

**CEBU INSTITUTE OF TECHNOLOGY**

**UNIVERSITY**

COLLEGE OF COMPUTER STUDIES

Software Requirements Specifications

for

**Pawlly**

Submitted by:

Baltazar, Rigel L.

Carbungco, Louie James F.

Tagarao, Keith Ruezyl

Submitted to:

Revilleza, Frederick

Change History

Table of Contents

Change History 2

Table of Contents 3

1. Introduction 5

1.1. Purpose 5

1.2. Scope 5

1.3. Definitions, Acronyms and Abbreviations 5

1.4. References 6

2. Overall Description 6

2.1. Product perspective 7

2.2. User characteristics 8

2.4. Constraints 8

2.5. Assumptions and dependencies 8

3. Specific Requirements 10

3.1. External interface requirements 10

3.1.1. Hardware interfaces 10

3.1.2. Software interfaces 10

3.1.3. Communications interfaces 10

3.2. Functional requirements 10

Module 1 10

Module 2 10

3.4 Non-functional requirements **11**

Performance 11

Security 11

Reliability 11

# Introduction

**Pawlly** is a community-driven platform that brings together animal lovers and provides a space for them to connect and support one another. The platform allows users to share information about pets available for adoption, report lost or found animals, and search for reports in their area through utilizing Geographic Information Systems (GIS) and Global Positioning Systems (GPS). Additionally, Pawlly has implemented an admin-reviewed application system ensuring a smooth and secure adoption process . By facilitating connections between people who care about animals, Pawlly aims to make a positive impact on the lives of pets and their human companions.

## Purpose

* Facilitate animal welfare efforts by connecting users to rehome pets, report lost animals, and help find missing pets.
* Promote collaboration among pet lovers to support the well-being of animals in their community.

## Scope

* DBMS for users and animals posted (rehome, adaption and lost and found)
* Report Management System (Generation, Status tracking)
* Adaption Request Management
* Geolocation Searching
* Identify the software product(s) to be produced by name (e.g., Host DBMS, Report Generator, etc.);
* Explain what the software product(s) will, and, if necessary, will not do;
* Describe the application of the software being specified, including relevant benefits, objectives, and goals;
* Be consistent with similar statements in higher-level specifications (e.g., the system requirements specification), if they exist.

## Definitions, Acronyms and Abbreviations

**API (Application Programming Interface)** – A set of rules and tools for building software applications, enabling communication between different systems or components.

**Database Management System (DBMS)** – A software that allows users to create, store, manage, and retrieve data efficiently. It provides tools for defining data structures, enforcing security, ensuring data integrity, and supporting concurrent access by multiple users.

**Geographic Information System (GIS)** – A system for capturing, storing, analyzing, and managing spatial or geographic data. It helps visualize relationships, patterns, and trends in geographic contexts.

**Global Positioning System (GPS)** – A satellite-based navigation system that provides location, velocity, and time data to users anywhere on Earth with a clear line of sight to multiple satellites.

**Hypertext Transfer Protocol Secure (HTTPS)** – A secure version of HTTP that encrypts data exchanged between a web browser and a website using SSL/TLS, ensuring privacy and security.

**iPhone Operating System (iOS)** – Apple's mobile operating system designed for iPhones, iPads, and iPods. It provides a secure and user-friendly interface, supports App Store applications, and integrates seamlessly with Apple's ecosystem.

**MTBF (Mean Time Between Failures)** – The average time between system or component failures, used to measure reliability and predict maintenance needs.

**MTTR (Mean Time to Repair)** – The average time required to diagnose, fix, and restore a failed system or component to full functionality.

**SRS (Software Requirements Specification)** – A document that describes the system's intended functionality, performance, and design constraints in detail.

**SQL (Structured Query Language)** – A standardized programming language used for managing and manipulating relational databases, allowing users to query, insert, update, and delete data.

**XSS (Cross-Site Scripting)** – A web security vulnerability where attackers inject malicious scripts into web pages, potentially compromising user data and site integrity.

## References

*[1]* ***Petfinder****, "Petfinder – Adopt a pet," [Online]. Available:* [*https://www.petfinder.com*](https://www.petfinder.com)*. [Accessed: 31-Jan-2025].*

*[2]* ***PawBoost****, "Reuniting lost pets with their families," [Online]. Available:* [*https://www.pawboost.com*](https://www.pawboost.com)*. [Accessed: 31-Jan-2025].*

*[3]* ***Nextdoor****, "Nextdoor – Lost and found pets," [Online]. Available:* [*https://nextdoor.com*](https://nextdoor.com)*. [Accessed: 31-Jan-2025].*

*[4]* ***Adopt-a-Pet****, "Find adoptable pets," [Online]. Available:* [*https://www.adoptapet.com*](https://www.adoptapet.com)*. [Accessed: 31-Jan-2025].*

*[5]* ***LostMyDoggie****, "Lost My Doggie – Pet recovery service," [Online]. Available:* [*https://www.lostmydoggie.com*](https://www.lostmydoggie.com)*. [Accessed: 31-Jan-2025].*

*[6]* ***Facebook****, "Local lost and found pets groups," [Online]. Available:* [*https://www.facebook.com*](https://www.facebook.com)*. [Accessed: 31-Jan-2025].*

*[7]* ***HomeAgain****, "HomeAgain Pet Recovery Service," [Online]. Available:* [*https://www.homeagain.com*](https://www.homeagain.com)*. [Accessed: 31-Jan-2025].*

# Overall Description

## Product perspective

Pawlly is a web and mobile-based application designed to streamline animal welfare services by providing a platform for adoption, lost pet reporting, and community-driven pet recovery efforts. It functions as an independent system with dedicated modules for pet listing, geo-location searching, adoption management, and user interactions.

**System Context**

Pawlly interacts with users through its web and mobile applications while integrating with third-party mapping services for geo-location functionalities. The system does not depend on external hardware but relies on cloud-based databases for data storage and retrieval.

**Modular Decomposition**

##### Module 1: User Management

* Transaction 1.1: Account Registration
* Transaction 1.2: User Profile Management

##### Module 2: Pet Management

* Transaction 2.1: Pet Listing (Rehome, Adoption, Lost & Found)
* Transaction 2.2: Geo-location Searching
* Transaction 2.3: Pet Locate System (Lost Pet Reports & Responses)

##### Module 3: Adoption Process

* Transaction 3.1: Adoption Requests
* Transaction 3.2: Adoption Auditing

##### Module 4: Notification System

* Transaction 4.1: Adoption Updates
* Transaction 4.2: Lost and Found Notifications

## User characteristics

*Pawlly caters to different types of users, each with specific roles and privileges:*

1. ***General Users***
   * *Register and manage their profiles*
   * *Browse pet listings and lost pet reports*
   * *Submit adoption applications*
   * *Report lost pets*
   * *Receive notifications on adoption and lost pet reports*
2. ***Pet Owners & Adopters***
   * *Post pets for rehoming*
   * *Submit adoption requests*
   * *Communicate with potential adopters or pet finders*
3. ***Administrators***
   * *Review and approve adoption requests*
   * *Moderate pet listings and lost pet reports*
   * *Manage user accounts*

## 2.4. Constraints

*The development and deployment of Pawlly are subject to the following constraints:*

* ***Regulatory Policies****: Compliance with animal welfare regulations and data protection laws.*
* ***Hardware Limitations****: The platform is optimized for web and mobile devices but does not support offline functionality.*
* ***Interfaces to Other Applications****: Integration with Google Maps API for geo-location services.*
* ***Parallel Operation****: The system supports concurrent users but may experience performance limitations under heavy traffic.*
* ***Audit Functions****: Adoption requests undergo admin review to ensure ethical rehoming.*
* ***Control Functions****: Admin users have moderation control over listings and reports.*
* ***Reliability Requirements****: The system must ensure uptime and data accuracy for user engagement.*
* ***Criticality of the Application****: Essential for pet recovery and adoption services but not life-critical.*
* ***Safety and Security Considerations****: Secure authentication and encrypted data storage to protect user information.*

## 2.5. Assumptions and dependencies

* The platform assumes users have internet access and compatible devices.
* Cloud database services will be available for data storage and retrieval.
* Users will accurately input information for listings and reports.
* The system will rely on external mapping services for geo-location functionalities.
* The mobile application will be compatible with Android and iOS platforms.
* Administrators will actively review and moderate pet adoption applications to prevent misuse.
* Notification services depend on third-party push notification providers.

# Specific Requirements

## External interface requirements

### Hardware interfaces

Pawlly will be accessible on the following devices:

* **Web Application:**  Desktop or laptop computers with modern web browsers.
* **Mobile Application**: Smartphones running Android (8.0 and above) or iOS (12.0 and above).

### 3.1.2. Software interfaces

The application includes:

* **Web Application**: Built for accessibility via browsers like Chrome, Firefox, and Safari.
* **Mobile Application**: Developed for Android and iOS.
* **Database**: Stores user profiles, pet listings, and adoption/lost pet reports.
* **Authentication**: Handles secure user logins.

### 3.1.3. Communications interfaces

Pawlly communicates using the following APIs:

* **Google Maps API**: For geolocation and location-based searches.
* **Firebase**: For notifications and authentication. - Subject-to-change
* **Spring Boot API**: Enables interaction between the frontend and backend.

## Functional requirements

### Web and Mobile Functions

1. Account Registration – Users must create an account to access the platform/
2. Pet Listing – Users can browse and view posted listings by others.
3. Geo-location Searching – Users can search for reports or listings specified to their location.
4. Pet Locate System – Users may generate a ‘lost pet’ report. Users may respond to lost pet reports by viewing listings and communicate with the owner off-app.
5. Adoption Requests – Users may respond to rehome listings by submitting an application form, which will then be reviewed by an admin.
6. Adoption Auditing – Users may respond to rehome listings by submitting an application form, which will then be reviewed by an admin.
7. Notification System – Users will receive new updates on new adaption listings, nearby lost and found listings, or adaption application status.
8. User Profile Management – Users can modify/delete their account.

## Non-functional requirements

### Performance

* The system should respond to user input within 2 seconds
* The system should be able to handle a minimum of 100 concurrent users
* The system should be able to process a minimum of 10 requests per second
* The system should have a maximum page load time of 4 seconds

### Security

* The system should ensure confidentiality, integrity, and availability of user data
* The system should comply with relevant security standards and regulations
* The system should protect against common web attacks such as SQL injection and cross-site scripting (XSS)
* The system should use secure communication protocols such as HTTPS

### Reliability

* The system should be available 98% of the time
* The system should be able to recover from failures within 24 hours
* The system should have a mean time between failures (MTBF) of at least 30 days
* The system should have a mean time to repair (MTTR) of less than 4 hours