Software Requirements Document for Reiman Reality

TEAM: SD-05

AUTHORS: Evan Hilton, Hoi Phung, Ian Gluesing, Jacob Gongora, Matthew Vozenilek

Version	Date	Author	Change
1.0	10 Nov. 2022	EH	

Table of Contents

1 Introduction 3
1.1 Purpose 3
1.2 Scope 3
1.3 Definitions, acronymns, abbreviations 3
1.4 References 3
2 Overall Description 4
2.1 Product Perspective 4
2.2 Product functions 14
2.3 User characteristics 16
2.4 Constraints 16
2.5 Assumptions and Dependencies 16
3 Specific Requirements 17
3.1 FEATURES 17
3.2 Performance requirements 17
3.3 Design Constraints 17
3.4 Software System Attributes (NON-FUNCTIONAL REQUIREMENTS) 17

1 Introduction

1.1 PURPOSE

The purpose of this document is to establish how the user application should interact with end users and how the administrator application interacts with administrators, and to establish all requirements for these applications.

1.2 SCOPE

This SRS covers a number of potential use cases that users may encounter, as well as an overview of the project and its intended uses. It also includes information on the project's UI sketches, but the primary purpose is to give detailed descriptions of anticipated use cases.

1.3 DEFINITIONS, ACRONYMNS, ABBREVIATIONS

Term	Description
Augmented Reality (AR)	The use of computer graphics overlaid on top of a camera input
AR model	The three-dimensional graphics (models) used by AR
Natural Feature Tracking (NFT)	The tracking of features in the environment by the use of a camera and computer
NFT marker	A parsing of a real-life object that allows the placement of AR models on top of the real-life object

1.4 REFERENCES

https://aframe.io/

https://ar-js-org.github.io/AR.js-Docs/

2 Overall Description

Reiman Gardens is looking for a new way for guests to experience the exhibits at the Gardens. Currently, some exhibits are only available during certain seasons (such as tulips), and some exhibits are rotated out each year. This project aims to use augmented reality (AR) to allow exhibits to be viewed indefinitely, as well as allow for the creation of AR-only exhibits that would be otherwise infeasible.

2.1 PRODUCT PERSPECTIVE

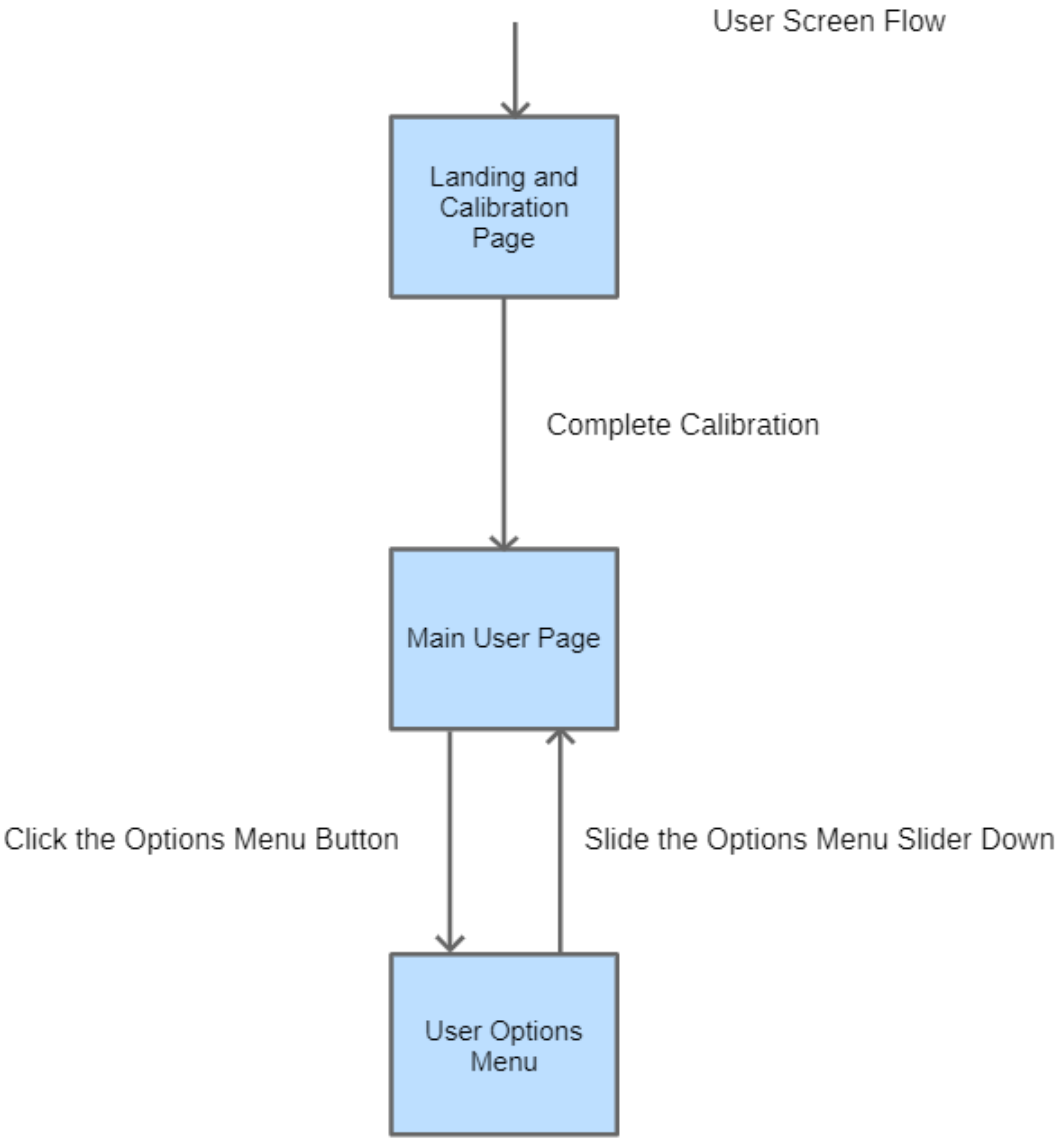
“Pokémon Go”, a mobile app that displays AR models of Pokémon on top of a camera feed
“Nintendo 3DS AR Games”, an application for the Nintendo 3DS that displays AR models on top of an NFT marker – usually a graphic on a card.

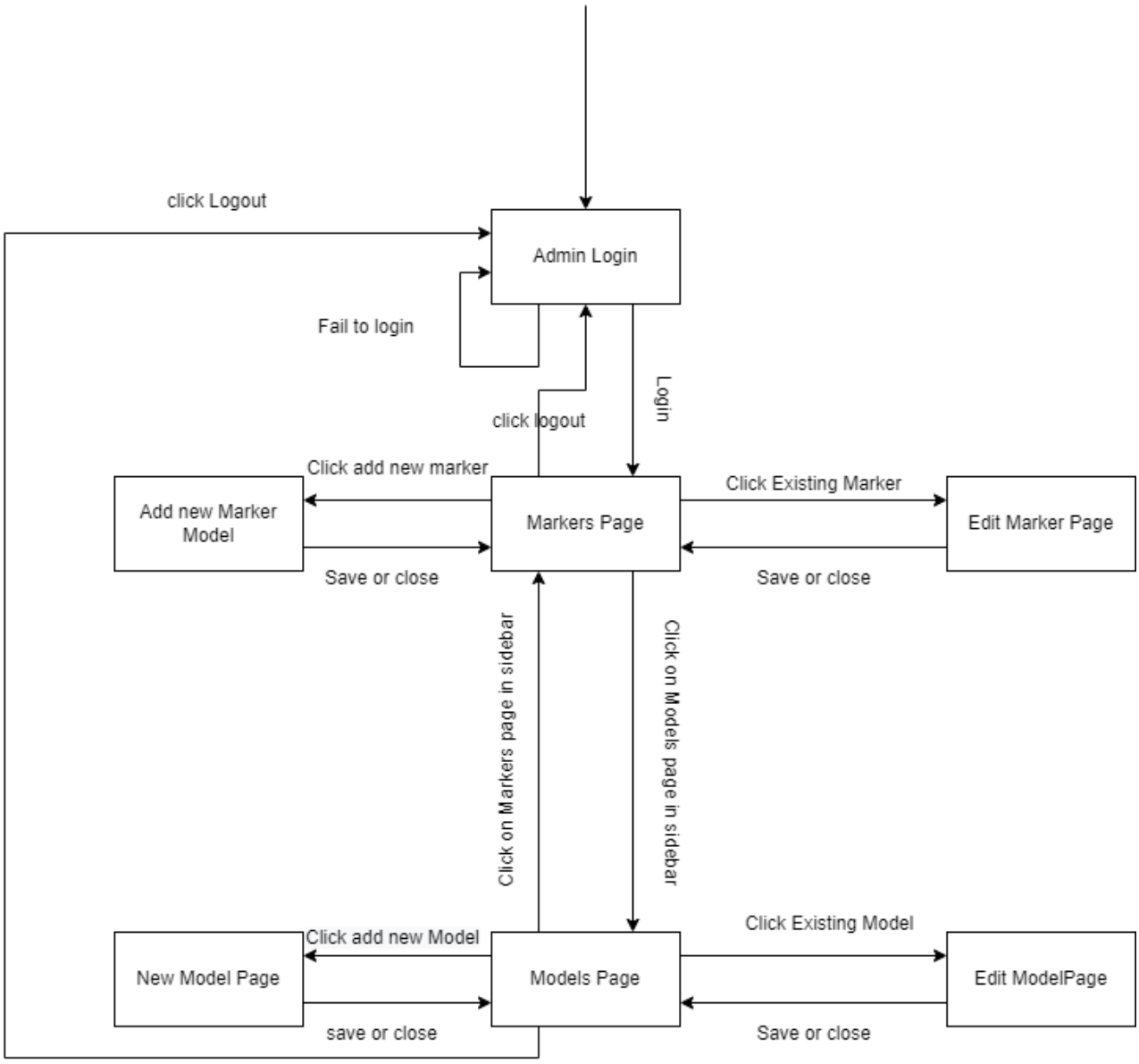
2.1.1 Concept of Operations

This web application will allow users to use their phone cameras to view AR models as designated exhibits in the Gardens. A separate webpage will allow administrators to create NFT markers of exhibit locations and add new AR models to be displayed at the locations encoded by the NFT markers. Administrators will also be able to edit existing NFT markers and AR models.

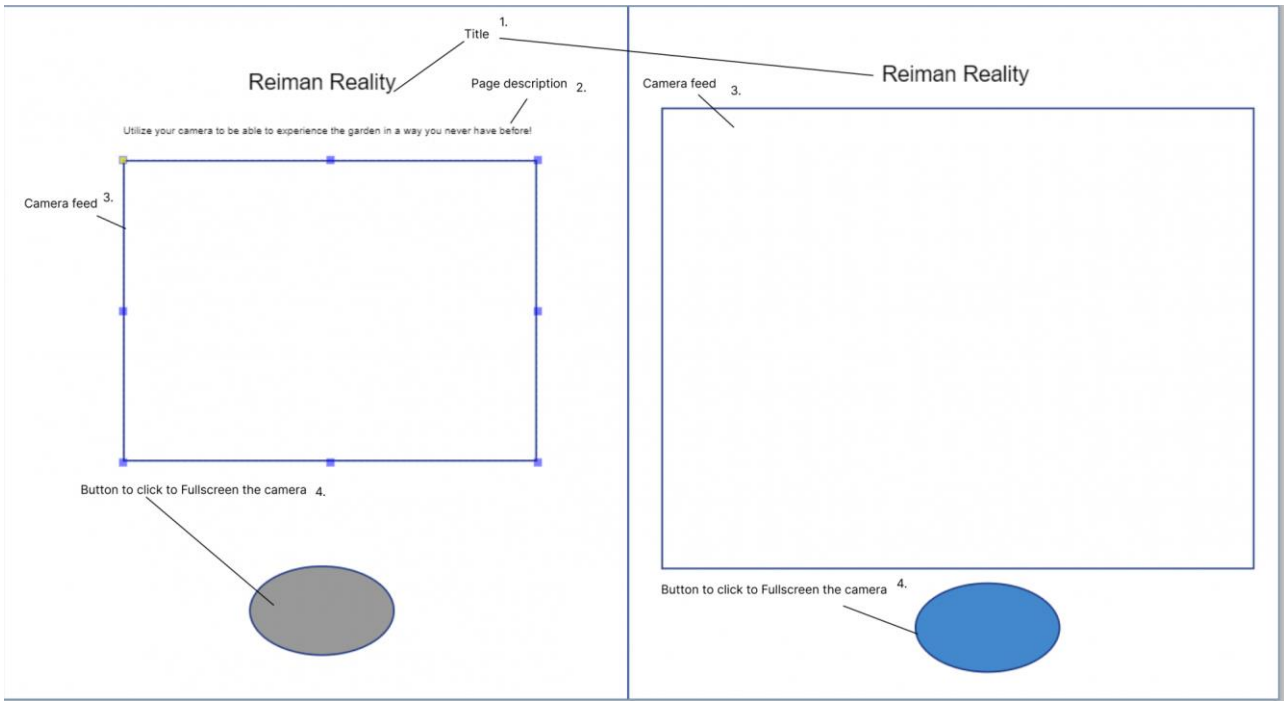
These applications will be backed by a database, which will store NFT markers, AR models, and the list of administrators.

2.1.2 Major User Interfaces

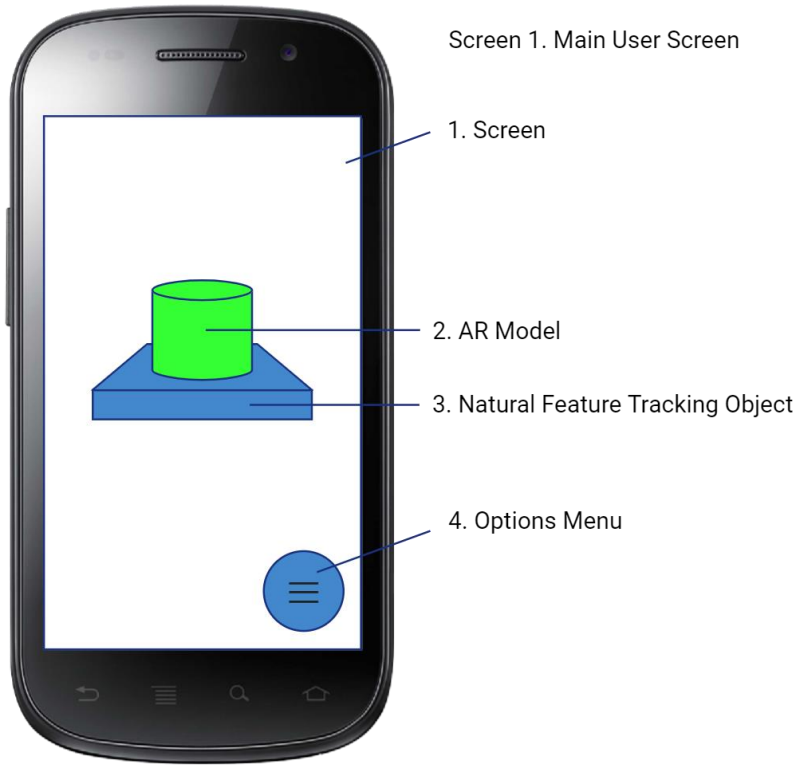

User Screen Flow
Landing and Calibration Page
Complete Calibration
Main User Page
Click the Options Menu Button
Slide the Options Menu Slider Down
User Options Menu


click Logout
Admin Login
Fail to login
click logout
Login
Click add new marker
Add new Marker Model
Markers Page
Click Existing Marker
Edit Marker Page
Save or close
Save or close
Click on Markers page in sidebar
Click on Models page in sidebar
Click add new Model
New Model Page
Models Page
Click Existing Model
Edit ModelPage
save or close
Save or close

2.1.2.1 User Calibration Screen



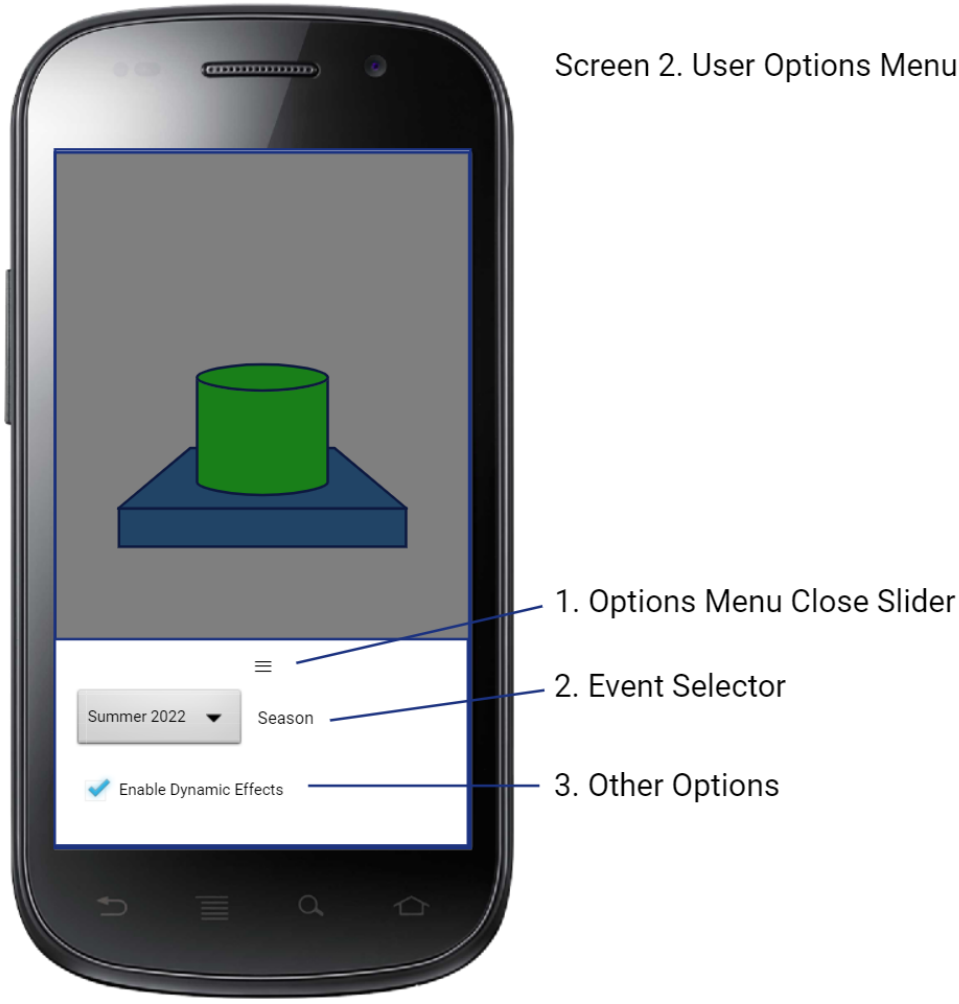
2.1.2.2 Main User Screen



The main user screen allows users to view AR models placed on top of Natural Feature Tracking (NFT) objects. This screen is displayed for the majority of the time the user is interactng with the application. AR models are only displayed when the appropriate NFT object is visible.

The screen (1) displays the current contents of the user's phone's back camera. A minimum resolution of 360 x 740 is supported. If a valid NFT object (3) is detected, an AR model (2) will be displayed. Note that each NFT object may have multiple AR models. If an NFT object has multiple AR models associated with it, the most recent AR model will be displayed by default. If a user wishes to view a previous AR model, it may be selected in the options menu (4).

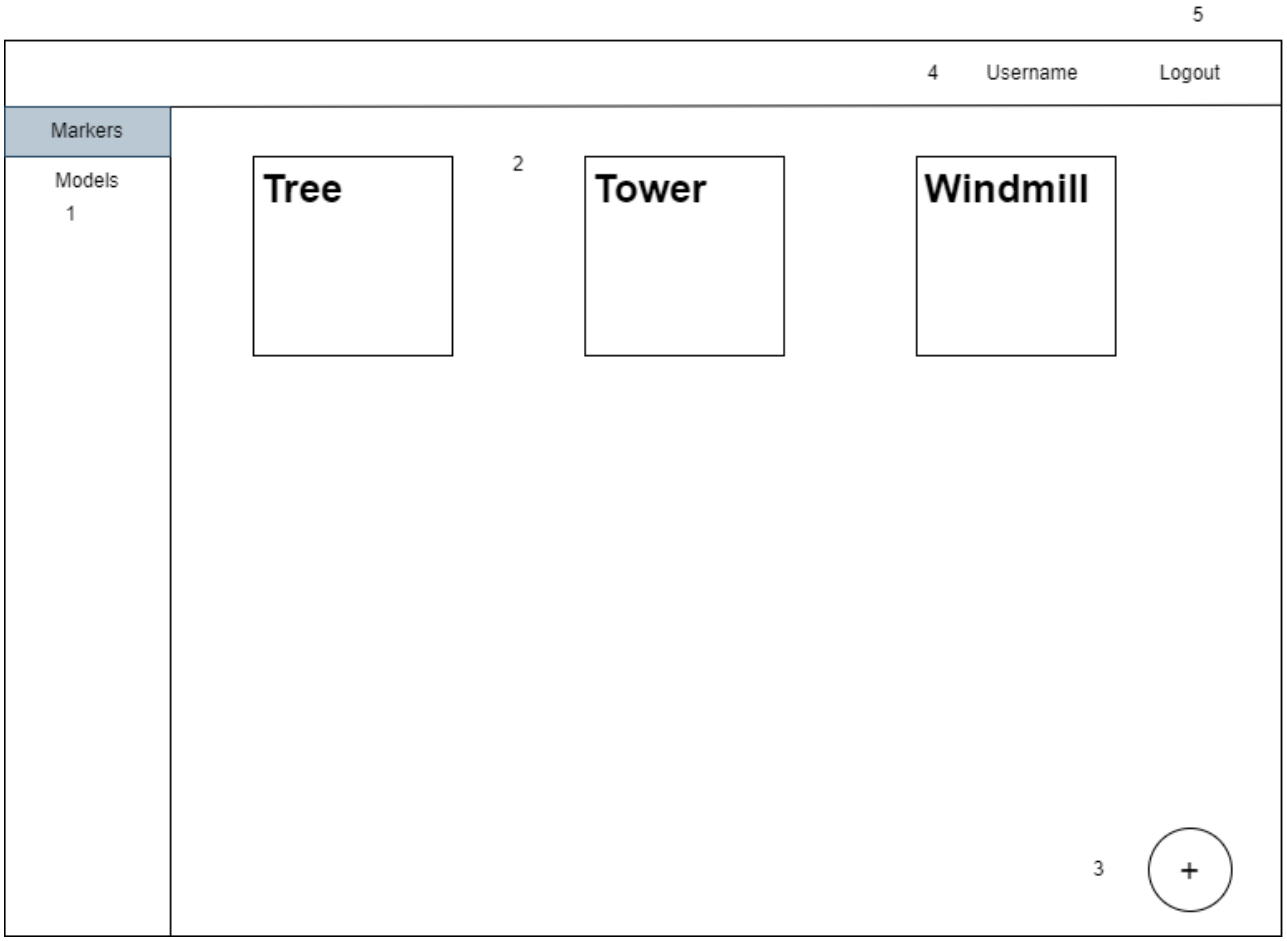
2.1.2.3 User Options Menu



The user options menu allows the user to configure some options of the application. This options menu appears when the user presses the option menu button on the main user screen.

The options menu close slider (1) allows the user to close the options menu and return to the main user screen. The event selector (2) allows users to choose alternative AR models associated with the current NFT object, as described in the screen sketch for the main user screen. In addition, some additional options may be available (3), such as enabling animated models (here referred to as "dynamic effects"). These can be disabled to save battery.

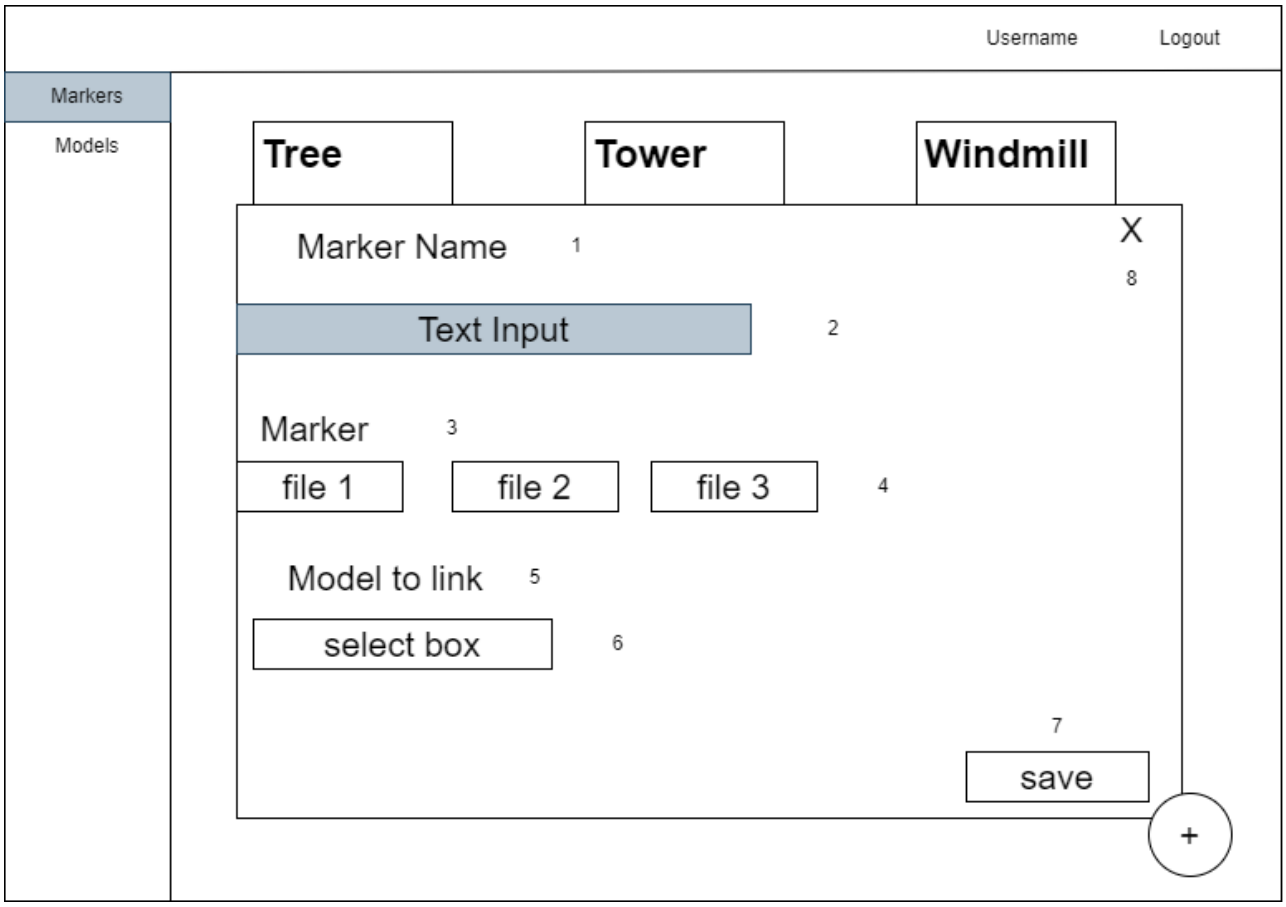
2.1.2.4 Admin NFT Markers Page



This is the markers page. It is the first screen an admin will see upon logging in this allows a administrator to view all markers that have been uploaded.

1. Screen select by default the admin is on the markers page upon login but they can swap between markers and models
2. Active models, these are the models currently saved by the user. they are clickable and bring up an interface on click that allows editing
3. Add new model: this button is when a user wants to create a new model they click this and a modal pops up that lets them enter all needed information
4. Username: static text displaying who is logged in
5. log out: takes a user back to the login screen for administrators

2.1.2.5 Admin Add New Marker Page



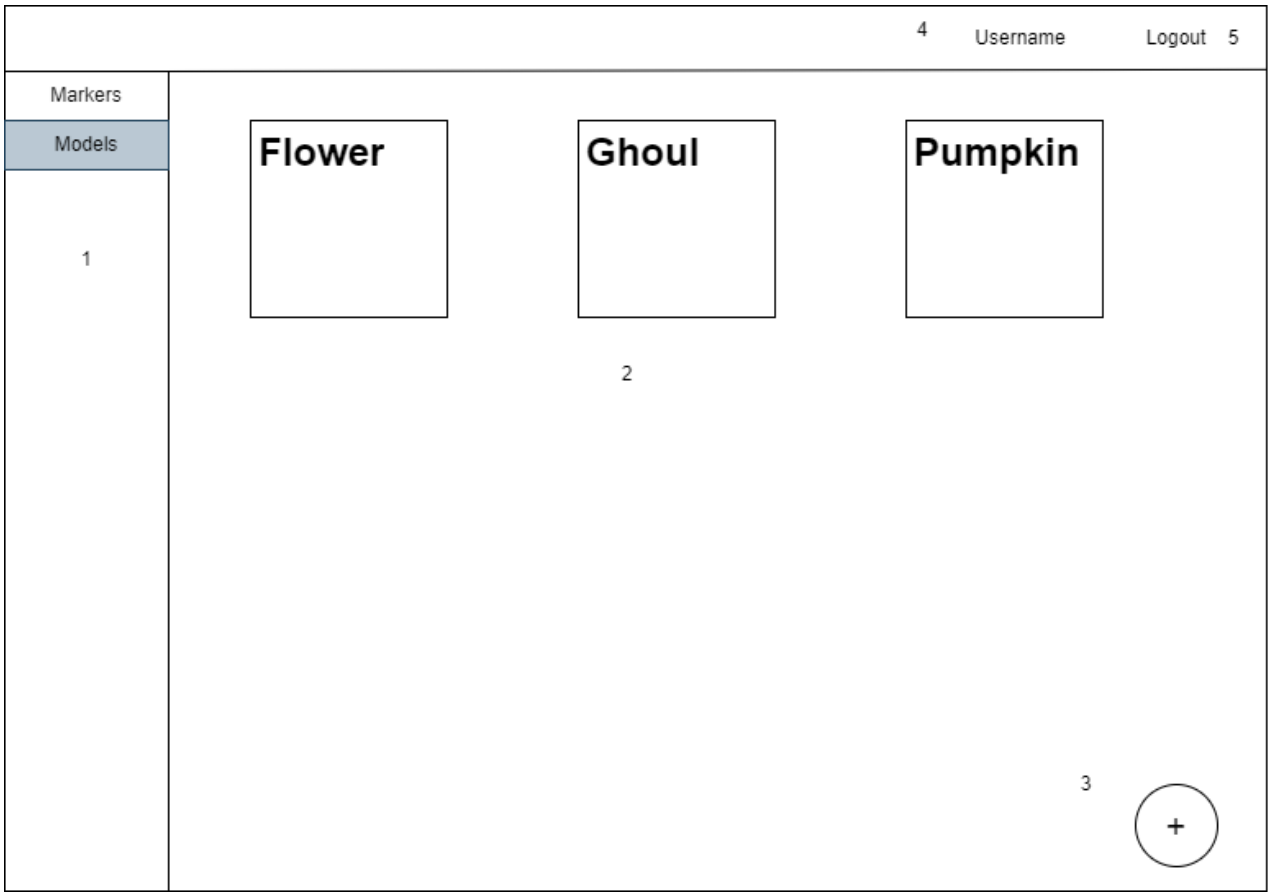
When clicking on a box or adding a new marker this modal shows up

1. Marker name is the heading for the marker
2. a text input allowing a user to enter a new name / rename a marker
3. Marker section identifier
4. File inputs for markers allowing a user to select the 3 required files for markers
5. section for linking a model

6 has all current models named and lets a user choose one in a select box

7. save your changes or add a new marker
8. exit the modal returning to the base page

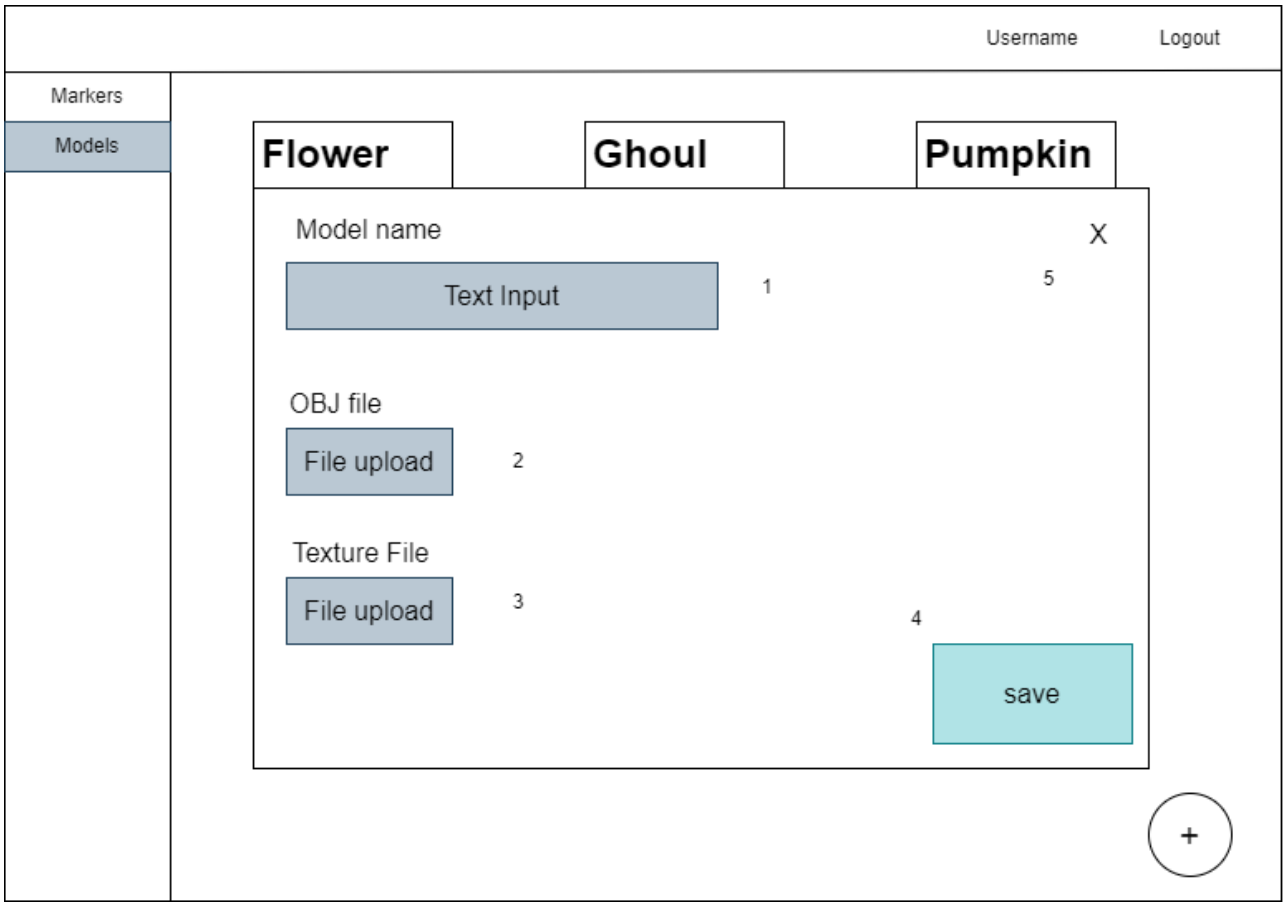
2.1.2.6 Admin AR Models Page



Models section

1. similar to the markers page version this section allows a user to see what page they are on to edit
2. All the active models overflow will be scrollable if required
3. Create a new model object
4. current user logged in as
5. log out and return to login screen

2.1.2.7 Admin Add New Model Page



1. Naming section for the model consists of its title and an input to change the name

2. OBJ file section. we will be using OBJ files so this input allows a user to upload their model to the server

3. Texture file upload, OBJ files require texture to be displayed so a user can upload a set texture for a file

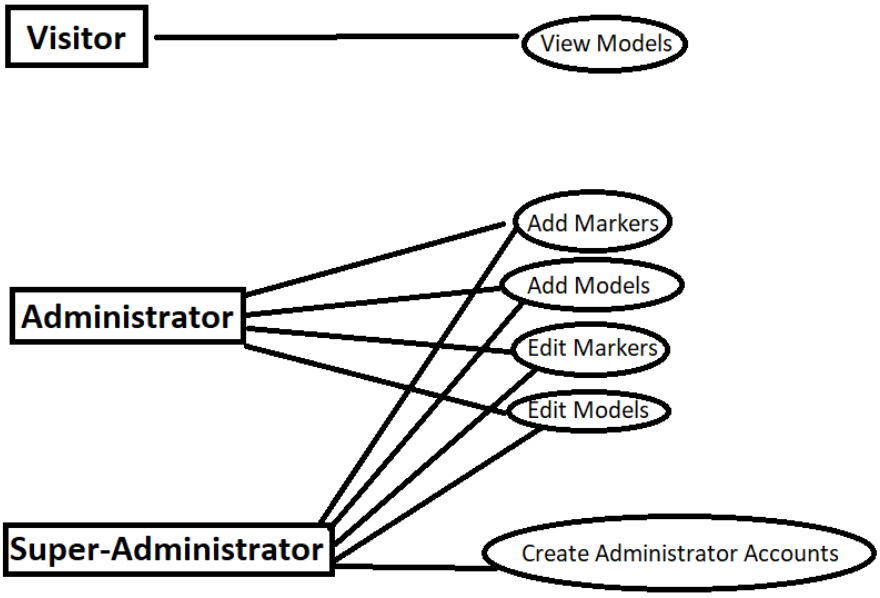
4. Save the object

5. Exit the edit modal.

2.1.3 Hardware Interfaces

Any device with (a) a web browser, (b) a screen with a minimum resolution of 360 x 740, (c) a camera, and (d) a gyroscope and accelerometer should function.

2.2 PRODUCT FUNCTIONS



These use cases describe the steps that a student, adviser, or administrator would need to take in order to accomplish some common/basic tasks, such as creating an account or designing a new 4-year plan.

2.2.1 UC-0

A visitor to the garden wants to see AR models. (User Goal / Visitor)

Main scenario:

1. User opens the application's website.
2. The user arrives at the calibration page.
3. The user gives the website permission to access the phone's camera, gyroscope, and accelerometer.
4. The user confirms that the application is calibrated and working correctly.
5. The user uses the application as long as they wish – the application will automatically adjust to new NFT markers.

Extensions:

1. The user wishes to see the plants from a previous year.
    a. The user completes steps 1-4 of the main scenario.
    b. The user presses the user options menu button.
    c. The user selects the desired season or event to view from the dropdown.

d. The user returns to the main user screen.
e. The user returns to the main scenario.

2.2.2 UC-1

An administrator wishes to add a new NFT marker for AR models to appear at to the application.

(User Goal / Administrator, Super-Administrator)

Main scenario:

1. The administrator opens the administrator website.
2. The administrator logs in with username and password.
3. The administrator clicks on the "Markers" tab on the main page.
4. The administrator clicks the plus icon in the bottom right-hand corner.
5. The administrator fills out the necessary information.

Extensions:

1. An administrator wishes to edit a currently existing NFT marker.
   a. The administrator completes steps 1-3 of the main scenario.
   b. The administrator clicks on the box with the NFT marker's name and thumbnail.
   c. The administrator edits the necessary information.
2. An administrator wishes to add a new AR model to the application.
   a. The administrator completes steps 1 and 2 of the main scenario.
   b. The administrator clicks on the "Models" tab on the main page.
   c. The administrator completes steps 4 and 5 of the main scenario – the Markers and Models pages are similar in format.
3. An administrator wishes to edit a currently existing AR model.
   a. The administrator completes steps 1 and 2 of the main scenario.
   b. The administrator clicks on the "Models" tab on the main page.
   c. The administrator clicks on the box with the AR model's name and thumbnail.
   d. The administrator edits the necessary information.

2.2.3 UC-2

A super-administrator wishes to add a new administrator account.

(User Goal / Super-Administrator)

Main scenario:

1. The super-administrator opens the administrator website.
2. The super-administrator logs in with username and password.
3. The super-administrator clicks on the "Add Administrator" tab on the main page – this tab only appears for the super-administrator.
4. The super-administrator sets the username and initial password for the new administrator.

2.3 USER CHARACTERISTICS

End users are expected to use the application only for the duration of their visit to the Gardens. Each individual's usage will be infrequent, but as a group usage should be considered continuous while the Gardens are open.

Expected administrator characteristics is to add most NFT markers at the beginning, with new NFT markers and AR models being added sporadically for the lifetime of the application. Only one administrator is expected to be online at any time.

2.4 CONSTRAINTS

Time constraints prevent the development of our own AR framework. As such, we must use a preexisting AR framework. There are very few such frameworks, which forms the primary constraint on this project.

2.5 ASSUMPTIONS AND DEPENDENCIES

2.5.1 Hardware Assumptions

A phone with a working camera, accelerometer, and gyroscope is required; as well as a minimum screen resolution of 360 x 740. In addition, phones with outdated processors may not function correctly.

2.5.2 Software Dependencies

The following dependencies are required for the functioning of the applications.

- AR.js
- AFrame.js
- Koa.js
- Node.js
- MariaDB

3 Specific Requirements

3.1 FEATURES

3.1.1 Display AR Models on Top of NFT Markers

The application must display AR models on top of NFT markers. That is to say, as an NFT marker is a parsing of a specific location, the application must display an AR model on top of a prespecified location. The application must display this model in a reasonable amount of time, and must continuously track the location as long as it remains in frame. The application must not cause the AR model to "jump around".

3.1.2 Allow Multiple AR Models per NFT Marker

The application must allow administrators to add multiple AR models for each NFT marker. This can be used, for instance, to show previous year's displays as an AR model overlayed on the current year's display. The application must allow administrators to select a default AR model for each NFT marker. The application must allow users to select non-default AR models for each NFT marker, if they exist.

3.2 PERFORMANCE REQUIREMENTS

The application must run at an acceptable frame rate on modern phones: at least 20 frames per second on devices no older than 5 years old. The application backend must support 100s of visitors at a given time.

3.3 DESIGN CONSTRAINTS

The application must run on a phone, which limits both processing power and screen real estate. Specifically, the application must be legible on a screen at least as small as 360 x 740 pixels.

3.4 SOFTWARE SYSTEM ATTRIBUTES (NON-FUNCTIONAL REQUIREMENTS)

3.4.1 Reliability

The application should always display the correct AR model for the given NFT marker and user settings. The application should always identify an NFT marker if present. Should an error occur, it should be handled gracefully: the application should not display corrupted graphics, the application

should not display -AR models not designed for the current NFT marker, and the application should not crash.

3.4.2 Availability

The application should always be available while the Gardens are open. The application back-end is designed to not require server downtime for maintenance. However, if such downtime is required, it should never need to occur while the Gardens are open to visitors. Failure to retrieve one AR model should not impact the availability of any NFT marker other than the one the model is associated with.

3.4.3 Security

Only the super-administrator (singular) should be able to create new administrator accounts. Administrator accounts must be password-protected. Cookies used to authenticate identity after the login page must be revokable and must not contain account information such as username, user ID, or password. Only administrators and the super-administrator should be able to add and edit NFT markers and AR models.

3.4.4 Maintainability

The application backend must support the ability to add NFT markers and AR models without taking the application offline for maintenance.

3.4.5 Portability

It is not necessary that the application be able to be deployed portably. However, as application development is on a server other than Reiman Gardens', some portability is required.