Logo

Description automatically generated

Technical Design Document

University of Maryland Global Campus

SWEN 670 – Team A

Spring 2023

Version 3.0

April 4, 2023

Document Control

Document Information

|  |  |
| --- | --- |
| © | Information |
| Document Id | ViroTour-IDE-TDD |
| Document Owner | UMGC SWEN 670 TEAM A |
| Issue Date | February 12, 2023 |
| Last Saved Date | April 4, 2023 |
| File Name | ViroTour\_TDD\_Team\_A.docx |

Document History

|  |  |  |
| --- | --- | --- |
| Version | Issue Date | Changes |
| 0.1 | 2/8/2023 | Initial Draft |
| 1.0 | 2/11/2023 | Milestone 2 Submission |
| 2.0 | 3/26/2023 | Milestone 3 Re-Submission |
| 3.0 | 4/4/2023 | Milestone 4 Re-Submission |

Contents

[1. Introduction 1](#_Toc131523010)

[1.1 Purpose 1](#_Toc131523011)

[1.2 Intended Audience 1](#_Toc131523012)

[1.3 Project Documents 1](#_Toc131523013)

[1.4 Overview 2](#_Toc131523014)

[1.5 Scope 2](#_Toc131523015)

[1.5.1 In Scope 2](#_Toc131523016)

[1.5.2 Out of Scope 2](#_Toc131523017)

[1.6 Definitions, Acronyms, and Abbreviations 3](#_Toc131523018)

[2. System Overview 3](#_Toc131523019)

[3. System Architecture 4](#_Toc131523020)

[3.1 Architectural Design 4](#_Toc131523021)

[3.2 Decomposition Description 5](#_Toc131523022)

[3.2.1 Tour Navigation Decomposition 5](#_Toc131523023)

[3.2.2 View Customization Decomposition 6](#_Toc131523024)

[3.2.3 Hotspot Customization Decomposition 6](#_Toc131523025)

[4. Data Design 7](#_Toc131523026)

[4.1 Application Data Handling 7](#_Toc131523027)

[4.2 Third Party Data Components 8](#_Toc131523028)

[4.2.1 JSON/JPEG Request 8](#_Toc131523029)

[4.2.2 JSON/JPEG Response 9](#_Toc131523030)

[4.3 Application Data Model 12](#_Toc131523031)

[5. Component Design 12](#_Toc131523032)

[5.1 Tour Navigation 13](#_Toc131523033)

[5.2 View Customization 14](#_Toc131523034)

[6. Human Interface Design 16](#_Toc131523035)

[6.1 User Flow 16](#_Toc131523036)

[6.1.1 User Main Screen 17](#_Toc131523037)

[6.2 Screen Images 20](#_Toc131523044)

[6.2.1 ViroTour User Main Screen 20](#_Toc131523045)

[6.2.2 Application Settings Screen 25](#_Toc131523064)

[7. Requirement Matrix 26](#_Toc131523065)

# Introduction

The Technical Design Document (TDD) for ViroTour is a comprehensive description of the application's design and architecture. This document outlines the scope of the application, including its intended functionality and use cases. It also covers the architectural, data, component, and human interface designs of the application. In addition, a supplementary document called the software test plan will be included to ensure the quality and reliability of the application.

## Purpose

The purpose of this document is to provide a detailed description of the design and architecture of the ViroTour application. The TDD outlines the technical details of the application, including its architecture, data models, and component designs. The document also describes the user interface design and that will be used to ensure the application's quality and reliability. By following this approach, we aim to create an application that is robust, reliable, and meets the needs of our users.

## Intended Audience

This document will be used by Software Developers/Testers, Project Managers, and Product Owners as a reference when implementing the design and for conveying the design to the ViroTour technology team.

## Project Documents

This Technical Design Document is part of a set of documents created to aid in developing the ViroTour and to provide artifacts with vital information for the application’s ongoing support and operation throughout its life cycle.

**The following documents are included in the entire documentation package:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Document** | **Version** | **Date** |
| 1 | Project Management Plan (PMP) | 2.0 | 2/12/2023 |
| 2 | Software Requirements Specification (SRS) | 2.0 | 2/12/2023 |
| 3 | Technical Design Document (TDD) | 1.0 | 2/12/2023 |
| 4 | Software Test Plan (STP) | 1.0 | 2/12/2023 |
| 5 | Programmers Guide (PG) | 1.0 | 3/25/2023 |
| 6 | Deployment and Operations (DepOps) | 1.0 | 3/25/2023 |
| 7 | User Guide (UG) | 1.0 | 4/1/2023 |
| 8 | Test Report (TR) | 1.0 | 4/1/2023 |

**Table 1.3 – Project Documents**

## Overview

The purpose of the ViroTour is to allow a user with a series of panoramic images to create a virtual tour by linking the images together so that a virtual tour of the place is available for others to visit the place virtually. . This tour can be enhanced by the addition of hotspots which will give additional information to what is seen in a given image. The photographs will be stored on a server and accessible via web, android, or apple for anyone with a working internet connection. The user experience will be similar regardless of the platform and intuitive with little to no training needed. The user will be able to quickly create virtual tours without having to manually indicate how locations should be connected together.

## Scope

The scope of this document is to provide a technical description of the design and architecture of the ViroTour App for users without advanced technical abilities. This document is broken down into sections that will detail the aspects of the application. The in-scope areas that this document will address are the following:

### In Scope

* + **Architectural Design**: This section describes the system architecture of the ViroTour application. It shows the flow of data. The application is built following the industry proven three-tier application architecture. Team A will focus on the presentation tier of the application.
  + **Data Design**: This section will detail how data is utilized by the application. This includes where it is coming from and being sent to, where it is being stored, and what type of data the system will utilize for the app to function.
  + **Component Design**: This section will describe the components of the application that make the application provide the intended features, including components implemented within the application and third-party resources utilized.
  + **Human Interface Design**: This section will detail the functionality of the human interface design of the ViroTour Application and how it is intended to interact with the user.

### Out of Scope

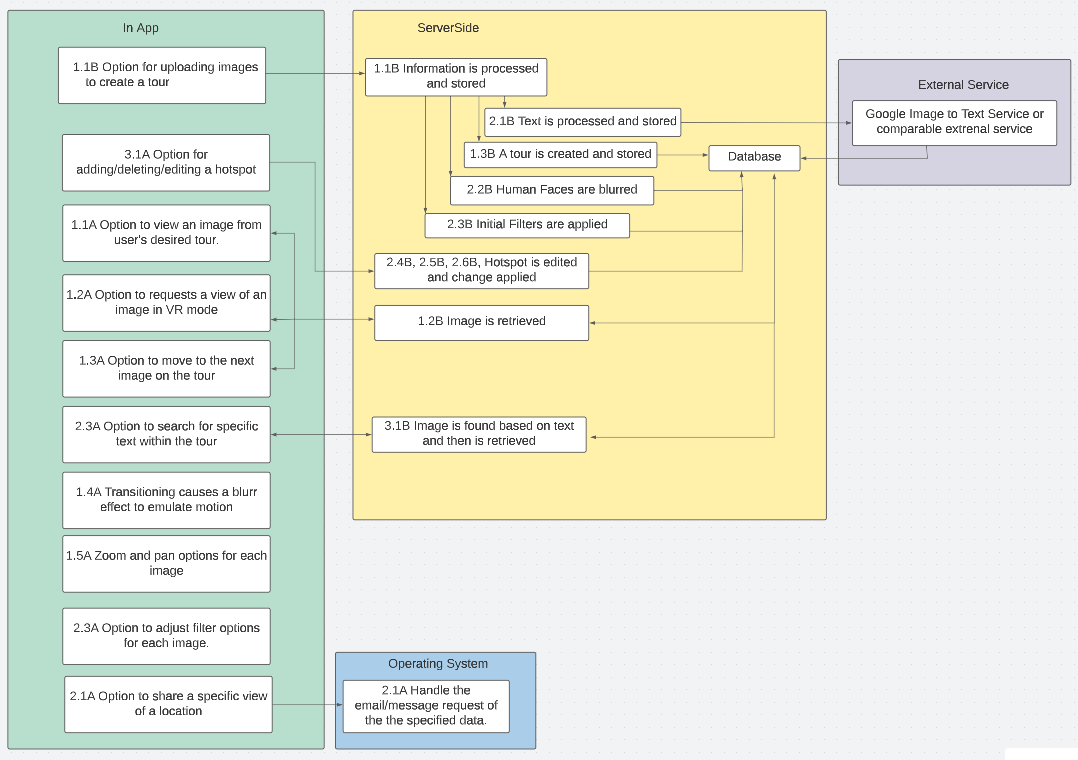
* + **Accessibility**: All accessibility features, including voice input and screen reading, are not within the scope of Team A’s requirements. All accessibility features will be covered by Team B’s documentation and/or are provided natively by the Operating System.
  + **Application and Data Tiers:** Any description of how the server will store or process the images for ViroTour is out of scope and will be covered by Team B.
  + **Image to Text Processing:** This functionality will be external to the application via Google Vision and is thus out of scope.
  + **User Authentication:** ViroTour will allow all users to access all functions and feature below. User authentication functionality is recommended as a follow-on feature and a distinction between Admin and user accounts.

## Definitions, Acronyms, and Abbreviations

* API - Application Program Interface
* Dart - an object-oriented client-optimized programming language used to build mobile and web applications on various platforms. It has easy-to-use applications that can be utilized to work on both the user and server end.
* DepOps - Deployment and Operations
* Flutter - an open-source UI software development kit created by Google
* GB - Gigabyte
* HTTP - Hypertext Transfer Protocol - is the foundation of the World Wide Web, and is used to load webpages using hypertext links. HTTP is an application layer protocol designed to transfer information between networked devices and runs on top of other layers of the network protocol stack.
* HTTPS - A combination of the Hypertext Transfer Protocol (HTTP) with the Secure Socket Layer (SSL)/Transport Layer Security (TLS) protocol. TLS is an authentication and security protocol widely implemented in browsers and Web servers.
* Hotspots - transition points, as in points that indicate a direction of a new image
* Hotspot Information Points – points of interest, information in the form of text and potentially a picture that is docked to a point in a picture (also listed as informational\_hotspots)
* iOS - iPhone Operating System
* JSON - (JavaScript Object Notation) is an open standard file format and data interchange format that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and arrays.
* OS - Operating System
* SRS – Software Requirement Specification
* STP – Software Test Plan
* TDD – Technical Design Documentation
* UMGC – University of Maryland Global Campus
* URL - Uniform Resource Locators
* VR – Virtual Reality

# System Overview

ViroTour is intended to be an application which can be utilized by anyone wishing to create a virtual tour of a physical space with the capability to access and share specific locations within the tour as well as search for words which may appear somewhere in the real-world location. The ViroTour application is meant to run on iOS/Android as well as via a web portal and should be easy to navigate utilizing a style familiar to many other modern applications. The below diagram outlines how each requirement will be handled within the system as a whole and how the subsystems will be related to each other.

**Figure 2.1 - System Requirement Overview**

To achieve the desired functionality, the software interactions will be broken up into in-app, operating system and server-based actions for each requirement outlined. For text processing an external resource will be required. The team has selected Google Cloud Vision due to its open-source nature making it a good fit for the project. See Requirement Matrix in section 7 for more details on the requirements themselves. The project will be completed by two teams. Team A will handle the application side (frontend) while Team B handles the server-side processing (backend).

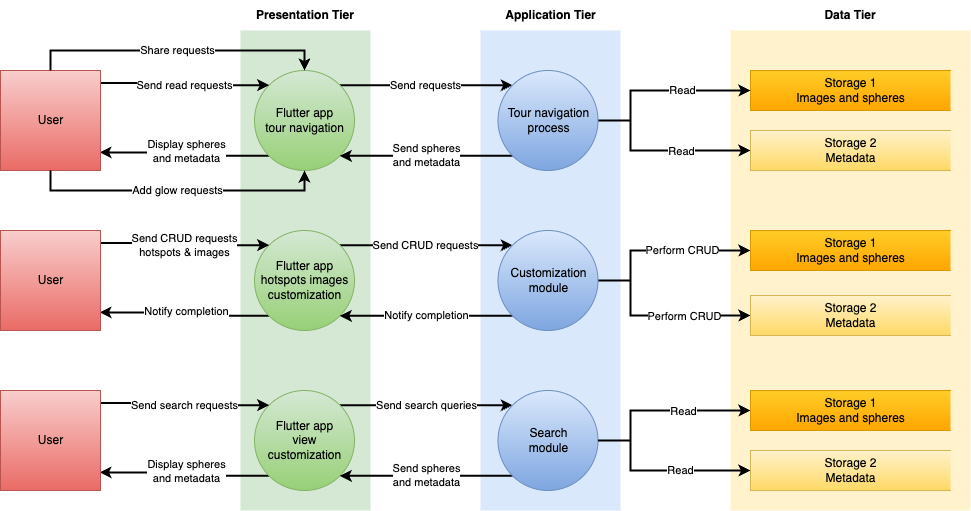
# System Architecture

## Architectural Design

This section describes the system architecture of the ViroTour application. It shows the flow of data. The application is built following the industry proven three-tier application architecture. Team A will focus on the presentation tier of the application. Team B will focus on the application and the data tier of the application. The application tier includes several processes. One process is used to process images uploaded by users. This process converts 2D images into a 360-degree image which creates an illusion of a sphere. It also connects several spheres to form a tour. Another process is used to handle the tour navigation that allows users to move to different locations in a tour. A third process enables users to search for any available text in a tour and returns the list of objects that contain the searched text.

The presentation tier of the application is built with Flutter - an open-source UI software development kit created by Google using Dart - a programming language designed for client development developed by Google.

The application is deployed on Google Play Store and Apple’s App Store. The Flutter application will interact with a number of services to provide the functionality for both the client-side and the server-side. The application leverages the features provided by Flutter and Google Cloud Platform to store and serve images.



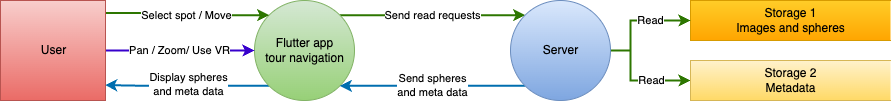
**Figure 3.1 - System Architecture Diagram**

## Decomposition Description

When the users open the application on their device, they will see a list of available tours. There are steps that need to occur for the app to perform its functionalities. All the steps are grouped into three main categories which are described below.

### Tour Navigation Decomposition

When the users open the ViroTour application on their mobile devices, they see the list of available tours. If they select a tour, they will see the first sphere of the tour. From the first sphere, they can move to adjacent spheres. The transition between each sphere is smoothened by the application to provide a better user experience. Within each sphere, the users can see different points of interest with provided text. They can pan and zoom to see the details within the sphere. There is also an option for the users to use their VR viewer to see the tour.

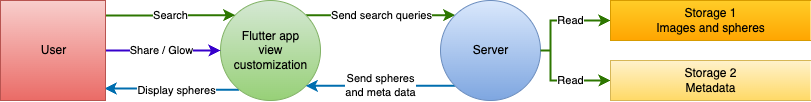


**Figure 3.2.1 - Tour Navigation Decomposition**

### View Customization Decomposition

The users can also use the search bar to search for any text. If the text is available in the tour, the server will handle the request and respond back with a list of objects that contain the searched text. The users then can choose to go back to the tour or select an object. If they select an object, the application will move the view to the sphere and the angle that contains the object.

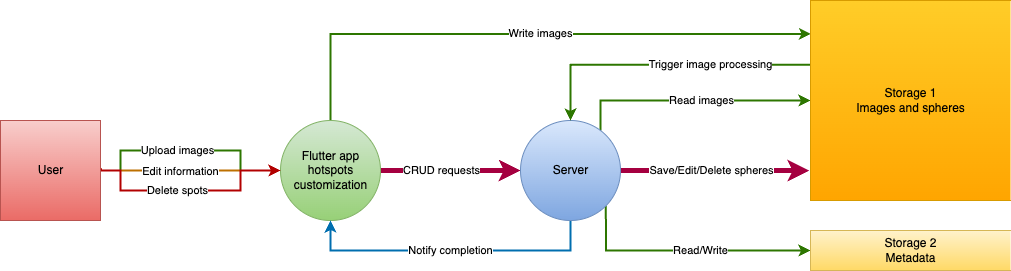
The users also have the option to adjust the birghtness and contrast of images within the tour. This option is intended to be used on images that are difficult to see by default.



**Figure 3.2.2 - View Customization Decomposition**

### Hotspot Customization Decomposition

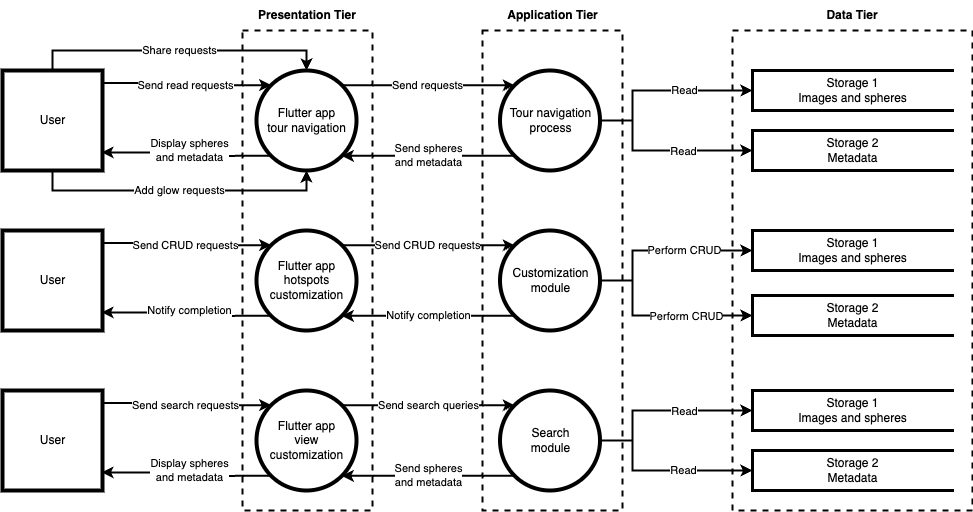
The application provides users with the option to upload images. After all the images have been uploaded successfully, the server will automatically create a tour out of the uploaded images. The users will be notified when the tour is ready to view. As the creation of a tour, a user can edit the information, object details, add or remove any spots within the tour.



**Figure 3.2.3 - Hotspot Customization Decomposition**

# Data Design

Data for the application will be transmitted over the internet, stored on the user’s mobile device, and sent to web services. The user can perform some app functions without an internet connection, but for full functionality, the application requires an internet connection.



**Figure 4 – ViroTour Data Flow Diagram**

## Application Data Handling

ViroTour’s frontend and backend will communicate and exchange data to provide users a seamless experience. The frontend will be responsible for displaying the user interface and capturing user inputs, which the backend will be responsible for processing and returning the expected output. The frontend will send HTTP/S requests to the backend, which will contain data including user inputs/desired actions. To ensure that data sent between the frontend and backend are uniform, multiple validation checks will occur before the data is sent between the systems. These checks will create uniform data objects and help mitigate any possible security threats, such as scripting attacks or data injection. No sensitive data will be stored on ViroTour’s database, which means no encryption will be necessary.

Data shall be transmitted and received using JavaScript Object Notation (JSON). JSON is a simple format for storing and transmitting key-value data pairs. This project has chosen to use JSON because it is lightweight, efficient, human-readable, cross-platform compatible, easy to parse, and can support complex data structures. Flutter has integral functions for converting JSON to application objects. JSON has simple syntax rules. Data is in key value pairs and separated by commas. Curly braces define objects, and square brackets represent arrays.

## Third Party Data Components

Communication between the ViroTour Frontend applications and the backend servers will be relayed through a web interface using the HTTP protocol. HTTP requests and responses to the APIs will consist of the transmission of data back and forth in form of base64 encoding for images and tours and JSON data for all textual content.

### JSON/JPEG Request

There will be two kinds of requests occurring within the app. One request will be to made by the user creating the virtual tour and will involve the tour name, location, date recorded, description and a series of image which are meant to be stitched together into the virtual tour. The images will be encoded as base64. The images will come in groups of four where each four images represent a singular real-world location. The server will automatically stitch these together into a singular base64 image and form a logical tour through the resulting images.

**4.2.1.1 Example tour creation request:**

{

“name”: “WWII air museum”,

“description”: “A virtual tour of the New Orleans World War II Museum “,

“images” {[

“image\_1”: “<image URL>”,

“image\_2”: “<image URL>”,

“image\_3”: “<image URL>”,

“image\_4”: “<image URL>”,

…

]}

}

|  |  |  |
| --- | --- | --- |
| **Object** | **Attribute** | **Description** |
| tourObject | tour\_name | A name given to the tour. |
|  | description | This is the descriptive information of the tour. |
|  | images | This is an array of base64 images that are sent over to the server to be stitched. |
|  | tour | A single image made up of multiple stitched images that is a response made by the server. |

The second kind of request comes from a user requesting to view a specific tour and will consist of the tour name and ID of the location they want to view.

**4.2.1.2 Example tour view request:**

{

“name:” “WWII air museum”,

“id:” “23”

}

|  |  |  |
| --- | --- | --- |
| **Object** | **Attribute** | **Description** |
| viewTourObject | name | A name given to the tour. |
|  | id | The internal ID number used to identify specific images within the specific tour. |

### JSON/JPEG Response

The server response will be through HTTP in the form of a JSON object. The object will hold the information for a singular location within the tour as a base64 image along with any transition points and hotspot data.

**4.2.2.1 Example tour view response:**

{

"name": "Virtual Tour 03\_19\_10\_20\_39",

"description": "test",

"id": 1,

"locations": [

{

"location\_id": 1,

"neighbors": [

{

"location\_id": 2,

"x": 4347,

"y": 1515

}

],

"pano\_file\_path": "panoramic\_images/T\_1\_L\_1\_pano\_blurred.jpg"

},

{

"location\_id": 2,

"neighbors": [

{

"location\_id": 1,

"x": 342,

"y": 297

}

],

"pano\_file\_path": "panoramic\_images/T\_1\_L\_2\_pano\_blurred.jpg"

}

],

"text\_matches": [

{

"content": "text match 1",

"location\_id": 1,

"x": 5032,

"y": 1451

},

{

"content": "text match 2",

"location\_id": 2,

"x": 4852,

"y": 1689

}

]

}

|  |  |  |
| --- | --- | --- |
| **Object** | **Attribute** | **Description** |
| locationObject | name | A name given to the tour. |
|  | description | This is the descriptive information of the tour. |
|  | pano\_file\_path | The singular image representing the user’s current location within the tour. This includes the state of the image (any postprocessing that was done on the image) and the image URL. |
| transitional\_  hotspots | locations | Locations that are available to be navigated to from this transitional hotspot |
| informational\_hotspots | position, y | The x and y coordinates where the transition point will be added to the image client side as an interactable button (clicking on it will take the user to the image with the location id specified above). |
|  | content | The content associated with the informational hotspot. This includes the hotspot name, position (x, y, and z coordinates), description, hotspot\_image, timestamp, and creator’s name. |

## Application Data Model

JavaScript Object notation (JSON) is the data encoding standard to be used when passing information between the server and applications. All images are converted to text based base64 data encoding before they are sent over a network. The JSON attributes will be the tour name, location, date recorded, description, the single image of where the user currently is, the list of transition points to be placed within that image as coordinates paired with which image they should link to and an array of hotspots if any are present. The single image will be displayed and transition icons will be added client-side to the specified coordinates and will be interactable buttons allowing the user to move throughout the tour. The user selecting one will trigger another request to the server which will then fetch the next image in the tour as necessary.

# Component Design

This section describes what each component does systematically. Each bullet point shows a primary class in the system and its detailed description and capabilities. Pseudocode will be provided to explain more complex functions as needed.

The design will provide more flexibility in development and more concrete test suites in test and bug fixing in maintenance. The application is divided into components by features. That way, each segment is abstracted from the main functionality.

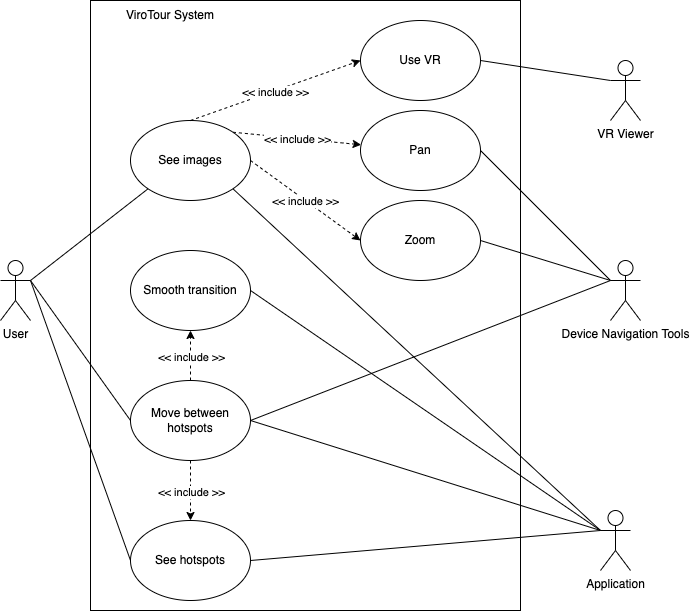
The application consists of three tiers. The presentation tier will allow users to interact with the application via their device’s screen. The application tier will handle the API requests from the users via the presentation tier. This tier will also manage all the business logic and processes of the application. The application tier accepts and returns data in a JSON format. The third tier is the data tier. This tier stores the uploaded images, the processed images or spheres of the tours, and the tours’ metadata.

The Flutter application will start with a main Dart file with a void main( ) function which contains information to start the application. Additional UI, management, and model files will be provided with corresponding files and class names.

## Tour Navigation

The User Interface will include graphical screens designed for a touchscreen mobile device. This allows the user to display the data for each process. The Tour Navigation Component is the primary component that users interact with when they scan the location. The navigation component of ViroTour refers to the features that allow users to move around the tour’s virtual environment and access different images throughout the tour. The navigation component has multiple sub-components: hotspots, navigation controls, zoom controls, and auto-play mode. All of the sub-components work together to create a seamless experience for the user, allowing them to move about the virtual space freely and at their own pace.

Hotspot identifies areas in a panoramic image for interactions, navigational or actions activation purposes. The ViroTour will take the user to the hotspot area by clicking the hotspot locator. The linkage process connects view orientation to the next nearest panoramic scenes. The links are associated with a hotspot, so the user can activate the link by pressing on the hotspot. During the stitching process, in addition to creating a connection between the panoramic images, the transition effect of the actual scene to the hotspot can also be added.

  
**Figure 5.1 - Tour Navigation**

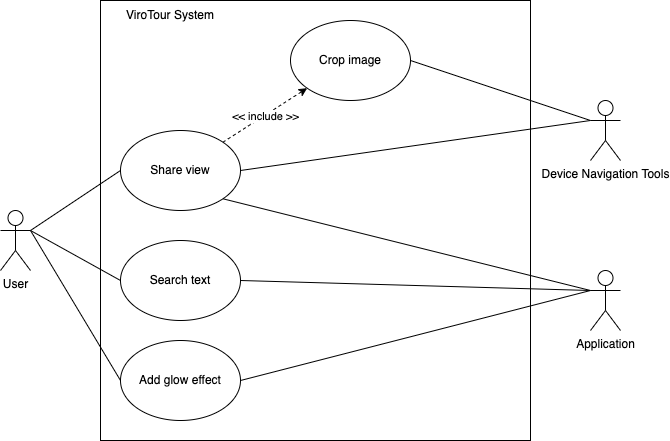
|  |  |
| --- | --- |
| **View** | **Description** |
| Main Menu (Home) | Main page for routing the user to application functions. Options to Create, Go to Tour Page, Search Text |
| **Create Tour** | |
| Create Tour - Add New Tour (Record feature) | Click ‘+’ to add new tour. Add Tour Name, Location Details (Address, Date recorded, etc.). When done, click ‘Add Tour’. After this, next screen will prompt/start camera view. |
| Transition of hotspot (Record feature) | After done, move to next spot 5-10 feet away from the last spot and repeat the process throughout the building. |
| **View/Edit/Publish** | |
| Search | Search Box allows looking up tour name that contains a string. |
| Select a tour for viewing | The screen displays a list of completed tours. The user selects one tour to start viewing. |
| Select viewing mode: as a sphere from hotspot (UC 1.1A) or using a VR device (UC 1.2A) | The screen displays viewing options. |
| VR viewer | A physical device allows user to see images in VR mode |
| Start a tour | Display 3D rendered images taken from the first hotspot. |
| Experience the tour (UC 1.3A) | In the sphere, the user can see and click on the link for the object information, add glow effect (refer to Module 2. View Customization – 2.2A and 2.3A), and the previous or next hotspot (1.3A). |
| The transition between each hotspot is smooth (UC 1.4A) | After clicking on the immediate previous/next hotspot, the view toggles from the current location to the chosen location. The movement of the transition between two locations should be smooth and feel like the human eyes are actually moving (in faster speed) within the space. |
| Device Build-in Navigation Tools - Zoom and pan at each hotspot (UC 1.5A) | Enable click / touch and type.  The user can zoom and pan what they are viewing using a little ruler at the corner of the screen or adjusting with 2 fingers. |

**Table 5.1 - User Views and Description**

Application data and functions will be handled by backend classes and processes, allowing the user interface to be reconstructed as needed when application states change.

## View Customization

The interface design for ViroTour will include customization capability of the user interface to search for a text that is available in the tour and to add glow effect to the view for better visibility and experience of the tour.

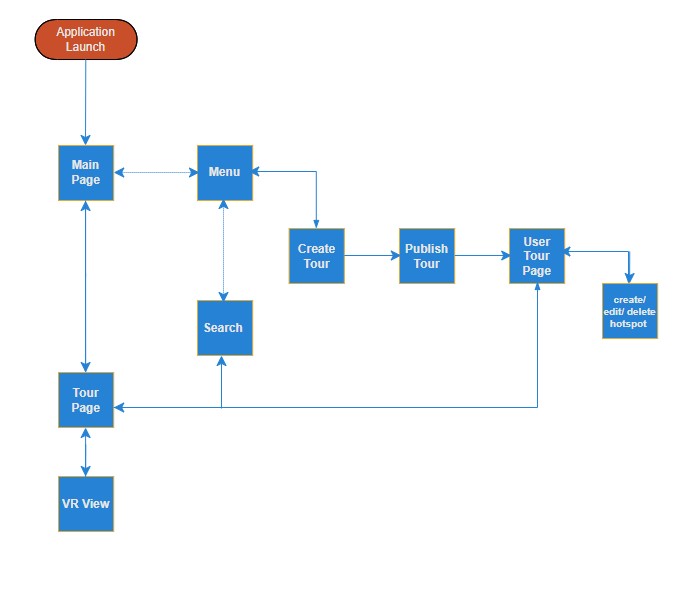
  
**Figure 5.2 - View Customization**

|  |  |
| --- | --- |
| **View** | **Description** |
| Start a tour | Display 3D rendered images taken from the first hotspot. |
| **Search for text that is available** | **User can customize the tour by searching a text in the tour to navigate to the location where the text is displayed in the virtual tour.** |
| Search bar | Search bar is always active from the first hotspot of the tour on the user interface. |
| Type and search text (UC 2.2A) | The search covers any major text that is available in the entire panoramic pictures of the tour. |
| Clicking on the search result | Viro Tour navigates to the nearest hotspot where the text is located. The user can start the tour from that hotspot and move to the next. |
| Zoom and pane on text (UC 1.5A) | Once the user finds the actual text on the tour anywhere in the panoramic image, the user can zoom and pane to read the actual text from the image itself. |
| **Add glow effect to the view** | **The user can customize the tour lighting effect by activating the glow effect functionality on the interface.** |
| Glow effect Slider (UC 2.3A) | Upon starting a tour on the first hotspot, the Glow Effect slider is visible but inactive. |
| Default setting | The default setting of the slider is at 0%. |
| Clicking on the Glow Effect slider | The slider will be bold with the option to drag to increase or decrease the tour lighting. |
|  |  |

# Human Interface Design

## User Flow

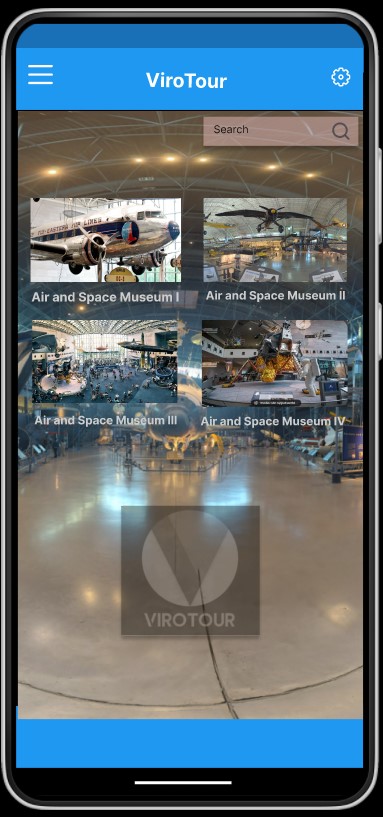
The users visit ViroTour to create and view a virtual tour created using a series of panoramic images. The three separate modules allow the users to navigate through the virtual tour, customize and share their views with the ease of hotspot customization. When a user opens the app, they see the main page which contains all the published tours ready for navigation. From the main page, user can also go to the hamburger menu at the left top corner to select from other options including Create Tour, Edit Tour, and Publish Tour. During the viewing experience, the user can zoom in and out within a space, move from one hotspot to the immediate previous and next one, adjust filters for better lighting and search for texts that are available in the tour. From a Tour Page, user can enable VR to go to VR Page.



**Figure 6.1 - Application User Flow**

### User Main Screen

The user dashboard screen is the screen that users see when they launch the application. The user dashboard displays the list of tours that user can choose from. From this screen, users can also access menu at the top left-hand side and application settings using the gear icon at the top right corner.

**Figure 6.1.1 ViroTour Main Screen and Hamburger Menu (Right)**

**6.1.2 Touring Screen**

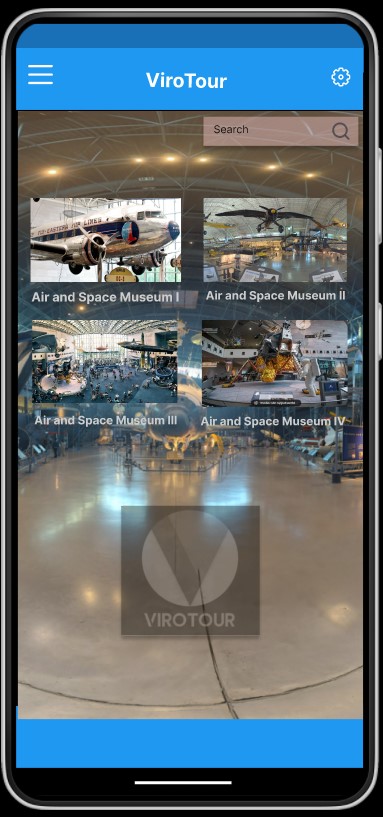
To navigate tours available on ViroTour, users need to select a tour and get to the view tour screen. There are two options to get to this screen, user can either go to the hamburger menu at the left top corner or select a tour directly from the user main screen (Figure 6.1.1). In the navigate tour screen below, the application shows the selected tour with full screen on the device. The navigation screen provides user with hotspots to easily navigate the tour. User can also pan by using finger or mouse and use the zoom in and zoom out options as needed. This screen also avails control options towards the top right to switch to VR view, edit hotspots, delete hotspots, and share location and image.



**Figure 6.1.2 Touring Screen**

**6.1.3 Text Search Screen**

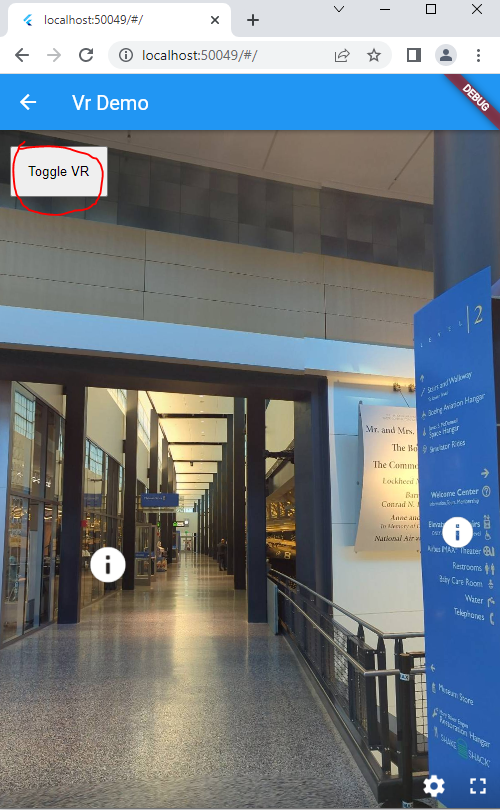
In the following user interface, the search option remains at the top of the interface and is available for the user to refine the results. The returned results are displayed chronologically below the search pane in descending order.



**Figure 6.1.3 Text Search Screen**

**6.1.5 VR View**

This view, when switched to, allows the user to view images in a 3 - dimensional environment which enables the user to explore and interact with a virtual surrounding in a way that approximates reality as perceived through the sense of the user. In the screen below, the VR view button is the first icon at the top right-hand corner of the interface.



**Figure 6.1.5 VR View**

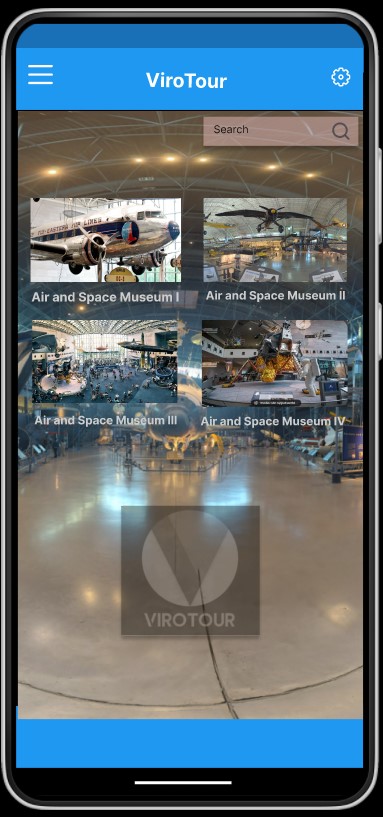
## Screen Images

The following mock-up designs are each accompanied by an explanation of the user experience when interacting with that component.

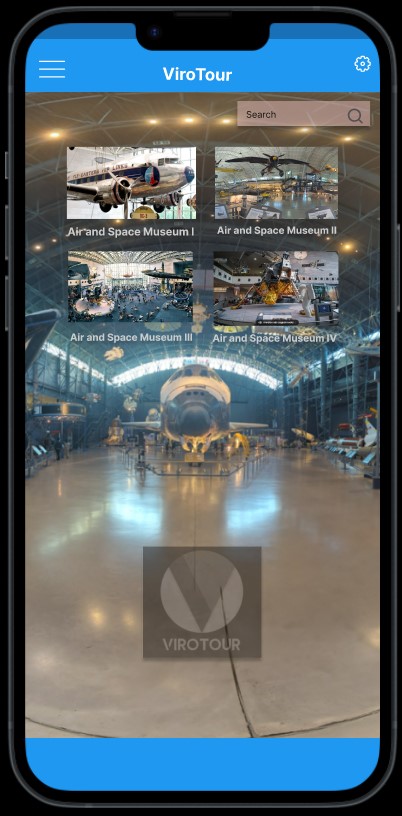
### ViroTour User Main Screen

When the application is launched, the tours available to choose from are displayed for the user in the main screen. User can select and start navigating tours or they can go to the menu to decide what to do by selecting options between view tour, create tour, edit tour or publish tour. At the top right corner of the screen, user can access application settings by clicking on the gear.

Since ViroTour follows a cross platform design approach, the main scree design reflects the design for Android, iOS and Web. However, the rest of the screen designs are based on Android device.



**Figure 6.2.1a ViroTour User Main Screen (Android)**



**Figure 6.2.1b ViroTour User Main Screen (iOS)**



**Figure 6.2.1 c ViroTour User Main Screen (Web)**

#### Navigate Tour

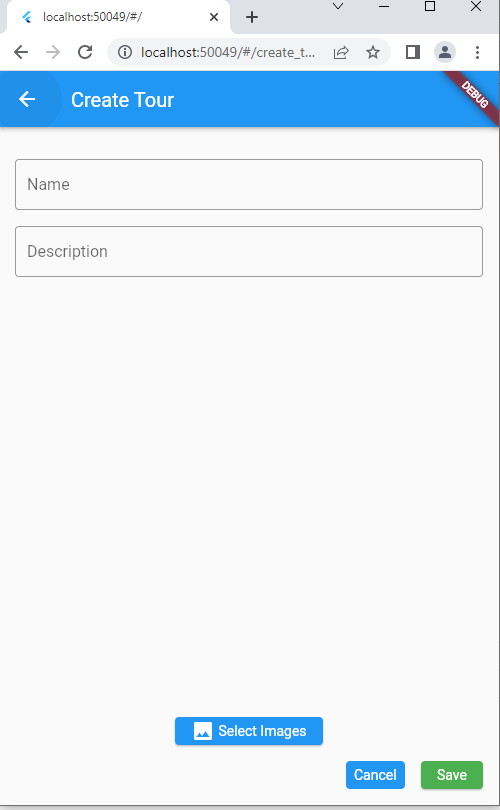
When user starts navigating tour, as it is shown below, they will be provided with navigation options. The options include hotspots, zoom in and zoom out, towards the top right side of the screen additions will also be provided including VR view, share, edit (hotspots).



**Figure 6.2.1.1 Navigate Tour**

#### Create Tour, add Metadata

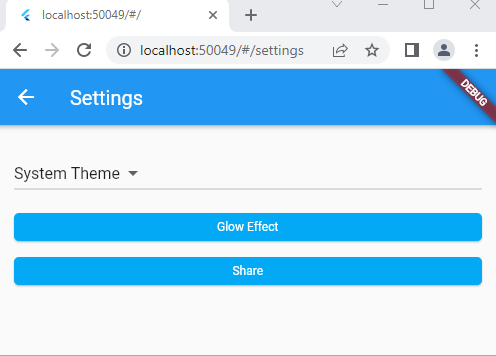
User selects the create tour from main menu to add new tour. The form to add tour information will be displayed in the screen requesting users to add Tour Name, Short description and images that comprise the tour.. When user is done entering the data, they select "save" to submit the information.



**Figure 6.2.1.2 Create Tour, Add Metadata**

### Application Settings Screen

In the gear menu located at the top right corner of the screen the user is able to access the glow effect, system theme and share location options.



**Figure 6.2.5 Application Settings**

# Requirement Matrix

The requirement matrix below outlines all identified requirements as either optional or mandatory and further identifies which team will oversee implementing said requirement.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirement Description** | **Category** | **Team** |
|  | Core |  |  |
| 1.1B | Uploads images to server – User can upload a series of images to the server which will later be processed into the virtual tour. The images should be in jpeg format. | Mandatory | B |
| 1.2B | Panoramic image created – The server will automatically process the pictures into a “stitched together” representation of the real physical space with transition points assigned among pictures which represent adjacent locations. | Mandatory | B |
| 1.3B | Determine transition points – The server will automatically assign an area to each picture which will be used as the transitional trigger when the user interacts with it via the input of the application which will trigger the next logical image in the tour to be displayed. | Mandatory | B |
|  | Editing |  |  |
| 2.1B | Extract text from images – Using an external service (Google Vision) extract all text available within the images provided and create a searchable database where words are stored along with the image location where they can be found, such that the user will be able to navigate easily to a specific piece of text within the environment. | Mandatory | B |
| 2.2B | Detect and blur human faces – When the server processes the images into a tour any human faces will be automatically identified and blurred to protect their privacy. | Mandatory | B |
| 2.3B | Apply filters – The images will have an automatic glow effect applied to them allowing for an easy adjustment to brightness and contrast within each image. | Mandatory | B |
|  | Search |  |  |
| 3.1B | Search text – Users have the option to search for a specific string of characters within the processed tour. They can quickly navigate to any location where the specified text exists. | Mandatory | B |
|  | Tour Navigation |  |  |
| 1.1A | View images in browser/application - Images from the tour are visible to users. | Mandatory | A |
| 1.2A | View images in VR view – Users can switch their preferred view to VR mode which will alter their view to google cardboard specifications for VR display | Mandatory | A |
| 1.3A | Move from one location to another – clicking on a transition point will carry the user to the next location in the tour. | Mandatory | A |
| 1.4A | Transition Between points is smooth – Changing from one location to another should be smooth and take no longer than 1 second to achieve with a blurring animation to make the motion feel more life-like. | Mandatory | A |
| 1.5A | Zoom and pan at each image – The user can zoom and pan around each image by using either their cursor or the standard “pinch-swipe” input method for most modern applications. | Mandatory | A |
|  | View Customization |  |  |
|  |  |  |  |
| 2.2A | Search for text available in the tour – User can search for a specific piece of text within the tour, and quickly navigate to all locations where said text appears. | Mandatory | A |
| 2.3A | Adjust the glow filter applied to the view – User can modify the filter to adjust brightness and contrast of the image. | Mandatory | A |

Table 7 Requirement Matrix

Appendix - A

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No:** | **Requirement** | **Component** | **Use Case ID** | **Use Case Name** | **Team A** | **Delivered** | **Partly Delivered** | **Not Delivered** |
| 1 | Use Matterport Axis device to automatically take images (See <https://buy.matterport.com/shop/camera/>) using an app written in Dart and Flutter and stitch the four images together to create a panoramic image. |  |  |  | Backend Requirement |  |  |  |
| 2 | Using the images, automatically find out where the transition hotspots be placed to transition from one image to another. |  |  |  | Backend Requirement |  |  |  |
| 3 | Use Google Vision (or some other tool such as Apache Tesseract) to read text from the images and note the location of the text to refer to it later. |  |  |  | Backend Requirement |  |  |  |
| 4 | In the app using Dart and Flutter, provide means for a user to get a tour of all the pictures put together via hotspots. | Tour Navigation | A-1.1 | View the images as a sphere from hotspots. | x | x |  |  |
| A-1.3 | Move from one hotspot to another. |
| 5 | Allow manual placement of hotspots as well as ability to delete and edit hotspots created earlier. | Hotspot Customization | A-3.1 | Create/Edit/Delete a tour by uploading images. | x |  | \*  Note 1 |  |
| 6 | Provide a facility to create, edit, and delete informational hotspots that display a blurb with possibly a smaller image about that item associated with it. |
| 7 | Allow ability to search for text read from images, display options of where that text appears in images, and take the user to where the selected image by the user such that the text appears in the middle of the view. | View Customization | A-2.2 | Search for text that is available in the tour. | x |  | \*  Note 2 |  |
| 8 | Provide ability to zoom and pan. | Tour Navigation | A-1.5 | Zoom and pan at each hotspot. | x | x |  |  |
| 9 | Allow the user to long press a point in an image to capture a convenient height and width clip around the point where it was long pressed, and then share with someone such that view that image clip and if the recipient selects the shared image, he or she can also view the image at the same location. | View Customization | A-2.1 | Long press share link | x |  |  | x |
| 10 | Allow methods (functions) to convert image coordinates between cartesian and spherical coordinates of the images. |  |  |  | Backend Requirement |  |  |  |
| 11 | Transitions from an image to the next should be animated and not abrupt. | Tour Navigation | A-1.4 | Zoom and pan at each hotspot. | x | x |  |  |
| 12 | Blur human faces and skin from all images |  |  |  | Backend Requirement |  |  |  |
| 13 | Provide option when processing images to apply the "Glow" filter along with reasonable parameters. This filter is used in Photoshop, TikTok, ImageMagick <http://www.fmwconcepts.com/imagemagick/glow/index.php>), and can be implemented using OpenCV (<https://stackoverflow.com/questions/68592934/implement-glow-filter-in-cv2-python>). It is very useful if the indoor lighting is inadequate. | View Customization | A-2.3 | Zoom and pan at each hotspot. | x | \*  Note 3 |  |  |
| 14 | Provide a VR view of the end-product    Resource:  <https://kuula.co/>  Kuula allows for manual placement of the hotspots and does not provide automatic placement like MatterPort does. Kuula does provide, like MatterPort, a VR view.   <https://www.capturingreality.com/> | Tour Navigation | A-1.2 | Enable VR View | x | \*  Note 3 |  |  |
| Functional Feature - 1 | Main Page: Tour List, Hamburger Menu, Wheel Menu | Application Navigation | N/A | N/A | x | x |  |  |
| Functional Feature - 2 | Search tours with result from the list of tours. | Application Navigation | N/A | N/A | x | x |  |  |

|  |  |
| --- | --- |
| **\* Notes Explained** | |
| **Note 1** | "Delivered Create/Edit/Delete Tour (Transitional Hotspots).    Create tour can select images on the device and redirect users to the list of tours after making a mock API call. The backend is not ready.    Did not deliver Create/Edit/Delete Informational Hotspots." |
| **Note 2** | Search informational hotspots with hard-coded URLs to objects. The search functionality works. Back-end API endpoints required. |
| **Note 3** | "VR view works as a standalone feature.  Glow effect only works on Chrome web browswer, not on iOS or Android." |