

UGAthens Transit: Software Design

A Website that Integrates UGA and Athens Transit Data

By

Eleanor Madderra, Keerthana Ramesh, Isra Naweed, Anjali Devarapalli,
Ryan Majd, and Shreyan Bose

December 8, 2024

Table of Contents

INTRODUCTION.....	3
Project Overview.....	3
Relevant Parties.....	3
DESIGN GOALS.....	3
SYSTEM BEHAVIOR.....	4
LOGICAL VIEW.....	5
High-Level Design.....	5
High-Level Components.....	5
1. Authentication Module.....	5
2. Bus Stop Information Module.....	6
3. Map Interaction Module.....	6
4. UI/UX Module.....	6
6. Dynamic Patterns of Interaction.....	7
6.1. User Authentication:.....	7
Mid-Level Design.....	7
UML Diagram.....	8
PROCESS VIEW.....	9
DEVELOPMENT VIEW.....	10
PHYSICAL VIEW.....	11
USE CASE VIEW.....	12

INTRODUCTION

Project Overview

This document documents the process and design of the UGAthens Transit website. UGAthens Transit is a website that combines live tracking of both UGA and Athens Transit buses. Our goal is to provide Athens residents with a more accessible way to navigate UGA's campus and the city, ultimately enhancing their public transportation experience. By making public transportation easy to navigate, we hope to encourage those who live in Athens to try taking the bus for the first time or use Athens public resources more often.

Relevant Parties

The purpose of this document is to describe the architect and design of the UGAthens Transit website for all potential users and any other interested parties. The major parties include:

Users	Users may want to understand how UGAthens Transit works and want to know where we get our data from.
Project Manager	The Project Manager is responsible for the schedule and assigning tasks.
Developers	Developers are the ones who build and develop the UGAthens website features.
Maintenance Programmers	The potential programmers in the future who will edit and build on the base UGAthens website that exists currently.

DESIGN GOALS

This is how our team defines good design:

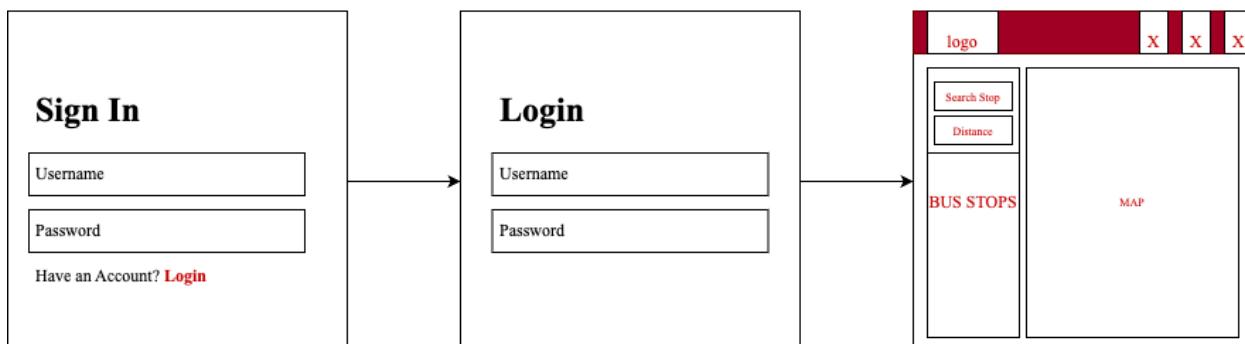
Can Easily Be Built Upon	We have an iterative design process so ensuring our project can be built upon in the future is important.
Good Documentation	We want future programmers to easily pick up where past programmers have left off. We will create detailed design documents documenting our process and write descriptive

	comments in our code when necessary. We use the JavaScript Doc standard which applies to our components and api calls.
As Simple As Possible	Our programming team will typically only have 2 to 4 people actively working at a time. We will focus on doing things as simply as possible to avoid later complications that may be time-consuming.
Reusable	Since our programming team is so small and we are working on a short schedule, we want to reuse as much code as possible.

SYSTEM BEHAVIOR

Our system is designed with a focus on user-friendliness and reliable component initialization. To prevent API overloading, the system does not offer a guest view. Instead, users are required to register using an email address and a strong password. Once authenticated via Firebase and successfully logged in, users gain access to the map interface, where they can search for and view bus stops for both UGA buses and Athens Transit buses

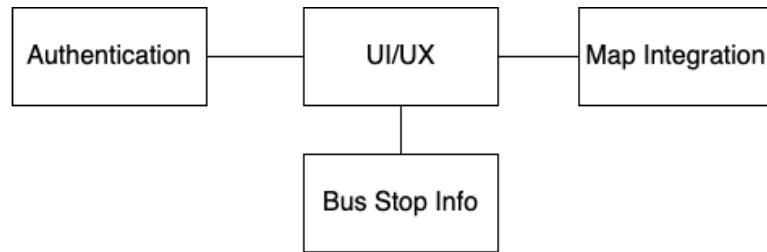
Version 2.1, No Guest View



System Behavior Diagram

LOGICAL VIEW

High-Level Design



Logical View Diagram

High-Level Components

The UGAthens Bus Stop website consists of the following main components:

1. Authentication Module

- **Purpose:** Ensures secure access to the app's functionalities.
- **Components:**
 - SignInComponent: Provides the user interface for logging in.
 - SignUpComponent: Allows new users to create an account.
- **Dynamic Interaction:**
 - The SignUpComponent communicates with Firebase Authentication to create new accounts.
 - The SignInComponent verifies credentials with Firebase Authentication before granting access.
- **Static Relationships:**
 - Both components depend on the Firebase configuration for authentication services.

2. Bus Stop Information Module

- **Purpose:** Displays and filters bus stops based on user input.
- **Components:**
 - SearchBarComponent: Allows users to search for specific bus stops.
 - DistanceFilterComponent: Filters bus stops based on proximity.
 - BusStopCardComponent: Displays individual bus stop details (e.g., name, location, time).
- **Dynamic Interaction:**
 - User input in the SearchBarComponent triggers a query to the backend API for relevant bus stops.
 - The DistanceFilterComponent refines the results based on user preferences.

3. Map Interaction Module

- **Purpose:** Visualizes bus stops on a map with real-time interactivity.
- **Components:**
 - MapDisplayComponent: Renders the map with bus stop markers.
 - MarkerComponent: Represents individual bus stops on the map.
- **Dynamic Interaction:**
 - The MapDisplayComponent retrieves geolocation data from the backend API and updates markers dynamically.
 - Interaction with markers triggers pop-ups with detailed bus stop information.

4. UI/UX Module

- **Purpose:** Manages the app's visual and functional layout.
- **Components:**
 - NavigationBar: Includes links for navigation (e.g., "Contact Us", "About Us").
 - ThemeToggle: Allows users to switch between light and dark modes.
- **Dynamic Interaction:**
 - Navigation buttons link to different views (e.g., Home, Map, Contact).
 - State management for theme preferences is persistent across user sessions.

5. Static Relationships

- The Authentication Module depends on the FirebaseService for secure authentication.

- The Bus Stop Information Module and Map Interaction Module interact with the same backend API to ensure synchronized data.
- The UI/UX Module interacts with all other modules to maintain a cohesive and consistent user experience.

6. Dynamic Patterns of Interaction

6.1. User Authentication:

- The user inputs credentials into the SignInComponent.
- The FirebaseService validates the credentials and, on success, redirects to the main app view.

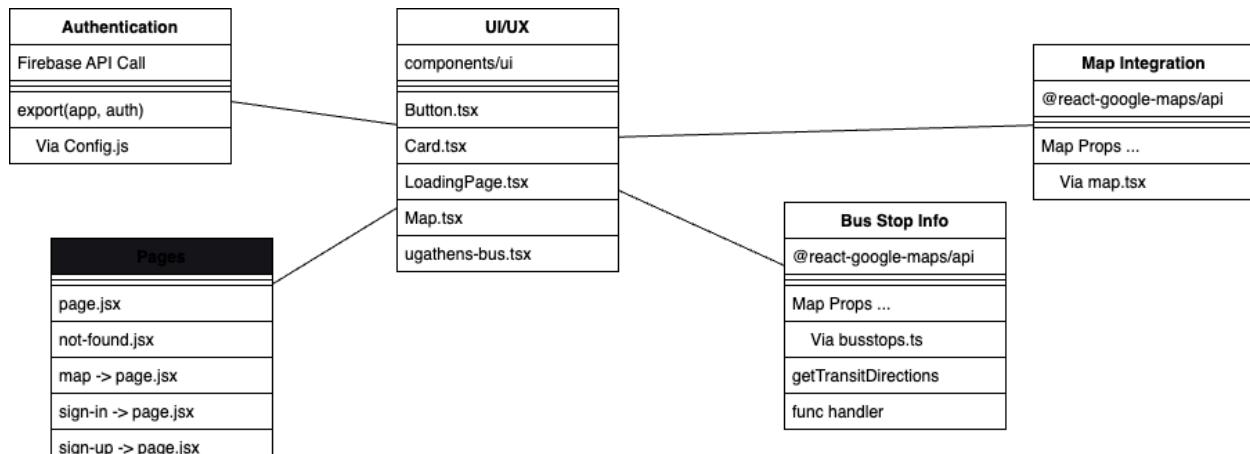
6.2. Bus Stop Search and Filter:

- The user enters a query in the SearchBarComponent and selects a distance filter.
- The BusStopService fetches relevant data from the backend API, which updates the BusStopCardComponent and MapDisplayComponent.

7. Map Interaction:

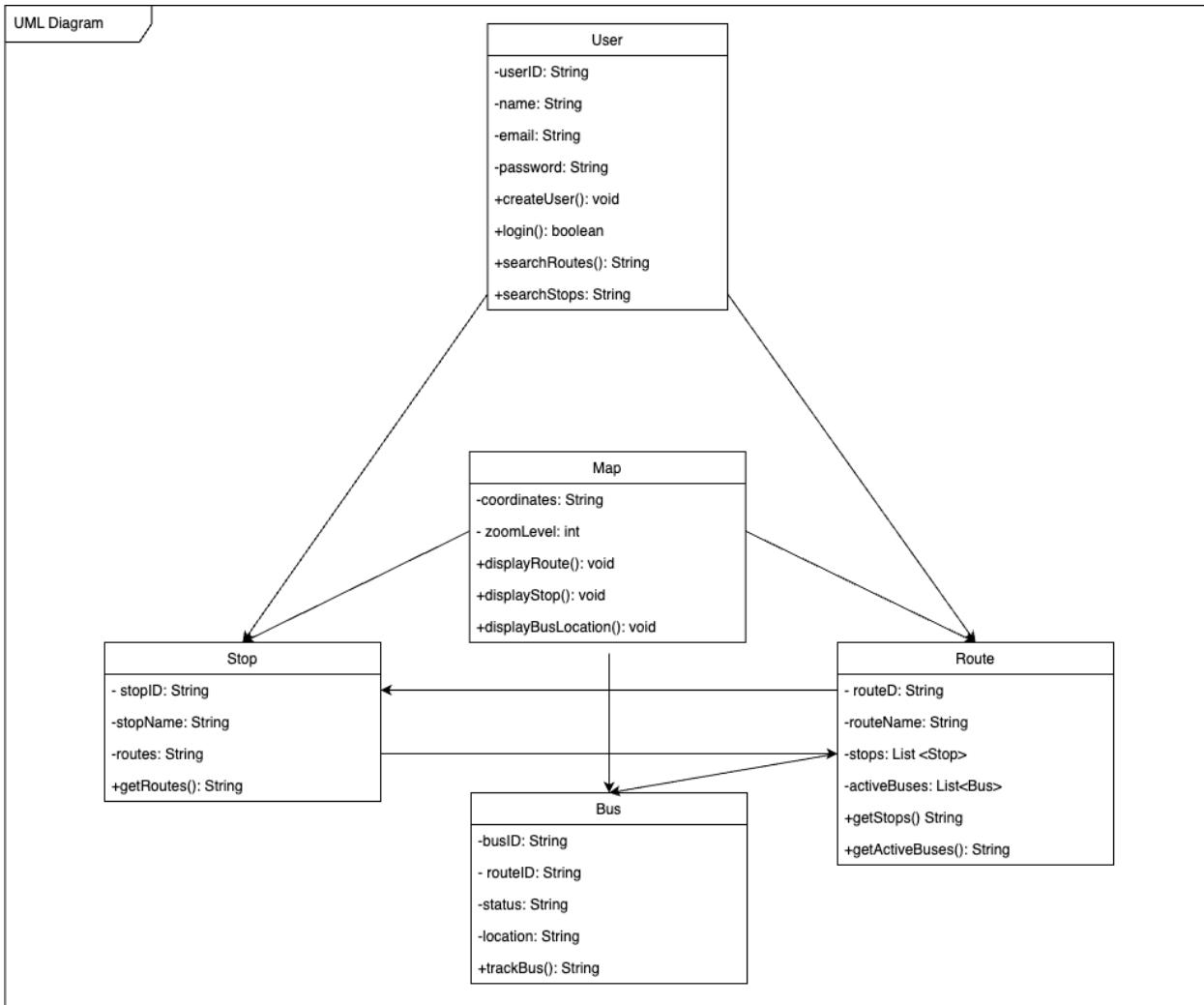
- The MapDisplayComponent initializes with markers fetched from the backend.

Mid-Level Design

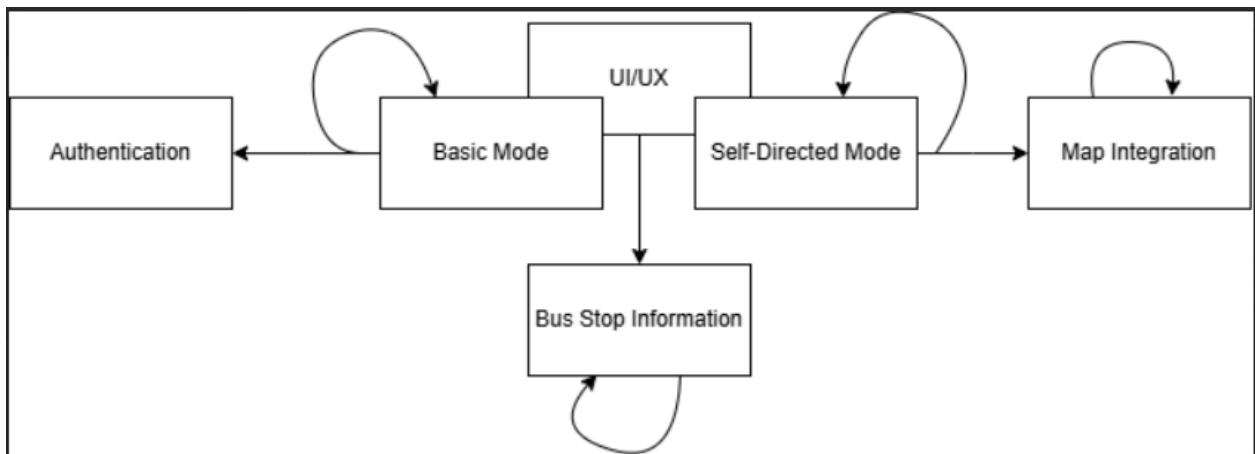


Mid Level View Design Diagram

UML Diagram



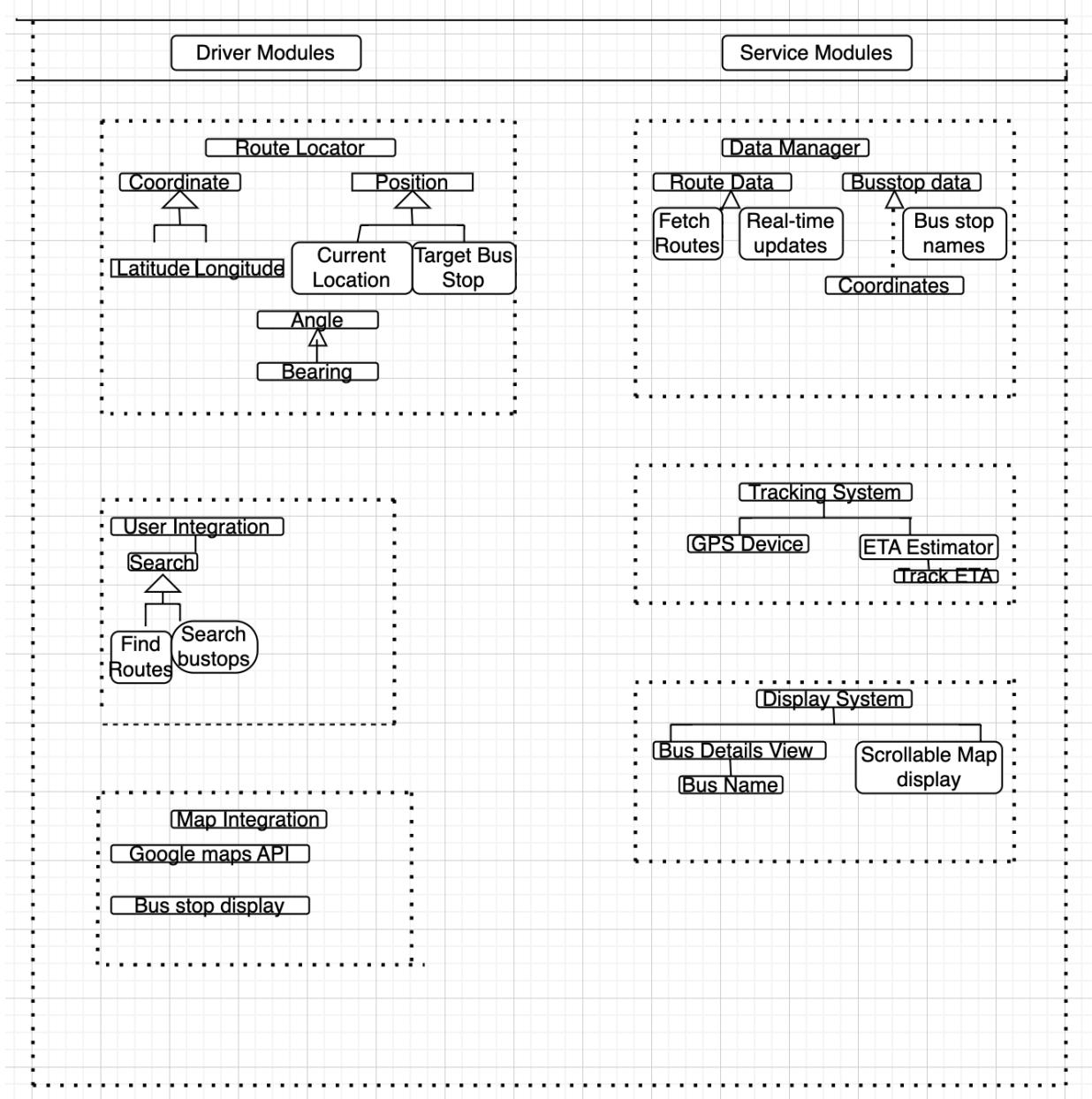
PROCESS VIEW



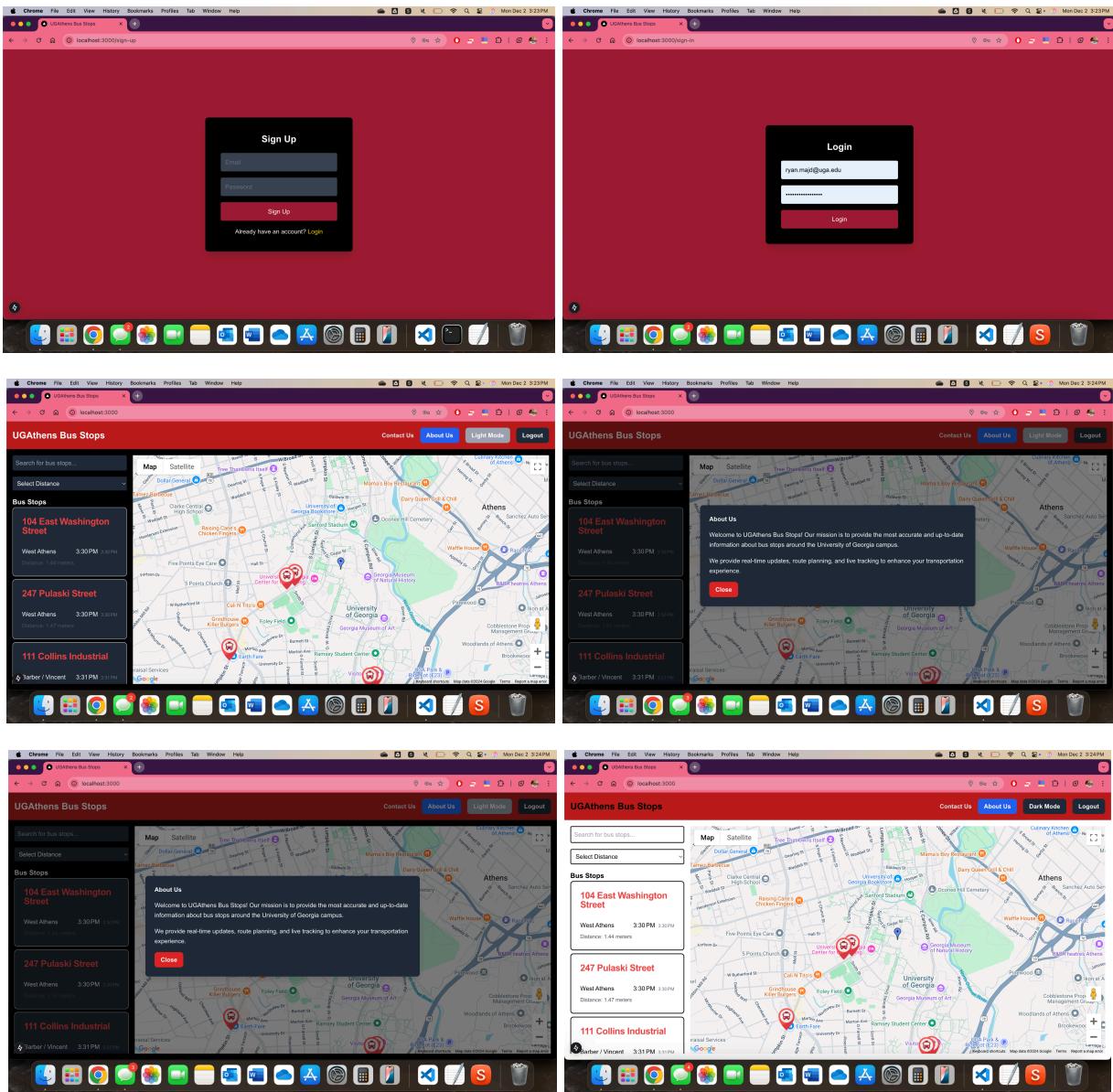
*Self-directed mode displays an advanced system with real-time data updates and proactive behavior.

*Looped arrows indicate processes that are continuously running

DEVELOPMENT VIEW



PHYSICAL VIEW



USE CASE VIEW

Use Case:	Sorting Through Bus Stops
Actors:	Users
Description:	This use case begins when an individual visits the website. The user will be able to sort through different bus stops of both UGA and Athens Transit systems. If a bus stop is selected, the website will display all routes that stop at that particular bus stop.

Use Case:	Sorting Through Bus Routes
Actors:	Users
Description:	This use case begins when an individual visits the website. The user will be able to sort through different bus routes of both UGA and Athens Transit systems. If a route is selected, the website will show all stops on that route.

