ESCI upf.
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

Table of Contents

1. Introduction
- 1.1 Purpose
- 1.2 Intended Audience
- 1.3 Project Scope
- 1.4 References **(...)**

2. Overall Description
- 2.1 Product Perspective **(...)**
- 2.2 Product Features **(...)**
- 2.3 User Classes and Characteristics **(...)**
- 2.4 Operating Environment **(...)**
- 2.5 Design and Implementation Constraints **(...)**
- 2.6 User Documentation **(...)**
- 2.7 Assumptions and Dependencies **(...)**

3. System Features (...)
- 3.1 System Feature 1
- 3.2 System Feature 2 (and so on)

4. External Interface Requirements (...)
- 4.1 User Interfaces
- 4.2 Hardware Interfaces
- 4.3 Software Interfaces
- 4.4 Communications Interfaces

5. Other Nonfunctional Requirements (...)
- 5.1 Performance Requirements
- 5.2 Safety Requirements
- 5.3 Security Requirements
- 5.4 Software Quality Attributes

6. Other Requirements (...)

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.>

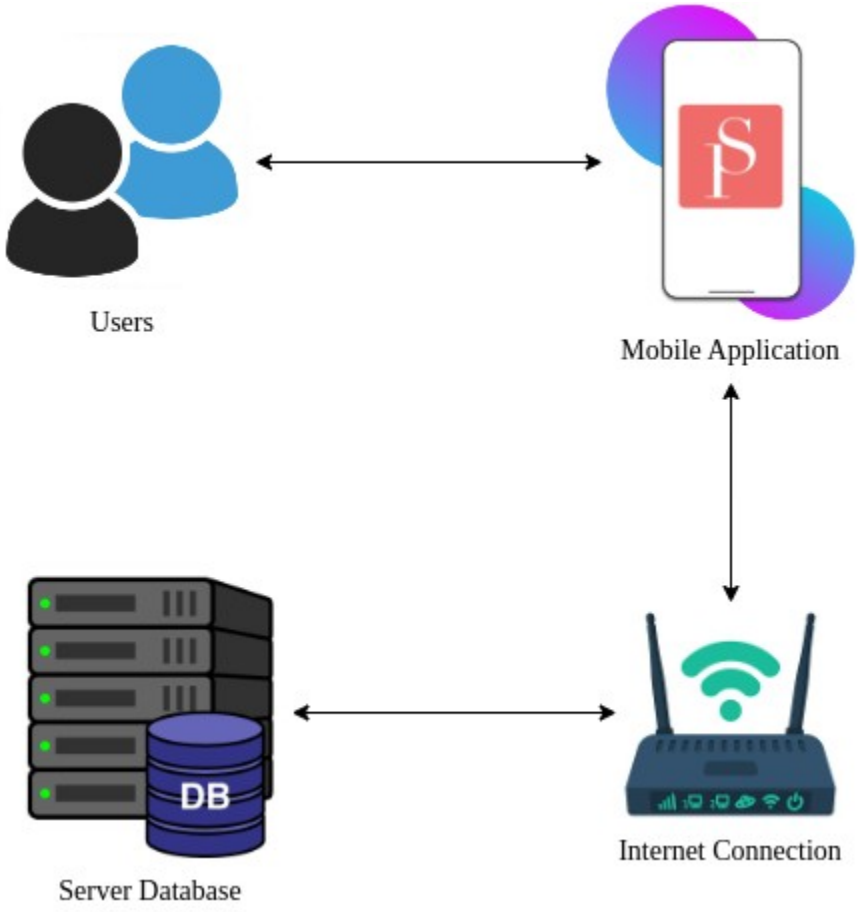

Fig 1. *Pictorial representation of the bigger system*

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 System Feature 1

<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 Registration:

- The app shall allow pet owners and pet sitters to register and create user accounts.
- The app shall collect and store user information, including name, contact information, pet details, service preferences, and other relevant information.

Search and Booking:

- The app shall allow pet owners to search for pet sitters based on location, availability, services required, pet type, and other search criteria.
- The app shall display a list of pet sitters that match the search criteria, including their profiles, ratings, reviews, and availability.
- The app shall allow pet owners to view and select pet sitters, send booking requests with date, time, duration, and other requirements.
- The app shall send booking confirmation to pet owners and pet sitters, including booking details, payment confirmation, and other relevant information.

Messaging and Communication:

- The app shall facilitate real-time messaging and communication between pet owners and pet sitters within the app.
- The app shall send notifications for new messages and updates.
- The app shall allow users to manage their messages, view chat logs, and communicate securely.

Secure Payment Integration:

- The app shall integrate a secure payment gateway to enable pet owners to make payments for bookings, subscriptions, premium services, or other transactions.
- The app shall store payment information securely and handle payment transactions in compliance with industry standards and regulations.

Ratings and Reviews:

- The app shall allow pet owners to provide ratings and reviews for pet sitters and their services.
- The app shall display ratings and reviews of pet sitters, helping pet owners make informed decisions.
- The app shall allow pet sitters to view and respond to ratings and reviews.

User Profiles:

- The app shall allow pet owners and pet sitters to create and manage their profiles.
- The app shall display user profiles, including information, photos, services offered, ratings, and reviews.

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.>