Constituency Connect

By

Muhammad Ahsan

21-ARID-737

Haseeb Ur Rehman

21-ARID-715

Huzaifa Bin Shahzad

21-ARID-719

Supervisor

Ms. Farkhanda Qamar

University Institute of Information Technology,

PMAS - Arid Agriculture University,

Rawalpindi, Pakistan

2025

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

This chapter introduces "Constituency Connect," a project designed to modernize and enhance communication between constituents and their elected representatives. Constituency Connect aims to address the persistent gap in citizen-representative communication with a focus on fostering transparent, efficient, and real-time interactions. This platform empowers constituents to voice concerns and hold representatives accountable by streamlining complaint submission, tracking, and resolution. Simultaneously, it provides representatives with valuable tools to engage meaningfully with their communities. This project combines civic engagement with technological innovation, strengthening democratic practices through user-centric features, such as virtual meetups, real-time notifications, and role-specific functions. Constituency Connect aspires to set a new standard in civic communication by making governance more responsive, inclusive, and transparent.

1.1. Brief Overview

The Constituency Connect project is designed to bridge critical communication gaps between constituents and their elected representatives. In many communities, citizens struggle to engage with representatives meaningfully, often facing limited channels for real-time interaction, delayed responses, and a lack of transparency in complaint tracking and resolution. These communication inefficiencies reduce civic engagement, decrease representative accountability, and contribute to a disconnect between representatives and the people they serve.

Recognizing the importance of timely and transparent communication, Constituency Connect leverages modern technology to provide a unified platform that enables complaint reporting, real-time status updates, virtual meetups, and performance tracking. The platform empowers citizens to voice concerns effectively and fosters accountability by providing representatives with tools to track, prioritize, and address issues systematically. By integrating role-specific access levels, automated notifications, and data visualizations, Constituency Connect enhances the quality of governance and civic engagement. It also includes multi-language support to ensure accessibility across diverse populations.

The project adopts a Kanban methodology to efficiently manage workflow, prioritize tasks, and ensure continuous progress with minimal team resources, ensuring that the development process remains flexible and responsive to evolving user needs. By continuously iterating feedback, Constituency Connect aims to build a responsive, accessible, and impactful platform that strengthens democratic communication and enhances public service effectiveness.

1.2. Relevance to Course Modules

The following are the relevant course modules.

- **Web Development**:

  This course provided the foundational skills necessary to develop the web-based system for Constituency Connect, focusing on front-end and back-end development aspects.

- **Object-Oriented Analysis and Design**:

  Through this course, we learned how to model and structure the system using use cases, activity diagrams, and sequence diagrams, which are crucial for developing efficient and organized architecture.

- **Database Management**:

  This course was instrumental in designing and implementing the database for the project, ensuring efficient data handling, storage, and retrieval.

- **Human-Computer Interaction**:

  This module informed us of our approach to creating a user-friendly interface, essential for promoting accessibility and ease of use for diverse users.

- **Software Project Management**:

  Key concepts from this course helped us plan, manage tasks, and use Kanban to streamline workflow and track project progress effectively with a small team.

1.3. Project Background

The "Constituency Connect" project addresses a critical need for improved communication and accountability between elected representatives and their constituents. Traditional channels of communication often lack transparency, timely feedback, and the ability to track the progress of complaints or requests. Constituency Connect aims to address these challenges by providing a digital platform that facilitates structured communication, complaint management, and real-time interaction.

The platform enables constituents to report issues, monitor their progress, and engage in virtual meetings with representatives, fostering greater civic participation and accountability. Constituents can also access public event calendars, participate in surveys, and view representative performance metrics. By utilizing modern web technologies and a streamlined user experience, the system enhances both accessibility and transparency.

Additionally, the project leverages role-based access control to ensure that representatives, constituents, and department officials can interact within their specific functional roles, promoting data security and efficient workflow management. This initiative highlights the importance of digital engagement in governance, presenting a scalable model for strengthening democratic processes and bridging the gap between government officials and the public.

1.4. Literature Review

Many democratic societies face a significant communication barrier between elected officials and their constituents. Direct interaction with representatives is often limited to individuals with special connections or high social status. In contrast, the average citizen's interaction is typically limited to brief encounters during election seasons or at local events. These irregular interactions rarely address the ongoing needs and concerns of the community. This lack of

meaningful engagement has led to a sense of disconnection among citizens who feel their voices are unheard in the democratic process. As a result, community issues frequently go unaddressed, and public trust in government institutions continues to diminish.

By addressing these existing problems, Constituency Connect has the potential to significantly transform the relationship between citizens and their government, leading to more responsive, efficient, and democratic governance.

- **FixMyStreet** is a platform designed to help citizens report local issues, such as potholes, broken streetlights, and other community problems, directly to their local government. This user-friendly tool allows residents to track the progress of their reports, ensuring that public concerns are addressed efficiently and effectively.
- **SeeClickFix** is a community-driven platform that enables residents to report non-emergency issues in their neighborhood, such as damaged infrastructure, public safety concerns, and other local problems. This interactive approach encourages active community engagement and helps to build a stronger, more connected neighborhood.
- **Neighbour-land** is a civic engagement platform that empowers residents to collaborate with their local government and community organizations on neighborhood improvement projects. It focuses on gathering ideas, feedback, and support from the community to shape public spaces and policies. By facilitating open dialogue and collective action, it helps to create more vibrant, inclusive, and responsive communities.
- **PM Citizen Portal** is a government platform started by the Prime Minister's Office in 2018 to connect citizens with the government more effectively. It works as a central system for handling complaints, allowing people to register issues, give suggestions, and find information about various public services. You can access the portal through a mobile app or a website, making it easy to file complaints against any government department or service provider. Once you submit a complaint, the system sends it to the right authorities for resolution and keeps you updated on its status in real time. The Pakistan Citizen Portal is a key part of the government's effort to improve transparency, accountability, and the quality of public services. By giving citizens, a direct way to voice their concerns and engage with the government, the portal empowers people and helps create a more responsive and inclusive system.

1.5. Analysis from Literature Review

Our analysis reveals key alignments and distinctions between Constituency Connect and other platforms that bridge communication gaps between government representatives and citizens. Constituency Connect complements and extends the functionality of existing platforms by integrating various civic engagement tools, promoting a continuous, interactive, and transparent relationship between representatives and constituents. This approach contributes to more responsive and democratic local government.

Features	**Constituency Connect**	**FixMyStreet**	**SeeClickFix**	**Neighbourland**
Origin	Pakistan	United Kingdom	United States	United States
Profile Management	Yes, detailed profiles with editable information	Limited to basic profile management	Yes, basic profile management	Yes, basic profile management
Complaint Reporting	Yes, including anonymous reporting and detailed tracking	Yes, report local problems to the council	Yes, allows reporting issues to local authorities	No, focuses on community engagement
Complaint Tracking	Yes, with full history and status updates	Limited tracking, focus on reporting	Yes, with status updates	No, Does not focus on tracking
Complaint Prioritization	Yes	No	No	No
Automated Complaint Routing	Yes	Yes	Yes	No

Virtual Meetups	Yes	No	No	Yes
Surveys and Polls	Yes, Representatives can create a survey	No	No	Yes
Event Calendar	Yes, Representatives can post upcoming events and public meetings	No	No	No
Security	Standard	Standard	Standard	Standard
Push Notifications	Yes	No	Yes	No
Multi-Language Support	Yes	No	No	No
Feedback	Yes, Feedback about problem resolution	Yes	Yes	Yes
Performance Metrics	Yes, tracks and displays the performance metrics of representatives	No	No	No

Table 1.1: Literature Review

Features	Constituency Connect	PM Citizen Portal
Origin	Pakistan	Pakistan
User Roles	Constituents, Representatives, Assistants, Complaint Handlers	Citizens, Government Officials
Profile Management	Yes, detailed profiles with editable information	Yes, detailed profiles with editable information
Complaint Reporting	Yes, including anonymous reporting and detailed tracking	Yes, with broad categorization
Complaint Tracking	Yes, with full history and status updates	Status updates are available through the portal
Complaint Prioritization	Yes	No
Automated Complaint Routing	Yes	Forwarded to concerned government departments
Virtual Meetups	Yes	Not available
Surveys and Polls	Yes, Representatives can create a survey	Not available
Event Calendar	Yes, Representatives can post upcoming events and public meetings	No
Security	Standard	Standard
Push Notifications	Yes	Yes

Multi-Language Support	Yes	Yes
Feedback	Yes, Feedback about problem resolution	Limited
Performance Metrics	Yes, tracks and displays the performance metrics of representatives	No

Table 1.2: Literature Review

1.6. Methodology and Software Lifecycle for This Project

For the Constituency Connect project, we have chosen the Kanban methodology. This approach suits our small team due to its flexibility, simplicity, and the emphasis it places on visualizing the work process. Kanban will enable us to manage tasks more effectively by ensuring continuous flow and minimizing bottlenecks, which is ideal for our team. Kanban's focus on visualizing tasks, setting priorities, and delivering features as they are completed will provide constant insight into our progress, allowing for better decision-making.

1. **Requirements Elicitation**

   The project will begin with an in-depth understanding of the system's needs and requirements. The team will carefully gather input from stakeholders to make sure the features we develop match user needs perfectly.

2. **Design**

   The design phase will focus on structuring the system architecture and database schema. We will also define the UI/UX components to ensure users have a seamless experience when engaging with the platform.

3. **Construction / Iteration**

   We will proceed with incremental development, breaking down tasks into smaller pieces of work. This will involve building and testing features iteratively, ensuring each function is completed before moving on to the next.

4. **Testing**

   Testing is integrated into each stage of development. Continuous testing will be done

as part of each iteration to ensure that features work correctly and meet quality standards. Automated testing tools will be used for quick feedback and bug resolution.

5. **Deployment**

   Deployment will occur as new features are completed and validated, following successful iterations. Once the features are validated, they will be deployed to the production environment.

6. **Feedback**

   Feedback from the stakeholders will be collected continuously, helping to refine the system and prioritize the next set of features for development. This feedback loop ensures that the product meets evolving user needs.

Key Roles:

Following are the key roles to develop our system.

- **Project Lead:**

  Acts as the Kanban "service delivery manager," prioritizing tasks based on project goals and managing workflow efficiency. This role ensures the project stays aligned with high-level objectives and communicates any adjustments to meet stakeholder needs.

- **Development Lead:**

  Oversees the technical aspects, functioning similarly to a "flow manager." They help address bottlenecks, guide the technical direction, and facilitate collaboration across tasks.

- **Developers:**

  Each developer focuses on their assigned tasks within the Kanban board. With clearly defined tasks in each column (e.g., To-Do, In-Progress, Testing, Done), developers work independently, pulling tasks as they progress.

1.6.1. Rationale behind Selected Methodology

The Kanban methodology was chosen for the following reasons:

1. **Visualizing Work:**

   Kanban provides a clear and visual representation of work items, allowing the team to prioritize tasks, track progress, and identify bottlenecks in real-time.

2. **Flexibility and Adaptability:**

   Kanban's flexibility allows us to adapt to changes and adjust our work priorities quickly, which is essential given the changing needs of constituents and representatives.

3. **Efficient Resource Management:**

   With our small team, Kanban helps optimize resources by enabling us to focus on manageable tasks, reducing the risk of overload and increasing efficiency.

4. **Continuous Delivery:**

   The Kanban approach supports the continuous delivery of incremental features, ensuring that the most valuable functionality is available to users as soon as it is ready.

5. **Minimizing Waste:**

   Kanban emphasizes efficiency, minimizing delays between stages, and ensuring the team can focus on high-priority tasks, reducing wasted time.

6. **Improved Collaboration:**

   Kanban encourages frequent communication between team members and stakeholders, improving collaboration and the overall quality of the project.

Chapter 2: Problem Definition

This chapter outlines the key challenges that the Constituency Connect project aims to address in improving communication and accountability between constituents and their elected representatives. Traditionally, communication is confined to election periods, which often fail to address ongoing, everyday issues of the public effectively and lack consistent follow-up.

Such limitations result in a disconnection between citizens and representatives, leading to under-addressed issues, and decreasing public trust in governmental institutions. This project intends to address these issues by creating a structured digital platform where constituents can report concerns, track updates, and directly engage with representatives, creating a continuous feedback loop. Additionally, representatives benefit from insights into public sentiment and an organized method to track and respond to issues effectively.

The goal is to develop a centralized system that facilitates transparent, accountable communication and provides actionable insights to both representatives and constituents. By identifying these gaps in civic engagement and public service accountability, this chapter lays the groundwork for understanding the project's potential to enhance civic participation and governmental responsiveness. This chapter explains how Constituency Connect plans to make communication more open, and easy by providing a web-based platform focused on the needs of both citizens and representatives.

2.1. Problem Statement

Although various complaint management systems exist, significant gaps remain in effectively connecting constituents with their representatives. Current communication channels are often inefficient and unreliable and lack real-time interaction, causing delays in complaint resolution and weakening civic engagement. This results in slow problem resolution, reduced accountability, and a growing disconnect between representatives and the public. Citizens struggle to file complaints, track progress, or receive timely responses, highlighting the need for a unified solution that improves communication, transparency, and governance.

2.2. Deliverable and Development Requirements

For the *Constituency Connect* project, defining clear deliverables and development requirements is essential for effective project management and achieving the intended impact.

Deliverables represent tangible outputs, while development requirements ensure that each component meets necessary technical and functional standards.

- **Deliverables:**

In *Constituency Connect*, the deliverables represent the measurable outputs or milestones to be achieved throughout the project. Key deliverables include:

1. **User-Centered Web Platform**:
   A fully functional web application with a seamless, responsive interface designed for constituents to submit issues, track progress, and receive updates from representatives, optimizing the user experience across different devices.
2. **Role-Based Access and Permissions System**:
   A secure role-management system to allow customized access levels for Constituents, Representatives, and Complaint Handlers, ensuring data privacy and appropriate permissions for each user.
3. **Complaint Management System**:
   A streamlined system that enables representatives to categorize, prioritize, and address issues efficiently, ensuring timely responses to constituent concerns and transparent tracking of issue status.
4. **Automated Notifications and Alerts**:
   Real-time notification system to inform users of updates or status changes on their submissions, enabling constituents and representatives to stay engaged with issue progress.
5. **Data Analytics Dashboard**:
   An integrated dashboard providing representatives with insights on frequently reported issues, resolution times, and response trends, supporting data-driven decision-making and accountability.
6. **Detailed Documentation:**
   Comprehensive documentation, including technical specifications, and setup instructions for end-users to maximize usability and functionality.

7. **Performance Reports**:

   Reports that summarize complaint management efficiency and other metrics, demonstrate the platform's impact on civic engagement.

These deliverables ensure Constituency Connect provides a secure, user-friendly, and effective platform that fosters transparent communication and accountability between constituents and their representatives.

- ## Development Requirements:

The development requirements define the technical aspects necessary to achieve the desired functionalities of *Constituency Connect*. The following requirements will ensure the platform's successful implementation and alignment with project goals:

1. **User-Friendly Interface**:

   Design an intuitive, accessible interface to facilitate easy navigation and usage for all users, including multi-language support to accommodate diverse communities.
2. **Role-Based Access Control**:

   Set up access control protocols to define permissions for different user roles (e.g., constituents, representatives, departmental officials), ensuring data integrity and efficient workflows.
3. **Continuous Improvement**:

   Incorporate tools and frameworks for regular monitoring, testing, and improvement of functionalities to maintain the high performance and reliability of the system.
4. **Ethical and Compliance Standards**:

   Adhere to ethical standards for fair data usage, transparency in communication, and respect for citizens' rights, aligning with democratic and legal principles.
5. **Scalability and Reliability:**

   Ensure the platform can handle increasing user traffic and data without performance degradation.
6. **Data Privacy and Security:**

   Implement robust security measures to protect sensitive user data, particularly considering the sensitive nature of constituent-representative communication.

7. **Efficient Data Management:**

   Implement efficient data handling and storage techniques to allow easy retrieval, filtering, and sorting of user-submitted issues and reports.

These requirements will guide the development process to meet high standards of usability, functionality, and security, ensuring that Constituency Connect fulfills its objective of improving democratic engagement and public service accountability.

Chapter 3: Requirement Analysis

This chapter delves into the requirements for the project, "Constituency Connect." The Software Requirements Specification (SRS) document is central to this analysis, comprehensively outlining the system's essential functionalities, user roles, and technical requirements for successful implementation. The SRS defines the project's scope, aligning user needs with system features and providing clarity on the platform’s purpose, performance goals, and expected outcomes.

The chapter underscores the importance of thorough requirement gathering to effectively guide the project's direction. Through detailed stakeholder engagement and iterative feedback, the team gains a deep understanding of constituent needs, representative functionalities, and system usability expectations. This comprehensive approach helps identify key use cases and user stories that drive the platform’s design and development, ensuring Constituency Connect meets its objective of improving communication and responsiveness between government representatives and their constituents.

3.1. Use Cases

The use cases outlined in this section provide a comprehensive overview of user interactions with the Constituency Connect system, detailing how various roles engage with the platform to accomplish key tasks. These use cases clarify core functionalities, validate requirements, and capture the essential interactions between constituents, representatives, and administrative roles. They include scenarios such as complaint submission, complaint tracking, response management, and notifications, demonstrating the system’s responsiveness to a wide range of user needs.

By illustrating how Constituency Connect addresses tasks from complaint lodging to resolution tracking, the use cases facilitate clear communication between stakeholders and the development team. This alignment of requirements and expectations ensures that the project is developed with a focus on user needs, guiding the creation of a platform that fosters efficient and transparent communication between constituents and their representatives.

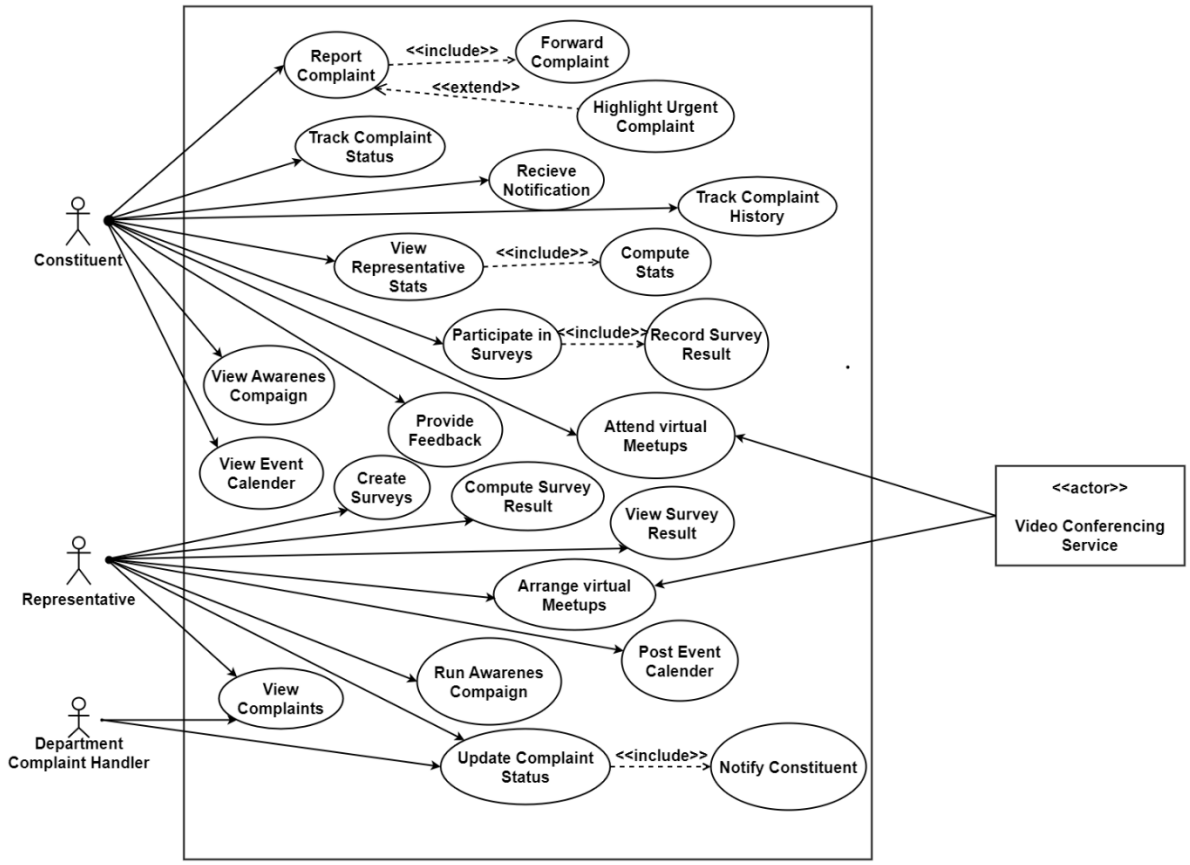

Fig 3.1: Use Case diagram

3.1.1 Actors Description:

The actors and the description of the actors is

1. **Constituent:**

   Description: Constituents are citizens who interact with the system to register complaints, provide feedback, and communicate with representatives. Their primary role involves submitting issues that need resolution, accessing their case status, and engaging with representatives via the platform's tools, including video conferencing as needed. Their interactions help ensure that representatives understand the needs of their constituencies.

2. **Representative:**

   Description: Representatives, such as parliamentarians, respond to constituent issues, initiate actions, and review feedback. They interact with constituents through various

communication modes, including virtual meetups. Their involvement is vital to resolving cases, addressing public concerns, and improving community welfare based on received complaints and feedback.

3. **Department Complaint Handler:**
   Description: Represents various government departments or agencies responsible for handling specific types of complaints (e.g., health services). They receive complaints, provide status updates, and close complaints upon resolution.

4. **Video Conferencing System (Secondary Actor):**
   Description: Integrated as a secondary actor, the video conferencing system facilitates direct communication between constituents and representatives. This functionality is essential for more complex cases that require in-depth discussion. It enhances engagement and improves case resolution efficiency by enabling real-time interaction.

3.2. Use Case Description:

The following are the Expanded Use Cases of our System.

3.2.1. Report Complaint

Use Case ID:	UC-01
Use Case Name:	Report Complaint
Actors:	Constituent
Description:	Enable a constituent to complain about an issue requiring attention
Trigger:	The constituent identifies a problem and wishes to report it.
Preconditions:	1. The constituent must be registered in the system. 2. The reported issue must fall within a valid category recognized by the system.
Postconditions:	1. The complaint is recorded in the system and assigned a unique identifier. 2. Notifications may be sent to relevant stakeholders.
Normal Flow:	1. Constituent identifies an issue they wish to report. 2. Constituent initiates the process to report a complaint.

	3. The system prompts the constituent to provide necessary details, including a description, location, and category. 4. The constituent submits the complaint. 5. The system validates the details and creates a complaint record. 6. The system confirms the successful submission to the constituent and generates a unique complaint ID. 7. The system routes the complaint to the relevant representative or department.
Alternative Flows:	**Missing Information:** If mandatory information is missing, the system prompts the user to complete the fields.
Includes:	Forward Complaint
Special Requirements:	The system should ensure the security of sensitive complaint information.
Assumptions:	Constituents have access to the internet to report complaints. Notes and Issues
Notes and Issues:	None

Table 3.1: Report Complaint

3.2.2. Forward Complaint

Use Case ID:	UC-02
Use Case Name:	Forward Complaint
Actors:	Constituent (Primary), Representative, Department Complaint Handler
Description:	Allows a constituent's complaint to be forwarded to the appropriate authority for resolution.
Trigger:	The complaint is determined to require attention from a specific authority.
Preconditions:	The complaint must already exist in the system.
Postconditions:	The complaint is successfully forwarded to the selected authority, and its status is updated.
Normal Flow:	1. The system determines where the complaint needs to be forwarded. 2. The forwarding process is initiated.

	3. The system sends complaints to the selected authority, including its status. 4. A confirmation is sent to the complainant, and the complaint status is updated.
Alternative Flows:	**Incorrect Category:** If the complainant selects the wrong category, DCH or the Representative can forward it to the right authority.
Includes:	None
Special Requirements:	Must ensure reliable tracking of forwarded complaints.
Assumptions:	Authorities are already registered in the system
Notes and Issues:	None

Table 3.2: Forward Complaint

3.2.3. Highlight Urgent Complaint

Use Case ID:	UC-03
Use Case Name:	Highlight Urgent Complaint
Actors:	Constituent
Description:	Allows a constituent to mark a complaint as urgent, ensuring it receives priority handling.
Trigger:	The system identifies the complaint as requiring immediate attention.
Preconditions:	The complaint must exist in the system.
Postconditions:	The complaint is marked as urgent, and notifications are sent to relevant stakeholders.
Normal Flow:	1. Constituent identifies an existing complaint that requires urgent attention. 2. The system verifies that the complaint qualifies for urgency. 3. The system marks the complaint as urgent and notifies the relevant stakeholders.
Alternative Flows:	None
Includes:	None
Exceptions:	None
Special Requirements:	None

Assumptions:	Stakeholders are equipped to handle urgent complaints.
Notes and Issues:	None

Table 3.3: Highlight Urgent Complaint

3.2.4. Track Complaint Status

Use Case ID:	UC-04
Use Case Name:	Track Complaint Status
Actors:	Constituent
Description:	Enables constituents to check the progress and current status of their complaints.
Trigger:	The constituent wishes to know the status of a complaint
Preconditions:	The constituent must have submitted at least one complaint.
Postconditions:	The constituent is informed of the complaint's current status and history.
Normal Flow:	1. Constituent initiates the request to track a complaint. 2. The system prompts for the complaint ID. 3. The constituent provides the complaint ID. 4. The system retrieves and displays the status and progress of the complaint.
Alternative Flows:	**Invalid Complaint ID:** If the ID provided does not exist, the system notifies the user and prompts them to retry.
Includes:	None
Exceptions:	None
Special Requirements:	None.
Assumptions:	Complaint records are stored securely and reliably.
Notes and Issues:	None

Table 3.4: Track Complaint Status

3.2.5. Receive Notification

Use Case ID:	UC-05
Use Case Name:	Receive Notification
Actors:	Constituent
Description:	Provides constituents with updates, announcements, or alerts regarding complaints, events, or surveys.
Trigger:	A notification is triggered by an event in the system.
Preconditions:	The constituent must be registered.
Postconditions:	The notification is successfully delivered to the constituent.
Normal Flow:	1. A triggering event (e.g., complaint update, survey invitation) occurs. 2. The system identifies the target constituent(s) for the notification. 3. The system sends the notification. 4. The constituent receives the notification.
Alternative Flows:	**Notification Delivery Failure:** The system logs the failure and retries to send the notification.
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Table 3.5: Receive Notification

3.2.6. Track Complaint History

Use Case ID:	UC-06
Use Case Name:	Track Complaint History
Actors:	Constituent
Description:	Allows a constituent to view the complete history and updates of their complaints
Trigger:	The constituent wants to review their past complaints.
Preconditions:	The constituent must have submitted at least one complaint.

Postconditions:	The constituent is presented with the complaint's historical details.
Normal Flow:	1. Constituent initiates the request to view complaint history. 2. The system retrieves all complaints associated with the constituent. 3. The constituent selects a specific complaint to view its history. 4. The system displays a detailed history, including updates and actions taken.
Alternative Flows:	**No Complaints Found:** If no complaints are associated with the constituent, the system notifies them accordingly.
Exceptions:	None
Includes:	None
Special Requirements:	Complaint history must be accurate and complete.
Assumptions:	Complaint records are maintained indefinitely.
Notes and Issues:	None

Table 3.6: Track Complaint History

3.2.6. View Representative Stats

Use Case ID:	UC-07
Use Case Name:	View Representative Stats
Actors:	Constituent.
Description:	Enable constituents to view statistical data regarding the performance and activities of their elected representatives.
Trigger:	A constituent wants to evaluate the effectiveness of a representative based on statistics.
Preconditions:	1. The representative must have performance data recorded in the system. 2. The constituent must have access to the representative’s public data.
Postconditions:	The constituent is presented with detailed statistics on the representative's activities.
Normal Flow:	1. Constituent requests to view representative statistics.

	2. The system retrieves the representative's statistical data from the database. 3. The system displays aggregated statistics such as the number of resolved complaints, surveys conducted, and events organized.
Alternative Flows:	**No Data Available:** If no statistics are recorded, the system notifies the constituent.
Includes:	Compute Stats
Exceptions:	None
Special Requirements:	Must ensure data accuracy and prevent manipulation of stats.
Assumptions:	Representatives' activities are tracked consistently.
Notes and Issues:	Data visualization tools should be understandable for non-technical users.

Table 3.7: View Representative Stats

3.2.8. Compute Stats

Use Case ID:	UC-08
Use Case Name:	Compute Stats
Actors:	System (Primary), Representative (Secondary)
Description:	Automatically calculates performance statistics based on data recorded in the system.
Trigger:	The system detects new data or a request to compute statistics.
Preconditions:	1. Representative activities are logged into the system. 2. The system must have pre-configured statistical algorithms.
Postconditions:	The updated statistics are accessible for constituents and representatives to view
Normal Flow:	1. The system identifies updated data from complaints, surveys, and events. 2. The system applies predefined algorithms to compute statistics such as response rates, resolution times, and survey participation. 3. The computed stats are stored for future retrieval.
Alternative Flows:	None
Exceptions:	None

Includes:	None
Special Requirements:	Algorithms must be optimized for large datasets.
Assumptions:	Representative activity data is updated frequently.
Notes and Issues:	None

Table 3.8 Compute Stats

3.2.9. Participate in Survey

Use Case ID:	UC-09
Use Case Name:	Participate in Survey
Actors:	Constituent
Description:	Enables constituents to participate in surveys created by representatives to collect community feedback.
Trigger:	A constituent is invited or wishes to respond to a survey.
Preconditions:	1. The constituent must have access to the survey. 2. The survey must be active.
Postconditions:	The survey response is recorded and linked to the respective constituent.
Normal Flow:	1. The constituent selects an active survey to participate in. 2. The system presents the survey questions. 3. Constituent responds to each question. 4. The system validates and submits the responses. 5. The system confirms successful submission to the constituent.
Alternative Flows:	None
Exceptions:	None
Includes:	Record Survey Result
Special Requirements:	Surveys must support multiple question types (e.g., multiple choice, open-ended).
Assumptions:	None
Notes and Issues:	None

Table 3.9 Participate in Survey

3.2.10. Record Survey Result

Use Case ID:	UC-10
Use Case Name:	Record Survey Result
Actors:	System (Primary), Representative (Secondary)
Description:	Captures and stores survey responses submitted by constituents
Trigger:	A constituent submits survey responses.
Preconditions:	The survey must exist, and questions must be defined.
Postconditions:	Survey responses are stored and linked to the respective survey.
Normal Flow:	1. The system receives survey responses from a constituent. 2. The system validates the responses against the survey structure. 3. The responses are stored in the database. 4. The system updates survey participation statistics
Alternative Flows:	**Invalid Responses:** The system rejects incomplete or invalid responses.
Exceptions:	None
Includes:	None
Special Requirements:	Ensure responses are stored properly.
Assumptions:	None
Notes and Issues:	None

Table 3.10 Record Survey Result

3.2.11. Attend Virtual Meetups

Use Case ID:	UC-11
Use Case Name:	Attend Virtual Meetups
Actors:	Constituent (Primary), Video Conferencing Service (Secondary)
Description:	Enables constituents to join virtual meetings hosted by representatives to discuss community issues.
Trigger:	A constituent is invited to a virtual meeting or wishes to join one.
Preconditions:	The meeting must be scheduled and active.
Postconditions:	The constituent successfully joins the virtual meeting.
Normal Flow:	1. Constituent receives an invitation to a virtual meeting.

	2. The system verifies the meeting details and provides a join link. 3. The constituent initiates the process of joining the meeting. 4. The system authenticates the constituent and redirects them to the video conferencing platform. 5. The video conferencing service establishes the connection.
Alternative Flows:	None
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Table 3.11 Attend Virtual Meetups

3.2.12. Provide Feedback

Use Case ID:	UC-12
Use Case Name:	Provide Feedback
Actors:	Constituent
Description:	Allows constituents to submit feedback on complaint resolution.
Trigger:	A constituent wishes to share their feedback regarding their experience.
Preconditions:	A constituent's complaint must be closed (resolved or rejected).
Postconditions:	Feedback is recorded and stored for review by relevant authorities.
Normal Flow:	1. Constituent accesses the feedback section. 2. Constituent provides detailed feedback, including optional ratings or attachments. 3. The system validates the feedback form. 4. Feedback is submitted and stored in the database. 5. The system acknowledges the feedback submission to the constituent.
Alternative Flows:	**Incomplete Form:** If required fields are left empty, the system prompts the constituent to complete them before submission.

Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	Constituents provide honest and constructive feedback.
Notes and Issues:	None

Table 3.12 Provide Feedback

3.2.13. View Awareness Campaign

Use Case ID:	UC-13
Use Case Name:	View Awareness Campaign
Actors:	Constituent
Description:	Allow constituents to view active awareness campaigns initiated by representatives.
Trigger:	A constituent wishes to learn about ongoing community awareness initiatives.
Preconditions:	1. Awareness campaigns must be published and active. 2. The constituent must have access to the platform.
Postconditions:	The constituent views details of the awareness campaign.
Normal Flow:	1. Constituent navigates to the awareness campaign section. 2. The system retrieves all active campaigns from the database. 3. The system displays campaigns, including descriptions, and schedules. 4. The constituent selects a campaign to view detailed information.
Alternative Flows:	None
Exceptions:	None
Includes:	None
Special Requirements:	Campaigns must include engaging content.
Assumptions:	Constituents regularly check for campaign updates.
Notes and Issues:	Ensure campaigns are accessible across devices and platforms.

Table 3.13 View Awareness Campaign

3.2.14. View Awareness Campaign

Use Case ID:	UC-14
Use Case Name:	View Event Calendar
Actors:	Constituent
Description:	Enables constituents to view scheduled events such as virtual meetups, surveys, or community initiatives.
Trigger:	A constituent wants to check upcoming events in their community
Preconditions:	The system must have an updated event calendar.
Postconditions:	The constituent views the list of events with corresponding details.
Normal Flow:	1. Constituent accesses the event calendar section. 2. The system retrieves the event details from the database. 3. The system displays the event calendar, showing dates and brief descriptions. 4. The constituent selects an event for detailed information.
Alternative Flows:	None
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	Events are updated in the system promptly.
Notes and Issues:	Ensure the calendar is visually clear and user-friendly.

Table 3.14 View Event Calendar

3.2.15 Create Survey

Use Case ID:	UC-15
Use Case Name:	Create Surveys
Actors:	Representative
Description:	Allows representatives to create surveys for constituents to gather feedback or opinions on various topics.
Trigger:	A representative decides to initiate a survey to collect data from constituents.
Preconditions:	The system must support survey creation functionality.

Postconditions:	The survey is saved and becomes available for constituents to participate in.
Normal Flow:	1. The representative navigates to the survey creation section. 2. The system prompts the representative to define the survey title. 3. The representative adds questions, specifying formats (e.g., multiple-choice, text response). 4. Representative sets survey parameters, including start and end dates and visibility. 5. The system validates the survey and saves it in the database. 6. The survey is published.
Alternative Flows:	None
Exceptions:	None
Includes:	None
Special Requirements:	The survey must support multiple question types and allow scheduling.
Assumptions:	None
Notes and Issues:	Ensure surveys are accessible on all devices.

Table 3.15 Create Surveys

3.2.16. Compute Survey Results

Use Case ID:	UC-16
Use Case Name:	Compute Survey Results
Actors:	Representative
Description:	Enables representatives to calculate and analyze survey responses for insights.
Trigger:	A representative wants to analyze data from a completed survey.
Preconditions:	The survey must be completed and responses stored in the database.
Postconditions:	Survey results are computed and presented in a report format.
Normal Flow:	1. The representative selects a complete survey for analysis. 2. The system retrieves survey responses from the database. 3. The system computes results, generating aggregated statistics and insights. 4. Representative views results in graphical and tabular form.

Alternative Flows:	**No Responses:** If there are no available responses, the system alerts the representative.
Exceptions:	None
Includes:	None
Special Requirements:	The system should support advanced data visualization.
Assumptions:	Responses are complete and accurately recorded.
Notes and Issues:	Ensure reports are exportable for external use.

Table 3.16 Compute Survey Results

3.2.17. View Survey Results

Use Case ID:	UC-17
Use Case Name:	View Survey Results
Actors:	Representative
Description:	Allows representatives to review computed results and gain insights from constituent responses.
Trigger:	A representative seeks to view results of a previously conducted survey
Preconditions:	Survey results must be computed and available in the database.
Postconditions:	Survey results are displayed to the representative.
Normal Flow:	1. The Representative accesses the survey results section. 2. The system lists all completed surveys with available results. 3. The representative selects a specific survey to assess detailed results. 4. The system displays computed results in visual formats.
Alternative Flows:	**No Results:** The system notifies the representative if results are unavailable.
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	Results are accurate and formatted for easy understanding.
Notes and Issues:	None

Table 3.17 View Survey Results

3.2.18. Arrange Virtual Meetups

Use Case ID:	UC-18
Use Case Name:	Arrange Virtual Meetups
Actors:	Representative, Secondary Actor: Video Conferencing Service
Description:	Enables representatives to organize virtual meetings with constituents or other stakeholders.
Trigger:	A representative decides to arrange a virtual meetup.
Preconditions:	The video conferencing service must be integrated with the platform.
Postconditions:	Virtual meetup is scheduled and invitations are sent to participants.
Normal Flow:	1. The representative accesses the virtual meetup scheduling section. 2. The system prompts the representative to set a title, date, and time for the meetup. 3. The representative adds participants and specifies the meeting agenda. 4. The system generates a meeting link through the integrated video conferencing service. 5. Invitations are sent to participants.
Alternative Flows:	None
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	Video conferencing service is reliable.
Notes and Issues:	None

Table 3.18 Arrange Virtual Meetups

3.2.19. Post Event Calendar

Use Case ID:	UC-19
Use Case Name:	Post Event Calendar
Actors:	Representative
Description:	Allows representatives to create and publish an event calendar to inform constituents about upcoming events.

Trigger:	A representative wants to share event details with constituents.
Preconditions:	The system must support calendar management.
Postconditions:	The event calendar is updated and visible to constituents.
Normal Flow:	1. The representative accesses the event calendar management section. 2. The system displays the current calendar. 3. The representative creates an event, specifying details such as title, date, time, and description. 4. The representative confirms and saves the changes. 5. The system updates the event calendar and notifies constituents.
Alternative Flows:	None
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	Representatives maintain accurate and relevant event details.
Notes and Issues:	None

Table 3.19 Post Event Calendar

3.2.20. Run Awareness Campaign

Use Case ID:	UC-20
Use Case Name:	Run Awareness Campaign
Actors:	Representative
Description:	Enables representatives to launch awareness campaigns for constituents on various topics.
Trigger:	A representative has decided to initiate a public awareness campaign.
Preconditions:	The representative must have access to campaign management features.
Postconditions:	The campaign is published and available for constituents to view.
Normal Flow:	1. The representative navigates to the campaign creation section.

	2. The system prompts for campaign details such as title, content, and duration. 3. The representative uploads any relevant media or resources. 4. Representative reviews and publishes the campaign. 5. The system broadcasts the campaign to constituents through appropriate channels.
Alternative Flows:	None
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	Monitor constituent engagement for campaign effectiveness.

Table 3.20 Run Awareness Campaign

3.2.21View Complaints

Use Case ID:	UC-21
Use Case Name:	View Complaints
Actors:	Representative, Department Complaint Handler
Description:	Enable representatives and department complaint handlers to view complaints submitted by constituents.
Trigger:	A user accesses the complaint viewing feature.
Preconditions:	1. Complaints must exist in the system database. 2. The user must have the necessary permissions.
Postconditions:	Complaints are displayed with relevant details.
Normal Flow:	1. The user logs into the system and navigates to the complaints section. 2. The system retrieves complaints from the database. 3. The user sorts of complaints based on criteria such as urgency, status, or date. 4. The system displays the list of complaints.
Alternative Flows:	None
Exceptions:	None

Includes:	None
Special Requirements:	None
Assumptions:	Complaint data is accurate and up to date.
Notes and Issues:	None

Table 3.21 View Complaints

3.2.22. Update Complaint Status

Use Case ID:	UC-22
Use Case Name:	Update Complaint Status
Actors:	Representative, Department Complaint Handler
Description:	Allow authorized actors to update the status of complaints.
Trigger:	A user decides to update the status of a specific complaint.
Preconditions:	1. Complaints must exist in the system database. 2. The user must have the necessary permissions.
Postconditions:	The complaint status is updated and reflected in the system.
Normal Flow:	1. The user selects a complaint from the list. 2. The system displays details of complaints, including the status. 3. The user updates the status and provides optional remarks. 4. The system validates and saves updated status. 5. The constituent linked to the complaint are notified of the status change.
Alternative Flows:	**Invalid Update:** If the status change violates predefined rules, the system notifies the user and rejects the update.
Exceptions:	None
Includes:	Notify Constituents
Special Requirements:	The system should log all status changes for audit purposes.
Assumptions:	None
Notes and Issues:	Ensure notifications are timely and correctly formatted.

Table 3.22 Update Complaint Status

3.3. Functional Requirements:

Users (Constituents and Representatives)

1. The system shall allow users (constituents and representatives) to register on the platform.
2. Users shall be able to log in to their accounts using valid credentials.

Constituents:

3. Constituents shall be able to report a complaint to their respective representatives through the system.
4. Constituents shall be able to track the status of their complaints.
5. Constituents shall have access to a history of their submitted complaints**.**
6. Constituents shall be able to view their representative's performance statistics, including (Total complaints received, number of complaints resolved, pending complaints, and Total meetups conducted etc)
7. Constituents shall be able to participate in surveys and polls organized by their representatives.
8. Constituents shall be able to attend virtual meetups arranged by their representatives.
9. Constituents shall be able to provide feedback on complaint resolutions and representative performance.

Representative:

10. Representatives should only register with their official email address (e.g., johndoe@na.gov.pk).
11. Representatives shall be able to create and distribute custom surveys to gather insights from constituents.
12. Representatives shall have the ability to schedule and host virtual meetings with constituents.
13. Representatives shall be able to update the status of complaints to reflect progress or resolution.
14. Representatives shall be able to forward complaints to the relevant departments for action**.**
15. Representatives shall be able to post-event calendars displaying upcoming community events and public meetings.

16. Representatives shall be able to run awareness campaigns.

Miscellaneous:

17. The system shall automatically route categorized complaints to the relevant department for efficient resolution.
18. Uncategorized complaints shall be routed to the representative for manual action.
19. The system shall notify constituents about updates on the status of their complaints and upcoming virtual meetings organized by their representatives.
20. The system shall record survey responses from participants.
21. Survey results shall be displayed using visualizations, such as charts and graphs, for easy interpretation.
22. The system should prioritize complaints containing sensitive keywords to ensure urgent issues receive prompt attention.
23. The Department Complaint Handler shall be able to update the status of complaints assigned to them, ensuring transparency and tracking.
24. The Department Complaint Handler shall be able to reroute miscategorized complaints to the appropriate representative for further action.

3.4. Non-Functional Requirements:

- Each page must be loaded within 2 seconds for 95% of user interactions.
- The system should be scaled to support up to 10,000 concurrent users.
- The system must ensure 99.9% uptime availability, allowing no more than 8 hours of downtime annually.
- Complaint submission confirmation must be provided within 2 seconds.
- Sensitive user data, such as CNIC and addresses, must be encrypted using AES-256 both in transit and at rest.
- The system should support browser compatibility for the latest two versions of Chrome, Firefox, Edge, and Safari.
- The system must handle complaint routing with a response time of under 2 seconds per action.
- User sessions should time out after 30 minutes of inactivity for enhanced security.

Chapter 4: Design and Architecture

This chapter explores the design and architecture of the **Constituency Connect** system, highlighting essential principles and architectural decisions crucial for the project's success. It emphasizes the role of well-structured design in ensuring system efficiency, scalability, and user-friendliness.

It details the system's structure and architectural components, providing a blueprint for translating functional requirements into a practical and reliable solution. The discussion covers user role interactions, complaint management workflows, authentication mechanisms, and external integrations.

It also outlines the technical foundation of the project, fostering a deeper understanding of how the system efficiently manages complaint resolution, feedback mechanisms, and representative engagement, ensuring seamless public service accessibility.

4.1. System Architecture

The system follows a layered architecture, ensuring modularity, scalability, and efficient data flow. The system is structured into the following key tiers:

Presentation Layer

Business Logic Layer

Data Access Layer

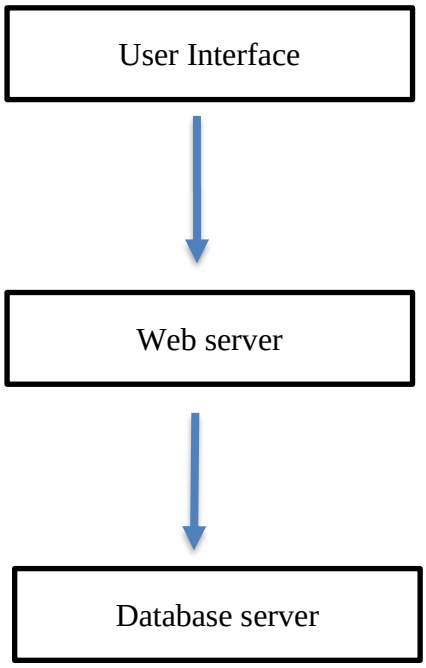

Fig 4.1: System Architecture

a. Presentation Layer (User Interface):

- This layer deals with the user interfaces, where users (constituents, representatives, and complaint handlers) interact with the system.
- Users can submit complaints, track statuses, manage events, provide feedback, and access other system functionalities.
- Ensuring a clear separation of the UI allows for an intuitive, responsive, and accessible design, even for non-technical users.

b. Business Logic Tier (Web Server):

- This layer handles the core logic of the system, including complaint processing, event management, notification handling, and user role-based access control.
- It validates complaints, categorizes them, triggers notifications, and ensures smooth communication between different roles.
- A modular service-oriented approach allows easy updates or expansions without affecting the UI or Data Layer.

c. Data Access Layer (Database Server):

- This layer is responsible for storing and managing system data, including user details, complaints, feedback, events, and notifications.
- Ensuring a separate data layer enhances security, privacy, and performance.

4.1.1. Block Diagram

The Block Diagram provides a high-level representation of the Constituency Connect System, illustrating its main functional components and their interactions. It offers a simplified view of how different modules connect, focusing on data flow and system structure without detailing specific implementation aspects.

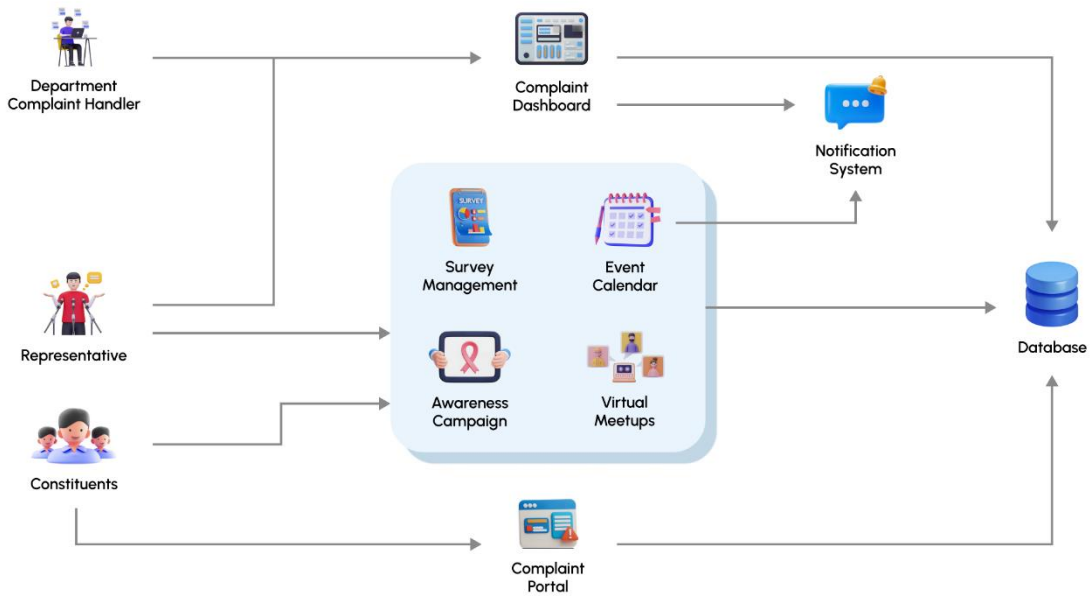

Fig 4.2: Block Diagram

4.2. System Design

System design is the process of defining the structure, components, modules, and interactions within a system to ensure it meets the specified requirements. The Constituency Connect system is designed to provide an efficient, scalable, and secure platform for managing complaints, feedback, events, and communication between constituents, representatives, and complaint handlers.

This section outlines the system design, detailing key components and their interactions.

4.2.1. UML Structural Diagrams

Structural UML diagrams show how the system is structured, including the classes, objects, components, etc. in the system and the relationships between the elements.

4.2.1.1 Domain Model:

The Domain Model provides a high-level conceptual representation of the system, defining key entities, their attributes, and relationships. It serves as a bridge between real-world concepts and system implementation.

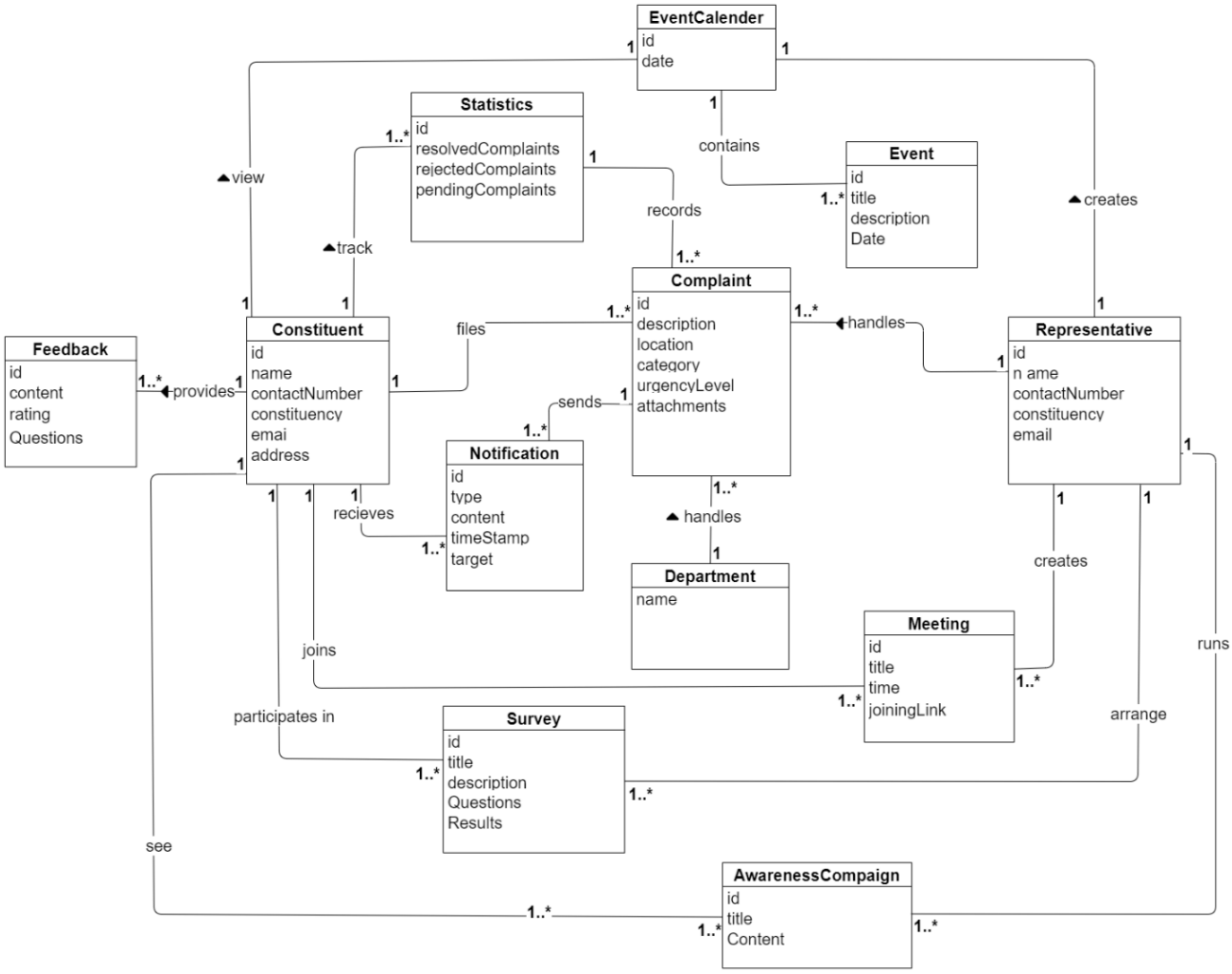

Fig 4.3: Domain Model

4.2.1.2. Class Diagram:

The Class Diagram provides a detailed static view of the system by defining its classes, attributes, methods, and relationships.

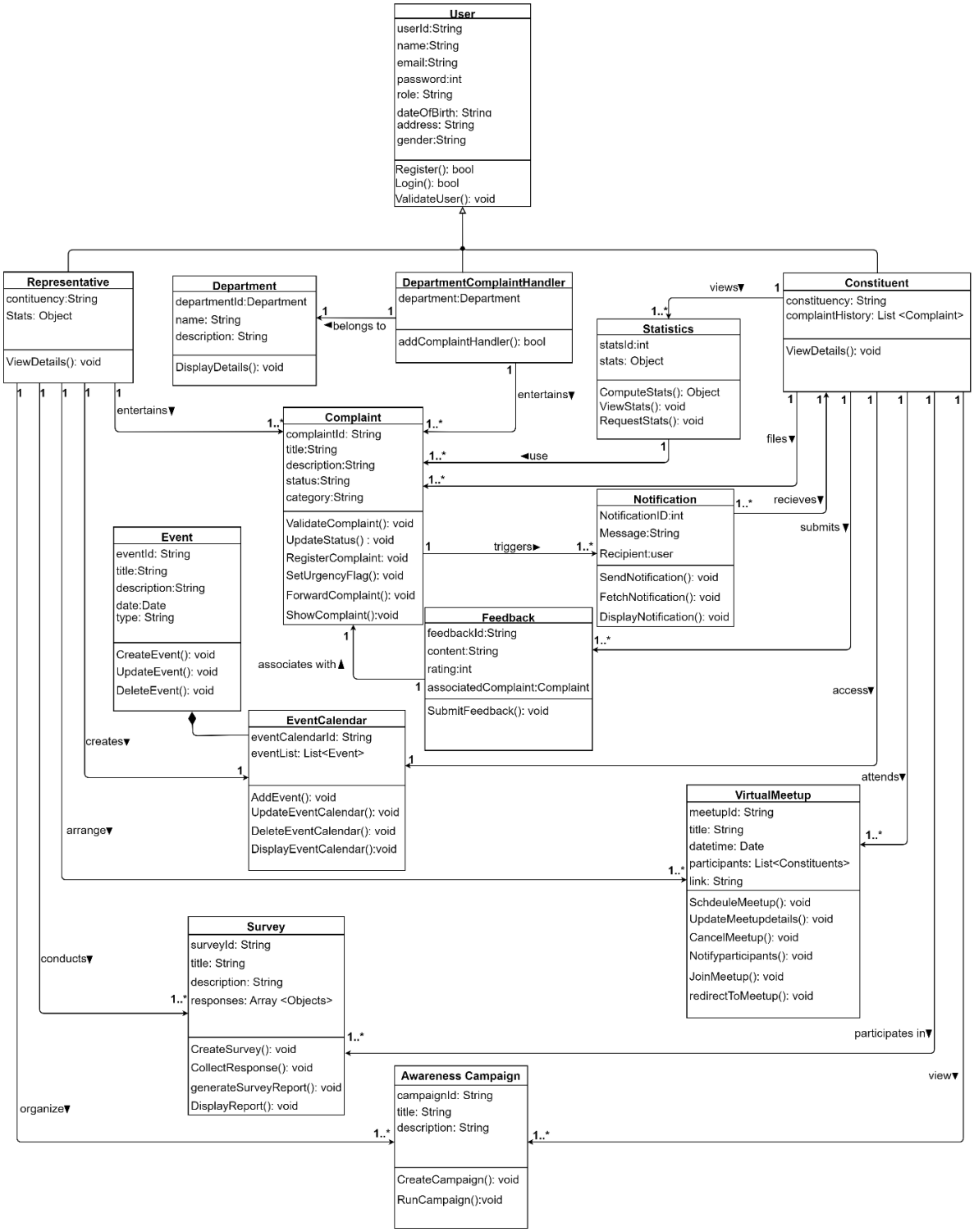

Fig 4.4: Class Diagram

4.2.1.3. Component Diagram:

Component Diagram displays the software components of the system and their dependencies, showcasing how different modules interact to provide system functionality.

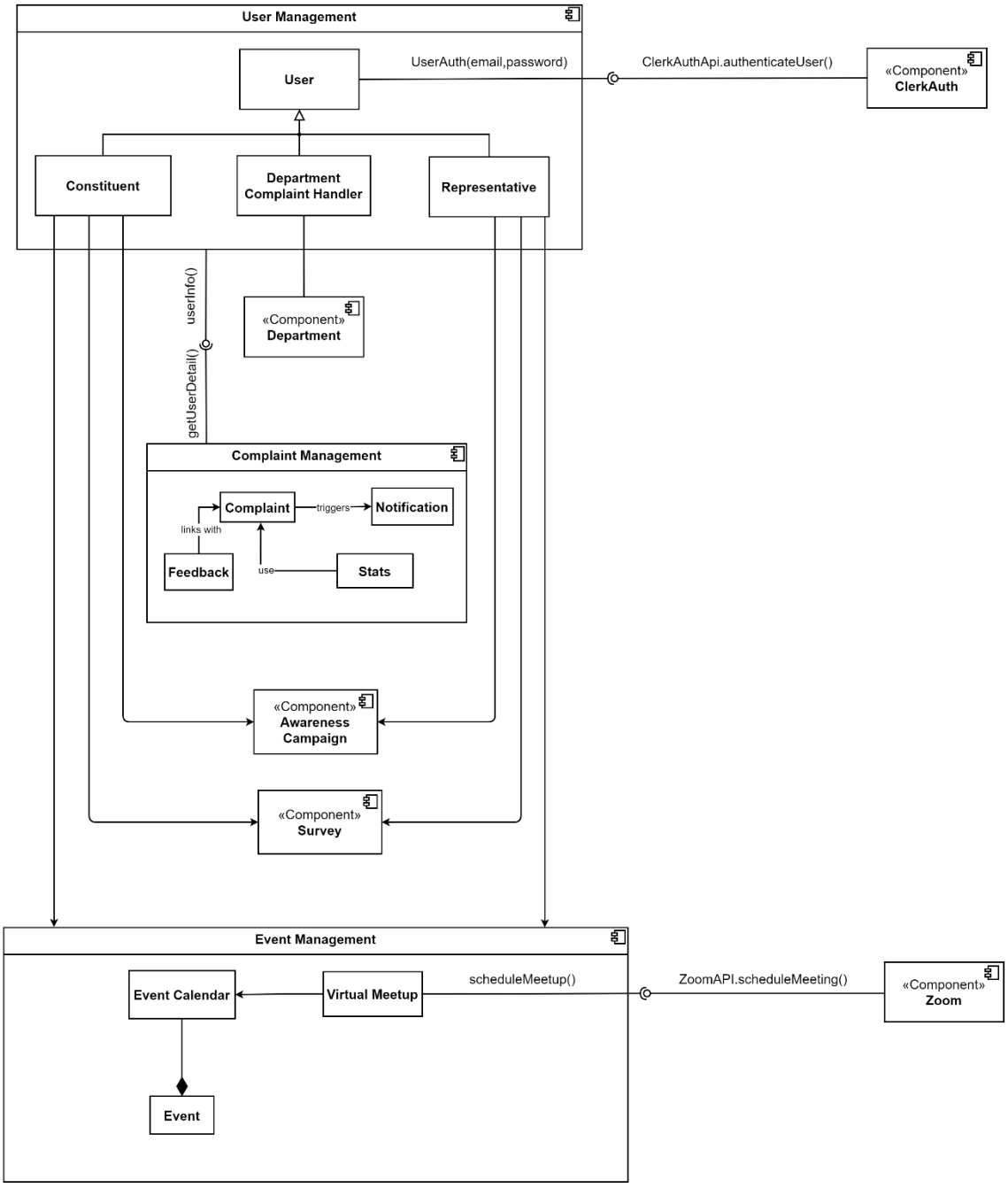

Fig 4.5: Component Diagram

4.2.1.4. Deployment Diagram:

This diagram depicts the physical deployment of software components across hardware nodes, illustrating how the system's modules are distributed and interconnected within a network infrastructure. It visualizes the deployment architecture, including servers, databases, and other resources, facilitating effective system deployment and management.

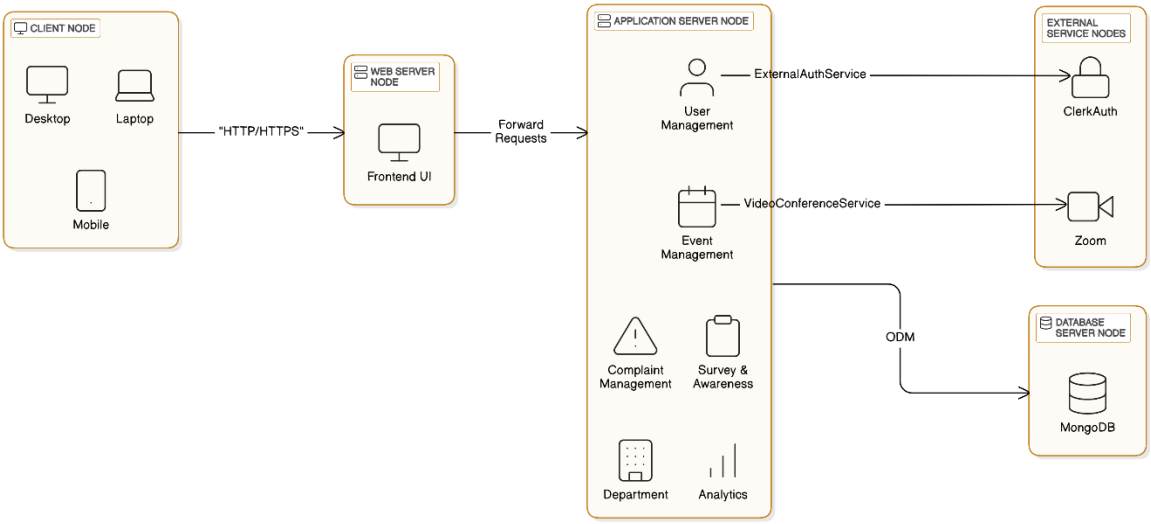

Fig 4.6: Deployment Diagram

4.2.1.5. Entity Relation Diagram

Defines the data model by representing entities, their attributes, and relationships, providing a clear structure for database design and management.

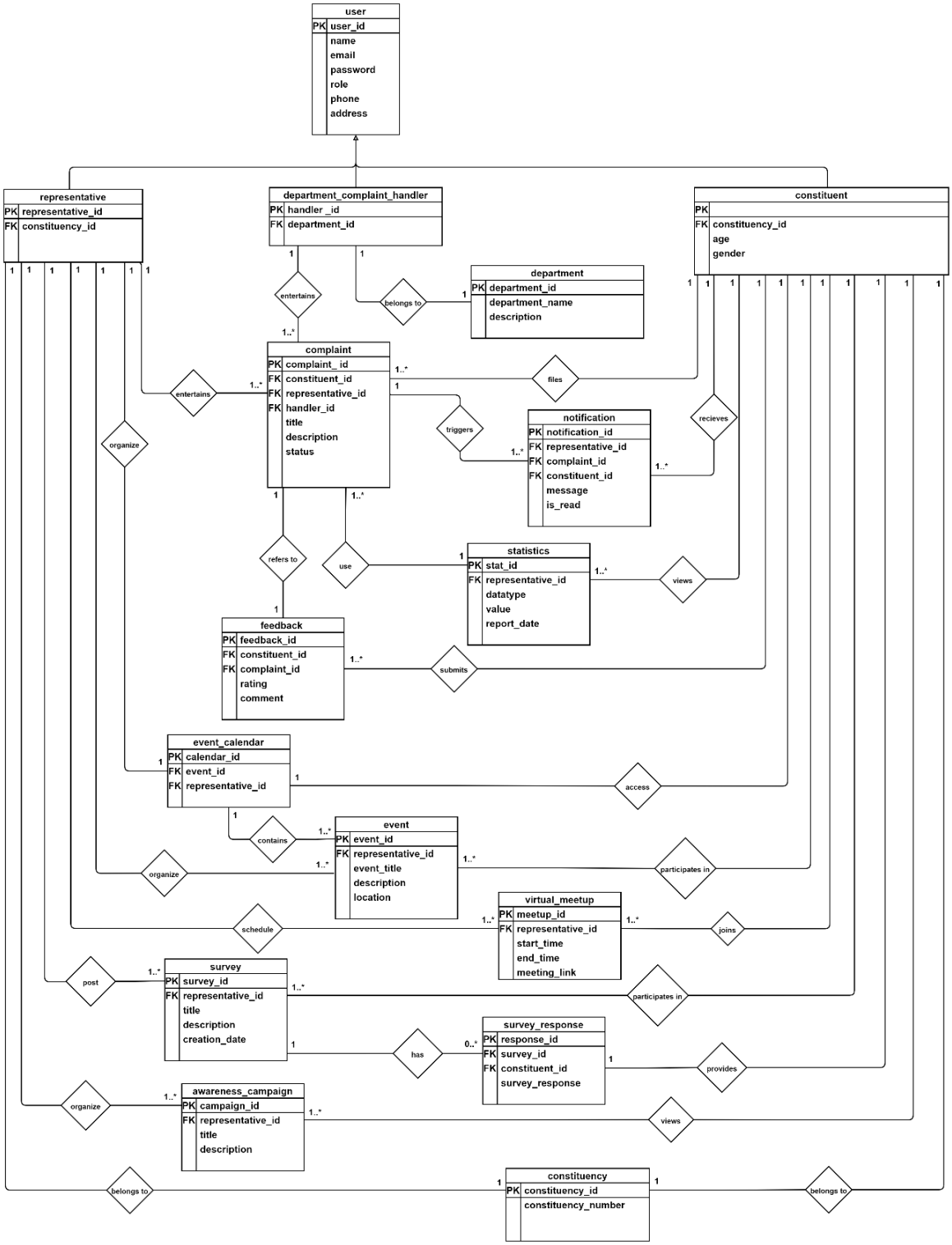

Fig 4.7: ER Diagram

4.2.2. UML Behavioral Diagrams

UML Behavioral Diagrams provide a visual representation of the system's dynamic behavior, showcasing how different components and entities interact to achieve specific tasks. These diagrams help in understanding functional workflows, message exchanges, and state transitions, ensuring a structured approach to system design and process modeling. By offering a clear view of system interactions and logic, they facilitate effective stakeholder communication and streamlined development.

4.2.2.1. Activity Diagrams:

The flow of activities within a system, illustrating the sequence of actions or steps required to accomplish a specific task or behavior.

Login – Registration Activity Diagram:

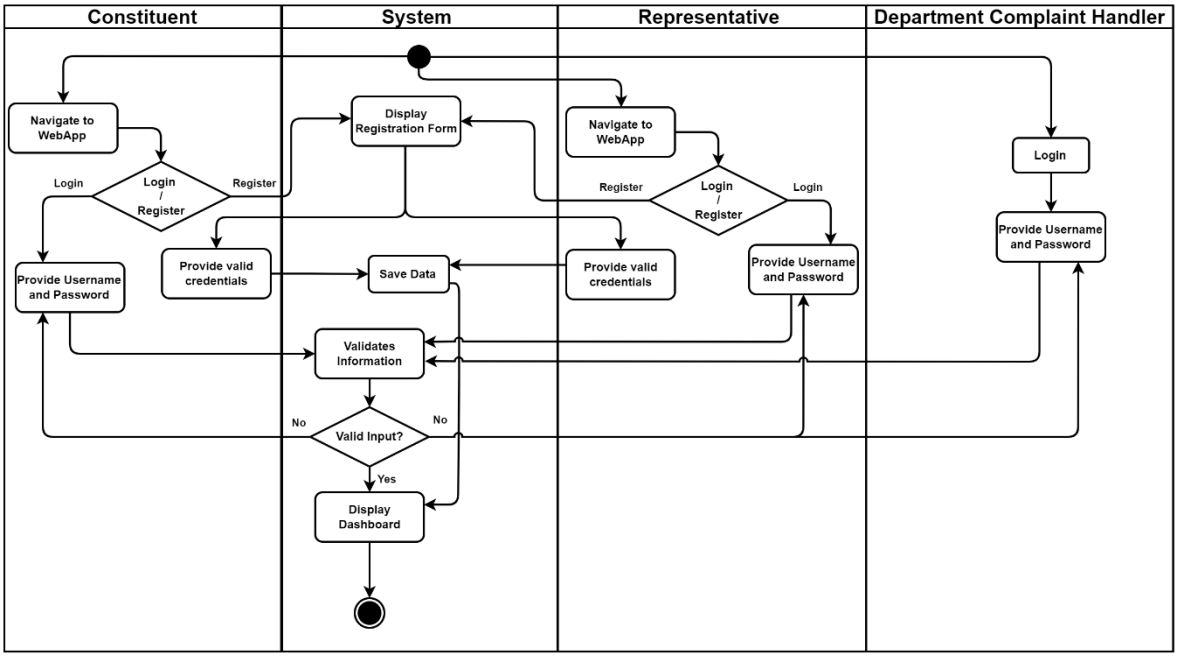

Fig 4.8: Login – Registration Activity Diagram

Complaint Resolution Activity Diagram:

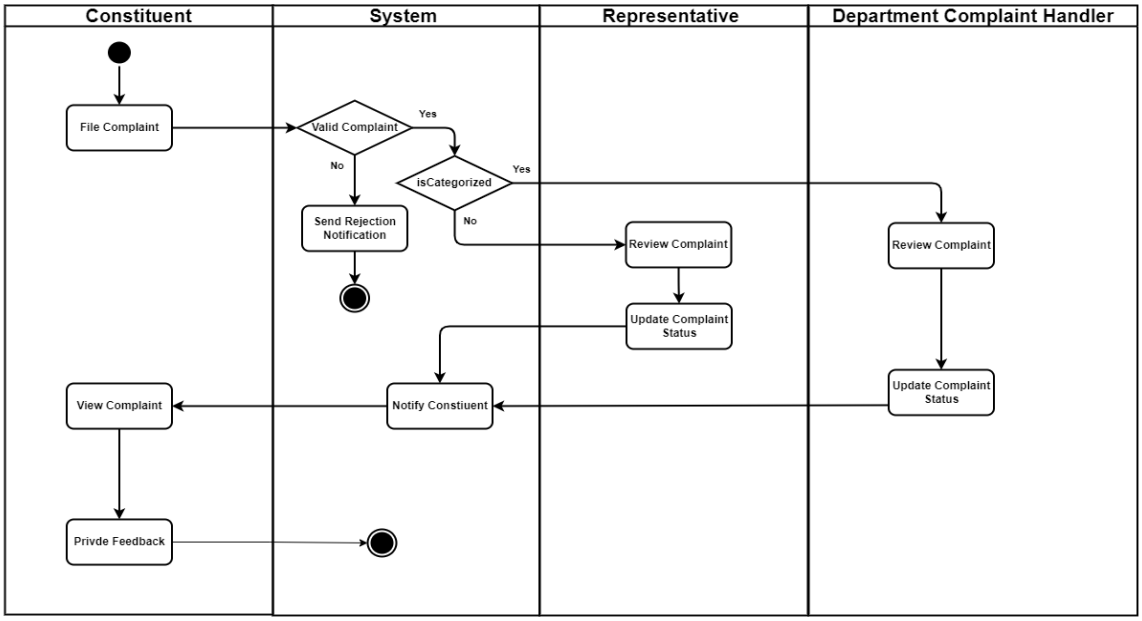

Fig 4.9: Complaint Resolution Activity Diagram

View Statistics Activity Diagram:

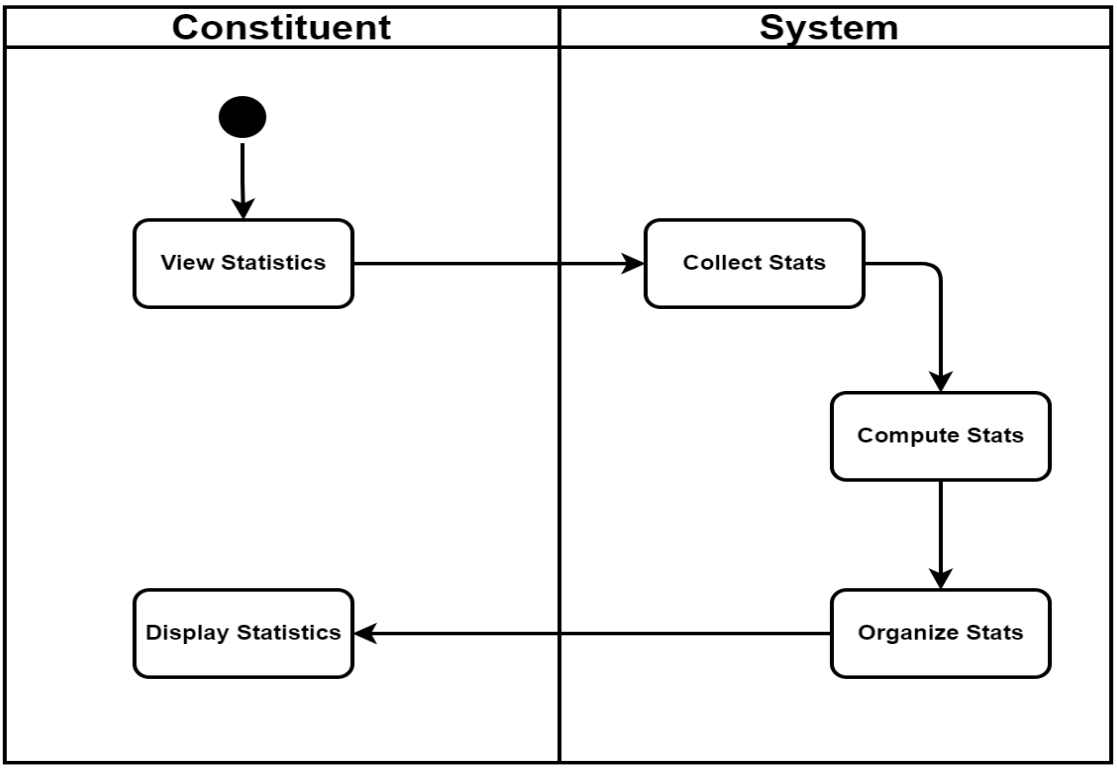

Fig 4.10: View Statistics Activity Diagram

Event Calendar Activity Diagram:

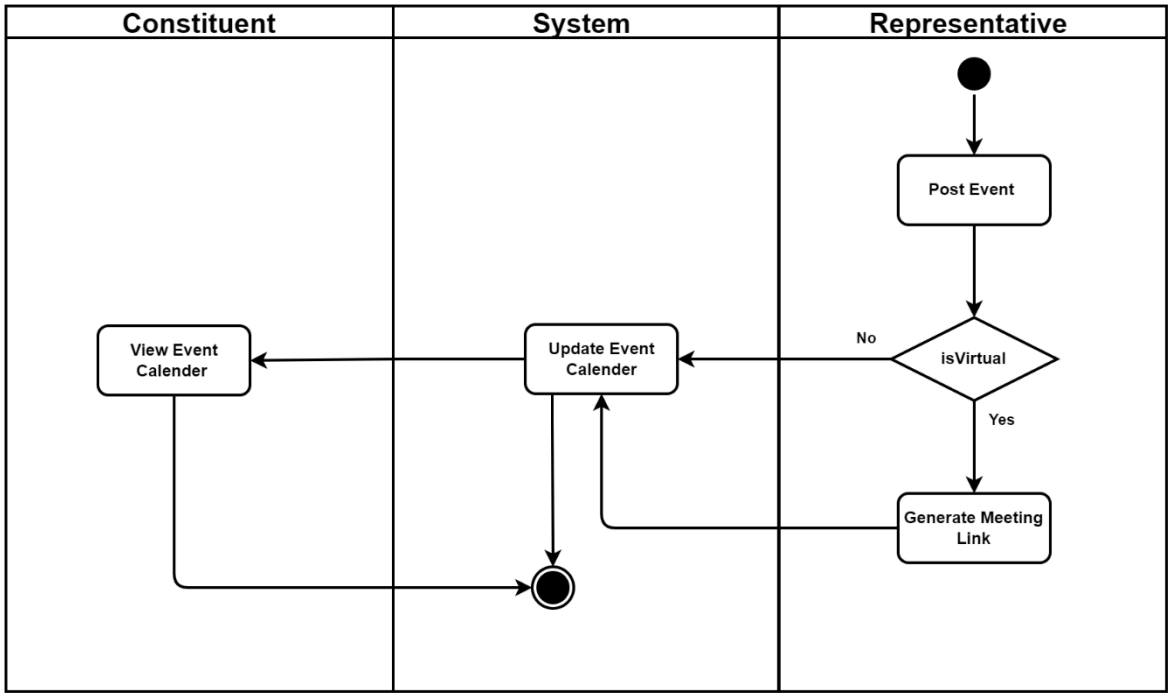

Fig 4.11: Event Calendar Activity Diagram

Survey Activity Diagram:

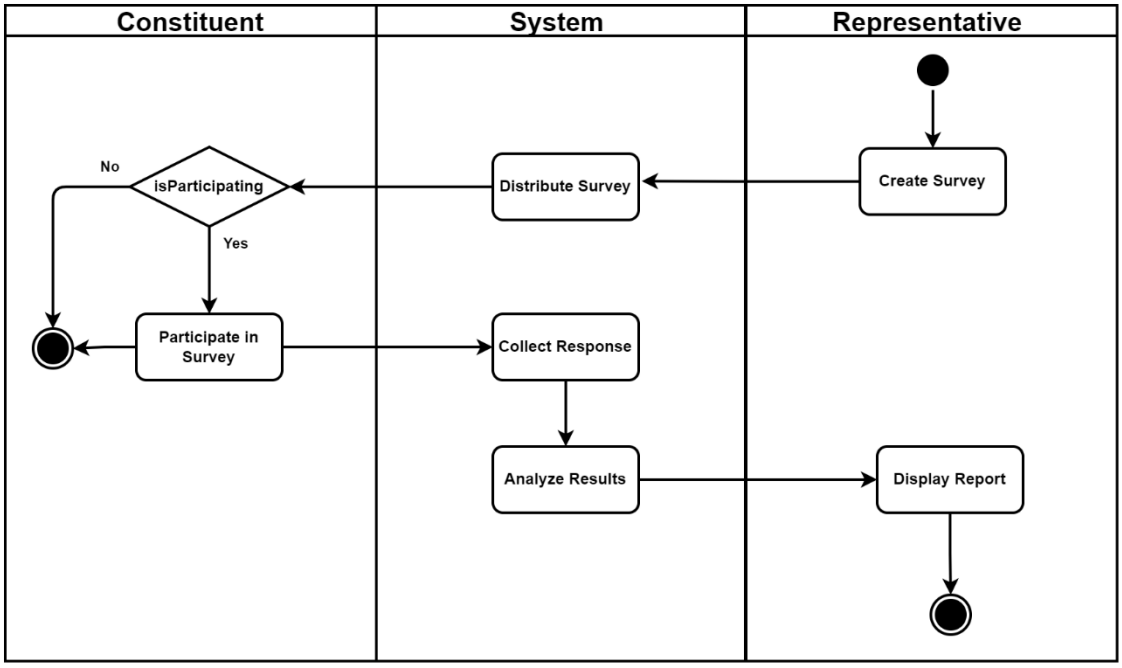

Fig 4.12: Survey Activity Diagram

4.2.2.2. State Machine Diagrams:

This diagram models the behavior of individual components or objects within the system, depicting their states and transitions between them. It offers a visual representation of how the system responds to events and stimuli, guiding the understanding of its dynamic behavior.

Login – Registration State Machine Diagram:

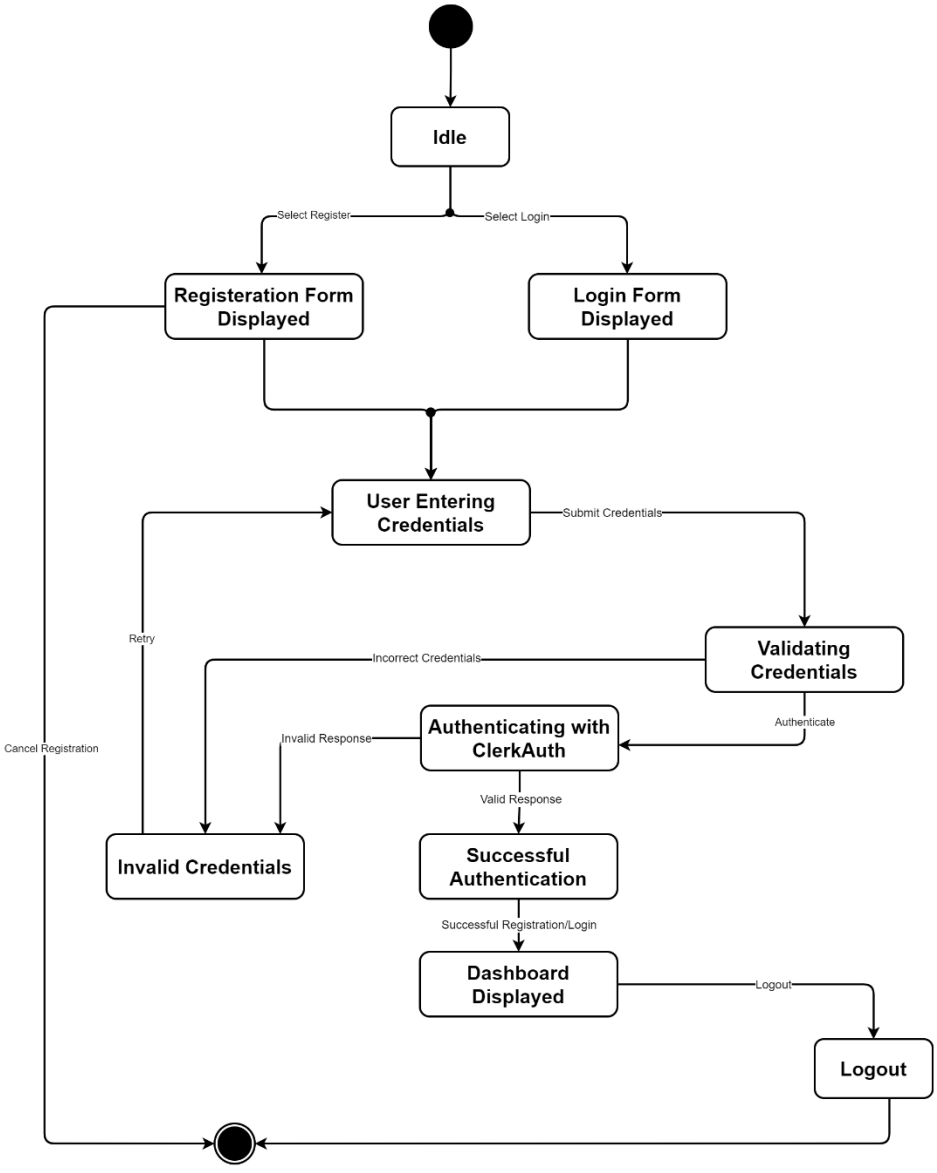

Fig 4.13: Register – Login State Machine Diagram

Complaint Resolution State Machine Diagram:

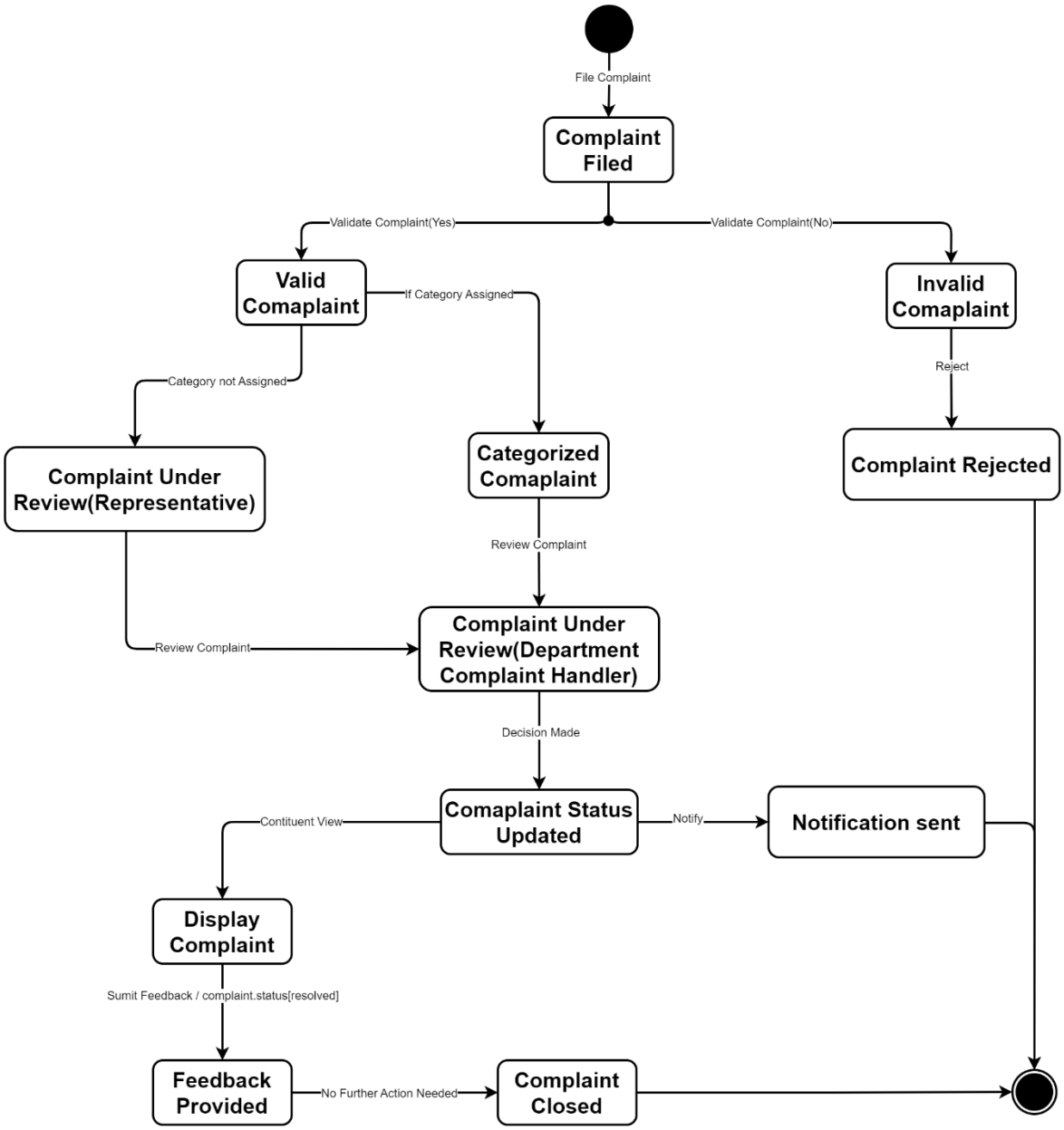

Fig 4.14: Complaint Resolution State Machine Diagram

View Statistics State Machine Diagram

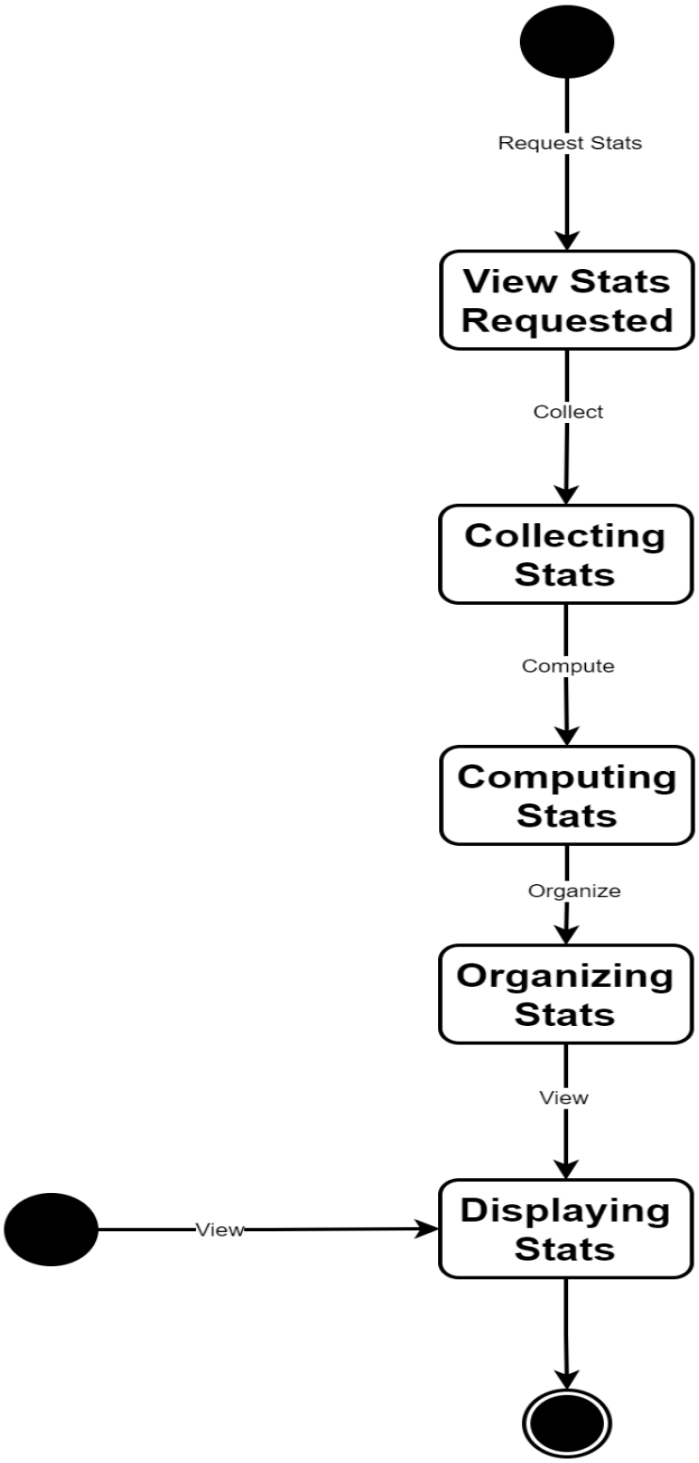

Fig 4.15: View Stats State Machine Diagram

Event Calendar State Machine Diagram:

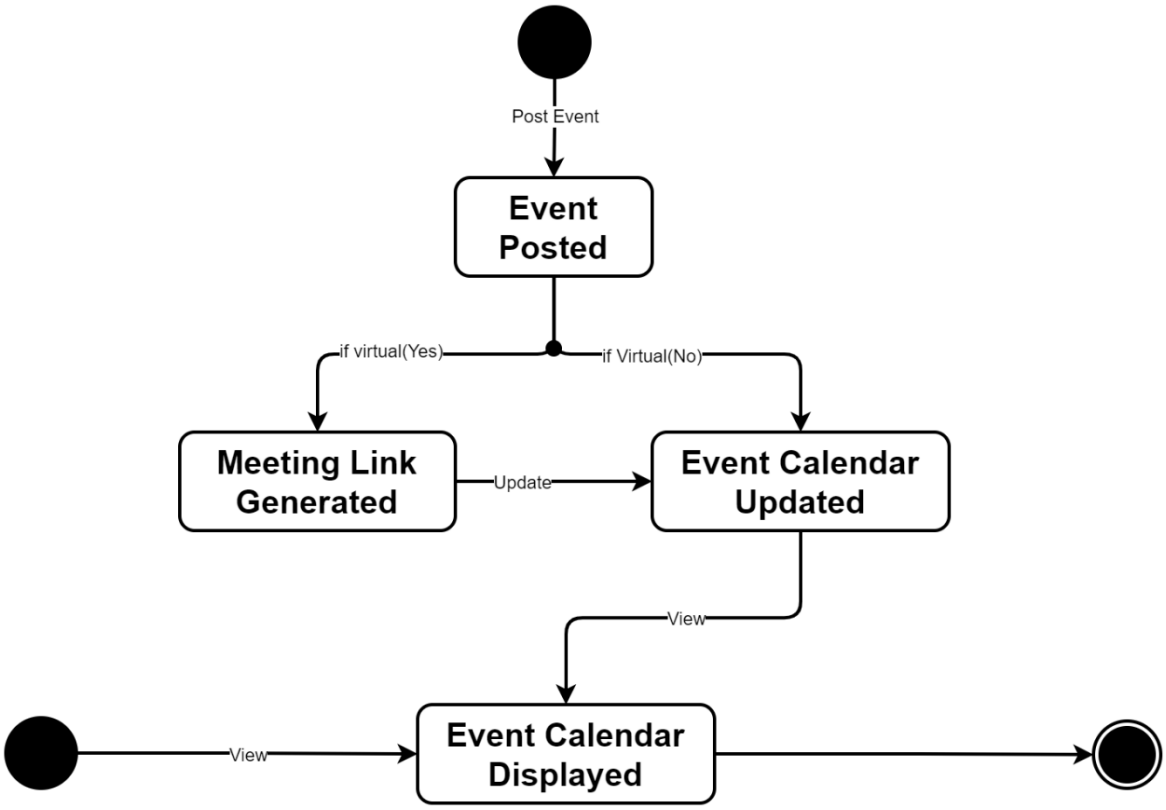

Fig 4.16: Event Calendar State Machine Diagram

Survey State Machine Diagram:

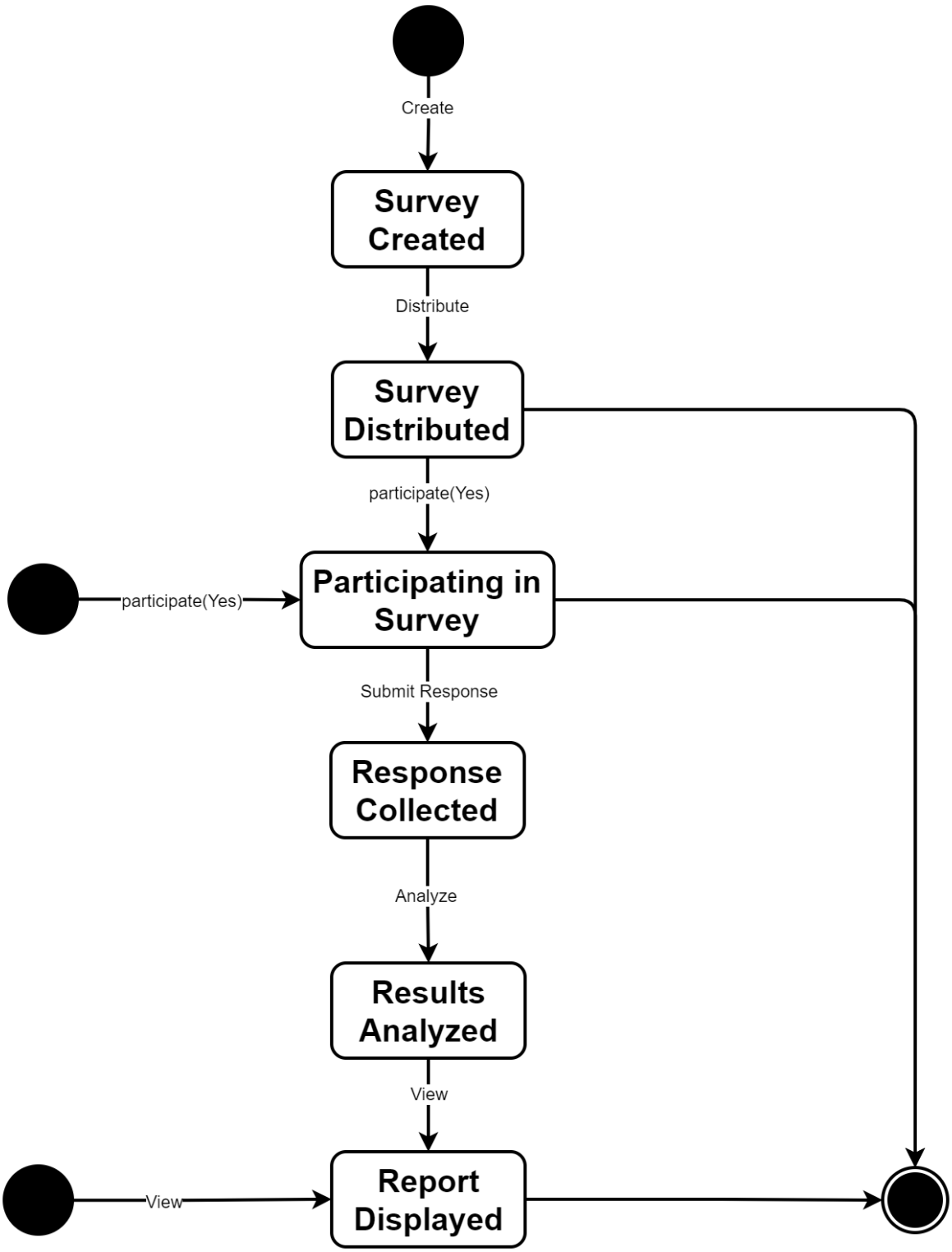

Fig 4.17: Survey State Machine Diagram

4.2.3. UML Interaction-diagrams

These diagrams detail how objects communicate in response to events, focusing on message exchanges between system components.

4.2.3.1. System Sequence Diagrams

It Shows how the system interacts with external actors in a sequence of operations.

Submit Complaint System Sequence Diagram

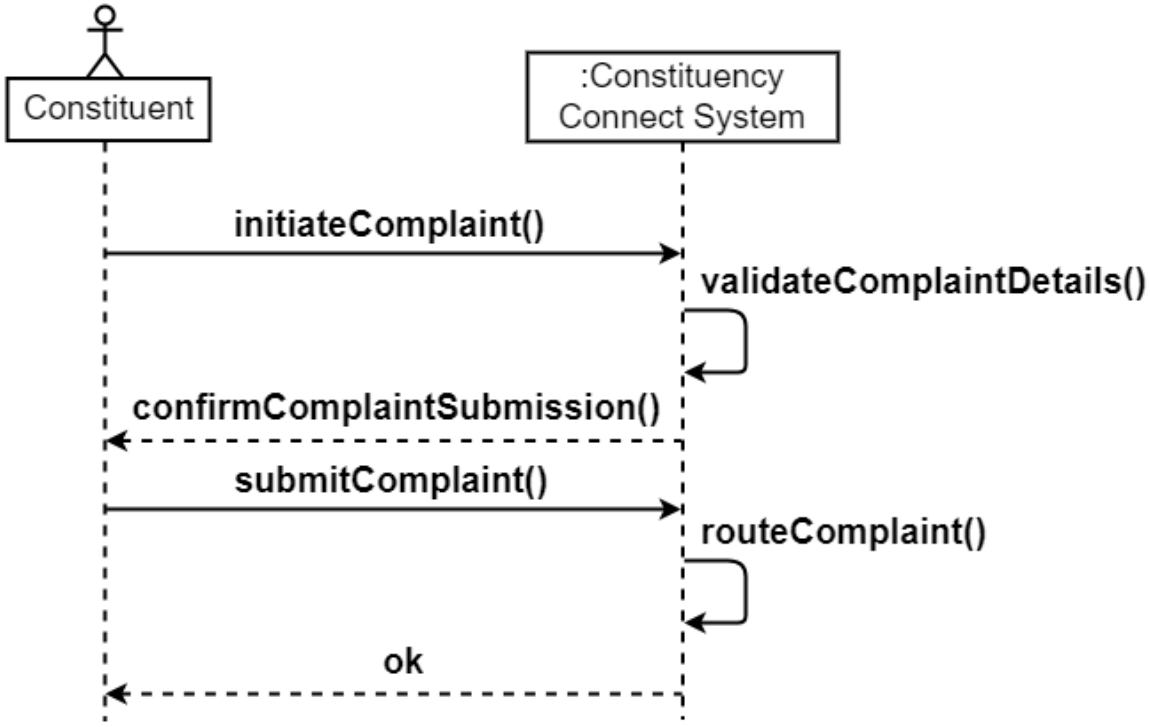

Fig 4.18: Submit Complaint System Sequence Diagram

Track Complaint Status System Sequence Diagram

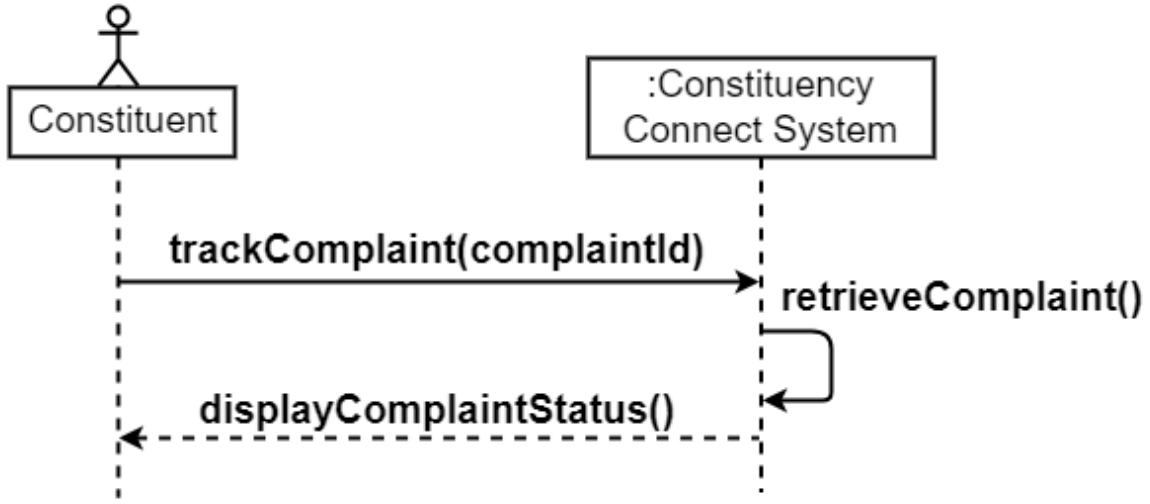

Fig 4.19: Track Complaint Status System Sequence Diagram

Track Complaint History System Sequence Diagram

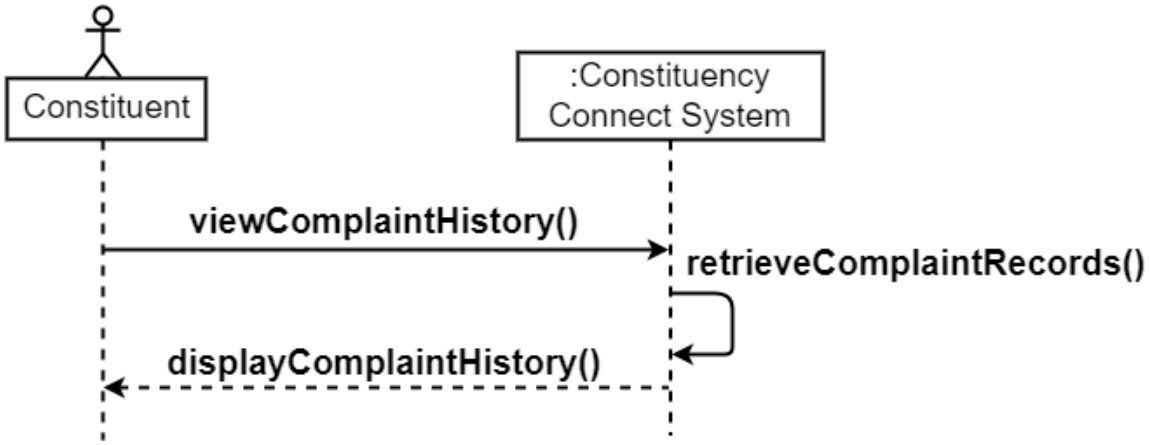

Fig 4.20: Track Complaint History System Sequence Diagram

View Representative Stats System Sequence Diagram

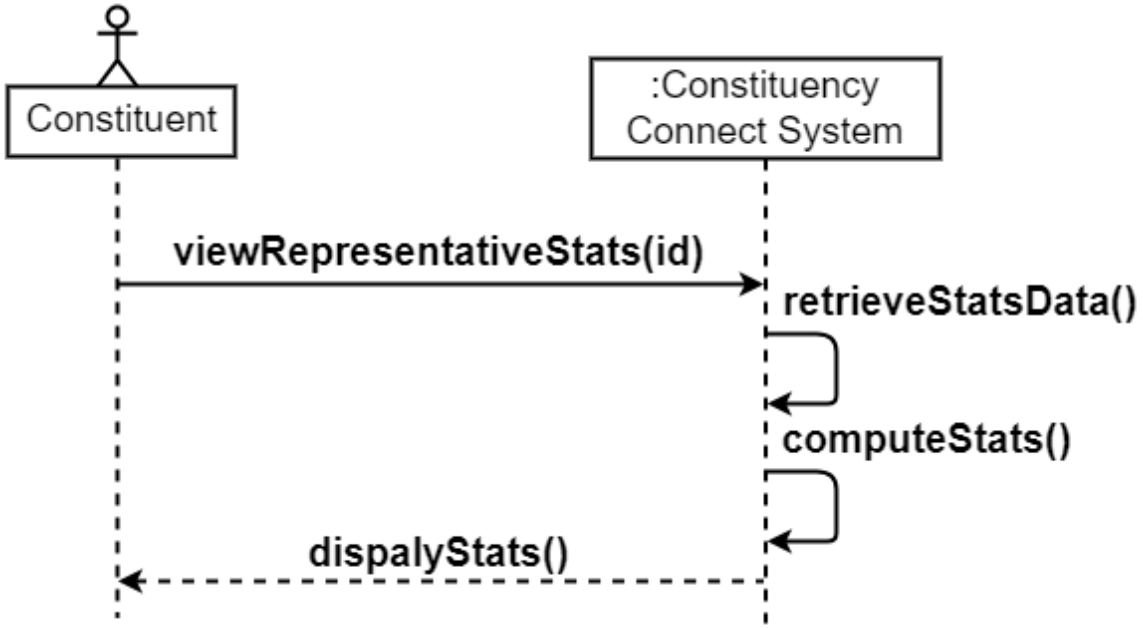

Fig 4.21: View Representative Stats System Sequence Diagram

Survey Participation System Sequence Diagram

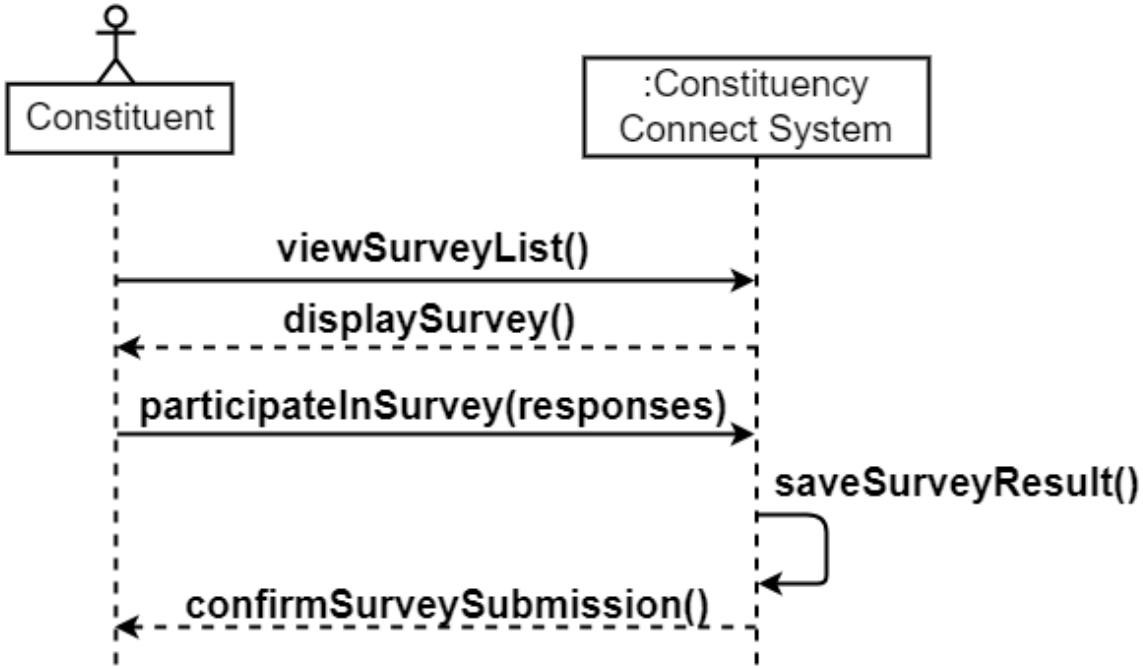

Fig 4.22: Survey Participation System Sequence Diagram

Provide Feedback System Sequence Diagram

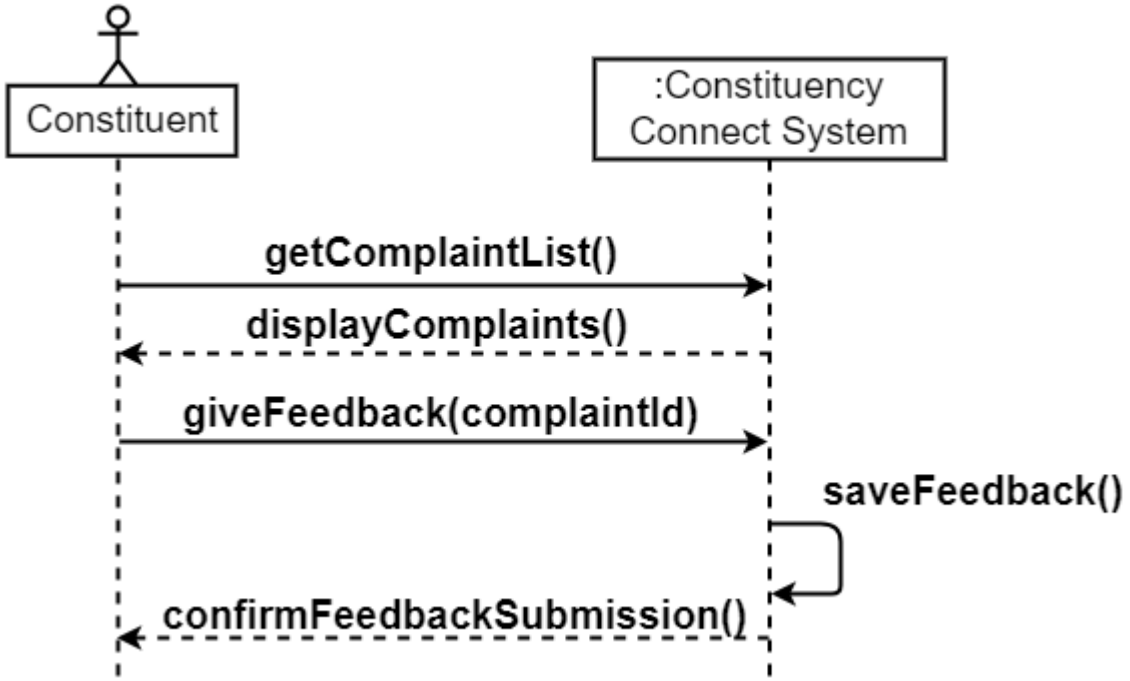

Fig 4.23: Provide Feedback System Sequence Diagram

Display Campaign System Sequence Diagram

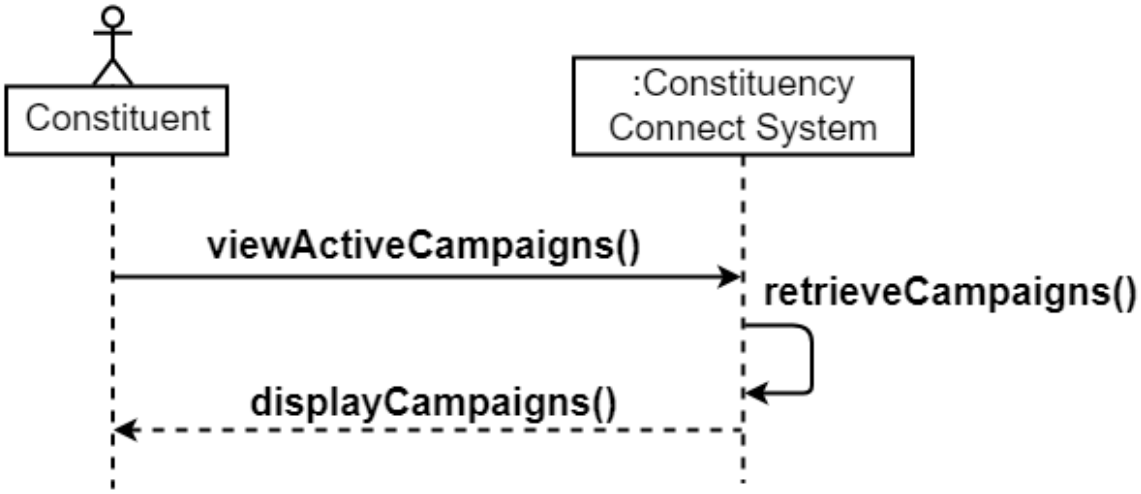

Fig 4.24: Display Campaign System Sequence Diagram

Display Event Calendar System Sequence Diagram

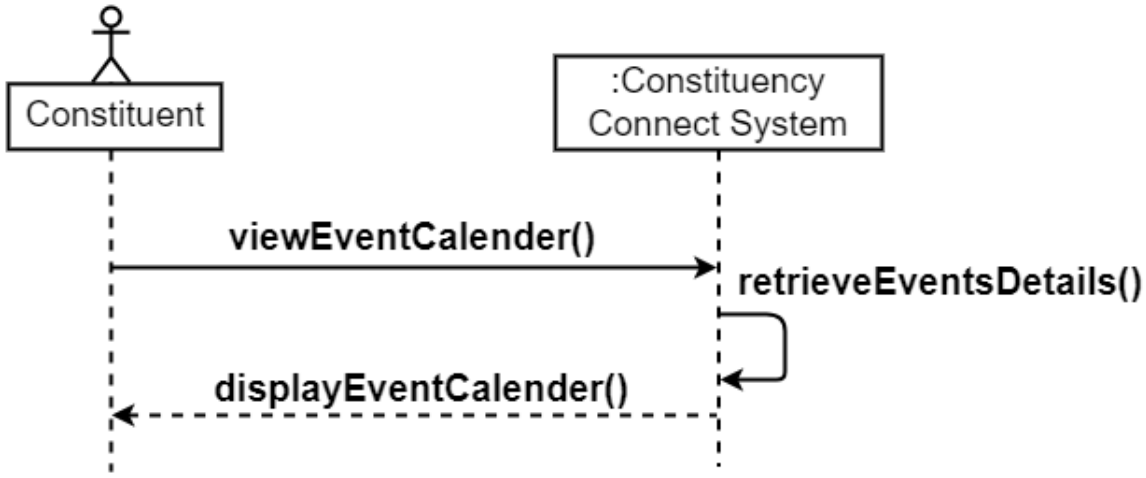

Fig 4.25: Display Event Calendar System Sequence Diagram

Join Virtual Meetup System Sequence Diagram

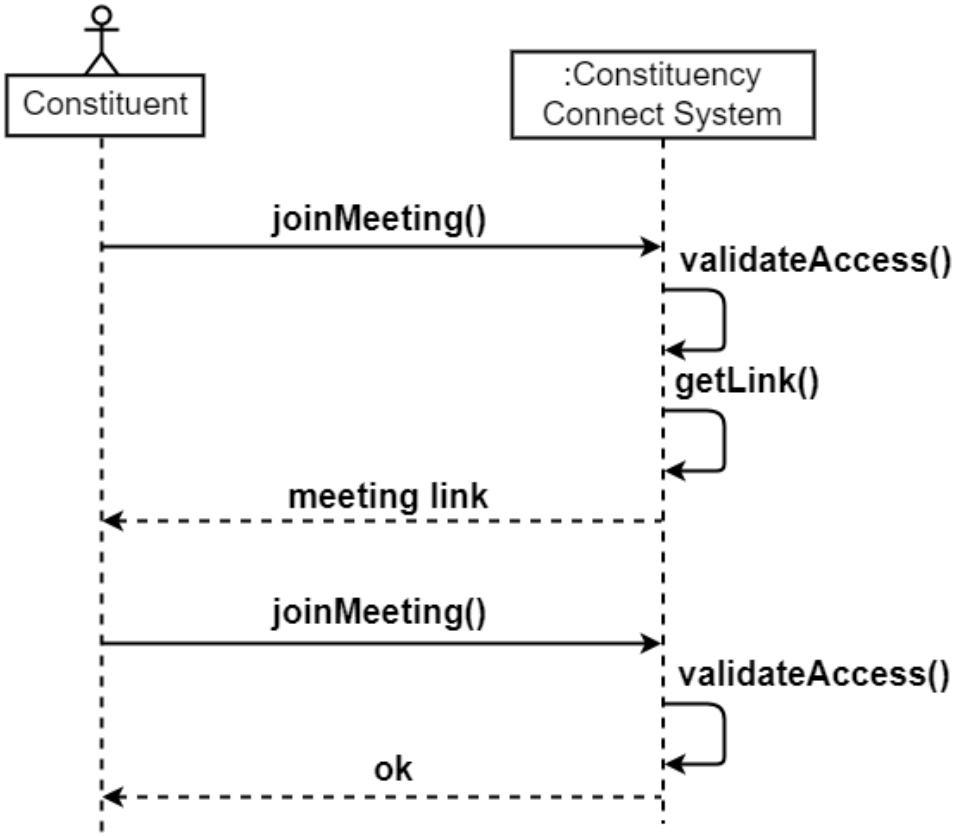

Fig 4.26: Join Virtual Meetup System Sequence Diagram

Create Survey System Sequence Diagram

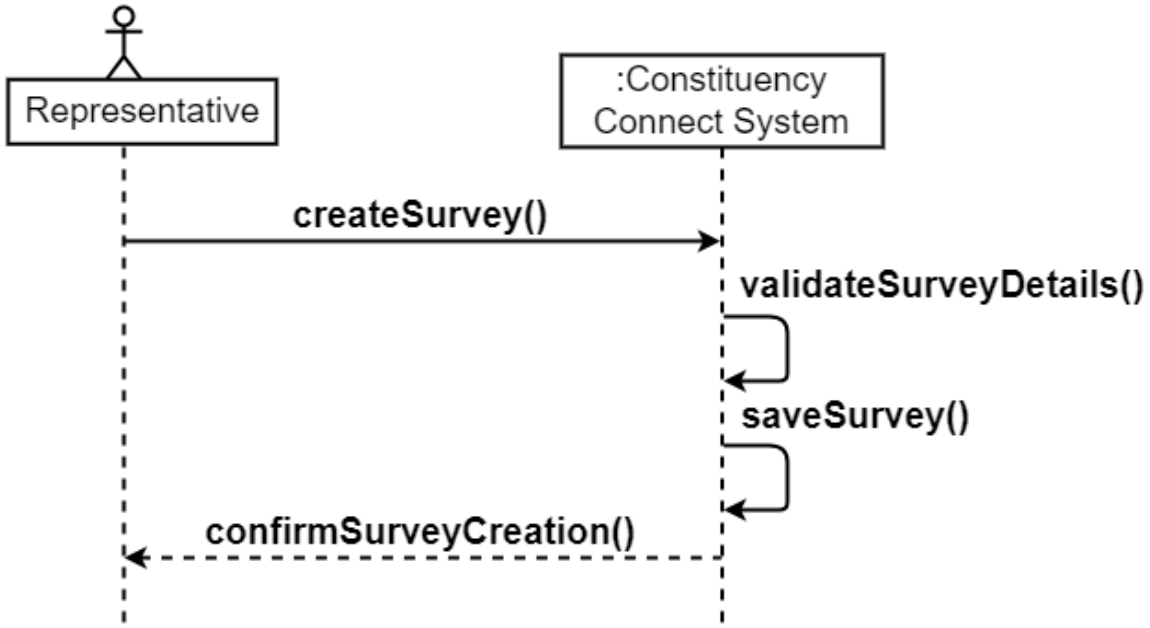

Fig 4.27: Create Survey System Sequence Diagram

Compute Survey Result System Sequence Diagram

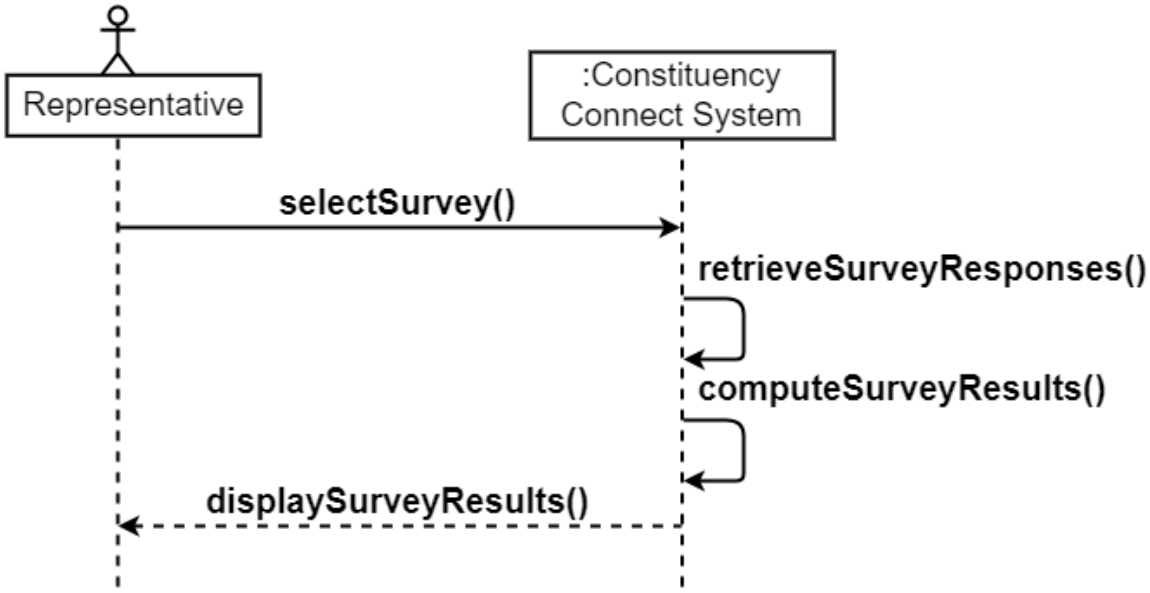

Fig 4.28: Compute Survey Result System Sequence Diagram

Post Event Calendar System Sequence Diagram

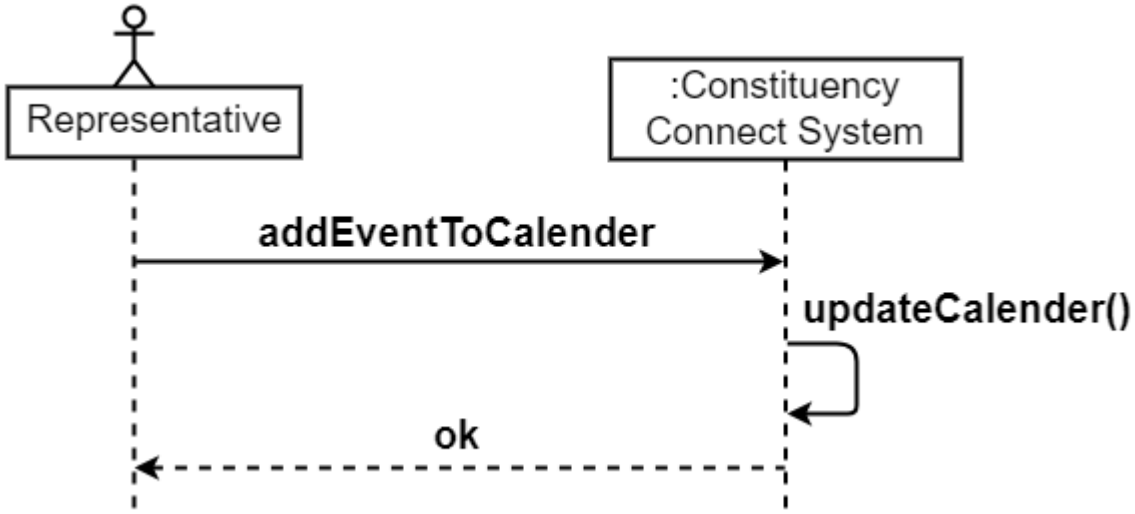

Fig 4.29: Post Event Calendar System Sequence Diagram

Post Awareness Campaign System Sequence Diagram

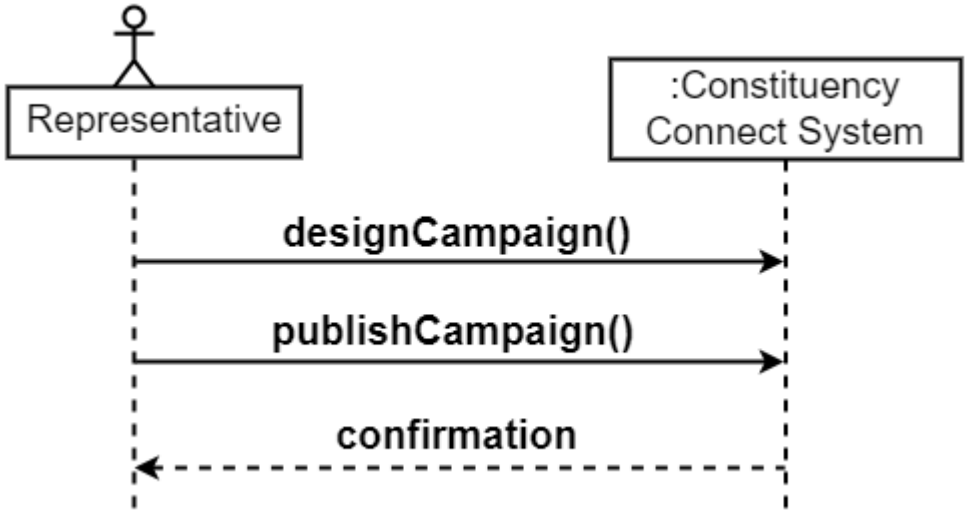

Fig 4.30: Post Awareness Campaign System Sequence Diagram

View Complaint System Sequence Diagram

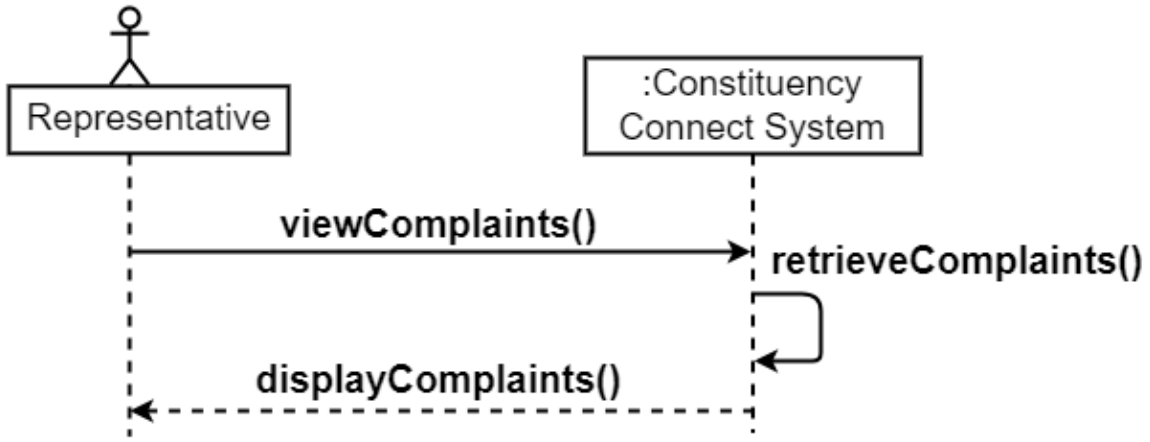

Fig 4.31: View Complaint System Sequence Diagram

Update Complaint Status System Sequence Diagram

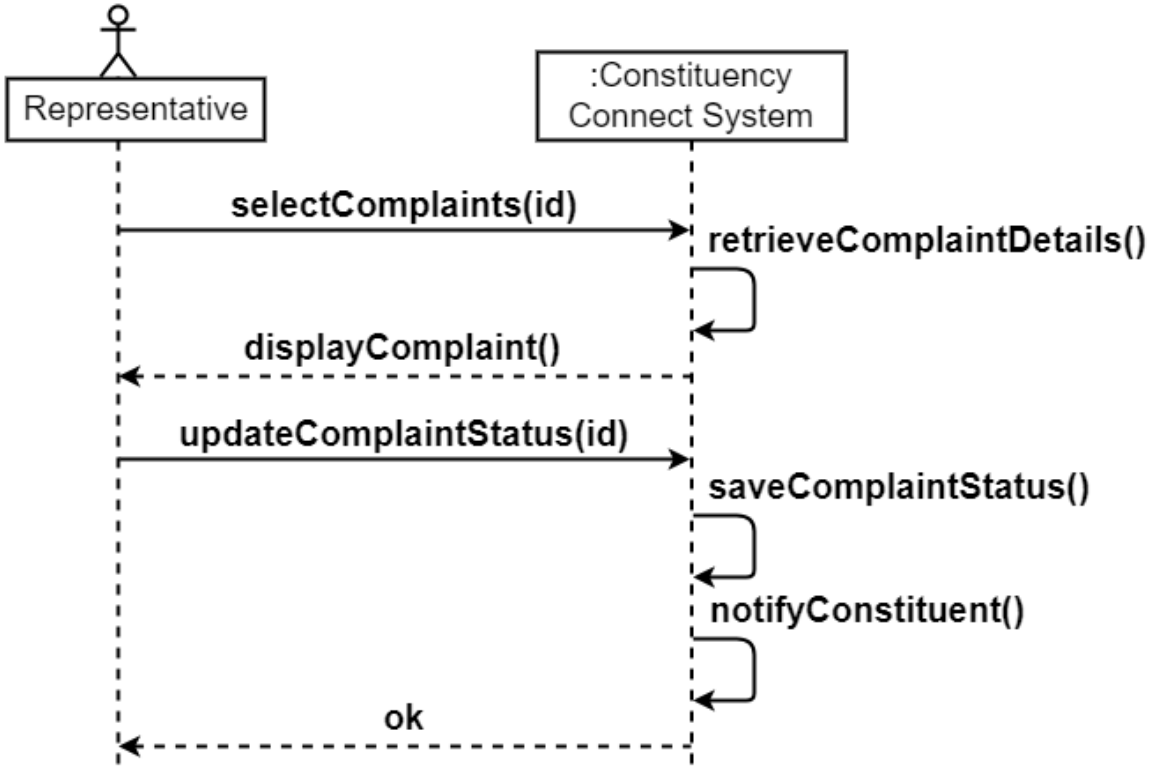

Fig 4.32: Update Complaint Status System Sequence Diagram

Arrange Virtual Meetup System Sequence Diagram

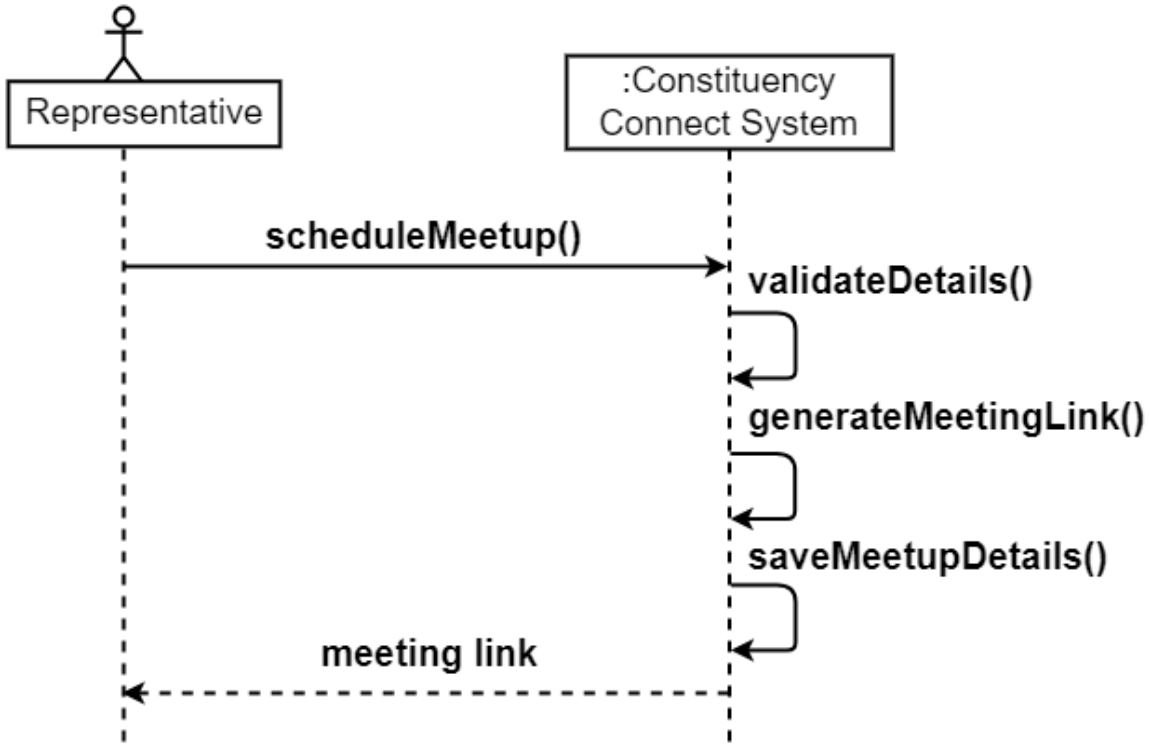

Fig 4.33: Arrange Virtual Meetup System Sequence Diagram

4.2.3.2. Sequence Diagrams

It represents the step-by-step flow of interactions between objects in a sequence**.**

Signup Sequence Diagram

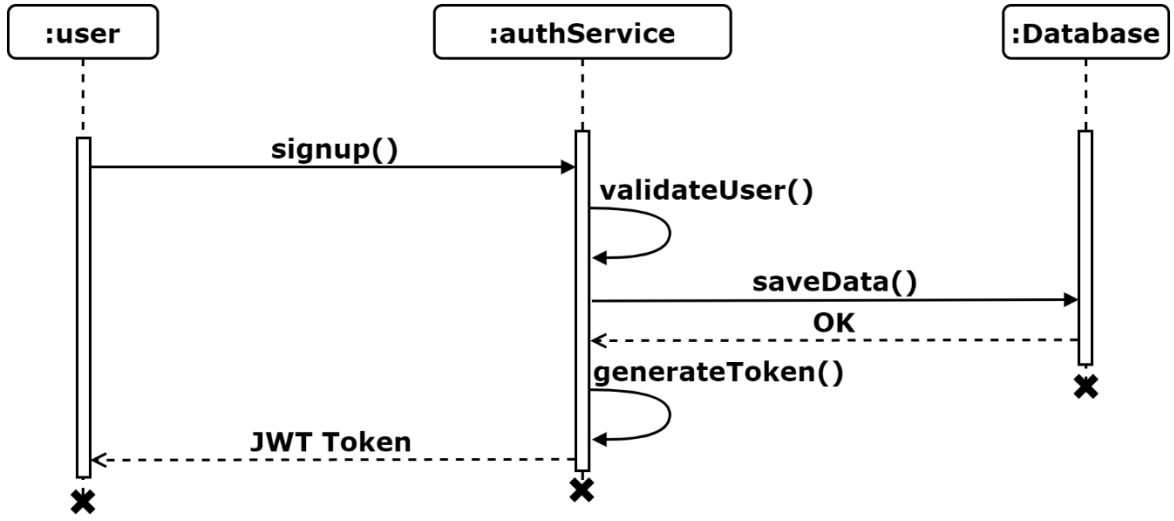

Fig 4.34: Signup Sequence Diagram

Login Sequence Diagram

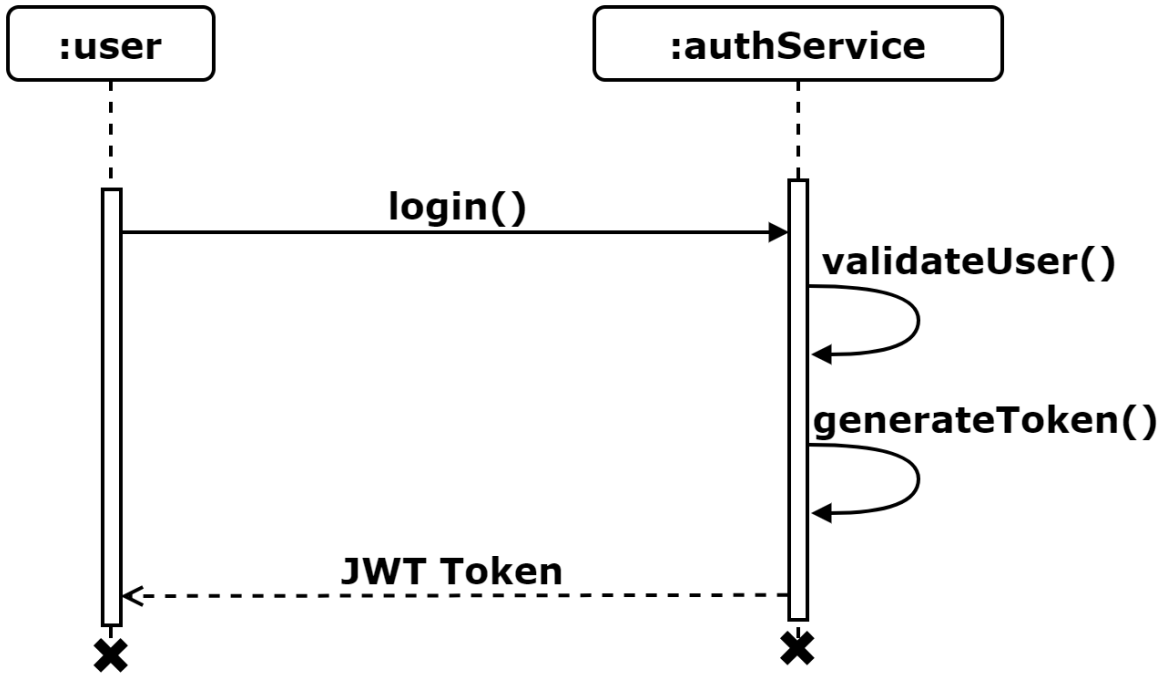

Fig 4.35: Login Sequence Diagram

Complaint Resolution Sequence Diagram

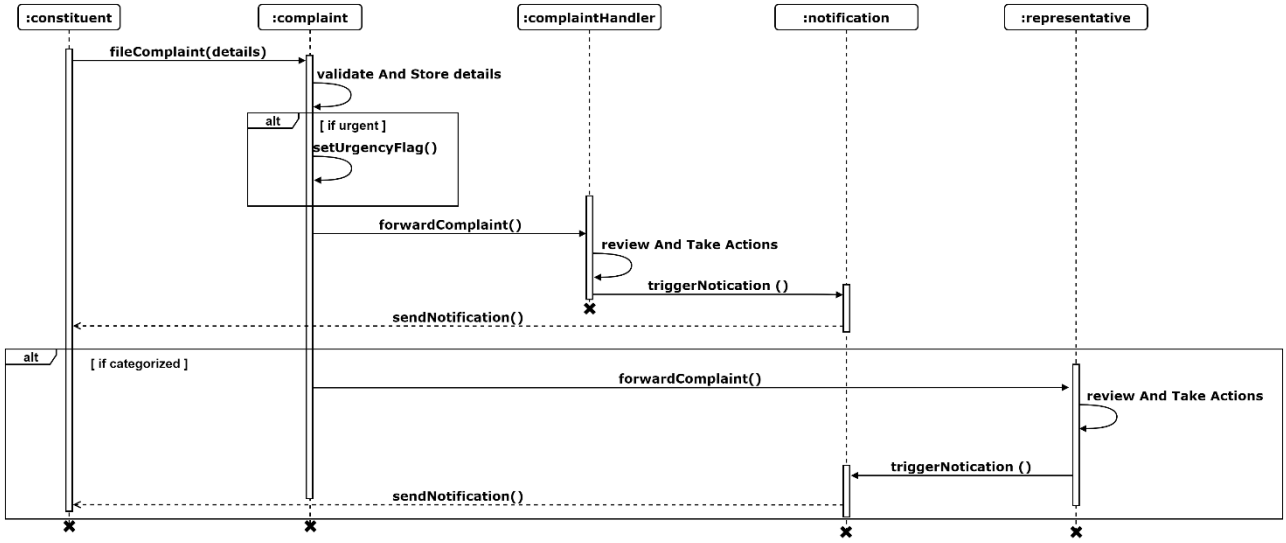

Fig 4.36: Complaint Resolution Sequence Diagram

Submit Feedback Sequence Diagram

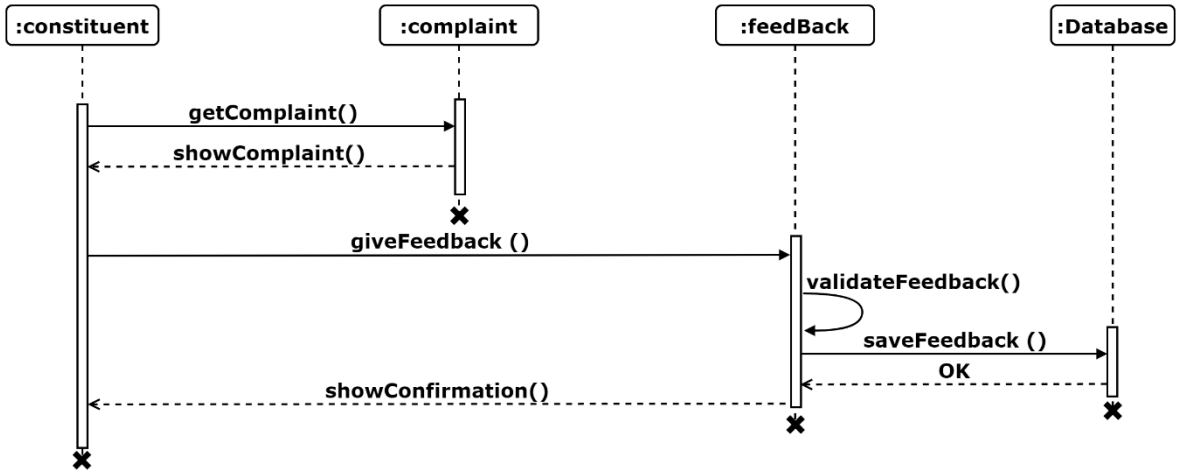

Fig 4.37: Submit Feedback Sequence Diagram

Post an Event Sequence Diagram

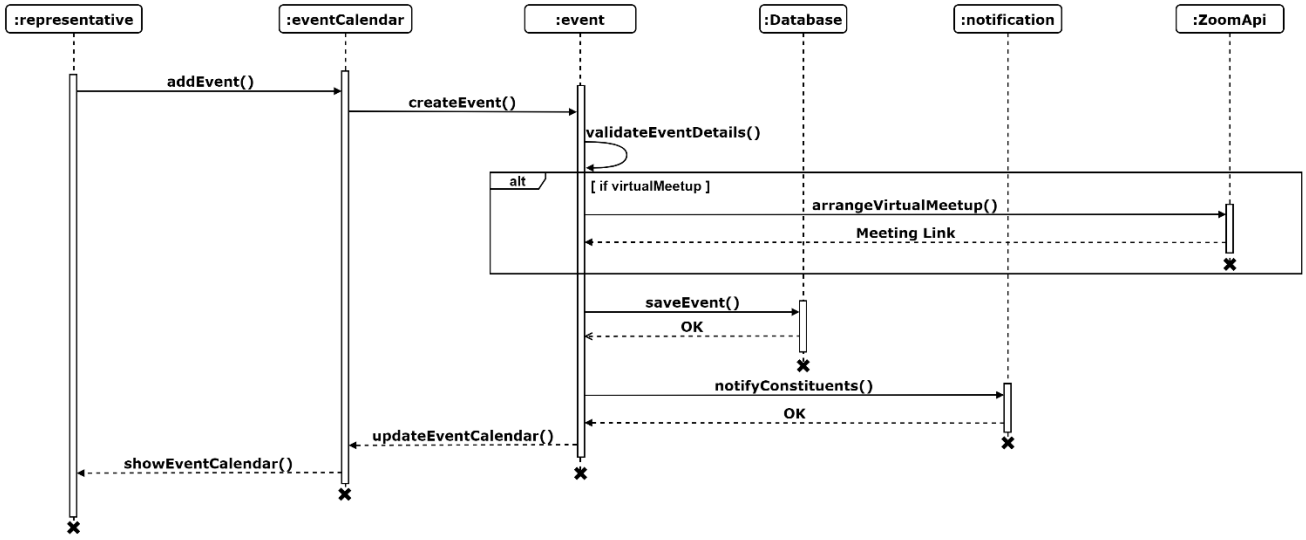

Fig 4.38: Post an Event Sequence Diagram

Join Virtual Meetup Sequence Diagram

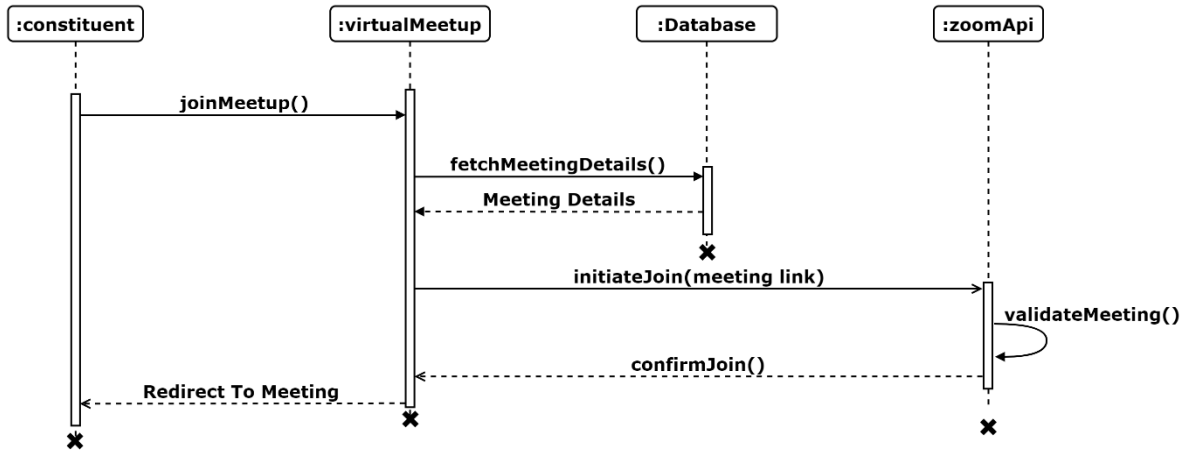

Fig 4.39: Join Virtual Meetup Sequence Diagram

Post Virtual Meetup Sequence Diagram

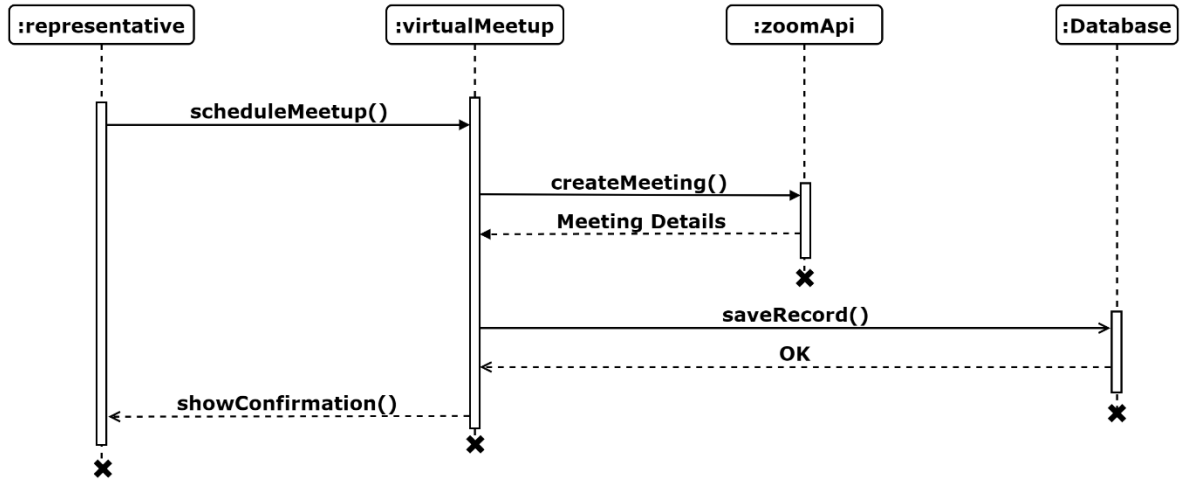

Fig 4.40: Post Virtual Meetup Sequence Diagram

Create Awareness Campaign Sequence Diagram

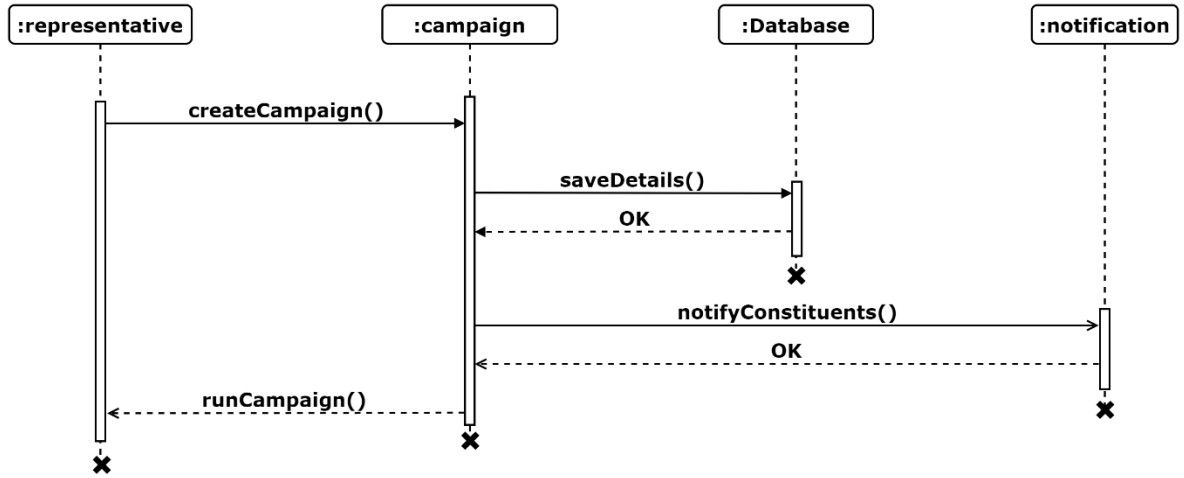

Fig 4.41: Create Awareness Campaign Sequence Diagram

View Notification Sequence Diagram

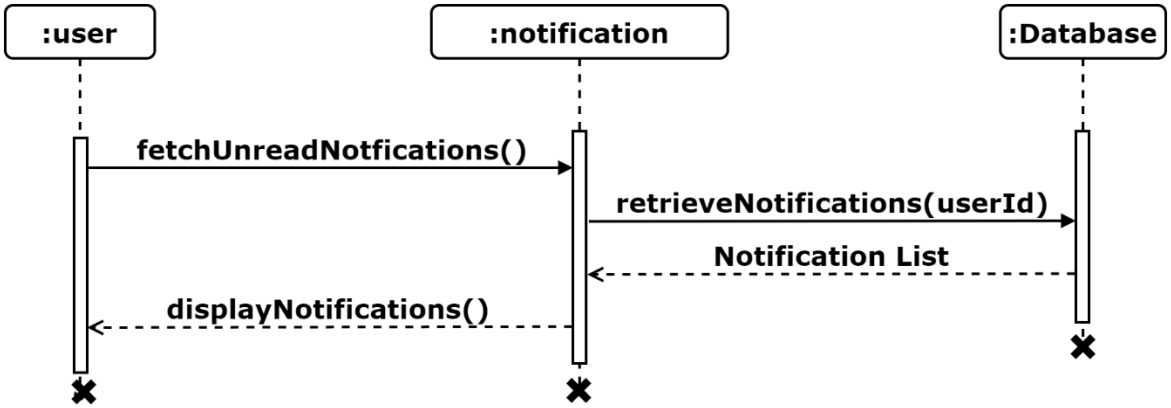

Fig 4.42: View Notification Sequence Diagram

View Stats Sequence Diagram

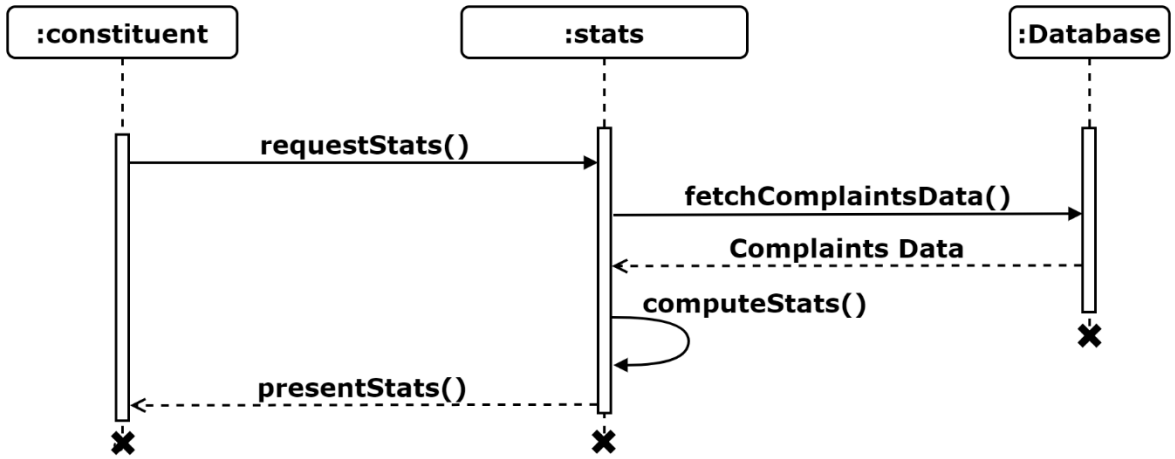

Fig 4.43: View Stats Sequence Diagram

Analyze Survey Sequence Diagram

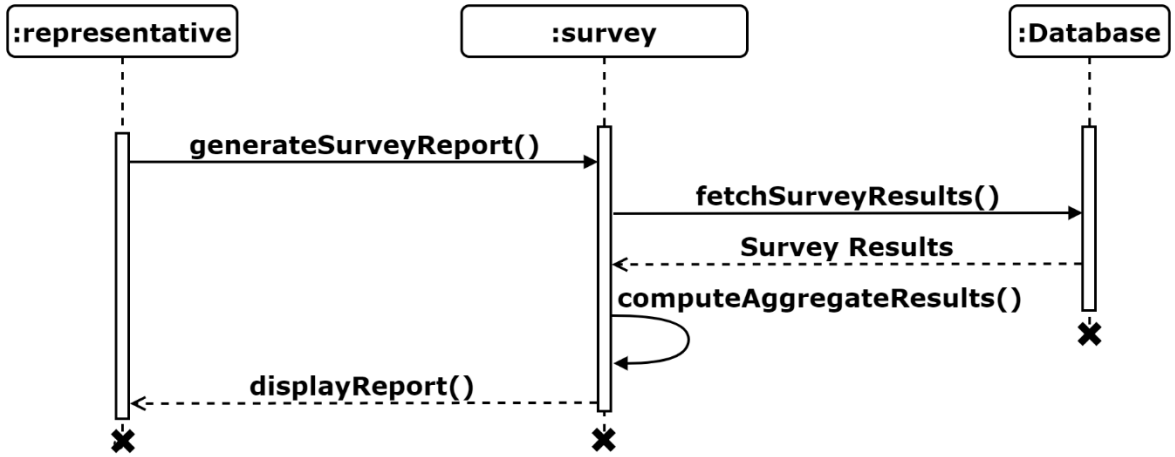

Fig 4.44: Analyze Survey Sequence Diagram

View Complaint Sequence Diagram

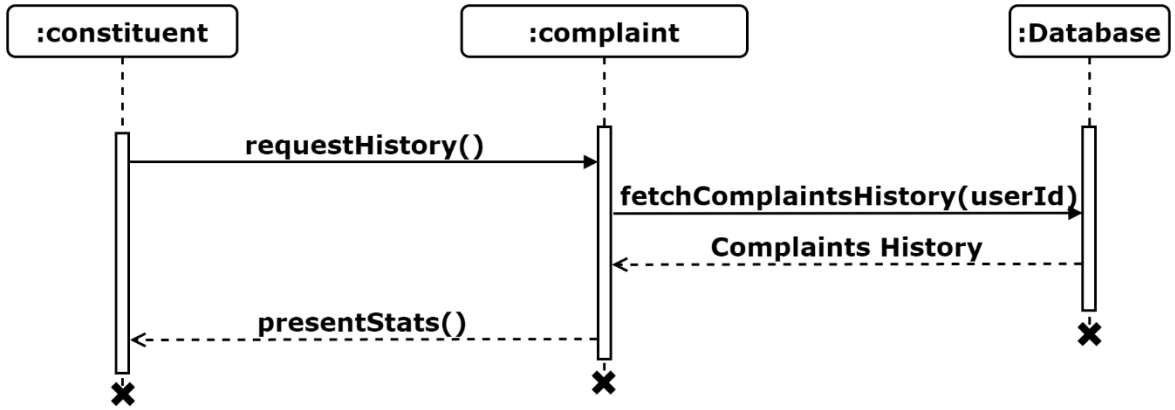

Fig 4.45: View Complaint Sequence Diagram

Survey Participation Sequence Diagram

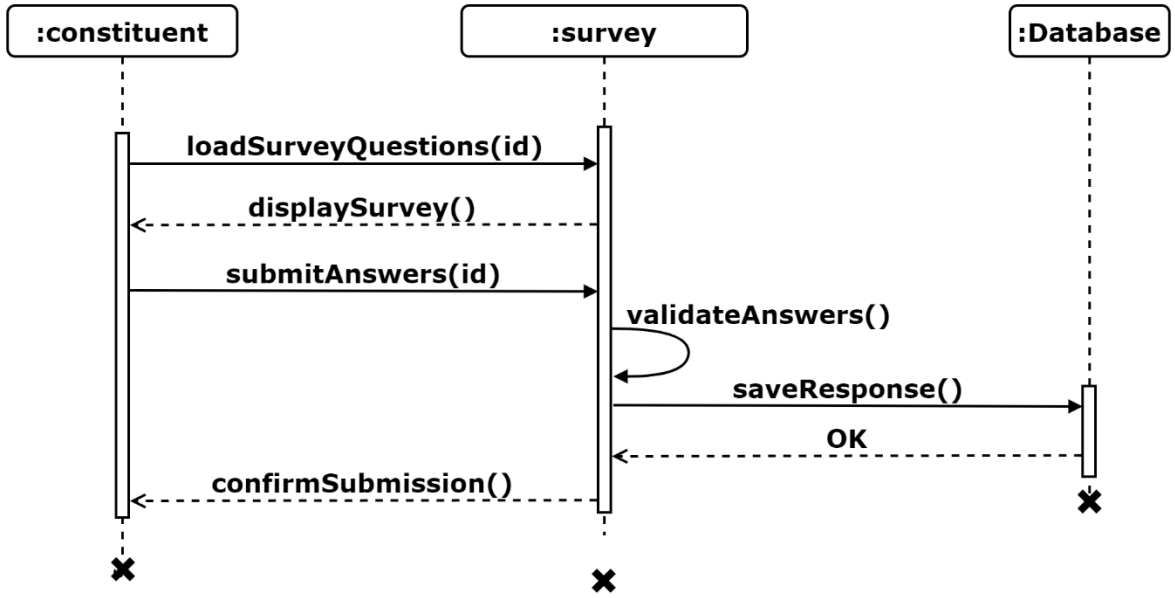

Fig 4.46: Survey Participation Sequence Diagram