SOFTWARE REQUIREMENTS SPECIFICATION

for

MAJULI ISLAND SIMULATION

Version 1.0

Prepared by: Anurag Ravi(200101017)
Bhanu Rajput(200101024)
Gaurav Kumar(200101033)
Himanshu Shekhar(200101042)
Himanshu Yadav(200101043)
Ravi Kumar(200101089)

Supervised by: Prof. Samit Bhattacharya

February 1, 2023

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the information system about “Majuli Island”. It will illustrate the purpose and complete declaration for the development of the system along with the system constraints. This document is primarily intended to be proposed to a customer (especially Prof. Samit Bhattacharya) for their approval and as a reference for developing the first version of the system for the development team.

1.2 Scope

The software will cover information on Majuli Island, its culture, geography, attractions, and activities. It will also provide information on how to reach the island and the various modes of transportation available.

1.3 Overview

Overview outlines the main goals and objectives of the software system designed to provide information about Majuli Island, a river island in the Brahmaputra river in Assam, India.

1.4 Document Convention

DESC	Description
RAT	Rational
OTP	One Time Password
Tourist	One who comes for only cultural visit
Researcher	Person who collects information about a particular subject
Local Resident	A person who lives in a particular place for tourists' benefit and provides suggestions to the administrator.
Administrator	Person who manages the software.

1.5 Intended Audience

This would likely be tourists, travellers, and anyone interested in learning about the culture, history, and attractions of Majuli island. This includes people planning a trip to the island, as well as students, researchers, or anyone who wants to learn more about the area. The information system can be accessible to a broad audience, including those who are not familiar with the region.

It should be designed to be user-friendly with clear and concise information, images, maps, and other resources that make it easy for users to learn about Majuli Island.

1.6 Intended Use

The information system could be used as a planning tool for people visiting the island, with information on how to get there, what to see and do, and where to stay. It could also be used by students and researchers to learn about the history and culture of the region with information on the local traditions, customs, and festivals.

The information system could also provide a platform for users to share their experiences and insights about the island. This could include user-generated content such as photos, videos, and reviews. This would help create a community around the island and provide a valuable resource for future visitors.

2 Overall Description

2.1 Product Perspective/Objectives

The Majuli Island project is intended to provide a comprehensive and user-friendly information system about the island. The information system should be accessible to a broad audience and provide information on a range of topics related to Majuli Island including geography, culture, history, and attractions.

To provide accurate and up-to-date information about Majuli Island.

To make the information system accessible and user-friendly with clear and concise information, images, maps, and other resources.

To encourage exploration and appreciation of the region by providing information on the local traditions, customs, and festivals.

To provide a platform for users to share their experiences and insights about the island.

To create a community around Majuli Island with user-generated content such as photos, videos, and reviews.

To be a valuable resource for tourists, travellers, students, researchers, and anyone interested in knowing about the area.

To promote tourism to the island while also highlighting the need to protect and preserve the unique environment and culture of the region.

2.2 Product Functions/System Overview

Product Functions:

The purpose of the software solution being developed for the Majuli Island project is to provide a comprehensive management system for the island's resources and infrastructure. The software is expected to automate various tasks related to resource management, including resource allocation, tracking, and utilization. The goal of the project is to improve efficiency, reduce waste, and ensure equitable distribution of resources for the benefit of the island's residents.

The software system for the Majuli Island project will consist of several modules, each responsible for a specific aspect of resource management. The main components of the system will include:

- Resource Inventory Management Module
- Resource Allocation Module
- Resource Tracking Module
- Resource Utilization Report Module

2.3 User Class, Functions and Interfaces

The user classes, functions and interfaces would be defined based on the requirements gathered from stakeholders such as tourists, local residents, and administrators. The specific user classes, functions and interfaces would be determined based on the goals and objectives of the project.

For example, some potential user classes could include:

- **Tourists:** people who visit the island for leisure or educational purposes

- **Local Residents:** people who live on the island which are affected by tourism activities and claim any improvement by requesting any multimedia content of that area and also about software-based requirements up to their depth of knowledge. They can also provide more details and cultural details to add to the software.
- **Administrators:** people who are allowed to modify maps can manage images/videos in the app, can see incoming traffic on the software, can monitor usage, and can configure project settings.

Some potential functions for the project could include:

- **Tourist functions :** Search for accommodations, activities, and tour packages, make bookings, view itineraries, provide feedback
- **Resident functions :** provide feedback on tourism activities, report issues, view information about the island
- **Administrator Functions :** manage any modification in the map and can configure any settings in the app.

The interfaces for these functions could include web-based portals, mobile applications, or desktop software.

2.4 Operating and Technical Environment and System and Performance Requirements

For an SRS (Software Requirements Specification) project on Majuli Island, it is important to consider the following factors in the operating environment:

Operating Environment:

- Limited connectivity: Access to the internet and other communication technologies may be limited, impacting the ability to collect and analyze data.
- High humidity and temperature: The tropical climate on the island may impact the performance and longevity of electronic equipment used for the project.
- Cultural sensitivity: Understanding and respecting local cultural norms and values is essential in order to ensure the successful implementation of the project.

Considering these factors in the operating environment on Majuli Island will help ensure the successful implementation of an SRS project in the region.

Technical Environment:

- Software and hardware compatibility with existing systems and equipment
- Ability to integrate with existing data management systems and processes
- Compliance with relevant regulations and standards
- Cybersecurity measures to protect sensitive data and systems

System Requirements:

- User-friendly interface for easy operation and management
- Robust data backup and disaster recovery systems
- Adequate storage capacity for data and information
- Scalability to accommodate future growth and changes

Performance Requirements:

- Efficient and effective data management and analysis
- Ability to handle high volumes of data in real-time
- Response times within acceptable limits
- High availability and reliability of the system
- These requirements will vary based on the specific needs and goals of the project, but they provide a general framework for developing a comprehensive and effective operating environment.

2.5 Design and Implementation Constraints

The design of the software solution will need to consider several constraints that could impact the functionality and usability of the system. These constraints may include:

- **Technical Constraints:** Technical limitations, such as the availability of hardware, software, and connectivity, must be taken into account when designing the software.
- **Time Constraints:** The project must be delivered within the specified time frame, and the design must take into account any deadlines that must be met.
- **User Constraints:** The software must be designed to meet the needs of its intended users, which may include administrators, residents, and other stakeholders. The design must take into account their technical proficiency, preferences, and limitations.

The implementation of the software solution for the Majuli Island project will also face several constraints that could impact the success of the project. These constraints may include:

- **Technical Constraints:** The availability of hardware, software, and connectivity could impact the implementation of the system.
- **Resource Constraints:** The availability of personnel, including developers, testers, and support staff will impact the implementation of the system.
- **Time Constraints:** The implementation must be completed within the specified timeframe and any delays could impact the overall success of the project.
- **User Acceptance:** The success of the project will depend on user acceptance of the system. The implementation must take into account user feedback and ensure that the system meets their needs and expectations.

2.6 Assumption and Dependencies

Assumptions and dependencies are an important part of any project including a project. They help to identify the conditions and factors that are necessary for the project to be successful and to inform project planning and decision-making. Some common assumptions and dependencies for this type of project might include:

Assumptions:

- Adequate funding and resources will be available
- The local community will support and participate in the project
- Regulatory requirements will be met and permits obtained
- The project team will have access to the necessary expertise and skills
- Weather conditions will not significantly impact the project timeline

Dependencies:

- Availability of materials and equipment
- Cooperation and support from government agencies and local organizations
- Cooperation from stakeholders and community members
- Compliance with relevant regulations and standards

By understanding these assumptions and dependencies, project managers can proactively address potential risks and ensure that the project stays on track. Additionally, regularly reviewing and updating these assumptions and dependencies can help to identify new risks and opportunities, and ensure that the project remains relevant and effective over time

3 External Interface Requirements

3.1 User Interfaces

The UI (User Interface) of a project would play a critical role in the success of the project by allowing users to interact with the system effectively and efficiently. The design of the UI should take into account the specific needs and requirements of the users, as well as the technical capabilities of the system.

By designing a user-friendly and effective UI, project managers can help ensure the success of the project by making it easier for users to interact with the system and access the information they need.

3.2 Input Interfaces

The input interface of a project refers to the way in which data and information are entered into the system. A well-designed input interface can help ensure the accuracy and completeness of the data as well as make the process of entering data more efficient and user-friendly. Some key considerations for the input interface might include:

- **User-friendly design:** The input interface should be designed to be easy to use, even for those with limited technical skills. This might involve using clear and simple language, as well as visual aids such as graphics and icons to support understanding.

- **Data validation:** The input interface should include data validation checks to help ensure that the data entered into the system is accurate and complete. For example, the system might check that a required field has been filled in or that a value falls within a specified range.

- **Data entry methods:** The input interface should allow users to enter data in a variety of ways such as through a web form, spreadsheet or mobile app.

- **Integration with other systems:** The input interface should be designed to integrate with other systems such as data management systems and geographic information systems to ensure that the data entered is consistent and up-to-date.

- **Automated data collection:** The input interface should allow for automated data collection where possible, such as through the use of sensors or APIs. This can help to reduce the time and effort required for data entry, as well as increase the accuracy of the data.

3.3 Output Interfaces

The output interface of a project refers to the way in which data and information are made available to users. A well-designed output interface can help users access the information they need to analyze and understand the data and make informed decisions.

3.4 Integration with other systems

Integration with other systems refers to the process of connecting and combining different software systems or applications to work together seamlessly and share data. A project could involve integrating various systems such as geographic information systems (GIS), remote sensing technologies, weather forecasting systems, and databases for storing and managing the collected data. The aim of integration is to improve the efficiency, accuracy and overall effectiveness of the project by bringing together diverse sources of information.

The specific integration methods and technologies used would depend on the particular requirements of the project and the systems being integrated.

3.5 Hardware Interface

Hardware interface refers to the connection between the software and hardware components of a system. In the context of a project, the hardware interface could include the interface between the software and the sensors and other equipment used to collect data and monitor the island's resources.

Examples of hardware components that might be integrated with the software include environmental sensors (e.g., temperature, humidity, and rainfall), remote sensing equipment (e.g., drones or satellites), GPS devices, and data loggers. The hardware interface is important to ensure that the software can receive and process the data collected by these devices and convert it into usable information. The specific hardware interface requirements and technologies would depend on the particular requirements of the project and the equipment being used.

3.6 Software Interface

The software interface refers to the connection between the different software systems and applications used in the project. This interface is important to ensure that the different software components can interact with each other and share data effectively.

Examples of software components that might be integrated into the project include geographic information systems (GIS), remote sensing software, data analysis tools and databases for storing and managing the collected data. The software interface could be achieved through APIs (Application Programming Interfaces) which allow for data transfer between the systems. Additionally, it could also involve using standard protocols and data formats to ensure compatibility between different software components.

The specific software interface requirements and technologies would depend on the particular requirements of the project and the software systems being used. It is essential to ensure that the software interface is designed to meet the needs of the project and support the efficient and effective integration of the different software components.

3.7 Communication Interface

The communication interface refers to the means by which the different components of the system, including hardware, software and personnel can exchange information and data.

Examples of communication interfaces that might be used in the project include wireless networks for transmitting data from sensors and other equipment, internet connectivity for accessing and sharing data and information and direct connections between software systems for data transfer. The communication interface is crucial for ensuring the effective and efficient operation of the system as well as the accuracy and reliability of the data collected.

The specific communication interface requirements and technologies would depend on the particular requirements of the project including the types of data being collected and size and complexity of the system and the availability of communication resources. It is important to ensure that the communication interface is secure and reliable to protect sensitive data and ensure the integrity of the project.

4 Functional requirements

4.1 User Class 1: Tourist

4.1.1 Authentication

4.1.1.1 Create account

ID: FNR_TST_1.1
INPUT: user information
OUTPUT: validation prompt message
DESC: create a new account using user details and display the account creation message.

4.1.1.2 Login

ID: FNR_TST_1.2
INPUT: user credentials
OUTPUT: login status
DESC: verify user credentials and provide access to user data.

4.1.1.3 Send One Time Password

ID: FNR_TST_1.3
INPUT: user email or mobile
OUTPUT: Send one-time password
DESC: An OTP is sent to the user's email or mobile number.

4.1.1.4 Set New Password

ID: FNR_TST_1.4
INPUT: OTP and new password
OUTPUT: password change verdict
DESC: OTP is verified, and the new password is entered and updated.

4.1.1.5 Update Password

ID: FNR_TST_1.5
INPUT: old password and new password
OUTPUT: password change verdict
DESC: old password is verified, and the new password is entered and updated.

4.1.2 Navigation

ID: FNR_TST_2.1
INPUT: mouse click on the navigation button or arrow keys on the keyboard
OUTPUT: show the view from a near point in the direction of input
DESC: the user's location gets shifted in the direction where he clicks/presses keys to move.

4.1.3 Map View

4.1.3.1 Open map

ID: FNR_TST_3.1
INPUT: button click
OUTPUT: map view with current location marker and landmarks
DESC: on getting a click on the map icon in 3d view switch to map view.

4.1.3.2 Teleportation

ID: FNR_TST_3.2
INPUT: a location in map view
OUTPUT: 3d view of the new location
DESC: teleport to the coordinates of the new location provided by the user.

4.1.3.3 Exit Map

ID: FNR_TST_3.3
INPUT: a click on ESC or close map button
OUTPUT: 3D view
DESC: When a user clicks on ESC or a button, the map will be closed and the system switches back to the 3d view.

4.1.4 Projection of Video

4.1.4.1 Start Video

ID: FNR_TST_4.1
INPUT: Button click
OUTPUT: A video
DESC: On clicking, the video will be projected on the wall

4.1.4.2 Play/Pause

ID: FNR_TST_4.2
INPUT: Button click
OUTPUT: Video in play or paused state
DESC: Video will be played or paused depending on the button click.

4.1.4.3 Seek In Video

ID: FNR_TST_4.3
INPUT: Time stamp
OUTPUT: Video at the selected time stamp.
DESC: The user moves/skips to the new position of the video.

4.1.4.4 Exit before video completion

ID: FNR_TST_4.4
INPUT: Button click or ESC button
OUTPUT: 3d view.
DESC: The video will be closed before its completion and the user gets switched back to the 3d view.

4.1.5 Settings

4.1.5.1 Open Settings

ID: FNR_TST_5.1
INPUT: A click on the settings icon
OUTPUT: settings view
DESC: on clicking on the settings icon the user switches from 3d view to settings view and vice versa

4.1.5.2 Set Sound Level

ID: FNR_TST_5.2
INPUT: slider movement
OUTPUT: sound level
DESC: Higher the level of the slider, the higher the level of sound

4.1.5.3 Compass On/Off

ID: FNR_TST_5.3
INPUT: a click on toggle button
OUTPUT: “compass turned on/off” prompt
DESC: compass is viewable to the user based on the toggle button status.

4.1.5.4 Navigation Arrow On/Off

ID: FNR_TST_5.4
INPUT: a click on toggle button
OUTPUT: “arrows turned on/off” prompt
DESC: direction arrows in 3d view are viewable to the user based on the toggle button status.

4.1.5.5 Hide/Show Map Icon

ID: FNR_TST_5.5
INPUT: a click on toggle button
OUTPUT: “map icon hidden/shown” prompt
DESC: control visibility of the map icon on the screen

4.1.5.6 Close Settings

ID: FNR_TST_5.6
INPUT: button click or ESC key press
OUTPUT: 3d view
DESC: Once the exit button is clicked or the ESC key is pressed by the user the system switches back to 3d view from the settings view.

4.2 User Class 2: Administrators

4.2.1 Modify Map

4.2.1.1 Add a landmark

ID: FNR_ADM_1.1
INPUT: landmark name, icon, and coordinates
OUTPUT: “landmark saved” prompt.
DESC: landmark will be added according to the details entered by the administrator.

4.2.1.2 Remove a landmark

ID: FNR_ADM_1.2
INPUT: landmark name/coordinates
OUTPUT: “landmark removed” prompt

DESC: When the administrator found that a landmark is not correct, they can remove landmark

4.2.2 Manage Content on the Software

4.2.2.1 Add Image/Video/Audio

ID: FNR_ADM_2.1
INPUT: Image/Video/Audio and description
OUTPUT: "Image/Video/Audio added" prompt
DESC: If administrators wish to add images or videos or audio, they can add them to the corresponding landmark with an auto generated ID.

4.2.2.2 Remove Images/Videos/Audio

ID: FNR_ADM_2.2
INPUT: Image/Video/Audio ID
OUTPUT: "Image/Video/Audio removed" prompt
DESC: Hypocritical images/videos/audio will be identified and removed.

4.2.3 See incoming traffic on the software

ID: FNR_ADM_3.1
INPUT: button click
OUTPUT: Incoming traffic
DESC: Incoming traffic on the software will be displayed on the screen

4.2.4 Remove Users

ID: FNR_ADM_4
INPUT: User Id
OUTPUT: Confirmation of User Deletion
DESC: Fraud or pretender users get flagged and removed for security purpose

4.2.5 Configure Project Settings

ID: FNR_ADM_5
INPUT: New project settings

OUTPUT: Confirmation of changes in settings
DESC: As per the requirement administrator can change the setting of "view type", "sound", "language" or restrict a user type from some areas

4.2.6 Approve/disapprove the modification request from Locals

ID: FNR_ADM_6
INPUT: Request ID
OUTPUT: Verdict about the request approved or disapproved
DESC: The administrator will be able to approve/disapprove requests/suggestions from the locals after verification.

4.3 User Class 3: Local Residents

4.3.1 Report or Suggest Information

4.3.1.1 Report Information

ID: FNR_LRS_1.1
INPUT: User details and complaint
OUTPUT: "Reported and sent for verification" status
DESC: Locals may report some information that they feel is not correct or adulterated, their complaint is verified and the information is modified/removed based on that.

4.3.1.2 Suggest Information

ID: FNR_LRS_1.2
INPUT: Information title, content
OUTPUT: "Request sent" confirmation
DESC: The information suggested by the local resident is verified and added by the administrator.

4.3.2 Provide multimedia content

ID: FNR_LRS_2
INPUT: Audio/Photo/Video

OUTPUT: “Sent for approval” message
DESC: Multimedia content can be provided by the local residents which after approval from the administration can be added to the project

4.4 User Class 4: Researchers

4.4.1 Download Content

4.4.1.1 Take a photo of the current view

ID: FNR_RSR_1.1
INPUT: key button or icon
OUTPUT: “photo downloaded” message
DESC: When a researcher clicks on the icon or key button, a 2D photo of the current view will start downloading.

4.4.1.2 Save the videos/360 deg images

ID: FNR_LRS_1.2
INPUT: A click on Download icon
OUTPUT: “Video/360-degree image saved” message
DESC: As per requirement, researchers can save the video or 360-degree images.

4.4.2 Save Any Spot

ID: FNR_RSR_2
INPUT: Spot coordinates
OUTPUT: “Spot saved” message
DESC: Researchers may save their spots of interest and past locations he has visited by clicking on the map and adding a spot button, the spot is saved within a system-generated spot id with the researcher’s provided coordinates.

4.4.3 Take Notes

4.4.3.1 Create a Note

ID: FNR_RSR_3.1
INPUT: a note from the user
OUTPUT: confirmation of note creation

DESC: a note will be saved with a note ID

4.4.3.2 View a Note

ID: FNR_RSR_3.2
INPUT: note ID
OUTPUT: note with node ID provided in input

4.4.3.3 Delete a Note

ID: FNR_RSR_3.3
INPUT: note ID
OUTPUT: confirmation of note deletion

5 Non-functional requirements

5.1 Performance requirements

Performance requirements refer to the specific goals or objectives that a system must meet in terms of its efficiency, accuracy and effectiveness.

Examples of performance requirements could include the following:

- **Data Collection:** The system should be able to collect accurate and relevant data about the resources and environment of Majuli Island in a timely and efficient manner.
- **Data Processing:** The system should be able to process the data collected and convert it into usable information in a timely and efficient manner.
- **Data Storage and Management:** The system should be able to store and manage the collected data effectively, ensuring that it is secure and easily accessible.
- **Data Analysis:** The system should be able to analyze the data and provide insights into the resources and environment of Majuli Island including any trends or changes.

- **Reporting:** The system should be able to generate reports on the data collected and provide stakeholders with relevant information.

The specific performance requirements and the associated metrics for measuring performance would depend on the particular requirements. It is important to define and document the performance requirements clearly to ensure that the system is designed and implemented to meet the needs of the project.

5.2 Security requirements

Security requirements refer to the measures that must be taken to protect the data, systems, and personnel involved in the project from threats and vulnerabilities.
Examples of security requirements could include the following:

- **Data Confidentiality:** The system should be designed to ensure that sensitive data is protected from unauthorized access and disclosure.
- **Data Integrity:** The system should be designed to ensure that data is not altered or corrupted during storage, processing or transmission.
- **Data Availability:** The system should be designed to ensure that data is always accessible to authorized personnel when needed.
- **Access Control:** The system should be designed to ensure that only authorized personnel can access sensitive data and systems.
- **Incident Response:** The system should include an incident response plan to address any security incidents that may occur.

The specific security requirements and the associated security controls would depend on the requirements and the types of data being collected and processed. It is important to ensure that the security requirements are documented and that appropriate security controls are implemented to protect the system and the data collected.

5.3 Compatibility requirements

Compatibility requirements refer to the standards and specifications that the components must meet to ensure that they work together effectively and efficiently.
Examples of compatibility requirements could include the following:

- **Hardware compatibility:** The hardware components used in the project, including sensors and equipment, must be compatible with the other hardware components and the overall system.
- **Software compatibility:** The software systems and applications used in the project must be compatible with each other and with the hardware components.
- **Data compatibility:** The data collected by the system must be in a format that can be easily processed and analyzed by the software systems used in the project.
- **Communication compatibility:** The communication protocols and methods used by the system must be compatible with the communication infrastructure and resources available in the project area.
- **Regulatory compatibility:** The system must comply with relevant laws, regulations and standards regarding data privacy, security and access.

The specific compatibility requirements and the associated standards and specifications would depend on the particular requirements. It is important to ensure that the compatibility requirements are clearly defined and that the components of the system are designed and implemented to meet these requirements.

5.4 Usability requirements

Usability requirements refer to the aspects that must be designed to be user-friendly and accessible to the people using the system.
Examples of usability requirements could include the following:

- **User-Centered Design:** The system should be designed with the users in mind taking into account their needs, preferences and abilities.
- **Easy to Use:** The system should be intuitive and easy to use with a simple and straightforward user interface.
- **Accessibility:** The system should be accessible to all users including those with disabilities and meet relevant accessibility standards.
- **User Assistance:** The system should include appropriate user assistance such as online help, tutorials and guides.

- **User Feedback:** The system should provide users with appropriate feedback such as confirmation messages, error messages and progress indicators.

The specific usability requirements and the associated design elements would depend on the particular requirements and the types of users who will be using the system. It is important to ensure that the usability requirements are clearly defined and that the system is designed and implemented to meet these requirements making it easy and efficient for users to interact with.

6 Contextual Inquiry

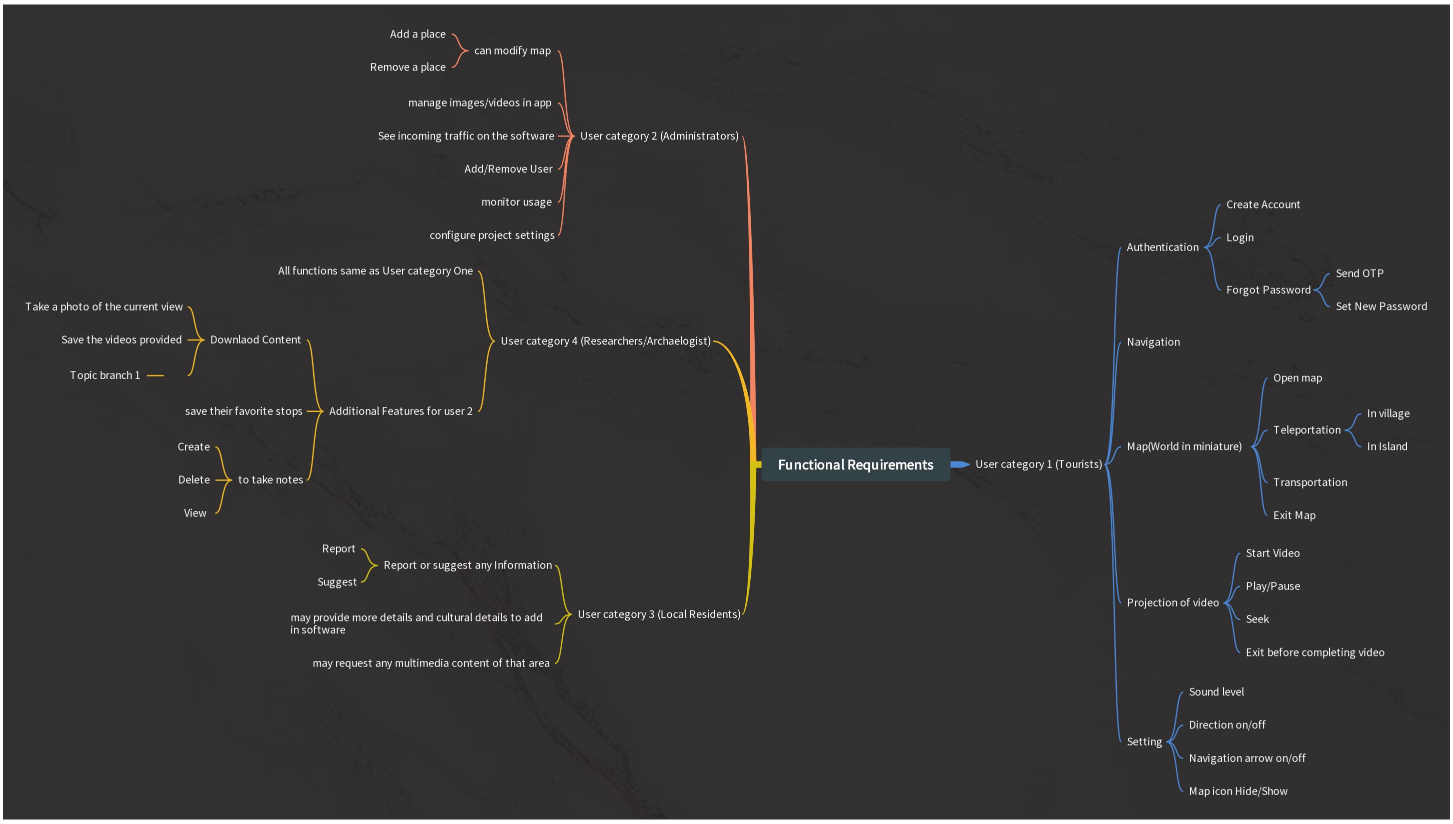
Contextual Inquiry is a qualitative research method used to understand the context in which people use a product or service. It involves observing users in their natural setting, as they perform tasks and engage with the product or service, and collecting data on their behaviors, attitudes, motivations, and challenges.

The goal of Contextual Inquiry is to gain a deep understanding of the user's experience and to uncover the underlying needs and requirements that drive their behaviors. This information can be used to inform the design of products, services, and processes that better meet the needs of users.

Since we have used a slightly different way of this inquiry. We have prepared a **mindmap**(a graphical representation of ideas and concepts, arranged around a central idea or theme) which is representing the features and requirements of the project in a structured and hierarchical manner based on the user's motivation and challenges.

Here is the link to whole mindmap SWE

Requirements look like this:



- Functional Requirements
  - User category 1 (Tourists)
    - Authentication
      - Create Account
      - Login
      - Forgot Password
        - Send OTP
        - Set New Password
    - Navigation
    - Map(World in miniature)
      - Open map
      - Teleportation
        - In village
        - In Island
      - Transportation
      - Exit Map
    - Projection of video
      - Start Video
      - Play/Pause
      - Seek
      - Exit before completing video
    - Setting
      - Sound level
      - Direction on/off
      - Navigation arrow on/off
      - Map icon Hide/Show
  - User category 2 (Administrators)
    - can modify map
      - Add a place
      - Remove a place
    - manage images/videos in app
    - See incoming traffic on the software
    - Add/Remove User
    - monitor usage
    - configure project settings
  - User category 4 (Researchers/Archaelogist)
    - All functions same as User category One
    - Additional Features for user 2
      - Downlaod Content
        - Take a photo of the current view
        - Save the videos provided
        - Topic branch 1
      - save their favorite stops
      - to take notes
        - Create
        - Delete
        - View
  - User category 3 (Local Residents)
    - Report or suggest any Information
      - Report
      - Suggest
    - may provide more details and cultural details to add in software
    - may request any multimedia content of that area