

# Screen Sketches

For “CityWatcher”

By 2\_jabir 2: Nicholas, Sam, Samuel

# Actors

- **Citizen**
  - Report Issue: Submit a new problem in the city with photo, description, and location.
  - View Issues: See a list of reported issues, including their own.
  - Track Issue Status: Monitor the progress of reported issues.
  - Receive Notifications: Get updates on issue resolutions or status changes.
  - Filter Issues: Sort and filter issues by various criteria (e.g., category, status, location).
  - Add Comments: Provide additional information or updates on existing issues.
- **City Official (Manager)**
  - View Assigned Issues: Access a list of issues assigned to them.
  - Update Issue Status: Change the status of issues (e.g., "Under Review," "In Progress," "Resolved").
  - Add Internal Notes: Attach private comments or updates to issues.
  - Prioritize Issues: Rank issues based on urgency or importance.
  - Assign Tasks: Delegate specific tasks related to issue resolution.
- **Administrator**
  - Manage User Accounts: Create, modify, or delete accounts for citizens and city officials.
  - Define Issue Categories: Create and manage categories for issue classification.

## Non-Functional Requirements

1. **Performance**
  - The mobile app must load and display the main screen within 3 seconds on average Android devices.
  - The user should not have to wait more than 5 seconds for confirmation that their issue report has been submitted successfully.
  - Any UI element in the app should respond within 0.5 seconds of user interaction.
2. **Scalability**
  - The system must support a minimum of 20 concurrent active users.
  - The database should be able to store and efficiently retrieve at least 1000 issue reports.
3. **Usability**
  - The mobile app interface should be intuitive, allowing new users to report an issue within 3 minutes of first use.
  - Error messages should be clear and actionable, guiding users on how to correct their input.

# Tables and Fields

## Table

- Users Table
  - UserID (Primary Key)
  - Username
  - Email
  - Role (Citizen, CityOfficial, Administrator)
- Issues Table
  - IssueID (Primary Key)
  - ReporterID (Foreign Key referencing Users)
  - AssignedOfficialID (Foreign Key referencing Users)
  - Category
  - Status
  - Title
  - Description
  - Latitude
  - Longitude
  - ReportedDate
  - LastUpdatedDate
- Comments Table
  - CommentID (Primary Key)
  - IssueID (Foreign Key referencing Issues)
  - UserID (Foreign Key referencing Users)
  - Content
  - Timestamp
  - IsInternalNote (Boolean)

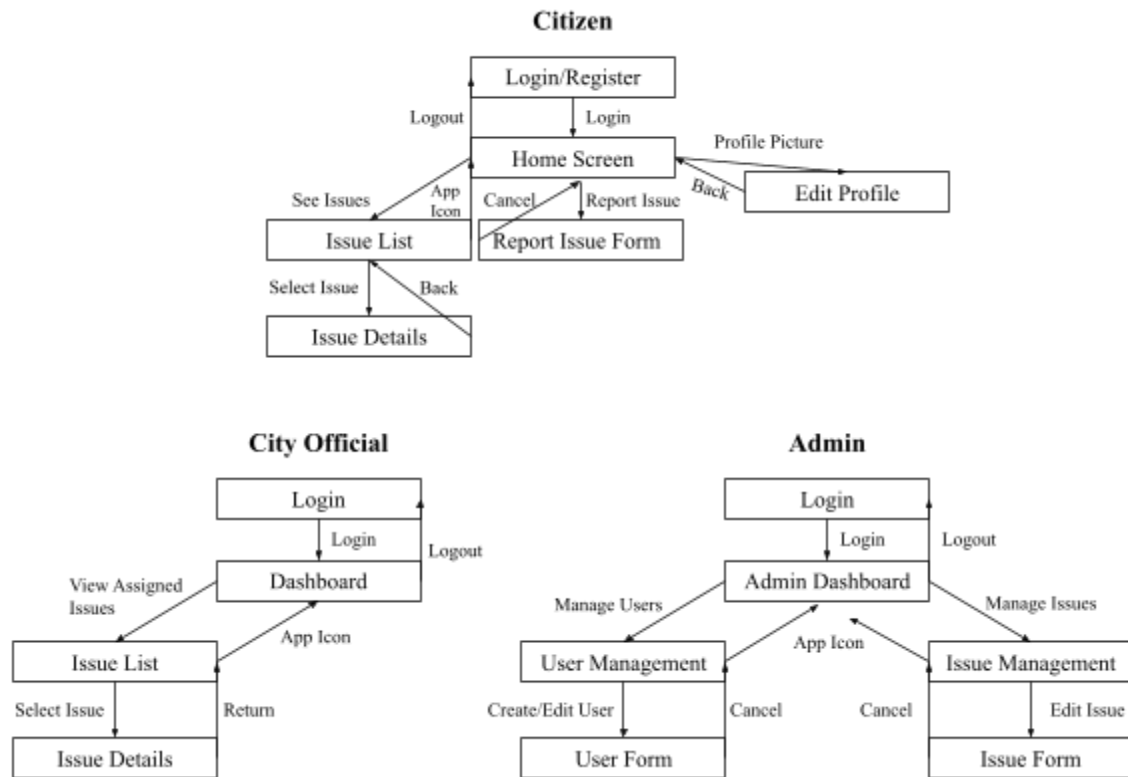
## Relationships

- Users to Issues: One-to-Many (One user can report many issues)
- Users to Comments: One-to-Many (One user can make many comments)
- Issues to Comments: One-to-Many (One issue can have many comments)

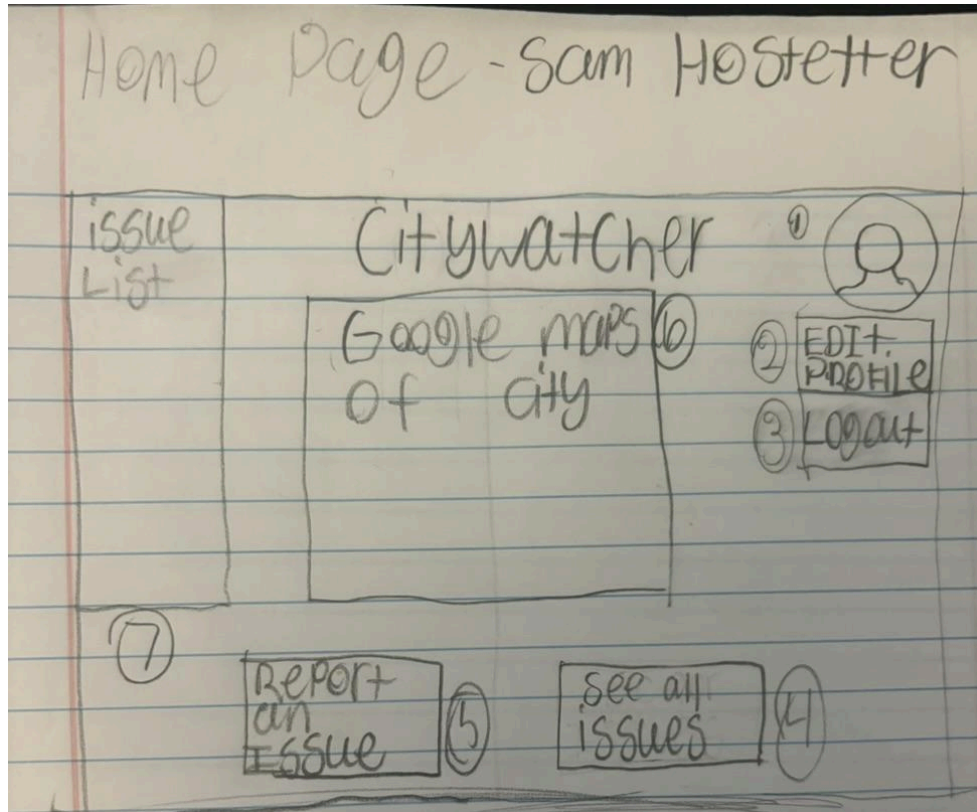
## File Storage

- Issue Images:
  - Stored on the file system with a naming convention like "IssueID\_Timestamp.jpg"
  - Path to the image file stored in the Issues table
  - Format: JPEG for compression and wide support

# Screen Flow Diagram



# Screen Sketches

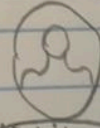


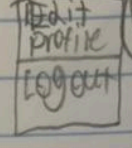
This screen is the first screen you will see after logging in or signing up for the app.

There are several different things you can do from this page. The focus is on the city, where you will be able to see issues on the maps. This is structured to focus on the map of the city but allow for users to see other items that they need to use the app.

This allows for a dropdown menu to appear to edit profile or logout (1), Opens a new menu to edit your profile (2), sends you back to the login screen, and logs this user out (3), opens the issue list so the user can see all of the issues (4), Takes the user to the Report Issue Form, where they can report their own issue (5), Shows the google maps view of the city, also shows users location (6), preview of the issues in the city, only a few issues here.

**Report Issue Form - Sam Hostetter**

**Back to Home Screen** (6) **Citywatcher** 

**Report Form**  (7)

**Category** (1)

**Title** (2)

**Description** (3)

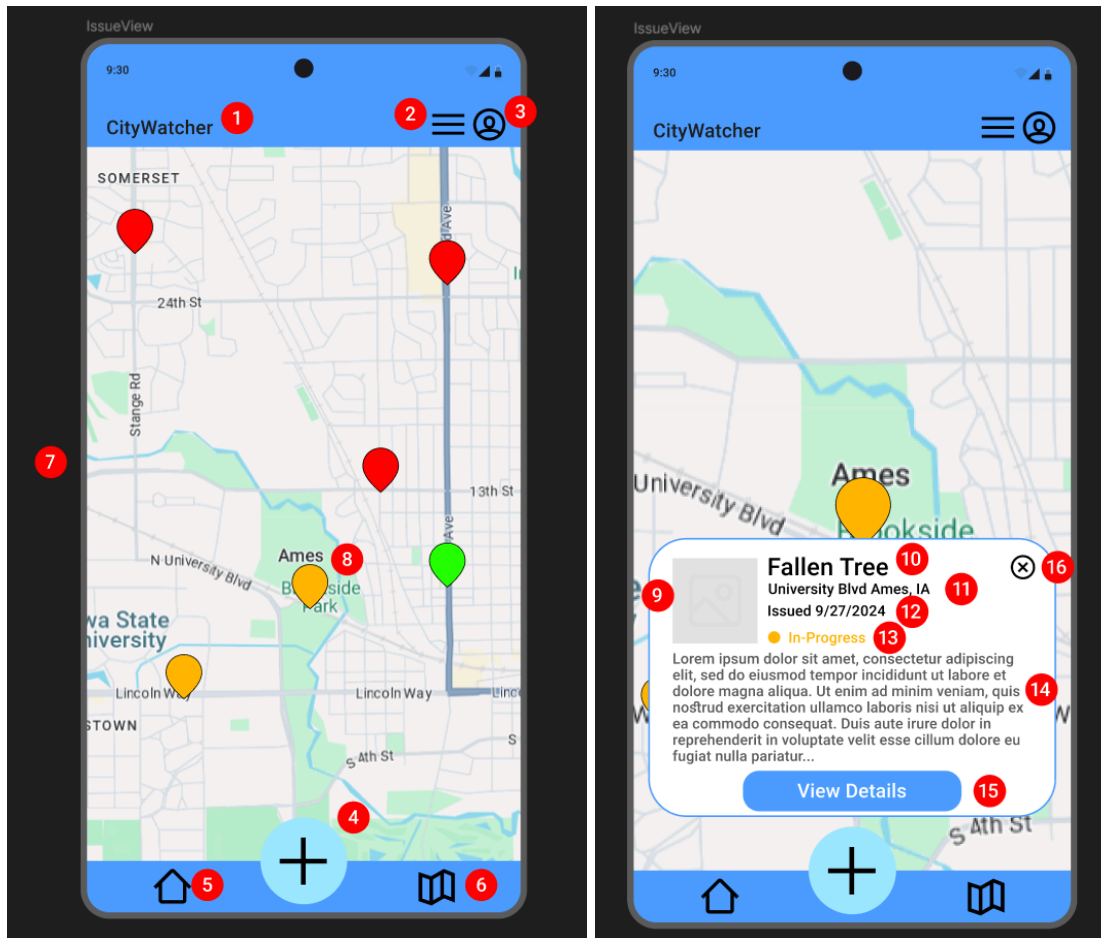
**Location** (4)

**Image (JPEG only)** (5)

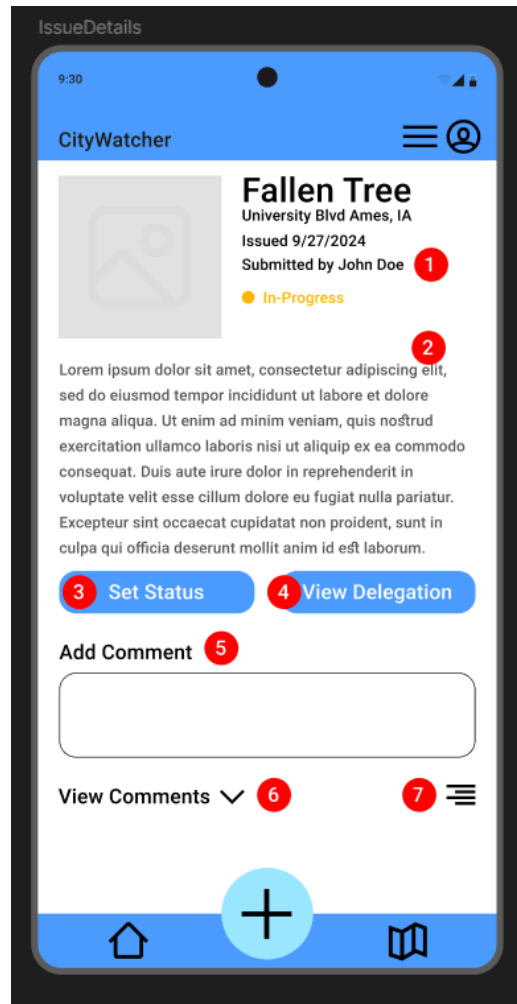
**upload image** (5)

This screen allows the user to input an issue to the issue list. Several different fields need to be filled for there to be an issue put into the database. This gives enough information for the city officials to be able to make a plan to fix said issue.

A dropdown menu of different categories to put the issue in (1), a textbox to input a title for the issue(2), a short description of the issue that the user can input (3), a location to assign the issue to (4), an image of the issue if possible (5), a way to get back to the home screen (6), a dropdown menu to logout or edit the users profile (7)



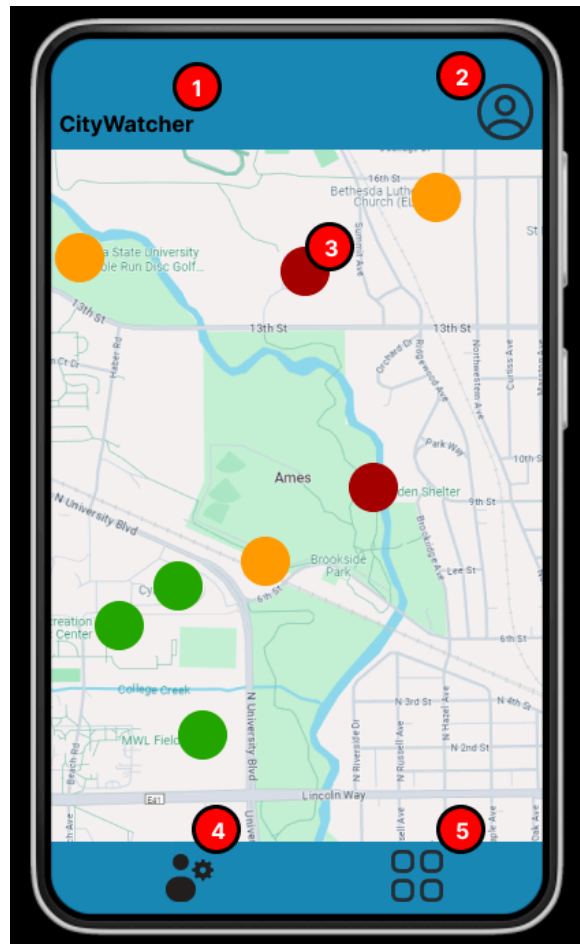
Issue View Screen: This is the screen which displays all of the current issues. In the navigation bar, the user can return to the home screen by tapping the logo/title (1). The menu (2) allows users to create an issue, view issues, etc. If the user is signed into an administrator account, their administrative privileges are accessed here. The user can also view their profile or log out (3). The bottom bar allows additional ways for users to add an issue (4), return to the home screen (5), or view all issues (6). The content of the screen includes an interactable map (7) and the locations of current issues are marked with a pin (8). The color of each pin signifies the status of the issue. Tapping on a pin pops up information about the corresponding issue. This includes the image provided (9), the issue name (10), location (11), date (12), status (13), and a shortened description (14). The user is able to view additional details about the issue (15) or close this pop up (16).



Issue Details Screen: This screen provides an expanded view of an issue and is accessed from the Issue View screen. In addition to the data provided on that screen, this screen displays the user who submitted the issue (1) and the full description of the issue (2). It also allows the user to add a comment (5), view comments (6), and sort comments by certain criteria (like count, date, etc.). If signed into an administrator account, admins can change the status of the issue (3) or view and assign delegation to an issue (4).



## Admin Dashboard (Nicholas Morrow)



The Admin Dashboard is the central hub for the administrator, providing quick access to key functions such as managing users, viewing issues, and navigating back to the main dashboard. From here, the admin can also logout or view individual issues marked on the city map. (1) Acts as a home button, making it easy for the admin to return to the main dashboard from anywhere in the app. (2) Provides access to account management actions, like logging out, ensuring the admin can easily exit the app. (3) Gives a live, visual representation of where issues are located in the city, helping the admin track problems geographically. (4) Quick access to user management, allowing the admin to control who has access to the app and their respective roles. (5) Directs the admin to the core functionality of managing and resolving issues reported by citizens.

## User Management (Nicholas Morrow)



This screen is designed for administrators to manage user accounts within the app. It provides functionalities to search for existing users, filter them based on criteria like role, access detailed user information, and potentially create or modify user accounts. (1) Clicking the logo navigates the administrator back to the main admin dashboard. (2) This section displays the currently logged-in administrator's profile and includes a logout option when clicked. (3) This search bar allows administrators to search for specific users by their username. (4) This area provides filtering options to refine the user search. Administrators can filter users based on their role (e.g., Citizen, City Official, Administrator). (5) When a user is found via the search, clicking on their entry in the search results will open a detailed user form. This form displays comprehensive information about the user and allows the administrator to edit user details and manage their account. (6) This button opens a blank user form, allowing administrators to create new user accounts. (7) Quick access to user management, allowing the admin to control who has access to the app and their respective roles (same as the admin dashboard). (8) Directs the admin to the core functionality of managing and resolving issues reported by citizens (same as the admin dashboard).