Software Requirements Specification

for

Change Detection in Temporal Satellite Imagery for Road Extraction and Mapping

Version 1.0

Prepared by

Group Name: Road Atlas

Sanjana Goskonda	SE22UCSE243	se22ucse243@mahindrauniversity.edu.in
Aditya Sahoo	SE22UCSE013	se22ucse013@mahindrauniversity.edu.in
Mahardhi Peddu	SE22UCSE203	se22ucse203@mahindrauniversity.edu.in
Samriddhi Singh	SE22UCSE240	se22ucse240@mahindrauniversity.edu.in
Bolla Shivatmika	SE22UCSE055	se22ucse055@mahindrauniversity.edu.in
Mogili Akshaya Reddy	SE22UCSE171	se22ucse171@mahindrauniversity.edu.in
G. Lishita	SE22UCSE093	se22ucse093@mahindrauniversity.edu.in

Instructor:	Prof Vijay Rao
Course:	Software Engineering
Lab Section:	CSE
Teaching Assistant:	Murali Krishna Bukkasamudram
Date:	10-03-2025

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Team Road Atlas	First draft of SRS with project purpose, scope, functional, non functional requirements	10/03/25

1 Introduction

This Software Requirements Specification (SRS) describes a project entitled Change Detection in Temporal Satellite Imagery for Road Extraction and Mapping, which leverages artificial intelligence (AI) and geographic information system (GIS) technologies to monitor and detect changes in road infrastructure over time. By processing satellite imagery, the system identifies newly constructed roads, updates geospatial databases, and provides real-time alerts to relevant stakeholders. This document outlines the project objectives, the intended features of the software, and the conventions followed to ensure consistency and clarity. It serves as a detailed guide for all stakeholders who wish to understand the underlying requirements and scope of the product.

1.1 Document Purpose

This SRS specifies the software requirements for the Change Detection in Temporal Satellite Imagery for Road Extraction and Mapping system (Release 1.0). It documents the functionalities, performance metrics, and operational constraints that guide the design, development, and deployment of the solution. The document focuses primarily on the core subsystems, namely satellite imagery processing, AI-based change detection, and data management for road updates.

This SRS also delineates the boundaries of the product. Although it provides a comprehensive view of how the system interacts with external sources of satellite data (e.g., the Bhoonidhi portal) and external services (e.g., email APIs for notifications), it does not cover any unrelated components that fall outside of the specified scope (such as in-depth geospatial data curation beyond the context of road detection).

1.2 Product Scope

The software being specified aims to perform road detection and mapping by analyzing temporal satellite imagery. It processes new satellite images from the Bhoonidhi Portal, extracts road features using a trained model, and stores them in a GIS database. When a user queries for road changes, the system fetches relevant images, compares them with existing road data, and detects modifications. If changes are found, the system displays the results to the user, allowing them to view the images and send alerts to other (subscribed) users.

By monitoring these road changes over time and presenting them in a user-friendly interface, the software saves significant time and manpower previously spent on manual image interpretation. It further supports emergency services by improving navigation to affected areas and offering a continuously updated road network. Authorities benefit from simplified data access, as well as the ability to verify and act on road changes quickly. Because the solution is designed to be intuitive and straightforward, non-experts in the field can also operate it with minimal training.

1.3 Intended Audience and Document Overview

This Software Requirements Specification (SRS) addresses the needs of several key readers:

- ***Developers*** *use it to implement the system's features in line with the stated requirements.*
- ***Testers*** *reference it to design and execute testing activities.*
- ***Project Managers and Clients*** *consult it to validate scope, manage timelines, and confirm alignment with business needs.*
- ***Professors (Academic Supervisors)*** *review it for technical feasibility, academic rigor, and completeness.*

The remainder of this SRS is structured to guide each reader through progressively more detailed topics:

- ***Sections 1–2*** *provide fundamental information about the product, including its purpose, scope, and an overall description.*
- ***Section 3*** *presents the core functional requirements—covering external interfaces, use case models, and other details essential for developers and testers.*
- ***Section 4*** *covers non-functional requirements such as performance and security, which ensure the system's robustness and scalability.*
- ***Section 5*** *captures additional requirements that may influence the product's design or deployment.*
- ***Appendices*** *offer supplementary materials, definitions, and logs for further clarity.*

Readers should begin with the introductory sections before proceeding to the more technical or role-specific sections. This order ensures that everyone gains a foundational understanding of the project's objectives and constraints before diving into specialized content.

1.4 Definitions, Acronyms and Abbreviations

Below is a list of key terms, abbreviations, and acronyms used throughout this document, sorted alphabetically:

- ***AI****: Artificial Intelligence*
- ***API****: Application Programming Interface*
- ***Bhoonidhi****: ISRO's geospatial data portal for satellite imagery*
- ***GIS****: Geographic Information Systems*
- ***ISRO****: Indian Space Research Organisation*
- ***PWA****: Progressive Web Application*
- ***SOW****: Statement of Work*
- ***SRS****: Software Requirements Specification*
- ***2FA****: Two-Factor Authentication*

1.5 Document Conventions

This document adheres to IEEE formatting standards to maintain uniformity and ease of reading. The guidelines applied in this Software Requirements Specification (SRS) are as follows:

1.5.1 Formatting Guidelines

- The document uses Arial font in size 11 or 12 for all text.
- Section and subsection headings are bolded for better visibility.
- Italicized text is reserved for comments or placeholders.
- The document is single-spaced with 1-inch margins on every side.

1.5.2 Naming Guidelines

- System components, like PreprocessingModule and QueryHandler, are presented in camel case for clarity.
- File names and database tables use snake_case (e.g., road_change_data).
- Variable and function names in the software follow consistent naming conventions based on the programming language used.

1.5.3 Terminology & Symbols

- Shall indicates a required condition.
- Should suggests a recommended but optional feature.
- May refers to an optional feature or potential future improvement.
- UML diagrams and flowcharts are utilized to visually depict workflows.

1.6 References and Acknowledgments

- ***Statement of Work**: "Change Detection in Temporal Satellite Imagery for Road Extraction and Mapping." Team RoadAtlas, 7 February 2025.*
- ***Xu, Y., Xie, Z., Feng, Y., & Chen, Z. (2018)**. Road extraction from high-resolution remote sensing imagery using deep learning. Remote Sensing, 10(9), 1461. https://www.mdpi.com/3389444*
- ***Nethravathi, B., Amitha, G., Saruka, A., Bharath, T.P., & Suyagya, S. (2020)**. Structuring natural language to query language: a review. Engineering, Technology & Applied Science Research, 10(6), 6521–6525. https://etasr.com/index.php/ETASR/article/view/3873/2407*
- ***Tan, H., Xu, H., & Dai, J. (2022)**. BSIRNet: A road extraction network with bidirectional spatial information reasoning. Journal of Sensors, 2022(1), 6391238. https://onlinelibrary.wiley.com/doi/full/10.1155/2022/6391238*
- ***Malczewski, J. (2004)**. GIS-based land-use suitability analysis: a critical overview. Progress in Planning, 62(1), 3–65. https://www.sciencedirect.com/science/article/abs/pii/S0305900603000801*
- ***Bhoonidhi Portal (ISRO)**: Geospatial data portal supplying satellite imagery for road extraction and mapping.*

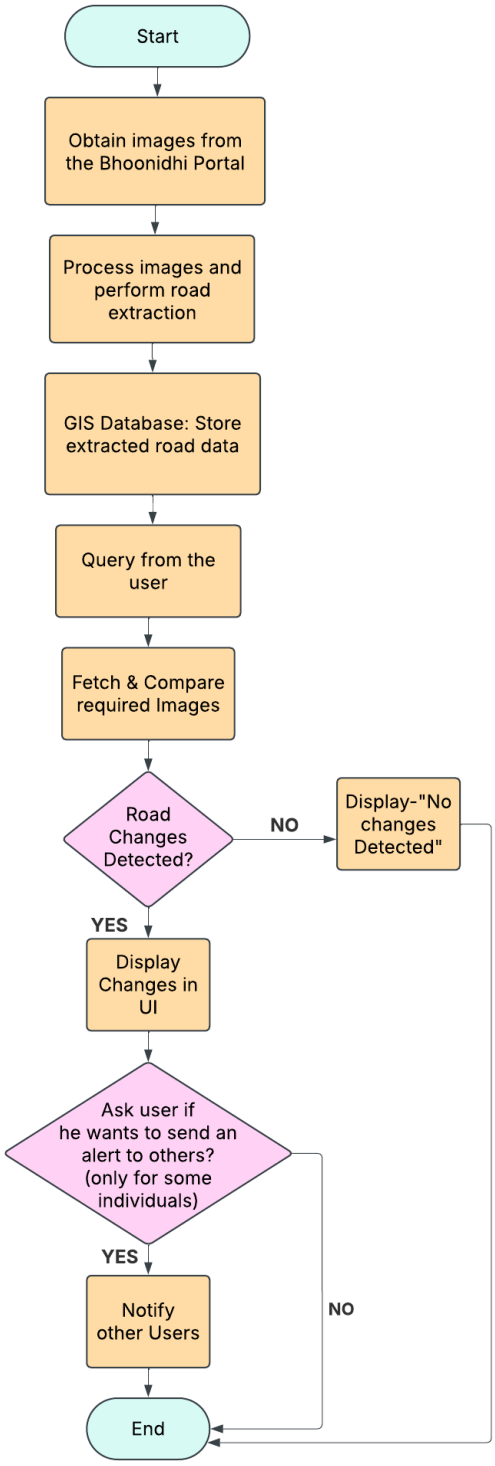
The authors wish to acknowledge the valuable input from all course instructors and external advisors, who contributed ideas and guidance to shape the scope and direction of this software solution.

2 Overall Description

2.1 Product Overview

The Road Change Detection System is a new, standalone product that identifies changes in roads using satellite images and GIS data. It combines image processing methods, trained models, and a GIS database to automatically find newly built, altered, or removed roads.
This system is not an extension of any existing product line, as it does not enhance or build on a previous commercial road change detection system. Although research has been done in this area, there is no leading system that this project directly relies on.
Furthermore, it does not serve as a substitute for any specific current system. Existing road mapping and change detection techniques often depend on manual GIS updates or semi-automated methods. In contrast, this system offers an automated process that reduces the need for human input and improves accuracy through machine learning and image processing.

You can find the diagram on the next page.


Start
Obtain images from the Bhoonidhi Portal
Process images and perform road extraction
GIS Database: Store extracted road data
Query from the user
Fetch & Compare required Images
Road Changes Detected?
NO
Display-"No changes Detected"
YES
Display Changes in UI
Ask user if he wants to send an alert to others? (only for some individuals)
YES
Notify other Users
NO
End

2.2 Product Functionality

User Functionalities

1. **Create Account**
    - Users can sign up using an email, username, and password.
    - Multi-factor authentication (MFA) can be enabled for security.
    - Upon account creation, users receive an email verification link.
    - Profile settings allow updating details like name, password, and contact info.
2. **Get Allocated Locations**
    - Users are assigned specific locations upon account creation or by an admin.
    - The system restricts user access to only their allocated locations.
    - Users can view a list of their assigned locations from their dashboard.
    - If they need additional locations, they must request access.
3. **Search Function with Read-Only Access to Allocated Locations**
    - Users can search for records within the database but only for their assigned locations.
    - They can filter search results based on criteria like date, name, or category.
    - The system ensures read-only access, preventing modifications or deletions.
    - Results can be exported (if permitted) in formats like CSV or PDF.
4. **Request Access**
    - If users need access to additional locations or permissions, they can submit a request.
    - Requests go to the admin responsible for review.
    - Users may be required to provide justification for the access request.
    - Admins approve or deny requests, with notifications sent to users.

Admin Functionalities

1. **View Information on Users Allocated to Them**
    - Admins can see a list of users assigned to them.
    - User profiles display details such as name, email, assigned locations, and roles.
    - Admins can filter users based on location, access level, or activity status.
2. **Change Permissions of Users on the Database**
    - Admins can grant or revoke read and write access for specific locations.
    - A role-based access control (RBAC) system ensures permissions are managed securely.
    - Logs track all permission changes for audit purposes.
    - Temporary access can be granted with expiration dates.
3. **View the Database**
    - Admins have full visibility into the database, including all locations.
    - They can perform queries, generate reports, and audit changes.
    - Role-based permissions ensure only authorized actions are allowed.
    - Logs track all admin actions for security and compliance.

2.3 Design and Implementation Constraints

Hardware Limitations:

1. Significant computational power is necessary for processing images and executing deep learning models.
2. Substantial storage capacity is needed for satellite imagery and Geographic Information System (GIS) data.
3. A high-speed internet connection is essential for downloading Bhoonidhi satellite images.

COMET Methodology:
The software should be developed in accordance with the COMET (Collaborative Object Modeling and Architectural Design Method) methodology, which advocates for an object-oriented approach to software engineering. This methodology prioritizes use case-driven development, iterative design processes, and architectural modeling, rendering it particularly effective for intricate systems such as Geographic Information Systems (GIS) and image processing applications.

Reference:
Gomaa, H. (2011). Designing Software Product Lines with UML: Applications to Embedded Systems. Addison-Wesley.

UML Modeling:
The design of the system must comply with the standards set forth by the Unified Modeling Language (UML) for depicting system architecture, workflows, and interactions. Various UML diagrams, including class diagrams, sequence diagrams, activity diagrams, and deployment diagrams, will be employed to effectively model the system.

Reference:
Object Management Group (OMG). (2017). Unified Modeling Language (UML) Version 2.5.1.

Programming Languages: The development of the system will utilize Python for image processing and React for the user interface visualization.

Libraries: The implementation will incorporate external libraries such as OpenCV for image processing, TensorFlow for deep learning applications, and PostgreSQL for the storage of user data, and PostGIS for the storage of GIS data.

Security: It is essential to establish authentication and access control measures to safeguard against unauthorized access to road data and satellite imagery.

2.4 Assumptions and Dependencies

Assumptions:

1. Access to Bhoonidhi Satellite Images: The system expects that necessary satellite images will be accessible via the Bhoonidhi Portal in a timely fashion.
2. It is presumed that roads exceeding 20 feet in width will be distinctly visible in satellite images, facilitating precise detection.
3. Furthermore, Bhoonidhi images are anticipated to be available in standard geospatial formats, such as GeoTIFF, to ensure seamless processing and integration.
4. Reliable Internet Connection: A reliable internet connection is needed for the software to retrieve and process satellite images.
5. High-Performance Computer: It is assumed that the system will operate on a powerful workstation with adequate GPU resources for deep learning tasks.

Dependencies:

1. Third-Party GIS Software: The system might work with GIS applications such as QGIS or ArcGIS for better visualization and analysis.
2. External Libraries: The software depends on OpenCV, TensorFlow, and PostgreSQL/PostGIS for handling and storing road data.
3. Cloud Storage: If satellite images are too large for local storage, cloud storage options like AWS S3 may be utilized.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

1. Progressive Web App (PWA) for Easy Access on Mobile and Laptops.
2. Users will have a personalized dashboard displaying allocated locations and recent activities. Requests for additional access can be made through a simple interface. Notifications will inform users about account updates, approvals, or important changes.
3. A search bar will allow them to filter and find relevant data quickly.
4. Admins will have a dedicated interface to manage users, permissions, and locations. A user management section will allow them to modify access and permissions.
5. The login page will support two-factor authentication (2FA) for enhanced security.

3.1.2 Hardware Interfaces

Hardware Interfaces

The software product will interact with the following hardware components:

1. **User Devices (Laptops, Desktops, Mobile Phones, Tablets)**
    - The software will be accessible via a **Progressive Web App (PWA)**, ensuring compatibility with various screen sizes and operating systems (Windows, macOS, Linux, Android, iOS).
    - Devices should have an internet connection for full functionality, but offline support will be available for certain features.
    - Minimum hardware requirements:
        - Mobile: 2GB RAM, Android 8.0+ / iOS 13+
        - Laptop/Desktop: 4GB RAM, modern web browser (Chrome, Firefox, Edge, Safari)
2. **Servers (Cloud or On-Premise Database and Application Servers)**
    - The backend system will be hosted on cloud-based or on-premise servers.
    - The system will interact with a **database server** for storing user data, locations, and permissions.
    - Secure API endpoints will handle communication between the frontend (PWA) and backend.
3. **External Storage Devices (Optional Data Export & Backup)**
    - Users may export reports and data to USB drives, external SSDs, or cloud storage services.
    - The system will support standard file formats (CSV, PDF) for data export.

3.1.3 Software Interfaces

1. **Database Management System (DBMS)**
    - The software will connect to a **PostgreSQL database** for storing user accounts, allocated locations, access permissions, and logs.
    - Read and write operations will be handled using secure API calls.
    - The system will enforce **role-based access control (RBAC)** to manage user permissions.
2. **Web Browser (Client Interface for PWA)**
    - The frontend will be a **Progressive Web App (PWA)** accessible through web browsers like Chrome, Firefox, Edge, and Safari.
    - The application will use **HTML, CSS, TypeScript** to provide an interactive UI.
3. **Authentication and Security Services**
    - Two-Factor authentication will be implemented .
    - The system will follow ISO 27001 guidelines for secure data handling and risk management. Multi-level user access control will restrict sensitive functions to authorized users only.
4. **Cloud Storage and Backup**
    - If the system supports exporting reports or backups, it will integrate with cloud storage providers like **AWS S3, Google Drive, or Dropbox**.
    - Users may download data in **CSV or PDF formats**.

3.2 Functional Requirements

3.2.1 F1: Image Processing

- The system shall automatically retrieve satellite images from the Bhoonidhi Portal.
- The system shall preprocess the retrieved images to enhance quality by applying noise reduction, contrast adjustment, and geometric corrections.
- The system shall extract roads from the preprocessed images using a trained deep learning model designed for road detection.
- The system shall store the extracted road data in the GIS Database for further processing and analysis.
- The system should provide an interface for administrators to manually review and validate the extracted road data before it is stored in the GIS Database.
- The system may integrate with other satellite image sources in addition to Bhoonidhi Portal to improve road extraction accuracy

3.2.2 F2: User Query for Road Changes

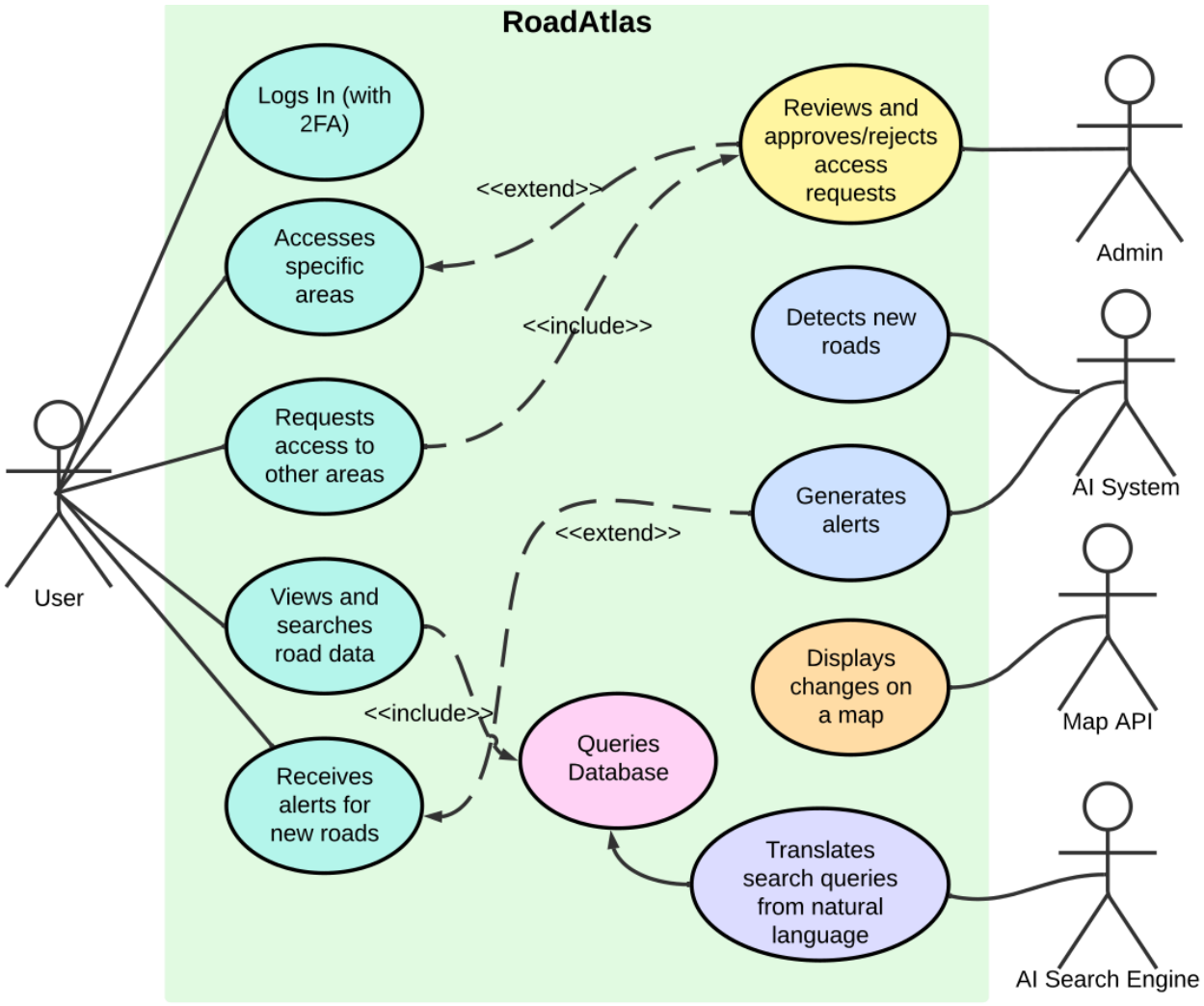
- The system shall allow users to search for road changes through a graphical User Interface (UI).
- The system shall process user queries and retrieve relevant images from the GIS Database based on the specified search criteria.
- The system shall compare the old satellite images with the new satellite images to detect changes in roads such as newly constructed roads, widened roads, or removed roads.
- The system shall visually display the detected road changes on an interactive map interface.

- The system should provide filtering options to allow users to refine results based on date, location, or road type (e.g., highways, streets, rural roads).
- The system may offer multiple map visualization modes, such as satellite view, street view, or hybrid view, to improve the clarity of detected road changes.

3.2.3 F3: Change Detection and Alerts

- The system shall identify and highlight specific areas where road changes have been detected based on the comparison of old and new satellite images.
- The system shall allow users to view and interact with the detected changes through a graphical User Interface (UI), which includes zooming, panning, and overlaying images for better analysis.
- The system shall provide users with an option to manually trigger an alert when they confirm a detected change is significant.
- The system shall automatically send a notification to all subscribed users when an alert is triggered, containing details such as location, type of change, and timestamp.
- The system should allow users to configure alert preferences, such as receiving notifications only for major road changes, or restricting alerts to specific geographic regions.
- The system may integrate with SMS services to send text message notifications to subscribed users when an alert is triggered.

3.3 Use Case Model



3.3.1 Use Case #1 Login with 2FA (U1)

- **Author**
  Team Member: Sanjana G
- **Purpose**
  Enable a registered user to authenticate securely into the system using a two-factor authentication (2FA) mechanism.
- **Requirements Traceability**
  - R1: The system shall support username/password authentication.
  - R2: The system shall require a secondary verification step (2FA code) upon login.
  - R3: The system shall grant users access to the appropriate system areas following successful authentication.

- **Priority**
  High. Reliable user authentication is critical for secure access to data.
- **Preconditions**
  - The user already has valid login credentials (username, password).
  - The system's 2FA service is active and able to generate and deliver verification codes.
  - The user's account is not locked or otherwise disabled.
- **Postconditions**
  - The user is granted access based on their role and permissions.
  - The system logs the successful login event for auditing.
  - The user can now proceed to access specific areas or request additional data privileges.
- **Actors**
  - **User**: Initiates the login process and enters credentials.
  - **2FA Service**: Generates and validates the secondary authentication code.
- **Extends**
  Not applicable; this is a primary use case.
- **Flow of Events**
  Basic Flow
  - The user navigates to the login page.
  - The user enters their username and password.
  - The system verifies the credentials. If valid, the system prompts for the 2FA code.
  - The user receives the 2FA code (e.g., via email or SMS).
  - The user inputs the 2FA code into the system.
  - The system verifies the 2FA code and grants access if it is correct.
- **Alternative Flow**
  - A1: *Incorrect 2FA Code*
    - If the user enters an invalid 2FA code, the system displays an error.
    - The system prompts the user to re-enter or request a new code.
- **Exceptions**
  - E1: *User Enters Incorrect Password*
    - The system denies access and displays an error message.
  - E2: *2FA Service Unavailable*
    - The system notifies the user that 2FA codes cannot be generated at this time.
    - The user cannot proceed until the service is restored.
- **Includes**
  Not applicable; the use case does not incorporate any other use cases directly.
- **Notes/Issues**
  - If the system repeatedly detects incorrect 2FA attempts, it may temporarily lock the user's account.
  - System logs should capture both successful and failed login attempts for auditing purposes.

3.3.2 Use Case #2 Generate and Send Alert for Road Changes (U2)

- **Author**
  Team Member: *Sanjana G*

- **Purpose**
  Automatically notify designated users when the system detects changes to existing roads or identifies newly constructed roads.
- **Requirements Traceability**
  - R10: The system shall monitor satellite imagery for detected road changes.
  - R11: The system shall generate an alert when changes exceed a predefined threshold.
  - R12: The system shall distribute alerts to users through email and/or in-app notifications.
- **Priority**
  High. Real-time notification of road changes is a core functionality supporting timely response and data accuracy.
- **Preconditions**
  - The AI-based detection component is operational and actively scanning new satellite imagery.
  - There are registered users or user groups designated to receive change alerts.
  - The system has valid email configuration or a functioning in-app notification mechanism.
- **Postconditions**
  - Relevant users are informed of road changes via the chosen alert medium(s).
  - The system records the alert event for audit and reference.
- **Actors**
  - **Detection Module**: Identifies new or changed roads.
  - **System** (Notification Service): Generates and dispatches the alerts.
  - **User** (Recipient): Receives the road change alerts.
- **Extends**
  Not applicable; this is a primary use case.
- **Flow of Events**
  Basic Flow
  - The AI detection module flags a road change in the satellite imagery.
  - The system verifies the severity or significance of the change against a predefined threshold.
  - If the threshold is met or exceeded, the system compiles an alert message (e.g., location, nature of change).
  - The system sends the alert to all designated recipients (e.g., via email or an in-app notification).
  - The system logs the alert for tracking and auditing purposes.
- **Alternative Flow**
  - A1: *Insufficient Road Change*
    - If the detected change does not meet or exceed the threshold, no alert is sent.
    - The system still records the detection event but categorizes it as minor or unworthy of alert.
- **Exceptions**
  - E1: *Notification Service Failure*
    - The system attempts to resend or switch to an alternative notification method.
    - If all methods fail, the system logs the error and alerts an administrator at the earliest opportunity.
  - E2: *Invalid User Email/Contact*

        - The system cannot deliver the alert to one or more recipients.
        - The system logs the failed attempt and sends a warning message to an admin.
- **Includes**
  Not applicable; no other use cases are incorporated.
- **Notes/Issues**
    - Administrators should be able to define or adjust thresholds for alerts.
    - Logs should include timestamps, relevant user IDs, and any data describing the specific road changes.

3.3.3 Use Case #3: Request Access to Other Areas (U3)

- **Author**
  Team Member:Aditya Sahoo
- **Purpose**
  Enable a user to submit a request for additional access privileges to view or modify road data in areas beyond their current permissions.
- **Requirements Traceability**
    - R7: The system shall allow users to request expanded access.
    - R8: The system shall notify an administrator about incoming access requests.
    - R9: The system shall grant or deny requests based on administrative review.
- **Priority**
  Medium. While not essential to basic usage, managing permissions ensures secure, role-based data access.
- **Preconditions**
    - The user is already logged into the system with valid credentials.
    - The user's current permissions do not cover the area or dataset they need to access.
- **Postconditions**
    - The user's request is recorded and awaits admin approval.
    - The user receives confirmation that the request has been submitted.
- **Actors**
    - **User** (Authenticated): Initiates the permission request.
    - **Admin**: Receives and reviews the request.
    - **System**: Routes the request to the admin and tracks its status.
- **Extends**
  Not applicable; this is a standalone use case.
- **Flow of Events**
  Basic Flow
    - The user navigates to the "Request Additional Access" feature in the dashboard.
    - The user selects the region or dataset for which they need elevated permissions.
    - The user provides an optional justification or notes for the request.
    - The system registers the request and forwards it to the admin.
    - The user receives a confirmation message that the request has been submitted.
- **Alternative Flow**
    - A1: *Request Form Incomplete*
        - The system detects missing or invalid information.
        - The user is prompted to correct the form and resubmit.
- **Exceptions**
    - E1: *Admin Unavailable*

        - The system queues the request until an admin is available to review it.
    - E2: *System Error During Submission*
        - The system informs the user that an error has occurred, and they should retry or contact support.
- **Includes**
  Not applicable; no sub-use cases are directly included.
- **Notes/Issues**
    - Administrators can approve, reject, or request additional details before granting access.
    - An audit log should track the history of all access requests for security compliance

3.3.4 Use Case #4: Specific Area Access (U4)

- **Author**
  Team Member: G.Lishita
- **Purpose**
  Enable users to access road data for a specific area within their authorized permissions.

- **Requirements Traceability**
  - R10: The system shall allow users to view road data within their assigned regions.
  - R11: The system shall restrict access to unauthorized regions.
  - R12:The system shall log all access attempts for auditing.

- **Priority**
  High. Ensuring users can efficiently access road data within their permissions is critical for the system functionality.

- **Preconditions**
  - The user is logged into the system with valid credentials.
  - The user has existing permissions to access certain areas.

- **Postconditions**
  - The user successfully accesses road data for an authorized area.
  - Unauthorized access attempts are blocked and logged.

- **Actors**
  - User (Authenticated): Requests access to road data for a specific area.
  - System: Verifies permissions and grants or denies access accordingly.

- **Extends**
  Not applicable; this is a standalone use case.

- **Flow of Events**
  Basic Flow
    1. The user navigates to the "Search" section in the dashboard.
    2. The user selects a specific area they want to access.
    3. The system verifies the user’s permissions for the requested area.
    4. If the user is authorized, the system grants access and displays the data.
    5. The system logs the access attempt for security and auditing purposes.

Alternative Flow

A1: User Requests Unauthorized Area
- The system detects that the user lacks permission for the requested area.
- The system displays an error message and suggests submitting an Access Request

- **Exceptions**
  E1: System Error While Fetching Data
  - The system informs the user that an error has occurred and suggests retrying later.

- **Includes**
  Not applicable; no sub-use cases are directly included.

- **Notes/Issues**
  - An audit log should maintain a record of all access attempts.

3.3.4 Use Case #5: Enabling Government Agencies to Manage Road Project (U5)

- **Author**
  Team Member: Mogili Akshaya

- **Purpose**
  Enable government agencies to monitor and manage road construction projects, track progress, and oversee upcoming developments.

- **Requirements Traceability**
  - R19: The system shall allow government agencies to monitor road construction projects.
  - R20: The system shall enable agencies to update progress reports.
  - R21: The system shall provide notifications for upcoming developments and milestones.

- **Priority :**
  High. Effective project management ensures timely completion and transparency in infrastructure development.

- **Preconditions**
  - The government agency is logged into the system with valid credentials.
  - The agency has the necessary access permissions to manage road projects.

- **Postconditions**
  - Project statuses and progress updates are recorded in the system.
  - Stakeholders receive notifications regarding project developments.
  - The system maintains a history of updates for auditing and compliance.

- **Actors**
  - Government Agency: Manages and updates road project details.
  - Admin: Grants or revokes access permissions.
  - System: Stores and tracks project statuses and updates.

- **Extends**
  Not applicable; this is a standalone use case.

- Flow of Events
  Basic Flow
  1. The government agency logs into the Road Monitoring System.
  2. The agency navigates to the "View Road Projects" section.
  3 . The agency selects an existing project from the available list.
  4 . The agency logs out or continues viewing other projects.

  Alternative Flow
  A1: Missing Information in Project Update
  - The system detects missing required fields.
  - The system informs the agency that there are currently no projects assigned for viewing.

- **Exceptions**
  E1: Unauthorized Access Attempt
  - The system denies access and notifies the admin of unauthorized access attempts.

  E2: System Downtime During Submission
  - The system informs the agency of temporary unavailability and suggests retrying later.

- **Includes**
  Not applicable.

- **Notes/Issues**
  - Audit logs should track all changes made by agencies for compliance purposes.
  - The system should allow collaboration between agencies on multi-region projects.

4 Other Non-functional Requirements

4.1 Performance Requirements

P1 - Concurrent Users – The system must support at least 100 users simultaneously without performance drops. This ensures that city planners, researchers, and engineers can access and analyze data smoothly, even during peak usage.

P2 - Image Pre-Processing Speed – Satellite images must be processed within 15 seconds, including cropping, resizing, and normalization. Faster pre-processing prevents delays in the pipeline and ensures the machine learning model gets optimized input for accurate road detection.

P3 - Road Detection Speed – The system must analyze each satellite image within 60 seconds to quickly identify new roads and update the database. This ensures timely infrastructure monitoring and decision-making.

P4: Data Storage & Retrieval – The database must efficiently store 1 year of road data while keeping query times under 5 seconds. Optimized indexing and compression will allow users to retrieve historical data quickly for comparisons and analysis.

P5: Website Load Time – The website must load within 60 seconds on a 10 Mbps connection. A fast-loading interface enhances usability, allowing smooth navigation across all devices.

P6: Cross-Device Support – The website must be fully responsive, adapting to mobiles, tablets, and desktops. The UI should dynamically adjust for different screen sizes, ensuring smooth interactions across all platforms.

4.2 Safety and Security Requirements

General Security Measures

The system will follow **ISO 27001** guidelines for secure data handling and risk management. Multi-level user access control will restrict sensitive functions to authorized users only.

Data Protection and Privacy

AES-256 encryption will secure stored data, ensuring protection against breaches. Users will have full control over their data with options to view, update, and delete it.

Database and API Security

The database will have restricted access with daily automated backups to prevent data loss. All API requests will require authentication keys to verify legitimate access.

Mobile and Network Security

End-to-end encryption will protect mobile communications, ensuring data security. Secure login mechanisms, such as two-factor authentication (2FA), will be enforced for enhanced security.

Incident Response and Compliance

A Security Incident Response Plan will be in place to detect and resolve breaches effectively. The system will comply with ISO 27001 to maintain security and privacy standards.

Cost-Effective Security Implementation

Open-source security tools will be used where possible to maintain security while reducing costs. Cloud-based security solutions will minimize infrastructure expenses and ensure scalability.

4.3 Software Quality Attributes

1. Reliability

The system must provide 99.9% uptime, ensuring uninterrupted access to road detection and data retrieval. Automated error handling, failover mechanisms, and periodic backups will prevent data loss and minimize downtime.

2. Security

End-to-end encryption, Role-Based Access Control (RBAC), and Two-Factor Authentication (2FA) will protect sensitive data. The database will be secured using ISO 27001 guidelines, and regular security audits will identify and mitigate vulnerabilities.

3. Scalability

The system must support at least 100 concurrent users while maintaining smooth performance. This will be achieved using cloud hosting, load balancing, and database optimization to handle increasing user traffic and data processing demands efficiently.

4. Usability

The interface must be intuitive, responsive, and accessible across desktops, tablets, and mobile devices. Features such as easy map navigation, clear data visualization, and customizable overlays will enhance user experience.

5. Maintainability

The system will follow modular development, clean coding standards, and proper documentation to ensure easy updates and bug fixes. Version control (Git) and CI/CD pipelines will streamline deployments and ensure long-term stability.

6. Efficiency

The system must process satellite images within 60 seconds and retrieve road data in under 2 seconds for a smooth experience. Optimized query execution, caching, and database indexing will ensure fast data access and system responsiveness.

5 Other Requirements

5.1 Database and GIS Requirements

- The system will integrate PostgreSQL as the primary database.
- PostGIS must be enabled to support geospatial queries and data storage.
- The database should be scalable and designed with optimized table structures.
- The system should be able to handle large geospatial data efficiently.
- Caching mechanisms should be considered to enhance performance.
- The system should support multiple spatial operations such as k-nearest neighbor (KNN) search, intersections, and other geospatial computations.

5.2 Legal and Compliance Requirements

- The system must adhere to data policies set by ISRO.
- Usage of satellite images must comply with ISRO licensing agreements and applicable regulations.

Appendix A – Data Dictionary

1. Constants

Name	Description	Value/Range	Related Operations	Requirement ID
IMAGE_RESOLUTION	Default resolution for satellite images	1024x1024 px	Image processing	F1
CHANGE_THRESHOLD	Minimum percentage change to trigger an alert	10%	Change detection	F3
ALERT_FREQUENCY	Time interval for checking road changes	Every 24 hours	Alert generation	F3
MAX_QUERY_RESULTS	Maximum results returned in a user query	100	User query processing	F2

2. State Variables

Variable	Description	Possible States	Related Operations	Requirement ID
USER_ROLE	Defines user access levels	Admin, User	User authentication, permissions	F1, F2
IMAGE_STATUS	State of the image in processing pipeline	Raw, Preprocessed, Analyzed	Image processing	F1
ROAD_CHANGE_STATUS	Status of detected road changes	Pending Review, Approved, Rejected	Change detection	F3

3. Inputs

Input Name	Description	Data Type	Source	Related Operations	Requirement ID
Satellite Image	Raw image from Bhoonidhi Portal	TIFF/JPEG/PNG	Bhoonidhi Portal	Image processing	F1
User Query	Search request for road changes	String (Location, Date)	User input	Query processing	F2
Manual Review	Admin verification of detected road changes	Boolean (Approve/Reject)	Admin	Change validation	F3

4. Outputs

Output Name	Description	Data Type	Destination	Related Operations	Requirement ID
Road Change Report	Summary of detected road changes	JSON/XML	GIS Database	Change detection	F3
User Alert	Notification sent to users on detected road changes	Email/Push Notification	User device	Alert generation	F3
Search Results	List of road changes based on user query	JSON/List	User Interface	Query processing	F2

Appendix B - Group Log

Meeting 1

1. Meeting Details

- **Date:** 06/02/2025
- **Location/Platform:** Main Campus
- **Attendees:**
    - Samriddhi
    - Lishita
    - Mahardhi
    - Sanjana
    - Shivatmika
    - Akshaya

2. Agenda

- Finalizing the problem statement
- Viability of the project
- Dividing roles

Meeting 2

1. Meeting Details

- **Date:** 17/02/2025
- **Location/Platform:** IT Lab
- **Attendees:**
    - Samriddhi
    - Lishita
    - Mahardhi
    - Sanjana
    - Shivatmika
    - Akshaya

2. Agenda

- Mapping inputs and outputs of the entire software
- Discussions of use cases
- Mapping the functionalities we plan to implement
- High level timeline

Meeting 3

1. Meeting Details

- **Date:** 04/03/2025
- **Location/Platform:** Hostel
- **Attendees:**
    - Sanjana
    - Lishita
    - Akshaya

2. Agenda

- Signup/ Login page
- Landing page of the site

Meeting 4

1. Meeting Details

- **Date:** 06/03/2025
- **Location/Platform:** Main Campus
- **Attendees:**
    - Samriddhi
    - Mahardhi
    - Shivatmika

2. Agenda

- Backend workflow
- Database integration

Meeting 5

1. Meeting Details

- **Date:** 09/03/2025
- **Location/Platform:** Hostel
- **Attendees:**
    - Akshaya
    - Lishita
    - Samriddhi

2. Agenda

- Front-end - Back-end connection

Meeting 6

1. Meeting Details

- **Date:** 10/03/2025
- **Location/Platform:** IT Lab
- **Attendees:**
  - Samriddhi
  - Sanjana
  - Aditya
  - Shivatmika
  - Akshaya
  - Lishita

2. Agenda

- Finalizing SRS
- Database services