Software Requirements Specification

for

CityMeet

Version 9 approved

Prepared by Darshan Patel, Elliot Gong, Alan Mai, Alejandro Vargas, Vivian Casas, Isidoro Flores, Deion Stapleton, Siying Chen

Group 5

30 April 2025

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Darshan Patel, Elliot Gong, Alan Mai, Alejandro Vargas, Vivian Casas, Isidoro Flores, Deion Stapleton, Siying Chen	2/28	Filled out the initial version of the Software Requirements Document for CityMeet.	1
Darshan Patel, Elliot Gong, Alan Mai, Alejandro Vargas, Vivian Casas, Isidoro Flores, Deion Stapleton, Siying Chen	3/5/2025	Add content for user personas, user journeys, and user stories. Better define functional and nonfunctional requirements/features.	2
Darshan Patel, Elliot Gong, Alan Mai, Alejandro Vargas, Vivian Casas, Isidoro Flores, Deion Stapleton, Siying Chen	3/12/2025	Add system architecture	3
Darshan Patel, Elliot Gong, Alan Mai, Alejandro Vargas, Vivian Casas, Isidoro Flores, Deion Stapleton, Siying Chen	3/19/2025	Update sections with new content from features and stories.	4

Darshan Patel, Elliot Gong, Alan Mai, Alejandro Vargas, Vivian Casas, Isidoro Flores, Deion Stapleton, Siying Chen	3/26/2025	Update sections with new content from features and stories.	5
Darshan Patel, Elliot Gong, Alan Mai, Alejandro Vargas, Vivian Casas, Isidoro Flores, Deion Stapleton, Siying Chen	4/8/2025	Update sections with new content from features and stories.	6
Darshan Patel, Elliot Gong, Alan Mai, Alejandro Vargas, Vivian Casas, Isidoro Flores, Deion Stapleton, Siying Chen	4/17/2025	Update sections with new content regarding features and stories.	7
Darshan Patel, Elliot Gong, Alan Mai, Alejandro Vargas, Vivian Casas, Isidoro Flores, Deion Stapleton, Siying Chen	4/22/2025	Update sections with new content regarding project updates and features.	8
Darshan Patel, Elliot Gong, Alan Mai, Alejandro	4/29/2025	Update sections with new content regarding project updates and features.	9

Vargas, Vivian Casas, Isidoro Flores, Deion Stapleton, Siying Chen			

1. Introduction

1.1 Purpose

This document aims to clarify the goals and projected outcomes of the CityMeet social media platform. As of now, this document highlights the frontend and backend resources that will be used to develop version 1 of CityMeet in addition to the scope of its user and system requirements.

1.2 Intended Audience and Reading Suggestions

This document is intended for those possessing a limited to intermediate understanding of web and social media development. This software requirements document contains information about the project features/scope and the tools/techniques needed to produce these components. The document intro and project overview can be found in Sections 1 and 2, which is then followed by the user requirements in Section 3. Sections 4 and 5 go over functional and nonfunctional system requirements while Section 6 acknowledges legal and ethical concerns and details.

Reading suggestions:

- Marketing - Introduction, Overall Description
- Ethics and Compliance - Legal and Ethical Considerations
- UX/UI - External Interface Requirements, Other Nonfunctional Requirements
- Developers/Programmers, Testers/Quality Assurance - Requirements Specifications, Other Nonfunctional Requirements

1.3 Product Scope

Version 1 of CityMeet will have the following features:

- Account creation and management
- Posting features
- Group creation/management
- Geospatial mapping and tracking algorithms for recommending content related to the users' area/region.
    - Won't require user addresses, only the city/zip code of their current location.
- Job posting/application features
- Access/links to social services provided by the Los Angeles city government.
- Messaging
- Search Filters
- Reporting/content moderation
- 2 factor authentication

1.4 Definitions, Acronyms, and Abbreviations

Glossary - Provides all the definitions, terms, and acronyms used in this SRD document.

1.5 References

Related Documents:

- CS 3337 Group 5 Project Proposals. https://docs.google.com/presentation/d/1zL862LbHzwp00Qv5E4HOdjGxCMRTBaEpJW9zUCYLK-k/edit#slide=id.p. - The CityMeet project's proposal document.
- Gong, Elliot, et al. Software Test Plan for Group 5 Version 1. Google Docs, 21 Feb. 2025, https://docs.google.com/document/d/1rr_VO-qoN1aIkb05d-1kpxe2-Py6Dr9wahwB9l4Xbvs/edit?tab=t.0. - A general outline of how CityMeet will be tested to ensure requirements are met.
- N/A. "PostgreSQL 17.4 Documentation." PostgreSQL Documentation, 20 Feb. 2025, https://www.postgresql.org/docs/17/index.html. - The relational database that can be used to store information about users and content.
- N/A. "Google Maps Platform Documentation." Google for Developers, https://developers.google.com/maps/documentation. Accessed 27 Feb. 2025. - The documentation page for the Google Maps API
- N/A. "Material Design." Material Design, https://m3.material.io. Accessed 27 Feb. 2025. - Google's open-source design system for web and mobile applications.
- N/A. Department of Public Social Services. https://dpss.lacounty.gov/en.html. Accessed 27 Feb. 2025. - The official website of the Los Angeles County Department of Public Social Services
- N/A. "California Consumer Privacy Act (CCPA)." State of California - Department of Justice - Office of the Attorney General, 15 Oct. 2018, https://oag.ca.gov/privacy/ccpa. - The official web page explaining the California Consumer Privacy Act.

2. Overall Description

This web application will be a centralized platform where individuals can learn where to access critical resources, engage with the local community, and explore new employment opportunities. We wish to integrate an interactive map to guide our users, as well as profiles where the users can engage, update, and connect with others in their community. As well as making a job board for users to apply for jobs. We want to make an intuitive interface that allows for broad and easy usage among our users that use our platform.

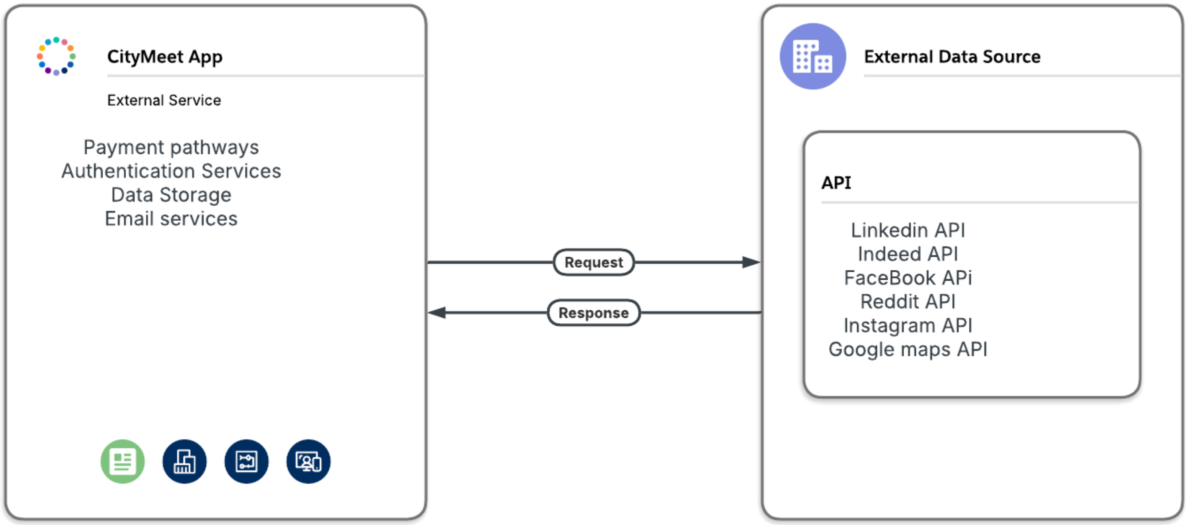
2.1 System Analysis

The problem we are trying to solve in Los Angeles County is that many individuals struggle to find essential services that the government provides. Things like food pantries, shelters, and community programs. Second, there is an absence of support and communication within local communities which makes it hard for people to share helpful information amongst each other. Lastly, we could help the unemployed look for jobs within the city. Jobs that range from the very top to the very bottom and want to be able to connect them and employ local individuals.

- Improve access to resources, enhance community engagement, and connect job seekers and employers.
- Challenges we might face include integrating and updating location data for maps, real-time updates and notifications, and ensuring data privacy and user authentication.
- By utilizing the Supabase database and integrating it with the Google Maps API we can get real time navigation. We will also use Postgresql to create profiles for users and connect the apis using next.js. For data privacy, we can implement a role-based access control for managing permissions and to also flag for inappropriate content on our platform.

2.2 Product Perspective

This web application is mostly independent from other products but it will integrate with several external services and APIs to improve the functionality of the product. It will interact with the Google Maps API, the Supabase Auth, PostgreSQL in Supabase, and can potentially incorporate the Linkedin or Indeed APIs for employment opportunities.



2.3 Product Functions

1. Accounts: Users can utilize a username/email and password to access and use the product and save data and settings to their preferences.
2. Content posting: Users can upload media files or write posts to share with other users.
3. Maps feature: We want the users to be able to search for nearby resources and other locations
4. Community engagement: The user will be able to create profiles and be able to post and comment in other discussions on the platform. We wish to create good and positive interactions.
5. Job listing and Applications: They will be able to browse job listings by location. You would also be able to submit an application directly through the platform and receive notifications when responded or other opportunities arise.
6. User Authentication: The sign up and login process will be through email or a third party authenticator.
7. Content moderation and Reporting: We want to ensure that it is safe and respectful by allowing users to report inappropriate content and we will also include fact checking to make sure no misinformation is being spread.

2.4 User Classes and Characteristics

As of right now we are going to have three main user classes in our web application CityMeet. The first one is Community Members which will be the general users which will be able to use the features that the app provides but they will have no moderation privileges. The next class which is higher will be local businesses and employers. We want to give them the required privileges to post job listings and manage them. We will also have profile verification so we can ensure they are legitimate. The highest class is the administrators they will moderate and be system managers. They will have access to logs and analytics to the system and review and moderate reported content. They can manage user accounts with the moderation tools we provide by doing account restrictions or bans.

2.5 Operating Environment

For the deployment environment this can be hosted on AWS and the backend server can be on Amazon linux. For the database we'll use Postgresql via Supabase. We would like for this application to be supported on mobile phones, desktop and laptops, and tablets. In the real world this application would be used in more urban areas where they can have access to libraries and community centers.

2.6 Design and Implementation Constraints

- Rules and Regulations:
  - Following privacy laws to ensure the protection of user data
  - Making sure that the platform is accessible to users who have disabilities
- Hardware Limits:
  - Ensuring servers can handle many users at once
  - Designing the app to work well on different devices, like phones and tablets
- Interfaces with other applications:
  - Integrate with maps for location tracking
  - Allowing the ability to share content on social media
- Parallel Operations:
  - Making sure the app works smoothly even with many users online
  - Use load balancing to distribute traffic evenly
- High-order language requirements
  - Use modern languages and frameworks for easier maintenance
  - Ensuring to follow the best coding practices and conduct code reviews
- Reliability requirements:
  - Keep the app running at all times
  - Implement error handling to fix issues quickly
- Safety and Security Considerations:
  - Protect user accounts with strong security measures
  - Regularly check for and fix security vulnerabilities
- Memory Constraints:
  - Manage memory efficiency to handle lots of data
  - Optimize database queries to reduce memory load

2.7 User Documentation

For our web application, CityMeet, we will provide several types of user documentation to help users understand and use the platform effectively. These will include the following:

- User manuals that will have detailed guides explaining how to use all the features of CityMeet. These manuals will be downloadable PDFs that can be acquired from the CityMeet website.
- Online Guides that show detailed step-by-step instructions that cover common tasks and features, and will be accessible through a dedicated "Help" section for easy access and download.
- Tutorials can also be made available and embedded in the "Help" sections for a more user-friendly experience.
- FAQs will list frequently asked questions as well as answers, this will also be available on the CityMeet website to help users quickly find solutions to common issues.

All the documents listed can be found and will be posted on the CityMeet website in the dedicated “help” or “support” section. With these documents being various types, such as PDFs, web pages, and videos, this will ensure that users have multiple resources to help navigate and make the most of CityMeet.

2.8 Assumptions and Dependencies

Some factors that may affect the requirements of the documents are the following:

- Operating systems: We plan to run CityMeet on common operating systems like Windows, macOS, iOS, and Android. If any of these operating systems are not available then we will need to update our requirements.
- Third-Party Components; We plan to use third-party services like Google for geospatial integration and user authentication. If these services change or become unavailable then alternatives will need to be considered.
- Browser compatibility; we assume that users will have access to CityMeet using up-to-date web browsers like Chrome, Firefox, Safari, and Edge. Any outdated browsers, can and may experience issues.
- Hardware Requirements; We assume that users’ devices will meet the minimum hardware requirements like having sufficient memory and processing power. The app may not perform optimally if users have older or less powerful devices.

These factors can affect the requirements and design of CityMeet should they be subject to change or become outdated thus, we would have to update our software requirements document accordingly.

2.9 Apportioning of Requirements

- Enhanced Security: adding a two-step authenticator to better secure users’ accounts and biometrics for easier ways to log in to profiles might be available in future updates.
- Customizable User Profiles; while adding basic user profiles will be implemented and available, more advanced customization options like themes, and personal information fields might be introduced in future updates
- Enhanced Media Sharing: Users will be able to share basic media such as photos and videos, but more advanced media sharing options like live streaming may be considered for future updates.
- Multilingual Support: While English will be available upon deployment of the web app, future updates may include additional languages
- Advanced Analytics; Features like detailed user engagement metrics and advanced reporting tools may also be considered for future updates

3. External Interface Requirements

3.1 User Interfaces

The user interface of the CityMeet platform will be designed to be intuitive and easy to access, ensuring a seamless experience on both web and mobile platforms.

- **Platform Compatibility:** The interface will be accessible through a website optimized for desktop and mobile browsers. A version of the Android mobile application may also be developed to provide a more interactive experience.
- **Accessibility:** The interface will be compliant with Americans with Disabilities Act standards to ensure accessibility for users with disabilities.
  - Provide compatibility with screen readers for visually impaired users.
  - Full keyboard navigation for users who are unable to use a mouse.
  - Visually impaired users with adjustable font sizes and high contrast modes.
- **Design Considerations:**
  - **Navigation Bar:** Each screen will have a consistent navigation bar that provides quick access to key features such as home, groups, posts, and profiles.
  - **Help Button:** Each screen will provide a help button where users can quickly access FAQs and support.
  - **Error Messages:** Error messages will be displayed in a clear and concise manner with suggestions for resolving the problem.
  - **Home:** Displays recent posts, trending topics, and updates on nearby events.
  - **Groups page:** Allows users to create, join, and manage groups. Each group will have a dedicated page for discussions and announcements.
  - **Post creation screen:** A simple interface for creating posts with options to add text, images, and location tags.

3.2 Hardware Interfaces

There are no specific hardware interface requirements for the CityMeet platform. However, the application will run on standard consumer devices such as PCs, laptops, and cell phones that can support web browser applications. Such devices must be able to support modern applications such as Google Chrome, Microsoft Edge, and Safari.

3.3 Software Interfaces

- **Geospatial API: Mapbox**
  - Version: The latest version available at the time of development.
  - Purpose: Provides geospatial integration, allowing users to tag locations in posts and search for nearby events or resources.
- **Social Media APIs (Facebook/Instagram/Snapchat):**
  - Version: The latest version available at the time of development.
  - Purpose: Allows users to share CityMeet posts to their social media accounts and integrate social media features into the platform.
- **Github**
  - Version: Current version
  - Purpose: Gives us a method for configuration and project management.
- **Vercel**
  - Version: 41.6.1. The current version.

    - Purpose: Lets us deploy our product from our Github repo to the web.
- **Supabase**
    - Version: The latest version available at the time of development.
        - MongoDB: version: 8.0.5
        - MySQL: version: 8.0
        - PostgreSQL: version: 17.0
    - Purpose: Used as the primary database for storing user data, posts, groups, and other platform-related information.
- **Node.js/Express.js:**
    - Version: Latest Long-Term Support version.
        - Node.js: v10.9.2
    - Purpose: Handles server-side logic and API integration.

3.4 Communications Interfaces

- **HTTP/HTTPS:** All communication between the client (web/mobile) and the server will be conducted over HTTPS to ensure data security and encryption.
- **Push Notifications:** For mobile apps, push notifications will be used to alert users about new posts, group activity, and emergency alerts.
- **Email Integration:** Sends account verification emails, password reset links, and notifications for users who opt-in for email reminders.
- **Data Transfer Rates:** The platform will be optimized to handle high data transfer rates, especially during peak usage, to ensure smooth performance.
- **Message Notifications:** If toggled, the user will receive a notification of any messages received from other users.
- **Security:**
    - All sensitive data (passwords, personal information) will be encrypted during transmission and storage.
    - User authentication will be handled through Supabase, which provides secure authentication mechanisms.

4. Requirements Specification

4.1 System Overview

The CityMeet website shall provide users with the ability to create and manage accounts. The opportunity to integrate for two-factor authentication shall also be available for users. The system shall allow users to post their own content and search for their desired content. If necessary, users shall have the chance to report/flag undesired/inappropriate content. The system shall integrate Geospatial Mapping and provide connections/links to official LA City welfare/social services. Users shall be able to directly message each other. Users shall be able to create and join groups/forums. Businesses/employers who own accounts shall be able to post job listings and accept applications for such advertised positions.

4.1.1 Maps

- **4.1.1.1** The system shall integrate a mapping service
    - **4.1.1.1.1** Neighborhood boundaries.
    - **4.1.1.1.2** User locations (with user consent).

    - **4.1.1.1.3** Locations of community events and job listings.
    - **4.1.1.1.4** Searchable points of interest within the neighborhood with keywords.
- **4.1.1.2** The system shall integrate map API authentication and error handling gracefully.

4.1.2 Community Engagement

- **4.1.2.1** The system shall provide a forum or discussion board interface for neighborhood communication.
    - **4.1.2.1.1** Users shall be able to create new threads and reply to existing posts.
    - **4.1.2.1.2** Users shall be able to search for content within the forums.
    - **4.1.2.1.3** The system shall support threaded discussions.
- **4.1.2.2** The system shall enable users to create and manage accounts.
    - **4.1.2.3.1** Users shall be able to upload profile pictures and add personal information.
    - **4.1.2.3.2** Users shall be able to control the visibility of their profile information.
    - **4.1.2.3.3** User profiles will show neighborhood affiliation.

4.1.3 Job Postings and Applications / Welfare Locator

- **4.1.3.1** The system shall provide an interface for employers to create and manage job listings.
    - **4.1.3.1.1** The interface shall include fields for job title, description, location, salary, and application instructions.
    - **4.1.3.1.2** The system shall allow employers to manage applications received.
    - **4..3.1.3** The system shall allow users to find filtered social services
- **4.1.3.2** The system shall provide a job search interface for users.
    - **4.1.3.2.1** Users shall be able to search for jobs based on keywords, location, and category.
    - **4.1.3.2.2** The system shall display job listings with relevant information and application links.
    - **4.1.3.2.3** Users shall be able to save job listings and receive job alerts.
    - **4.1.3.2.4** Users shall be able to have navigation routes displayed based on social services
    - **4.1.3.2.5** Users shall be able to search for social services with keywords and phrases

4.1.4 User Authentication

- **4.1.4.1** The system shall implement secure user authentication (e.g., OAuth, JWT).
- **4.1.4.2** The system shall support user registration and login.
- **4.1.4.3** The system shall provide password recovery functionality.
- **4.1.4.4** The system shall encrypt user credentials.

4.1.5 Content Moderation and Reporting

- **4.1.5.1** The system shall provide content moderation tools for moderators
- **4.1.5.2** Users shall be able to report inappropriate content
- **4.1.5.3** Administrators shall be able to review and respond to user-reported content
- **4.1.5.4** The system shall log moderation actions

4.1.6 Handling Abnormal Cases

- **4.1.6.1 User Authentication Errors**
    - **4.1.6.1.1** The system shall display a clear and informative error message to the user if they enter incorrect login credentials (e.g., "Invalid username or password").

  - **4.1.6.1.2** The system shall implement a lockout mechanism after a specified number of failed login attempts to prevent brute-force attacks.
  - **4.1.6.1.3** The system shall provide a password recovery mechanism for users who have forgotten their passwords.
  - **4.1.6.1.4** The system shall log all failed login attempts, including the timestamp and IP address.
- **4.1.6.2 Mapping Service Errors**
  - **4.1.6.2.1** If the mapping service is unavailable, the system shall display a message to the user indicating that the map cannot be loaded.
  - **4.1.6.2.2** If the mapping service returns an error due to an invalid request, the system shall log the error and display a generic error message to the user.
  - **4.1.6.2.3** If map data is partially loaded, the system shall display the loaded data and indicate that some data may be missing.

4.1.7 Messaging/ Communications Feature

- **4.1.7.1** The system shall provide a comprehensive messaging service
  - **4.1.7.1.1** The system shall have message sending and receiving capabilities
  - **4.1.7.1.2** The system shall have group chat functionalities
  - **4.1.7.1.3** The system shall have multimedia attachments to send and receive
  - **4.1.7.1.4** The system shall have an implemented message search with keywords
- **4.1.7.2** The system shall integrate secure messaging protocols and error handling optimally.

4.1.8 Two-Factor Authentication (2FA) Requirements

- **4.1.8.1 2FA Setup**
  - **4.1.8.1.1** The application shall provide users with the option to enable Two-Factor Authentication (2FA) from their account settings.
  - **4.1.8.1.2** The application shall support authentication methods including authentication apps (Google Authenticator, Authy), security keys, and SMS-based authentication.
  - **4.1.8.1.3** The system shall generate a QR code or setup key for users choosing an authentication app for 2FA setup.
  - **4.1.8.1.4** Users shall be required to confirm 2FA setup by entering a verification code before activation is completed.
- **4.1.8.2 Login Process with 2FA**
  - **4.1.8.2.1** If 2FA is enabled, the application shall prompt users to enter a verification code after successfully entering their password.
  - **4.1.8.2.2** The system shall verify the entered 2FA code before granting access to the user account.
  - **4.1.8.2.3** If an incorrect 2FA code is entered, access shall be denied, and an error message shall be displayed.
  - **4.1.8.2.4** Users shall have the option to mark a device as trusted, reducing repeated 2FA prompts for future logins.
- **4.1.8.3 Backup and Recovery Options**
  - **4.1.8.3.1** The system shall generate and provide users with a set of recovery codes upon enabling 2FA.
  - **4.1.8.3.2** Recovery codes shall be used to regain account access in the event of lost access to the authentication method.

  - **4.1.8.3.3** The system shall allow users to update or reset their 2FA settings after verifying their identity via email confirmation.

4.1.9 User Accounts

- **4.1.9.1 Logging in Logging Out**
  - **4.1.9.1.1** The CityMeet website shall grant access CityMeet features upon a successful login.
  - **4.1.9.1.2** Logging out should disable access to CityMeet features and account info.
  - **4.1.9.1.3** A new account session and data will be created when I log in.
  - **4.1.9.1.4** My account session and sensitive data will securely close/terminate when I log out.
  - **4.1.9.1.5** Session tracking should show all the periods correlating to when users log in and out of the system.
- **4.1.9.2 Signing Up**
  - **4.1.9.2.1** The customer shall be able to sign up by creating and saving a username/password pair.
  - **4.1.9.2.2** The customer shall be able to input other data that will be relevant to the type of account they will have.
  - **4.1.9.2.3** The username/password pair shall be saved in some external database/storage.
  - **4.1.9.2.4** The username/password pair shall be used to gain access to the CityMeet platform.
- **4.1.9.3 Deleting Accounts**
  - **4.1.9.3.1** Users shall verify their identity before deleting their account.
  - **4.1.9.3.2** CityMeet shall ensure users read and accept the confirmation/dialog before proceeding with their decision.
  - **4.1.9.3.3** Users shall be able to cancel their account deletion before their final submission.
  - **4.1.9.3.4** Once an account is deleted, the credentials attached to that account shall no longer function.
  - **4.1.9.3.5** Once an account is deleted, CityMeet will hold onto some of its data before permanently deleting it.

4.1.10 User Posting/Content Uploads

- **4.1.10.1 Create a Post**
  - **4.1.10.1.1** The system shall allow users to create posts with text, images, and videos.
  - **4.1.10.1.2** The system shall validate file types and sizes.
  - **4.1.10.1.3** The system shall display a preview of the post before submission.
  - **4.1.10.1.4** The system shall provide a confirmation message when the post is successfully uploaded.
- **4.1.10.2 Tag Posts with Location**
  - **4.1.10.2.1** The system shall allow users to tag posts with location data (e.g. city, zip code).
  - **4.1.10.2.2** The system shall display the location tag on the post for other users to see.
- **4.1.10.3 Categorize Posts**
  - The system shall provide tags for users to categorize their posts.
  - The system shall display the category on the post for other users to see.

- **4.1.10.4 Edit or Delete Posts**
  - The system shall allow users to edit the text, images, or videos in their posts.
  - The system shall allow users to delete their posts.
  - The system shall confirm with the user before deleting a post.

4.1.11 Content Reporting and Flagging

- **4.1.11.1 Report Button**
  - A report button would be added in a place easy to access to report the post or comment
- **4.1.11.2 Report Context**
  - Reason and Explanation would be an option for users to input to explain the context or any other additional information
- **4.1.11.3 Review of the Report**
  - Once submitted, moderators and admins would review the submission to decide if action will be taken against the flagged content.

4.1.12 Links to Los Angeles City Social Services

- **4.1.12.1 Find offered welfare services**
  - The CityMeet website shall flag city points of interest that support city government welfare/assistance.
  - Each welfare/social assistance facility shall provide links to dedicated web pages/profiles for more in-depth information.
- **4.1.12.2 Connect with a live government spokesperson**
  - The call/message feature will directly connect users with an official spokesperson/representative.
  - The system shall allow users to select the option in which they desire to communicate.
  - The system shall log the communication in the user's account history.
- **4.1.12.3 Update official government content**
  - Based on the account currently in use, the user shall be able to update official government content on CityMeet.
  - The system shall perform another identity verification to ensure the person making these updates is an official government employee/official.

4.2 External Interface Requirements

A. User Authentication Interface

   A. **Name of item:** User Login

   B. **Description of purpose:** To authenticate users and grant access to the system.

   C. **Source of input:** User input (username/email, password).

   D. Valid range, accuracy, and/or tolerance:

    a. Username/email: Valid email format or alphanumeric string.
    b. Password: Minimum length, character complexity (e.g., alphanumeric, special characters).

  E. **Units of measure:** N/A.
  F. **Timing:** Real-time authentication upon user submission.
  G. **Relationships to other inputs/outputs:**
  H. **Input:** User registration data.
  I. **Output:** Session token or error message.
  J. **Screen formats/organization:** Login form with username/email and password fields.
  K. **Window formats/organization:** Modal or dedicated login page.
  L. **Data formats:** JSON (for API), HTML form data.
  M. **Command formats:** POST request to `/api/login`.
  N. **End messages:** "Login successful," "Invalid credentials," "Account locked."

B. Geospatial Mapping API Interface
  A. **Name of item:** Map Display
  B. **Description of purpose:** To display interactive maps with neighborhood boundaries, user locations, and event markers.
  C. **Source of input:** System requests for map data, user interactions (zoom, pan).
  D. **Destination of output:** Web browser display.
  E. **Valid range, accuracy, and/or tolerance:**
  F. Latitude/longitude: Valid geographic coordinates.
  G. Zoom level: Defined range by the mapping API.
  H. **Units of measure:** Degrees (latitude/longitude), zoom level (integer).
  I. **Timing:** Asynchronous loading of map tiles and data.
  J. **Relationships to other inputs/outputs:**
    a. Input: Neighborhood boundary data, event location data, user location data.
    b. Output: Map tiles, markers, and interactive elements.
  K. **Screen formats/organization:** Map embedded within the web page.
  L. **Window formats/organization:** Full-screen map or map section within a page.
  M. **Data formats:** JSON (from API), map tile images.
  N. **Command formats:** API requests to the mapping service.
  O. **End messages:** "Map loaded," "Map service unavailable."

C. Job Listing API Interface
  A. **Name of item:** Job Search
  B. **Description of purpose:** To enable users to search and view job listings.
  C. **Source of input:** User search queries (keywords, location, category).
  D. **Destination of output:** Web browser display of job search results.
  E. **Valid range, accuracy, and/or tolerance:**
  F. Keywords: Alphanumeric strings.
  G. Location: Valid address or coordinates.
  H. Category: predefined categories.
  I. **Units of measure:** N/A.
  J. **Timing:** Real-time search and display of results.
  K. **Relationships to other inputs/outputs:**
    a. Input: job database.
    b. Output: job search results.
  L. **Screen formats/organization:** Search bar and job listing display.

M. **Window formats/organization:** Dedicated job search page.
N. **Data formats:** JSON (for API).
O. **Command formats:** API requests to `/api/jobs/search`.
P. **End messages:** "Search results found," "No results found."

D. Email API Interface
  a. **Name of item:** Event Notification
  b. **Description of purpose:** To send email notifications for event reminders and updates.
  c. **Source of input:** System-generated event data.
  d. **Destination of output:** User email inbox.
  e. **Valid range, accuracy, and/or tolerance:**
    i. Email address: Valid email format.
    ii. Message content: text.
  f. **Units of measure:** N/A.
  g. **Timing:** Scheduled or triggered event-based emails.
  h. **Relationships to other inputs/outputs:**
  i. Input: Event data, user email address.
  j. Output: Email delivery status.
  k. **Data formats:** JSON (for API).
  l. **Command formats:** API requests to the email service.
  m. **End messages:** "Email sent," "Email delivery failed."

E. Database interface
  a. **Name of item:** Data Storage and Retrieval
  b. **Description of purpose:** To store and retrieve user data, neighborhood information, event details, and job listings.
  c. **Source of input:** System data, user input.
  d. **Destination of output:** System data, user display.
  e. **Valid range, accuracy, and/or tolerance:** Based on the logical database requirements.
  f. **Units of measure:** N/A.
  g. **Timing:** Database queries are executed upon system requests.
  h. **Relationships to other inputs/outputs:** All parts of the system rely on the database.
  i. **Data formats:** SQL queries and database responses.
  j. **Command formats:** SQL queries (SELECT, INSERT, UPDATE, DELETE).
  k. **End messages:** Database query results, error messages.

4.3 Logical Database Requirements

- Account
  - Name
  - Emails
  - Address
  - Location
  - Biometric data
- Posts
  - Content(image(s)/video(s)/text)

- Caption
- Tags
- Likes/Dislikes
- Comments

- Groups
  - Members(Elected/official positions)
  - Group rules(member limit, age range)
  - Group Posts(announcements, polls, etc.)
  - Group settings
- Job Listings/Applications
  - Location
  - External Links
  - Resume/CV upload
  - Position information
- 2 Factor Authentication
  - The database shall store 2FA status for each user(Enabled/Disabled)
  - The database shall store encrypted recovery codes for each user
  - The database shall log 2FA authentication attempts with timestamps and status(success/failure)

4.4 Design Constraints

There are some constraints with access to certain APIs like for Google Map, as some may be locked behind a pay hall. As the application gains more traction the need to scale up the database might be needed. Use a relational database for the backend as information will not be changing often This will help with having structured data and enforce data integrity.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- The software shall support up to 500 terminals
- The system must handle 1000 simultaneous users without performance degradation
- 95% of user queries shall be answered within 2 seconds
- The system shall manage and store up to 100 MB of data

5.2 Safety Requirements

- Possible loss, damage, or harm
  - Identify potential risks associated with the use of the product.
  - Implement measures to prevent loss, damage, or harm to users and property.
- Safeguard and actions
  - Establish robust encryption protocols to protect sensitive data.
  - Enforce multi-factor authentication (MFA) for user access.
  - Develop and maintain an incident response plan for data breaches.
- Preventative actions
  - Regularly update software to address security vulnerabilities.
  - Conduct periodic security audits and vulnerability assessments.
  - Anonymize user data to minimize the impact of data breaches.
- External policies and regulations
  - Ensure compliance with relevant data protection laws (e.g., GDPR, CCPA).
  - Adhere to industry standards and best practices for data security.
- Safety certifications
  - Obtain necessary safety certifications (e.g., ISO/IEC 27001) to validate the security measures implemented.
  - Conduct regular training sessions for employees on data protection and security practices.

5.3 Security Requirements

Security and Privacy Issues:

- Implement data encryption for data in transit and at rest to protect sensitive information.
- Ensure compliance with data protection regulations (e.g., GDPR, CCPA) to maintain user privacy and security.
- The application shall enforce strong authentication mechanisms, including Two-Factor Authentication (2FA).
- The application shall ensure that 2FA data, including recovery codes, is stored securely using encryption.
- The application shall comply with industry security standards for authentication, such as NIST 800-63 guidelines.
- The application shall implement rate-limiting mechanisms to prevent brute-force attacks on 2FA verification codes.

User Identity Authentication:

- Require multi-factor authentication (MFA) for all user accounts to enhance security.
- Implement role-based access control (RBAC) to ensure that users have access only to the data and functions necessary for their roles.

External Policies and Regulations:

- Adhere to industry standards and best practices for security (e.g., NIST, ISO/IEC 27001).
- Regularly review and update security policies to comply with evolving regulations and threats.

Security and Privacy Certifications:

- Obtain relevant security certifications (e.g., ISO/IEC 27001) to demonstrate compliance with industry standards.
- Conduct regular security audits and assessments to ensure ongoing compliance and identify potential vulnerabilities.

5.4 Software Quality Attributes

Adaptability:

- The system shall support integration with at least three external systems without significant changes to the codebase.

Availability:

- The software must achieve 99.9% uptime, ensuring minimal downtime and high availability for users.

Correctness:

- All calculations and data processing must be verified with a 0% error rate in critical functionalities.

Flexibility:

- The system should allow customization of user interfaces and workflows to accommodate different user needs.

Interoperability:

- The software shall support standard communication protocols (e.g., HTTP, REST) to interact seamlessly with other systems.

Maintainability:

- The codebase shall follow best practices for readability and modularity, allowing updates and maintenance with minimal disruption.

Reliability:

- The system must have an MTBF (Mean Time Between Failures) of at least 1,000 hours, ensuring reliable operation.

Re-usability:

- At least 50% of the code should be reusable in other projects or components to maximize efficiency.

Robustness:

- The software must handle unexpected inputs and conditions gracefully without crashing or producing incorrect results.

Testability:

- The system shall be designed to facilitate automated testing, with at least 80% code coverage by unit tests.

Usability:

- The user interface should be intuitive and user-friendly, achieving a satisfaction score of 4.5 out of 5 in user surveys.

5.5 Business Rules

Role-Based Access:

- **Administrators:** Can access and modify all system settings, perform user management tasks, and view all data.
- **Managers:** Can approve or reject user actions, view reports, and modify specific project-related settings.
- **Users:** Can perform tasks related to their specific roles and access data relevant to their tasks only.

Approval Workflows:

- Certain actions (e.g., data export, and system configuration changes) must be approved by a manager before execution.
- Project creation and deletion must be approved by an administrator.

Data Retention:

- User activity logs must be retained for a minimum of 6 months for audit and compliance purposes.
- Financial data must be archived annually and retained for at least 7 years.

Security Protocols:

- Passwords must be changed every 90 days and must meet complexity requirements (e.g., minimum 8 characters, including a mix of letters, numbers, and symbols).
- User accounts will be locked after 5 unsuccessful login attempts and require administrator intervention to unlock.

Operational Hours:

- System maintenance will occur every Sunday from 2 AM to 4 AM, during which the system may be unavailable.
- Customer support is available from 8 AM to 8 PM, Monday to Friday.

Data Privacy:

- All personal data must be anonymized before being used for analytics or reporting.
- Users must be informed of any changes to data privacy policies and obtain their consent where necessary.

6. Legal and Ethical Considerations

- Data Privacy
  Firstly, there are a couple of laws and regulations our web application will take into consideration. The first is the EU's General Data Protection. We will do our best to protect consumers' data and safeguard against bad actors but also obtain consent before collecting data and strictly comply with user wishes to delete or obtain their data. Similarly, we will be complying with the CCPA, California Consumer Privacy Act. As a result of our app including wellness or welfare features in the event that we implement some health features in the future we will be following the Health Insurance and Accountability Act, which are strict guidelines that protect user health data.

- Security

  To ensure protection for our users we will be following the global standard in Security. ISO 27001 is a global standard for information and security management systems. PCI-DSS is the industry standard for protecting sensitive payment card details. Additionally, our web app will be adhering to the OWASP guidelines, which were established to prevent common vulnerabilities like SQL injections.

- Accessibility

  Given the short time constraints, we will do our best to make the web application as accessible as possible but will not strictly follow all guidelines at the moment, however with more time we will do our best to strictly adhere to these policies. Now with that in mind, we will do our best to follow the WCAG or web content accessibility guidelines.

- Consumer Protection and E-commerce Laws

  In the case that we establish an e-commerce aspect to our application or sell products to our users we will be following ftc guidelines to protect users from fraud.

  We will also be complying with e-commerce laws that allow for refunds, taxes, etc.

- Ethical Concerns

  By holding our developers to a stringent standard in adhering to policies that were established to protect consumer data, I think we have put to rest the most ethical concerns when it comes to data protection. However, given the nature of our app, I think there are some ethical concerns that may arise. For starters, some level of app att tracking will be required to better optimize our user experience. Environmental impact is also a major concern, however, we are not including AI elements in our app, and as a result, our carbon footprint is smaller if not similar to most websites.

Another concern may be the digital divide where varying socioeconomic levels can impact a user's experience. However, we decided to do a web application for this exact reason. A user can access our app from a phone, tablet, or computer.

Appendix A: Glossary

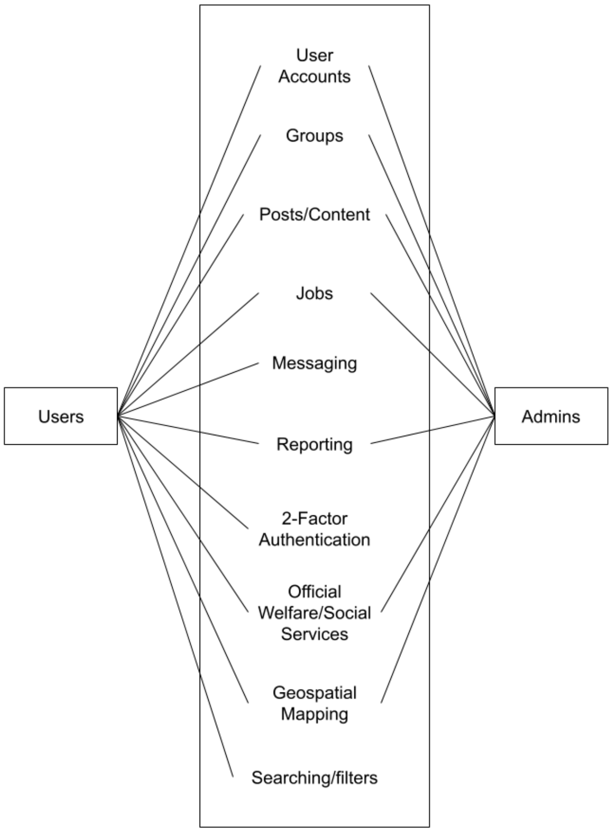
Definitions:

- **Adaptability**- the ability of a system to adjust to changing conditions
- **API**- Application Programming Interface is a set of rules that allows software applications to communicate with each other
- **Availability-** a measure of how often a system is operational and accessible
- **AWS**- a cloud computing platform that offers services for storage, computing, databases, and more
- **Biometrics-** the process of using unique physical or behavioral characteristics to identify people
- **CCPA-** California Consumer Privacy Act is a state law that protects the personal data of California Residents
- **CityMeet** - A web-based social media platform that doubles as a community outreach tool supported by the Los Angeles city government.
- **Data Retention**- the practice of storing for a specific period of time
- **Downtime**- when a computer system is unavailable or not working
- **Encryption-** the process of converting information or data into a code, especially to prevent unauthorized access.
- **FAQs**- a collection of questions and answers about a product, service, or topic
- **Flexibility-** the ability of any major unit to be connected by a data path to any other major unit within a digital computer.
- **Interoperability**- the ability of different systems to communicate and exchange information
- **ISO/IEC**- an international standard to manage information security
    - International Organization for Standardization (ISO)
    - International Electrotechnical Commission(IEC)
- **GDPR-** General Data Protection Regulation is an European Union(EU) law that protects the personal data of EU residents
- **Geospatial:** relating to or denoting data that is associated with a particular location
- **Middleware** - Computer software that provides services for user/client devices and external/server applications.
- **NIST**- National Institute of Standards and Technology is a non-regulatory laboratory and agency within the U.S. Department of Commerce
- **PostgreSQL** - A free and open-source relational database management system that supports SQL and JSON
- **Project-scope:** a document that outlines the work required to complete a project

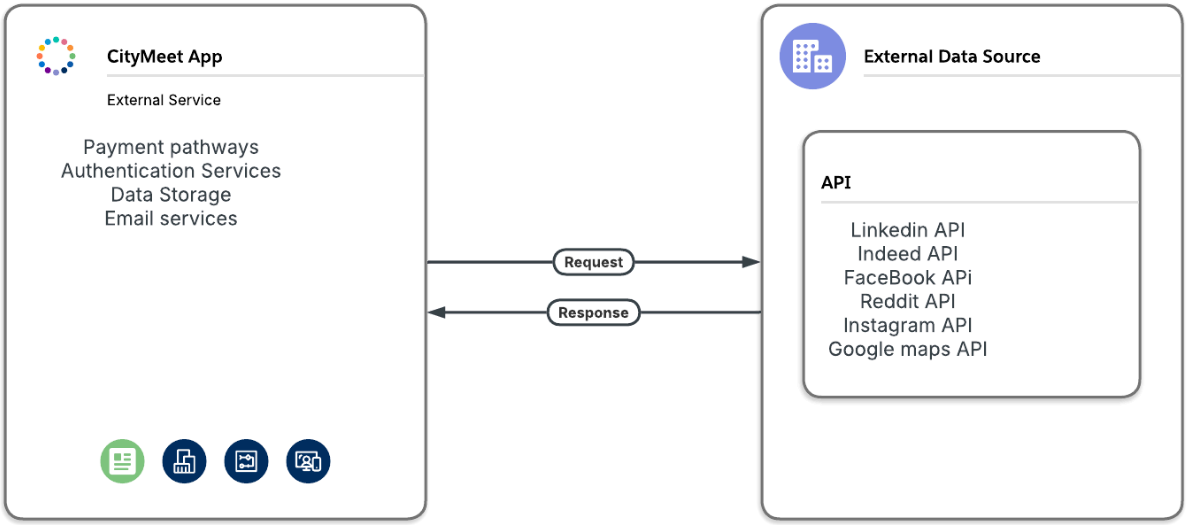
- **Query(Queries)-** a request for information from a database or information system
- **Robustness-** the ability of a system to handle errors and unexpected input
- **Supabase**- an open-source Backend platform that helps developers manage databases and backends
- **Terminals-** a text-based interface that lets users interact with a computer by typing commands
- **Uptime**- the amount of time a computer or service is available and operational
- **WCAG**- Web Content Accessibility Guidelines are international standards that help make websites accessible to people with disabilities
- **Web-base**- something that is accessible through a web browser or the internet

Appendix B: Analysis Models

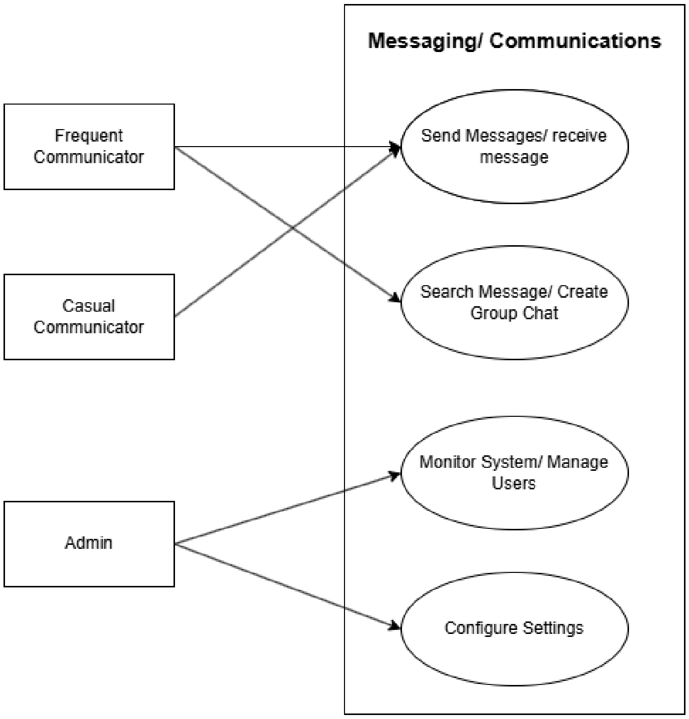
- Use Case Diagram



- User interaction with Map api and IT Admin



- Product perspective graph


Messaging/ Communications
Frequent Communicator
Casual Communicator
Admin
Send Messages/ receive message
Search Message/ Create Group Chat
Monitor System/ Manage Users
Configure Settings

Appendix C: To Be Determined List

1. Backlog Creation
    a. Work with stakeholders to prioritize features and functionality. We will convert stakeholder requirements into User Stories to identify user activities that must be possible with our product.
2. Sprint Planning
    a. Prioritize stories to be developed in the upcoming sprints
3. Prototyping
    a. Create initial prototypes to receive feedback from our end-users/stakeholders
4. Product Development
    a. Our developers will work on the prioritized user stories in iterations.
5. Testing and Quality Assurance
    a. Run unit, component, and system tests to monitor progress of software.
6. Sprint Review
    a. Showcase completed features to stakeholders for feedback.
    b. Based on feedback, we shall adjust and reprioritize the back;og based on the changing user requirements

7. Continuous Development and Iteration
    a. We will release features incrementally to allow continuous user feedback, have the ability to respond faster to changing needs, and prioritize features based on real-environment usage data