School of International Studies

Software Requirements Specification (SRS)

‘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	—	0.1
Irene/Alba/Paula	03-05-2023	Clean the contents and 1. Introduction, 2. Overall description (2.1, 2.2, 2.3) **[(...)** = TO DO**]**	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

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 Project 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 (...)

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

2. Overall Description

This section will cover general information about the project perspective, functions and various requirements and constraints.

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.

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

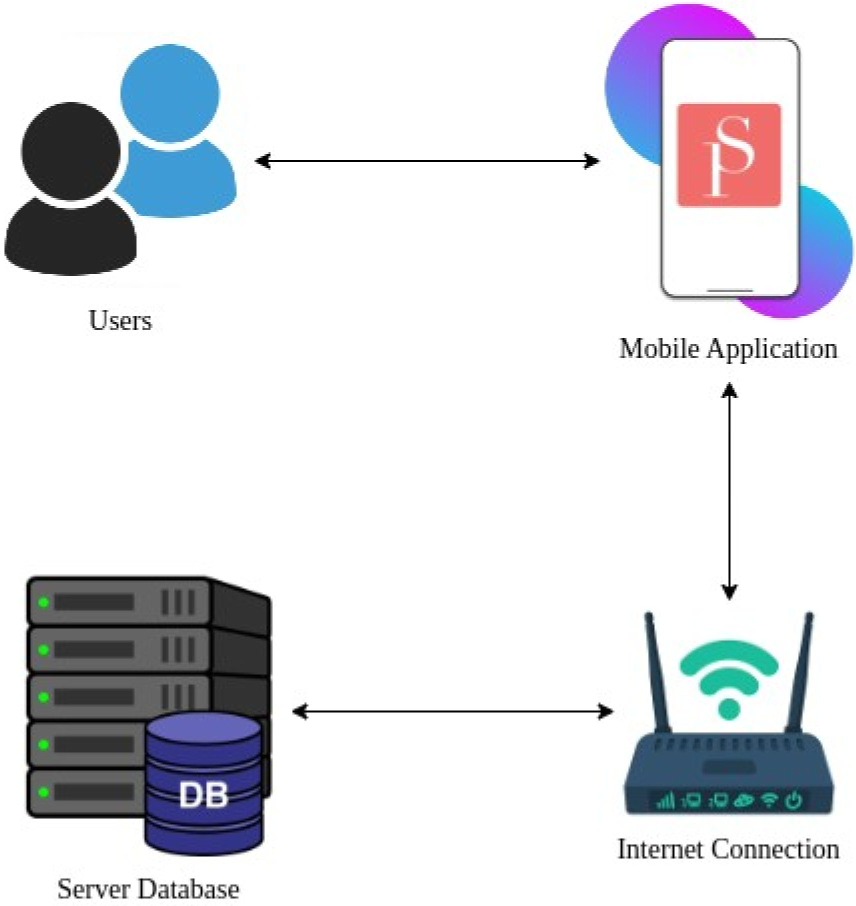

Fig 1. *Pictorial representation of the bigger system*

2.2 Product Features (...)

Profile creation with information about: Owner and pet/s (1 or more pets) and sitter

Short recommendations about the sitter process

Searcher for owner's offers or sitters in a previously specific zone range

In-app messages between both parts

Offers mailbox

Being able to pay the sitters from the app

Tasks register (with photos, hours done, ...)

Video call for emergencies

GPS tracking while walking the dogs

<Summarize the major features the product contains or the significant functions that it performs or lets the user perform. Details will be provided in Section 3, so only a high level summary is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or a class diagram, is often effective.>

User authentication and authorization system: The app allows users to sign up, log in, and manage their profiles, credentials, and preferences securely.

Pet owner dashboard: The app provides pet owners with a dashboard that displays their pet's information, appointment history, pending and upcoming appointments, and payment history.

Pet sitter dashboard: The app provides pet sitters with a dashboard that displays their schedule, appointment requests, accepted appointments, and payment history.

Pet sitter search functionality: The app enables pet owners to search for pet sitters based on their location, availability, and pet sitting experience, using filters and a map view.

Appointment request and management system: The app allows pet owners to request appointments with pet sitters, and pet sitters to accept or reject appointments based on their availability and preferences. The app also manages appointment reminders, cancellations, and rescheduling.

Messaging system: The app enables pet owners and pet sitters to communicate with each other through an in-app messaging system, to discuss appointments, pet care instructions, and other relevant information.

Payment gateway integration: The app integrates with a payment gateway service to enable secure and convenient payments between pet owners and pet sitters, and manage payment disputes and refunds.

2.3 User Classes and 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 Operating Environment (...)

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

2.5 Design and Implementation Constraints (...)

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

2.6 User Documentation (...)

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

2.7 Assumptions and Dependencies (...)

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

3. System Features (...)

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

3.1 Functional requirements

<Don't really say "System Feature 1." State the feature name in just a few words.>

3.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

3.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

3.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

User case list:

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 make informed decisions when selecting a suitable sitter for their pet.
UC-04	Request Appointment	Pet Owner	Allows pet owners to request an appointment with a selected pet sitter. Pet owners can specify the preferred date, time, duration, and any additional requirements or instructions for the pet sitting service. The request is sent to the chosen pet sitter for review and confirmation. This use case facilitates the booking process between pet owners and pet sitters.
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.

System requirements table:

Functional requirement	Non-functional requirement
UC-01 Identify user role as pet owner or pet sitter	UC-01 Performance: The registration process should be fast and responsive, allowing users to complete the registration without significant delays. Design: The registration form should be user-friendly and easy to understand, with clear instructions and input validation to assist users in providing accurate information. It should be securely stored and protected from unauthorized access, including appropriate security measures, such as password encryption to prevent fraudulent registrations. It should be capable of handling a large number of concurrent user registrations without compromising performance.
UC-02 Allow pet owners to search for pet sitters	UC-02 Performance: The search functionality should provide fast and efficient results, even when dealing with a large number of pet sitters in the system. Design: The search interface should be intuitive and easy to use, allowing pet owners to specify search criteria easily and providing accurate results based on the specified criteria, considering factors such as location, availability, and pet-specific requirements. The search system should be scalable to accommodate a growing number of pet sitters and ensure optimal performance even with increased data. The search functionality should consider the preferences and previous interactions of the pet owner, providing personalized recommendations for suitable pet sitters.
UC-03 Enable pet owners to view pet sitter details	UC-03 Performance: The profile page should load quickly, allowing pet owners to access the information without significant delays. Design: The profile layout should be clear and well-organized, presenting relevant information about the pet sitter in a user-friendly manner. The profile information should be available and up to date, reflecting the current status, experience, and service offerings of the pet sitter. The profile page should be responsive and adapt to different screen sizes and devices to ensure a consistent user experience.
UC-04 Enable pet owners to request appointments	UC-04 The appointment request system should be reliable, ensuring that requests are delivered

	to the pet sitter without loss or delay and it should validate input data to ensure that the provided information is accurate and complete before sending the request. The system should send notifications to the pet owner and the pet sitter to inform them about the received appointment request. It should log and store relevant information about the request, including the date, time, and additional requirements tracking the status of the appointment request, allowing pet owners to know whether the request is pending, accepted, or rejected.
UC-05 Allow pet sitters to accept/reject requests	UC-05 The interface for managing appointments should be intuitive and user-friendly, allowing pet sitters to view, accept, or reject requests efficiently and should handle a high volume of requests and provide fast response times to pet sitters. The system should be available and accessible to pet sitters at all times, enabling them to manage appointments as needed notifying them about new appointment requests, allowing them to respond promptly.
UC-06 Enable users to view and manage appointments	UC-06 The interface for managing appointments should be user-friendly and intuitive, allowing pet owners to view, modify, reschedule, or cancel appointments easily. The appointment management system should handle a large number of appointments efficiently, providing fast response times to pet owners being able to them at all times, allowing to manage their appointments according to their needs. Pet owners should receive timely notifications regarding appointment confirmations, changes, or cancellations, keeping them informed about the status of their appointments. It should synchronize appointment updates across multiple devices and platforms to ensure consistent information for pet owners.
UC-07 Facilitate communication between users	UC-07 The messaging functionality should support real-time messaging between pet owners and pet sitters, allowing for quick and efficient communication and supporting the exchange of multimedia content such as photos, videos, or documents to facilitate communication related to pet care. The messaging system should ensure the privacy and confidentiality of messages exchanged between pet owners and pet sitters, using encryption and secure communication protocols. Users should receive notifications for new messages or replies, enabling them to stay

	updated on the conversation without actively monitoring the app. 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.
UC-08 Allow pet owners to make secure payments	UC-08 The payment system should ensure secure transactions, protecting sensitive payment information and utilizing encryption and authentication mechanisms. The system should support multiple payment options, such as credit cards, debit cards, digital wallets, or other popular payment methods, to accommodate user preferences. The payment system should be reliable, minimizing the risk of transaction failures or errors and ensuring accurate processing of payments. The payment process should be fast and responsive, with minimal delays to provide a smooth and convenient payment experience for pet owners. The payment system should comply with relevant financial regulations and standards, ensuring the security and integrity of financial transactions.
UC-09 Enable pet owners to rate and review pet sitters	UC-09 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. The rating and review system should include moderation mechanisms to prevent inappropriate or offensive content from being published. The system should provide an accessible interface for pet owners to provide ratings and write reviews, following accessibility guidelines. The rating and review system should foster trust among users by displaying authentic and reliable feedback, helping pet owners make informed decisions when selecting a pet sitter.
UC-10 Allow users to manage their profiles	UC-10 The user profile management interface should be user-friendly, allowing pet owners and pet sitters to easily update and maintain their profile information. 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. The profile management functionality should be fast and responsive, enabling users to update their profile information without significant delays. Privacy: The system
UC-11 Allow pet owners to select appointment days in a calendar view	UC-11 The calendar view for selecting appointment days should be intuitive and user-friendly, allowing pet owners to navigate, select, and mark their availability easily. The calendar functionality should provide a smooth and responsive user experience, even when dealing with a large number of selectable dates. The calendar view should support various date selection options, including single dates, date ranges, or recurring appointments, to accommodate different scheduling needs of pet owners. The calendar view should support different date formats and regional settings to cater to users from various locations. The calendar view should be compatible with different devices and screen sizes, ensuring a consistent experience across platforms.
UC-12 Allow pet owners to confirm selected appointment days	UC-12 The confirmation process should accurately display the selected appointment details to pet owners, allowing them to review and verify the information before finalizing the booking. The system should provide immediate acknowledgment to pet owners upon successfully confirming the appointment, ensuring that both parties are aware of the scheduled pet sitting arrangement. The system should send a confirmation notification to the pet sitter, informing them about the confirmed appointment and relevant details. The confirmation process should be reliable, minimizing the risk of errors or inconsistencies in the booking information. The confirmation process should provide information about the cancellation policy, ensuring pet owners are aware of any applicable fees or restrictions related to cancellations.

UC-13 Enable pet owners to update selected appointment days	UC-13 The appointment update functionality should allow pet owners to modify selected appointment days, time, or other details easily. The system should provide a fast and responsive experience when updating appointments, ensuring that changes are reflected promptly. The appointment update process should validate the modified information to ensure accuracy and prevent conflicts or errors. The system should send notifications to the pet sitter about the updated appointment, keeping them informed about any changes made by the pet owner. The system should maintain a revision history or log of appointment updates, allowing pet owners and pet sitters to track and review previous changes if necessary.
UC-14 Allow pet owners to cancel a previously booked appointment	UC-14 The cancellation process should be straightforward and accessible, allowing pet owners to cancel appointments without undue complications. 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. The cancellation process should be timely, notifying the pet sitter as soon as possible to allow them to adjust their schedule or make alternative arrangements. If applicable, the system should handle refund processes promptly and accurately, ensuring that pet owners receive appropriate refunds based on the cancellation policy. The cancellation process should facilitate communication between pet owners and pet sitters, allowing them to discuss and resolve any issues related to the cancellation.

3.2 System Feature 2 (and so on)

4. External Interface Requirements (...)

4.1 User Interfaces

- Pet Owner Interface: The app shall provide a user-friendly interface for pet owners to create profiles, search for pet sitters, request services, schedule appointments, communicate with pet sitters, and make payments.

- Pet Sitter Interface: The app shall provide a user-friendly interface for pet sitters to create profiles, search for pet owners, accept or decline service requests, manage appointments, communicate with pet owners, and receive payments.
- Admin Interface: The app may provide an administrative interface for app administrators to manage user accounts, verify pet sitters, handle payment disputes, and monitor overall app activity.

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

4.2 Hardware Interfaces

Mobile Device: The app shall interface with the mobile device's hardware components, such as camera (for uploading pet photos), GPS (for location-based services), and push notifications (for alerts and notifications).

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

4.3 Software Interfaces

- Payment Gateway: The app shall interface with a third-party payment gateway for secure payment processing and transaction management.
- Mapping Service: The app may interface with a mapping service (e.g., Google Maps) for location-based services, such as searching for pet sitters or displaying pet sitters' locations on a map.
- Messaging Service: The app may interface with a messaging service (e.g., Twilio, Firebase Cloud Messaging) for real-time communication between pet owners and pet sitters.

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

4.4 Communications Interfaces

- Internet Connectivity: The app shall require internet connectivity for various functionalities, such as searching for pet sitters, submitting service requests, and processing payments.
- Email/Phone Notifications: The app may send email or phone notifications to users for important updates, such as appointment confirmations, payment receipts, or service request status changes.

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

5. Other Nonfunctional Requirements (...)

5.1 Performance Requirements

- The app shall have fast response times and load times, ensuring smooth and efficient user experience.
- The app shall be able to handle a large number of concurrent users and bookings without performance degradation.

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

5.2 Safety Requirements

-

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

5.3 Security Requirements

- The app shall implement industry-standard security measures to protect user information, payment data, and other sensitive information.
- The app shall use encryption and authentication mechanisms to secure user data and communications.
- The app shall comply with relevant data privacy and security regulations, such as GDPR, CCPA, and others.

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

5.4 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

6. Other Requirements (...)

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary (...)

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models (...)

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: Issues List (...)

< This is a dynamic list of the open requirements issues that remain to be resolved, including TBDs, pending decisions, information that is needed, conflicts awaiting resolution, and the like.>