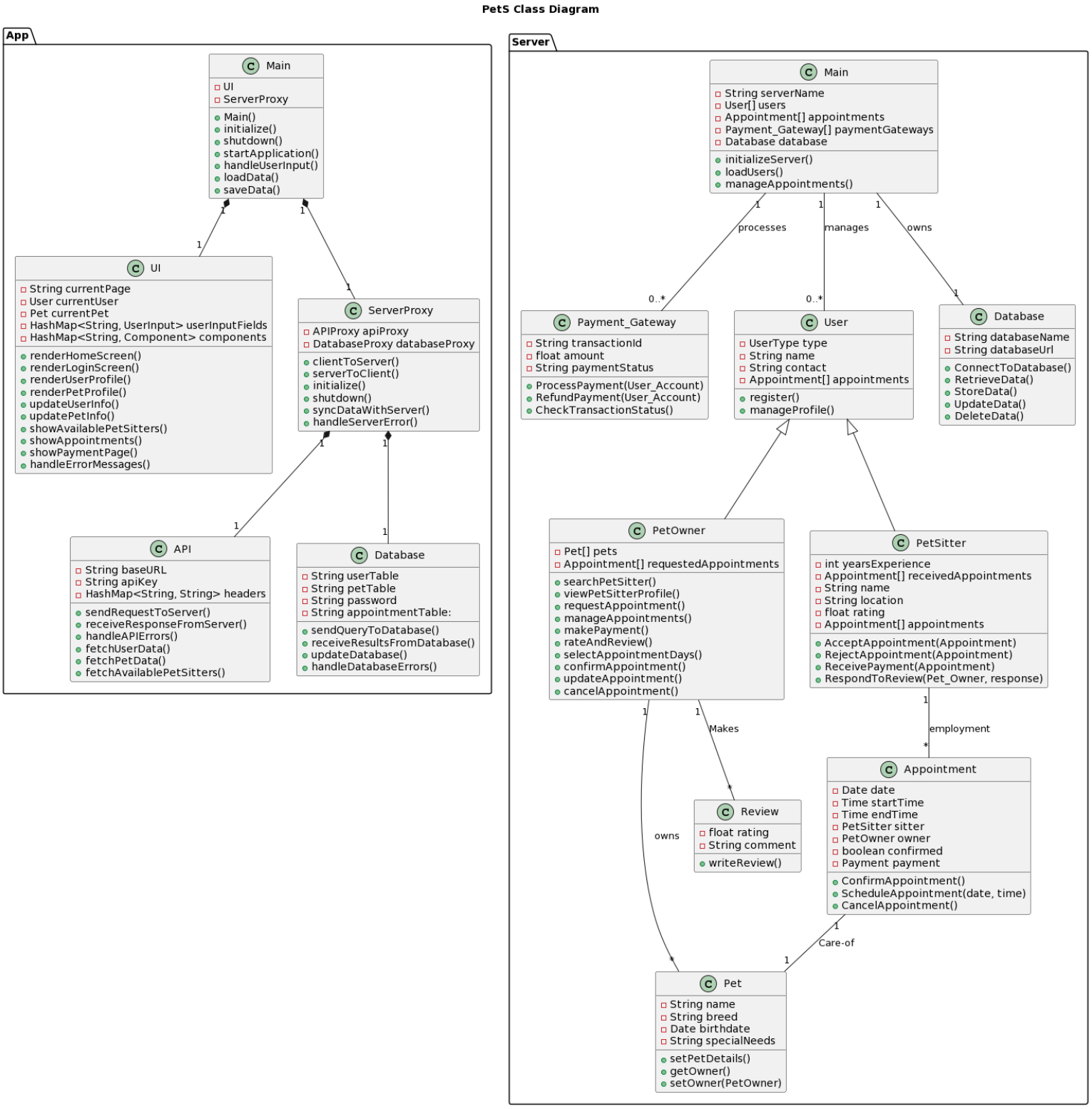
Description:

In this sequence diagram, the Pet Owner initiates the process by accessing appointment details. The UI then interacts with the App Core, which makes an API call to the API Server to retrieve appointment details from the Database. After the appointment details are displayed, the Pet Owner confirms the appointment.

Upon confirmation, the App Core communicates with the API Server to update the appointment status in the database. Once the update is confirmed, a confirmation message is displayed on the UI.

Finally, the Pet Owner asks for verification of the confirmation, which initiates a similar process to retrieve the confirmation status from the Database and show it on the UI.

4.5 Class Diagram



Repository file path:

- PNG: Documents/PlantUML/SystemClassDiagram.png
- PUML: Documents/PlantUML/SystemClassDiagram.puml

Description:

This is a comprehensive class diagram showing the architecture of a Pet Sitting Application System which is divided into two main packages: the App package, and the Server package, which provide an overview of the application's front-end and back-end architecture, respectively.

In the App package: (user-facing part of the system)

- The *Main* class is the entry point of the application. It is responsible for initializing, managing and shutting down the application's lifecycle, starting the application, handling user input, and loading and saving data.
- The *UI* class manages the user interface of the application. It renders different screens including home, login, user profile, pet profile and others, and also handles user data updates, error messages, and displays available pet sitters, appointments, and the payment page.
- The *ServerProxy* serves as the bridge between the application and the server. It sends data to the server and receives data back from the server, handles server errors and synchronizes data with the server.
- The *API* class is used to send requests to and receive responses from the server. It contains the base URL, the API key, and headers for requests. It provides methods for sending requests to the server, receiving responses, handling API errors, and fetching data.
- The *Database* class manages database operations. It contains the names of the user, pet, and appointment tables, and a password for database access. It provides methods for querying the database, receiving results, updating the database, and handling database errors.

In the Server package: (server-side part of the system)

- The *Main* class is responsible for initializing the server, loading user data, and managing appointments. It contains server names, users, appointments, payment gateways, and a database.
- The *User* class represents a user in the system, and the subclasses PetOwner and PetSitter inherit from this class, each with their own additional attributes and methods.
- The *Pet* class represents a pet owned by a PetOwner, it has attributes like name, breed, birthdate, and special needs, and methods to get and set pet details and owner.
- The *Appointment* class represents an appointment between a pet owner and a pet sitter. It contains details about the appointment such as the date, start and end time, whether it is confirmed, and the payment details.
- The *Review* class represents a review made by a PetOwner about a PetSitter. It contains a rating and a comment.
- The *Database* class manages the server-side database connection, updates, and handles data storage and retrieval.
- The Payment_Gateway class manages payment operations. It contains the transaction ID, amount, and payment status, and provides methods for processing payments, refunding payments, and checking transaction status.

The diagram also shows the relationships between the classes, indicating how they interact with each other in the system. For example, a PetOwner owns multiple Pets, makes multiple Reviews, and a PetSitter has multiple Appointments. This diagram gives a high-level overview of the architecture of the Pet Sitting Application System.