Software Requirements Specification

University of Maryland Global Campus
SWEN 670 : Team A
Spring 2023
Version 4.0

April 4, 2023

Document Control

Document History

Version	Issue Date	Changes
0.1	1/25/2023	Initial Draft.
1.0	1/28/2023	Milestone 1 Submission
2.0	2/12/2023	Milestone 2 Re-submission
3.0	3/25/2023	Milestone 3 Re-submission
4.0	4/4/2023	Milestone 4 Re-submission

1. Introduction

ViroTour is an application that will automatically generate virtual tours. The application will provide features to enable users to customize the virtual tours according to their needs.

1.1. Purpose

The purpose of this Software Requirements Specification (SRS) is to outline the functional and non-functional requirements for the ViroTour Application Development (VAD). Defining the requirements within this document ensures that the development team understands the intent of the application to meet the needs of the client. In addition, the SRS serves as a communication tool between development teams and the customer to establish a common agreement on how the application should function. This SRS shall describe the interactions between the application and the user in the form of use cases.

1.2. Scope

ViroTour is a standalone product that is independent of location or site. By making use of its built-in algorithms, the application provides users with the functionality to upload pictures and create virtual tour. On top of this, the ViroTour application delivers the capability of searching, editing and sharing.

The development effort is shared between two teams, Team A and Team B where the former focuses on the front-end development and the later on back-end development. In addition to making sure that the front-end development is ready to be integrated with the back-end development, the project gives emphasis on the navigation or the virtual tour interface implementation. To address the requirements for this project, in scope for Team A are:

- UI visual layout and design (regarding pages the user can route to within the application, and the components the user can interact with on screen)
- Capability to zoom and pan through the panorama images the application loads up
- Transitioning between images (when user clicks a hotspot) to be animated and not to be abrupt
- Functional and Performance Testing

1.2.1. Out of Scope

The elements out of scope for Team A are:

- All backend processing including database to store images, locations and hotspots
- API to processing based on user input
- User Authentication

1.3. Definitions, Acronyms and Abbreviations

The following are definitions, acronyms and abbreviations that might be mentioned in this document and need more clarity.

- API – Application Programming 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.
- Flutter - an open-source UI software development kit created by Google
- Informational Hotspots – points of interest, information in the form of text and potentially a picture that is docked to a point in a picture
- iOS – iPhone Operating System
- Lucid.app - a web-based diagramming application that allows users to visually collaborate on drawing, revising and sharing charts and diagrams
- Mbps - megabits per second
- SRS – Software Requirement Specification
- STP – Software Test Plan
- Transitional Hotspots - Images with their associated transition to new images.
- UMGC – University of Maryland Global Campus
- UML – Unified Modeling Language
- VAD - ViroTour Application Development
- VR – Virtual Reality
- VT – ViroTour

1.4. Project Documents

There are various documents created for this effort to provide the stakeholders, namely the project team, the client, and external users with sufficient information and understanding for the success of the project. These documents are summarized below.

The following documents are included in the entire documentation package:

	Document	Version	Date
1	Project Management Plan (PMP)	4.0	4/1/2023
2	Software Requirements Specification (SRS)	4.0	4/1/2023
3	Technical Design Document (TDD)	3.0	4/1/2023
4	Software Test Plan (STP)	3.0	4/1/2023
5	Programmers Guide (PG)	2.0	4/1/2023

6	Deployment and Operations (DepOps)	2.0	4/1/2023
7	User Guide (UG)	1.0	4/1/2023
8	Test Report (TR)	1.0	4/1/2023

Table 1.4 - Project Documents

1.5. References

- Assadullah, M. (2023). Software Engineering Project. Retrieved from: https://learn.umgc.edu/d2l/home/732302
- MacQuarrie, A. (2018) The Effect of Transition Type in Multi-View 360° Media. Retrieved from: https://www.researchgate.net/publication/322566741_The_Effect_of_Transition_Type_in__Multi-View_360_Media

1.6. Overview

The second section of this SRS covers the overall description of the application where the functional and non-functional requirements are presented in general terms. The remaining part of the section presents the specific requirements through use case diagrams where detailed information is provided regarding the interaction of the actor, the system, and the outcome of the action performed; followed by assumptions and dependencies. Section 3 of this SRS focuses on providing detailed use cases for functional requirements describing preconditions, triggers and post conditions along with screen mockups.

ViroTour application has the following functional requirements:

1. In the app using Dart and Flutter, provide means for a user to get a tour of all the pictures put together via hotspots.
2. Allow manual placement of hotspots as well as ability to delete and edit hotspots created earlier.
3. 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.
4. Allow the user to apply long press to pick an image, choose the height/width of the location inside the 3D image, and create a sharable link, which will bring another user to the exact spot.
5. Transitions from an image to the next should be animated and not abrupt.
6. Provide option when processing images to apply the "Glow" filter along with reasonable parameters if the indoor lighting is inadequate.
7. Program should accept mouse clicks as well as keyboard input for transitioning between images.
8. Transitions from an image to the next should be positional relative to the current image, current perspective, and location use clicks (or enters keyboard command)
9. Automatic Virtual Tour, moving between several images without additional User Input

10. Provide a VR view of the end product.

2. Overall Description

The ViroTour Application Development covers the front-end portion of the ViroTour application. This application has multiple requirements that have been derived in collaboration with the client. The ViroTour Application Development deals with the user interaction of ViroTour. These requirements are associated with three separate categories: Tour Navigation, View Customization and Hotspot Customization.

2.1. Use-Case Model Survey
There are 9 use cases that were defined. The table below provides a listing of use cases and a description.

Functions	Description	Use Case ID	Use Cases
1. Tour Navigation	This module allows users to navigate through the virtual tour	A 1.1	View the images as a sphere from hotspots.
		A 1.2	View the images using a VR viewer.
		A 1.3	Move from one hotspot to another.
		A 1.4	The transition between each image is smooth.
		A 1.5	Zoom and pan at each hotspot.
2. View Customization	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.	A 2.1	Share View
		A 2.2	Search for text that is available in the tour.
		A 2.3	Add glow effect to the view.
3. Hotspots Customization	Ability to add, delete, and edit tours.	A 3.1	Create/Edit/Delete a tour by uploading images.

Table 2.1 - Use Case Model
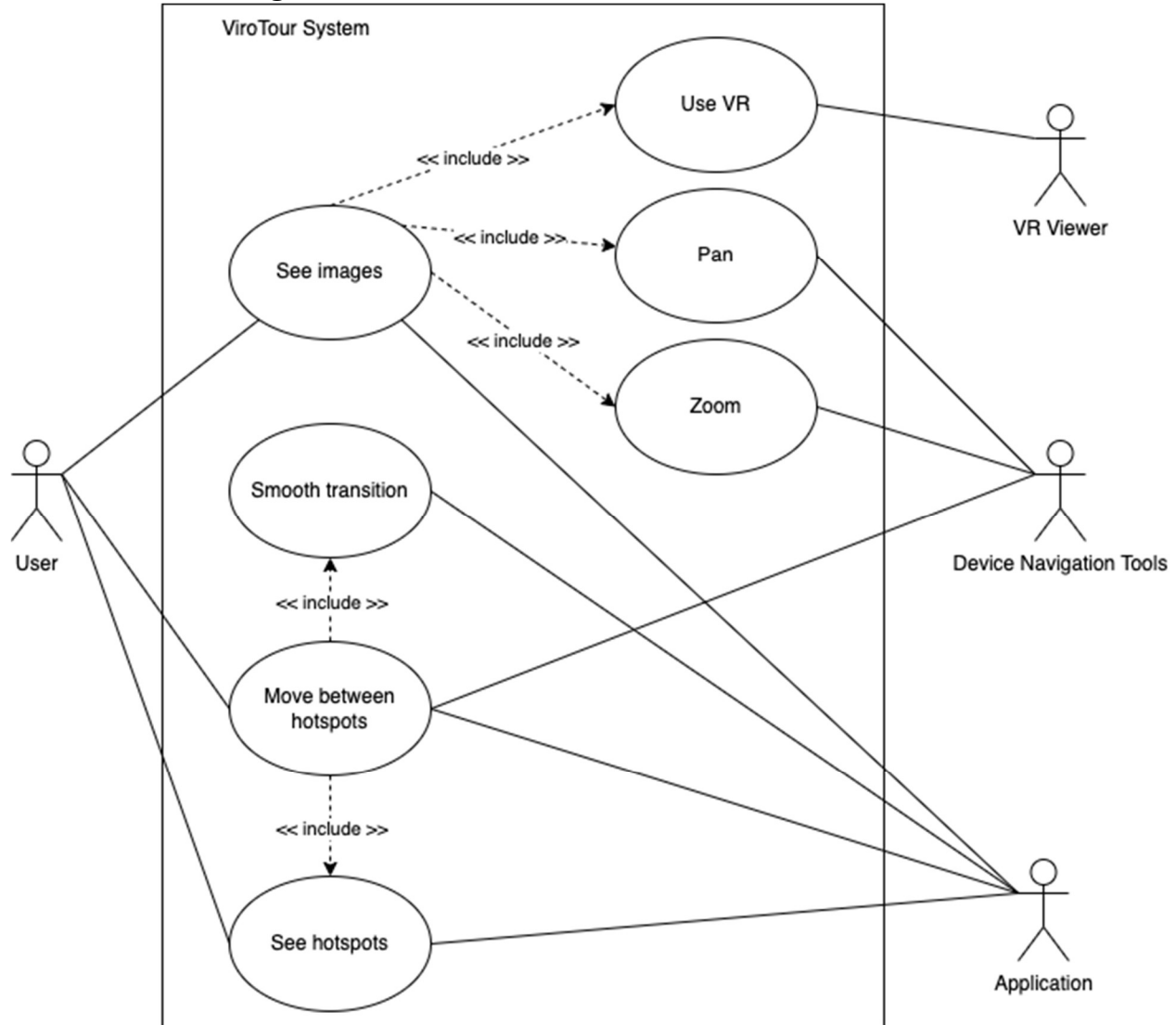
2.2.Use Case Diagrams

2.2.1. Tour Navigation

Figure 2.2.1 - Tour Navigation

Description:
This diagram describes the functionalities that allow users to navigate through the virtual tour.
Use Cases:
1. View the images as a sphere from hotspots.
2. View the images using a VR viewer.
3. Move from one hotspot to another.
4. The transition between each hotspot is smooth.
5. Zoom and pan at each hotspot.
Primary Actor:

User: the users of the application.

Supporting Actor:
  Device Build-in Navigation Tools: enable click / touch and type.
  VR viewer: a physical device allows user to see images in VR mode.
  Application: the application.

Stakeholders and Interests:
  User desires to be provided with the abilities to navigate through the virtual tour.

Pre-Conditions:
  The tour is successfully loaded

Post Conditions:
 User successfully navigates through the virtual tour.

Minimal Guarantee:
  User can perform actions using gestures.

Main Success Scenarios:
  - User selects the desired tour.
  - User sees the first hotspot of the tour.
  - User pans and zooms the images like the user is in a sphere.
  - User moves to the next hotspot.
  - User sees the smooth transition between each hotspot.

2.2.2. View Customization
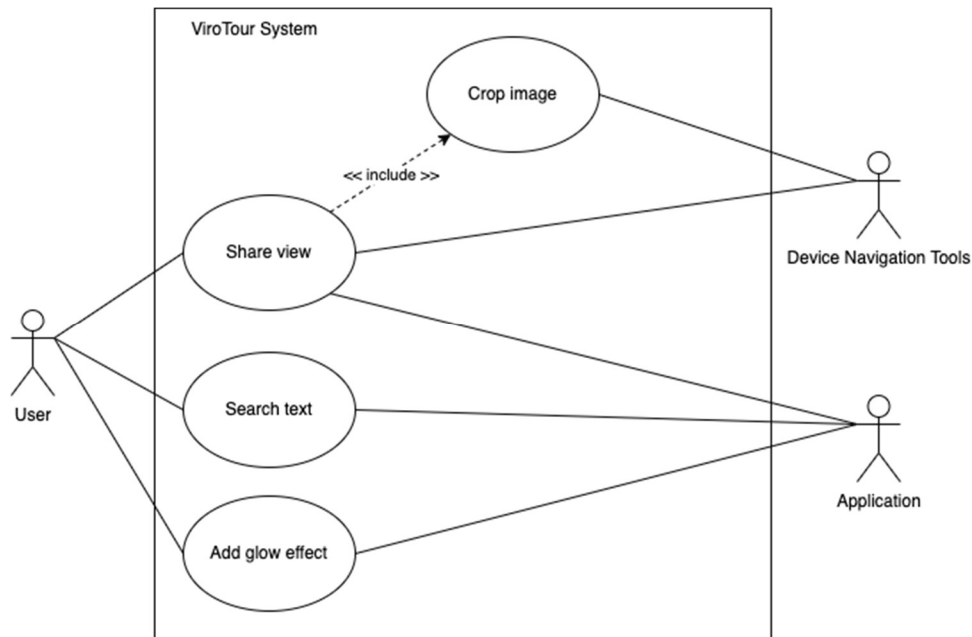
Figure 2.2.2 - View Customization

Description:
This diagram describes the functionalities that allow users to customize their views.

Use Cases:
  1. Share View
  2. Search for text that is available in the tour.

3. Add glow effect to the view.

Primary Actor:
      User: the users of the application.

Supporting Actor:
      Device Build-in Navigation Tools: enable click / touch and type.
      Application: the application.

Stakeholders and Interests:
      User desires to be provided with the abilities to customize their views of the tour and share the information.

Pre-Conditions:
      The tour is successfully loaded.

Post Conditions:
      User successfully uses the gestures to customize their views.

Minimal Guarantee:
      User can perform actions using gestures.

Main Success Scenarios:
- User presses and holds on the screen of the tour.
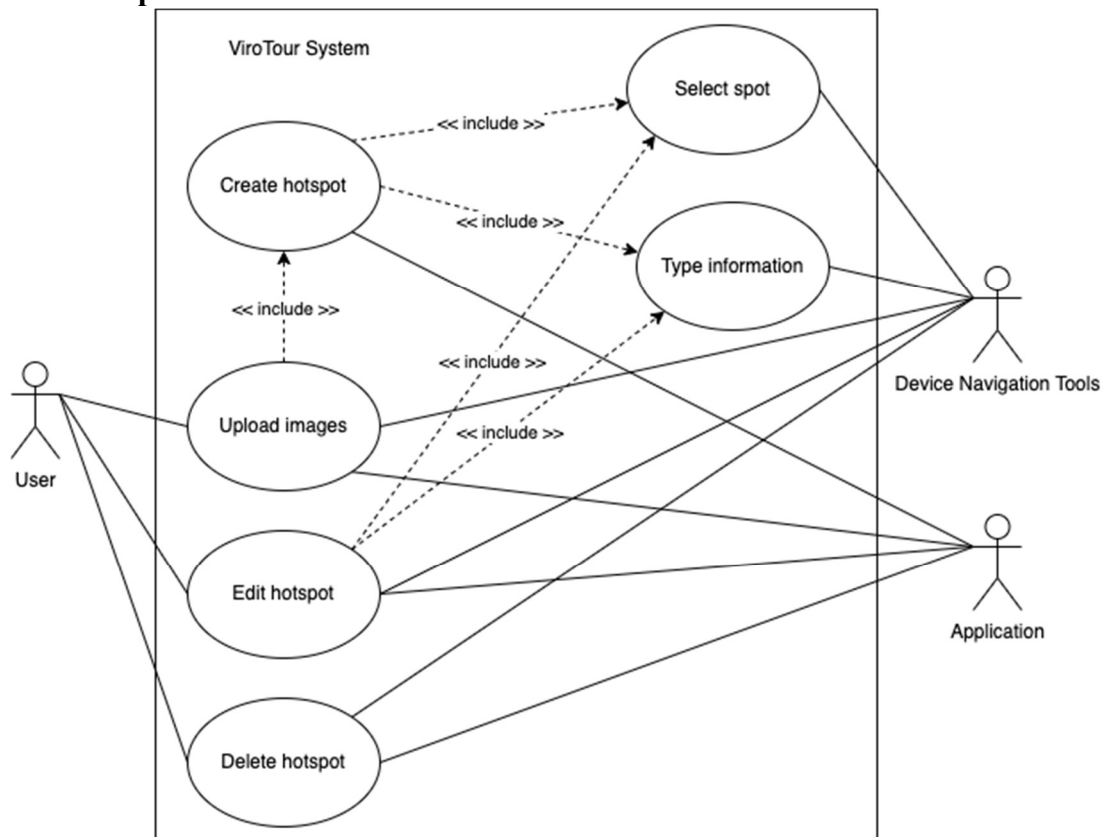- User sees the share options.

2.2.3. Hotspots Customization

Figure 2.2.3 - Hotspots Customization

Description:
This diagram describes the functionalities that allow users to customize the tours.

Use Cases:
1. Create an tour by uploading images.
2. Edit a tour.
3. Delete a tour.

Primary Actor:
 User: the users of the application.

Supporting Actor:
 Device Build-in Navigation Tools: enable click / touch and type.
 Application: the application.

Stakeholders and Interests:
 User desires to be provided with the abilities to customize hotspots.

Pre-Conditions:
 The application is successfully loaded.

Post Conditions:
 User successfully performs the tasks.

Minimal Guarantee:
 User can perform actions using gestures.

Main Success Scenarios:
- User enters the information of the tour.
- User confirms the creation of the new tour.
- User selects any tour and sees the options to modify the tour.
- User selects any field of the tour to modify its information.
- User deletes a tour.

2.3. Assumptions and Dependencies

2.3.1. Assumptions

- The software will be hosted on a dedicated server with appropriate cybersecurity protocols in place to allow customers access to it.
- Matterport Device will be made available and in working condition throughout the project.
- The frontend portion of the application will be available for the backend integration.
- Project team members will use open-source software development tools to complete the project.
- End-user training, beyond the user's guide, will not be provided.
- The Personnel costs associated with the project will be covered pro-bono.
- Milestones will be completed according to the project schedule.
- Team members will be available throughout the project.

2.3.2. Dependencies

- ViroTour front end: The front end must be functioning to interact with the backend functionalities.
- Matterport device: Valid photos captured using any phone camera mounted upon the Matterport device.

- Data Store: Storage medium to store hotspots and images.

3. Specific Requirements

This section provides the specific requirements for each implementation of the 9 use cases discussed in section 2 with the action taken, the system response, and the pre and post conditions.

Use-Case Reports

3.1. Tour Navigation

3.1.1. Tour Navigation – Use Case ID A 1.1 – View the Images as a Sphere

Summary: Users can automatically view all existing panoramic images by transitioning through them as a tour.

Preconditions: None.

Triggers: User selects the tour button in the UI.

Post Conditions: After successful deselection, no auto-tour is selected, and user returns to manual tour.

3.1.2. Tour Navigation – Use Case ID A 1.2 – View the images using a VR viewer

Summary: This process allows the user to have a virtual reality experience of a series of locations uploaded as images.

Preconditions: There exists already processed and stitched images in the local storage or on the cloud storage of the application.

Triggers: The user selects one of the panoramic tours.

Basic course of events:

[option =
View Tours]

alt

[option =
Selected a Tour]

Requests all Tours as a list

Returns a list of tours

Displays the list of tours

Request the tour specified

Request the tour specified

To View
UML

Display Tour

Returns the specified tour as JSON
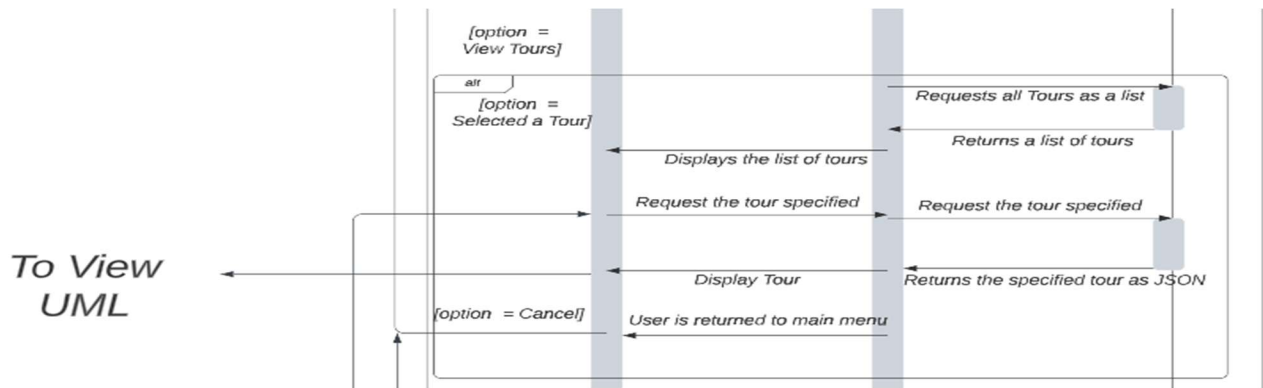
[option = Cancel]

User is returned to main menu

Figure 3.1.2 - Tour Navigation – View the Images Using a VR Viewer

3.1.3. Tour Navigation – Use Case ID A 1.3 – Move from one transition point to another

Summary: Users can successfully move from one image to the next by selecting the transition point within the first image.

Preconditions: A tour was uploaded from a series of photos uploaded to the ViroTour server. The user has successfully opened the tour and is viewing an image a transition point.

Triggers: User selects a transition point.

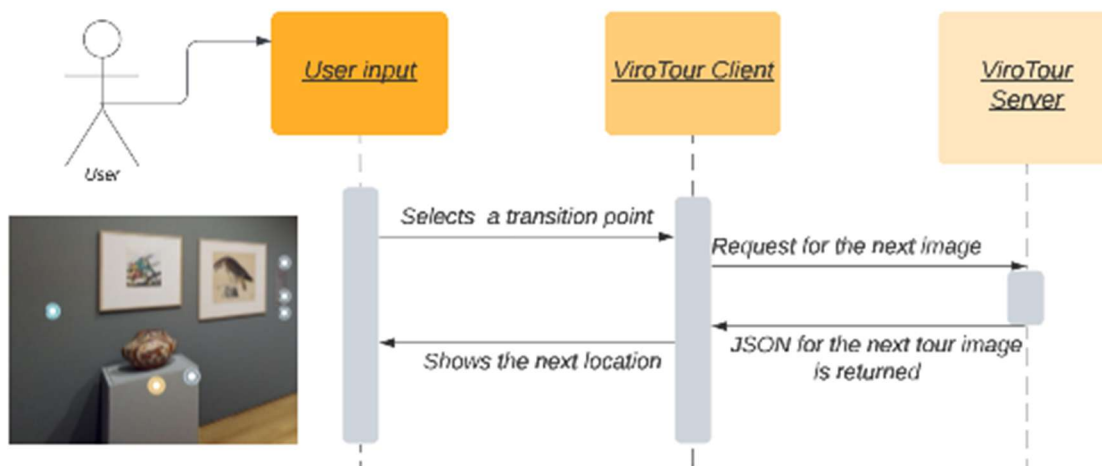**Basic course of events (Scenario):**

Figure 3.1.3 - Tour Navigation – Move from one Transition Point to Another

Post Conditions: The user is now in the next image along the tour.

3.1.4. Tour Navigation – Use Case ID A 1.4 – The transition between each image is smooth.

Summary: The user can quickly navigate through whichever images they have either chosen to capture themselves or have had shared with them from other users, during the transitions between

images on a tour the transition should not be abrupt maximizing the user experience of taking a tour.

Preconditions: A series of images have been created into a tour and loaded into ViroTour server.

Triggers: The user opens the ViroTour application.

Basic course of events:
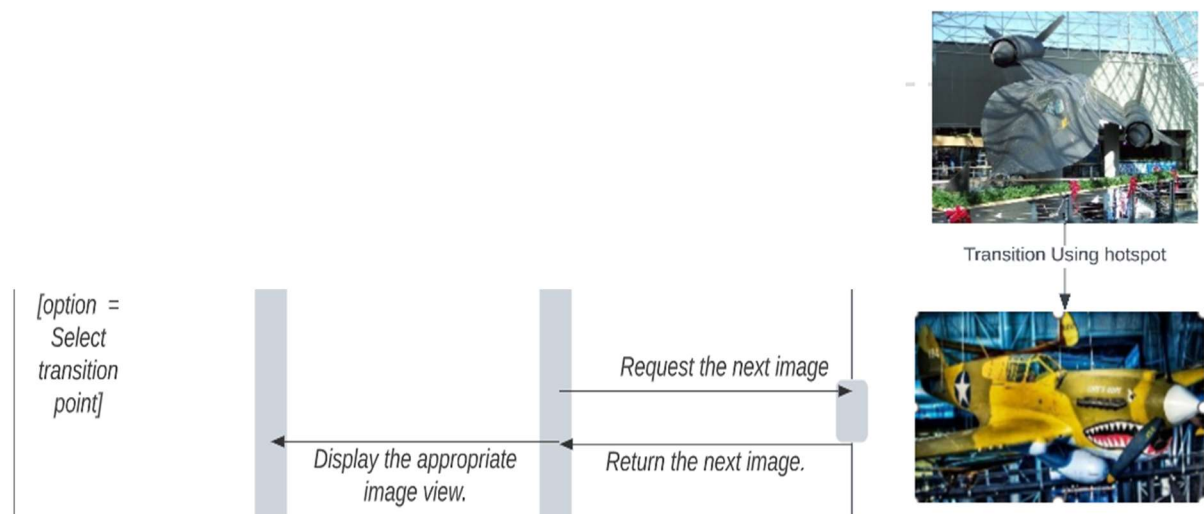
Figure 3.1.4 - Tour Navigation – Smooth Transitioning Between Each Image

Post Conditions: The user is looking at the next image they chose in the tour.

3.1.5. Tour Navigation – Use Case ID A 1.5 – Zoom and pan at each hotspot.
Summary: User can manually zoom in/out of the image and can pan the image view around to their preference.

Preconditions: User opens an image.

Triggers: User scrolls their mouse wheel in, pinches their fingers on the touch screen, or uses the zoom bar to zoom in.

Basic course of events:

Actor	System	Screen
1. User scrolls their mouse wheel forward.		

	2. The image zooms in.	
3. User uses the zoom bar to zoom in.		
	4. The image zooms in.	
5. The user pinches their fingers on the touch screen.		
	6. The image zooms in.	

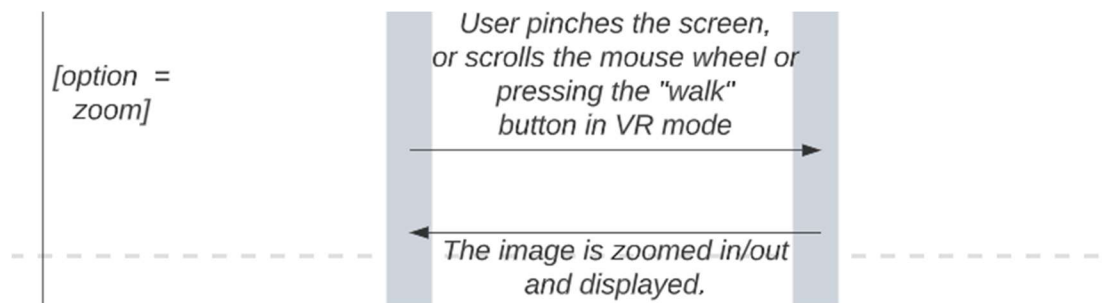*Table 3.1.5a - Tour Navigation – Zoom and Pan at Each Hotspot – Zoom In*

Triggers: User scrolls their mouse wheel backwards, pinches their fingers on the touch screen, or uses the zoom bar to zoom out.

Basic course of events:

Actor	System	Screen
1. User scrolls their mouse wheel backward.		
	2. The image zooms out.	

3. User uses the zoom bar to zoom out.		
	4. The image zooms out.	
5. The user pinches their fingers on the touch screen.		
	6. The image zooms out.	

Table 3.1.5b - Tour Navigation – Zoom and Pan at Each Hotspot – Zoom Out

Figure 3.1.5 - *Tour Navigation – Zoom and Pan at Each Hotspot*

Post Conditions: The image is zoomed to the user's preference.

3.2. View Customization

3.2.1. View Customization - Use Case ID A 2.2 – Search for text that is available in the tour.

Summary: User can search for specific text that may or may not appear throughout the physical space that's being toured. The app will provide references to all instances of the specified string of text and allows the user to easily navigate their view to the location where each instance of the text appears.

Preconditions: All images have been uploaded and processed.

Triggers: The user selects the search bar at the top of the screen and types in a series of characters they wish to locate.
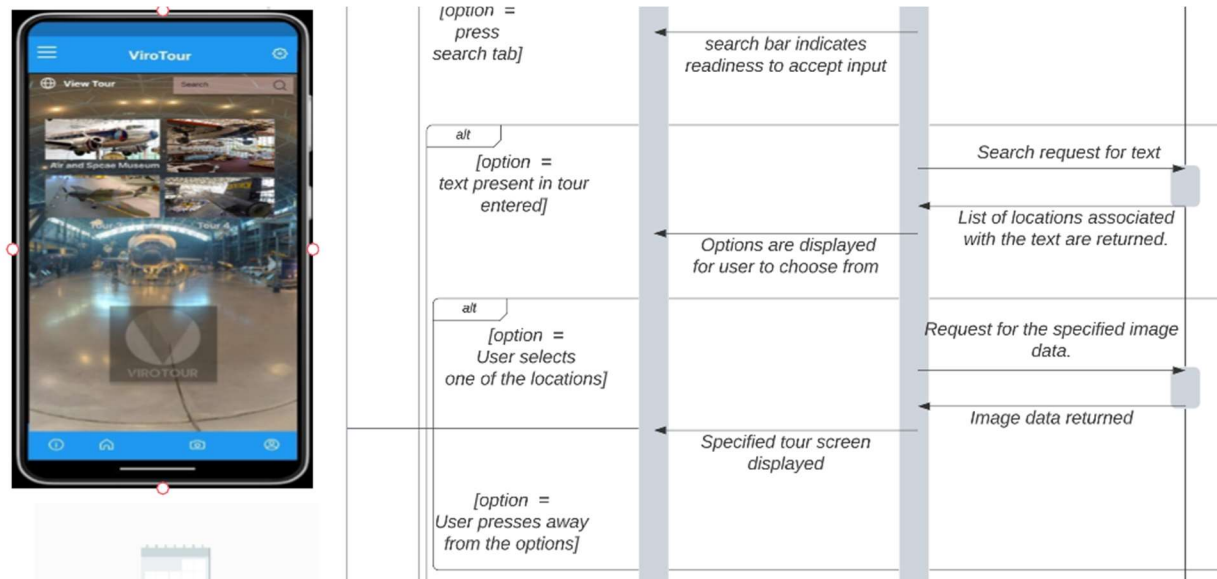
Basic course of events:

Figure 3.1.7 - View Customization – Search for Text

Post Conditions: The user is made aware of and may navigate to instances of the desired string of words.

3.2.2.   View Customization – Use Case ID A 2.3 – Adjust the glow filter over image.

Summary: The glow filter effect will allow the application users to adjust the brightness and contrast of an image. The glow effect adds light to an image when visibility is low due to low lighting.

Preconditions: There exists images on the app's local or cloud storage.

Triggers: The user selects an image from the gallery.

Basic course of events:

Figure 3.1.8 - View Customization – Glow Filter

Post Condition: A panoramic image glow effect used in the tour is adjustable by the user to optimum lighting.
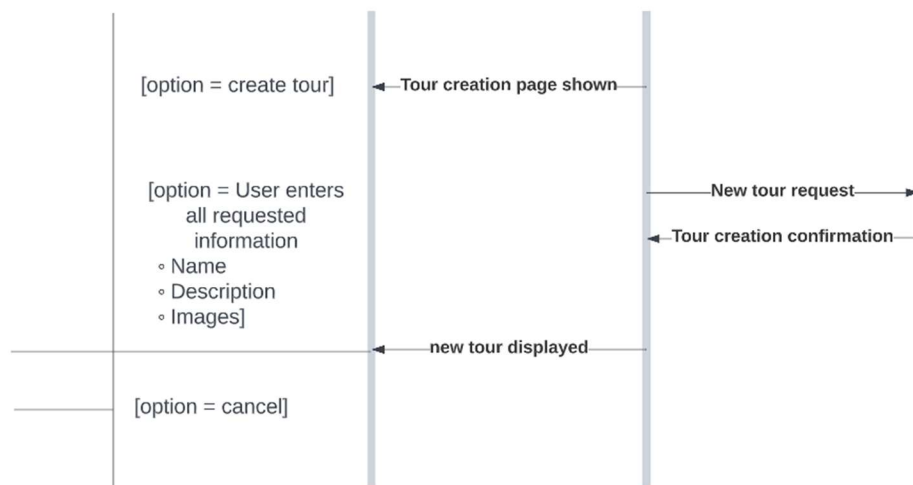
3.1. Hotspots Customization

3.1.1. Hotspots Customization - Use Case ID A 3.1 – Create a Tour

Summary: User can create a tour by entering a name, description, and images used to create the tour.

Preconditions: User can navigate to the create tour page by clicking the create tour button available in the hamburger menu. User then enters information required for the tour.

Triggers: User selects to save the tour.

Basic course of events:

Figure 3.1.9.1 - Hotspot Customization – Create

Post Conditions: After successful selection, tour is created.

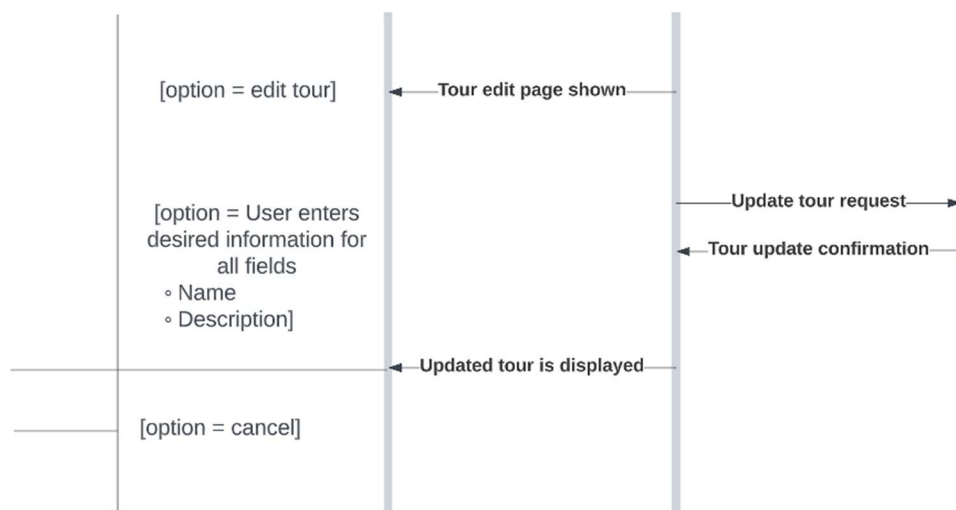### 3.1.2. Hotspots Customization - Use Case ID A 3.2 - Edit Tour

Summary: User can edit a tour on the that the user has created. The name and description fields can be changed,

Preconditions: User selects a tour.

Triggers: User selects edit tour.

Basic course of events:

Figure 3.1.9.2 - Hotspot Customization – Edit

Post Conditions: After successful selection, tour is edited.

3.1.3. Hotspots Customization - Use Case ID A 3.3 - Delete Tour

Summary: User can delete a tour that the user has selected.

Preconditions: User selects a tour to edit.

Triggers: User selects delete tour.

Basic course of events:

Figure 3.1.9.3 - Hotspot Customization – Delete

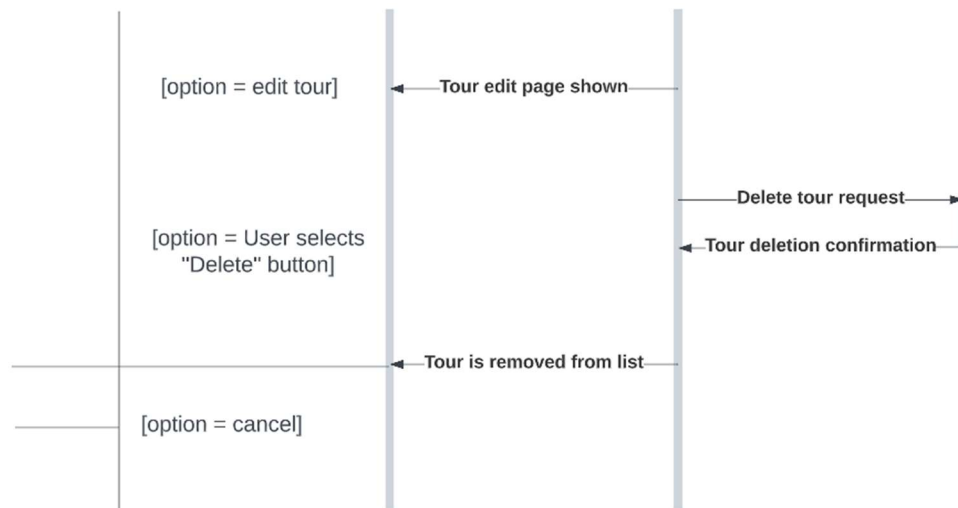**Post Conditions:** After successful selection, tour is deleted.

3.2. Full UML Diagram
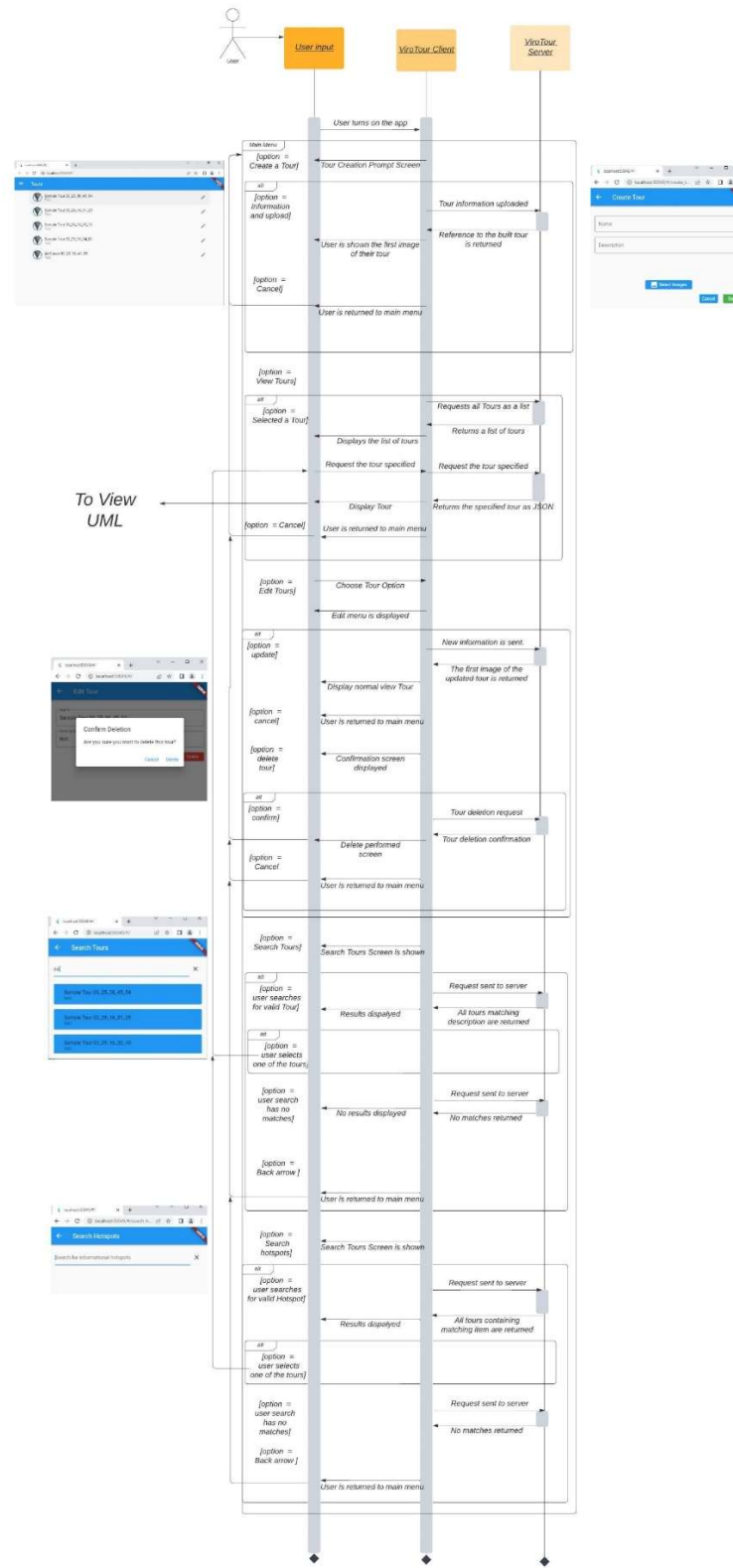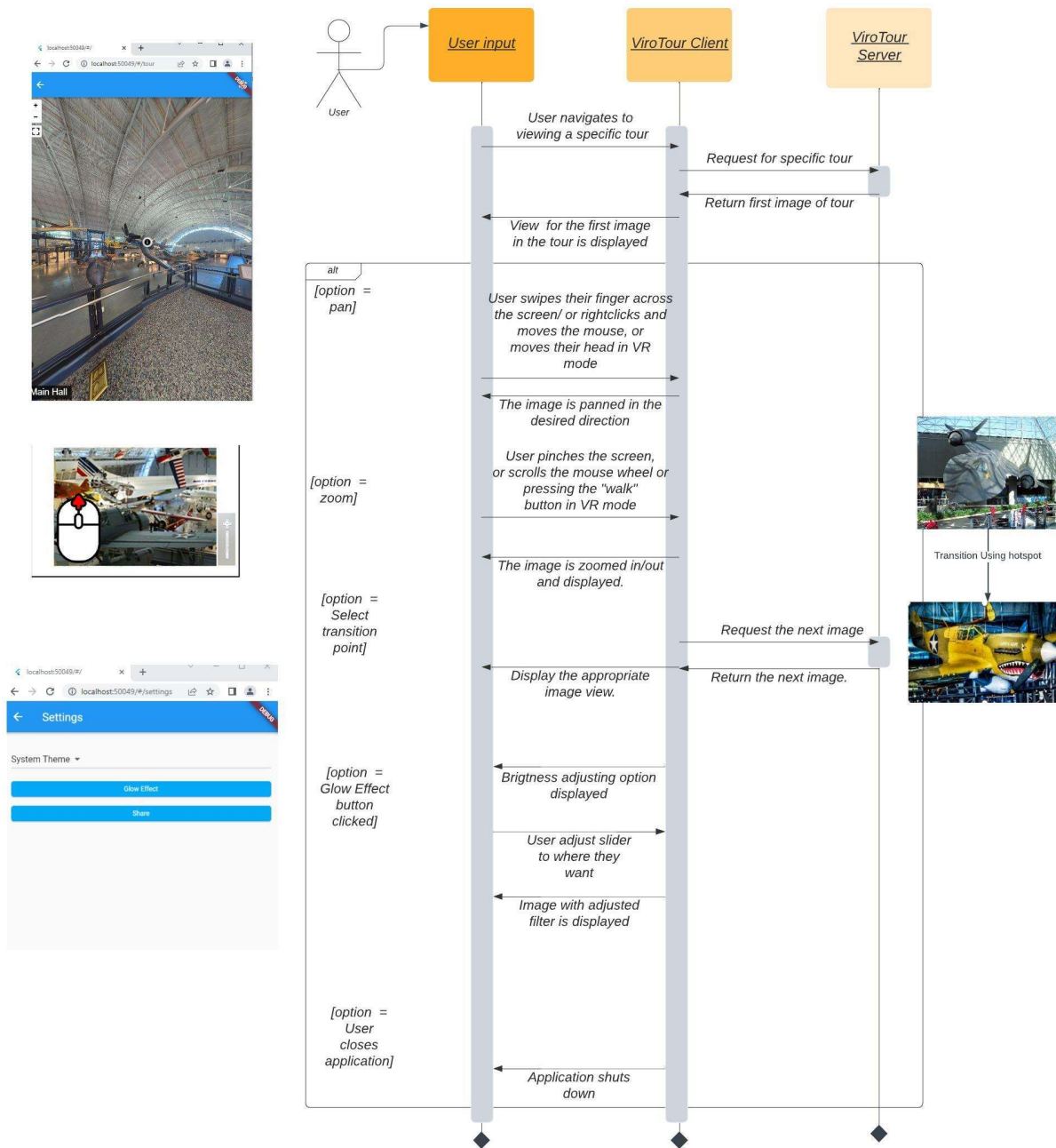
Table names can be click to see UML in Lucid.app.

Figure 3.2.1 Menu UML

Figure 3.2.1 Tour View UML

3.3. Supplementary Requirements

There are no additional supplementary requirements for this implementation.

3.4. **Non-functional Requirements**

1. The image processing component is technology and platform neutral; the app needs to be in Dart and Flutter.
2. It is possible to use a third-party software to process images to automatically place the hotspots, but the resulting data must be consumed, interpreted, and presented in Dart and Flutter.
3. Image processing should take no more than one minute per image.
4. Post-processed images, associated text, any metadata, and hotspot locations can be stored on a cloud location in a separate directory for each tour to support multiple tours simultaneously.
5. Tour viewing should display images in less than a second on Wi-Fi with a speed of 10Mbps+.

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	View Customization	A-2.3	Zoom and pan at each hotspot.	x	* Note 3		

	using OpenCV (https://stackoverflow.com/questions/68592934/implement-glow-filter-in-cv2-python). It is very useful if the indoor lighting is inadequate.						
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 browser, not on iOS or Android."