|  |
| --- |
|  |
|  |
| SOFTWARE REQUIREMENTS SPECIFICATION (SRS) |
| FOR |
| Aerogotchi |

Team Name: OoeyGUI

Class: 490

Instructor: Professor Dantes

Revision History

| Revision Letter | By | Change Description | Date |
| --- | --- | --- | --- |
| - | OoeyGUI | Initial Revision | 09  October, 2023 |
| A | Keith Chua, Jarrett McIntire | Removed Headers, Footers. Added Team Name, Logo, and Header. Fixed Rev History and Titles. Removed constraints that did not apply from Functional Requirement and Qualification Provision | 19 November, 2023 |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

Page

1. INTRODUCTION 1

1.1 Scope 1

1.2 Product Value 1

1.3 Intended Audience 1

1.4 Intended Use 1

2. FUNCTIONAL REQUIREMENTS 2

3. EXTERNAL INTERFACE REQUIREMENTS 4

3.1 User Interface Requirements 4

3.2 Hardware Interface Requirements 4

3.3 Software Interface Requirements 5

4. NON FUNCTIONAL REQUIREMENTS 5

4.1 Security 5

4.2 Capacity 5

4.3 Compatibility 5

4.4 Reliability 5

4.5 Scalability 6

4.6 Usability 6

4.7 Other 6

5. QUALIFICATION PROVISIONS 7

6. NOTES 11

6.1 Acronyms and Abbreviations 11

# INTRODUCTION

## Scope

This Software Requirements Specification (SRS) defines the requirements for the Aerogotchi Project Software for COMP 490 developed by OoeyGUI.

The scope of this Software Requirements Specification document is to list the requirements of Aerogotchi, an autonomous pet-like drone developed by the coding wizards at OoeyGUI for California State University of Northridge. These requirements include everything encompassing the design and function of Aerogotchi. This document also lists the verification methods of each requirement.

## Product Value

Aerogotchi is a drone companion that will interact with the user providing a delightful experience, without the usual responsibilities of feeding, cleaning up after, and walking them.

## Intended Audience

The intended audience are people and children who want the experience of interacting with a pet without the commitment level and permanence of adding a pet to their family. Aerogotchi is also a fun way for kids to interact and learn about caregiving and technology at the same time.

## Intended Use

The Aerogotchi is a drone companion that is intended to create long lasting bonds with the user and to provide entertainment in their life. Users are meant to take care of their pet as well as engage in fun activities through the mobile app that will further enlighten their experience.

# 2. FUNCTIONAL REQUIREMENTS

Design Requirements (FUNC\_SRS\_001):

* FUNC\_SRS\_001.1 - The drone shall be autonomous, but will have the option to be controlled manually using a physical controller and a smartphone application.
* FUNC\_SRS\_001.2 - The drone shall use image recognition to enable the drone pet to identify and interact with specific objects or patterns in its environment.
* FUNC\_SRS\_001.3 - The drone shall display interactive behaviors, including responding to user commands.
* FUNC\_SRS\_001.4 - The drone shall display emotional expressions and engage in playful activities.
* FUNC\_SRS\_001.5 - The application shall allow the user to manually control the drone using a controller and have access to the game that is included for entertainment.
* FUNC\_SRS\_001.6 - The game shall present periodic in-game challenges to maintain user engagement.

System Requirements (FUNC\_SRS\_002):

* FUNC\_SRS\_002.1 - The software minimum requirements shall be Windows 10, macOS 10.15, Android 8.0, or iOS 12.0 (or equivalent versions).
* FUNC\_SRS\_002.2- The software shall include permission access to the device's camera, microphone, and location services for drone control and image recognition.
* FUNC\_SRS\_002.3 - The software shall include mouse and keyboard (for desktop) or touchscreen (for mobile), as well as game controller support.
* FUNC\_SRS\_002.4 -The application shall request and handle Bluetooth permissions when needed for pairing and connecting with the drone.

Constraints (FUNC\_SRS\_003)

* FUNC\_SRS\_003.1 - The drone shall not be operated in extreme weather conditions.
* FUNC\_SRS\_003.2 - The drone shall have safety mechanics such as collision avoidance, distance to prevent accidents and safety concerns.
* FUNC\_SRS\_003.3 -The application shall assume the user has access to a drone.
* FUNC\_SRS\_003.4 - The drone shall be programmed and developed in Python using ArduPilot, MediaPipe, Tello SDK.
* FUNC\_SRS\_003.5 - The application shall be designed and developed in Flutter.
* FUNC\_SRS\_003.6 - The application design shall run on any mobile device > 2018.
* FUNC\_SRS\_003.7 - The application design shall assume the user will have enough storage to install.
* FUNC\_SRS\_003.8 - The application shall store user data within its own database.

## 3.1 User Interface Requirements

In game icons shall be positioned around the borders of the screen to provide room for main visuals

Character Status Elements simple to discern in specified sections (Status Menu and/or Noise Cues)

Visual (drone motion and/or In-game emote) and Audio (Drone speaker or mobile device speaker) Feedback for Game Actions (feeding, playing, teaching, etc.)

* UIR\_SRS\_001 - The game shall have a Pet View where you can see the pet character with several Menu Icons surrounding the border to provide room for the main visuals
* UIR\_SRS\_001.1 - Pet View shall have 8-bit visuals of drone pet.
* UIR\_SRS\_001.2 - Drone Pet visual shall have motions based on game status and mechanics.
* UIR\_SRS\_002 - The Settings Menu shall be accessible from the Pet View screen.
* UIR\_SRS\_002.1 - Settings Menu shall have a “Go to Title” option that allows the User to
* UIR\_SRS\_003 - The Drone Control Menu shall be accessible from the Pet View screen.
* UIR\_SRS\_003.1 - Drone Control Menu shall connect to the drone camera for viewing and flying purposes.
* UIR\_SRS\_003.2 - Drone Control Menu shall allow the user to fly the drone with a controller directly connected to the mobile device.
* URI\_SRS\_003.3 - Drone Control Menu shall have a button titled “Go Home” that tells the drone to fly to its “Home” spot.
* UIR\_SRS\_004 - The Status Menu shall be accessible from the Pet View screen.
* UIR\_SRS\_004.1 - Status Menu shall have an Energy Bar to display the drone pet’s “energy” or actual battery percentage.
* UIR\_SRS\_004.2 - Status Menu shall have a Food Bar to display the drone pet’s “hunger.”
* UIR\_SRS\_004.3 - Status Menu shall have a Fun Bar to display the drone pet’s level of “entertainment”.
* UIR\_SRS\_005 - The Feeding Menu shall be accessible from the Pet View screen.
* UIR\_SRS\_005.1 - Feeding Menu shall have options of different Food to serve the drone pet depending on Drone Personality and other game factors.
* UIR\_SRS\_006 - The Playing Menu shall be accessible from the screen.
* UIR\_SRS\_006.1 - Playing Menu shall have access to different predefined game modes that the User and drone pet can interact with.

## 3.2 Hardware Interface Requirements

* HIR\_SRS\_001 - The application shall run on iOS and Android mobile devices.
* HIR\_SRS\_002 - The application shall require Internet connection to download software.
* HIR\_SRS\_003 - The application shall require Internet connection to communicate with the drone, preferably 4G/5G

## 3.3 Software Interface Requirements

* SIR\_SRS\_001 - The Aerogotchi mobile application will be used to control the drone including directional movement and interaction.

# 4. NON FUNCTIONAL REQUIREMENTS

## 4.1 Security

* SEC\_SRS\_001 - The application shall require data backups on a frequent basis to allow up to date data in case of an issue.
* SEC\_SRS\_002 - The drone shall have data security features where only the members of Aerogotchi have access to the data.
* SEC\_SRS\_003 - The drone shall have flight safety mechanisms such as collision avoidance mechanism and no flights in no fly zone.
* SEC\_SRS\_004 - The drone shall be in compliance with regulations since this is essential to safety, security and legality.

## 4.2 Capacity

* CAP\_SRS\_001 - The mobile application shall require 30 MB.

## 4.3 Compatibility

* COMP\_SRS\_001 - The application shall be compatible with at least android 9 (2018) and onwards for android based devices(Phones and Tablets).
* COMP\_SRS\_002 - The application shall be compatible with at least iOS 12 (2018) and onwards for apple based devices(Phones and Tablets).

## 4.4 Reliability

* RELI\_SRS\_001 - Aerogotchi shall function as intended until the battery life of the drone runs out.

## 4.5 Scalability

* SCAL\_SRS\_001 - N/A

## 4.6 Usability

* USA\_SRS\_001 - The initial setup of the drone pet through the app will be straightforward, with clear instructions and minimal technical knowledge required.
* USA\_SRS\_002 - The application shall provide access to instructional videos for basic drone skills.

## 4.7 Other

* N/A

# 5. QUALIFICATION PROVISIONS

Table I. Requirements Verification

| SRS Req. ID | Paragraph Title | Verification Method |
| --- | --- | --- |
| FUNC\_SRS\_001.1 | The drone shall be autonomous, but will have the option to be controlled manually using a physical controller and a smartphone application. | Demonstration |
| FUNC\_SRS\_001.2 | The drone shall use image recognition to enable the drone pet to identify and interact with specific objects or patterns in its environment. | Demonstration |
| FUNC\_SRS\_001.3 | The drone shall display interactive behaviors, including responding to user commands. | Demonstration |
| FUNC\_SRS\_001.4 | The drone shall display emotional expressions and engage in playful activities | Test |
| FUNC\_SRS\_001.5 | The application shall allow the user to manually control the drone using a controller and have access to the game that is included for entertainment. | Test |
| FUNC\_SRS\_001.6 | The game shall present periodic in-game challenges to maintain user engagement. | Demonstration |
| FUNC\_SRS\_002.1 | The software minimum requirements shall be Windows 10, macOS 10.15, Android 8.0, or iOS 12.0 (or equivalent versions). | Inspection |
| FUNC\_SRS\_002.2 | The software shall include permission access to the device's camera, microphone, and location services for drone control and image recognition. | Inspection |
| FUNC\_SRS\_002.3 | The software shall include mouse and keyboard (for desktop) or touchscreen (for mobile), as well as game controller support. | Inspection |
| FUNC\_SRS\_002.4 | The application shall request and handle Bluetooth permissions when needed for pairing and connecting with the drone. | Inspection |
| FUNC\_SRS\_003.1 | The drone shall not be operated in extreme weather conditions. | Analysis |
| FUNC\_SRS\_003.2 | The drone shall have safety mechanics such as collision avoidance, distance to prevent accidents and safety concerns. | Demonstration |
| FUNC\_SRS\_003.3 | The application shall only work if the user has access to the drone. | Test |
| FUNC\_SRS\_003.4 | The drone shall be programmed and developed in Python using ArduPilot, MediaPipe, Tello SDK. | Inspection |
| FUNC\_SRS\_003.5 | The application shall be designed and developed in Flutter. | Inspection |
| FUNC\_SRS\_003.6 | The application design shall run on any mobile device > 2018. | Analysis |
| FUNC\_SRS\_003.7 | The application design shall assume the user will have enough storage to install. | Analysis |
| FUNC\_SRS\_003.8 | The application shall store user data within its own database. | Analysis |
| UIR\_SRS\_001 | The game shall have a Pet View where you can see the pet character with several Menu Icons surrounding the border to provide room for the main visuals | Demonstration |
| UIR\_SRS\_001.1 | Pet View shall have an 8-bit visual of a drone pet. | Demonstration |
| UIR\_SRS\_001.2 | Drone Pet shall have features based on game status and mechanics. | Demonstration |
| UIR\_SRS\_002 | The Settings Menu shall be accessible from the Pet View screen. | Demonstration |
| UIR\_SRS\_002.1 | Settings Menu shall have a “Go back” option that allows the User to return to Pet View. | Demonstration |
| UIR\_SRS\_003 | The Drone Control Menu shall be accessible from the Pet View screen. | Demonstration |
| UIR\_SRS\_003.1 | Drone Control Menu shall connect to the drone camera for viewing and flying purposes. | Analysis |
| UIR\_SRS\_003.2 | Drone Control Menu shall allow the user to fly the drone with a controller directly connected to the mobile device. | Analysis |
| UIR\_SRS\_003.3 | Drone Control Menu shall have a button titled “Go Home” that tells the drone to fly to its “Home” hub. | Analysis |
| UIR\_SRS\_004 | The Status Menu shall be accessible from the Pet View screen. | Demonstration |
| UIR\_SRS\_004.1 | Status Menu shall have an Energy Bar to display the drone pet’s “energy” or actual battery percentage. | Demonstration |
| UIR\_SRS\_004.2 | Status Menu shall have a Food Bar to display the drone pet’s “hunger” (or replaced with something else) | Demonstration |
| UIR\_SRS\_004.3 | Status Menu shall have a Fun Bar to display the drone pet’s level of “entertainment”. | Demonstration |
| UIR\_SRS\_005 | The Feeding Menu shall be accessible from the Pet View screen. | Demonstration |
| UIR\_SRS\_005.1 | Feeding Menu shall have options of different Food to serve the drone pet depending on Drone Personality and other game factors. | Demonstration |
| UIR\_SRS\_006 | The Playing Menu shall be accessible from the screen. | Demonstration |
| UIR\_SRS\_006.1 | Playing Menu shall have access to different predefined game modes that the User and drone pet can interact with. | Demonstration |
| HIR\_SRS\_001 | The application shall run on iOS and Android mobile devices. | Test |
| HIR\_SRS\_002 | The application shall require Internet connection to download software | Inspection |
| HIR\_SRS\_003 | The application shall require Internet connection to communicate with the drone, preferably 4G/5G | Inspection |
| SIR\_SRS\_001 | The Aerogotchi mobile application shall control the drone including directional movement and interaction. | Demonstration |
| SEC\_SRS\_001 | The application shall require data backups on a frequent basis to allow up to date data in case of an issue. | Inspection |
| SEC\_SRS\_002 | The drone shall have data security features where only the members of Aerogotchi have access to the data. | Demonstration |
| SEC\_SRS\_003 | The drone shall have flight safety mechanisms such as collision avoidance mechanism and no flights in no fly zone. | Demonstration |
| SEC\_SRS\_004 | The drone shall be in compliance with regulations since this is essential to safety, security and legality. | Inspection |
| CAP\_SRS\_001 | The mobile application shall require 30 MB | Inspection |
| COMP\_SRS\_001 | The application shall be compatible with at least android 9 (2018) and onwards for android based devices(Phones and Tablets) | Inspection |
| COMP\_SRS\_002 | The application shall be compatible with at least iOS 12 (2018) and onwards for apple based devices(Phones and Tablets). | Inspection |
| RELI\_SRS\_001 | Aerogotchi shall function as intended until the battery life of the drone runs out. | Test |
| USA\_SRS\_001 | The application setup shall be straightforward, with clear instructions and minimal technical knowledge required. | Demonstration |
| USA\_SRS\_002 | The application shall provide access to instructional videos for basic drone skills. | Inspection |

# 6. NOTES

6.1 Definitions

6.1.1 Controller - a physical device that takes in user input that will communicate with the device.

6.1.2 Interactive Behaviors - certain types of behaviors that happen through interaction.

6.1.3 ArduPilot - open source software used for controlling autonomous vehicles specifically in unmanned aerial vehicles (UAV). Frameworks available to use Flight Control, Autonomous Navigation.

6.1.4 MediaPipe - open source software for computer vision and machine learning related to Video Analysis. Frameworks available to use Augmented Reality (AR), gesture recognition, image tracking.

6.1.5 Tello SDK - Documentation of commands for programming the drone.

6.1.6 Flutter - mobile development software that has user interfaces (UI) and user experiences (UX). Works well with MediaPipe since both are from Google.

## 6.2 Acronyms and Abbreviations

Table II. Acronyms and Abbreviations

| Abbreviation | Full name |
| --- | --- |
| SDK | Software Development Kit |
| iOS | IPhone Operating System |
| macOS | Mac Operating System |