Winfra

Software Design Description (SDD)

Version 2.0

Document Number: SDD-002

Project Team Number: B13

Project Team Members: Jenesis Blancaflor (jb7801), Ruthvik Mukkamala (rcm8412), Sanjida Orpi (so2189), Elijah Wilson (esw9386)

REVIEW AND APPROVALS

Team Members	Function	Date	Signature
Jenesis Blancaflor	Project Manager	5/4/25	JB
Sanjida Orpi	Project Manager	5/4/25	SO
Elijah Wilson	Project Manager	5/4/25	EW
Ruthvik Mukkamala	Project Manager	5/4/25	RM

REVISION LEVEL

Date	Revision Number	Purpose
3/3/2025	Version 1.0	Initial Release
5/5/2025	Version 2.0	Revised Release

TABLE OF CONTENTS

1. INTRODUCTION

1.1. PURPOSE

The purpose of this document is to specify the software design of the proposed infrastructure improvement application, *Winfra*. This document aims to detail the architecture of the project at each level of integration throughout the life cycle of the project, and is to be referenced by project managers, technical support, and the quality assurance group.

1.2. SCOPE

The *Winfra* system is entirely delivered as software; in particular, the system is experienced as a public web application. The system will include the following functions:

- Infrastructural suggestion
- Community connection
- Accessibility promotion
- Interactive map interface
- Location authentication
- CTA

The platform will allow users to post reports of issues in real time through a simple location pin-drop feature on an interactive map interface. Each post will provide a detailed report of an issue along with the citizen's recommendations and exigencies for improvement.

Users will be able to view reported problems within their area and engage in discussions to build community awareness. The system will enable direct outreach to local government representatives, which will streamline efforts to address infrastructure needs in underrepresented areas.

The system will be implemented as a web application to be supported on desktop and mobile environments, to support time sensitive reports regarding infrastructure. The system will centralize infrastructural reports onto a single, accessible platform that enables users to track issues in their communities.

1.3. IDENTIFICATION

Winfra Software Design Description SDD-001 Version 1.0

1.4. DOCUMENT SUMMARY

The motivation for this document is to involve stakeholders in the design of the *Winfra* system in order to demonstrate that the project requirements are reflected in the system design. Thus the intended audience of the SDD includes all stakeholders, clients, and project developers.

1.5. SYSTEM OVERVIEW

The Winfra system intends to improve New Yorkers' satisfaction with infrastructure and initiate infrastructural development to enhance the quality of living in NYC. The web-based system aims to build an interactive network for citizens to report damaged or ineffective infrastructure, which will facilitate development and improvement where it is urgently needed. The secondary purpose of the system is to aggregate firsthand civilian testimonies in a way that allows them to be interpreted verbally, graphically, or numerically.

1.6. DOCUMENT OVERVIEW

Introduction. Preliminary information regarding the motivation of the project and the purpose of the software design description.

Reference Documents. List of previous releases that are referenced in the body of the software design description.

System-Wide Design Decisions. Descriptions of the software components of the *Winfra* project.

Software Item Detailed Design. Descriptions of the software items of the *Winfra* project.

Deployment Architecture. Descriptions of the hardware design including the user interface, application server, and database.

Dictionaries. Descriptions of the classes of the *Winfra* system and their attributes and methods.

Requirements Traceability. Depiction of how design components are backwards-traceable to requirements and forwards-traceable to modules of code.

System Design Testing. Descriptions of quality assurance testing plans including peer review, self-checks, walkthroughs, inspection, product testing, and acceptance testing.

Appendices. Selected tables and diagrams for context.

2. REFERENCE DOCUMENTS

Blancaflor, Jenesis et. al. “Project Information Sheet.” Version 1.0. *Winfra,* 24 Feb. 2025.

Blancaflor, Jenesis et. al. “Project Management Plan.” Version 2.0. *Winfra,* 5 Feb. 2025. [SPMP-002]

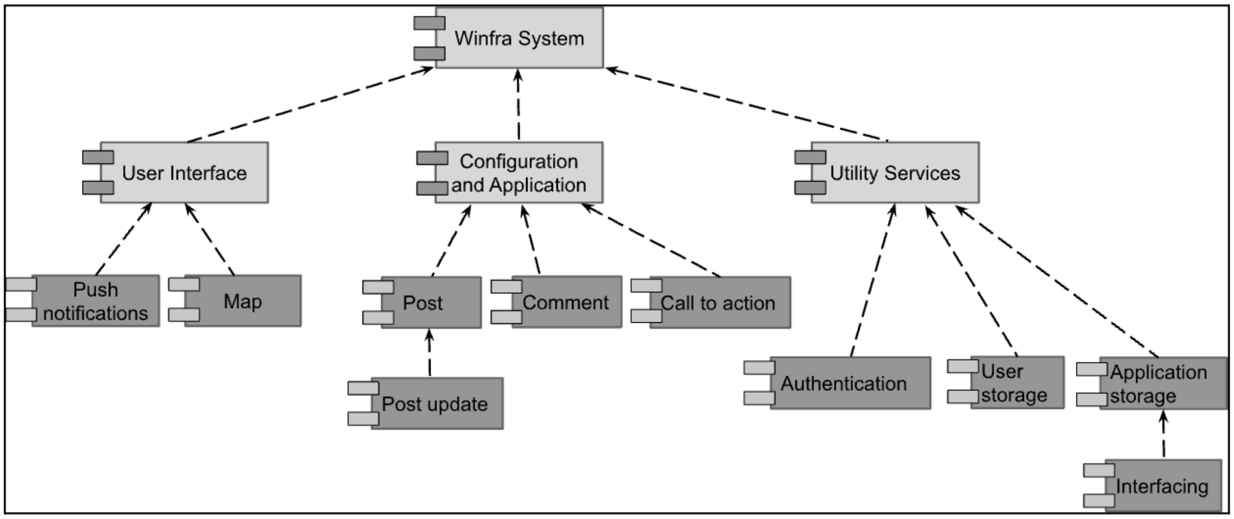
Blancaflor, Jenesis et. al. “Project Proposal.” Version 1.0. *Winfra,* 19 Feb. 2025.

Blancaflor, Jenesis et. al. “System Requirements Specification.” Version 4.0. *Winfra,* 19 Feb. 2025. [SRS-004]

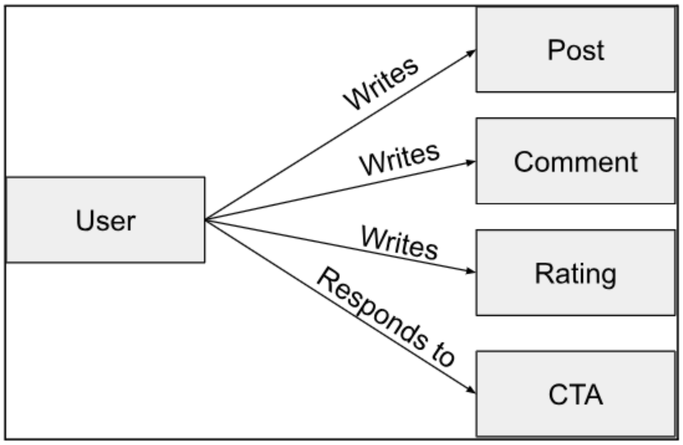
3. SYSTEM-WIDE DESIGN DECISIONS

The functional and nonfunctional requirements of the *Winfra* project along with all use cases and use case diagrams have been maintained since the release of *SRS-004* and can be reviewed in detail in §6 and §8 of that document. In short, the functional requirements are infrastructural suggestion, community building, an interactive map, promotion of accessibility, real-time updates, and an ability and incentive to contact local representatives. The nonfunctional requirements are security, performance, usability, and maintainability. These requirements inform the architectural decisions detailed in this section.

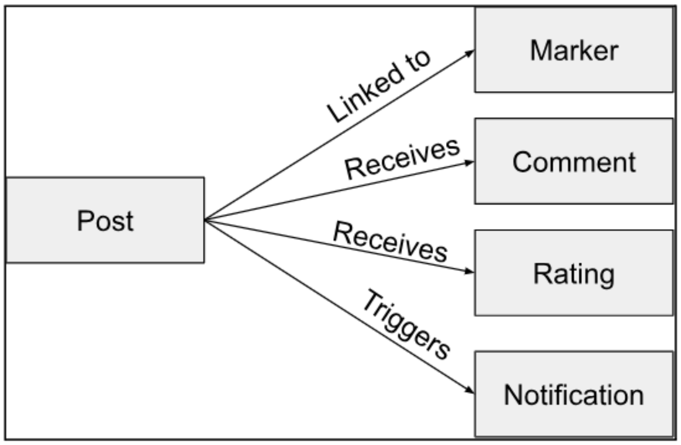
3.1. SOFTWARE COMPONENT ARCHITECTURAL DESIGN



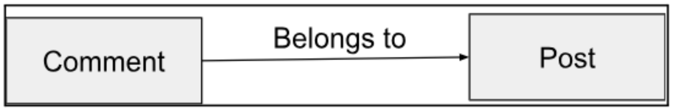
User Component



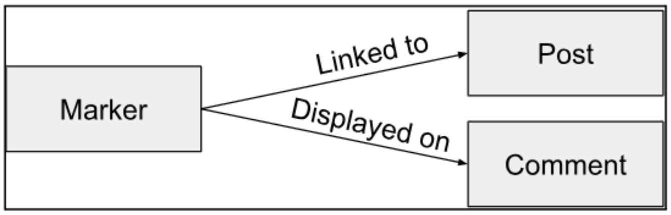
Post Component



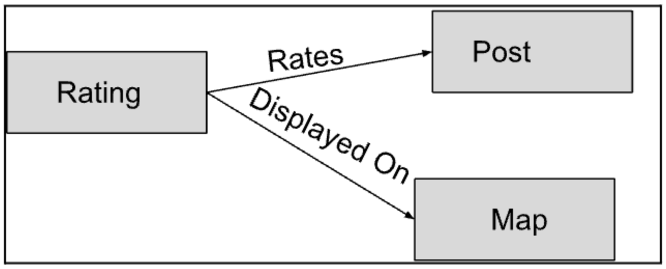
Comment Component



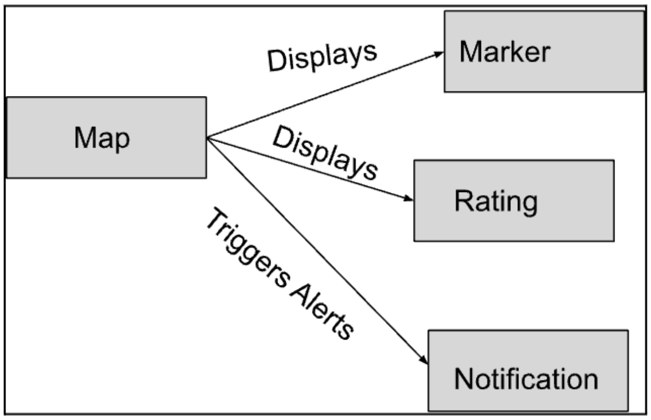
Marker Component



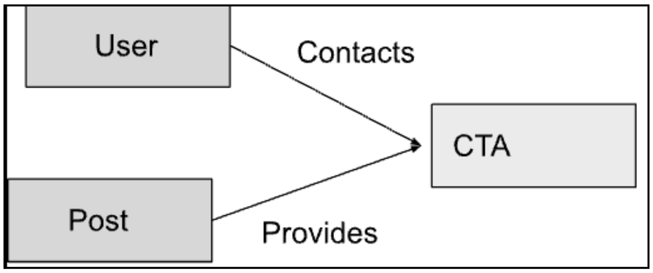
Rating Component



Map Component



Call to Action (CTA) Component



3.2. SOFTWARE ARCHITECTURE GENERAL DESCRIPTION

The subsystems of the system architecture include the user interface, configuration and application, and utility services. The user interface handles the interaction between the end user and graphical interface by displaying the content of the website and making its services accessible. This includes forms to post an issue, view posts, and the interactive map. Another subsystem is the configuration and application layer which processes requests from the end user and accesses the database layer to provide responses to the user. The last subsystem is the utility services associated with the database layer which stores and retrieves user information, user posts, relationships between components, and external data needed for the functionality of the requirements of the application.

3.3. SOFTWARE ITEM COMPONENTS

- The User component manages individual user accounts and authentication. It stores user information, including first name, last name, password, phone number, email, and home address. The User ensures that users can log in, interact with the platform, and contribute to its content and services that rely on user contribution. This component relates to the infrastructural suggestion, community building, and contacting local representatives requirements.
- The Post component enables users to submit infrastructure reports that are marked on the map interface and displayed to users in the area. Each post contains a description, severity rating, author, location, status, accessibility information, comments, and an optional photo attachment. This component facilitates the submission and visibility of reported issues within the community, resting to the infrastructural suggestion, promotion of accessibility, and real-time updates requirements.

- The Comment component allows users to engage with infrastructure reports by adding responses under posts. It consists of user-made text responses that connect to posts where users can ask questions, give updates, feedback, reactions, and engage in real-time discussions. This component relates to the community building and real-time updates requirements.
- The Marker component represents the geographical location of reported infrastructure concerns and is displayed on the map interface. It stores latitude and longitude data and links a user's post to a specific point on the interactive map. Markers help visualize reported issues and assist users in navigating to affected locations. This component relates to the interactive map and real-time updates.
- The Rating component allows users to rate and categorize infrastructure accessibility. This includes a dropdown option for users to assess the accessibility of an infrastructure concern, which is then displayed on the corresponding post. This feature helps highlight accessibility issues if it is prominent at the posted location. This component relates to the promotion of accessibility and interactive map requirements.
- The Map component is an interactive, real-time interface that displays infrastructure issues using markers and icons categorized by severity and type. It provides filtering options, allowing users to sort posts by issue type, severity, and location. This component ensures that infrastructure problems are visualized

effectively for easy navigation and awareness. This component relates to the interactive map and real-time update requirement.

- The Call to Action (CTA) component enables users to connect with government officials responsible for infrastructure improvements. This feature enhances accountability by facilitating direct communication between the community and local authorities and fulfills the call to action requirement.

3.4. COMPONENT INTERFACE IDENTIFICATION

Interface Name	**Description**	**Component 1**	**Component 2**
User Authentication API	Manages user login, logout, and session authentication.	Frontend UI	Backend API
Post Submission API	Allows users to submit infrastructure issue reports.	Frontend API	Backend API
Comment API	Enables users to comment on infrastructure issue posts.	Frontend API	Backend API
Map Data API	Retrieves infrastructure issue locations and displays them on an interactive map.	Frontend API	Google Maps API

Interface Name	Description	Component 1	Component 2
User Data Access	Manages user-related data retrieval and authentication details.	Backend API	PostgreSQL Database
Post Management API	Handles storage, retrieval, and updates of infrastructure issue posts.	Backend API	PostgreSQL Database
Comment Management API	Stores and retrieves comments associated with infrastructure posts.	Backend API	PostgreSQL Database
Geolocation API	Fetches geolocation data related to reported infrastructure issues.	Backend API	Google Maps API
Local Representative Lookup	Provides contact details of local government representatives responsible for infrastructure.	Backend API	External Representative Database
Infrastructure Marker API	Displays reported infrastructure issues with location markers and severity levels.	Google Maps API	PostgreSQL Database

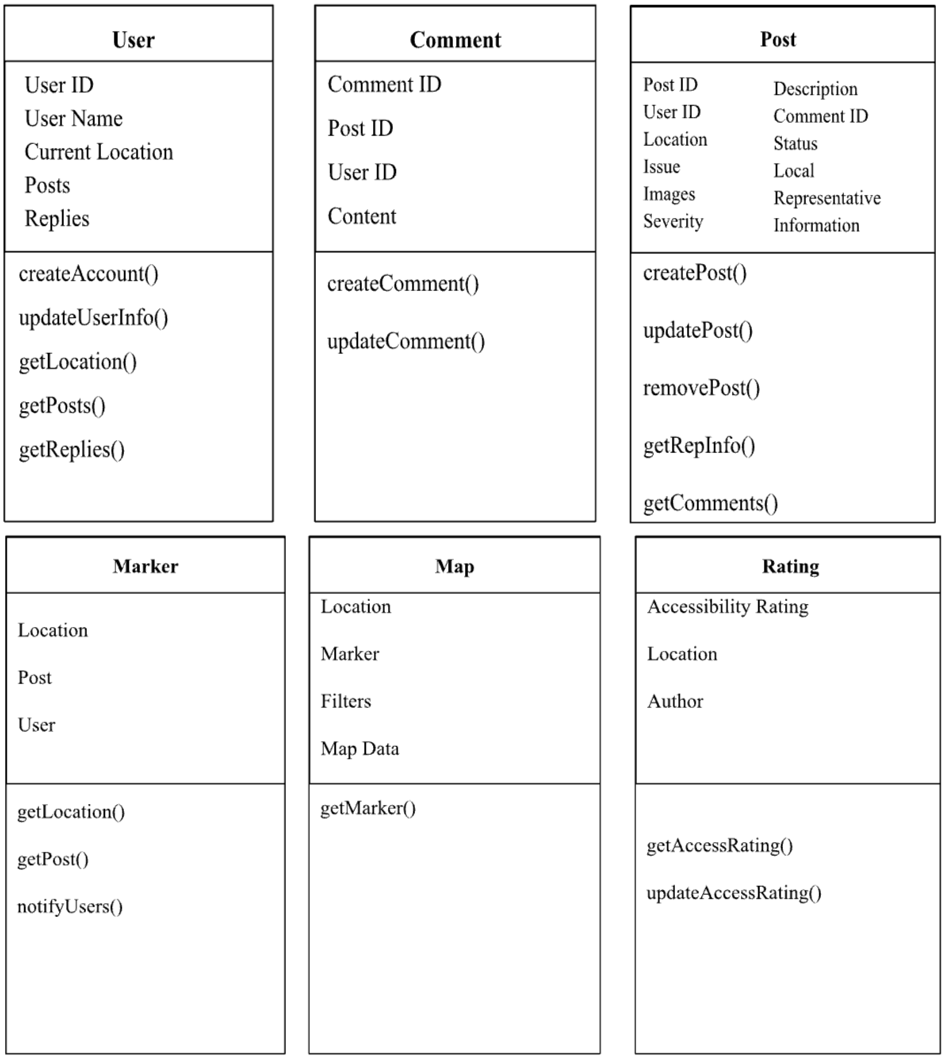
3.5. SOFTWARE COMPONENT CONCEPT OF EXECUTION

The flow of execution of the system is as follows: the database is launched, then the server is launched, and a connection is established between the two. Next, the user interface is launched, and the connection between the UI and the map API is established. The map API is also connected to the database in order to accurately select users' home locations. . The user interface enters execution when a user interacts with the system by selecting buttons such as "Post," "Comment," or "Rate," prompting requests to the backend. The backend API is triggered when it receives a request from the UI, processes data, and communicates with the database or external services, returning a response to the UI. The database server executes queries upon API calls, retrieving, storing, or updating data, ensuring persistence and consistency of infrastructure reports, comments, and ratings.

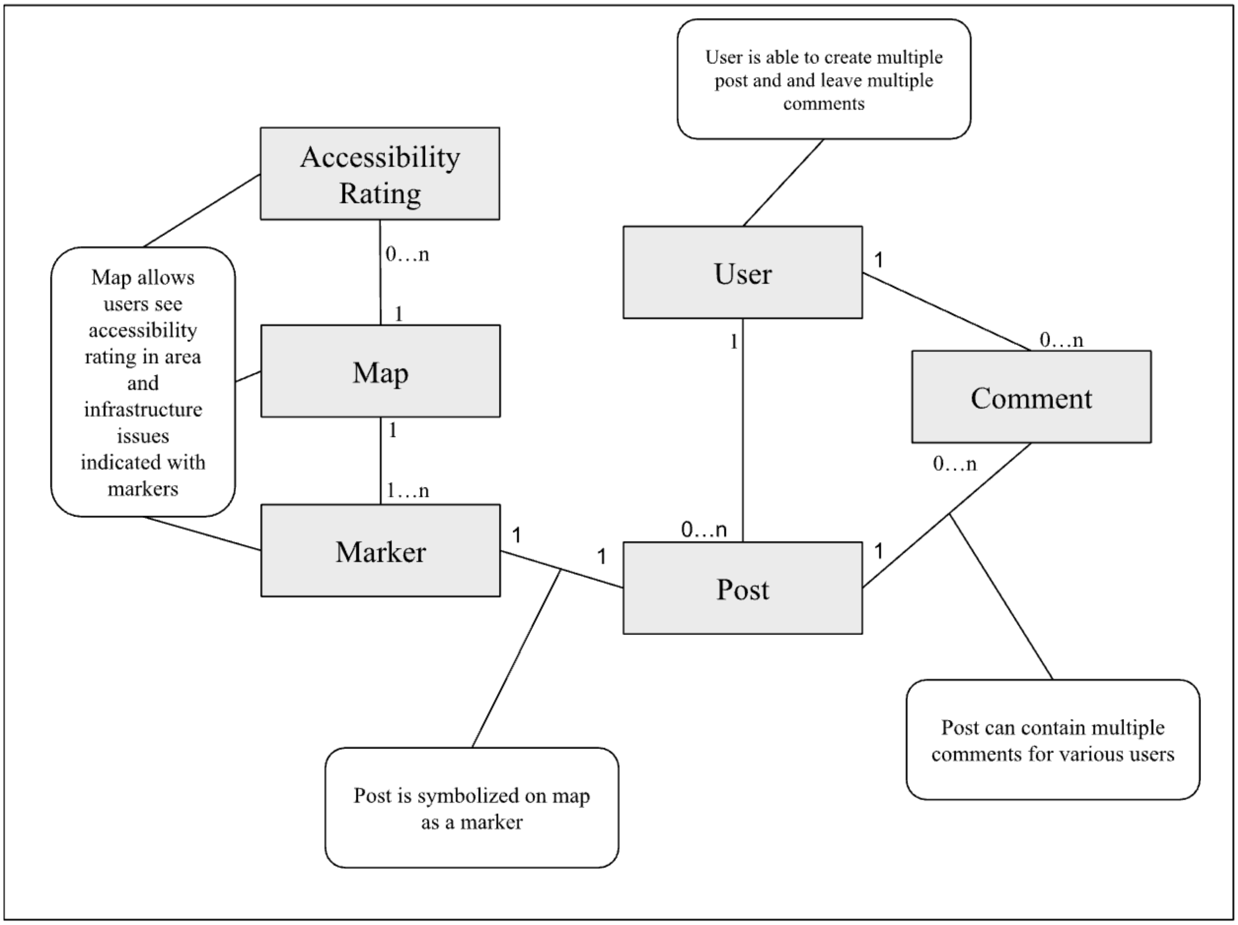
4. SOFTWARE ITEM DETAILED DESIGN

4.1. STRUCTURE

4.1.1. Software Unit Detailed Design



4.2. STATIC RELATIONSHIP OF SOFTWARE UNIT


User is able to create multiple post and and leave multiple comments
Accessibility Rating
0…n
1
Map allows users see accessibility rating in area and infrastructure issues indicated with markers
Map
1
1…n
Marker
1
1
User
1
0…n
1
Comment
0…n
0…n
Post
1
Post is symbolized on map as a marker
Post can contain multiple comments for various users

Above is the class collaboration diagram for the components of the *Winfra* project, which displays the relationships between components and the cardinalities of those relationships.

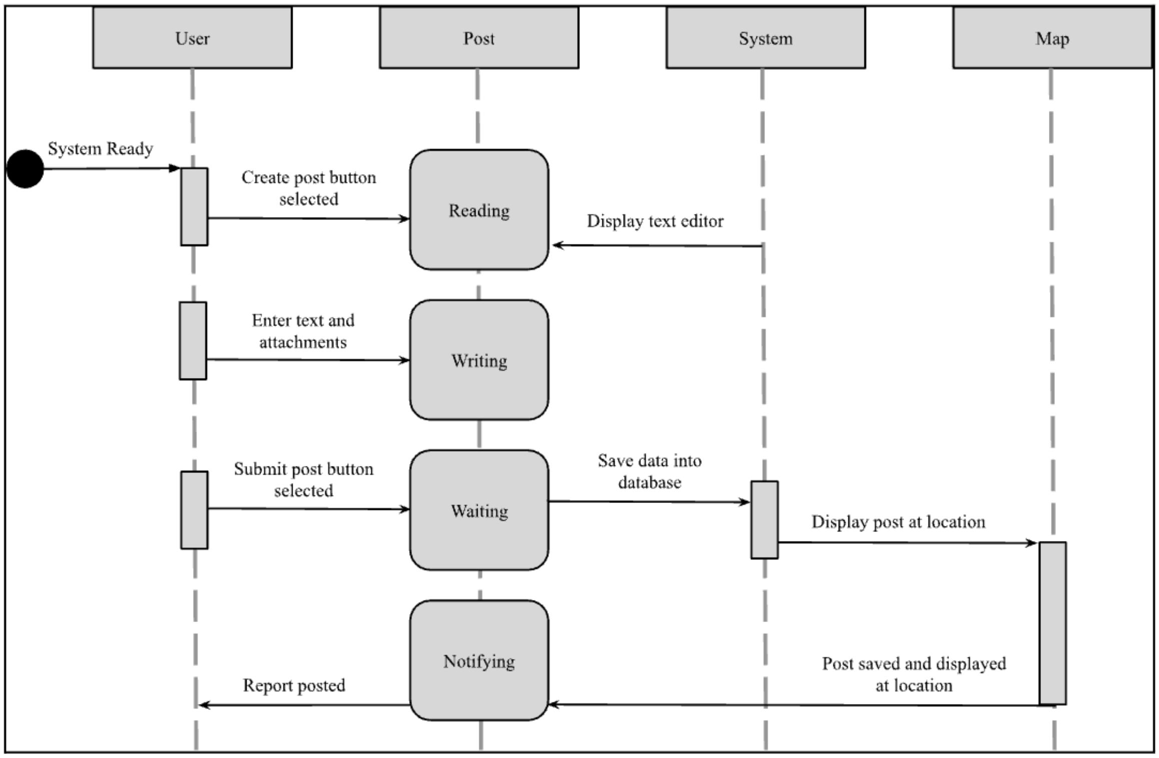
4.2.1. Run-time Object Instances

Ratings are generated at runtime, because they are subjective to the users' target areas, of which there are an infinite number. They are deleted when their scope is exceeded.

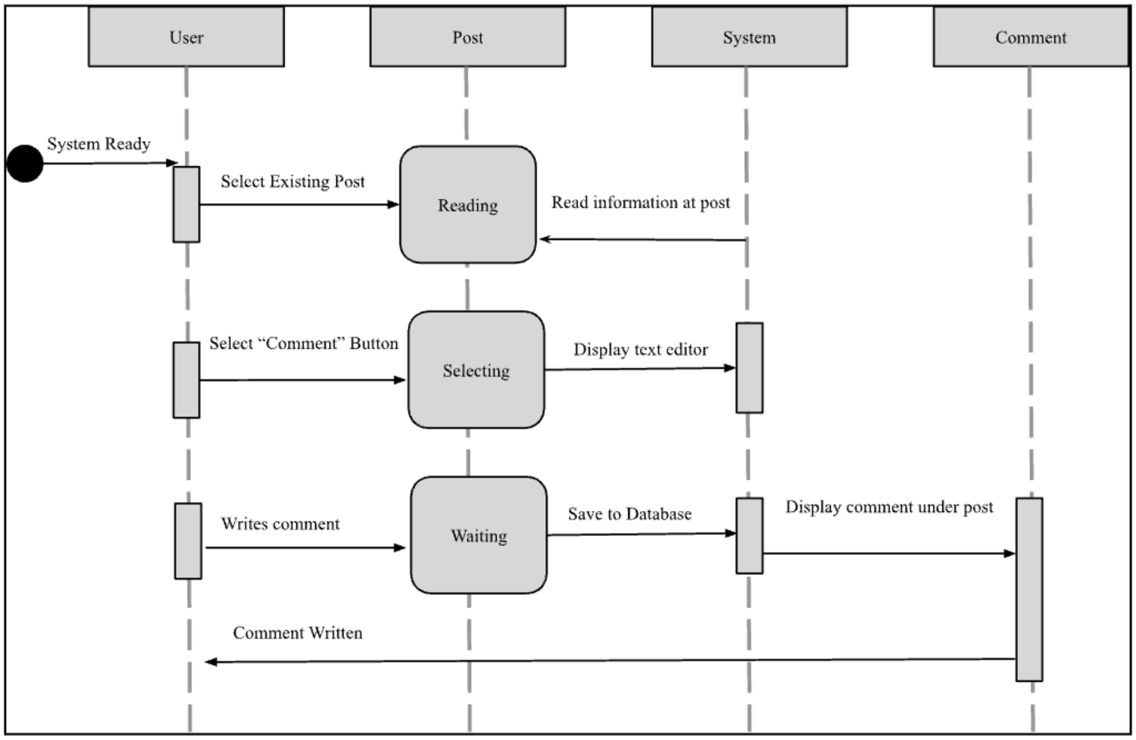
4.3. BEHAVIOR

4.3.1. Sequence Diagrams

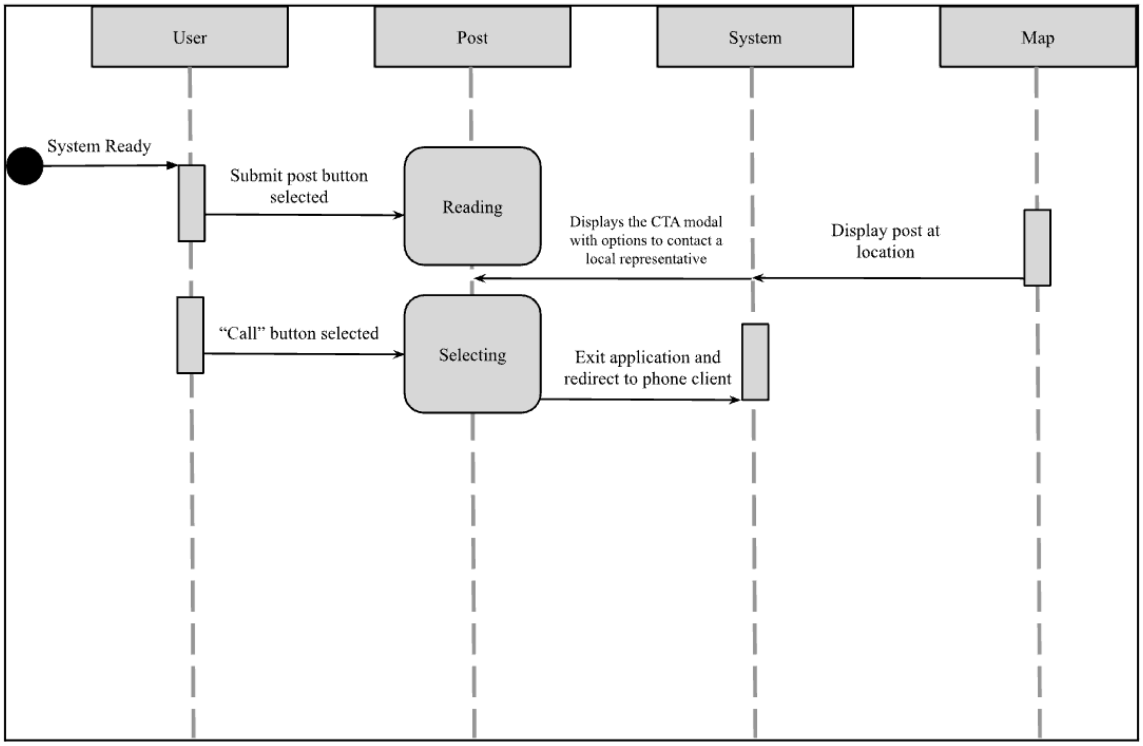
MIS-CNP



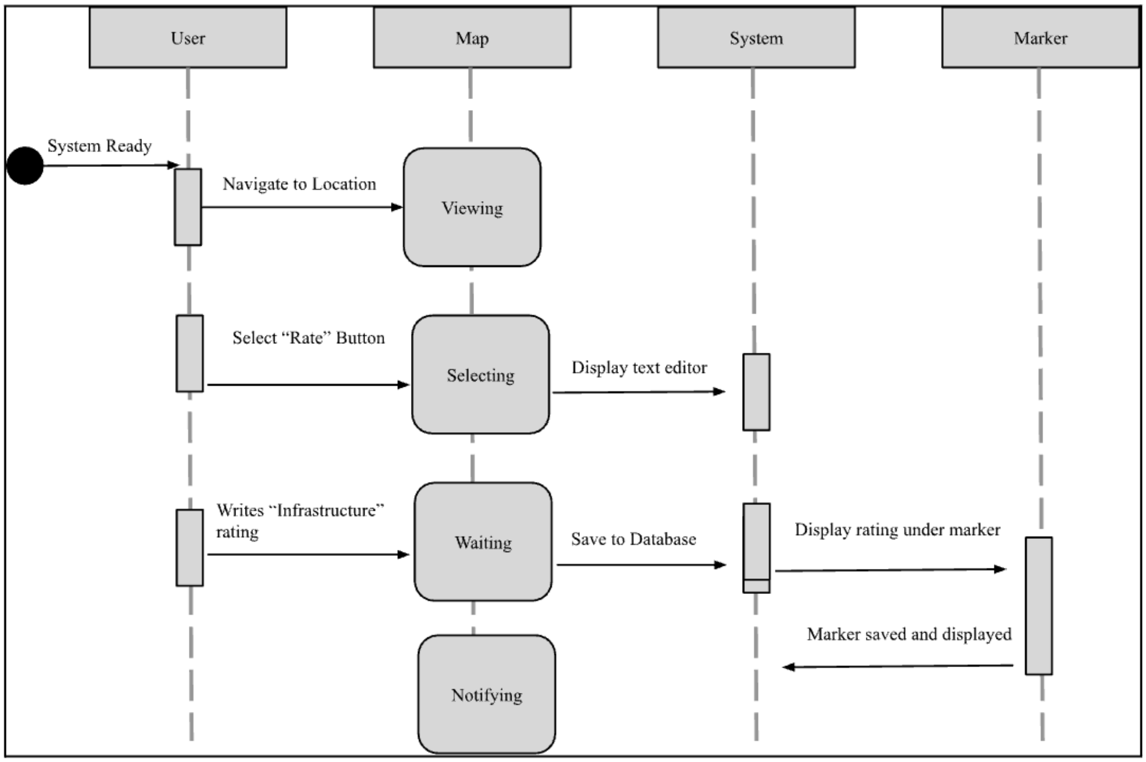
MIS-CUP


User
Post
System
Comment
System Ready
Select Existing Post
Reading
Read information at post
Select “Comment” Button
Selecting
Display text editor
Writes comment
Waiting
Save to Database
Display comment under post
Comment Written

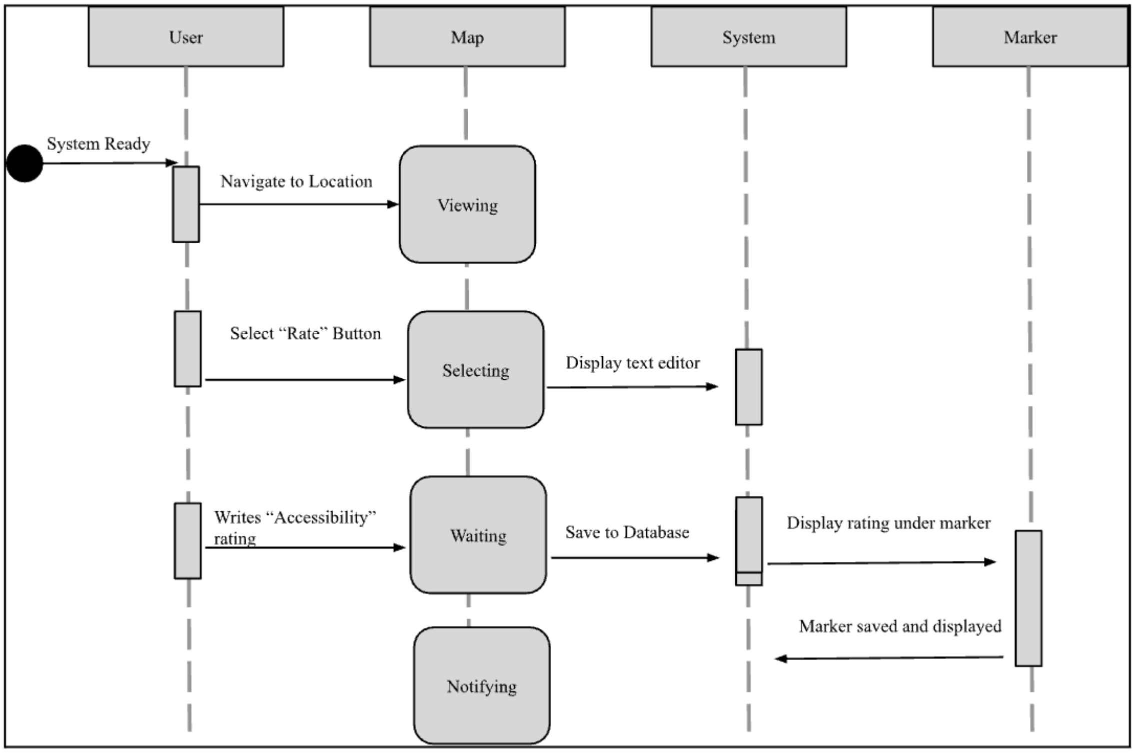
CTA


User
Post
System
Map
System Ready
Submit post button selected
Reading
Displays the CTA modal with options to contact a local representative
Display post at location
"Call" button selected
Selecting
Exit application and redirect to phone client

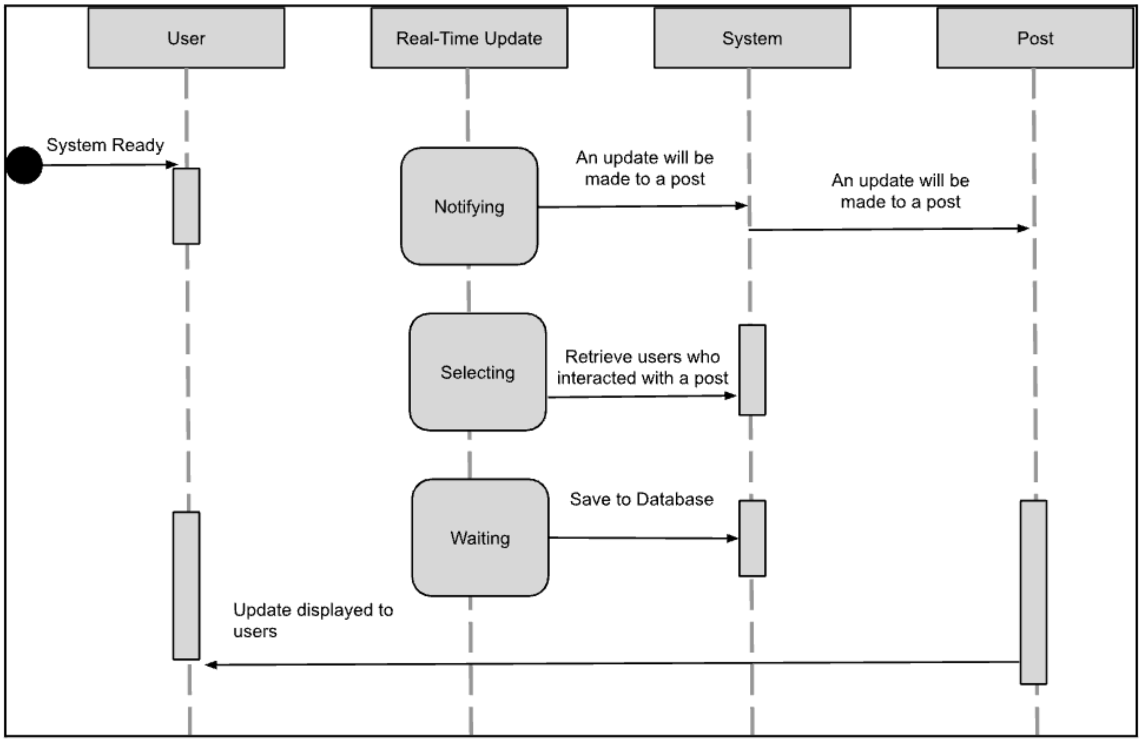
MVA-VIR


User
Map
System
Marker
System Ready
Navigate to Location
Viewing
Select “Rate” Button
Selecting
Display text editor
Writes “Infrastructure” rating
Waiting
Save to Database
Display rating under marker
Marker saved and displayed
Notifying

MVA-VAR

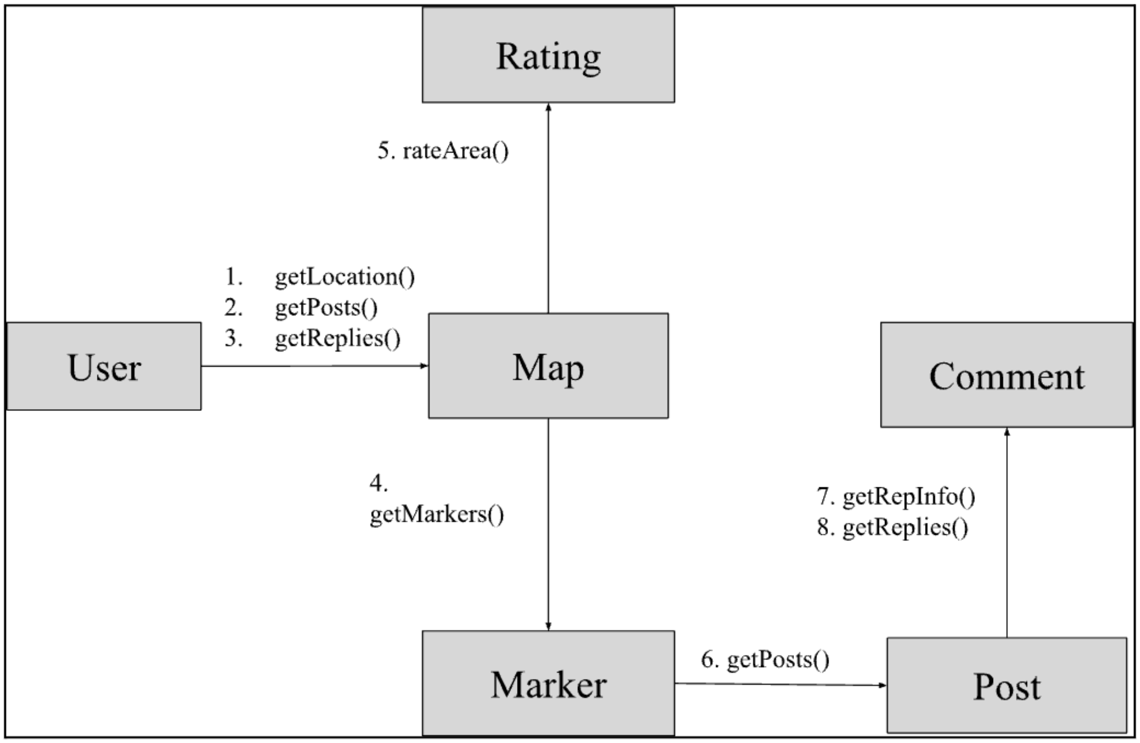

User
Map
System
Marker
System Ready
Navigate to Location
Viewing
Select “Rate” Button
Selecting
Display text editor
Writes “Accessibility” rating
Waiting
Save to Database
Display rating under marker
Marker saved and displayed
Notifying

RTU


User
Real-Time Update
System
Post
System Ready
Notifying
An update will be made to a post
An update will be made to a post
Selecting
Retrieve users who interacted with a post
Waiting
Save to Database
Update displayed to users

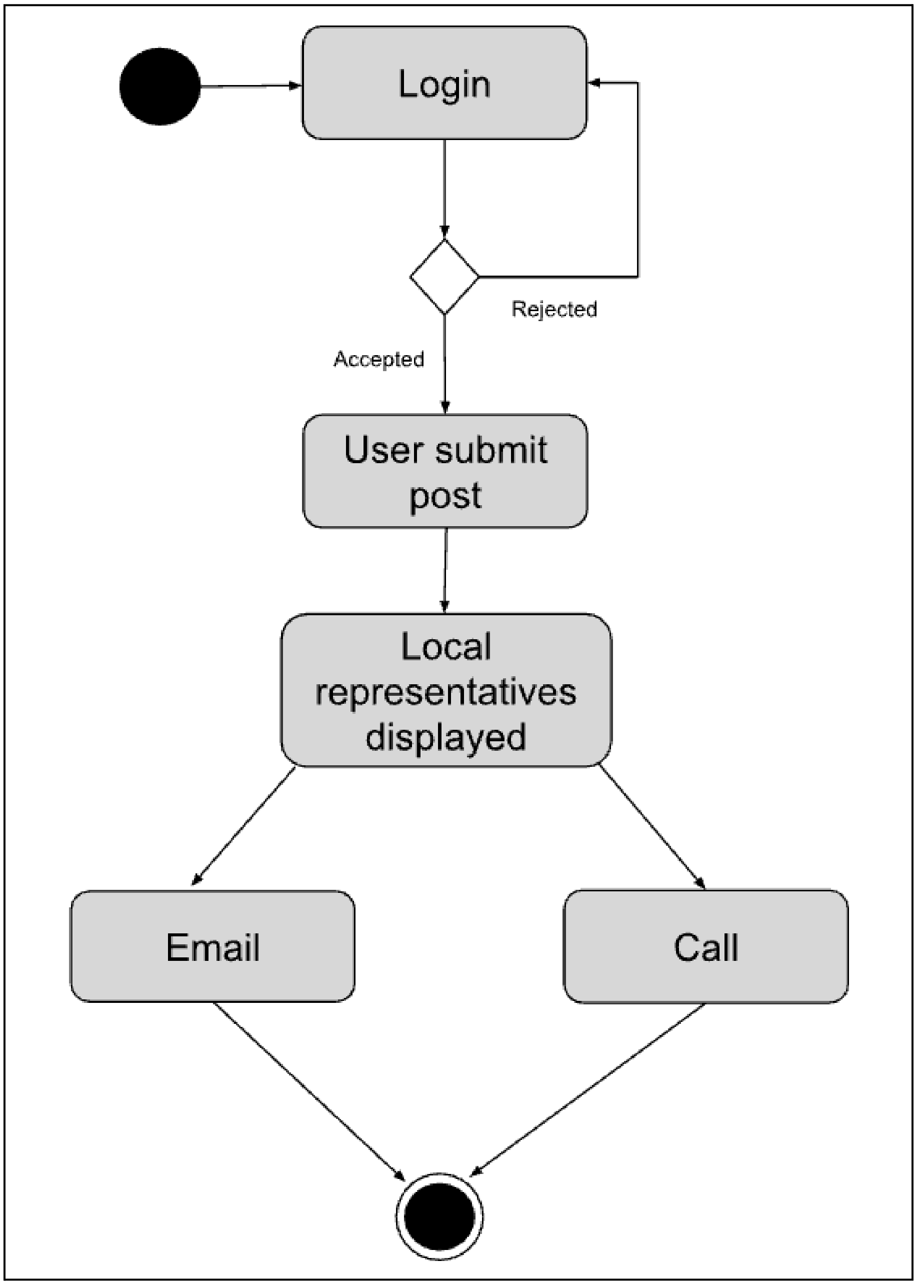
4.3.2. Collaboration Diagram

The collaboration diagram below depicts the flow of information between independent components of the system. The methods described in the diagram represent abstraction and may not have literal implementations.

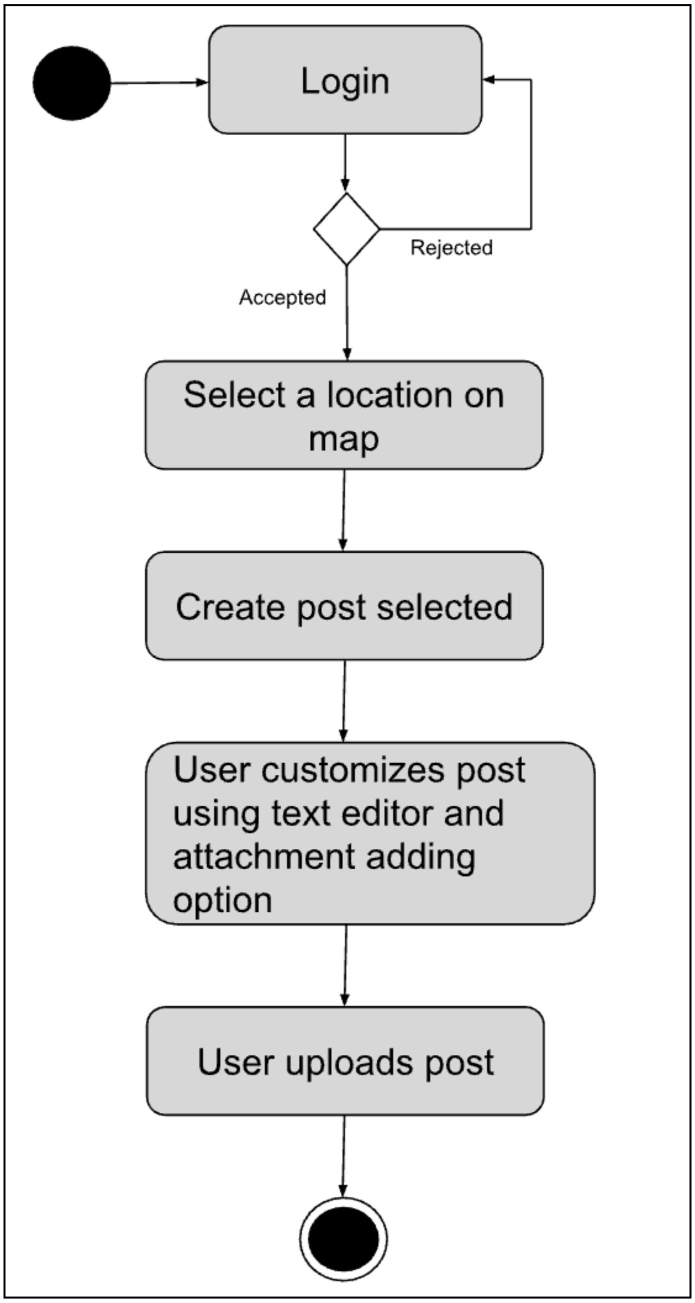


4.3.3. Activity Diagrams

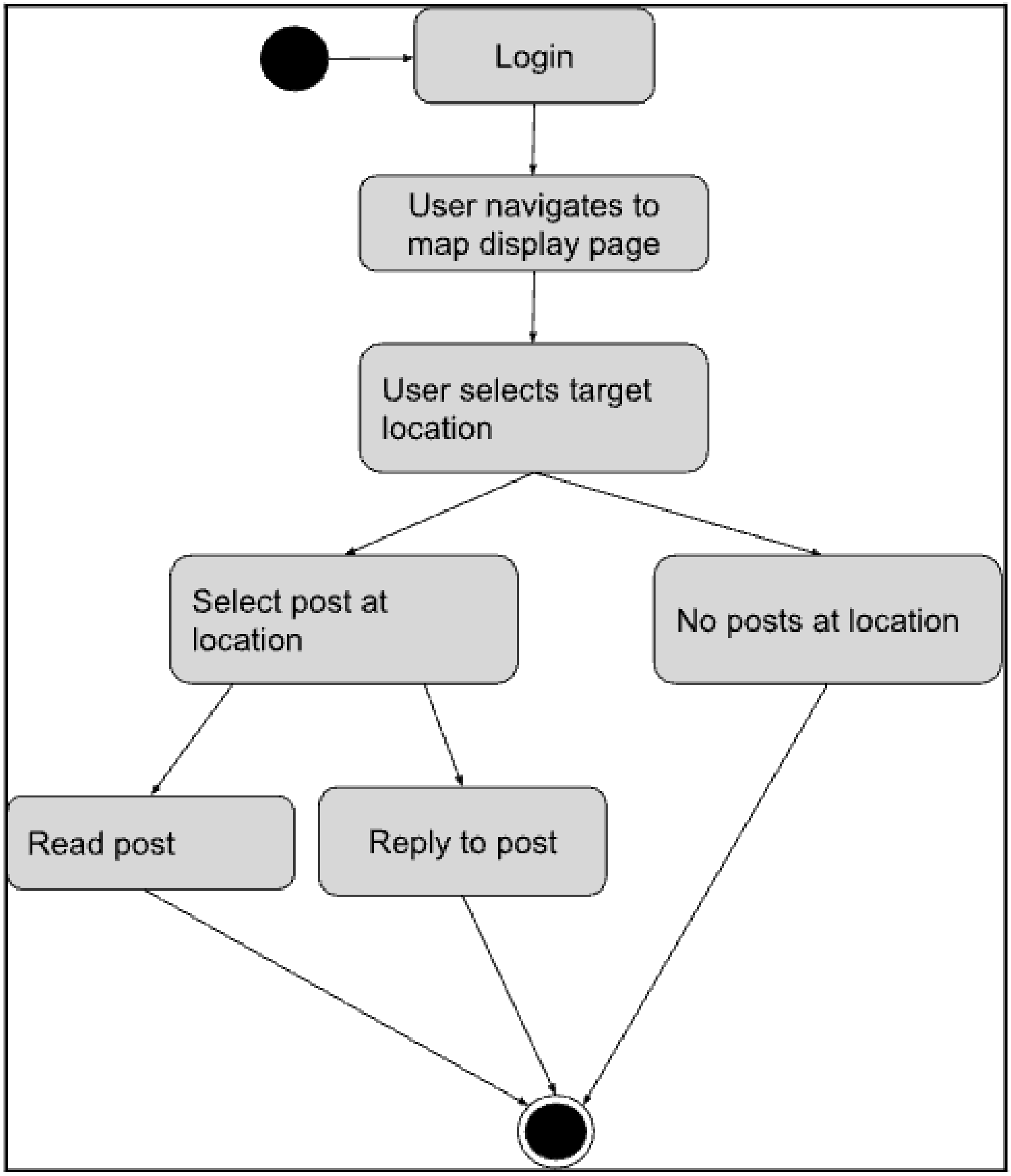
CTA-CLR



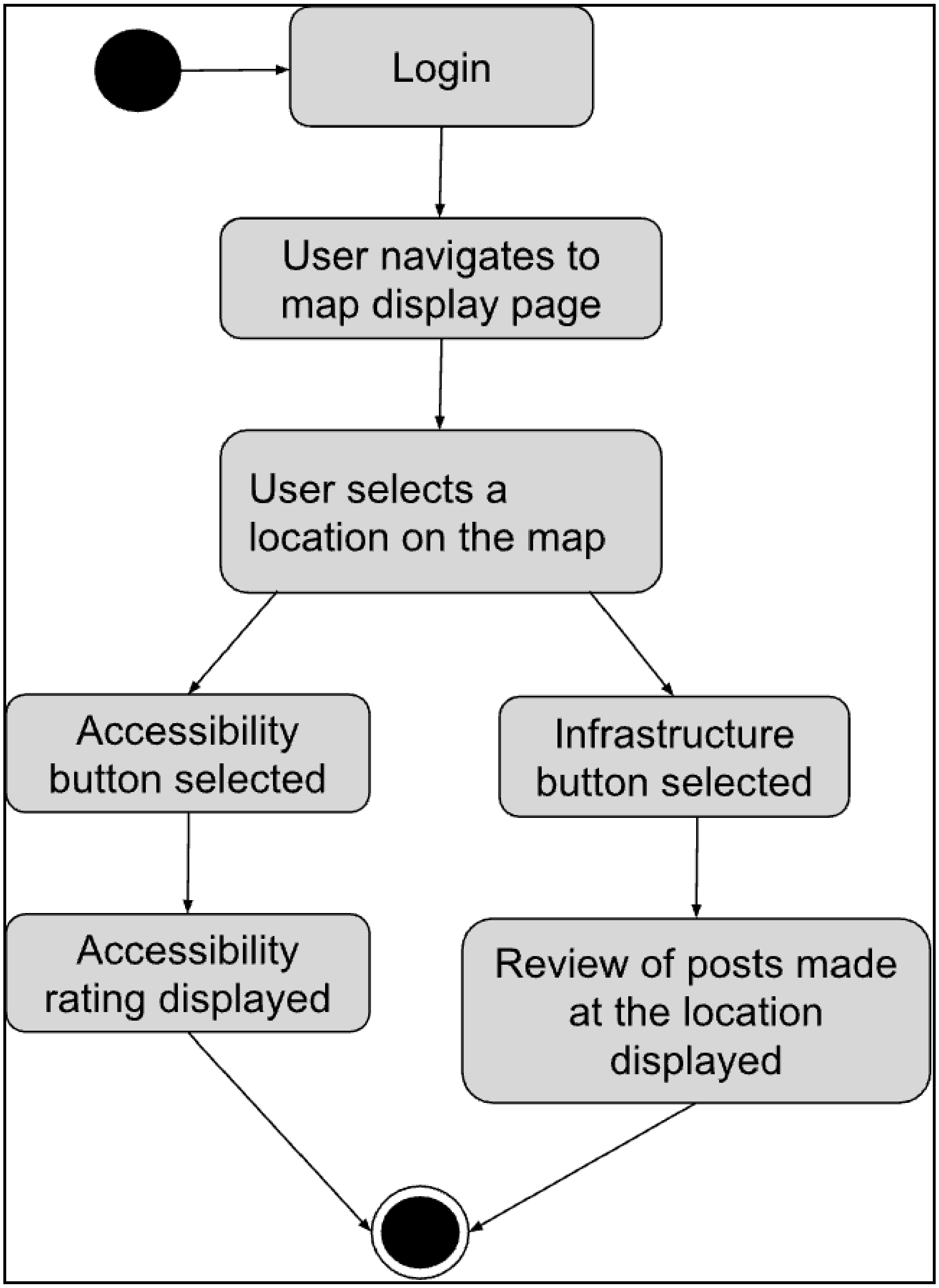
MIS–CNP


Login
Rejected
Accepted
Select a location on map
Create post selected
User customizes post using text editor and attachment adding option
User uploads post

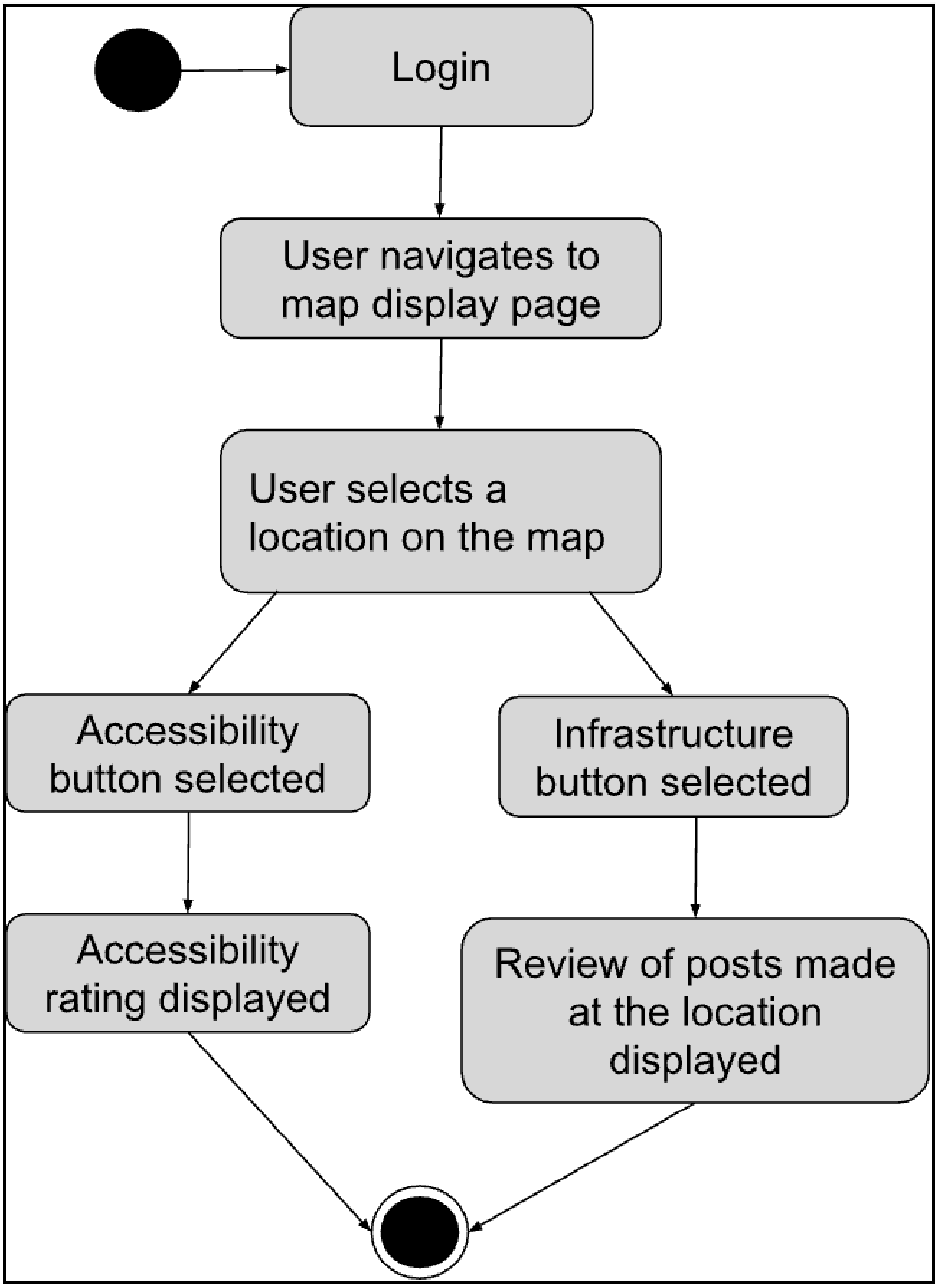
MIS-CUP


Login
User navigates to map display page
User selects target location
Select post at location
No posts at location
Read post
Reply to post

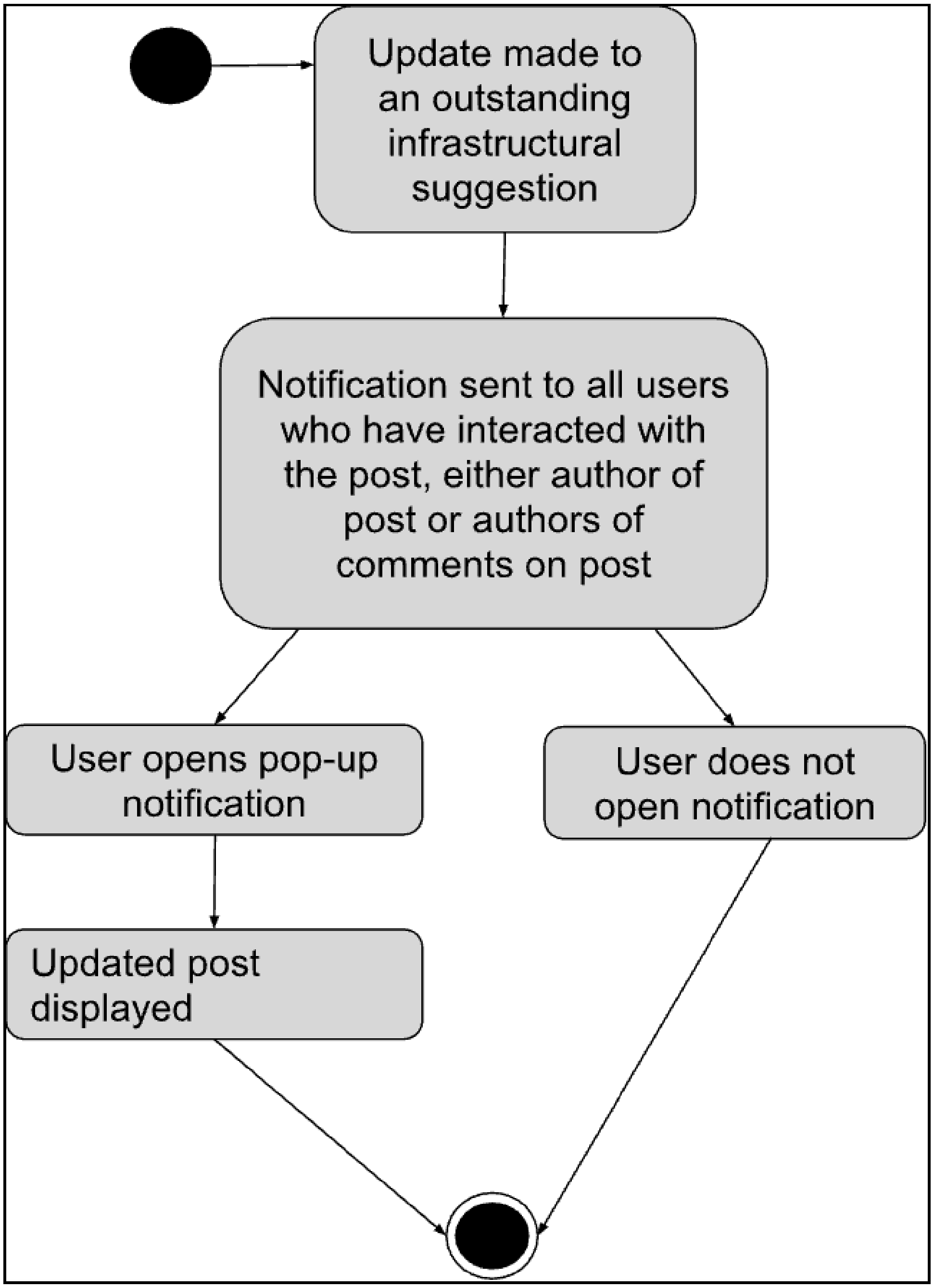
MVA-VIR


Login
User navigates to map display page
User selects a location on the map
Accessibility button selected
Infrastructure button selected
Accessibility rating displayed
Review of posts made at the location displayed

MVA-VAR


Login
User navigates to map display page
User selects a location on the map
Accessibility button selected
Infrastructure button selected
Accessibility rating displayed
Review of posts made at the location displayed

RTU


Update made to an outstanding infrastructural suggestion
Notification sent to all users who have interacted with the post, either author of post or authors of comments on post
User opens pop-up notification
User does not open notification
Updated post displayed

4.4. CONCEPT OF EXECUTION

Upon launch of the *Winfra* web application, the user interface is launched, and its goal in its initial interaction with the end user is to receive as input a candidate username-password pair. By this point, the database has been launched, and checks the given pair against existing user data, using the username as a key. If the credentials are accepted, the map API is launched, with the purpose of continuously displaying the markers for the runtime of the application. If a marker is selected, the database is accessed to return the post and comments associated with that marker, and whenever a post or comment is made, it is returned to the database. Ratings are generated at runtime computationally, since they are subjective to users' target areas, of which there are an infinite number.

4.5. INTERFACE DESIGN

4.5.1. Unique Identifier of Interface

The interfaces of the *Winfra* project are uniquely identified as follows:

User Authentication API	UA
Post Submission API	PS
Comment API	CT
Map Data API	MD
User Data Access	DA

Post Management API	PM
Comment Management API	CM
Geolocation API	GL
Local Representative Lookup	LR
Infrastructure Marker API	MK

4.5.2. Interface Identification and Diagrams

UA	Implemented using the DA interface.
PS	Implemented using the PM interface.
CT	Implemented using the CM interface.
MD	Implemented using Leaflet API.
DA	Reusable; implemented using Postgres.
PM	Reusable; implemented using Postgres.
CM	Reusable; implemented using Postgres (similar to PM).
GL	Implemented using Leaflet API.
LR	Implemented using Postgres; data scraped from public records.
MK	Implemented using Leaflet API.

5. IMPLEMENTATION ARCHITECTURE

5.1. ALL ACTIVE AND PASSIVE CLASSES ASSIGNED TO COMPONENTS

To be completed for a future release

5.2. DIAGRAMS OF PHYSICAL PACKAGING OF LOGICAL COMPONENTS

To be completed for a future release

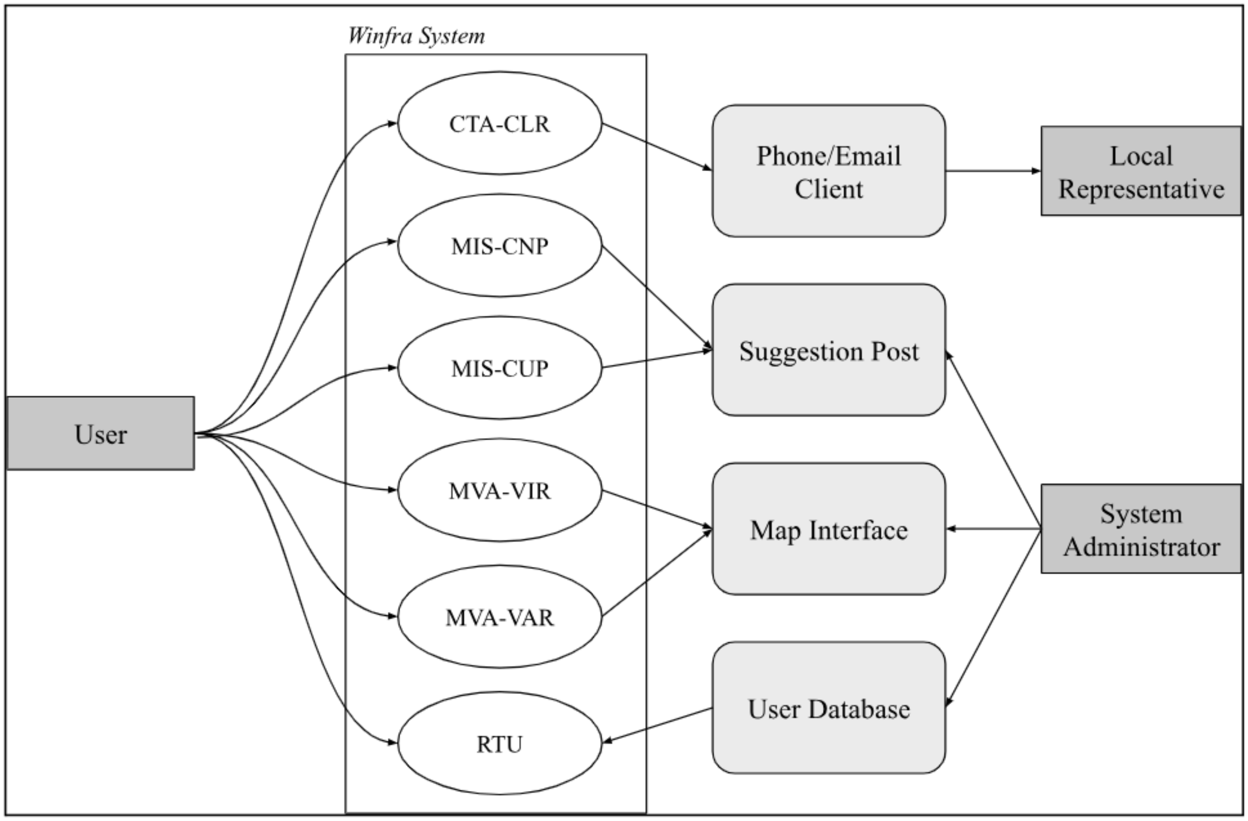
6. DEPLOYMENT ARCHITECTURE

6.1. PHYSICAL DEPLOYMENT ARCHITECTURE ALGORITHM

The breakdown of our business needs requires the team to develop a website, therefore we will incorporate an interactive map API to support location-based logging of infrastructure issues, allowing users to pin their location and visualize problems geographically. Users will also be able to report cases of inaccessibility in their area requiring the implementation of a community forum. Users will be able to quickly access their local representatives' contact information for issues concerning public infrastructure. Users will also be able to contact local contractors for issues concerning private infrastructure. These interactions will require location-based services, communication tools, and data management within the application. Additionally, the development team must decide on the best languages to construct the front-end and back-end of the application. We also require the implementation of scalable software. We need to maintain a database of users and their activity; this can be accomplished using PostgreSQL. We also require an application testing interface, for our application we can use Selenium WebDriver. Security measures will include SSL encryption, Web Application Firewall (WAF), and Role-Based Access Control (RBAC). Regarding the LAN connection between the UI and App Server, a 100 Mbps network is recommended for handling moderate traffic, API calls, and map interactions, but for high user loads and real-time updates, a 1 Gbps connection would be ideal to ensure seamless performance and low latency.

7. REQUIREMENTS

7.1. USE CASE DIAGRAM



In the use case diagram above, actors are denoted by rectangles, use cases are denoted by ellipses, and information/systems are denoted by rounded rectangles. Arrows represent the flow of information and decisions.

7.2. USE CASE DESCRIPTIONS

Make Infrastructural Suggestion—Create New Post (MIS-CNP)		
Description	Iteration: 1, last modification: Oct. 14 by E. Wilson. Primary actor: End user (or “user”). Goal in context: To create and post an infrastructural suggestion. Trigger: User is dissatisfied with perceived infrastructural shortcomings.	
Pre-Conditions	User is logged into the *Winfra* system; user has location services enabled; user location matches target location.	
Flows	**Basic or Normal Flows**	1. The user logs into *Winfra* (see use case **ULA**). 2. The user selects the “post” button from the major function buttons. 3. The system displays a text editor. 4. The user writes and customizes their post detailing the infrastructure problem including problem, severity, and images. 5. The user selects the “submit” button.
	Alternative Flows	1. The user selects an existing post on the map interface—see **MIS-CUP**.
Post Conditions	New user post is to be entered into the database and displayed on the map.	
Special Requirement	1. *Security*: Unfinished/discarded posts must be properly disposed and deleted. 2. *Usability*: The text editor must be simple and familiar. 3. *Maintainability*: The target location must be an attribute of the post.	
Extension Points	1. Location services are not enabled—optional **Enable Location Services (ELS)** use case.	

<table>
<tr><th colspan="3">Call To Action—Contact Local Representative (CTA-CLR)</th></tr>
<tr><td><b>Description</b></td><td colspan="2">Iteration: 1, last modification: Oct. 14 by J. Blancaflor.
Primary actor: <i>Winfra</i> system.
Goal in context: To prompt end user to contact local representatives about their infrastructure concerns.
Trigger: User selects the “submit” button when creating a post (Basic Flow 7 in <b>MIS-CNP</b>).</td></tr>
<tr><td><b>Pre-Conditions</b></td><td colspan="2">User is logged into the <i>Winfra</i> system and has just posted a new infrastructural suggestion; user has phone and/or email services enabled.</td></tr>
<tr><td rowspan="2"><b>Flows</b></td><td><b>Basic or Normal Flows</b></td><td>1. The user logs into <i>Winfra</i> (see use case <b>ULA</b>).
2. The user submits a new post (see use case <b>MIS-CNP</b>).
3. The system displays the CTA modal with options to contact a local representative.
4. The user selects the “Call” button.
5. The user is redirected to their phone client.
6. The user makes a call to their local representative (exits scope of <i>Winfra</i>).</td></tr>
<tr><td><b>Alternative Flows</b></td><td>1. The user selects the “Email” button from the modal. They are redirected to their email client, where their post has been copied into a new draft to their local representative (exits scope of <i>Winfra</i>).</td></tr>
<tr><td><b>Post Conditions</b></td><td colspan="2">The user is in contact with a local representative regarding their infrastructure concerns.</td></tr>
<tr><td><b>Special Requirement</b></td><td colspan="2">1. Security: The system must protect the privacy of users’ contact information and correspondences.
2. Performance: The display of the modal should occur immediately after the user submits a new post in order to retain their attention.</td></tr>
<tr><td><b>Extention Points</b></td><td colspan="2">None</td></tr>
</table>

Make Infrastructural Suggestion—Comment Under Post (MIS-CUP)		
Description	Iteration: 1, last modification: Oct. 14 by E. Wilson. Primary actor: End user (or "user"). Goal in context: To comment under an existing infrastructural suggestion. Trigger: User reacts to a sentiment held by another user.	
Pre-Conditions	User's target location has outstanding suggestions.	
Flows	**Basic or Normal Flows**	1. The user logs into *Winfra* (see use case **ULA**). 2. The user navigates to their target location. 3. The user selects an existing post at their target location. 4. The user reads the suggestion contained in the selected post. 5. The user selects the "Reply" button under the post or an existing reply. 6. The system displays a text editor. 7. The user writes and customizes their post. 8. The user selects the "submit" button.
	Alternative Flows	1. No post exists at the user's target location—see **MIS-CNP**. 2. The user selects the "Rate" button from the major function buttons—see **MVA-VIR, MVA-VAR**.
Post Conditions	The user's post is public and can be viewed by other users, including local representatives and contractors.	
Special Requirement	1. *Usability*: The user must be able to read the post they are replying to while writing their reply. 2. Maintainability: There must be an ownership relationship between "Posts" and "Comments".	
Extention Points	1. Location services are not enabled—optional **Enable Location Services (ELS)** use case.	

<table>
<tr><th colspan="3">Map-View Analysis—View Infrastructure Rating (MVA-VIR)</th></tr>
<tr><td><b>Description</b></td><td colspan="2">Iteration: 1, last modification: Oct. 14 by R. Mukkamala.
Primary actor: End user (or “user”).
Goal in context: To analyze the infrastructure rating of an area as reported by <i>Winfra</i> users.
Trigger: User logs onto <i>Winfra</i>.</td></tr>
<tr><td><b>Pre-Conditions</b></td><td colspan="2">User is logged into the <i>Winfra</i> system; user has location services enabled.</td></tr>
<tr><td><b>Flows</b></td><td><b>Basic or Normal Flows</b></td><td>1. The user logs into <i>Winfra</i> (see use case <b>ULA</b>).
2. The user navigates to the target location.
3. The user selects the “Rate” button from the major function buttons.
4. The system presents options for infrastructure and accessibility.
5. The user selects the “Infrastructure” button.
6. The system presents a review of the target location based on <i>Winfra</i> posts made in the selected bounds.</td></tr>
<tr><td></td><td><b>Alternative Flows</b></td><td>1. The user selects the “Accessibility” button—see <b>MVA-VAR</b>.</td></tr>
<tr><td><b>Post Conditions</b></td><td colspan="2">The user has learned qualitative information about the infrastructural features of their target location.</td></tr>
<tr><td><b>Special Requirement</b></td><td colspan="2">1. <i>Usability</i>: The view of the target location must reduce the analytical work that the user would undertake by reviewing the suggestion map as presented to them.
2. <i>Maintainability</i>: The collection and manipulation of data must remain efficient as posts grow more dense at target locations.</td></tr>
<tr><td><b>Extention Points</b></td><td colspan="2">1. Location services are not enabled—optional <b>Enable Location Services (ELS)</b> use case.</td></tr>
</table>

Map-View Analysis—View Accessibility Rating (MVA-VAR)		
Description	Iteration: 1, last modification: Oct. 13 by E. Wilson. Primary actor: End user (or “user”). Goal in context: To analyze the accessibility of an area as reported by *Winfra* users. Trigger: User logs onto *Winfra*.	
Pre-Conditions	User is logged into the *Winfra* system; the user has location services enabled.	
Flows	**Basic or Normal Flows**	1. The user logs into *Winfra* (see use case **ULA**). 2. The user navigates to their target location. 3. The user selects the “Rate” button from the major function buttons. 4. The system presents options for infrastructure and accessibility. 5. The user selects the “Accessibility” button. 6. The system presents a review of the target location based on *Winfra* posts made in the selected bounds.
	Alternative Flows	1. The user selects the “Infrastructure” button—see **MVA-VIR**.
Post Conditions	The user has learned qualitative information about the accessibility of their target location.	
Special Requirement	1. *Usability*: The view of the target location must reduce the analytical work that the user would undertake by reviewing the suggestion map as presented to them. 2. *Maintainability*: The collection and manipulation of data must remain efficient as posts grow more dense at target locations.	
Extention Points	1. Location services are not enabled—optional **Enable Location Services (ELS)** use case.	

Real-Time Update (RTU)		
Description	Iteration: 2, last modification: May 4 by E. Wilson. Primary actor: *Winfra* system. Goal in context: To notify the end user that a change has been made relevant to their outstanding suggestion(s). Trigger: Infrastructure that a user has reviewed has been changed.	
Pre-Conditions	User has interacted with the post that has been updated.	
Flows	**Basic or Normal Flows**	1. The *Winfra* system receives notice of an update to an infrastructural feature with outstanding suggestions. 2. The *Winfra* system notifies all users that have posted or replied to suggestions regarding this feature that updates have been made. 3. The user selects the pop-up notification. 4. The system displays the suggestion post that the user has previously interacted with, now including the update to the infrastructural feature in question.
	Alternative Flows	1. The user is not logged into *Winfra*—a system notification is sent to the user.
Post Conditions	The user is notified about an update made concerning their suggestion.	
Special Requirement	1. *Security:* Users should not see who else has been notified regarding updates to the feature in question. 2. *Usability*: Users should not receive redundant updates concerning multiple suggestions about the same infrastructural feature.	
Extention Points	1. The user does not have notifications enabled for *Winfra*—optional **Enable Notification Services (ENS)** use case.	

8. PSEUDOCODE

The Winfra web application utilizes computational resources to provide real-time insights into system performance and availability such as the Google Maps API service. For The web app, it requires CPU resources for data processing and real-time analytics. RAM is critical for caching user information, handling concurrent user sessions, and supporting live data for the pin interface. The system's disk I/O is primarily driven by logging, historical data storage, and database transactions, ensuring efficient retrieval and up to date user states. Additionally, network utilization depends on the frequency of data polling, API interactions with Google Maps API, and real-time notifications. To optimize performance, the web app plans to employ load balancing to minimize resource overhead.

9. DICTIONARIES

Classes

Name	Description	Methods	Attributes
User	Account on the *Winfra* system	create_user(), get_user_by_username(), verify_user()	User ID, User Name, First Name, Last Name, Email, Password, Posts, Comments
Post	Primary function of *Winfra*, contains qualitative and quantitative reviews of infrastructural features	create_post(), delete_post(), get_posts_by_user(), get_posts_by_location(), get_posts()	Post ID, Title, Description, Location Name, Latitude, Longitude, Status Issue, Image URL, Accessibility Level, Created At, Author Username, Author, Comments
Comment	Subsidiary class whose instances belong to Post instances	create_comment(), delete_comment(), get_comments_by_post()	Comment ID, Content, Created At, Author Username, Post ID, Author, Post

Methods

Class	Name	Description	Arguments
User	create_user()	Set up a new *Winfra* account	First Name, Last Name, Email, Username, Password
	get_user_by_username()	Return user with given username	Username
	verify_user()	Verify login information	User ID, Password
Post	create_post()	Write new post (infrastructural suggestion)	Title, Description, Location Name, Latitude, Longitude, Accessibility Level, Image URL, Status, Issue

	delete_post()	Remove the indicated post	Post ID
	get_posts_by_user()	Get all the posts made by a given user	User ID
	get_posts_by_location()	Get all the posts within a radius of a given point	Latitude, Longitude, Radius
	get_posts()	Get posts sorted by date of creation	Limit (number of posts)
Comment	create_comment()	Write new comment under a post	Post ID, Content
	delete_comment()	Delete given comment	Comment ID
	get_comments_by_post()	Get all comments under a given post	Post ID

Attributes

Class	**Name**	**Description**	**Simple/Complex**	**Type**
User	User ID	Unique identification for a user	Simple	Integer
	User Name	Unique display name of user	Simple	String
	First Name	First name of user	Simple	String
	Last Name	Last name of user	Simple	String
	Email	User’s email	Simple	String
	Password	User’s login password	Simple	String
	Posts	Posts user has submitted	Complex	Relationship
	Comments	Comments under a Post	Complex	Relationship

Post	Post ID	Unique identifier for user’s post	Simple	Integer
	Title	Title of a post	Simple	String
	Description	Description of post	Simple	String
	Location Name	Name of current user location	Simple	String
	Latitude	Latitude of current user location	Simple	Float
	Longitude	Longitude of current user location	Simple	Float
	Status	Status of infrastructure	Simple	String
	Issue	Elaborated infrastructural issue	Simple	String
	Image URL	URL to external file to supplement text suggestions	Simple	String
	Accessibility Level	Qualitative rating of accessibility	Simple	String
	Created At	Time of post submission	Complex	DateTime
	Author Username	Username of creator of post	Simple	String
	Author	Author of post	Complex	Relationship
	Comments	Comments under post	Complex	Relationship
Comment	Comment ID	Unique identifier for user’s comment	Simple	Integer
	Content	Text of comment	Simple	String

	Created At	Time of comment submission	Complex	DateTime
	Author Username	Username of creator of comment	Simple	String
	Post ID	Post under which comment was left	Simple	Integer
	Author	Author of post	Complex	Relationship
	Post	Post under which comment was left	Complex	Relationship

10. SOFTWARE ITEM COMPUTER RESOURCE UTILIZATION

The analysis of the resource utilization of the system begins with the respective sizes of the software components. Each post can contain 2KB of text, and each comment can contain 1KB of text. The markers that hold the posts store their latitude and longitude, which are represented as floating point numbers, usually either 4B or 8B in size. Notifications can store 100B of text, not including the link to the related post, which will also be allocated 100B, bringing the total size of a Notification to 200B. Ratings are generated at runtime but carry four coordinates that delineate the target area as well as a floating point number to indicate the score and 3 links to related posts. Thus the size of a Rating ranges from 320B to 340B. Finally, the CTA stores an email-generating link and a phone number, yielding an allocated size of 120B.

The system is projected to have, on average, 5-15 posts per user, with 10-20 comments per post. Since each new post is represented by a marker, there will be 5-15 markers per user as well. Users are expected to generate 3 Ratings and to receive 0-1 Notifications and 1 CTA (after the creation of a post) per system login. The number of expected posts in a user's target area ranges from 15-30. The number of expected users ranges from 1000 to 500000.

Then at a minimum, the system will require 5 • (2056B + 10 • 1024B) • 1000 = 61.48MB to store and 15 • 1032B + 3 • 320B + 120B = 16.17KB to run locally, and at a maximum, the system will require 15 • (2062B + 20 • 1024B) • 500000 = 169.10GB to store and 30 • 1040B + 3 • 340B + 200B + 120B = 31.78KB to run locally. Thus the scalability of the project is dependent on the scalability of the database, while users' experience will not vary significantly as the user base grows.

11. REQUIREMENTS TRACEABILITY

The requirements traceability matrix below illustrates how elements of the *Winfra* project are traceable forward to work products and backward to requirements.

Requirements Traceability Matrix

Requirement ID	**Requirement Desc.**	**Use Case ID**	**Software Component**	**Deliverable** (*None at this time)*
Req. 1.0	Infrastructural Suggestion	**MIS-CNP**	Post, Comment	
Req. 2.0	Community Building	**MIS-CUP**	Comment	
Req. 3.0	Interactive Map	**MVA-VIR, MVA-VAR**	Marker, Rating	
Req. 4.0	Promotion of Accessibility	**MVA-VAR**	Rating, CTA	
Req. 5.0	Real-Time Updates	**RTU**	Notification	
Req. 6.0	Contacting Local Representatives	**CTA-CLR**	CTA	

12. SYSTEM DESIGN TESTING

The software quality group will ensure that the objectives of the system design are met by testing using four methods including peer review, self-checks, walkthroughs, and inspection. When there are changes caused by the design, requirements, or updates, the software quality group will update the testing plan.

To validate the system's design and ensure compliance with requirements, the following review methods will be conducted before formal testing. Peer reviews will be used by developers to conduct collaborative code and design reviews to identify issues throughout development, to ensure best practices and validate the performance and reliability of the system. Self checks will be used during testing, as such, developers will be expected to review their own code before submission of their implementation to prevent errors from being integrated with the main system. Walkthroughs will be conducted as structured sessions where developers and the software quality team walk will review the system design, user flows, and requirements implementation to confirm correctness. Lastly, formal inspections will be conducted where quality engineers will perform evaluations of the system's architecture and detect defects.

Product testing will be done by the software quality group with the assistance of the development team and will focus on verifying that all system components function as intended. For acceptance testing, the website will be presented to potential end users in beta to ensure it meets the requirements outlined by the clients and fix any bugs or errors they may encounter. If the tests are failed the system must go back into rework, otherwise the system can go into operation.

13. RATIONALE

None at this time

14. NOTES

None at this time

15. APPENDICES

15.1. DICTIONARIES

Classes

Name	Description	Methods	Attributes
User	Account on the *Winfra* system	create_user(), get_user_by_username(), verify_user()	User ID, User Name, First Name, Last Name, Email, Password, Posts, Comments
Post	Primary function of *Winfra*, contains qualitative and quantitative reviews of infrastructural features	create_post(), delete_post(), get_posts_by_user(), get_posts_by_location(), get_posts()	Post ID, Title, Description, Location Name, Latitude, Longitude, Status Issue, Image URL, Accessibility Level, Created At, Author Username, Author, Comments
Comment	Subsidiary class whose instances belong to Post instances	create_comment(), delete_comment(), get_comments_by_post()	Comment ID, Content, Created At, Author Username, Post ID, Author, Post

Methods

Class	Name	Description	Arguments
User	create_user()	Set up a new *Winfra* account	First Name, Last Name, Email, Username, Password
	get_user_by_username()	Return user with given username	Username

	verify_user()	Verify login information	User ID, Password
Post	create_post()	Write new post (infrastructural suggestion)	Title, Description, Location Name, Latitude, Longitude, Accessibility Level, Image URL, Status, Issue
	delete_post()	Remove the indicated post	Post ID
	get_posts_by_user()	Get all the posts made by a given user	User ID
	get_posts_by_location()	Get all the posts within a radius of a given point	Latitude, Longitude, Radius
	get_posts()	Get posts sorted by date of creation	Limit (number of posts)
Comment	create_comment()	Write new comment under a post	Post ID, Content
	delete_comment()	Delete given comment	Comment ID
	get_comments_by_post()	Get all comments under a given post	Post ID

Attributes

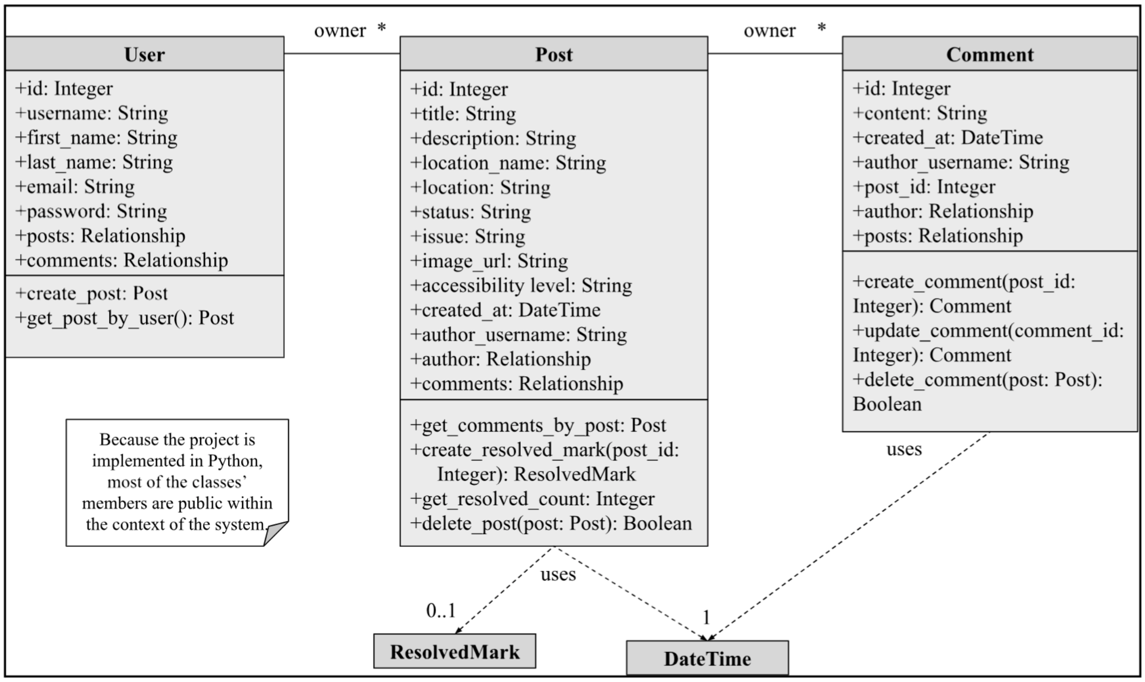
Class	**Name**	**Description**	**Simple/Complex**	**Type**
User	User ID	Unique identification for a user	Simple	Integer
	User Name	Unique display name of user	Simple	String
	First Name	First name of user	Simple	String

	Last Name	Last name of user	Simple	String
	Email	User’s email	Simple	String
	Password	User’s login password	Simple	String
	Posts	Posts user has submitted	Complex	Relationship
	Comments	Comments under a Post	Complex	Relationship
Post	Post ID	Unique identifier for user’s post	Simple	Integer
	Title	Title of a post	Simple	String
	Description	Description of post	Simple	String
	Location Name	Name of current user location	Simple	String
	Latitude	Latitude of current user location	Simple	Float
	Longitude	Longitude of current user location	Simple	Float
	Status	Status of infrastructure	Simple	String
	Issue	Elaborated infrastructural issue	Simple	String
	Image URL	URL to external file to supplement text suggestions	Simple	String
	Accessibility Level	Qualitative rating of accessibility	Simple	String
	Created At	Time of post submission	Complex	DateTime
	Author Username	Username of creator of post	Simple	String

	Author	Author of post	Complex	Relationship
	Comments	Comments under post	Complex	Relationship
Comment	Comment ID	Unique identifier for user’s comment	Simple	Integer
	Content	Text of comment	Simple	String
	Created At	Time of comment submission	Complex	DateTime
	Author Username	Username of creator of comment	Simple	String
	Post ID	Post under which comment was left	Simple	Integer
	Author	Author of post	Complex	Relationship
	Post	Post under which comment was left	Complex	Relationship

15.2. UML DIAGRAMS

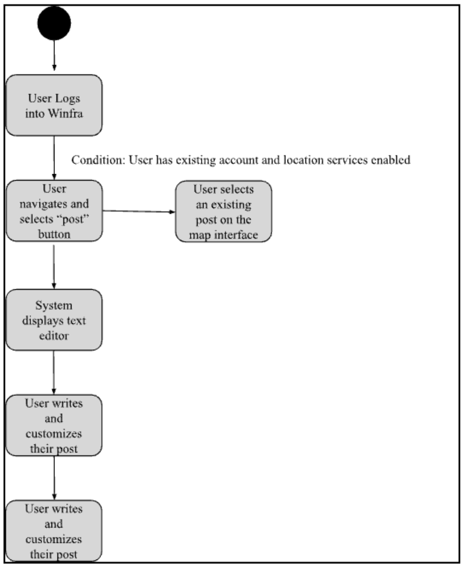
Class Diagram



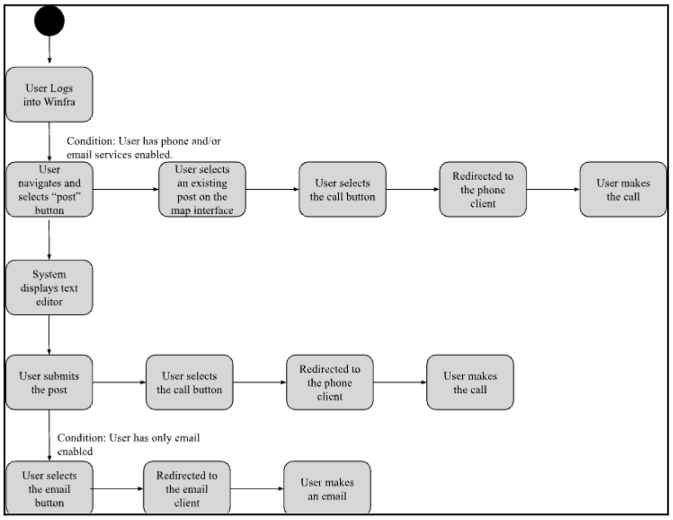
The class diagram above depicts the relationships between the classes implemented in the *Winfra* project. Solid lines indicate associations, while dotted lines indicate usage. Plus signs prepended to class members indicate public visibility. Multiplicity/cardinality is indicated either by numbers next to relationship lines or by asterisks, which indicate that one instance of some class can be related to many instances of another class.

Event Diagrams

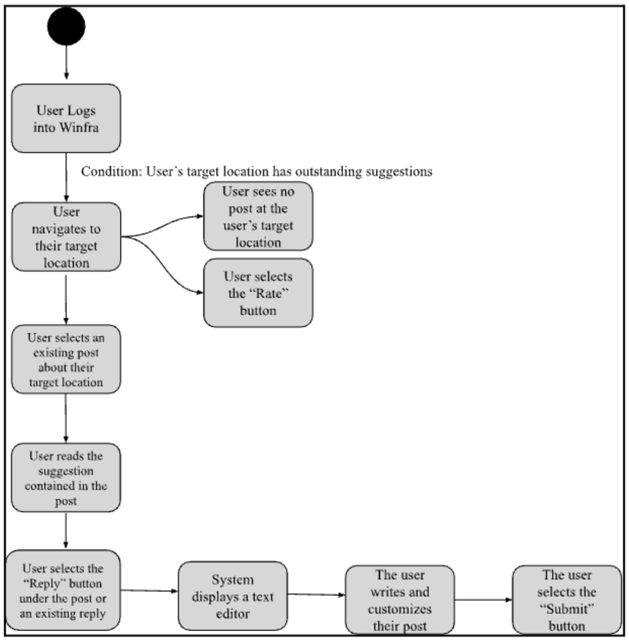
MIS-CNP


User Logs into Winfra
Condition: User has existing account and location services enabled
User navigates and selects "post" button
User selects an existing post on the map interface
System displays text editor
User writes and customizes their post
User writes and customizes their post

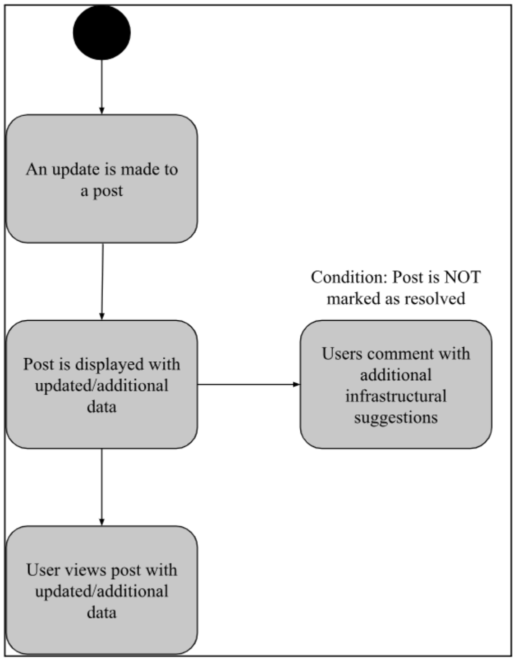
CTA


User Logs into Winfra
Condition: User has phone and/or email services enabled.
User navigates and selects "post" button
User selects an existing post on the map interface
User selects the call button
Redirected to the phone client
User makes the call
System displays text editor
User submits the post
User selects the call button
Redirected to the phone client
User makes the call
Condition: User has only email enabled
User selects the email button
Redirected to the email client
User makes an email

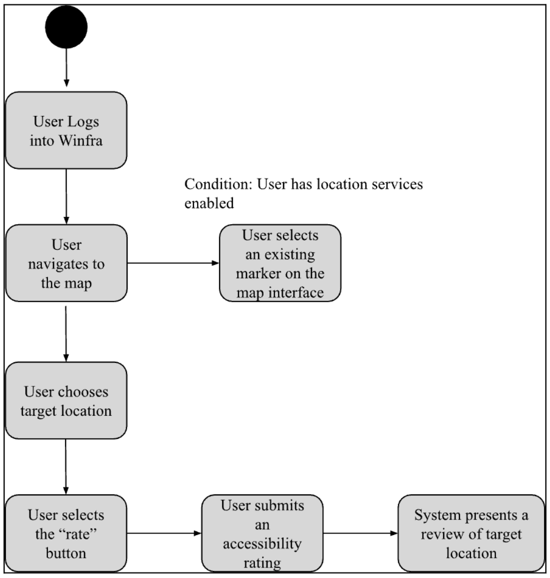
MIS-CUP


User Logs into Winfra
Condition: User's target location has outstanding suggestions
User navigates to their target location
User sees no post at the user's target location
User selects the "Rate" button
User selects an existing post about their target location
User reads the suggestion contained in the post
User selects the "Reply" button under the post or an existing reply
System displays a text editor
The user writes and customizes their post
The user selects the "Submit" button

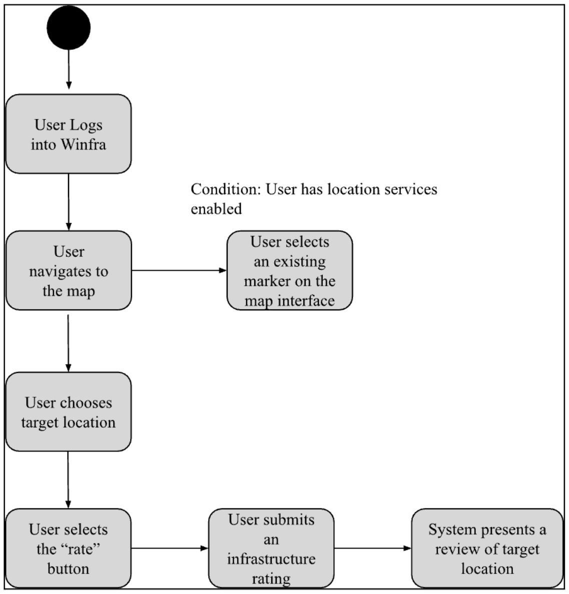
RTU


An update is made to a post
Condition: Post is NOT marked as resolved
Post is displayed with updated/additional data
Users comment with additional infrastructural suggestions
User views post with updated/additional data

MVA-VAR

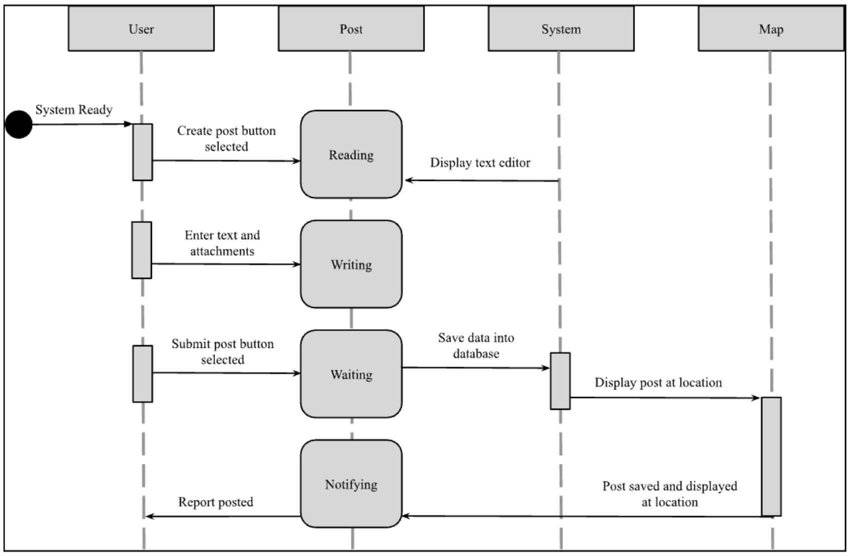

User Logs into Winfra
Condition: User has location services enabled
User navigates to the map
User selects an existing marker on the map interface
User chooses target location
User selects the “rate” button
User submits an accessibility rating
System presents a review of target location

MVA-VIR

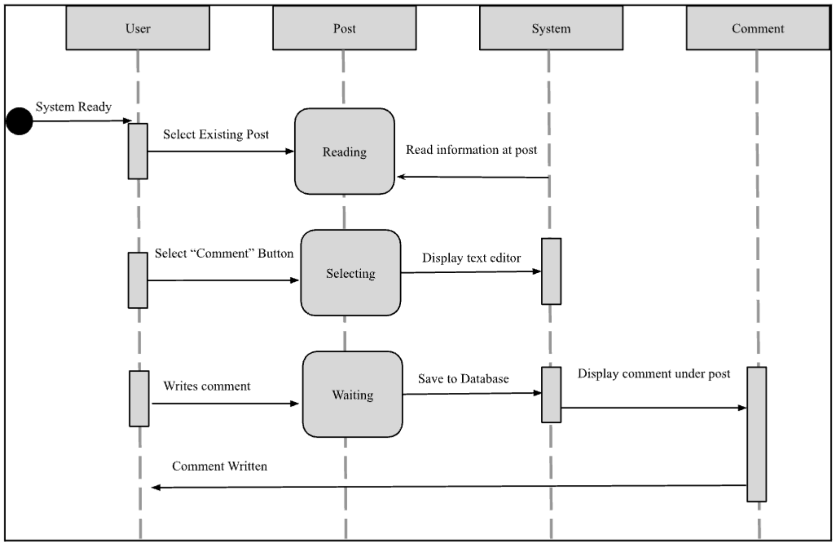

User Logs into Winfra
Condition: User has location services enabled
User navigates to the map
User selects an existing marker on the map interface
User chooses target location
User selects the “rate” button
User submits an infrastructure rating
System presents a review of target location

Sequence Diagrams

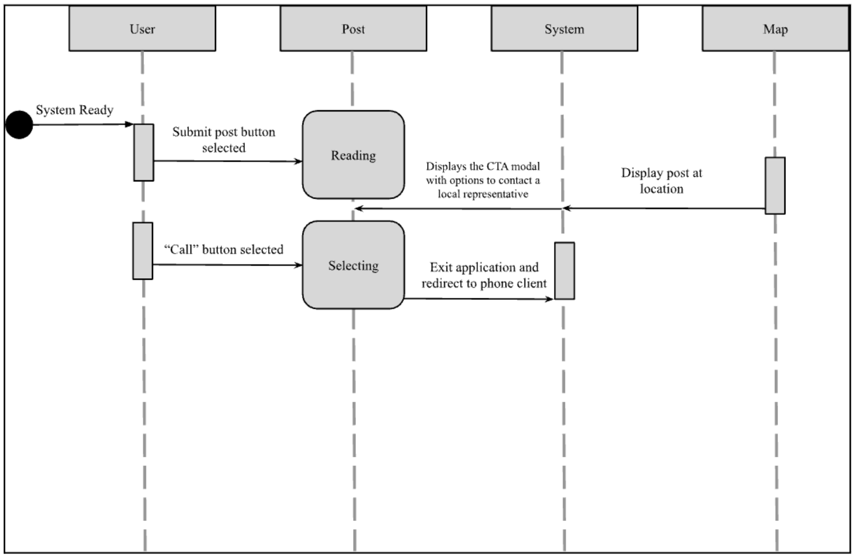
MIS-CNP



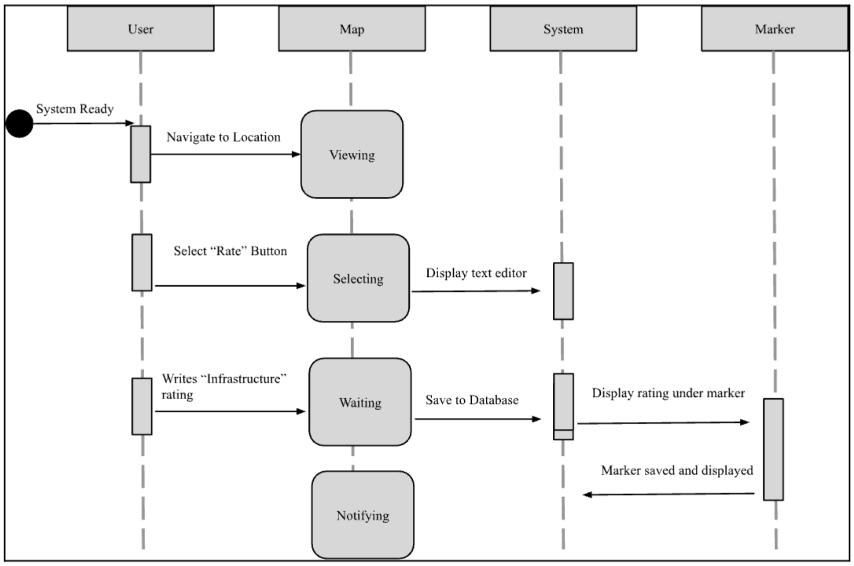
MIS-CUP



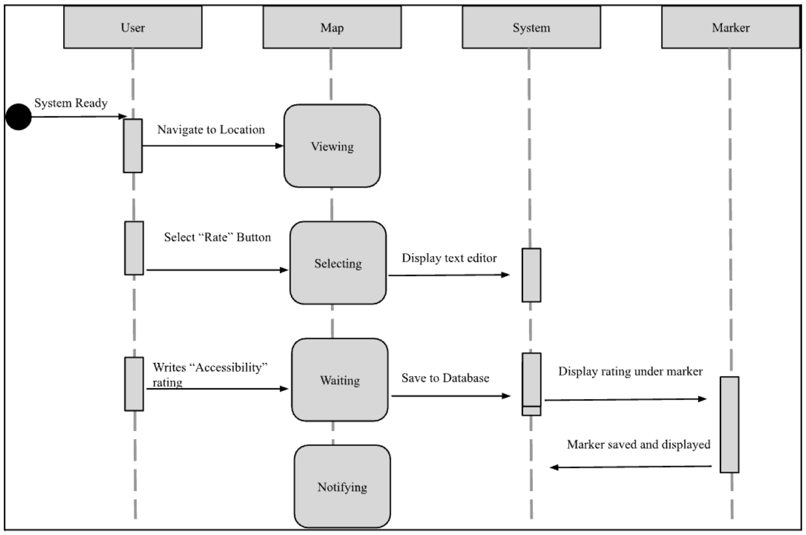
CTA


User
Post
System
Map
System Ready
Submit post button selected
Reading
Displays the CTA modal with options to contact a local representative
Display post at location
"Call" button selected
Selecting
Exit application and redirect to phone client

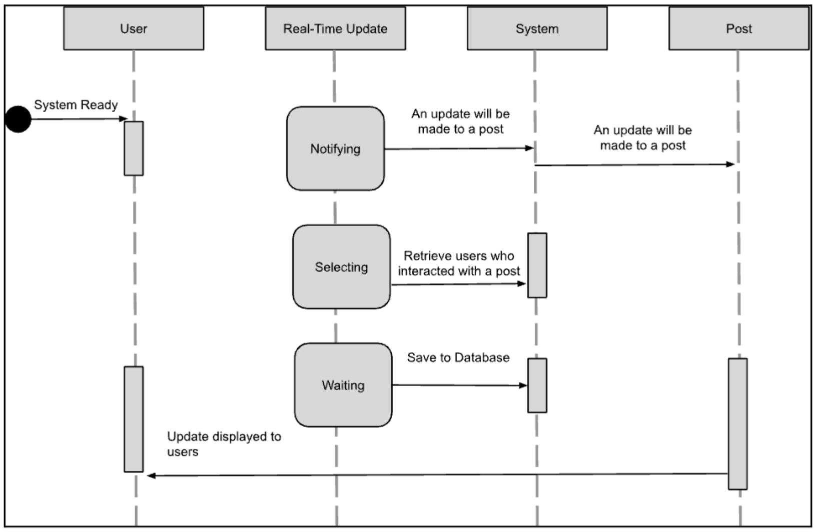
MVA-VIR


User
Map
System
Marker
System Ready
Navigate to Location
Viewing
Select "Rate" Button
Selecting
Display text editor
Writes "Infrastructure" rating
Waiting
Save to Database
Display rating under marker
Marker saved and displayed
Notifying

MVA-VAR


User
Map
System
Marker
System Ready
Navigate to Location
Viewing
Select "Rate" Button
Selecting
Display text editor
Writes "Accessibility" rating
Waiting
Save to Database
Display rating under marker
Marker saved and displayed
Notifying

RTU


User
Real-Time Update
System
Post
System Ready
Notifying
An update will be made to a post
An update will be made to a post
Selecting
Retrieve users who interacted with a post
Waiting
Save to Database
Update displayed to users

15.3. SCHEDULE TRACKING

Hours

Artifact or Deliverable	Who (individual or Team)	Estimated	Actual	Difference
SRS	Jenesis Blancaflor	3.5 hrs	6 hrs	+2.5 hrs
	Ruthvik Mukkamala	2 hrs	5 hrs	+3 hrs
	Sanjida Orpi	2 hrs	6 hrs	+4 hrs
	Elijah Wilson	2 hrs	5 hrs	+3 hrs
	Summary for the Entire Team	9.5 hrs	22 hrs	+12.5 hrs

Artifact or Deliverable	Who (individual or Team)	Estimated	Actual	Difference
SPMP	Jenesis Blancaflor	2 hrs	2 hrs	+0 hrs
	Ruthvik Mukkamala	2 hrs	1.5 hrs	-0.5 hrs
	Sanjida Orpi	2 hrs	2.5 hrs	+0.5 hrs

	Elijah Wilson	2 hrs	3.0 hrs	+1.0 hrs
	Summary for the Entire Team	8 hrs	9.0 hrs	+1.0 hrs

Artifact or Deliverable	Who (individual or Team)	Estimated	Actual	Difference
SDD	Jenesis Blancaflor	2 hrs	3 hrs	+1 hr
	Ruthvik Mukkamala	2 hrs	5 hrs	+3 hrs
	Sanjida Orpi	2 hrs	6 hrs	+4 hrs
	Elijah Wilson	4 hrs	11 hrs	+7 hrs
	Summary for the Entire Team	10 hrs	25 hrs	+15 hrs

15.4. DEFECT TRACKING

Counts

Artifact or Deliverable	Who (individual or Team)	Estimated	Actual	Difference
SRS	Jenesis Blancaflor	3 Counts	5 Counts	+2 Counts

	Ruthvik Mukkamala	4 Counts	5 Counts	+1 Count
	Sanjida Orpi	3 Counts	5 Counts	+2 Counts
	Elijah Wilson	4 Counts	4 Counts	0 Counts
	Summary for the Entire Team	14 Counts	17 Counts	+3 Counts

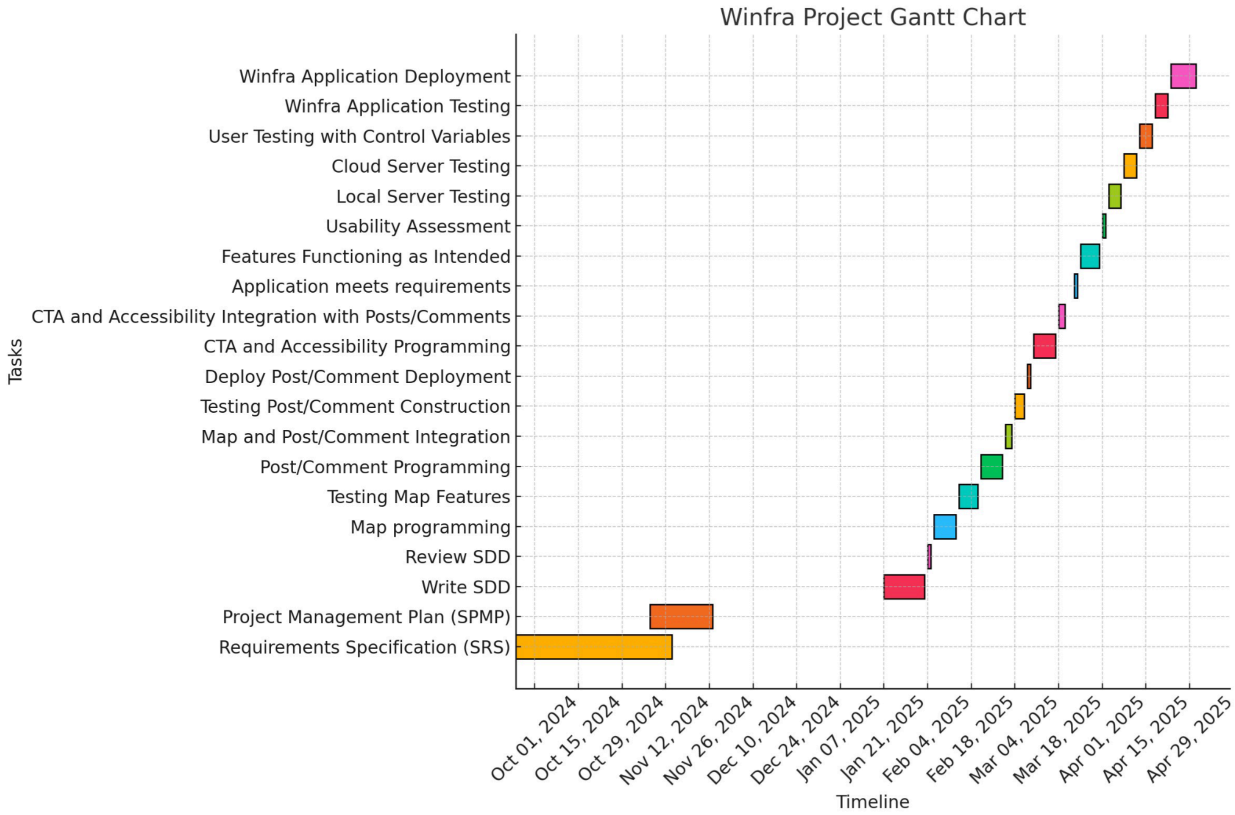
Artifact or Deliverable	Who (individual or Team)	Estimated	Actual	Difference
SPMP	Jenesis Blancaflor	2 Counts	2 Counts	+0 Counts
	Ruthvik Mukkamala	3 Counts	1 Count	-2 Counts
	Sanjida Orpi	2 Counts	0 Counts	-2 Counts
	Elijah Wilson	3 Counts	2 Counts	-1 Counts
	Summary for the Entire Team	10 Counts	5 Counts	-5 Counts

Artifact or Deliverable	Who (individual or Team)	Estimated	Actual	Difference
SDD	Jenesis Blancaflor	4 Counts	2 Counts	-2 Counts
	Ruthvik Mukkamala	4 Counts	5 Counts	+1 Count

	Sanjida Orpi	3 Counts	5 Counts	+2 Counts
	Elijah Wilson	4 Counts	6 Counts	+2 Counts
	Summary for the Entire Team	15 Counts	18 Counts	+3 Counts

15.5. GANTT CHART / MICROSOFT PROJECT SCHEDULE

Below is the Gantt chart for the work tasks of the *Winfra* project. The chart visualizes the project schedule that was originally described in *SPMP-002* §5.2.1.


Winfra Project Gantt Chart
Tasks
Winfra Application Deployment
Winfra Application Testing
User Testing with Control Variables
Cloud Server Testing
Local Server Testing
Usability Assessment
Features Functioning as Intended
Application meets requirements
CTA and Accessibility Integration with Posts/Comments
CTA and Accessibility Programming
Deploy Post/Comment Deployment
Testing Post/Comment Construction
Map and Post/Comment Integration
Post/Comment Programming
Testing Map Features
Map programming
Review SDD
Write SDD
Project Management Plan (SPMP)
Requirements Specification (SRS)
Oct 01, 2024
Oct 15, 2024
Oct 29, 2024
Nov 12, 2024
Nov 26, 2024
Dec 10, 2024
Dec 24, 2024
Jan 07, 2025
Jan 21, 2025
Feb 04, 2025
Feb 18, 2025
Mar 04, 2025
Mar 18, 2025
Apr 01, 2025
Apr 15, 2025
Apr 29, 2025
Timeline