**Software Requirements Specification**

**for**

**Leaflings**

**Version 1.1 approved**

**Prepared by João Pedro Santos**

**Grupo 10**

**26-09-2024**

**Table of Contents**

[1. Introduction 1](#_Toc180006189)

[1.1 Purpose 1](#_Toc180006190)

[1.2 Intended Audience and Reading Suggestions 1](#_Toc180006191)

[1.3 System Proposal 2](#_Toc180006192)

[1.4 Product Scope 3](#_Toc180006193)

[1.5 References 4](#_Toc180006194)

[2. Overall Description 5](#_Toc180006195)

[2.1 Product Perspective 5](#_Toc180006196)

[2.2 Product Functions 5](#_Toc180006197)

[2.3 User Classes and Characteristics 6](#_Toc180006198)

[2.4 Operating Environment 6](#_Toc180006199)

[2.5 Design and Implementation Constraints 6](#_Toc180006200)

[2.6 Assumptions and Dependencies 7](#_Toc180006201)

[3. External Interface Requirements 7](#_Toc180006202)

[3.1 User Interfaces 7](#_Toc180006203)

[3.2 Hardware Interfaces 8](#_Toc180006204)

[3.3 Software Interfaces 8](#_Toc180006205)

[3.4 Communications Interfaces 8](#_Toc180006206)

[4. Use Cases 9](#_Toc180006207)

[4.1 User Management 9](#_Toc180006208)

[4.2 Plant Database and Matchmaking 13](#_Toc180006209)

[4.3 Administration 17](#_Toc180006210)

[4.4 Other Functional Requirements 21](#_Toc180006211)

[5. Other Nonfunctional Requirements 22](#_Toc180006212)

[5.1 Security Requirements 22](#_Toc180006213)

[5.2 Software Quality Attributes 22](#_Toc180006214)

[Appendix A: Glossary 23](#_Toc180006215)

[Appendix B: Analysis Models 23](#_Toc180006216)

[Appendix C: To Be Determined List 23](#_Toc180006217)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| João Santos | 03/10/2024 | Corrections to | **1.1** |
| João Santos | 16/10/2024 | Adding Use Cases | **1.2** |

## Introduction

### Purpose

This document is dedicated to the 1.0 release of the application called **Leaflings,** by Group 10, for the course unit of software development lab at Escola superior de gestão e tecnologia (ESTG).  
  
This document will go over the main functionalities to be implemented in this release, that can be consulted at “**1.5 Product Scope**”.  
  
Group 10 is formed by the following students:   
 **. João Pedro Santos -** 8220256 **. Hugo Ricardo Guimarães -** 8220337

**. Sónia Raquel Degtyareva -** 8220114 **. Pedro Santos Pinho -** 8220307

### Intended Audience and Reading Suggestions

This document is intended for in team developers, testers and documentation writers, and for the accompanying teacher. This software requirement specification document will be going over the general idea for the project, the requirements, functional, and nonfunctional, the functionalities to be developed until the end of this version (1.0).

It is organized in parts, the **introduction**, where a general idea of the system, and the scope (what will be built in this 3 month stretch of work) are located;

The **overall description** provides a broad view of the app, describing the product, key features, user demographics, running platforms, and constraints to development;  
  
The **External Interface Requirements** section describes how the system interacts with users, hardware, and other software. Each subsection details specific interfaces;

The **Use Cases** describe how users should interact with the system;

The **Nonfunctional requirements** go over goals for the system to achieve, regarding security, performance and usability.  
  
The document can and should be read by the order it is numbered in.

### System Proposal

***Overview***

*The system our developing team proposes in this document aims to match users to houseplants best suited for them, having in mind certain characteristics, ranging from available time for taking care of the plants, to conditions such as natural light. The system aims to tackle the problem a lot of beginners face when it comes to starting to care for plants, which is a lack of experience leading to having their plants die, wasting their money and time on subpar care for houseplants. The system’s main goal is to ease users into the world of caring for plants, aiming to create a higher consciousness for wildlife and nature in general, while also helping said wildlife by preventing users from killing plants they thought would fit them, and preventing said users from also wasting their money and resources.*

***System Features***

*This system will primarily focus on easing the journey of a beginner plant enthusiast into the world of caring for plants and gardening. as such its core functionalities cover:*

**. User profile:** a customizable user profile, where the user will be able to input their living conditions, fields such as “available space “, humidity, natural lighting, time available for the plants. It would also provide a dedicated space for plants that the user currently has selected and is assumed to be taking care of in their house, where they’ll be able to track their progress across time.

**. Matching Plants to Users:** Based on the user’s parameters, the system will provide the user with a list of potential matches, from which the user will be able to select the ones they desire to try and cultivate.This is done with the intent of matching houseplants to their favored environments, raising the likelihood of said plants thriving.

**. Plant Almanac:** The system will provide a sort of plant almanac, accessible to users through a graphic interface, containing all plants available in the database. Through navigating this almanac, users will be able to filter, select and analyze the data regarding each plant, which should include a general description of the plants, one or multiple pictures, general tips on how to care for it, and its necessities.

**. Social Aspect** The system will provide a form of community engagement, through the implementation of public streaks (tracking how often a user cares for their plants and how well), leaderboards (tracking the number of plants a user successfully takes care of), and a graveyard feature where all the plants a user has not taken care well of will be recorded at.

**. Streaks –** much like Duolingo, the app will have a notification system that reminds the user to water their plants, and will keep track of how many days **in a row** a user has successfully done so.

**. XP system and Leaderboards –** Users will be pushed to compete based on who maintains their plants the best, or who maintains the most plants.

**Target Audience**

We aim to build our system’s user base targeting people that fit the following descriptions:   
  
**. Beginners:** People who are looking to get into gardening or add some green to their house life but lack the experience and confidence to do so alone**.**

**. Plant Experts:** people with more experience and comfortable with houseplants, possibly looking for a more exotic experience.

**. Plant Shops:** Potential partners with our app, there is the possibility of suggesting shops to our user base depending on their location.

***Conclusion***

*Our app aims to revolutionize how people interact with the prospect of caring for plants, through personalized plant recommendations and easy access to information about their plants.*

*The initial development phase will focus on creating the foundational components of the application, with the potential for future enhancements that could include a larger plant database.*

### Product Scope

**Objectives**

With this project, we intend to create a platform within which plant lovers of all levels of experience have a better chance at picking out houseplants that fit their needs, and also their capability to care for them.

The main purpose is to facilitate the daunting task of doing extensive research and risking making the wrong decisions about which plants are fitting for one’s home, whilst also teaching our future users the best practices for plant care.

As stated before, the team only has 3 months of developing time for this project, so we will be working towards building a minimum viable product of our system, so all features that are not considered to be necessary for the prototype of this app will not be implemented in this version of the project.

**Scope Description**

The team is committed to implementing the platform in two distinct versions, a website and a mobile application.

The features to be developed in this initial phase are as such:

* **Customizable user profiles**, in which the user will be able to describe their home environment, available time for plant caretaking, and other preferences;
* **An extensive dictionary of plants,** complete with care guides for each entry;
* **An algorithm which upon request will match a user to suitable plants**, through the analysis of the user profile, matching it with the needs and characteristics of the available plants at our database, such as luminosity, difficulty of care, water needs and such;
* **The option to select which plants a user has effectively planted**, in order to easily access their dictionary entries
* **A reminder system**, in order to incentivize the users to take consistent care of their plants through the use of reminders (in the form of notifications), including a streak component, which will keep count of the consecutive entries of caretaking, rewarding responsibility.

**All other features described in system proposal are therefore out of scope, and not to be developed in the following 3 months for the course unit of software development lab.**

### References

*All references this document will make reference to can be found at the group’s Gitlab project, where we will have the following documents listed:*

**. Configuration Management Plan –** in this document we outline processes and procedures used to track changes in the development of the system. It can be found at:

**https://gitlab.estg.ipp.pt/8220256/lds\_24\_25\_leaflings/-/wikis/Leaflings%20Project%20CMP**

**. Test Master Plan -** in this document we outline the overall testing strategy the group will be using in this project., objectives, and schedules for when it comes to testing our code. It can be found at:

**https://gitlab.estg.ipp.pt/8220256/lds\_24\_25\_leaflings/-/wikis/Leaflings%20Project%20TMP**

**. User Manual -** document designed to help end users understand how to use a product, typically software or hardware**.** It can be found at:

**https://gitlab.estg.ipp.pt/8220256/lds\_24\_25\_leaflings/-/wikis/User%20Manual**

**. Diagrams – This section contains diagrams that illustrate the architecture and interactions within the system.** It can be found at:

**https://gitlab.estg.ipp.pt/8220256/lds\_24\_25\_leaflings/-/wikis/Diagrams  
  
. Test Documentation -** This section includes test cases, test scenarios, results, and any reports generated from the testing phase. It can be found at:

**https://gitlab.estg.ipp.pt/8220256/lds\_24\_25\_leaflings/-/wikis/Test%20Documentation**

## Overall Description

### Product Perspective

Our group noticed that for many activities, like learning a language, hiking, and others, there are plenty of apps that are aimed at helping the process and tracking said process of its user base.

However, we noticed that we had never heard of such an app for when it comes to caring for plants, and after searching, although we noticed there are tracking apps for this effect, none of them really “gamifies” the process, in aims of making it more engaging and fun.   
  
The above description summarizes where the idea for the application came from; With further discussion the group decided it should be a self-contained product with the potential for future expansion through the addition of new features, much alike the application used as inspiration for the project “Duolingo “. It is intended as an app that the userbase would interact with on a day-to-day basis, where people can get educated, and share their experiences with the gardening world.

We found it to be a fun to work in project that would, at the same time, bring a positive weight to the social landscape, through engaging people in a community revolving around plants, making it so there would be more places around filled with greens, investing people into caring more heavily for plant life, and doing all this while using the game aspect of the system in order to drive engagement instead of lecturing the general population.

### Product Functions

The product **must** allow for the following functions:

**. User profile –** all users should have to **make a profile** before delving further into the system, this is intended in order to generate user engagement, by creating a beginning sense of commitment. In the profile creation the user will have the possibility to **enter information** about their **time available to care for the plants**, select qualitative options regarding their **space availability**, the **luminosity** for the plants.

**. Plant Manual –** the system ought to possess a graphical interface, representing a manual of the plants covered (by covered we mean, are in the database and have all the information necessary to support a user that chooses to take care of said plant).

**. Streak Record –** much like the inspiration system (Duolingo), we aim to implement a streak system, that’ll track for how many days a user has, for example, correctly watered their plant(s).

### User Classes and Characteristics

*<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>*

*An initial in our section* **1.3 System Proposal,**

### Operating Environment

The **Leaflings** application will be available in a web app format, ensuring accessibility on various operating environments, including Windows, Linux, and macOS. It is designed with the following web browsers in mind:

**. Google Chrome (latest two versions)**

**. Mozilla Firefox (latest two versions)**

Other than the webapp format, the group will be developing a mobile version of the system destined to the android and iOS platforms, we’ll be making use of **react native** for this end.

### Design and Implementation Constraints

Our project will be subject to several constraints, which might limit our options:

**. Time Constraints**: The team only has **3 months** to complete the project, necessitating a focus on delivering a Minimum Viable Product (MVP) with core functionalities.

**. Budget Limitations**: Due to a lack of funds, we will not utilize any paid interfaces, libraries, or web services.

We will be using **react** for the webapp version and **Kotlin** for the mobile version so anything outside of the capability of these frameworks will be beyond the scope our project, it may also cause the application to not be compatible with old browser versions, or older android and iOS versions.

### Assumptions and Dependencies

The development process of the Leaflings application will be affected by and based on multiple *assumptions and dependencies which influence the flow of development, depending on how accurate these turn out to be.*

*The assumptions are as follows:*

* *It is assumed that the future users of the platform have sufficient understanding of technology (namely, web and mobile applications) in order to navigate the system without the need of extensive training beyond short guides and/or tutorials;*
* *We assume that users will have access to hardware and software environments that allow for the application to run smoothly, such as OS compatibility, system capability and stable network connection;*
* *It is assumed that the initial database (to be implemented upon release) will be sufficiently comprehensive, providing users with information most pertinent to them, which is to say, include the most common houseplants recommended for ownership;*
* *It is assumed that all members of the development team will be available for meetings, willing to collaborate, and aid in decision making throughout the progress of the project, as well as possess all the technical skills to effectively implement the platform.*

*The dependencies are as such:*

* *The choice of React and React Native for the development of the project, demands that these frameworks and the tools necessary for their support are available and stable during the development period and beyond;*
* *Any external API’s or resources used will be presumed to remain available for the foreseeable future, in order to maintain all functionality during and post development and deployment;*
* *As GitLab will be used for version control and collaboration, these elements rely heavily on its availability for all team members;*
* *Lastly, the project will be heavily influenced by user feedback during testing and prototyping, hence it will be dependent on the users who will be submitting their thoughts and any issues encountered. Any delays encountered during these phases will impact the development.*

## External Interface Requirements

### User Interfaces

*<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.>*

### Hardware Interfaces

*<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.>*

### Software Interfaces

*<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.>*

### Communications Interfaces

*<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.>*

## Use Cases

### User Management

* + 1. **Register User**

|  |  |
| --- | --- |
| Name | Register User |
| Actors | User |
| Pre-Conditions | Users must have access to the platform. |
| Typical-Flow | A new user will access the platform and choose the **Register** option, upon which they will be prompted to complete a form with the basic information needed to create an account. After completing the form, the user may then select the **Submit** button, effectively creating their account. |
| Alternative-Flow | **Improper form data:**  In case the user fails to properly fill out the form, they will not be able to submit their information, until the form is correctly filled out.  **Redundancy:** If a user’s unique identifier (email) coincides with one already present in the database, the user will be notified, and will not be able to create an account. |
| Extensions | **N/A** |
| Post-Conditions | The user will now have access to all of the app’s functionalities, and will be able to log in with the credentials they have chosen. |

#### 

* + - 1. **Functional Requirements**

|  |  |
| --- | --- |
| **Category** | **User Management** |
| **ID** | FRUM-01 |
| **Name** | Login and Registration |
| **Description** | Users must be able to register in the application, upon which their data will be securely kept in the database. If a user is registered, they will then be able to log into their account, allowing them to use the platform. |
| **Priority** | 1 - Essential |
| **State** | Proposed |
| **Restrictions** | N/A |

* + 1. **Login**

|  |  |
| --- | --- |
| Name | Login |
| Actors | User, Admin |
| Pre-Conditions | Users must be registered. |
| Typical-Flow | A registered user will arrive at the entry screen, and, if they aren’t logged in, they can then enter their credentials, upon which they will be granted access to their respective dashboard. |
| Alternative-Flow | **Improper form data:**  In case the user fails to properly fill out the form, they will not be able to submit their information, until the form is correctly filled out. |
| Extensions | **N/A** |
| Post-Conditions | The user will now have access to all of the app’s functionalities, and will be able to log in with the credentials they have chosen. |

* + - 1. **Functional Requirements**

|  |  |
| --- | --- |
| **Category** | **User Management** |
| **ID** | FRUM-01 |
| **Name** | Login and Registration |
| **Description** | Users must be able to register in the application, upon which their data will be securely kept in the database. If a user is registered, they will then be able to log into their account, allowing them to use the platform. |
| **Priority** | 1 - Essential |
| **State** | Proposed |
| **Restrictions** | N/A |

#### 

* + 1. **Create User Profile**

|  |  |
| --- | --- |
| Name | Create User Profile |
| Actors | User |
| Pre-Conditions | Users must have access to the platform, and must be registered |
| Typical-Flow | Once a user is registered, they will be able to complete their profile with their caretaking information, which is to say, describe their environment and conditions (by filling out a multiple-choice form). To do this, a user will access their profile, **Edit Profile**, and fill out all of the relevant fields. Once complete, the user will select **Submit** in order to persist their data. |
| Alternative-Flow | **Improper form data:**  In case the user fails to properly fill out the form, they will not be able to submit their information, until the form is correctly filled out. |
| Extensions | A user will be able to change and update their profile at any given time. |
| Post-Conditions | The user will now have access to a customized feed of suitable plants for their environment. |

* + - 1. **Functional Requirements**

|  |  |
| --- | --- |
| **Category** | **User Management** |
| **ID** | FRUM-02 |
| **Name** | Profile Creation |
| **Description** | Registered users should be able to create a profile detailing the available environment for plant care (caretaker profile). |
| **Priority** | 1 - Essential |
| **State** | Proposed |
| **Restrictions** | Only registered users should have access to this functionality |

#### 

### 

* + 1. **Alter User Profile**

|  |  |
| --- | --- |
| Name | Alter User Profile |
| Actors | User |
| Pre-Conditions | Users must have access to the platform, and must be registered |
| Typical-Flow | Once a user has a profile created, they can then access it, selecting the **Edit** option, and updating any information they deem necessary. |
| Alternative-Flow | **Improper form data:**  In case the user fails to properly fill out the form, they will not be able to submit their information, until the form is correctly filled out. |
| Extensions | **N/A** |
| Post-Conditions | The user will now have access to a customized feed of suitable plants for their environment. |

* + - 1. **Functional Requirements**

|  |  |
| --- | --- |
| **Category** | **User Management** |
| **ID** | FRUM-02.1 |
| **Name** | Profile Alteration |
| **Description** | A user with an existing profile should be able to change it at any time, updating their information, as well as their caretaker profile |
| **Priority** | 1 - Essential |
| **State** | Proposed |
| **Restrictions** | Only registered users with an existing profile should have access to this functionality. |

## 

### Plant Database and Matchmaking

* + 1. **Access Plant Directory**

|  |  |
| --- | --- |
| Name | Access Plant Directory |
| Actors | User, Admin |
| Pre-Conditions | Users must have access to the platform, and must be registered, and the system and database must be available. |
| Typical-Flow | User selects the **Almanac** section of the application. |
| Alternative-Flow | **N/A** |
| Extensions | Once in the **Almanac** page, a user can use the search bar to look for specific plants.  A user can also access a specific plant, and explore the **Plant Care Guide** or simply read the plant’s description |
| Post-Conditions | The user will now have access to a feed of all the plants available in the database, along with their information. |

* + - 1. **Functional Requirements**

|  |  |
| --- | --- |
| **Category** | **Plant Database and Matchmaking** |
| **ID** | FRP-01 |
| **Name** | Plant Directory Access |
| **Description** | Users should be able to browse the entirety of the plant catalog, accessing information about each entry. |
| **Priority** | 1 - Essential |
| **State** | Proposed |
| **Restrictions** | Only registered users should have access to this functionality |

|  |  |
| --- | --- |
| **Category** | **Plant Database and Matchmaking** |
| **ID** | FRP-01.1 |
| **Name** | Plant Information Display |
| **Description** | Upon selection of a plant present in the catalog, a user should be able to inspect its characteristics. |
| **Priority** | 1 - Essential |
| **State** | Proposed |
| **Restrictions** | Only registered users should have access to this functionality |

|  |  |
| --- | --- |
| **Category** | **Plant Database and Matchmaking** |
| **ID** | FRP-01.2 |
| **Name** | Caretaking Guide |
| **Description** | Associated with each plant, besides their general information and characteristics, there must be a caretaking guide, detailing the steps and precautions to take when caring for this plant. |
| **Priority** | 1 - Essential |
| **State** | Proposed |
| **Restrictions** | Only registered users should have access to this functionality, the user’s caretaker profile must be completed, and there must be plants in the database capable of thriving according to user’s profile. |

|  |  |
| --- | --- |
| **Category** | **Plant Database and Matchmaking** |
| **ID** | FRP-01.3 |
| **Name** | Searching and Filtering |
| **Description** | Users should be able to look for plants within the database, applying filters in regards to the type of plant. |
| **Priority** | 2 - High Priority |
| **State** | Proposed |
| **Restrictions** | Only registered users should have access to this functionality |

* + 1. **Matching Plants to Users**

|  |  |
| --- | --- |
| Name | Matching Plants to Users |
| Actors | User |
| Pre-Conditions | Users must have access to the platform, and must be registered, their profile must be complete, and the system and database must be available. |
| Typical-Flow | User accesses the **Almanac** and selects the **For Me** option.  The system will compare the user’s available environment and the plant’s needs, and create a feed containing the plants suitable for the user. |
| Alternative-Flow | **No suitable plants:** If for some reason the user’s profile does not match with any plants in any capacity, the filter will return no results. |
| Extensions | Much like in the **Almanac** page, a user can use the search bar to look for specific plants.  A user can also access a specific plant, and explore the **Plant Care Guide** or simply read the plant’s description |
| Post-Conditions | The user will now have access to a customized feed of suitable plants for their environment. |

* + - 1. **Functional Requirements**

|  |  |
| --- | --- |
| **ID** | FRP-02 |
| **Name** | Plant Matchmaking |
| **Description** | Through the use of a matchmaking algorithm, and taking into account the user profile and plant characteristics, the program should be able to provide a personalized catalog containing only plants which abide by the user’s caretaker profile specifications. |
| **Priority** | 1 - Essential |
| **State** | Proposed |
| **Restrictions** | Only registered users should have access to this functionality, the user’s caretaker profile must be completed, and there must be plants in the database capable of thriving according to the user's profile. |

* + 1. **Adding Plants to User Favorites**

|  |  |
| --- | --- |
| Name | Adding Plants to User Favorites |
| Actors | User |
| Pre-Conditions | Users must have access to the platform, and must be registered, their profile must be complete, and the system and database must be available. |
| Typical-Flow | A user accesses a specific plant’s page through the **Almanac**. Then, within the plant page, a user can select the **Add** option, which will add a plant to the user’s favorites (**Garden**), making it easily accessible to check any information. |
| Alternative-Flow | **N/A** |
| Extensions | **N/A** |
| Post-Conditions | The plant is added to the user’s collection, for easier access, and will allow the user to enable reminders to care for it, if they effectively own this plant. |

* + - 1. **Functional Requirements**

|  |  |
| --- | --- |
| **Category** | **Plant Database and Matchmaking** |
| **ID** | FRP-03 |
| **Name** | Add Plant to Garden |
| **Description** | If the user decides to effectively care for one of the plants present in the catalog, they can then add it to their **Garden**, which will be a separate collection of plants selected by the user. Here, their information will be readily available. |
| **Priority** | 1 - Essential |
| **State** | Proposed |
| **Restrictions** | Only registered users should have access to this functionality, and the user must have selected at least one plant to care for, in order for the Garden to be available. |

### Administration

* + 1. **Remove User**

|  |  |
| --- | --- |
| Name | Remove User |
| Actors | Admin |
| Pre-Conditions | User must have administrative permission and be authenticated |
| Typical-Flow | An admin will access the **User Directory**, and select the user they wish to remove. They will then be prompted with a confirmation message, and upon answering affirmatively, the user’s account will be **Deactivated**. |
| Alternative-Flow | **Negative confirmation:**  In case the admin does not confirm the deletion, the action will not be performed. |
| Extensions | **N/A** |
| Post-Conditions | The user’s account will no longer be available. |

* + - 1. **Functional Requirements**

|  |  |
| --- | --- |
| **Category** | **Administration** |
| **ID** | FRA-01 |
| **Name** | User Deletion |
| **Description** | The platform admins will be able to remove user accounts from the platform |
| **Priority** | 5 - Essential |
| **State** | Proposed |
| **Restrictions** | Only authorized admins have access to this functionality |

* + 1. **Access User Information**

|  |  |
| --- | --- |
| Name | Access User Information |
| Actors | Admin |
| Pre-Conditions | User must have administrative permission and be authenticated |
| Typical-Flow | An admin will access the **User Directory**, upon which they will be able to navigate the list of registered users. |
| Alternative-Flow | **N/A** |
| Extensions | An admin will be able to view a specific user’s information. |
| Post-Conditions | **N/A** |

* + - 1. **Functional Requirements**

|  |  |
| --- | --- |
| **Category** | **Administration** |
| **ID** | FRA-02 |
| **Name** | User Information Access |
| **Description** | The platform admins will have access to user information, except passwords |
| **Priority** | 5 - Essential |
| **State** | Proposed |
| **Restrictions** | Only authorized admins have access to this functionality |

* + 1. **Plant Creation**

|  |  |
| --- | --- |
| Name | Plant Creation |
| Actors | Admin |
| Pre-Conditions | User must have administrative permission and be authenticated |
| Typical-Flow | An admin will access the **Plant Directory**, and select the **Add new Plant** option. They will then be prompted to fill out a form detailing the plant information. Once the form is completed, they will be able to **Submit** the new plant, upon which it will be persisted in the database. |
| Alternative-Flow | **Improper form data:**  In case the user fails to properly fill out the form, they will not be able to submit the plant, until the form is correctly filled out.  **Redundancy:** If a plant’s unique identifier (scientific name) coincides with one already present in the database, the user will be notified, and will not be able to proceed with the addition. |
| Extensions | **N/A** |
| Post-Conditions | The plant will be available in the directory and ready for operations. |

* + - 1. **Functional Requirements**

|  |  |
| --- | --- |
| **Category** | **Administration** |
| **ID** | FRA-03 |
| **Name** | Plant Creation |
| **Description** | The platform admins will be able to add plants to the database, without directly accessing it |
| **Priority** | 5 - Essential |
| **State** | Proposed |
| **Restrictions** | Only authorized admins have access to this functionality |

* + 1. **Edit Plant Information**

|  |  |
| --- | --- |
| Name | Edit Plant Information |
| Actors | Admin |
| Pre-Conditions | User must have administrative permission and be authenticated |
| Typical-Flow | An admin will access the **Plant Directory**, and select the plant they wish to edit, and edit the fields as they see fit. After this is done, they can **Submit** the update. |
| Alternative-Flow | **Improper form data:**  In case the user fails to properly fill out the form, they will not be able to submit the plant, until the form is correctly filled out.  **Redundancy:** If a plant’s unique identifier (scientific name) coincides with one already present in the database, the user will be notified, and will not be able to proceed with the edition. |
| Extensions | **N/A** |
| Post-Conditions | The plant’s information is now updated and will be used in the platform. |

* + - 1. **Functional Requirements**

|  |  |
| --- | --- |
| **Category** | **Administration** |
| **ID** | FRA-04 |
| **Name** | Plant Edition |
| **Description** | The platform admins will be able to edit/update plant information |
| **Priority** | 5 - Essential |
| **State** | Proposed |
| **Restrictions** | Only authorized admins have access to this functionality |

### 

### Other Functional Requirements

* + 1. **Engagement**

|  |  |
| --- | --- |
| **Category** | **Engagement** |
| **ID** | FRE-01 |
| **Name** | Reminder System |
| **Description** | Given the user’s Garden collection, reminders will be sent out, to remind the user of when they need to water their plant, or simply log the day into the application. |
| **Priority** | 3 - Medium Priority |
| **State** | Proposed |
| **Restrictions** | Only registered users should have access to this functionality, and the user must have selected at least one plant to care for. |

|  |  |
| --- | --- |
| **Category** | **Engagement** |
| **ID** | FRE-02 |
| **Name** | Streak System |
| **Description** | With continuous logging of caretaking into the application, the system will award the user by recording the consecutive entries, incentivizing consistency and responsibility. |
| **Priority** | 3 - Medium Priority |
| **State** | Proposed |
| **Restrictions** | Only registered users should have access to this functionality, and the user must have selected at least one plant to care for. |

# 

## Other Nonfunctional Requirements

### Security Requirements

|  |  |
| --- | --- |
| **Category** | **Safety and Security** |
| **ID** | NFRS - 01 |
| **Name** | Password Encryption |
| **Description** | The user’s password will never be directly kept in the database, without first being encrypted, for security reasons, ensuring a robust layer of protection. |
| **Priority** | 1 - Essential |
| **State** | Closed |
| **Restrictions** | N/A |

|  |  |
| --- | --- |
| **Category** | **Safety and Security** |
| **ID** | NFRS - 02 |
| **Name** | Token Authentication |
| **Description** | The authentication will be done with the use of access tokens, providing security and efficiency, with server side keys. |
| **Priority** | 1 - Essential |
| **State** | Closed |
| **Restrictions** | N/A |

## 

### Software Quality Attributes

|  |  |
| --- | --- |
| **Category** | **Accessibility** |
| **ID** | NFRA - 01 |
| **Name** | Ease of Use |
| **Description** | It is crucial that the system is developed in such a way that users will find it intuitive and easy to navigate. As such, the development team will use the www.acessibilidade.gov.pt website in order to gauge the level of accessibility as the application gets developed, to ensure the best possible experience. |
| **Priority** | 2 - High Priority |
| **State** | Closed |
| **Restrictions** | N/A |

## 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: To Be Determined List

*<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>*