Requirements Specification

for

FLOODLIGHT

Version 0.1

Prepared by

CODE FOR PAKISTAN

Jul 18, 2023

Revision History

Name	Date	Reason For Changes	Version
Mubassir Hayat	Jul 24, 2023	Initial Draft	0.1
Mubassir Hayat	Mar 14, 2024	Reviewing for any new additions, typos, etc. Simplified the TOC.	0.2

Table of Contents

1. Introduction

1.1 Purpose

The purpose of the Requirements Specification document for Floodlight is to provide a comprehensive and detailed description of the functional and non-functional requirements for the development of the Floodlight Disaster Management Toolkit. This document serves as a crucial communication tool between stakeholders, including project managers, developers, designers, and testers, ensuring a common understanding of the project's objectives and scope.

The requirements document outlines the specific features, functionalities, and capabilities that the Floodlight platform must possess to effectively collect, analyze, and visualize data related to disaster management. It defines the intended use cases, user interactions, and system behavior, ensuring that all project stakeholders are aligned on the expected outcomes.

Additionally, the requirements document serves as a basis for project planning, development, and testing. It aids in estimating project timelines, resource allocation, and budgeting. Moreover, it acts as a reference for quality assurance and testing teams, enabling them to validate the system against the stated requirements.

Ultimately, the requirements document aims to provide a comprehensive blueprint for the development of Floodlight, guiding the entire software development life cycle to ensure the successful creation of a robust, user-friendly, and impactful Disaster Management Toolkit.

1.2 Intended Audience

The Requirements Specification document for the Floodlight Disaster Management Toolkit is intended for a diverse audience of stakeholders involved in the development, implementation, and evaluation of the Floodlight project. The primary audience includes

1.2.1 **Development Team:** Software developers, engineers, and programmers who will be responsible for building the Floodlight platform based on the specified requirements.
1.2.2 **Project Managers:** Those overseeing the Floodlight project, responsible for planning, coordinating, and monitoring progress.
1.2.3 **Quality Assurance and Testing Team:** Individuals responsible for verifying that the developed Floodlight system meets the specified requirements and performs as expected.
1.2.4 **Designers:** User interface and user experience (UI/UX) designers who will create the visual elements and user interactions for the Floodlight platform.
1.2.5 **Domain Experts:** Experts in disaster management, emergency response, and related fields who can provide valuable insights into the functionality and relevance of the Floodlight system.

1.2.6 **Stakeholders:** All parties with a vested interest in the Floodlight project's success, including investors, sponsors, government organizations, NGOs, and disaster management authorities.

1.3 Project Scope

The Project Scope for the Floodlight Disaster Management Toolkit defines the boundaries, objectives, deliverables, and constraints of the project. It outlines what the project will accomplish, the features and functionalities that will be included, and the specific outcomes that are expected to be achieved. The scope helps in defining the extent of the project and provides clarity on what will be included and excluded from the final product.

By clearly defining the project scope, all stakeholders can have a common understanding of what the Floodlight Disaster Management Toolkit will entail, and the project can proceed with a well-defined direction and focus. Any changes to the scope should be managed through a formal change control process to ensure that they are properly evaluated, approved, and implemented.

1.3.1 Objectives:

The objective of the Floodlight Disaster Management Toolkit is to develop and implement a comprehensive and effective software solution that enhances disaster management efforts and preparedness response and recovery capabilities. The main goal is to empower stakeholders, including government organizations, NGOs, and community members, with real-time data collection, analysis, and communication tools to better understand, prepare for, and respond to disasters.

The specific objectives of the Floodlight project include:

1.3.1.1 Real-time Data Collection:

Enable the toolkit to collect and aggregate data from various sources, such as crowdsourcing, survey data, social media, satellite imagery, and sensor networks, to provide up-to-date and accurate information about disaster events.

1.3.1.2 Data Analysis and Visualization:

Implement data analysis algorithms and visualization tools to process the collected data and present it in a clear and informative manner through charts, maps, graphs, and other visualizations.

1.3.1.3 Early Warning and Alerting:

Develop a robust early warning system that can quickly detect and assess potential disaster threats, issuing timely alerts to relevant authorities and affected communities.

1.3.1.4 Crowdsourcing and Community Engagement:

Facilitate community engagement and participation in disaster management by allowing users to contribute data and feedback through crowdsourcing mechanisms.

1.3.1.5 Integration and Interoperability:

Ensure seamless integration with existing disaster management systems, government databases, and communication platforms to foster collaboration and information sharing.

1.3.1.6 Resilient Recovery and Reconstruction:

Support post-disaster recovery and reconstruction efforts by providing data-driven insights that aid in decision-making and resource allocation.

1.3.1.7 Data Privacy and Security:

Implement stringent data privacy and security measures to protect user information and maintain the confidentiality and integrity of the data collected.

1.3.1.8 User-Friendly Interface:

Create an intuitive and user-friendly interface that caters to the diverse needs of stakeholders, making the toolkit accessible and easy to use for individuals with varying technical expertise.

1.3.1.9 Scalability and Sustainability:

Design the Floodlight system to be scalable and sustainable, accommodating future growth and adapting to evolving disaster management needs.

Ultimately, the objective of the Floodlight Disaster Management Toolkit is to enhance disaster preparedness, response, and recovery efforts, ultimately reducing the impact of disasters and saving lives. By providing a centralized platform for data-driven decision-making and communication, the toolkit aims to promote resilience and support the collaborative efforts of organizations and communities working together to manage and mitigate the effects of disasters in Pakistan.

1.3.2 Functional Features:

The Functional Features of the Floodlight Disaster Management Toolkit are the specific capabilities and functionalities that the system will provide to fulfill its objectives. These features enable the toolkit to effectively collect, analyze, and disseminate data related to disaster management.

1.3.2.1 Data Collection and Aggregation:

- **Crowdsourcing:** Enable users to contribute disaster-related information, photos, and videos through mobile apps or web interfaces.
- **Social Media Integration:** Gather data from social media platforms to capture real-time updates and user-generated content.
- **Sensor Data Integration:** Collect data from various sensors, such as weather stations and environmental sensors, to monitor relevant parameters.
- **Satellite Imagery:** Utilize satellite imagery to assess the extent of disaster impacts and changes over time.

1.3.2.2 Data Analysis and Visualization:

- **Real-Time Analysis:** Process incoming data in real-time to identify patterns, trends, and anomalies related to disaster events.
- **Geographic Information System (GIS):** Visualize data on maps, allowing users to geospatially analyze disaster information and identify hotspots.
- **Graphs and Charts:** Present statistical data and trends through graphs, charts, and infographics for easy comprehension.

1.3.2.3 Early Warning and Alerting:

- **Disaster Event Detection:** Employ machine learning algorithms to detect and classify potential disaster events from data streams.
- **Alert Generation:** Automatically generate and disseminate alerts through SMS, email, or push notifications to relevant stakeholders.

1.3.2.4 Communication and Collaboration:

- **Two-Way Communication:** Facilitate real-time communication between disaster management authorities, response teams, and affected communities.
- **Collaboration Spaces:** Provide online platforms for different organizations to share information, coordinate efforts, and plan response strategies.

1.3.2.5 Resilient Recovery and Reconstruction:

- **Post-Disaster Assessment:** Conduct damage assessments through data analysis to inform recovery and reconstruction efforts.
- **Resource Allocation:** Assist decision-makers in allocating resources based on data insights and needs assessment.

1.3.2.6 Security and Access Control:

- **User Authentication:** Implement secure login mechanisms and user authentication to control access to the toolkit.
- **Data Privacy:** Ensure data privacy compliance and protect sensitive information through encryption and access controls.

1.3.2.7 Cross-Platform Compatibility:

- **Web and Mobile Applications:** Provide web and mobile interfaces to enable users to access the toolkit on various devices.

1.3.2.8 Reporting and Data Export:

- **Custom Reports:** Allow users to generate customized reports based on selected criteria and data filters.
- **Data Export:** Enable data export in standard formats (e.g., CSV, JSON) for further analysis and reporting.

1.3.2.9 System Administration and Management:

- **User Management:** Facilitate user roles and permissions management for different stakeholders.
- **System Configuration:** Provide an admin interface to configure system settings and parameters.

The Functional Features of Floodlight form the backbone of the Disaster Management Toolkit, empowering users with comprehensive tools and insights to efficiently respond to disasters, assess damages, and enhance overall disaster preparedness and recovery efforts.

1.3.3 Non-Functional Requirements:

Non-Functional Requirements for the Floodlight Disaster Management Toolkit define the qualities and constraints that govern the system's performance, security, usability, and other attributes. These requirements focus on the system's behavior and characteristics rather than its specific functionalities.

1.3.3.1 Performance:

- **Response Time:** The system should respond to user interactions within a reasonable time frame to ensure efficient data access and analysis.
- **Data Processing Speed:** Real-time data analysis and visualization should be performed with minimal delays to provide timely insights.
- **Scalability:** The system should be scalable to accommodate increasing data volumes and user loads during disaster events.

1.3.3.2 Security:

- **Data Privacy:** Ensure that user data is securely stored and transmitted, adhering to privacy regulations and protecting sensitive information.
- **User Authentication:** Implement robust user authentication mechanisms to prevent unauthorized access to the system.
- **Data Encryption:** Encrypt data transmissions and storage to protect against unauthorized interception and data breaches.

1.3.3.3 Usability:

- **User-Friendly Interface:** Design an intuitive and easy-to-navigate user interface to facilitate user adoption and reduce the learning curve.
- **Accessibility:** Ensure the toolkit is accessible to users with diverse abilities and needs, complying with accessibility standards.

1.3.3.4 Reliability:

- **System Availability:** The system should be available and accessible during critical times, even in the event of high user traffic or server failures.
- **Error Handling:** Implement effective error handling mechanisms to gracefully handle system errors and minimize downtime.

1.3.3.5 Compatibility:

- **Cross-Platform Compatibility:** Ensure compatibility with major web browsers and mobile devices to cater to a wide range of users.
- **Integration Compatibility:** Facilitate seamless integration with existing disaster management systems and APIs.

1.3.3.6 Maintainability:

- **Code Modularity:** Design the system with modular components for ease of maintenance and future enhancements.
- **Documentation:** Provide comprehensive technical documentation to assist developers in maintaining and extending the toolkit.

1.3.3.7 Scalability:

- **System Performance:** The system should maintain acceptable performance levels even as the user base and data volumes grow.
- **Resource Management:** Efficiently utilize server resources and minimize resource contention during peak usage.

1.3.3.8 Resilience:

- **Fault Tolerance:** The system should gracefully recover from failures and continue to function without significant disruptions.
- **Disaster Recovery:** Implement backup and recovery procedures to ensure data integrity and system continuity in the face of disasters.

1.3.3.9 Compliance:

- **Regulatory Compliance:** Ensure compliance with relevant industry standards, data protection regulations, and disaster management protocols.

The Non-Functional Requirements of Floodlight are critical for delivering a high-quality, reliable, and secure disaster management toolkit that can effectively support stakeholders in their disaster preparedness, response, and recovery efforts.

1.3.4 Inclusions and Exclusions:

The Inclusions and Exclusions section of the Floodlight Disaster Management Toolkit's Requirements Specification document outlines the specific features and functionalities that will be included and excluded from the project scope. This section helps to clearly define the boundaries of the project and set realistic expectations for stakeholders.

1.3.4.1 Inclusions:

- **Data Collection Module:** The Floodlight toolkit will include a data collection module that allows users to contribute disaster-related information through various channels, such as crowdsourcing, social media integration, and sensor data inputs.
- **Real-Time Data Analysis:** The system will perform real-time data analysis using machine learning algorithms to detect and assess potential disaster events and trends.
- **Early Warning and Alerting:** Floodlight will provide an early warning system to issue timely alerts and notifications to relevant authorities and affected communities based on incoming data.
- **Geographic Information System (GIS):** The toolkit will incorporate GIS capabilities to visualize disaster-related data on interactive maps, facilitating spatial analysis.
- **User Authentication:** The system will implement secure user authentication mechanisms to control access to sensitive data and functionalities.
- **Cross-Platform Compatibility:** Floodlight will be accessible through both web and mobile interfaces, ensuring compatibility with major browsers and devices.
- **Reporting and Data Export:** Users will be able to generate customized reports and export data in standard formats for further analysis and reporting.

1.3.4.2 Exclusions:

- **Physical Hardware:** The Floodlight project will not include the provision of physical hardware devices or infrastructure.
- **Physical Installation:** Installation and setup of physical hardware, sensors, or satellite equipment will not be part of the project scope.
- **External Data Sources:** The project will not be responsible for data collection from external sources not directly integrated into the Floodlight toolkit.
- **Hardware Maintenance:** Ongoing maintenance of hardware devices used for data collection, such as sensors or satellite systems, will not be handled by the project.
- **Disaster Response Operations:** Floodlight will not manage the on-ground disaster response operations itself but will support stakeholders with data-driven insights.
- **End-User Devices:** The project will not provide or manage end-user devices (e.g., smartphones) for data submission.
- **Integration with Third-Party Systems:** Any integration with external systems or APIs not specified in the project scope will be considered outside the scope.

Defining clear inclusions and exclusions helps prevent scope creep and ensures that all stakeholders have a shared understanding of the project's boundaries and deliverables. Any changes to the scope should be managed through a formal change control process to maintain project focus and alignment with objectives.

1.3.5 Constraints:

Constraints in the context of the Floodlight Disaster Management Toolkit refer to the limitations and restrictions that may impact the project's development, implementation, and deployment.

These constraints may arise from various factors, and it is essential to identify and address them to ensure the project's success.

Following are the constraints for Floodlight:

- **Budgetary Constraints:** The project may have budget limitations that restrict the allocation of resources for development, infrastructure, and ongoing maintenance.
- **Time Constraints:** The project may have strict timelines and deadlines to meet, necessitating efficient project planning and execution.
- **Resource Availability:** Availability of skilled human resources, technical expertise, and hardware may be limited, impacting the speed and scale of the project.
- **Data Accessibility:** Access to real-time and historical data, such as satellite imagery, may be restricted, affecting the completeness and accuracy of the information in the toolkit.
- **Regulatory and Legal Constraints:** Compliance with data privacy regulations, disaster management protocols, and legal requirements may impose constraints on data collection and usage.
- **Interoperability:** The need to integrate with existing disaster management systems, government databases, or external APIs may present compatibility challenges.
- **Internet Connectivity:** Limited or unreliable internet connectivity in disaster-prone areas may affect the usability and effectiveness of the toolkit.
- **Security and Privacy:** Strict security measures and privacy concerns may limit data sharing and access to certain functionalities.
- **Hardware and Infrastructure Limitations:** The performance and capacity of servers, databases, and other infrastructure components may impose technical constraints.
- **Accessibility:** Ensuring accessibility for users with disabilities or in areas with limited technology infrastructure can be a constraint.
- **Stakeholder Collaboration:** Coordination and collaboration among various stakeholders, such as government authorities, NGOs, and communities, may present challenges.
- **Disaster Response Logistics:** Challenges in on-ground logistics and communication during disaster events may impact the toolkit's effectiveness.

Identifying and understanding these constraints is crucial for project planning and risk management. Properly addressing these constraints through mitigation strategies and contingency plans can help ensure that the Floodlight Disaster Management Toolkit is developed and implemented successfully, meeting the needs of its intended users and stakeholders despite any challenges that may arise.

1.3.6 Deliverables:

The Deliverables for the Floodlight Disaster Management Toolkit are the tangible outputs and outcomes that the project aims to produce upon successful completion. These deliverables represent the concrete results and assets that stakeholders can expect from the project.

Following are the key deliverables for Floodlight:

- **Floodlight Software Application:** The fully developed and functional Floodlight Disaster Management Toolkit, accessible through web and mobile interfaces.
- **User Documentation:** Comprehensive user manuals and guides that provide instructions on how to use the Floodlight system effectively.
- **Technical Documentation:** Detailed technical documentation that describes the system architecture, database schema, APIs, and other technical aspects of the toolkit.
- **GIS Integration:** Implementation of Geographic Information System (GIS) capabilities within the toolkit, allowing for spatial analysis and data visualization.
- **Data Collection Module:** A fully operational data collection module that enables users to contribute disaster-related information through various channels, including crowdsourcing, social media integration, and sensor data inputs.
- **Real-Time Data Analysis Algorithms:** The development and integration of real-time data analysis algorithms, such as machine learning models, to detect potential disaster events and trends.
- **Early Warning System:** A functioning early warning system that generates and disseminates alerts to relevant authorities and affected communities based on incoming data.
- **Communication and Collaboration Tools:** Features that facilitate real-time communication and collaboration between disaster management stakeholders, response teams, and affected communities.
- **Custom Reports and Data Export:** A reporting module that allows users to generate customized reports and export data in standard formats for further analysis and reporting.
- **Security and Privacy Measures:** Implement robust security measures, including data encryption, user authentication, and privacy controls, to protect sensitive information.
- **Integration with External Systems:** Successful integration with existing disaster management systems, government databases, and relevant APIs.
- **System Administration Interface:** An admin interface that allows system administrators to manage user roles, configure settings and monitor system health.
- **Training Materials:** Training materials and resources for stakeholders and end-users to help them effectively utilize the Floodlight toolkit.
- **Testing and Quality Assurance Reports:** Reports on testing procedures, test results, and quality assurance efforts to ensure the reliability and stability of the toolkit.
- **Documentation of Assumptions and Constraints:** A comprehensive record of assumptions made during the project and documentation of identified constraints and their impact on the toolkit.

Each of these deliverables contributes to successfully implementing and utilizing the Floodlight Disaster Management Toolkit, empowering stakeholders with the necessary tools and insights to improve disaster preparedness, response, and recovery efforts.

1.3.7 Assumptions:

Assumptions in the context of the Floodlight Disaster Management Toolkit refer to the factors and conditions that are considered to be true, but may not be verified or confirmed at the time of defining the project scope. These assumptions influence the project planning and decision-making process and can have implications for the project's success. It is essential to identify and document assumptions to understand their potential impact on the project.

Following are the assumptions for Floodlight:

- **Data Availability:** It is assumed that relevant data sources, such as satellite imagery, social media feeds, and sensor networks, will be available and accessible for real-time data collection and analysis.
- **Stakeholder Cooperation:** The assumption is that key stakeholders, including government authorities, NGOs, and disaster management agencies, will actively participate and collaborate in the use of the Floodlight toolkit.
- **Internet Connectivity:** It is assumed that Internet connectivity will be available in disaster-prone areas, allowing for seamless data submission and system access.
- **Data Accuracy:** The assumption is that data contributed through crowdsourcing and other channels will be accurate and reliable, providing valuable insights for disaster management.
- **Hardware and Infrastructure:** It is assumed that the required hardware and infrastructure, such as servers and databases, will be in place to support the Floodlight system's operation.
- **User Adoption:** The assumption is that end-users, including disaster management personnel and community members, will adopt and utilize the Floodlight toolkit effectively.
- **Security Compliance:** It is assumed that the project will adhere to relevant data privacy and security regulations and implement robust security measures to protect user data.
- **Integration Compatibility:** The assumption is that integration with existing disaster management systems, APIs, and government databases will be feasible and compatible.
- **Resource Availability:** It is assumed that the project will have access to the necessary human resources, technical expertise, and funding to successfully develop and implement the Floodlight toolkit.
- **Disaster Response Engagement:** The assumption is that disaster management authorities and response teams will actively engage with the early warning system and alerts generated by Floodlight.
- **System Scalability:** It is assumed that the Floodlight toolkit can be scaled to handle increased data volumes and user loads during disaster events.
- **Regulatory Compliance:** The assumption is that the project will comply with relevant regulatory and legal requirements related to disaster management and data handling.

Documenting these assumptions is crucial as it helps stakeholders recognize potential risks and uncertainties. As the project progresses, it is essential to validate and verify these assumptions to

ensure that they hold true and make any necessary adjustments to the project plan based on actual conditions and challenges encountered during development and implementation.

1.3.8 Dependencies:

Dependencies in the context of the Floodlight Disaster Management Toolkit refer to external factors, resources, or activities that the project relies on for its successful development and implementation. These dependencies can have a significant impact on the project's timeline, scope, and overall success. Identifying and managing dependencies is crucial to ensure a smooth project execution.

Following are the dependencies for Floodlight:

- **Data Sources and Providers:** The project is dependent on access to various data sources, such as satellite imagery providers, social media platforms, sensor networks, and other data providers, to collect real-time and historical disaster-related data.
- **Government and Stakeholder Cooperation:** Successful implementation of Floodlight relies on the active cooperation and engagement of government organizations, NGOs, and other disaster management stakeholders in contributing data and utilizing the toolkit.
- **Internet Connectivity and Infrastructure:** The project depends on reliable internet connectivity and the availability of the necessary hardware and infrastructure to host and operate the Floodlight system.
- **GIS and Mapping Services:** Integration with Geographic Information System (GIS) platforms and mapping services is a dependency for visualizing disaster data on interactive maps.
- **External APIs and Systems:** Integration with existing disaster management systems, government databases, and external APIs is a dependency to enable seamless data exchange and collaboration.
- **Availability of Skilled Resources:** The project relies on the availability of skilled software developers, data scientists, GIS experts, and other technical resources to design, develop, and maintain the Floodlight toolkit.
- **Funding and Budgetary Support:** Adequate funding and budgetary support are essential dependencies to ensure the project's development, hosting, and ongoing maintenance.
- **Data Privacy and Legal Compliance:** The project is dependent on adherence to data privacy regulations, legal requirements, and disaster management protocols to ensure proper data handling and usage.
- **Disaster Management Protocols and Standards:** The toolkit must align with established disaster management protocols and standards to ensure compatibility and interoperability with existing practices.
- **Community Participation:** Successful adoption and use of the Floodlight toolkit depend on active community participation in data submission, feedback, and collaboration.
- **Satellite Imagery and Sensor Data Availability:** Access to satellite imagery and sensor data is a critical dependency for real-time monitoring and analysis of disaster events.

- **Collaboration and Communication Platforms:** The project relies on collaboration and communication platforms to facilitate real-time interactions and data sharing among stakeholders.

Identifying and managing these dependencies throughout the project's lifecycle is essential to address potential risks and ensure that the necessary resources and conditions are in place for the successful development and implementation of the Floodlight Disaster Management Toolkit.

1.3.9 Acceptance Criteria:

The Acceptance Criteria for the Floodlight Disaster Management Toolkit are the specific conditions and requirements that must be met for the project to be considered successfully completed and accepted by stakeholders. These criteria define the expected outcomes, functionalities, and performance of the toolkit, ensuring that it aligns with the project objectives and meets the needs of its intended users. Acceptance criteria serve as the basis for evaluating whether the project deliverables meet the desired quality and functionality standards.

Following are the acceptance criterion for Floodlight:

- **Data Collection Module:** The data collection module should be fully functional and allow users to submit disaster-related information through multiple channels, such as crowdsourcing, social media, and sensor inputs.
- **Real-Time Data Analysis:** The system should accurately analyze incoming data in real time, detecting and classifying potential disaster events and trends with a specified level of accuracy.
- **Early Warning System:** The early warning system should generate and disseminate timely alerts to relevant stakeholders, including authorities and affected communities, based on incoming data.
- **GIS Integration:** The Geographic Information System (GIS) capabilities should allow for interactive data visualization on maps, enabling spatial analysis of disaster-related information.
- **User Authentication and Security:** The system should implement secure user authentication mechanisms and robust data encryption to protect user data and ensure data privacy.
- **Cross-Platform Compatibility:** The toolkit should be accessible through both web and mobile interfaces, compatible with major browsers and mobile devices.
- **Reporting and Data Export:** Users should be able to generate customized reports and export data in standard formats (e.g., CSV, JSON) for further analysis and reporting.
- **Integration with External Systems:** Successful integration with existing disaster management systems, government databases, and relevant APIs should be achieved.
- **Usability and User-Friendliness:** The user interface should be intuitive, user-friendly, and accessible, catering to users with varying levels of technical expertise.

- **Performance and Scalability:** The system should demonstrate acceptable performance levels, even during peak usage and data volumes, and be scalable to handle increased demand.
- **Compliance with Standards and Regulations:** The toolkit should comply with relevant data privacy regulations, disaster management protocols, and industry standards.
- **Training and Documentation:** Comprehensive training materials and technical documentation should be provided to assist users in effectively using the Floodlight toolkit.

Meeting the acceptance criteria is essential for obtaining approval from stakeholders and ensuring that the Floodlight Disaster Management Toolkit meets its intended goals and delivers the expected value to disaster management efforts.

1.4 Stakeholders

Role/Name	Contact	Expectations
NGOs	Email: Phone:	• Expect timely alerts and accurate data to effectively coordinate disaster response efforts. • Collaborate with other stakeholders and access real-time information for informed decision-making. • Contribute data on the ground to support early warning systems and disaster mapping.
Government Authorities	Email: Phone:	• Receive real-time updates and early warnings to facilitate prompt disaster response and assistance. • Access accurate GIS data for effective disaster assessment and resource allocation. • Ensure the toolkit complies with data privacy regulations and government standards.
Community Members	Email: Phone:	• Receive timely alerts and safety information to protect themselves and their communities. • Contribute disaster-related information and report incidents through the toolkit's data collection. • Expect an easy-to-use and accessible platform to participate in disaster management efforts.
World Bank	Email: Phone:	• Evaluate the impact of Floodlight in enhancing disaster management and

		resilience. ● Assess the effectiveness of data-driven decision-making for disaster recovery and reconstruction. ● Collaborate with Floodlight developers to integrate data into the World Bank's disaster management.
UN OCHA	Email: Phone:	● Use Floodlight's data for situational analysis and coordination of humanitarian assistance. ● Expect real-time information sharing and collaboration with other humanitarian agencies. ● Support and contribute to Floodlight's development and interoperability with existing systems.
Technical Team	Email: Phone:	● Develop and maintain the Floodlight Disaster Management Toolkit according to specified requirements. ● Ensure the toolkit's performance, security, and scalability align with project objectives.
Funding Organizations	Email: Phone:	● Evaluate Floodlight's potential impact and alignment with funding organization's objectives. ● Expect regular progress reports and accountability in resource utilization. ● Assess the project's sustainability and potential for future funding opportunities.

2. Overall Description

2.1 Product Perspective

The Floodlight Disaster Management Toolkit is a standalone software product specifically designed to improve disaster management capabilities in Pakistan. It is a new and self-contained system that does not replace any existing solutions but complements and enhances disaster response efforts through innovative geospatial and data-driven functionalities.

2.1.1 Origin and Context:

The origin of the Floodlight Toolkit lies in the recognition of the challenges faced during the 2022 floods in Pakistan. The existing disaster management practices often struggle with obtaining real-time and accurate information, leading to delays in response and resource allocation. The

Floodlight project was initiated to address these gaps and empower disaster management stakeholders with timely and actionable data.

2.1.2 Uniqueness and Geospatial Aspects:

One of the key differentiators of the Floodlight Toolkit is its strong geospatial component. The system will extensively utilize geospatial data, including satellite imagery, GIS, and real-time sensor inputs, to visualize disaster events, assess their spatial distribution, and identify affected areas. This geospatial aspect provides valuable insights for decision-makers, allowing them to prioritize response efforts based on the severity and location of the disaster.

2.1.3 System Interfaces:

The Floodlight Toolkit interacts with various external systems and stakeholders through different interfaces:

- **User Interfaces:** The system provides web-based and mobile interfaces for different user groups, including government authorities, NGOs, and community members. These interfaces offer data submission, visualization, and communication tools.
- **Hardware Interfaces:** Floodlight integrates with external sensors and data sources, such as satellite providers and weather monitoring systems, to collect real-time data.
- **Software Interfaces:** The toolkit may interface with existing disaster management systems, government databases, and APIs to share and access relevant data.

2.1.4 Dependencies and Interactions:

The Floodlight Toolkit depends on:

- **Data Sources:** Access to reliable data sources, such as satellite imagery providers, social media platforms, and sensor networks, for real-time data collection and analysis.
- **Government and Stakeholder Cooperation:** Active engagement and collaboration with government authorities, NGOs, and community members to contribute data and utilize the toolkit.
- **Internet Connectivity and Infrastructure:** Reliable internet connectivity and the necessary hardware and servers to host and operate the Floodlight system.

2.1.5 External Interfaces and Stakeholders:

Floodlight interacts with various external entities:

- **Government Agencies:** Collaboration with government disaster management authorities for data sharing, coordination, and resource mobilization during disasters.
- **NGOs and Humanitarian Organizations:** Interaction with NGOs for real-time updates, coordination, and information sharing to enhance disaster response efforts.
- **Community Members:** Engaging with community members through social media integration and crowdsourcing to gather valuable on-the-ground data.
- **Satellite Imagery Providers:** Accessing satellite imagery and GIS data providers for real-time mapping and analysis of disaster situations.

2.1.6 System Boundaries:

The Floodlight Toolkit operates as a self-contained system, with clear boundaries defined in its scope. It does not replace or integrate with existing disaster management systems but aims to complement and enhance their capabilities through advanced data analysis and geospatial features.

Overall, the Floodlight Disaster Management Toolkit is a novel and innovative solution that brings together geospatial insights, real-time data, and collaboration tools to empower stakeholders in making data-driven decisions during disaster events in Pakistan. By addressing the unique geospatial aspects of disaster management, Floodlight strives to improve preparedness, response, and recovery efforts, ultimately minimizing the impact of disasters on affected communities.

2.2 Product Features

The Floodlight Disaster Management Toolkit offers a range of essential features that empower users to effectively respond to disasters and manage crisis situations. These features include:

2.2.1 Real-Time Data Collection:

The toolkit allows users to collect real-time data from various sources, including crowdsourcing, social media, and sensor networks, to provide up-to-date information on disaster events.

2.2.2 Geospatial Data Visualization:

Floodlight incorporates geospatial data and GIS capabilities to visualize disaster situations on interactive maps, enabling users to analyze and understand the spatial distribution of disasters.

2.2.3 Early Warning System:

The system utilizes advanced analytics to detect potential disaster events and issues early warnings to relevant stakeholders, ensuring timely response and preparedness.

2.2.4 Collaboration and Communication:

Floodlight facilitates seamless communication and collaboration among government agencies, NGOs, and communities, promoting effective coordination during disaster response efforts.

2.2.5 Data Analysis and Insights:

The toolkit includes data analysis tools that offer insights into disaster trends, impact assessment, and resource allocation, aiding decision-making processes.

2.2.6 Automated Report Generation:

Users can generate customized reports summarizing disaster-related data and analysis for monitoring and evaluation purposes and sharing with stakeholders.

2.2.7 Integration with External Systems:

Floodlight integrates with external disaster management systems, databases, and APIs, fostering interoperability and data exchange with relevant stakeholders.

2.2.8 Early Recovery Planning:

The toolkit assists in post-disaster early recovery planning, supporting stakeholders in strategizing and implementing recovery and rehabilitation efforts.

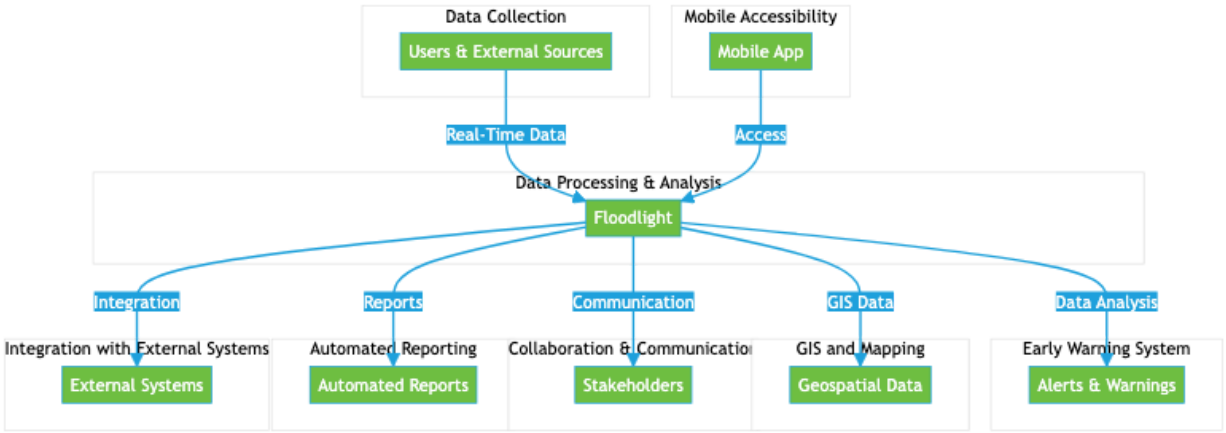
2.2.9 Disaster Mapping and Incident Tracking:

Users can create incident reports on the map, track the progress of incidents, and analyze trends over time for better situational awareness.

2.2.10 Mobile Accessibility:

Floodlight is accessible via mobile devices, enabling users to report and access disaster information even in areas with limited internet connectivity.

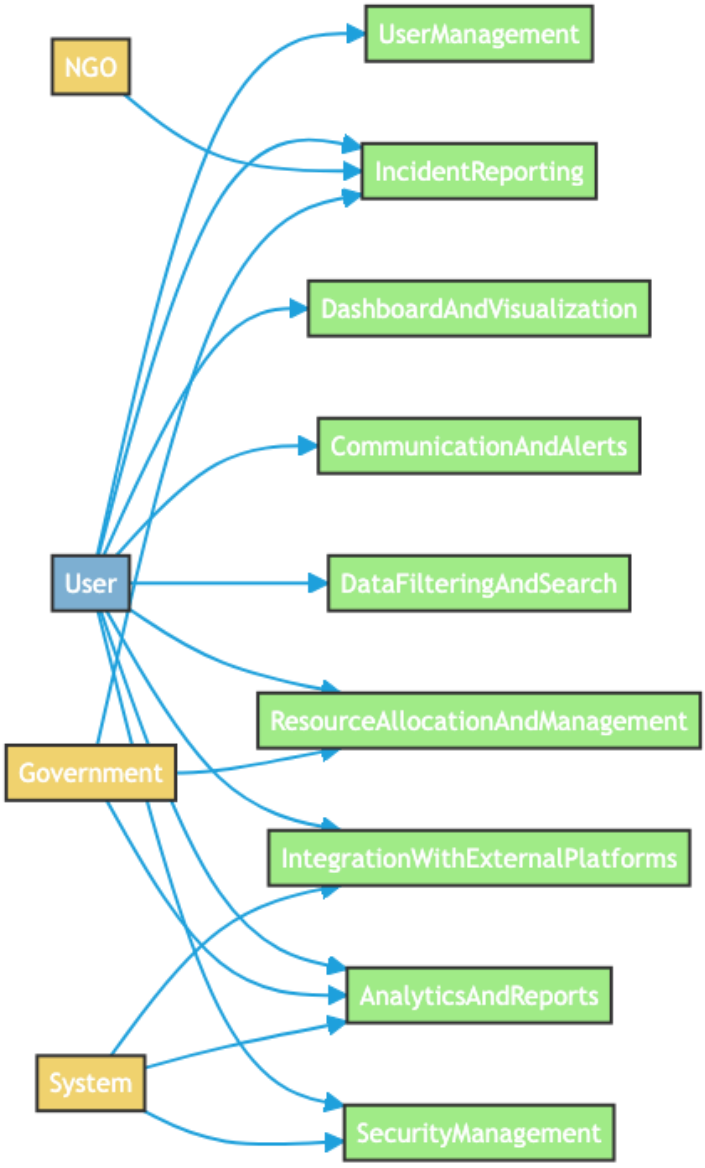
2.3 High-Level Data Flow Diagram:



The data flow diagram provides an overview of the major components and their interactions within the Floodlight system. It illustrates how data is collected, processed, and shared among stakeholders, highlighting the seamless flow of information for effective disaster management.

The Product Features of Floodlight, as summarized above, form the foundation of the toolkit's capabilities, empowering users with data-driven insights, collaboration tools, and geospatial visualization to enhance disaster preparedness, response, and recovery efforts.

2.4 User Classes and Characteristics



2.4.1 Government Authorities and Disaster Management Agencies:

Characteristics: These users will have high-frequency use of the toolkit during disaster events. They are experienced in disaster management, possess technical expertise, and require access to all product functions. They have higher privilege levels for data access and decision-making.
Importance: Favored user class as they play a critical role in coordinating disaster response efforts and require comprehensive access to all features.

2.4.2 Non-Governmental Organizations (NGOs) and Humanitarian Agencies:

Characteristics: NGOs will use the toolkit regularly for disaster response and resource allocation. They may have varied technical expertise and experience in disaster management. They need access to data sharing, collaboration tools, and reporting features.
Importance: Favored user class as they are key stakeholders in disaster relief operations and rely on the toolkit for efficient coordination.

2.5 Local Community Members:

Characteristics: These users may have limited technical expertise but can actively participate during disaster events through the mobile app's crowdsourcing features. They need user-friendly interfaces and access to incident reporting and early warning alerts.
Importance: Favored user class as they provide crucial on-the-ground data and play a vital role in disaster information collection.

2.6 Social Media Contributors:

Characteristics: These users contribute disaster-related data via social media integration. They may not frequently use the toolkit but play a significant role during crisis situations. They require secure and efficient data submission methods.
Importance: Important user class as their contributions enrich real-time data collection and situational awareness.

2.7 Technical Experts and Data Analysts:

Characteristics: These users are responsible for managing and analyzing data within the toolkit. They have a high level of technical expertise and require advanced data processing capabilities, data visualization, and reporting tools.
Importance: Favored user class as they ensure the accuracy and quality of data analysis and insights.

2.8 Decision-Makers and Senior Management:

Characteristics: These users rely on the toolkit for strategic decision-making and resource allocation during disaster response and recovery. They need access to high-level summaries, reports, and geospatial analysis.
Importance: Favored user class as their decisions are critical for effective disaster management.

2.9 External Researchers and Donors:

Characteristics: These users may not use the toolkit frequently but require access to aggregated data and impact assessment reports. They may have different technical expertise and educational levels.

Importance: Important user class as their involvement may contribute to future enhancements and funding.

It is essential to prioritize government authorities, NGOs, and local community members as the favored user classes, as they are the primary stakeholders in disaster management efforts. While all user classes are important, addressing the specific needs of these favored user groups will ensure that the Floodlight Disaster Management Toolkit meets its primary objectives effectively.
2.10

2.11 Operating Environment and Technology Stack

2.11.1 Operating Environment:

The FloodLight Disaster Management Toolkit will operate in a dynamic environment that combines existing open-source disaster management platforms, such as Ushahidi and Sahana, with custom-developed integrations and additional features. The operating environment includes:

Web-based Platform: FloodLight will provide a web-based interface accessible through modern web browsers, allowing users to access the combined functionalities seamlessly.
Mobile App: To enable real-time data collection and reporting from the field, FloodLight will offer a mobile application compatible with Android and iOS devices. The app will integrate with open-source platforms and custom developments.
Internet Connectivity: Since FloodLight relies on cloud-based open-source platforms and custom developments, a stable internet connection will be necessary for data synchronization, real-time updates, and communication.
Cloud Infrastructure: FloodLight may utilize cloud services, such as AWS, Azure, or Google Cloud, to host the integrated open-source platforms, databases, and custom-developed components, ensuring scalability and accessibility.

2.11.2 Technology Stack:

The technology stack for FloodLight Disaster Management Toolkit combines open-source platforms, custom developments, and interfacing components to create a comprehensive disaster management solution:

Open-Source Platforms:

- **Ushahidi:** A crowd-sourcing platform for data collection and real-time incident reporting.
- **Sahana Eden:** An emergency management platform for resource tracking, incident management, and logistics coordination.

Other open-source platforms for specific functionalities, such as early warning systems or geospatial visualization.

Custom Development:

- **Programming Languages:** JavaScript (Node.js), Python, or other suitable languages for integrating and extending functionalities.
- **APIs and Webhooks:** Custom APIs and webhooks to connect and communicate between open-source platforms and custom components.

Database Management System:

- MongoDB, PostgreSQL, or other compatible databases for storing and managing disaster-related data from the integrated platforms.

Geospatial Tools and Libraries:

- Geospatial libraries like Leaflet or Mapbox for interactive map visualization and integration with the open-source platforms.

Real-time Data Processing:

- Custom scripts and data processing workflows to handle real-time data streams from the open-source platforms.

Communication and Collaboration:

- Integration of real-time communication and collaboration features from the open-source platforms and custom development.

Security and Authentication:

- Utilization of security features provided by the open-source platforms and custom development for user authentication and data protection.

Containerization and Orchestration:

- Docker for containerization to ensure consistency and easy deployment of integrated components.

DevOps and CI/CD:

- Git for version control and collaborative development of custom components and integrations.

Cloud Services:

- Cloud services like AWS, Azure, or Google Cloud to host the integrated open-source platforms and custom components.

2.12 System Architecture:

FloodLight operates in a distributed and modular architecture, where open-source platforms and custom components are integrated seamlessly. The web-based platform, mobile app, and cloud infrastructure form the core components, allowing stakeholders to interact with the system in real-time. Custom-developed APIs and webhooks facilitate data exchange between open-source platforms and enable a cohesive user experience.

The entire system is containerized using Docker, promoting portability and ease of deployment across various environments. Cloud services further enhance scalability, ensuring the system can handle increased user loads during disaster events.

In summary, FloodLight Disaster Management Toolkit thrives in an adaptive environment, combining existing open-source platforms with tailored developments to offer a unified solution for effective disaster preparedness, response, and recovery.

2.13 Design and Implementation Constraints

- **Open-Source Platforms Compatibility:** The FloodLight toolkit's design and implementation must ensure seamless integration and compatibility with the selected open-source disaster management platforms, such as Ushahidi and Sahana Eden. Any custom developments and integrations must adhere to the APIs and data formats provided by these platforms.
- **Regulatory and Privacy Compliance:** The development of FloodLight must adhere to relevant regulatory policies and privacy laws. Data handling and storage practices must prioritize user privacy and comply with data protection regulations.
- **Security Considerations:** FloodLight's design must prioritize robust security measures to protect sensitive disaster-related data and ensure secure user authentication and authorization. Encryption and secure communication protocols should be implemented to prevent data breaches.
- **Hardware Limitations:** The toolkit's design and implementation should consider hardware limitations, especially for mobile devices used by field personnel. Optimal performance and efficiency must be maintained across various hardware configurations.
- **Communication Protocols:** To ensure smooth data exchange between the toolkit and various stakeholders, adherence to standard communication protocols, such as HTTP, HTTPS, WebSockets, or WebRTC, is essential.
- **Geospatial Data Standards:** FloodLight should adhere to established geospatial data standards to ensure interoperability with geographic information systems (GIS) and spatial analysis tools.
- **Parallel Operations:** The toolkit should be designed to handle concurrent operations, especially during peak disaster events, to maintain responsiveness and scalability.

- **Technology Stack Selection:** The choice of programming languages, databases, and other technologies must be compatible with the overall system architecture and align with the toolkit's performance and scalability goals.
- **User Interface and User Experience (UI/UX) Design:** FloodLight's UI/UX should follow established design conventions and be intuitive, ensuring ease of use for stakeholders with varying levels of technical expertise.
- **Data Visualization and Interpretation:** The design should focus on presenting data in a user-friendly manner, enabling stakeholders to interpret and analyze information effectively.
- **Maintainability and Extensibility:** The implementation should consider code maintainability and extensibility, allowing future updates, enhancements, and integration with new features or open-source platforms.
- **Documentation and Knowledge Transfer:** Comprehensive documentation should be provided to aid system administrators, developers, and stakeholders in understanding the toolkit's architecture, functionalities, and usage.
- **Resource Constraints:** The design and implementation should consider resource constraints, such as limited internet bandwidth and intermittent connectivity during disaster situations.
- **Cultural and Linguistic Considerations:** The toolkit's user interface should be culturally sensitive and accommodate multiple languages to facilitate usability among diverse communities.

By acknowledging and addressing these design and implementation constraints, the FloodLight Disaster Management Toolkit can be developed with careful consideration for its functionality, usability, security, and compliance requirements, resulting in an effective and valuable tool for disaster management and response.

2.14 User Documentation

- **User Manual:** A comprehensive user manual will be provided, offering detailed instructions on how to use the FloodLight toolkit's various features and functionalities. The manual will cover all aspects of the platform, including data collection, incident reporting, data visualization, and collaboration.
- **Online Help:** An interactive online help system will be available within the web-based platform and mobile app. Users can access context-sensitive help to get immediate assistance on specific tasks or features they are using.
- **Tutorials and Walkthroughs:** Video tutorials and step-by-step walkthroughs will be created to guide users through essential processes, such as setting up accounts, creating surveys, interpreting data visualizations, and managing incidents. These tutorials will help users get started with FloodLight quickly and efficiently.
- **FAQs and Troubleshooting Guide:** A list of frequently asked questions (FAQs) will be provided, addressing common queries and challenges users may encounter. Additionally,

a troubleshooting guide will assist users in resolving any technical issues they may face during their interaction with the toolkit.

- **Data Dictionary:** A comprehensive data dictionary will be included, explaining the meaning and context of various data fields and attributes used in the system. This will help users better understand and interpret the collected data.
- **Accessibility Guidelines:** The documentation will adhere to accessibility guidelines to ensure that users with disabilities can access and understand the information effectively.
- **Multilingual Support:** The user documentation will be available in multiple languages to accommodate the diverse user base and facilitate usability across different regions and communities.
- **Delivery Formats:** The user documentation will be provided in electronic formats, such as PDFs and web-based help systems, to allow easy access and distribution. Video tutorials will be accessible through online platforms like YouTube.
- **Standardization and Consistency:** The documentation will follow industry-standard formatting, design, and language conventions, ensuring consistency and ease of navigation for users.
- **Release Notes:** Regularly updated release notes will be included, summarizing the changes and enhancements made in each software update, ensuring users stay informed about new features and improvements.
- **Integration Guidelines:** For organizations and stakeholders integrating FloodLight into their existing systems, integration guidelines will be provided, detailing the APIs, data formats, and procedures for seamless integration.

The User Documentation will play a vital role in ensuring that users can effectively utilize the FloodLight Disaster Management Toolkit to its full potential, empowering them to make informed decisions, respond to disasters, and collaborate for effective disaster management.

2.15 Assumptions and Dependencies

2.15.1 Assumptions:

- **Open-Source Platform Compatibility:** Assumed that the integration of Ushahidi and Sahana Eden with custom-developed components will be seamless and compatible, allowing data exchange and collaboration without major obstacles.
- **Stable Internet Connectivity:** Assumes that users will have access to a stable internet connection for real-time data synchronization and communication with the FloodLight toolkit.
- **Regulatory Compliance:** Assumes that the development and operation of FloodLight will adhere to relevant regulatory policies and privacy laws, ensuring data protection and compliance with applicable standards.

- **Availability of Mobile Devices:** Assumes that field personnel and stakeholders will have access to smartphones or other mobile devices with internet connectivity to use the FloodLight mobile app for data collection and reporting.
- **Cloud Service Reliability:** Assumes that the chosen cloud infrastructure service (e.g., AWS, Azure, or Google Cloud) will provide reliable and scalable resources for hosting the FloodLight toolkit.
- **Geospatial Data Standardization:** Assumes that open-source platforms and custom developments will follow established geospatial data standards to ensure interoperability and seamless geospatial data exchange.

2.15.2 Dependencies:

- **Third-Party Open-Source Platforms:** FloodLight's functionality heavily depends on the stable and continuous operation of third-party open-source platforms like Ushahidi and Sahana Eden. Any changes or disruptions to these platforms may impact the functionality of FloodLight.
- **API and Data Format Dependencies:** The successful integration of open-source platforms and custom developments relies on the availability and proper functioning of APIs and data formats provided by these platforms.
- **Development Team Expertise:** The development of FloodLight depends on the expertise of the development team in integrating different open-source platforms, writing custom APIs, and ensuring smooth data exchange.
- **Software Components Reuse:** The project may rely on reusing specific software components or libraries from other projects to enhance functionality or improve performance.
- **External Service Providers:** The project's success may depend on the reliability and performance of external service providers, such as cloud infrastructure services or geospatial data providers.
- **User Adoption and Participation:** The effectiveness of FloodLight depends on user adoption and active participation by stakeholders and users, including NGOs, government agencies, and communities.
- **Data Availability:** FloodLight's functionality relies on the availability of relevant and accurate data from various sources, including crowdsourced data, government data, and remote sensing data.
- **Continuous Support and Maintenance:** The project's long-term success depends on ongoing support and maintenance to ensure the system remains operational, secure, and up-to-date.

It is essential to identify and manage these assumptions and dependencies throughout the development and deployment process of the FloodLight Disaster Management Toolkit to mitigate risks and ensure successful implementation.

3. System Features

3.1 Incident Reporting and Crowdsourcing

3.1.1 Description and Priority

This feature allows users to report incidents and disaster-related information, promoting crowdsourced data collection for real-time response efforts. It is of High priority, as it forms the foundation of Floodlight's data aggregation and engagement.

3.1.2 Stimulus/Response Sequences

User submits incident report via web form:

1. User accesses the Floodlight platform and fills out the incident report form.
    - System validates and processes the submitted information.
    - System generates a unique incident ID and timestamp.
    - System sends a confirmation notification to the user.
    - User reports incident via SMS:
2. User sends an SMS with incident details.
    - SMS is received by the Floodlight system.
    - System extracts relevant information and converts it into structured data.
    - System responds with an acknowledgment SMS.

3.1.3 Functional Requirements

3.1.3.1 REQ-1: Incident Submission via Web Form

- Users should be able to access the incident report form through the Floodlight platform.
- The form must include fields for incident type, location (geographical coordinates or text), description, and optional media attachments (photos, videos).
- The system should validate user inputs for accuracy and completeness.
- If any fields are incomplete or invalid, the system should provide appropriate error messages.
- Upon successful submission, the system should generate a unique incident ID and timestamp for reference.

3.1.3.2 REQ-2: SMS Incident Reporting

- The system should support incident reporting through SMS using a designated phone number.
- Users must be able to send SMS containing incident details, such as "FLOOD" followed by location and description.
- The system should have an SMS gateway to receive and process incoming SMS.
- In cases of ambiguous or incomplete messages, the system should respond with requests for clarification.
- Upon successful processing, the system should reply with an acknowledgment SMS confirming the incident report.

3.1.3.3 REQ-3: Incident Categorization

- The system should categorize incidents based on type, such as floods, earthquakes, medical emergencies, etc.
- Incident types should be predefined and structured for consistent data aggregation.
- Users must select the appropriate incident type when submitting a report through the web form.

3.1.3.4 REQ-4: Crowdsourced Data Validation

- The system should implement moderation and validation mechanisms for crowdsourced data.
- Moderators should review and approve incoming incident reports to ensure accuracy.
- If an incident report is deemed inaccurate or misleading, moderators should have the ability to reject or request clarification from the user.

3.1.3.5 REQ-5: Incident Timestamping

- The system should automatically timestamp each incident report upon submission.
- Timestamps must be accurate and reflect the time of submission in the user's local time zone.
- Timestamps will be used for tracking incident chronology and response timelines.

3.2 Dashboard and Visualizations

3.2.1 Description and Priority

This feature involves the creation of intuitive dashboards and visualizations to present real-time disaster data in a comprehensive and user-friendly manner. It is of High priority as it enables stakeholders to gain insights and make informed decisions during emergency response.

3.2.2 Stimulus/Response Sequences

1. User accesses the dashboard:
    - User logs in to the Floodlight platform.
    - User navigates to the dashboard section.
    - System retrieves and loads relevant data for display.
    - Dashboard interface and visualizations are rendered for the user.
2. User interacts with visualizations:
    - User interacts with charts, graphs, and maps on the dashboard.
    - System responds to user interactions, providing dynamic updates.
    - Users can filter data based on time range, incident type, and location.
    - Visualizations adjust to reflect the selected filters.

3.2.3 Functional Requirements

3.2.3.1 REQ-1: Interactive Dashboard Interface

- The dashboard interface should provide an overview of key disaster-related metrics.
- Users should be able to customize their dashboard layout and arrangement.
- Visualizations should be organized logically and offer easy navigation.

3.2.3.2 REQ-2: Real-time Data Integration

- The dashboard should seamlessly integrate with the underlying data sources, ensuring real-time updates.
- Data should be fetched, processed, and displayed without noticeable delays.
- Users should see accurate and up-to-date information at all times.

3.2.3.3 REQ-3: Visualizations for Incident Distribution

- The dashboard should include a map visualization showing the geographic distribution of reported incidents.
- Users should be able to interact with the map, zooming in/out and clicking on markers for incident details.
- Incident markers should differentiate between incident types through color or icons.

3.2.3.4 REQ-4: Graphical Representation of Trends

- The dashboard should offer various graphical representations of incident trends over time.
- Line graphs, bar charts, and pie charts can illustrate incident frequency, distribution, and proportions.
- Users should be able to adjust the time range for trend analysis.

3.2.3.5 REQ-5: Filter and Drill-down Functionality

- Users should have the ability to filter data by incident type, location, and time period.
- Filtered data should be presented in both map and chart visualizations.
- Users can drill down into specific data points to obtain more detailed information.

3.2.3.6 REQ-6: Responsive Design

- The dashboard should be designed to provide a consistent experience across different devices.
- Visualizations and interface elements should adapt to different screen sizes.

3.2.3.7 REQ-7: Error Handling and Notifications

- If the dashboard encounters errors in data retrieval or processing, users should be notified.
- Clear error messages should guide users on possible actions or provide support.

3.3 Communication and Alerts

3.3.1 Description and Priority

This feature involves setting up effective communication channels and alerts to facilitate real-time dissemination of critical information to stakeholders during disaster situations. It is of High priority as timely communication is vital for coordinating response efforts.

3.3.2 Stimulus/Response Sequences

1. Incident Report Triggering Alerts:
    a. User submits an incident report.

b. System validates and processes the report.
c. If the incident's severity exceeds a threshold, an alert is triggered.
d. Alert is sent to relevant stakeholders via selected communication channels.

2. Alerts to Emergency Responders:
   a. Upon receiving an alert, emergency responders are notified immediately.
   b. Notification can be via SMS, email, or in-platform notifications.
   c. Responders acknowledge receipt of the alert.
3. User Subscription Alerts:
   a. Users can subscribe to receive alerts about specific incident types or locations.
   b. When a subscribed incident occurs, users receive an alert via their chosen communication channel.
   c. Users can adjust their subscription preferences at any time.

3.3.3 Functional Requirements

3.3.3.1 REQ-1: Automated Alert Triggers

- The system should automatically trigger alerts based on predefined incident severity criteria.
- Incident reports exceeding these criteria should initiate alerts to relevant stakeholders.

3.3.3.2 REQ-2: Communication Channel Options

- Stakeholders should have the option to receive alerts via SMS, email, and platform notifications.
- Users can select their preferred communication channels in their profile settings.

3.3.3.3 REQ-3: Customizable Alert Preferences

- Users should be able to set their alert preferences, indicating which incident types they want to be alerted about.
- Users can adjust these preferences through their profile settings.

3.3.3.4 REQ-4: Immediate Alert Delivery

- Alerts should be delivered to stakeholders in real-time.
- There should be minimal delay between incident report submission and alert receipt.

3.3.3.5 REQ-5: Acknowledgment and Tracking

- Emergency responders receiving alerts should be able to acknowledge receipt.
- The system should track the acknowledgment status of alerts.

3.3.3.6 REQ-6: Hierarchical Alert Escalation

- If an alert remains unacknowledged for a specified duration, the system should escalate the alert to higher-level responders.
- Escalation can be based on a hierarchical structure defined in the system.

3.3.3.7 REQ-7: Multilingual Alerts

- Alerts should be available in multiple languages, catering to diverse stakeholders.
- Users can select their preferred language for alert messages.

3.3.3.8 REQ-8: Error Handling and Retry

- If an alert fails to be delivered, the system should attempt to resend it.
- Clear error messages should be provided if repeated delivery attempts fail.

3.4 Data Filtering and Search

3.4.1 Description and Priority

This feature involves enabling users to efficiently filter and search through the flood-related data within the system. It is of Medium priority as it enhances usability by allowing stakeholders to access specific information quickly.

3.4.2 Stimulus/Response Sequences

1. Filtering Incidents by Type:
    - User selects an incident type (e.g., flood, earthquake) from a dropdown menu.
    - System retrieves and displays incidents matching the selected type.
    - Filtered data is presented to the user.
2. Searching by Location:
    - User enters a location (e.g., city, district) in the search bar.
    - System performs a search based on the entered location.
    - Search results display incidents and relevant data related to the specified location.
3. Time-Based Filtering:
    - User selects a time range (e.g., last 24 hours, last week) from a list.
    - System fetches incidents that occurred within the chosen timeframe.
    - Filtered data is presented to the user.

3.4.3 Functional Requirements

3.4.3.1 REQ-1: Incident Type Filter

- The system should provide a filter option allowing users to select incident types.
- Filtered results should only display incidents matching the selected type.

3.4.3.2 REQ-2: Location-Based Search

- Users should be able to search for incidents based on specific locations (e.g., city, district).
- The system should return results that match the entered location.

3.4.3.3 REQ-3: Time Range Filter

- Users should have the option to filter incidents based on specified time ranges.
- Filtered data should include incidents that occurred within the chosen timeframe.

3.4.3.4 REQ-4: User-Friendly Search Interface

- The search and filtering interface should be intuitive and easy to use.
- Clear labels and instructions should guide users through the process.

3.4.3.5 REQ-5: Real-Time Filtering

- Filtering and search results should update in real-time as users make selections.
- This ensures users receive immediate feedback on their filtering choices.

3.4.3.6 REQ-6: Search Error Handling

- If a search term doesn't yield any results, the system should provide relevant suggestions or guidance to refine the search.

3.4.3.7 REQ-7: Consistent Results Presentation

- Filtered and search results should be presented consistently, providing key incident details and data points.

3.4.3.8 REQ-8: Mobile Responsiveness

- The filtering and search interface should be responsive and user-friendly on both desktop and mobile devices.

3.4.3.9 REQ-9: Accurate Time-Based Filtering

- The system should accurately retrieve incidents that occurred within the specified time range.
- Timezone considerations should be taken into account to ensure accuracy.

3.5 Resource Allocation and Management

3.5.1 Description and Priority:

This feature is of high priority as it directly impacts the efficient allocation and management of resources during disaster response. Effective resource allocation is crucial for timely assistance and minimizing the impact of disasters.

3.5.2 Stimulus/Response Sequences:

- **Stimulus:** Government agencies input data on available resources, including personnel, equipment, and relief supplies, into the Floodlight system.
- **Response:** Floodlight processes this data and makes it accessible through dashboards and reports. NGOs and government authorities can view the available resources and their locations in real-time.

3.5.3 Functional Requirements:

3.5.3.1 REQ-1: Resource Input:

- Government agencies should be able to input data about available resources, including type, quantity, location, and availability status.

3.5.3.2 REQ-2: Resource Visualization:

- Floodlight should provide visual representations of available resources on maps, including markers for the location of relief centers, medical facilities, and distribution points.

3.5.3.3 REQ-3: Resource Search:

- Users should be able to search for specific resources based on criteria like type, location, and availability.

3.5.3.4 REQ-4: Resource Status Updates:

- The system should allow agencies to update the availability status of resources in real-time.

3.5.3.5 REQ-5: Resource Allocation:

- Floodlight should facilitate the allocation of resources to specific incident locations based on priority and need.

3.5.3.6 REQ-6: Reporting:

- The system should generate reports on resource allocation, utilization, and availability for decision-makers.

3.5.4 Error Handling:

- Floodlight should provide error messages or alerts when there are discrepancies or inconsistencies in the resource data input.
- In cases of resource shortage or unavailability, the system should alert users and recommend alternative resources or locations.

3.6 Integration with External Platforms

3.6.1 Description and Priority:

This feature is of high priority as it enables Floodlight to integrate seamlessly with external platforms, enhancing its functionality and interoperability.

3.6.2 Stimulus/Response Sequences:

- **Stimulus:** Users initiate integration with external platforms, such as social media networks, messaging apps, and government databases.
- **Response:** Floodlight establishes connections with these platforms, retrieves data, and displays it within the system.

3.6.3 Functional Requirements:

3.6.3.1 REQ-1: Social Media Integration:

- Floodlight should be capable of integrating with popular social media platforms to monitor user-generated content related to disaster incidents.

3.6.3.2 REQ-2: Messaging App Integration:

- The system should allow integration with messaging apps like WhatsApp for real-time communication with affected individuals.

3.6.3.3 REQ-3: Government Database Integration:

- Floodlight should be able to connect with government databases to access official disaster-related data.

3.6.3.4 REQ-4: Data Retrieval:

- The system should retrieve data from external platforms, including text, images, and videos.

3.6.3.5 REQ-5: Data Filtering:

- Floodlight should provide tools to filter and categorize incoming data from external sources based on relevance and reliability.

3.6.3.6 REQ-6: Automated Alerts:

- The system should generate automated alerts and notifications based on incoming data, ensuring timely responses.

3.6.3.7 REQ-7: Error Handling:

- Floodlight should gracefully handle errors related to failed data retrieval or integration connection issues, providing informative error messages.

3.6.4 Error Handling:

- Floodlight should provide clear error messages and logs in case of failed integration attempts.
- The system should have mechanisms to reattempt integration if a connection is temporarily lost.

3.7 Analytics and Reports

3.7.1 Description and Priority:

This feature is of high priority as it empowers users with the ability to analyze data and generate reports, which are critical for informed decision-making during disaster incidents.

3.7.2 Stimulus/Response Sequences:

- **Stimulus:** Users initiate data analysis and reporting requests through the system interface.
- **Response:** Floodlight processes the requests, performs data analytics, and generates reports in various formats.

3.7.3 Functional Requirements:

3.7.3.1 REQ-1: Data Analysis Tools:

- Floodlight should provide built-in data analysis tools to allow users to explore and analyze disaster-related data, including historical incident trends and patterns.

3.7.3.2 REQ-2: Report Generation:

- The system should enable users to generate customized reports based on selected data parameters, such as incident type, location, and time frame.

3.7.3.3 REQ-3: Report Formats:

- Floodlight should support the generation of reports in multiple formats, including PDF, Excel, and interactive web-based reports.

3.7.3.4 REQ-4: Visualizations:

- The system should offer a variety of data visualization options, such as charts, graphs, and maps, to enhance data understanding.

3.7.3.5 REQ-5: Report Scheduling:

- Users should have the option to schedule automated report generation at specified intervals.

3.7.3.6 REQ-6: Export and Share:

- Floodlight should allow users to export reports and share them with stakeholders via email or other communication channels.

3.7.3.7 REQ-7: Error Handling:

- The system should gracefully handle errors related to data analysis, report generation, or data source unavailability, providing clear error messages.

3.7.4 Error Handling:

- Floodlight should log errors related to data analysis or report generation for debugging purposes.
- The system should provide user-friendly error messages in case of data analysis failures or invalid report requests.

4. External Interface Requirements

4.1 User Interfaces

To describe the User Interfaces for Floodlight, we'll outline the logical characteristics and elements of the interfaces between the system and its users. Please note that specific design details should be documented in a separate User Interface Specification. Here's an overview:

4.1.1 Dashboard Interface:

The dashboard provides an overview of the current disaster situation and relevant data visualizations.

4.1.1.1 Characteristics:

- Clean and intuitive design.
- Real-time data updates.
- Customizable widgets for users to select relevant information.
- Accessible via web browsers.
- Mobile-responsive for on-the-go access.

4.1.2 Incident Reporting Interface:

This interface allows users to report incidents and contribute data.

4.1.2.1 Characteristics:

- User-friendly incident reporting forms.
- Option to attach photos or documents.
- Geolocation tagging for precise incident location.
- Progress indicators for form completion.
- Immediate confirmation upon submission.

4.1.3 Data Visualization Interface:

This interface presents data visualizations based on the collected data.

4.1.3.1 Characteristics:

- Interactive charts, graphs, and maps.
- Filtering and sorting options for data exploration.
- Export capabilities to download visualized data.
- Drill-down features for more detailed insights.

4.1.4 Communication and Alerts Interface:

Handles communication between users, government agencies, and organizations.

4.1.4.1 Characteristics:

- Integrated messaging system.
- Automated alerts and notifications.
- WhatsApp integration for real-time updates.
- User profiles with contact information.
- Broadcast and targeted messaging options.

4.1.5 Search and Filtering Interface:

Enables users to search and filter data based on specific criteria.

4.1.5.1 Characteristics:

- Advanced search with multiple filters.
- Keyword-based search functionality.
- Saved search queries for quick access.
- Clear error messages for invalid search inputs.

These are the logical characteristics of the main user interfaces within Floodlight. Specific design elements, layouts, and standards should be documented in a separate User Interface Specification to ensure a consistent and user-friendly experience.
4.2

4.3 Hardware Interfaces

The hardware interfaces involve both logical and physical characteristics to ensure seamless interaction between the software product and the hardware components of the system. Here are the main hardware interfaces:

4.3.1 Web Browsers:

4.3.1.1 Logical Characteristics:

- Floodlight is accessible through standard web browsers, including Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.

- It supports the latest web standards (HTML5, CSS3) for a rich and responsive user experience.

4.3.1.2 Physical Characteristics:

- Users can access Floodlight via desktop computers, laptops, tablets, and smartphones.
- No additional hardware components are required to interact with Floodlight through web browsers.

4.3.2 Mobile Devices:

4.3.2.1 Logical Characteristics:

- Floodlight provides a mobile-responsive design for various screen sizes and resolutions.
- It offers a consistent user experience across mobile platforms.

4.3.2.2 Physical Characteristics:

- Users can access Floodlight on Android and iOS devices, including smartphones and tablets.
- Native mobile apps may be developed for specific platforms to enhance user experience, if required.

4.3.3 GPS and Geolocation Hardware:

4.3.3.1 Logical Characteristics:

- Floodlight utilizes GPS and geolocation data to accurately pinpoint incident locations.
- It integrates with the hardware's geospatial capabilities to collect precise location information.

4.3.3.2 Physical Characteristics:

- Devices with GPS and geolocation capabilities (e.g., smartphones, tablets) can provide accurate incident location data.

4.3.4 Communication Hardware:

4.3.4.1 Logical Characteristics:

- Floodlight incorporates communication hardware to facilitate messaging and alerts.
- It may utilize standard communication protocols such as SMS and WhatsApp.

4.3.4.2 Physical Characteristics:

- Users can send and receive messages through devices with communication capabilities (e.g., mobile phones).

4.3.5 Desktop Computers:

4.3.5.1 Logical Characteristics:

- Floodlight is designed to be compatible with various desktop operating systems (e.g., Windows, macOS, Linux).
- It offers a consistent user interface on desktop platforms.

4.3.5.2 Physical Characteristics:

- Desktop computers with internet connectivity can access Floodlight via web browsers.

4.3.6 Server Infrastructure:

4.3.6.1 Logical Characteristics:

- Floodlight relies on server infrastructure to host the web application and databases.
- It communicates with the server-side components for data storage and processing.

4.3.6.2 Physical Characteristics:

- The server infrastructure may include cloud-based servers, on-premises hardware, or a combination to ensure system reliability and scalability.

These hardware interfaces enable Floodlight to function effectively across a wide range of devices and platforms, ensuring accessibility and usability for both government agencies and citizens. The use of standard communication protocols allows for seamless data exchange and real-time alerts.

4.4 Software Interfaces

Various software interfaces are established to enable communication and data exchange with other software components, databases, and integrated tools. These interfaces play a crucial role in ensuring the functionality and effectiveness of the system. Here are the key software interfaces:

4.4.1 Web-based User Interface (UI):

- **Purpose:** Provides a graphical interface for users to interact with Floodlight.
- **Data:** Users input data through forms and interfaces (e.g., incident reports, location data).
- **Services:** Allows users to submit reports, view dashboards, and access visualizations.
- **Communication:** Uses standard web protocols (HTTP/HTTPS) for client-server interactions.

4.4.2 Backend Server:

- **Purpose:** Serves as the core of Floodlight, handling data storage, processing, and user requests.
- **Data:** Receives, processes, and stores user-generated data (e.g., incident reports, geospatial information).
- **Services:** Provides data storage, retrieval, and processing services.
- **Communication:** Employs APIs and protocols (RESTful APIs, GraphQL) for communication between the UI and server components.

4.4.3 Geospatial Databases:

- **Purpose:** Stores geolocation data, incident coordinates, and geospatial information.

- **Data:** Stores geographic coordinates, maps, and spatial data layers.
- **Services:** Provides geospatial data retrieval and mapping services.
- **Communication:** Utilizes geospatial database connectors and queries to access and update geographic data.

4.4.4 External Communication Services:

- **Purpose:** Facilitates communication with external platforms and messaging services.
- **Data:** Sends and receives messages, alerts, and notifications (e.g., SMS, WhatsApp).
- **Services:** Integrates with external communication APIs and services.
- **Communication:** Utilizes specific communication protocols (e.g., Twilio API for SMS).

4.4.5 Data Visualization Libraries:

- **Purpose:** Enables the creation of interactive visualizations and dynamic dashboards.
- **Data:** Retrieves data for visualization from the backend and databases.
- **Services:** Offers charting and visualization services (e.g., D3.js, Chart.js).
- **Communication:** Utilizes JavaScript and libraries to render data in graphical formats.

4.4.6 Mapping and GIS Tools:

- **Purpose:** Supports geospatial data visualization and mapping.
- **Data:** Retrieves geographic data and incident coordinates.
- **Services:** Offers mapping services (e.g., Leaflet, Google Maps API).
- **Communication:** Utilizes APIs provided by mapping tools for displaying maps and overlays.

4.4.7 External Data Sources:

- **Purpose:** Gathers external data (e.g., weather forecasts, social media feeds) for enhanced situational awareness.
- **Data:** Ingests data feeds and APIs from various sources.
- **Services:** Integrates external data into Floodlight's information ecosystem.
- **Communication:** Utilizes RESTful APIs, web scraping, or data ingestion techniques.

4.4.8 Authentication and Authorization Services:

- **Purpose:** Ensures secure user access and data protection.
- **Data:** Manages user credentials, roles, and permissions.
- **Services:** Provides user authentication and authorization services.
- **Communication:** Utilizes security protocols (e.g., OAuth, JWT) for user authentication.

These software interfaces form the foundation for Floodlight's functionality, enabling it to communicate with users, databases, external services, and visualization tools. The proper

functioning of these interfaces is essential for seamless data exchange, real-time alerts, and effective disaster management.

4.5 Communications Interfaces

Floodlight relies on various communication interfaces to ensure the exchange of information, alerts, and data between different components and external entities. These communication interfaces are critical for the system's effectiveness in disaster management. Here are the key communication interfaces and their associated requirements:

4.5.1 Web-Based Communication:

- **Purpose:** Facilitates communication between the web-based user interface and the backend server.
- **Requirements:**
    - Use of HTTP/HTTPS protocols for secure data transfer.
    - Support for RESTful API endpoints for data retrieval and submission.
    - Data formats specified in JSON for efficient and standardized communication.

4.5.2 External Communication Services:

- **Purpose:** Enables the system to send alerts and notifications to external platforms and users.
- **Requirements:**
    - Integration with external communication APIs (e.g., Twilio for SMS alerts).
    - Support for asynchronous messaging to reach users on various channels.
    - Secure and authenticated communication with external services.

4.5.3 Geospatial Data Communication:

- **Purpose:** Supports the exchange of geospatial data between Floodlight and external mapping services.
- **Requirements:**
    - Integration with mapping and GIS tools (e.g., Leaflet, Google Maps API).
    - Utilization of geospatial data formats (e.g., GeoJSON) for data interchange.
    - Real-time updates of map data for incident visualization.

4.5.4 External Data Integration:

- **Purpose:** Ingests data from external sources (e.g., weather forecasts, social media feeds) to enhance situational awareness.
- **Requirements:**
    - Integration with external data APIs and services.
    - Support for data ingestion methods (e.g., RESTful APIs, web scraping).

- Data synchronization mechanisms to keep external data up-to-date.

4.5.5 Authentication and Authorization:

- **Purpose:** Ensures secure user access and data protection.
- **Requirements:**
  - Implementation of secure authentication protocols (e.g., OAuth, JWT).
  - Role-based access control to restrict user actions and data access.
  - Encryption of sensitive data during transmission.

4.5.6 Reporting and Alerts:

- **Purpose:** Facilitates communication of incident reports and alerts to relevant stakeholders.
- **Requirements:**
  - Automated alert generation and delivery to specified recipients.
  - Support for various communication channels (e.g., email, SMS, in-app notifications).
  - Timely and accurate dissemination of critical information.

4.5.7 Data Transfer Rates:

- **Purpose:** Specifies the speed at which data is transferred between components.
- **Requirements:**
  - High-speed data transfer for real-time incident reporting and visualization.
  - Scalability to handle increased data transfer during disaster events.

4.5.8 Message Formatting:

- **Purpose:** Defines the structure and format of messages exchanged between system components.
- **Requirements:**
  - Consistent use of standardized message formats (e.g., JSON) for data integrity.
  - Support for plain text, HTML, and multimedia formats for different types of messages.

These communication interfaces and associated requirements are integral to Floodlight's functionality, ensuring that information flows seamlessly between users, the system, and external entities during disaster management scenarios.

5. Other Nonfunctional Requirements

5.1 Open Source and Open Data

Open source principles are integral to Floodlight's ethos, ensuring transparency, collaboration, and accessibility of its software.

5.1.1 Open Source Requirements:

5.1.1.1 Licensing:

- Floodlight shall utilize open source licenses approved by recognized authorities (e.g., GNU General Public License, Apache License).
- The licensing information for all components, including third-party libraries, must be well-documented and comply with open source principles.

5.1.1.2 Collaboration:

- Floodlight development shall actively encourage community involvement, including contributions, feedback, and peer review.
- The project shall maintain an open repository on platforms like GitHub to facilitate collaboration.

5.1.1.3 Transparency:

- Comprehensive documentation of the software's architecture, design, and source code shall be available to the public.
- Decision-making processes and project governance shall be transparent, including discussions, meetings, and decision logs.

5.1.1.4 Version Control:

- Floodlight shall employ a version control system (e.g., Git) to manage source code, promoting collaboration and facilitating code tracking.

5.1.1.5 Continuous Integration:

- Implement a continuous integration (CI) and continuous deployment (CD) pipeline to ensure code quality, automated testing, and efficient release cycles.

5.1.2 Open Data Requirements:

Open data is fundamental for Floodlight's functionality, ensuring access to vital information for users and stakeholders.

5.1.2.1 Data Accessibility:

- Floodlight shall make all collected and generated data publicly accessible, unless privacy or security concerns dictate otherwise.
- Data shall be published in common open formats (e.g., CSV, JSON) for maximum accessibility.

5.1.2.2 Data Licensing:

- Clearly define licensing terms for Floodlight's data, adhering to open data principles (e.g., Creative Commons licenses).

5.1.2.3 Data Quality:

- Establish data quality control processes to uphold data accuracy, reliability, and relevance.

5.1.2.4 Data Privacy and Security:

- Floodlight shall employ robust data privacy and security measures, safeguarding sensitive information while respecting open data principles.
- Personal data shall be anonymized or aggregated where necessary to protect individuals' privacy.

5.1.2.5 Data Integration:

- Enable seamless integration of Floodlight data with other open data platforms, APIs, and services to facilitate data sharing and interoperability.

5.1.2.6 Data Sharing Agreements:

- Develop data sharing agreements with pertinent government agencies, NGOs, and stakeholders to ensure smooth data exchange.

These detailed requirements serve as guidelines to ensure that Floodlight adheres to open source and open data principles while effectively addressing the unique challenges posed by disaster management and response. If you wish to add specific details or requirements to any of these points, please feel free to specify.

5.2 Data management

Effective data management is crucial for Floodlight's functionality, ensuring the security, integrity, and accessibility of the information it collects and processes.

5.2.1 Data Collection:

- **Data Sources:** Floodlight shall collect data from various sources, including crowdsourced reports, governmental agencies, NGOs, and sensors.
- **Data Capture:** Ensure data capture methods are reliable, efficient, and adaptable to various data types, including text, images, and location data.
- **Data Validation:** Implement validation mechanisms to ensure the accuracy and integrity of collected data.

5.2.2 Data Storage:

- **Scalability:** The data storage infrastructure shall be scalable to accommodate the increasing volume of data during crisis situations.
- **Data Partitioning:** Implement data partitioning strategies for efficient data retrieval and to prevent bottlenecks.

5.2.3 Data Security:

- **Access Control:** Implement robust access control mechanisms to restrict data access to authorized users only.
- **Data Encryption:** Sensitive data, including personal information, shall be encrypted both in transit and at rest.
- **Data Backup:** Regularly back up data to ensure its availability in case of system failures or data corruption.

5.2.4 Data Retention:

- **Data Lifecycle Management:** Define clear data retention policies, including the duration data is stored and when it should be archived or purged.
- **Archiving:** Implement archiving mechanisms for historical data to reduce storage costs while preserving accessibility.

5.2.5 Data Privacy:

- **Compliance:** Ensure compliance with relevant data privacy regulations and standards, such as GDPR or HIPAA.
- **User Consent:** Implement mechanisms for obtaining and managing user consent for data collection and usage.

5.2.6 Data Interoperability:

- **Standard Formats:** Data shall be stored and transmitted in standardized formats (e.g., JSON, XML) to enable interoperability with other systems and platforms.
- **APIs:** Develop APIs to facilitate data exchange with external systems and applications.

5.2.7 Data Quality Assurance:

- **Data Cleansing:** Implement data cleansing processes to detect and rectify data anomalies, inconsistencies, or duplicates.
- **Data Validation:** Regularly validate data against predefined criteria to ensure its accuracy and reliability.

5.2.8 Performance Optimization:

- **Data Indexing:** Implement indexing strategies to optimize data retrieval speed, particularly for search and filter operations.
- **Caching:** Utilize caching mechanisms to enhance data retrieval performance.

5.2.9 Data Governance:

- **Data Ownership:** Define clear data ownership roles and responsibilities within the organization.
- **Data Policies:** Establish data governance policies and procedures for data management, including data classification and handling guidelines.

5.2.10 Disaster Recovery:

- **Backup and Redundancy:** Maintain off-site backups and redundancy measures to ensure data recovery in case of catastrophic events.

These detailed data management requirements are essential to ensure that Floodlight effectively handles data collection, storage, security, and quality while adhering to legal and ethical considerations. Please feel free to specify any additional details or requirements as needed.

5.3 Analytics / Preparedness

Effective analytics and preparedness capabilities are critical for Floodlight to provide actionable insights and support disaster response efforts.

5.3.1 Data Analysis:

- **Real-time Analysis:** Floodlight shall support real-time data analysis to provide immediate insights during emergencies.
- **Historical Analysis:** Provide tools for historical data analysis to identify trends and patterns in past disaster events.

5.3.2 Visualization:

- **Interactive Dashboards:** Implement interactive and customizable dashboards for users to visualize data in real-time.

- **Geospatial Visualization:** Support geospatial visualization to display incidents, resources, and needs on maps.
- **Chart Types:** Offer a variety of chart types (e.g., bar, pie, line) for diverse data representation.

5.3.3 Alerts and Notifications:

- **Alert Generation:** Automatically generate alerts and notifications based on predefined criteria (e.g., threshold breaches, critical incidents).
- **Multichannel Delivery:** Send alerts via multiple communication channels, including SMS, email, and in-app notifications.

5.3.4 Reporting:

- **Standard Reports:** Provide standard reports for incident summaries, resource allocation, and disaster impact assessments.
- **Custom Reporting:** Allow users to generate custom reports tailored to their specific needs.

5.3.5 Performance Optimization:

- **Query Optimization:** Optimize database queries for efficient data retrieval during data analysis.
- **Scalability:** Ensure analytics modules are scalable to handle increased data loads during crises.

5.3.6 Machine Learning and AI:

- **Predictive Analytics:** Implement machine learning models for predictive analytics to anticipate disaster impacts and resource needs.
- **Natural Language Processing:** Utilize NLP techniques for sentiment analysis of crowdsourced data to gauge public sentiment and emotional needs.

5.3.7 Disaster Preparedness:

- **Resource Planning:** Provide tools for disaster preparedness agencies to plan resource allocation and mobilization in advance.
- **Simulation:** Support disaster simulation scenarios to evaluate readiness and response strategies.
- **Training:** Offer training modules for disaster response teams to familiarize them with Floodlight's capabilities.

5.3.8 Accessibility and Usability:

- **User Training:** Provide user training resources and documentation to ensure users can effectively utilize analytics and preparedness features.
- **Accessibility Standards:** Comply with accessibility standards (e.g., WCAG) to ensure the platform is usable by individuals with disabilities.

5.3.9 Performance Metrics:

- **Response Time:** Define response time metrics for analytics and preparedness features to ensure optimal system performance.
- **Resource Allocation Efficiency:** Measure the efficiency of resource allocation and utilization during disaster response.

These detailed requirements for analytics and preparedness cover aspects like data analysis, visualization, alerts, reporting, and performance optimization. They are essential for enabling Floodlight to provide actionable insights and support preparedness efforts effectively.

5.4 Performance Requirements

5.4.1 Response Time:

The system shall have a response time of no more than 3 seconds for users to submit incident reports during peak usage times.

5.4.2 Data Retrieval:

The system shall retrieve incident data for visualizations and analytics within 5 seconds for a dataset of up to 10,000 incidents.

5.4.3 Scalability:

The system shall be designed to scale horizontally to handle a 100% increase in concurrent users during disaster events without degrading performance.

5.4.4 Dashboard Loading:

The system shall load standard dashboards within 3 seconds of a user's request, even when multiple users are accessing them simultaneously.

5.4.5 Report Generation:

The system shall generate custom reports for users with specific data requirements within 10 seconds for datasets of up to 100,000 records.

5.4.6 Alert Dissemination:

The system shall disseminate critical alerts to all targeted recipients within 30 seconds of their generation.

These performance requirements are specific and address key aspects of Floodlight's functionality during both normal and peak usage conditions. They aim to provide a responsive and efficient platform for disaster management and response.

5.5 Safety Requirements

5.5.1 Data Accuracy:

The system shall ensure the accuracy of incident data reported. Data accuracy is essential to prevent misinformation that could lead to incorrect emergency responses.

5.5.2 User Authentication:

The system shall implement robust user authentication to prevent unauthorized access. Unauthorized access could lead to misuse or malicious actions within the platform.

5.5.3 Data Security:

The system shall employ encryption and secure data transmission protocols to protect sensitive data, such as personally identifiable information (PII) and sensitive reports.

5.5.4 Disaster Recovery:

The system shall have a disaster recovery plan in place to ensure data integrity and availability in the event of system failures or disasters.

5.5.5 Regular Backups:

The system shall perform regular backups of critical data to prevent data loss in case of system failures.

5.5.6 Accessibility:

The system shall adhere to accessibility standards to ensure that users with disabilities can access and use the platform during emergencies.

5.5.7 Training and User Education:

The system shall provide training materials and resources to educate users on the correct and safe use of the platform.

5.5.8 Compliance with Regulatory Standards:

The system shall comply with all relevant data protection and privacy regulations to protect user data and privacy.

5.5.9 Incident Response Protocol:

The system shall have a well-defined incident response protocol to handle security incidents promptly and effectively.

These safety requirements are designed to mitigate risks associated with Floodlight's usage and to ensure that the system can be relied upon during critical emergency situations.

5.6 Security Requirements

5.6.1 User Authentication:

The system shall implement strong user authentication mechanisms, including multi-factor authentication (MFA), to verify the identity of users.

5.6.2 Access Control:

The system shall enforce role-based access control (RBAC) to ensure that users only have access to data and features relevant to their roles.

5.6.3 Data Encryption:

The system shall encrypt data both in transit and at rest to protect sensitive information from unauthorized access.

5.6.4 Audit Trails:

The system shall maintain detailed audit trails of user actions and system activities to facilitate monitoring and forensic analysis.

5.6.5 Security Patching:

The system shall receive regular security updates and patches to address vulnerabilities promptly.

5.6.6 Data Privacy:

The system shall comply with all relevant data protection regulations and guidelines, including GDPR, HIPAA, or other applicable standards.

5.6.7 Incident Response Plan:

The system shall have a well-defined incident response plan in place to address security incidents, including data breaches or cyberattacks.

5.6.8 Security Training:

The system shall provide security training to users to educate them about best practices and potential security risks.

5.6.9 Third-party Integrations:

The system shall ensure that third-party integrations, if used, comply with security and privacy standards.

5.6.10 Secure Development Practices:

The system's development process shall adhere to secure coding practices to minimize vulnerabilities.

These security requirements are designed to protect the integrity, confidentiality, and availability of data within Floodlight and to ensure that user information is safeguarded from unauthorized access or malicious activities.

5.7 Software Quality Attributes

5.7.1 Reliability:

The system shall operate with a system uptime of at least 99.9% to ensure that it is available when needed during disaster events.

5.7.2 Availability:

The system shall be available 24/7/365, with minimal planned downtime for maintenance.

5.7.3 Performance:

The system shall respond to user requests within 2 seconds on average, even during peak usage.

5.7.4 Usability:

The system shall adhere to established user experience (UX) and user interface (UI) design standards to ensure ease of use and a minimal learning curve for users.

5.7.5 Scalability:

The system shall be scalable to handle a growing number of users and data as adoption increases.

5.7.6 Interoperability:

The system shall support standard data formats and APIs to facilitate integration with external systems, including government agencies and NGOs.

5.7.7 Maintainability:

The system shall be designed with modularity and code maintainability best practices to simplify future updates and enhancements.

5.7.8 Testability:

The system shall include a comprehensive test suite to validate its functionality and security.

5.7.9 Robustness:

- The system shall gracefully handle unexpected errors or inputs to prevent service disruptions.

5.7.10 Flexibility:

- The system shall allow for the configuration of disaster-specific parameters and data sources to adapt to changing disaster scenarios.

These quality attributes collectively contribute to the overall effectiveness, reliability, and usability of the Floodlight system, making it a valuable tool for disaster management.

6. Architecture Constraints

6.1 Open Source Dependency:

The system shall be built using open-source technologies and libraries to align with the project's commitment to open-source principles.

6.2 Platform Compatibility:

The system shall be compatible with common web browsers (e.g., Chrome, Firefox, Safari) to ensure accessibility for a wide range of users.

6.3 Data Privacy Regulations:

The system shall adhere to data privacy regulations, including but not limited to GDPR, HIPAA, and local data protection laws, to protect user data.

6.4 Third-Party Integrations:

The system shall integrate with external platforms such as Ushahidi and Sahana to leverage existing disaster management tools and data sources.

6.5 Cloud Infrastructure:

The system shall be hosted on a cloud infrastructure (e.g., AWS, Azure) to ensure scalability, availability, and disaster recovery capabilities.

6.6 Internet Connectivity:

The system shall assume that users have intermittent or limited internet connectivity during disaster events and must be designed to accommodate offline data collection and synchronization.

6.7 Multilingual Support:

The system shall support multiple languages to accommodate users from diverse linguistic backgrounds.

6.8 Geographic Adaptability:

The system shall be adaptable to different geographic regions and disaster scenarios, including floods, earthquakes, and epidemics.

7. Risks and Technical Debts

7.1 Data Accuracy Risk:

- **Risk:** There is a risk of inaccurate or incomplete data being reported by crowdsourced users during a disaster event, potentially leading to incorrect decisions and responses.
- **Mitigation:** Implement data validation algorithms and user feedback mechanisms to improve data quality over time.

7.2 Scalability Challenge:

- **Risk:** As the user base and data volume increase, there is a risk that the system may not scale effectively, leading to performance issues during peak usage.
- **Mitigation:** Continuously monitor system performance, implement load balancing, and optimize database queries to ensure scalability.

7.3 Security Vulnerabilities:

- **Risk:** Security vulnerabilities, such as data breaches or unauthorized access, may compromise sensitive user information or disrupt system functionality.
- **Mitigation:** Regularly conduct security audits, employ encryption mechanisms, and stay updated with security patches to protect against vulnerabilities.

7.4 User Adoption Challenge:

- **Risk:** Users may not fully adopt the system, resulting in underutilization and reduced effectiveness during disaster events.
- **Mitigation:** Conduct user training, engage in community outreach, and gather feedback to improve the system's usability and relevance.

7.5 Technical Debt - Code Complexity:

- **Debt:** The codebase may accumulate complexity over time, making it challenging to maintain and extend the system.
- **Mitigation:** Enforce coding standards, conduct regular code reviews, and refactor complex code sections to reduce technical debt.

7.6 Technical Debt - Outdated Dependencies:

- **Debt:** Dependencies on third-party libraries or frameworks may become outdated, posing security and compatibility risks.
- **Mitigation:** Implement dependency management processes to regularly update and replace outdated components.

7.7 Technical Debt - Documentation Gap:

- **Debt:** Inadequate documentation can lead to difficulties for developers and users in understanding and using the system effectively.
- **Mitigation:** Maintain comprehensive documentation, including user guides, developer manuals, and system architecture documentation.

7.8 Disaster Event Variability:

- **Risk:** The system may not fully address the unique aspects and variability of different disaster events, impacting its effectiveness.
- **Mitigation:** Collaborate with disaster management experts to ensure the system's adaptability to various disaster scenarios.

7.9 Legal and Compliance Risks:

- **Risk:** Changes in data privacy regulations or legal requirements may necessitate updates to the system, potentially leading to legal and compliance risks.
- **Mitigation:** Stay informed about relevant legal changes and adapt the system accordingly to maintain compliance.

8. Other Requirements

8.1 Internationalization and Localization:

- **Requirement:** The Floodlight system must support internationalization (i18n) and localization (l10n) to ensure accessibility to users from diverse linguistic and cultural backgrounds.
- **Rationale:** Internationalization and localization are essential to make the system user-friendly and accessible to a global audience during disaster events.

8.2 Database Requirements:

- **Requirement:** The system must employ a robust and scalable database system capable of efficiently storing and retrieving large volumes of data generated during disaster events.
- **Rationale:** A reliable and scalable database is critical for handling the influx of data during disaster events and ensuring data integrity.

8.3 Legal and Compliance Requirements:

- **Requirement:** The Floodlight system must adhere to all relevant legal and compliance requirements, including data privacy regulations and intellectual property rights.
- **Rationale:** Compliance with legal and regulatory standards is essential to protect user data and ensure the system's ethical operation.

8.4 Reusability Objectives:

- **Requirement:** The project should aim to create modular and reusable components, facilitating their integration into other disaster management systems or extensions of Floodlight.
- **Rationale:** Promoting reusability enhances the value of the project by allowing others to leverage its components for similar applications.

8.5 Disaster Preparedness Training:

- **Requirement:** The system should include a training module to educate disaster management personnel and volunteers on how to effectively use the Floodlight toolkit.
- **Rationale:** Proper training ensures that users can make the most of the system's capabilities during disaster events.

8.6 Continuous Improvement and Feedback Mechanism:

- **Requirement:** Implement a mechanism for collecting user feedback and suggestions for system improvement, which will be used to enhance the system in subsequent iterations.
- **Rationale:** User feedback is invaluable for identifying areas of improvement and tailoring the system to better meet user needs.

8.7 Accessibility Standards:

- **Requirement:** The system should adhere to established accessibility standards (e.g., WCAG) to ensure that individuals with disabilities can use the platform effectively.
- **Rationale:** Accessibility is a fundamental requirement to make the system inclusive and accessible to all, including those with disabilities.

8.8 Disaster Simulation and Testing:

- **Requirement:** Conduct periodic disaster simulation exercises and testing to validate the system's readiness and effectiveness in real-world disaster scenarios.
- **Rationale:** Regular testing and simulations help identify and rectify weaknesses in the system's disaster response capabilities.

8.9 Documentation Standards:

- **Requirement:** All system documentation, including user manuals, technical guides, and architecture documentation, should adhere to standardized templates and formats for clarity and consistency.

- **Rationale:** Standardized documentation ensures that information is presented in an organized and comprehensible manner.

8.10 User Support and Helpdesk:

- **Requirement:** Establish a user support and helpdesk system to assist users in case of technical issues, questions, or emergencies.
- **Rationale:** Providing user support enhances user experience and ensures that users can effectively utilize the system during critical situations.

8.11 Disaster Response Best Practices:

- **Requirement:** The system should incorporate recognized disaster response best practices and guidelines to inform users and guide their actions during disaster events.
- **Rationale:** Incorporating best practices enhances the system's effectiveness in disaster response.

8.12 Disaster-Specific Modules:

- **Requirement:** Develop disaster-specific modules within the system that can be activated or customized based on the type and nature of the disaster event (e.g., flood, earthquake, pandemic).
- **Rationale:** Tailoring the system to specific disaster types improves its relevance and utility during different scenarios.

8.13 Reporting and Accountability:

- **Requirement:** Implement reporting and accountability features that allow users and administrators to track and review actions, decisions, and responses during disaster events.
- **Rationale:** Reporting and accountability mechanisms enhance transparency and facilitate post-event analysis and improvement.

8.14 Disaster Recovery Plan:

- **Requirement:** Develop and maintain a disaster recovery plan for the Floodlight system to ensure data backup, system restoration, and continuity of operations in case of system failures or disasters affecting the platform itself.
- **Rationale:** A disaster recovery plan is essential to minimize downtime and data loss during critical events.

8.15 Ethical Use Guidelines:

- **Requirement:** Establish ethical use guidelines for the system, emphasizing responsible and ethical behavior in reporting, data collection, and system usage.
- **Rationale:** Ethical guidelines promote responsible use of the system and ethical conduct during disaster events.

8.16 Integration with External Systems:

- **Requirement:** Ensure that the Floodlight system is designed to seamlessly integrate with external disaster management systems, databases, and platforms for data sharing and collaboration.
- **Rationale:** Integration with external systems enhances the system's interoperability and extends its capabilities.

8.17 Disaster Communication Standards:

- **Requirement:** Adhere to recognized disaster communication standards and protocols to ensure that messages and alerts are consistent with established practices.
- **Rationale:** Following communication standards promotes clarity and consistency in disaster messaging.

8.18 Disaster Response Research Collaboration:

- **Requirement:** Collaborate with disaster response research organizations to share data, findings, and insights to contribute to the improvement of disaster management practices.
- **Rationale:** Research collaboration enhances the system's knowledge base and supports advancements in disaster response.

8.19 Data Retention and Privacy:

- **Requirement:** Define data retention policies that balance the need for historical data analysis with user privacy considerations, ensuring compliance with relevant data protection laws.
- **Rationale:** Data retention policies safeguard user privacy while allowing for valuable data analysis.

8.20 Integration with Public Warning Systems:

- **Requirement:** Integrate with public warning systems to disseminate alerts and critical information to the general public during disaster events.

- **Rationale:** Integration with public warning systems extends the reach of disaster alerts and ensures broader public awareness.

8.21 Public Awareness Campaigns:

- **Requirement:** Launch public awareness campaigns to educate communities about the Floodlight system's capabilities and encourage their participation in disaster reporting and response.
- **Rationale:** Public awareness campaigns promote community engagement and active participation in disaster response efforts.

8.22 Inclusivity and Diversity Considerations:

- **Requirement:** Consider inclusivity and diversity in all aspects of system development, ensuring that the system caters to the needs of diverse user groups and communities.
- **Rationale:** Inclusivity and diversity considerations promote equity in disaster response.

8.23 Disaster Event Data Archive:

- **Requirement:** Maintain an archive of historical disaster event data, which can be used for research, analysis, and improving disaster response strategies.
- **Rationale:** An archive of historical data supports ongoing learning and refinement of disaster response approaches.

8.24 Feedback Utilization:

- **Requirement:** Actively utilize feedback from users and stakeholders to inform system enhancements and updates, demonstrating a commitment to continuous improvement.
- **Rationale:** Feedback utilization ensures that the system remains responsive to user needs and evolving disaster scenarios.

8.25 Collaboration with Educational Institutions:

- **Requirement:** Collaborate with educational institutions to engage students and researchers in projects related to disaster management, data analysis, and system improvement.
- **Rationale:** Collaboration with educational institutions fosters innovation and knowledge sharing in disaster management.

8.26 Disaster Risk Reduction Integration:

- **Requirement:** Integrate disaster risk reduction principles into the system to promote proactive measures and preparedness efforts in addition to response capabilities.
- **Rationale:** Disaster risk reduction contributes to overall disaster resilience.

8.27 Transparency and Accountability Reports:

- **Requirement:** Generate and publish transparency and accountability reports detailing system usage, data handling, and disaster response outcomes.
- **Rationale:** Transparency reports enhance trust and accountability in system operations.

8.28 Continuous Evaluation and Improvement:

- **Requirement:** Implement mechanisms for continuous evaluation, assessment, and improvement of the Floodlight system's performance, effectiveness, and user satisfaction.
- **Rationale:** Continuous evaluation and improvement are essential for the system's ongoing relevance and success.

8.29 Public Access to Data:

- **Requirement:** Ensure that non-sensitive, non-personal data collected by the system is made publicly accessible for research and analysis purposes, contributing to disaster management knowledge.
- **Rationale:** Public access to data supports research and enhances the broader understanding of disaster events.

8.30 Integration with Early Warning Systems:

- **Requirement:** Integrate with early warning systems to enable real-time data sharing and alert dissemination for timely disaster response.
- **Rationale:** Integration with early warning systems enhances the system's ability to respond rapidly to emerging disaster situations.

8.31 Cultural Sensitivity Training:

- **Requirement:** Provide cultural sensitivity training to system administrators and support personnel to ensure respectful and culturally appropriate interactions with diverse user communities.
- **Rationale:** Cultural sensitivity training promotes respectful engagement during disaster events.

8.32 User Privacy Education:

- **Requirement:** Develop educational materials and resources to inform users about their privacy rights, data usage, and privacy protection measures within the system.
- **Rationale:** User privacy education enhances user trust and confidence in the system.

8.33 Integration with Mobile Alert Systems:

- **Requirement:** Integrate with mobile alert systems and apps to deliver disaster alerts, updates, and information directly to mobile devices.
- **Rationale:** Mobile integration enhances the system's reach and accessibility during disaster events.

8.34 Disaster Event Knowledge Repository:

- **Requirement:** Establish a knowledge repository within the system, aggregating best practices, lessons learned, and successful disaster response strategies from past events.
- **Rationale:** A knowledge repository supports informed decision-making during disaster events.

8.35 Data Verification and Validation:

- **Requirement:** Implement data verification and validation mechanisms to ensure the accuracy and reliability of user-contributed data.
- **Rationale:** Data verification and validation enhance the credibility of the information provided by users.

These requirements are essential to address various aspects of the Floodlight system's functionality, usability, ethics, and integration with external systems and stakeholders. They contribute to the system's effectiveness, inclusivity, and ability to support disaster management efforts comprehensively.

Appendix A: Glossary

- **Floodlight:** The disaster management toolkit designed to empower communities in tracking and responding to floods in Pakistan.
- **Ushahidi:** An open-source platform for crowdsourcing and visualizing crisis information.
- **NGOs:** Non-Governmental Organizations involved in humanitarian and disaster response efforts.
- **API:** Application Programming Interface, a set of rules and protocols for building and interacting with software applications.

- **GUI:** Graphical User Interface, the visual interface through which users interact with software.
- **KPI:** Key Performance Indicator, a measurable value that indicates the effectiveness of a system in achieving its objectives.
- **HTTP:** Hypertext Transfer Protocol, the foundation of data communication on the World Wide Web.
- **GIS:** Geographic Information System, a system designed to capture, store, manipulate, analyze, manage, and present spatial or geographic data.
- **SMS:** Short Message Service, a text messaging service component of most telephone, Internet, and mobile device systems.
- **API:** Application Programming Interface, a set of rules and protocols for building and interacting with software applications.
- **IoT:** Internet of Things, the network of physical objects ("things") embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet.
- **GDPR:** General Data Protection Regulation, a regulation in EU law on data protection and privacy.
- **IT:** Information Technology, the use of computers, software, and networks to store, process, and transmit data.

Appendix B: Analysis Models

- **Use Case Diagram:** This diagram illustrates the various interactions between users and the Floodlight system. It identifies the different use cases, actors, and their relationships.
- **Data Flow Diagram (DFD):** A DFD can be used to depict the flow of data within the Floodlight system. It can show how data is input, processed, and output, including interactions with external systems.
- **Entity-Relationship Diagram (ERD):** An ERD can be used to model the database schema of Floodlight, showing the relationships between different data entities.
- **Sequence Diagrams:** These diagrams can illustrate the sequence of interactions between objects or components in the system, showing how specific use cases are executed step by step.
- **State Transition Diagrams:** If Floodlight has components or processes with different states, these diagrams can be used to model the transitions between states and the events triggering those transitions.
- **Class Diagrams:** Class diagrams can depict the various classes and their relationships within the Floodlight system, providing an overview of the system's object-oriented structure.
- **Activity Diagrams:** These diagrams can be used to model the flow of activities or processes within the system, showing how different activities are performed in response to events.

Appendix C: Issues List

- **TBD: Performance Metrics:** The specific performance metrics, such as response times and data processing rates, need to be defined for each functional requirement. (Priority: High)
- **Conflict: User Authentication:** There's a conflict between two requirements regarding user authentication. One suggests using two-factor authentication, while another recommends a single sign-on approach. A decision is needed. (Priority: Medium)
- **Information Needed: Regulatory Compliance:** Information is needed on specific regulatory compliance requirements that Floodlight must adhere to, especially regarding data privacy and security. (Priority: High)
- **TBD: Disaster Classification:** The criteria and classification for different types of disasters are yet to be defined, affecting the design of the system. (Priority: Medium)
- **Pending Decision: Open Data Licensing:** A decision is pending regarding the licensing model for open data used in Floodlight. (Priority: High)
- **Conflict: User Interface Design:** There is a conflict between two teams regarding the design of the user interface. The conflict needs to be resolved. (Priority: High)
- **Information Needed:** Hardware Compatibility: Information is needed regarding the specific hardware components that Floodlight must be compatible with. (Priority: Medium)
- **TBD: Disaster Reporting Formats:** The formats for reporting different types of disasters are yet to be determined. (Priority: Medium)
- **Conflict: Third-party Integration:** There's a conflict between requirements for integrating with two different third-party platforms. A decision is needed. (Priority: High)
- **Pending Decision:** Data Retention Policy: A decision is pending regarding the data retention policy, including how long data will be stored and when it should be archived or deleted. (Priority: High)