Software Requirements Specification

ViroTour Server Processing

University of Maryland Global Campus

Software Engineering 670

Spring Cohort 2023

Team B

Document Control

Document History

Version	Issue Date	Changes
0.1	1/28/2023	Initial Draft.
0.2	2/12/2023	Revisions after milestone 1.

TABLE OF CONTENTS

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 Server Processing (VSP) application. Defining the requirements within this document ensures that the development team understands the intent of the application to meet the needs of Dr. Mir Assadullah, our customer. 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 VSP application and the user in the form of use cases.

1.2 Scope

ViroTour is a cross platform application that shall support web and mobile environments. ViroTour shall ingest images to create panoramic views used to facilitate virtual tours. We will provide editing and search capabilities.

The development team was divided into two groups. Each development team shall focus on one area of the application. Team A's development efforts are based on front-end development whereas Team B's efforts will be focused on back-end development. This SRS will define Team B's requirements.

The scope of this SRS is to deliver the project's identified requirements that utilize backend functionality and capability. Below are two categories that respectively detail what is to be *in scope* and what is to be *out of scope* for this project.

In Scope for Team B:

- Database to store Images, Locations, Hotspots, and Text.
- Core Module: Image processing to enable tour navigation.
- Editing Module: Perform modifications on Images, Locations and Hotspots.
- Search Module: Perform text-based search.
- Backend support for any number of Virtual Tours.
- Functional testing.

Out of Scope for Team B:

- 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 smooth.
- Capability for user to capture a particular section of the image on screen and to send it over/share with others.
- Features related to security, logging, auditing, or monitoring.
- Performance testing.

1.3 Definitions, Acronyms and Abbreviations

Common terminology is seen and presented throughout this document. To remove any ambiguity, below is a list of terms and their intended meanings:

Term	Definition
Dart	An object-oriented programming language developed by Google.
Flutter	An open-source framework developed by Google.
Hotspot	A preset location in the app that acts as a destination point of interest users can interact with.
Image Stitching	Process of combining multiple pictures that overlap and producing a panorama image.
Matterport Axis	A motorized phone-mountable device (developed by Matterport) that aids in taking panorama pictures with a phone.
Panorama	An unobstructed view of an area that is visible in every direction.
Post-processed Images	Original images that have been further digitally enhanced .
SRS	Software Requirements Specification.
UI	User Interface.
VSP	ViroTour Server Processing.
Dart	An object-oriented programming language developed by Google.
Flutter	An open-source framework developed by Google

1.4 Project Documentation

1.4.1 Project Suite of Documents

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

	Document	Version	Date
1	Project Management Plan (PMP)	0.2	02/12/2023
2	Software Requirements Specification (SRS)	0.2	02/12/2023
3	Technical Design Document (TDD)	0.1	02/12/2023
4	Software Test Plan (STP)	0.1	02/12/2023
5	Programmers Guide (PG)	0.1	TBD
6	Deployment and Operations (DevOps)	0.1	TBD
7	User Guide (UG)	0.1	TBD
8	Test Report (TR)	0.1	TBD

1.4.2 Document References

Brown, M. (2020). Capstone project guide. *University of Maryland Global Campus.* https://learn.umgc.edu/d2l/le/content/732302/viewContent/29895181/View

Matterport. (n.d.). *Meet Axis.* https://matterport.com/axis

Stack Overflow. (2021). *Implement glow filter in CV2 Python.* https://stackoverflow.com/questions/68592934/implement-glow-filter-in-cv2-python

Weinhaus, F. (n.d.). Glow. *Fred's ImageMagick Scripts.* http://www.fmwconcepts.com/imagemagick/glow/index.php

1.5 Overview

The VSP's SRS is made up of the Overall Description, and Specific Requirements sections. The *Overall Description* section gives a general overview of the requirements including any assumptions or dependencies affecting those requirements. The *Specific Requirements* section goes into detail for each functional and nonfunctional requirement. Use Cases are provided for the functional requirements only.

Having identified the main modules to be integrated into the ViroTour, the team reviewed three tools for development: Flutter & Dart, Google Vision, and Matterport. Matterport satisfies all the client's requirements and allows the integration of all the critical components of a virtual tour application. As a result, Matterport has been selected for all the panoramic image processing and Flutter & Dart for developing the application user interface and business processes.

Functional Requirements:

- Use Matterport Axis device to automatically take images using an app written in Dart and Flutter and stitch the four images together to create a panoramic image.
- Using the images, automatically find out where the transition hotspots are.
- Use a tool to read text from the images and note the location of the text.
- Provide means for a user to get a tour of all the pictures put together via hotspots.
- Allow manual placement of hotspots as well as ability to delete and edit hotspots created earlier.
- 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.
- Allow 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.
- Provide option when processing images to apply the "Glow" filter along with reasonable parameters.

2 Overall Description

The ViroTour Server Processing (VSP) application is the backend portion of the whole ViroTour application. This VSP application has multiple requirements that have been derived in collaboration with the client. VSP deals with panoramic image processing, the storing and processing of data that the ViroTour front end application will interact with. These requirements are associated with three separate categories.

1. **Core:** The Core category of requirements deals with the initial creation of the hotspots and their panoramic images that the frontend ViroTour will present.
2. **Editing:** The Editing category of requirements deals with alteration of the initial creation of data supplied to the frontend ViroTour application.
3. **Search:** The Search category of requirements deals with the retrieval specific data retrieved from the frontend ViroTour application.

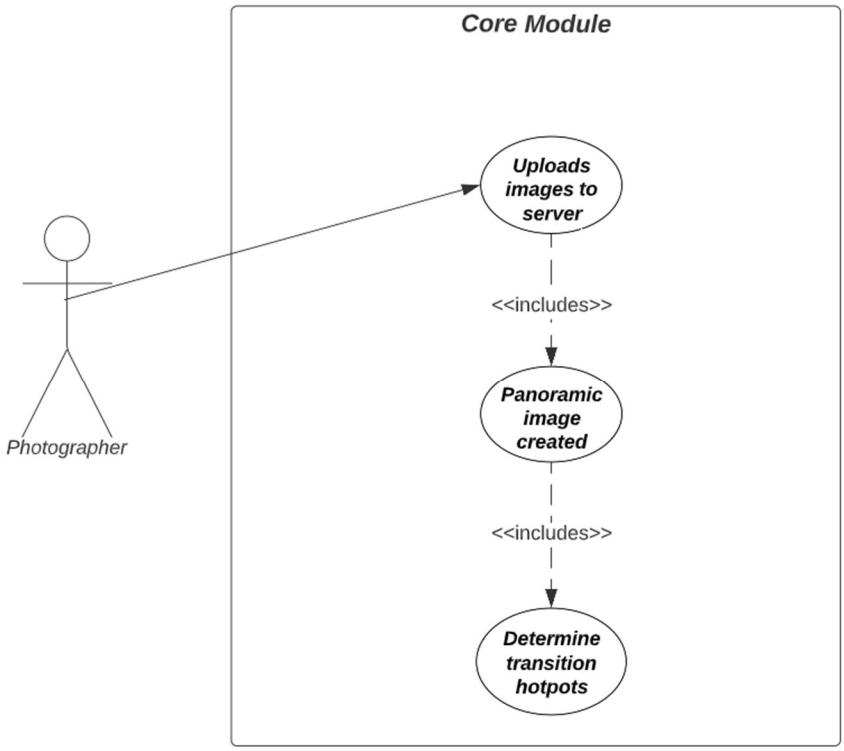
2.1 Use-Case Model Survey

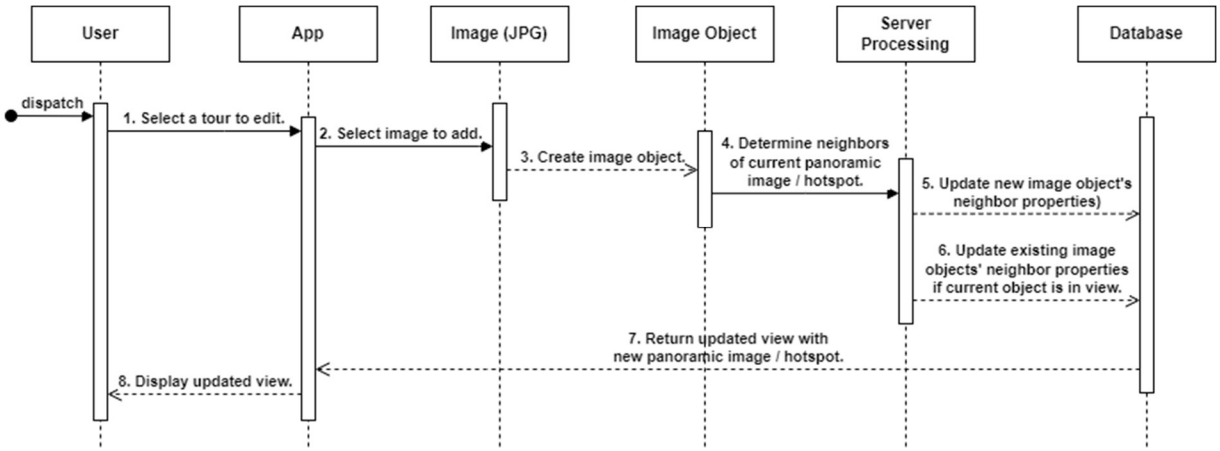
There are 10 use cases that were defined. The table below provides a listing of use cases and a description.

ID	Use Case	Description
1.1	Core: Uploads images to server	Upload images to server as groups which represent a location ("image group" or "location")
1.2	Core: Panoramic image created	Determine and provide a panoramic view of a location.
1.3	Core: Determine transition hotspots	Determine and provide transitions between locations ("transition hotspot").
2.1	Editing: Extract text from images	For each location, detect text from images and store results.
2.2	Editing: Detect and blur human faces	Blur human faces and skin from all images.
2.3	Editing: Apply filters	Ability to apply filters to images and store results.
2.4	Editing: Add hotspot	Ability to add a transition hotspot.
2.5	Editing: Edit hotspot	Ability to edit a transition hotspot.
2.6	Editing: Delete hotspot	Ability to delete a transition hotspot.
3.1	Search: Search text	Ability to search all text from B-2.1.

2.2 Use Case Diagrams

2.2.1 Core Module





Description: Upload images to ViroTour Server Processing.
Use Cases:

- Upload images to server.
- Panoramic image created.
- Determine transition hotspots.

Primary Actor: The photographer user

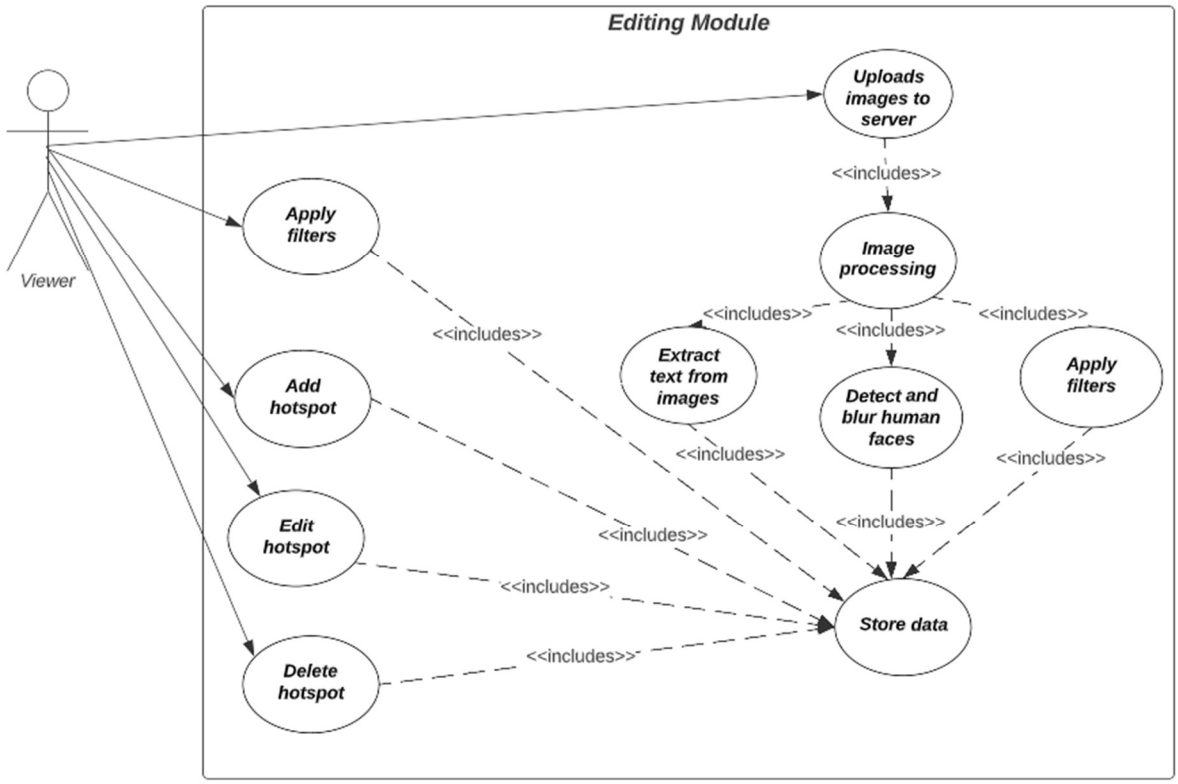
Stakeholder and Interests: The photographer who wants to generate a virtual tour from images.
Preconditions: The ViroTour Server Processing system must be functioning, and the user must have basic computer operation skills.
Post Conditions: The transition hotspots can be determined from the uploaded images.
Main Success Scenario:

1. The User uploads images.
2. The ViroTour Server Processing system processes the uploaded images and creates the Panoramic images.

Extensions:
Unsuccessful image upload

- The system notifies the user that the image upload was unsuccessful.

2.2.2 Editing Module



Description: Images are uploaded, processed, and hotspots can be modified.
Use Cases:

- Uploads images to server.
- Extract text from images.
- Detect and blur human faces.
- Image processing.
- Apply filters.
- Add hotspot.
- Edit hotspot.

- Delete hotspot.
- Store data.

Primary Actor: The photographer user
Stakeholder and Interests: The photographer who wants to generate a virtual tour from images.
Preconditions: ViroTour Server Processing system must be functioning, and the user has basic computer operation skills
Post Conditions: The user can successfully upload an image and add, edit, and delete hotspot.
Main Success Scenario:

1. The user uploads an image.
2. The image gets processed (extract text, detect and blur human face, apply filter).
3. The user adds, edits, and deletes hotspots from the stored image data or can apply additional filters to images.

Extensions:
Unsuccessful image upload.

- The system notify user that image upload was unsuccessful.

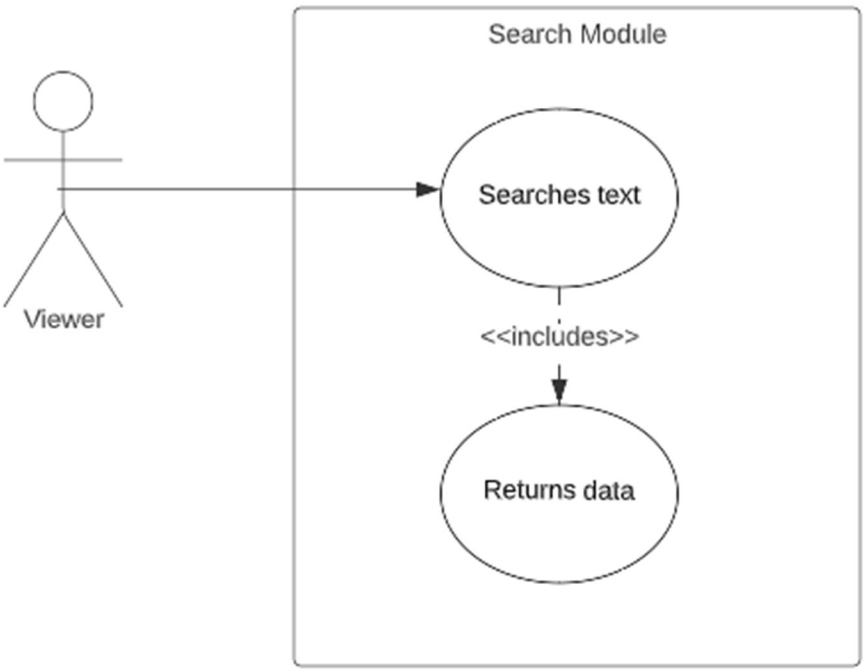
Unsuccessful add, edit, or delete hotspot action.

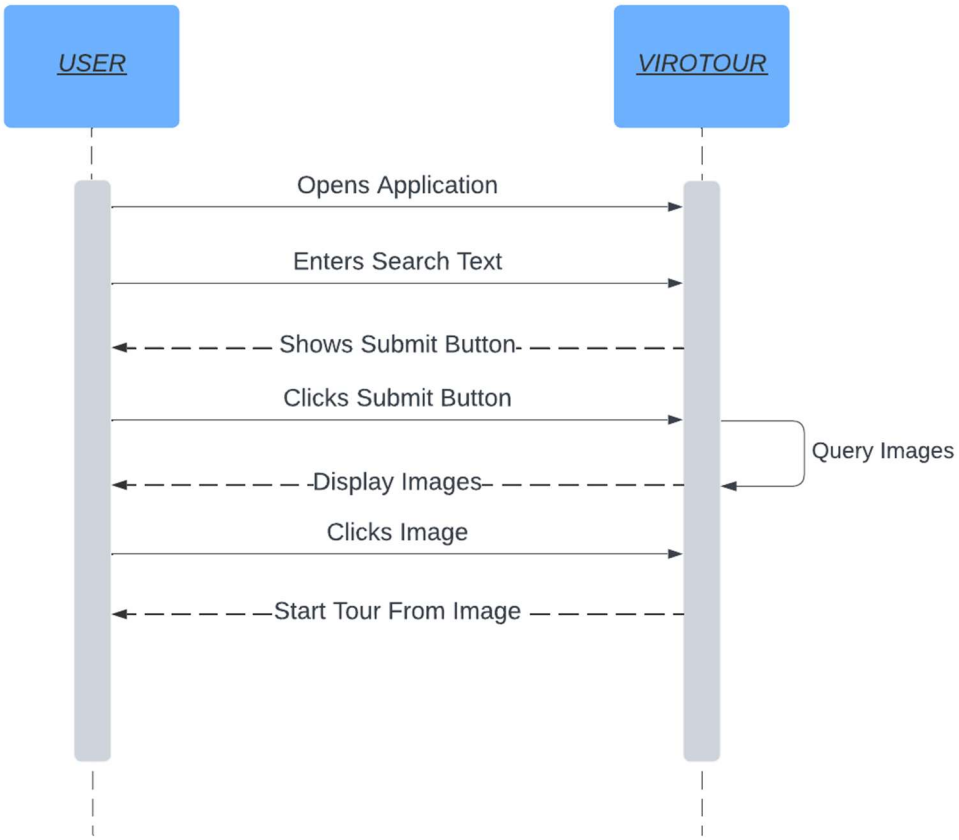
- The system notifies the user that the requested action cannot be performed.

Unsuccessful filter application.

- The system notifies the user that the requested action cannot be performed.

2.2.3 Search Module





Description: Search text read from images.
Use Cases:

- Searches text.
- Returns data.

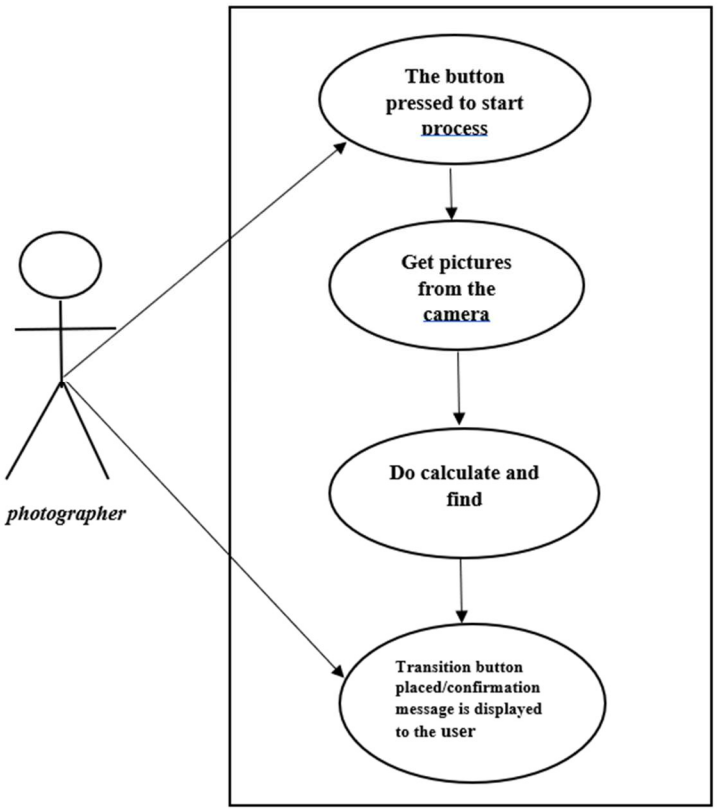
Primary Actor: The viewer user
Stakeholder and Interests: The view user who wants to perform text searches within the virtual tour.
Preconditions: ViroTour Server Processing system must be functioning, and the user has basic computer operations skills.
Post Conditions: The system return searched text.
Main Success Scenario:

1. The user search for text from an uploaded image.
2. The System return the text.

Extensions:
Invalid text or Text is not included in uploaded image.

- The system return massage indicating that the text was not found.

2.2.4 Transition Hotspot Module



Description: Determination of transition hotspot.
Use Cases:

- Receive images.
- Image analysis and review.
- Calculate and find hotspots.
- Save them.

Primary Actor: system
Stakeholder and Interests: Users want to see buttons on the images so that they can navigate through all parts of the panorama images.
Preconditions: Images sent from the camera should be provided in sufficient number and appropriate size.
Post Conditions: None
Main Success Scenario:

1. By pressing on a transition in any direction, the image for that location will be replaced slowly.

Extensions: None

2.3 Assumptions and Dependencies

2.3.1 Assumptions

- The frontend portion of the application will be available for backend integration.
- Project team members will use open-source software development tools.
- 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.
- Matterport camera: Valid photos captured using Matterport camera.
- Data Store: Storage medium to store hotspots and images.

3 Specific Requirements

3.1 Use-Case Reports

3.1.1 Feature Area: Core

3.1.1.1 Use case name: Upload images to server.

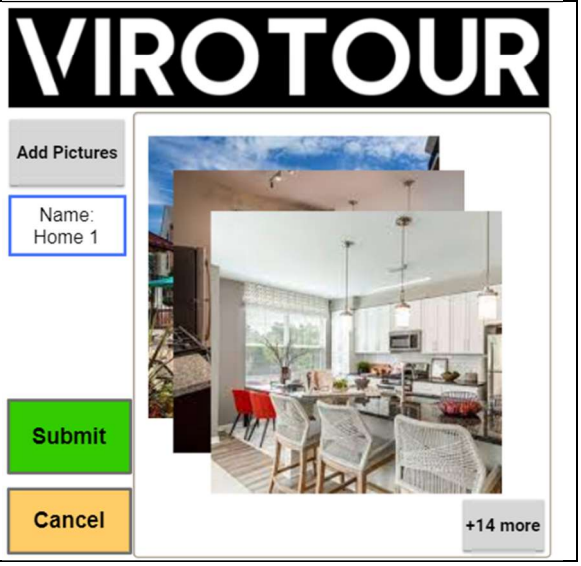
Summary: The application will upload the group of pictures to a server.

Preconditions: None.

Triggers: The user clicks “Create Tour”.

Basic course of events (Scenario):

Internal Precondition:

Actor	System	Screen
1. User clicks “Create Tour”.		
2. The user adds pictures, types a name, and clicks “Submit”.		

Actor	System	Screen
	3. The system updates the database to include the images under a new name. and sends a confirmation.	
	4. The system navigates use case "Ability to apply filters to images and store results*" (See below).*	

Alternative Course of Events: Unable to upload images

Actor	System	Screen
	2. When the user clicks "Upload", the system fails to process the images and prompts the user to try again.	

Alternative Course of Events: Name of tour already exists.

Actor	System	Screen
	2. When the user clicks "Upload", the system alerts the user that a name with that project already exists.	
3. User clicks "Append Images to Project".		
	4. The system updates the database to include the images under a new name and sends a confirmation.	

Inter Post Condition:

1. The system shall attempt to update transition links for images.

3.1.1.2 Use case name: Panoramic image created

Summary: The application shall stitch a group of images to create a panoramic view of a location.

Preconditions: The images should be stored within a folder on a server.

Triggers: The user uploads images to the server.

Basic course of events (Scenario):

Internal Precondition: User has access to the server.

Actor	System	Screen
	1. System retrieves a group of image files from the server.	
	2. System generates a panoramic image.	
	3. System stores panoramic image on the server.	
	4. System displays a success message notifying the user that the panoramic image was successfully created.	

	5. The success message disappears and the system displays the panoramic image.	
6. The user selects the Dashboard option.		
	7. The system navigates to the user dashboard.	
8. The user selects the "View Tours" option.		

	9. The system displays a list of tours	ç VIROTOUR Tours Home 1 Home 2 Home 3 Home 4 Home 5 Home 6 Home 7
10. The user clicks on a tour in the listing.		
	11. The system displays the panoramic image.	VIROTOUR 

Alternative Course of Events:

Actor	System	Screen
	1. System retrieves a group of image files from the server.	
	2. System could not generate panoramic image.	

	3. System displays an error message.	

3.1.1.3 Use case name: Determine transition hotspots

Summary: This use case is about how transition hotspots are placed on images.

Preconditions: The images sent from the camera should be provided in sufficient number and appropriate size.

Triggers: When the user enters the images, the process starts. After receiving the images, the operation of this process starts.

Basic course of events (Scenario): The images are received and their size is calculated by the program. Then the hotspots are found by the algorithm and their positions are saved in the program and then the corresponding buttons are placed on the image.

Internal Precondition:

Actor	System	Screen
	1. The system receives the information of the images	
	2. The system checks the size and quality of the images and in case of lack of quality or inappropriate size, it gives an error message and stops the operation	

	3. The information system processes images and finds important hotspots	
	4. The sensitive points are identified on the images and the output is displayed to the user, and then a message is displayed whether the output is confirmed or not.	
5. The user confirms the output		
	6. The success message allows the operation	
7. The user does not confirm the result		

	8. The process is repeated by the system	

Inter Post Condition:

1. All transition hotspots are stored on the server.

3.1.2 Feature Area: Editing

3.1.2.1 Use case name: Extract text from images

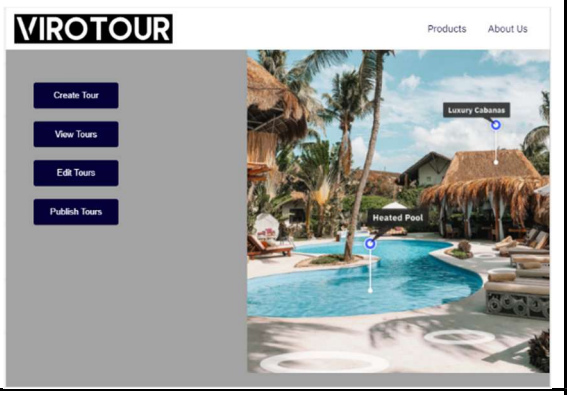
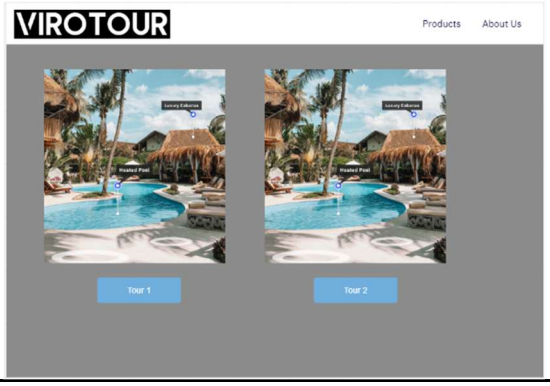
Summary: This use case is about a specific application of Optical Character Recognition (OCR) technology that allows for the extraction of text information from user-input pictures. The process of extracting text from images involves several steps, including image pre-processing, text detection, and text recognition. The user would first need to provide an image that contains the text they want to extract. The image is then passed through a pre-processing step to enhance the quality of the image, making it easier for the OCR algorithm to detect and recognize the text. Next, the OCR algorithm detects the text regions within the image and then proceeds to recognize the characters within those regions. Finally, the recognized text is outputted to the user in a format that they can easily read and use.

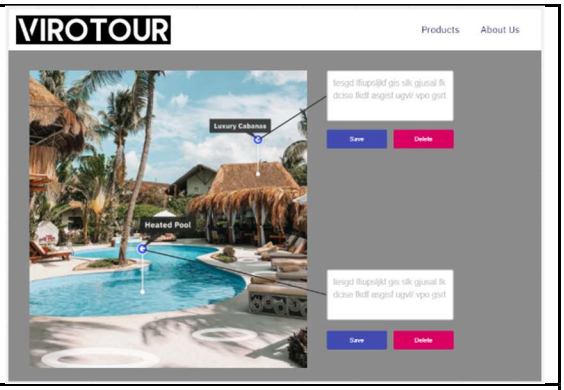
Preconditions: The images provided by the user are crucial for the success of the OCR process, and it is essential that they meet certain requirements in order to ensure the accuracy of the text recognition. The images need to have a certain level of clarity, meaning they should be focused and free of any blurriness or distortion. Additionally, the lighting conditions in the image need to be sufficient in order to allow the OCR algorithm to detect and recognize the text accurately. Poor lighting can cause the text to be too dark or too light, making it difficult for the algorithm to process the image accurately. A stable lens is also important in order to avoid any distortion or warping of the image that can make it difficult for the OCR algorithm to detect and recognize the text accurately. Without meeting these requirements, the process may not be able to recognize any text information in the images provided by the user, resulting in inaccurate or incomplete results.

Triggers: When the user enters the image, the program will first create a panorama of the image. This step is important because it allows the program to capture more of the text in the image, even if it is located in different areas of the image. Once the panorama is created, the program will initiate the text recognition process. This process involves using OCR algorithms to detect and

recognize the text in the image. The OCR algorithm scans the image and identifies the regions of the image that contain text. It then proceeds to recognize the characters within those regions and converts them into a machine-readable format. The recognized text is then provided to the user in the form of a list. The list will contain all the text that was recognized in the image, along with the location of the text within the image. This allows the user to easily review and edit the recognized text, as well as to extract the necessary information in a more efficient way. Additionally, the program can provide the recognized text in an editable format such as a text file or a document, which makes it easier for the user to copy and paste the text or to use it in another application.

Basic course of events (Scenario):

Actor	System	Screen
	1. System generates a panoramic image.	
2. The user selects Edit tours.		
	3. System shows all tours the user saved.	
4. The user selects Tour 1		

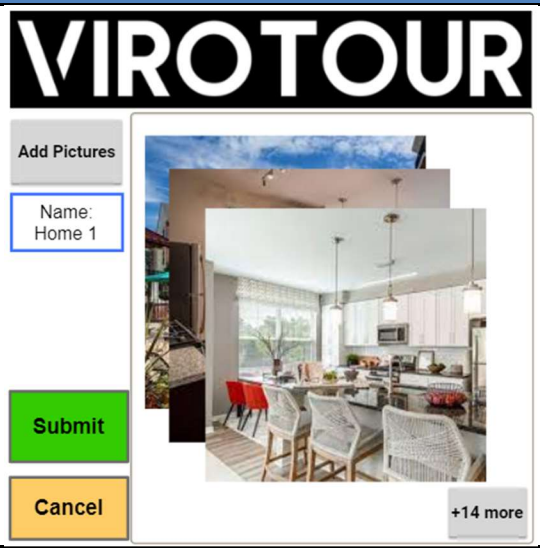
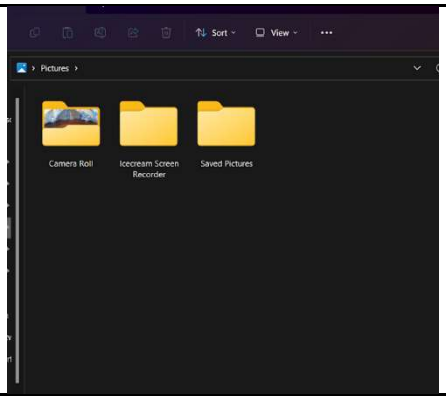
	5. System shows tour with extracted text information	
6. The user clicks the text input to edit text		
7. The user clicks save button to save the change		
	8. The system saves the text.	
9. The user clicks delete button to delete the text from tour.		
	10. The system saves the changes.	

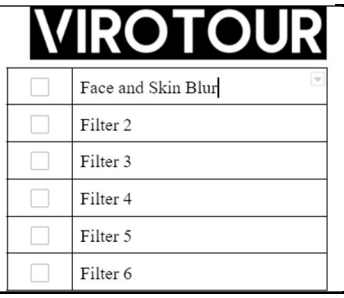
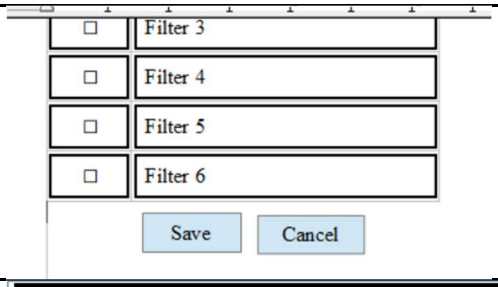
Alternative Course of Events:

Actor	System	Screen
1.The user opens the tour for editing		
	2. no text has been detected in the tour	

3.1.2.2 Use case name: Detect and blur human faces

Summary: The Blur human faces and skin from all images features will detect human faces and skin from an image. After that, detected faces and skin will be automatically blurred and displayed inside the app.
Preconditions: None.
Triggers: The customer uploads photos containing human faces or skin.
Basic course of events (Scenario):
Internal Precondition: None

Actor	System	Screen
1. The User clicks on the "Add Pictures" button.		
	2. The system navigates the user to the uploads page.	
3. The clicks the + button to add photos.		
	4. The system opens up file explorer on the user's device.	
5. The User clicks the photos they want to upload.		
	6. The system stages the photos.	

7. The User clicks the apply filter button.		
	8. The system displays filter options to the user.	
9. The user click the "Face and Skin Blur" option.		
	10. The system applies the filter to the select photos.	
11. The user clicks the "save" button.		
	12. The system saves the images to the database.	

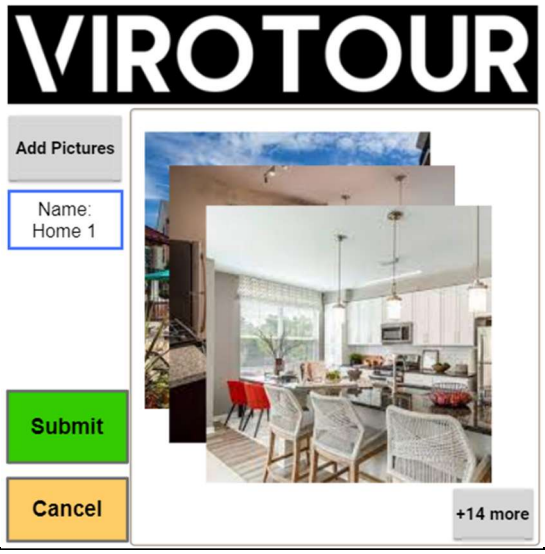
Alternative Course of Events:

Actor	System	Screen
1. The user clicks the "cancel" button.		
	2. The system takes the user to the previous screen.	

3.1.2.3 Use case name: Apply filters

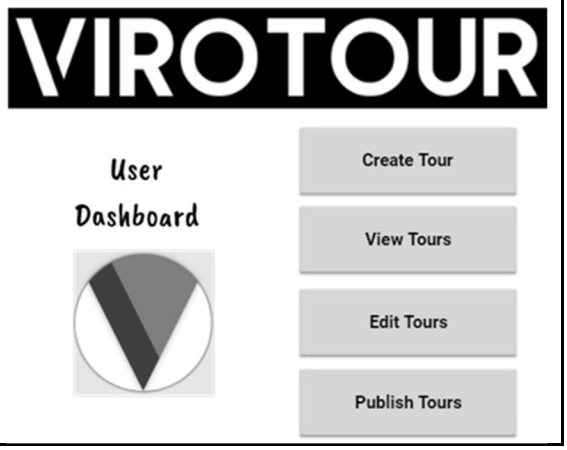
Summary: This use case allows filters to be applied to a panoramic image and the resulting image be stored on the server.
Preconditions: The current panoramas to edit and its respective metadata are stored or about to be stored within a folder on a server.
Triggers: The User has made a request to apply a filter to an image.
Basic course of events (Scenario):
Internal Precondition: ViroTour is running and on the home screen. The data for the tour has previously been uploaded to create the initial tour. The photographer user is ready to interact with the ViroTour application.

Actor	System	Screen
1. User clicks "Create Tour".		
	2. The system takes you to the Create Tours Page.	
3. The User selects the Filter to apply to panoramic creation.		

	4. The filter dropdown has a selection	
5. The user adds pictures, types a name, and clicks “Submit”		

	6. The system updates the database to include the filtered images under a new name and sends a confirmation or error.	

Alternative Course of Events: The user applies filter from the edit screen.

Actor	System	Screen
1. User clicks “Edit Tours”.		

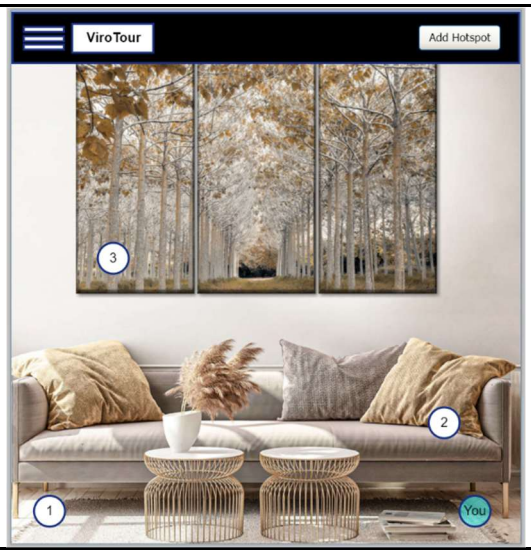
	2. The system takes you to the Edit Tours Page.	
3. The User selects the Filter to apply to panoramic creation.		
	4. The filter dropdown has a selection.	
5. The User clicks the 'Apply Filter' button.		

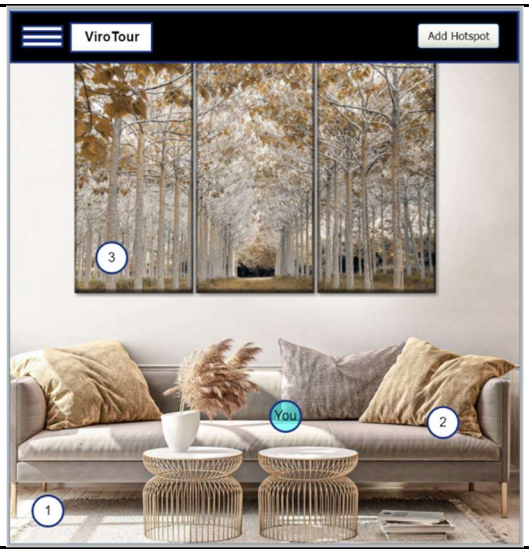
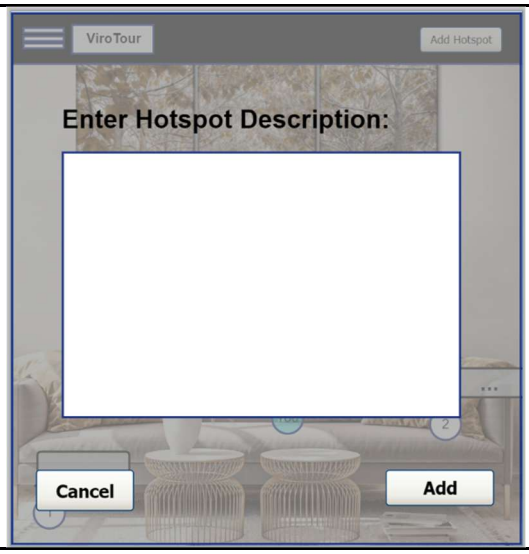
	6. The system updates the database to include the filtered images under a new name and sends a confirmation or error.	

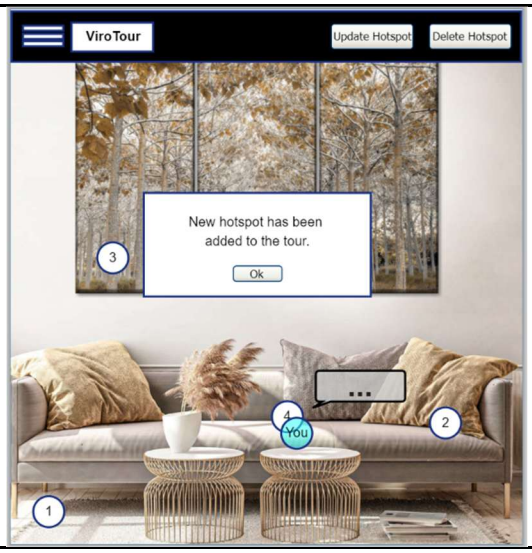
3.1.2.4 Use case name: Add hotspot

Summary: This feature allows users to add a marked point of interest to the application. The hotspot is comprised of a set location and a self-chosen text to display when users view the to-be created hotspot.
Preconditions: None.
Triggers: The user clicks the add hotspot button.
Basic course of events (Scenario):
Internal Precondition:

Actor	System	Screen
1. The user clicks the edit tours button.		

	2. The system navigates the user to a list of available tours to edit.	
3. The user selects the tour they want to edit.		
	4. The system navigates the user into the selected tour.	
5. The user navigates to any location in the panoramic view they want to add a hot spot to.		

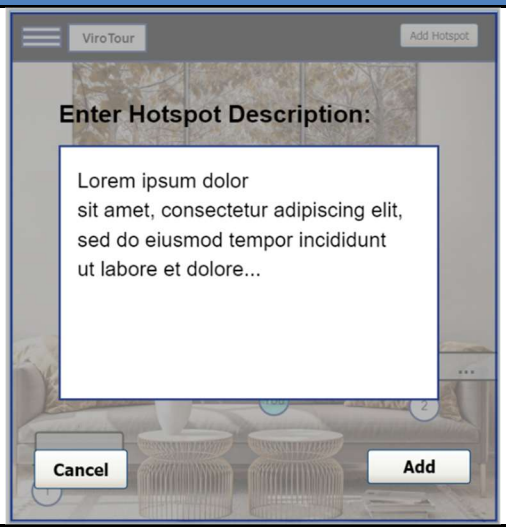
	6. The system moves the user to where they just clicked.	
7. The user clicks the add hotspot button.		
	8. The system prompts a text field for user to add the to-be hotspot's description.	
9. The user fills in the text field that they want displayed with the clickable hotspot.		
10. The user clicks the create hotspot button.		
	11. The system asks user to confirm that they want to create this hotspot.	

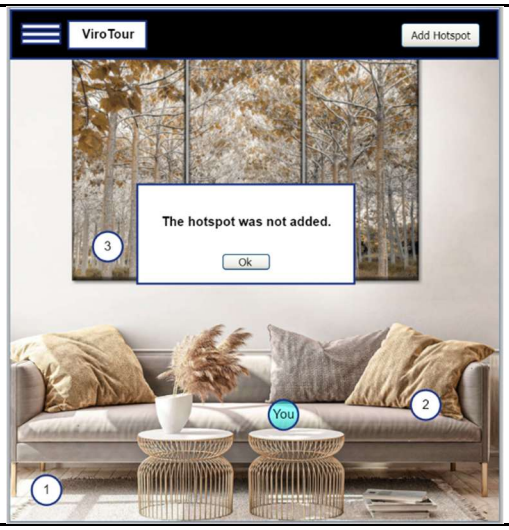
12. The user clicks the confirm button.		
	13. The system creates the hotspot at the user's selected location and attaches the description they just wrote.	

Inter Post Condition:

1. The hotspot (based on the user's selected location and text field) is created and is live on the application.

Alternative Course of Events: The user cancels just before adding the hotspot.

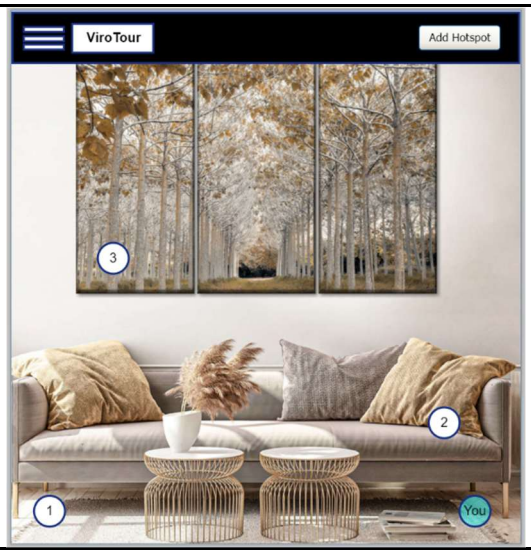
Actor	System	Screen
1. The user clicks the create hotspot (add) button after filling in text field.		
	2. The system asks user to confirm that they want to create this hotspot.	

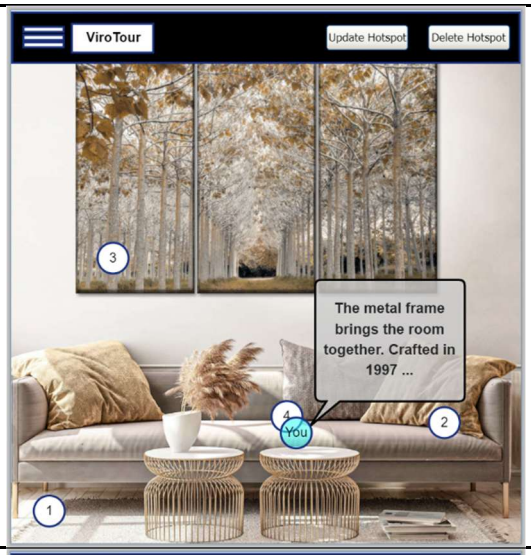
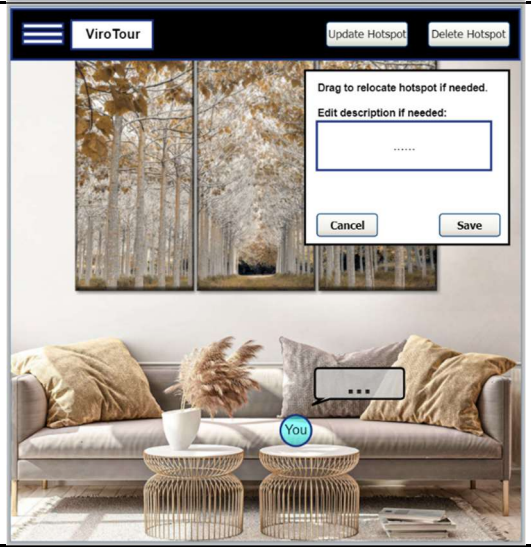
3. The user clicks the cancel button.		
	4. The system displays hotspot creation cancelled and takes user back to panoramic tour view.	

3.1.2.5 Use case name: Edit hotspot

Summary: This feature allows the user to change an already added hotspot's location and description text.
Preconditions: The user must have clicked and be currently on a pre-set hotspot.
Triggers: The user clicks on the edit hotspot button.
Basic course of events (Scenario):
Internal Precondition:

Actor	System	Screen
1. The user clicks the edit tours button.		

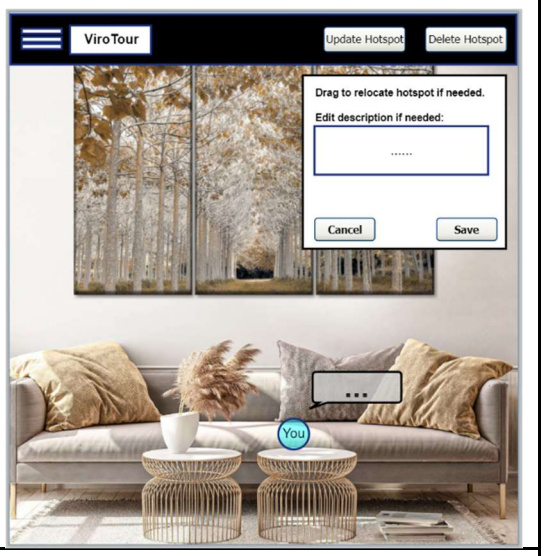
	2. The system navigates the user to a list of available tours to edit.	
3. The user selects the tour they want to edit.		
	4. The system navigates the user into the selected tour.	
5. The user clicks on one of many existing hotspots.		
	6. The system takes the user to the selected hotspot.	

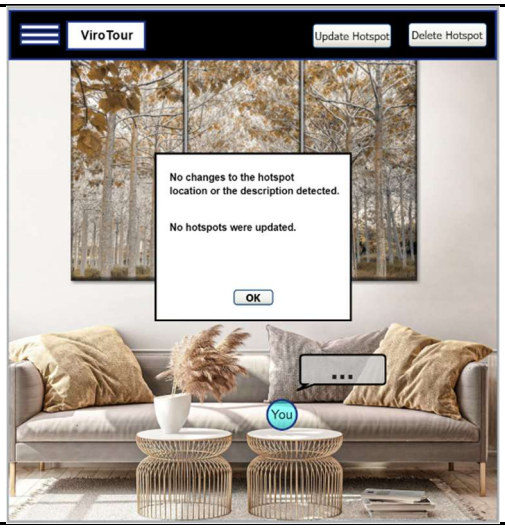
7. The user clicks the update hotspot button.		
	8. The system enables the screen for the user to adjust the hotspot's location by movement input, and an editable description field that belongs to said hotspot (populated with its current text value).	
9. The user (after editing either the location, description text field, or both) clicks on the save button to save any edits.		
	10. The system adjusts the hotspot accordingly and provides user a successful edit confirmation.	

Inter Post Condition:

1. The edited hotspot has changed its location, changed its description text, or both.

Alternative Course of Events: The user tries to confirm the edit hotspot functionality of an existing hotspot, but neither the location or description has changed.

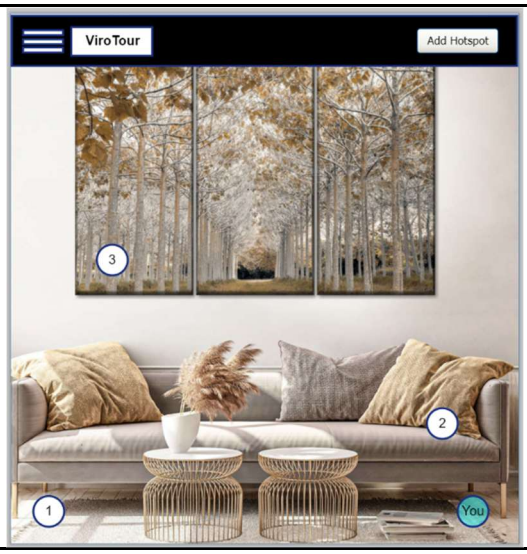
Actor	System	Screen
1. The user clicks the edit hotspot button (after clicking on an existing hotspot).		
	2. The system enables the screen for the user to adjust the hotspot's location by movement input, and an editable description field that belongs to said hotspot (populated with its current text value).	
3. The user leaves the previous hotspot location and description text as is, and clicks the confirm edit button.		

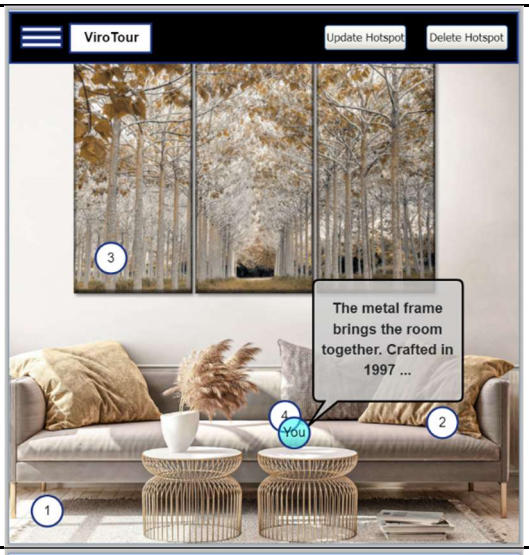
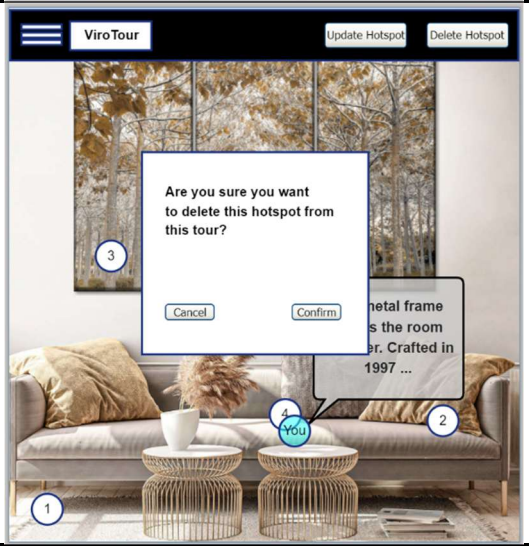
	1. The system goes back to the panoramic tour view and prompts user the current hotspot's values (location and description) have not changed and no edit changes have occurred.	

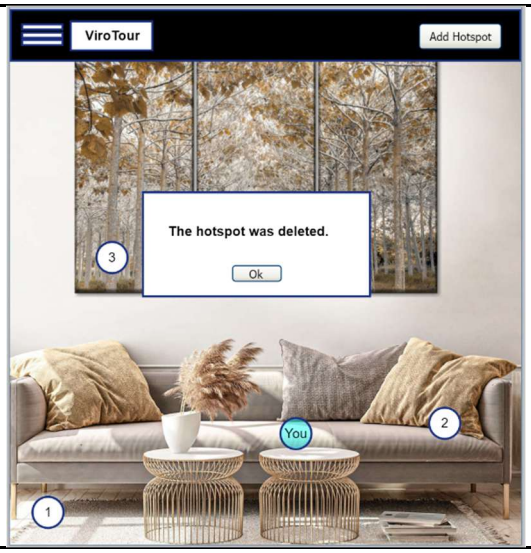
3.1.2.6 Use case name: Delete hotspot

Summary: This feature allows the user to delete a pre-set hotspot from the application and view.
Preconditions: The user must have clicked and be currently on a pre-set hotspot.
Triggers: The user clicks the delete hotspot button.
Basic course of events (Scenario):
Internal Precondition:

Actor	System	Screen
1. The user clicks the edit tours button.		

	2. The system navigates the user to a list of available tours to edit.	
3. The user selects the tour they want to edit.		
	4. The system navigates the user into the selected tour.	
5. The user clicks on one of many existing hotspots.		
	6. The system takes the user to the selected hotspot.	

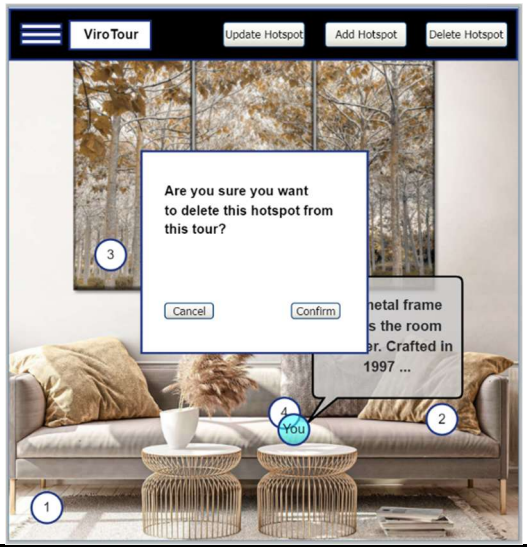
7. The user clicks the delete hotspot button.		
	8. The system asks the user to confirm if they really want to delete the hotspot they are currently at.	
9. The user clicks the confirm button.		

	10. The system deletes the hotspot from the panoramic tour view, providing the user a successful deletion message.	

Inter Post Condition:

1. The system successfully deletes the hotspot (the pre-set location and its description) from the application and view.

Alternative Course of Events: The user changes their mind and cancels their request to delete a hotspot.

Actor	System	Screen
1. The user clicks the delete hotspot button (when they are on a pre-set hotspot).		
	2. The system asks the user to confirm if they really want to delete the hotspot they are currently at.	

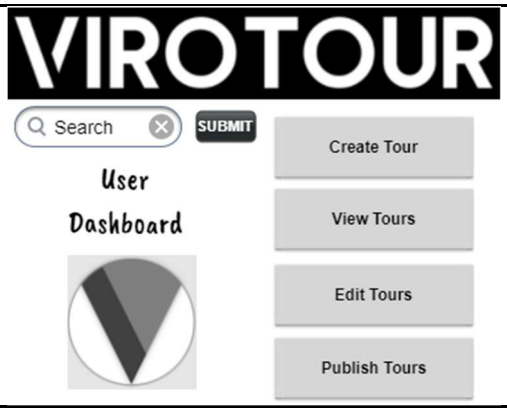
3. The clicks the cancel button.		
	3. The system returns the user back to the panoramic tour view, without any changes or hotspot deletions.	

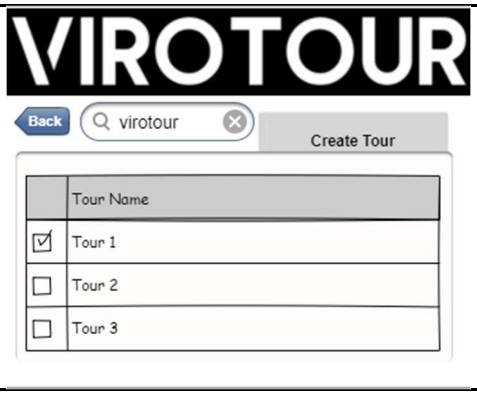
3.1.3 Feature Area: Search

3.1.3.1 Use case name: Ability to search all text from B-2.1

Summary: The application shall allow the search of images by taking a textual query.
Preconditions: The images and their respective metadata should be stored within a folder on a server.
Triggers: The user inputs text in the search box.
Basic course of events (Scenario):
Internal Precondition: The User has the App installed on the mobile device or access to the internet.

Actor	System	Screen
1. The User opens the App on Android, IOS, or Web		
2. The User taps or clicks in the text search box on the screen		
	3. The System focuses the text cursor on the search textbox.	
4. The User enters the search text		

query on the textbox.		
	5. The System shows the "Submit" button beside the search textbox.	
6. The User clicks the "Submit" button.		
	7. The System calls the service to process the image's search based on the entered query text.	
	8. The System displays a popup list of found images on the page/screen by their respective names.	
9. The User views the list of clickable images' names in a tabular format.		

10. The User clicks on a desired image name.		
	11. The System calls the service to process the image's search based based on the selected or clicked image's name from the tabular list.	
	12. The system displays a single image on the page/screen.	
13. The User clicks on the found image to start a virtual tour.		

Alternative Course of Events:

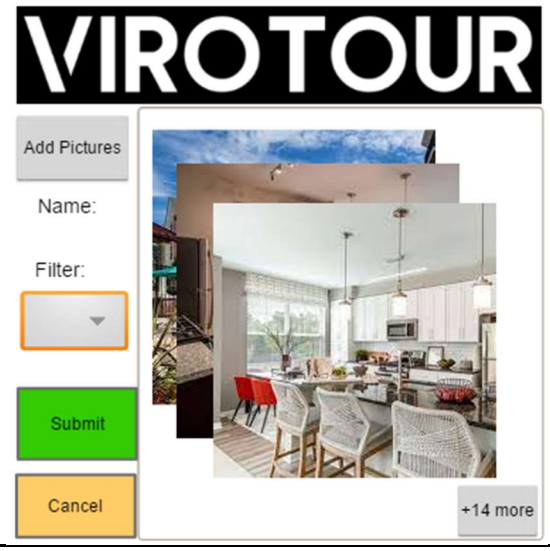
Actor	System	Screen
	7. The System calls the service to process the image's search based on the entered query text.	
	8. The System could not find any image based on the entered query text.	

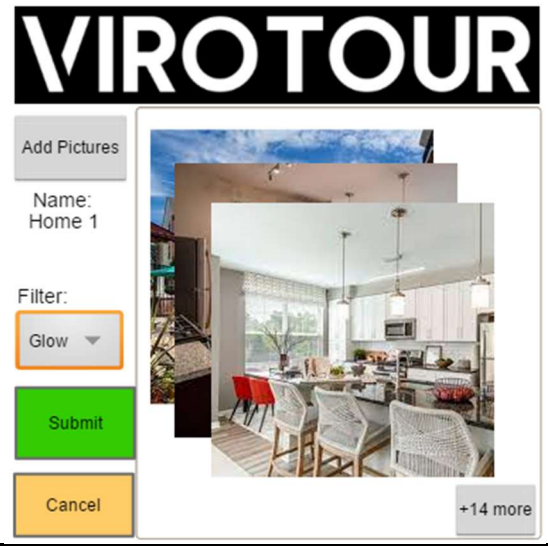
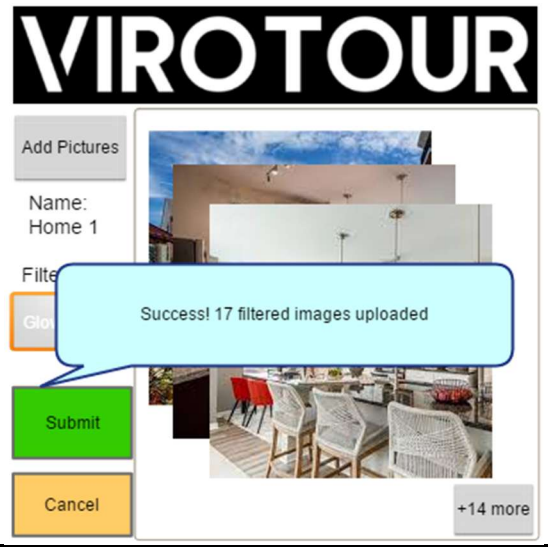
	9. The System displays the “Not found” message.	
10. The User views the “Not found” message on the page/screen.		
11. The User clicks the “X” button to hide the message.		
	12. The System focuses the cursor on the search text box again.	

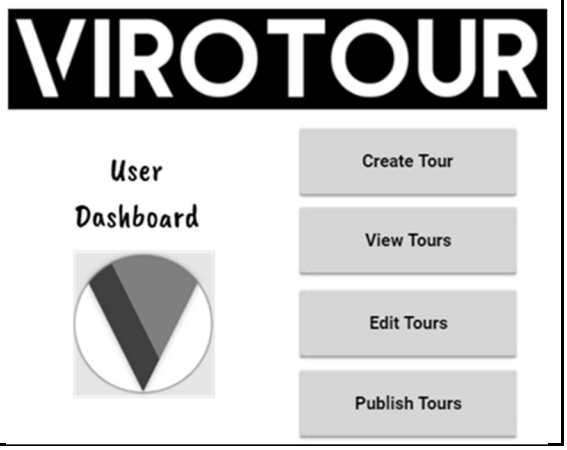
3.1.3.2 *Use case name:* Apply filters

Summary: This use case allows filters to be applied to the panoramic images and the resulting images be stored on the server.
Preconditions: The current image to edit and its respective metadata are stored within a folder on a server.
Triggers: Images were uploaded to the server and panoramic image was created.
Basic course of events (Scenario):
Internal Precondition: The system is running and the photographer user is ready to interact.

Actor	System	Screen

1. User clicks “Create Tour”.		
	2. The system takes you to the Create Tours Page.	
3. The User selects the filter to apply to initial panoramic creation.		

	4. The filter dropdown has a selection.	
5. The user adds images, types a name, and clicks “Submit”		
	6.) The system updates the database to include the filtered images under the name, and sends a confirmation or error.	

	7.) The system navigates to the home screen.	

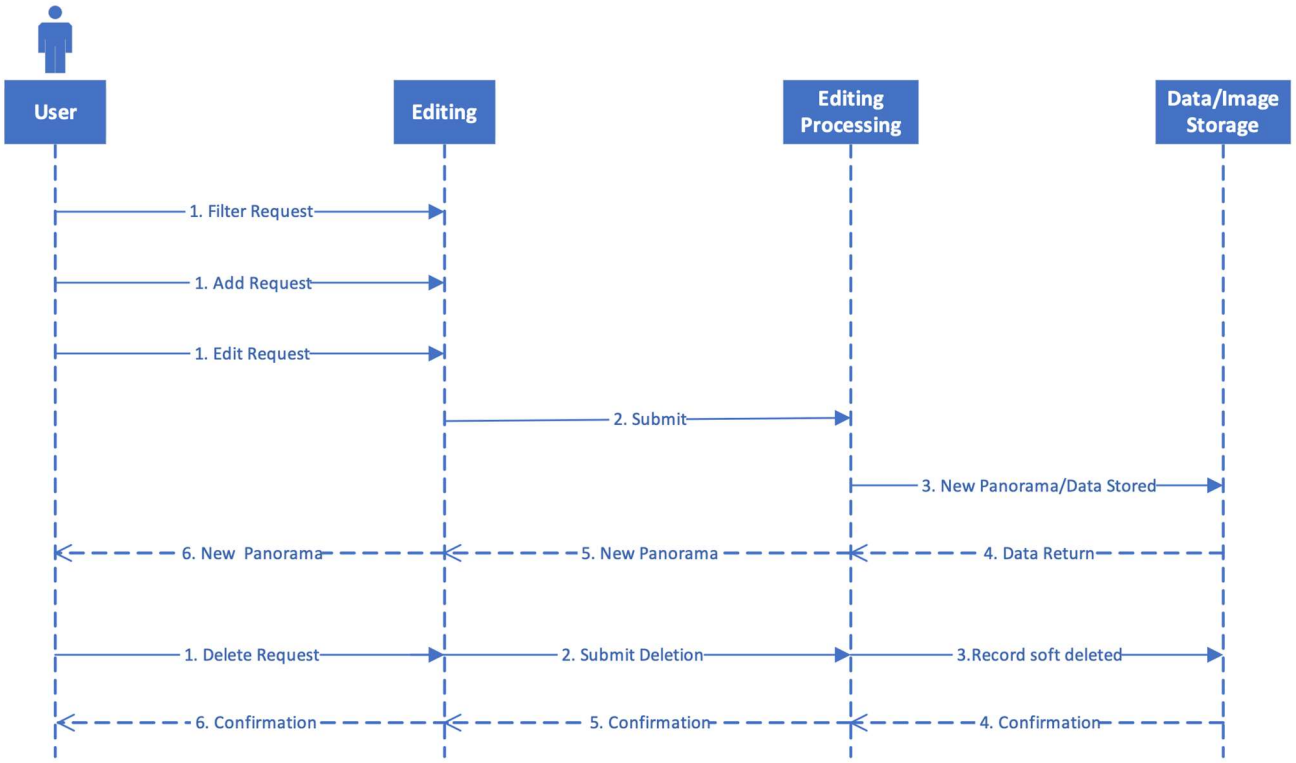
Alternative Course of Events:

Internal Precondition: The system is running and on the Edit Tours page, already has the initial panoramic files created, is on a hotspot, and the photographer user is ready to interact.

Actor	System	Screen
1.) The user selects a filter to apply to the hotspot panoramic.		
	2.) The filter dropdown has a selection.	
3.) The user clicks the 'Apply Filter' button.		

	4.) The system updates the database to include the filtered panoramic if it doesn't have it, and sends a confirmation or error.	

3.1 Editing Module Sequence Diagram



3.2 Supplementary & Non-Functional Requirements

- Support multiple tours within the application.
- Image processing should take less than an hour for 60 images, but it is a flexible limit.
- Image stitching should take less than 10 seconds on a cellphone when taking the four images to create a panoramic image.
- Tour viewing should display images in less than 1 seconds.
- Tour viewing should not freeze for more than 5 seconds when navigating between panoramic images.
- Post-processed images, associated text, any metadata, and hotspot locations can be on a cloud location, but does not necessarily need to be.