
SOFTWARE REQUIREMENTS SPECIFICATION

for

Majuli River Island Tour

Version 2.0

Prepared by : 1. Pradeep Kumar (200101080)
2. Prakhar Pandey (200101081)
3. Pratham Pekamwar (200101087)
4. Agney S Talwarr (200101116)
5. Dheeraj Garg (200101118)
6. Lovish Aggarwal (200101125)

Supervised by : Prof. Samit Bhattacharya

February 21, 2023

Contents

1	Introduction	4
1.1	Purpose	4
1.2	Document Conventions	4
1.3	Intended Audience and Project Scope	4
2	Overall Description	5
2.1	Product Perspective	5
2.2	Product Functions	5
2.3	User Classes and Characteristics	5
2.4	Usability Requirements	5
2.5	Operating Environment	6
3	Interface Requirements	7
3.1	User Interface Requirements	7
3.2	Hardware Requirements	7
3.3	Software Requirements	7
4	Functional Requirements	8
4.1	Tourist	8
4.2	Devotee	12
4.3	Researcher	15
5	Other Non-functional Requirements	22
5.1	Performance requirement	22
5.2	Software Quality Attributes	22

Revision History

Sr. No.	Date	Reasons for changes	Version
1	03/02/2023	Original	1.0
2	21/02/2023	Adding Functional Heirarchy for each user class	2.0

1 Introduction

1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the project. This project attempts to capture and virtually preserve the geography, important sites, ecology and daily life on the river island. We aim to provide a guided tour of Jangraimukh tribal village with support for interactive navigation to the user. This document is primarily intended to be proposed to a customer for their approval and as a reference for developing the first version of the software.

1.2 Document Conventions

Term	Definitions
DESC	Description
RAT	Rational
VR	Virtual Reality

1.3 Intended Audience and Project Scope

The software is meant to be used by common people who would like to have a virtual tour of the island via Google Cardboard. This software will allow the user to navigate and explore seamlessly across the virtual island.

2 Overall Description

2.1 Product Perspective

This product is intended to give a virtual tour of Majuli island to encourage the user to visit while appreciating the culture, heritage and ecology of island. This will also preserve the culture and heritage of the people.

2.2 Product Functions

The software is a VR application that virtually recreates the Majuli island. First, the island is introduced to the user followed by placing the user on the virtual island, where the user can move around. The product leverages the gyroscope sensor in the Android devices used by the user which determines the orientation factor of the user. The application will also allow the user to see where the user is present on the island through mini map (World in Miniature) and give user ways to teleport and move across the island. It also provides audio-visual cues such as ambient music, 3D models and images taken of the real site in the virtual world.

2.3 User Classes and Characteristics

We can classify the users of the software into 3 categories:

Devotees- Users who are interested in visiting religious sites such as temples.

Tourists- Users who wish to visit the river island to learn and experience the culture.

Researchers- Users who want to learn more about the island for academic purposes such as studying the geography and ecology of the river island.

All these users are novice users (layman users) and hence the user centric design would be easy to use instead of having more complexity.

2.4 Usability Requirements

- Guided tour
 - Introduce the user to the island via narration, map and text.
 - The tour should provide all the necessary information about the location and it's cultural, religious and historical significance.
- Immersive experience

- It must provide ambient sound, text, narration, video and other multimedia to provide a truly immersive experience.
 - There should be no lag and responsiveness should be prioritised over quality of image because of computational limits of a mobile phone.
- Navigation
 - It must allow seamless navigation from one location to another and a map to allow users to identify their location.

2.5 Operating Environment

The software will operate on Android mobile phones using Google Cardboard. The Android device then will use the gyroscope to take various parameters like device orientation and other things into account to determine the orientation of user's head to show appropriate view at the current location.

3 Interface Requirements

3.1 User Interface Requirements

User Interface should be intuitive and easy-to-use with a clear display of various options and information. It should provide smooth interactions and quick response time for a fast and responsive virtual environment.

3.2 Hardware Requirements

The user's mobile device should have gyroscope. User would also require a Google Cardboard. Also, the devices should have enough specification to handle the computation required and enough space to be able to run the software.

3.3 Software Requirements

User would be using Unity software to have a virtual view of the island. Smartphone should be running Android 4.1 or later. Since, Unity Cardboard integration requires Android Lollipop or greater.

4 Functional Requirements

4.1 Tourist

1. Introduction with Ambient sound

1.1 Intro

- ID: FR_1.1
- Input: Start button
- Output: Rotating globe view
- DESC: This would zoom in on the location of the island.

1.2 Skip Intro

- ID: FR_1.2
- Input: Google Cardboard Input
- Output: User Guide
- DESC: This would skip the intro and display user guide.

1.3 Voice over

- ID: FR_1.3
- Input: Start button
- Output: Human voice
- DESC: Voice would give introductory description of the island.

2. User Guide

2.1 Manual

- ID: FR_2.1
- Input: Google Cardboard Input
- Output: User Manual
- DESC: This would display user control guide.

2.2 Skip Manual

- ID: FR_2.2
- Input: Google Cardboard Input
- Output: Initial location

- DESC: This would skip user control guide and take user to the initial location.

3. Walk-through

3.1 Walk-through of Island

3.1.1 Visuals

- ID: FR_3.1.1
- Input: Coordinates of current location
- Output: Video/Images of current location
- DESC: Corresponding visuals according to the user's location would be displayed.

3.1.2 Voice over

- ID: FR_3.1.2
- Input: Coordinates of current location
- Output: Human voice
- DESC: Voice would give introductory description about the current location to the user.

3.2 Walk-through of village

3.2.1 Manual

- ID: FR_3.2.1
- Input: Google Cardboard Input
- Output: User Manual
- DESC: This would display user control guide.

3.2.2 Skip Manual

- ID: FR_3.2.2
- Input: Google Cardboard Input
- Output: Village Initial location
- DESC: This would skip user control guide and take user to the village initial location.

3.2.3 Visuals

- ID: FR_3.2.3
- Input: Coordinates of village
- Output: Video/Images/3d Model of village
- DESC: Corresponding visuals of the village.

3.2.4 Voice over

- ID: FR_3.2.4
- Input: Coordinates of location in village
- Output: Human voice
- DESC: Voice would give introductory description of the village.

3.2.5 Move Around

- ID: FR_3.2.5
- Input: Coordinates of current location inside village
- Output: New location inside village
- DESC: This will move the user to the new location inside village.

3.3 360 degree Viewing of Locations

- ID: FR_3.3
- Input: Orientation details of device
- Output: Corresponding view of that place
- DESC: Turning our heads would adjust the view of location accordingly.

3.4 Information about location

- ID: FR_3.4
- Input: Coordinates of current location
- Output: Corresponding description of the location
- DESC: Displays information related to current location

4. Mini Map (World-in-Miniature)

4.1 Display Map

- ID: FR_4.1
- Input: Coordinates of current location
- Output: Map of current location
- DESC: This will allow the user to know nearby location on the virtual island and help in navigating the virtual space.

4.2 Pointer

- ID: FR_4.2
- Input: Coordinates of current location
- Output: Pointer of current location
- DESC: This will allow the user to know it's location on the virtual island.

5. Teleportation

5.1 Open Main Map

- ID: FR_5.1
- Input: Google cardboard input
- Output: Main map
- DESC: This will open main map of the island.

5.2 Change Main Map Pointer

- ID: FR_5.2
- Input: Coordinates of current location and google cardboard input
- Output: Pointer to next location
- DESC: Pointer will point to the next location on main map.

5.3 Close Main Map

- ID: FR_5.3
- Input: Google cardboard input
- Output: New location
- DESC: This will close main map and user will be teleported to the location pointed by main map pointer.

6. **Exit**

6.1 Exit

- ID: FR_6.1
- Input: Google Cardboard Input
- Output: Closed app
- DESC: Closes the app

4.2 Devotee

1. Introduction with Ambient sound

1.1 Intro

- ID: FR_1.1
- Input: Start button
- Output: Rotating globe view
- DESC: This would zoom in on the location of the island.

1.2 Skip Intro

- ID: FR_1.2
- Input: Google Cardboard Input
- Output: User Guide
- DESC: This would skip the intro and display user guide.

1.3 Voice over

- ID: FR_1.3
- Input: Start button
- Output: Human voice
- DESC: Voice would give introductory description of the island.

2. Simplified User Guide

2.1 Manual

- ID: FR_2.1
- Input: Google Cardboard Input
- Output: User Manual
- DESC: This would display user control guide.

2.2 Skip Manual

- ID: FR_2.2
- Input: Google Cardboard Input
- Output: Initial location
- DESC: This would skip user control guide and take user to the initial location which is temple for this user.

3. Walk-through

3.1 Walk-through of Island

3.1.1 Visuals

- ID: FR_3.1.1
- Input: Coordinates of current location
- Output: Video/Images of current location
- DESC: Corresponding visuals according to the user's location would be displayed.

3.1.2 Voice over

- ID: FR_3.1.2
- Input: Coordinates of current location
- Output: Human voice
- DESC: Voice would give introductory description about the current location to the user.

3.2 Walk-through of village

3.2.1 Manual

- ID: FR_3.2.1
- Input: Google Cardboard Input
- Output: User Manual
- DESC: This would display user control guide.

3.2.2 Skip Manual

- ID: FR_3.2.2
- Input: Google Cardboard Input
- Output: Village Initial location
- DESC: This would skip user control guide and take user to the village initial location.

3.2.3 Visuals

- ID: FR_3.2.3
- Input: Coordinates of location in village
- Output: Video/Images/3d Model of village
- DESC: Corresponding visuals according to location in village.

3.2.4 Voice over

- ID: FR_3.2.4
- Input: Coordinates of location in village
- Output: Human voice
- DESC: Voice would give introductory description of the village.

3.2.5 Move Around

- ID: FR_3.2.5
- Input: Coordinates of current location inside village
- Output: New location inside village
- DESC: This will move the user to the new location inside village.

3.3 360 degree Viewing of Locations

- ID: FR_3.3
- Input: Orientation details of device
- Output: Corresponding view of that place
- DESC: Turning our heads would adjust the view of location accordingly.

4. Mini Map (World-in-Miniature)

4.1 Display Map

- ID: FR_4.1
- Input: Coordinates of current location
- Output: Map of current location
- DESC: This will allow the user to know nearby location on the virtual island and help in navigating the virtual space.

4.2 Pointer

- ID: FR_4.2
- Input: Coordinates of current location
- Output: Pointer of current location
- DESC: This will allow the user to know it's location on the virtual island.

5. Teleportation

5.1 Teleport

- ID: FR_5.1
- Input: Coordinates of current location and google cardboard input
- Output: Next location
- DESC: This will teleport user to the next location.

6. Exit

6.1 Exit

- ID: FR_6.1
- Input: Google Cardboard Input
- Output: Closed app
- DESC: Closes the app

4.3 Researcher

1. Introduction with Ambient sound

1.1 Intro

- ID: FR_1.1
- Input: Start button
- Output: Rotating globe view
- DESC: This would zoom in on the location of the island.

1.2 Skip Intro

- ID: FR_1.2
- Input: Google Cardboard Input
- Output: User Guide
- DESC: This would skip the intro and display user guide.

1.3 Voice over

- ID: FR_1.3
- Input: Start button
- Output: Human voice
- DESC: Voice would give introductory description of the island.

2. User Guide

2.1 Manual

- ID: FR_2.1
- Input: Google Cardboard Input
- Output: User Manual
- DESC: This would display user control guide.

2.2 Skip Manual

- ID: FR_2.2
- Input: Google Cardboard Input
- Output: Initial location
- DESC: This would skip user control guide and take user to the initial location.

3. Walk-through

3.1 Walk-through of Island

3.1.1 Visuals

- ID: FR_3.1.1
- Input: Coordinates of current location
- Output: Video/Images of current location
- DESC: Corresponding visuals according to the user's location would be displayed.

3.1.2 Detailed Voice over

- ID: FR_3.1.2
- Input: Coordinates of current location
- Output: Human voice
- DESC: Voice would give introductory description about the current location to the user.

3.2 Walk-through of village

3.2.1 Manual

- ID: FR_3.2.1
- Input: Google Cardboard Input
- Output: User Manual
- DESC: This would display user control guide.

3.2.2 Skip Manual

- ID: FR_3.2.2
- Input: Google Cardboard Input
- Output: village Initial location
- DESC: This would skip user control guide and take user to the village initial location.

3.2.3 Visuals

- ID: FR_3.2.3
- Input: Coordinates of current location in village
- Output: Video/Images/3d Model of village
- DESC: Corresponding visuals according to the location in village.

3.2.4 Detailed Voice over

- ID: FR_3.2.4
- Input: Coordinates of village
- Output: Human voice
- DESC: Voice would give introductory description of the village.

3.2.5 Move Around

- ID: FR_3.2.5
- Input: Coordinates of current location inside village
- Output: New location inside village
- DESC: This will move the user to the new location inside village.

3.3 360 degree Viewing of Locations

- ID: FR_3.3
- Input: Orientation details of device
- Output: Corresponding view of that place
- DESC: Turning our heads would adjust the view of location accordingly.

3.4 Detailed Information about location

- ID: FR_3.4
- Input: Coordinates of current location
- Output: Corresponding description of the location
- DESC: Displays information related to current location

4. Mini Map (**World-in-Miniature**)

4.1 Display Map

- ID: FR_4.1
- Input: Coordinates of current location
- Output: Map of current location
- DESC: This will allow the user to know nearby location on the virtual island and help in navigating the virtual space.

4.2 Pointer

- ID: FR_4.2
- Input: Coordinates of current location
- Output: Pointer of current location
- DESC: This will allow the user to know it's location on the virtual island.

5. Teleportation

5.1 Open Main Map

- ID: FR_5.1
- Input: Google cardboard input
- Output: Main map
- DESC: This will open main map of the island.

5.2 Change Main Map Pointer

- ID: FR_5.2
- Input: Coordinates of current location and google cardboard input
- Output: Pointer to next location
- DESC: Pointer will point to the next location on main map.

5.3 Close Main Map

- ID: FR_5.3
- Input: Google cardboard input
- Output: New location
- DESC: This will close main map and user will be teleported to the location pointed by main map pointer.

6. Calendar

6.1 Open Calendar

- ID: FR_6.1
- Input: Main map and Google Cardboard Input
- Output: Calendar
- DESC: This will display calendar on the screen which details different festivals celebrated in the region.

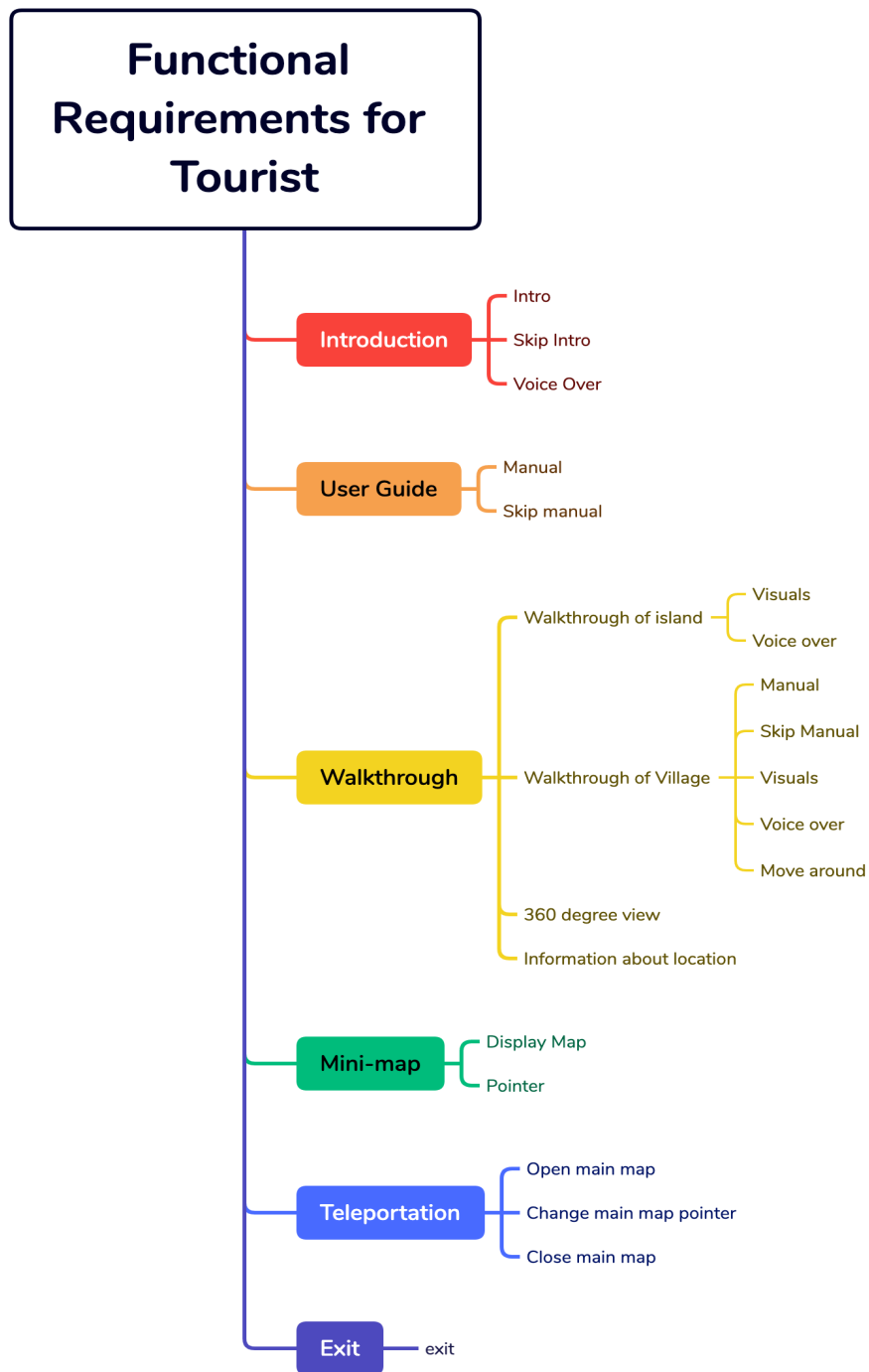
6.2 Close Calendar

- ID: FR_6.2
- Input: Google Cardboard Input
- Output: Main map
- DESC: This will close the calendar and display the main map.

7. Exit

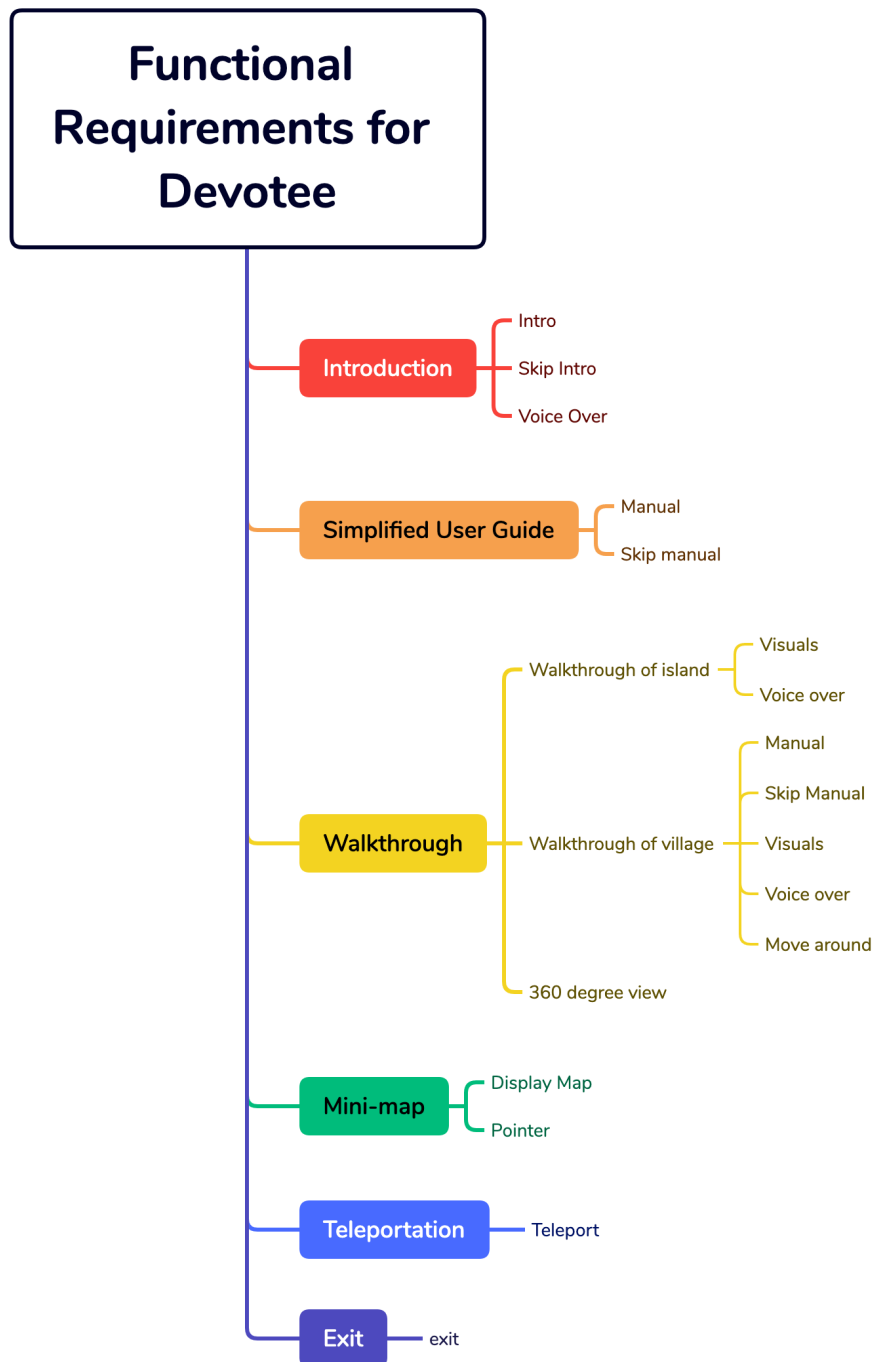
7.1 Exit

- ID: FR_7.1
- Input: Google Cardboard Input
- Output: Closed app
- DESC: Closes the app



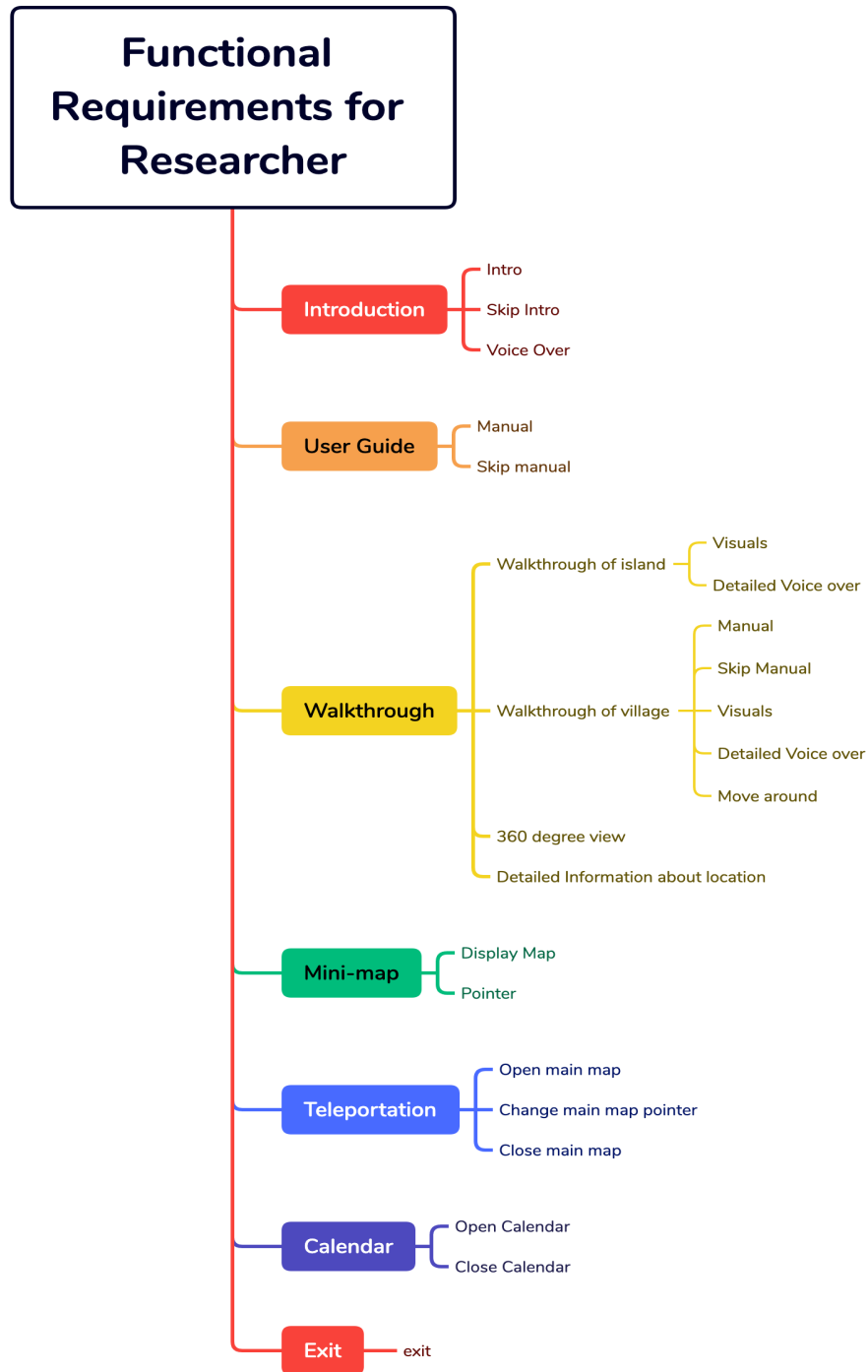
Presented with xmind

Figure 4.1: Functional Requirements flowchart.



Presented with xmind

Figure 4.2: Functional Requirements flowchart.



Presented with xmind

Figure 4.3: Functional Requirements flowchart.

5 Other Non-functional Requirements

5.1 Performance requirement

- **Smooth Rendering of Images** - The processing would be done on mobile phone, the processing power would be lower than other devices like laptops or PCs. Hence, optimizations need to be made and the image quality needs to be reduced enough that there are no glitches and slowdown of the software during use.
- **No Lag in Teleportation** - User would use teleportation to view various places on the virtual island. Hence, there should be smooth visuals shown while the user is teleporting from one place to other.

5.2 Software Quality Attributes

- **Code Modularity** - Code should be modular and well written, so as to facilitate further addition and implementation of new features and improvements.
- System should be robust enough to not crash during runtime.