

Design Document

1. Overview	1
1.1 Scope	1
1.2 Purpose	1
1.3 Intended Audience	1
1.4 Conformance	2
2. Definitions	2
3. Conceptual Model for Software Design Descriptions	2
3.1 Software Design in Context	2
3.2 Software Design Descriptions within the Life Cycle	3
4. Design Description Information Content	4
4.1 Introduction	4
4.2 SDD Identification	4
4.3 Design Stakeholders and Their Concerns	4
4.4 Design Views	5
4.4.1 User Interaction View	5
4.4.2 Backend Data Processing View	5
4.4.3 Database View	6
4.5 Design Viewpoints	6
4.5.1 Frontend Viewpoint	6
4.5.2 Backend Viewpoint	6
4.5.3 Database Viewpoint	6
4.6 Design Elements	7
4.6.1 WireFrame Layout	7
4.7 Design Overlays	8
4.8 Design Rationale	8
4.9 Design Languages	8
5. Design Diagrams	9
5.1 UML Diagram	9
5.2 Sequence Diagram	10
5.3 Entity-Relationship Diagram (ERD)	11
6. Human Interface Design	12
6.1 Overview of User Interface	12
6.2 Screen Images	12
7. Requirements Matrix	15

1. Overview

The RealEase project is a web-based real estate platform that allows users to easily access property information, neighborhood insights, and financial analysis tools, similar to Zillow. The platform will feature core functionalities such as a Neighborhood Insights Dashboard, ROI Calculator, and Home Comparison Tool. Our goal is to deliver a user-friendly platform that empowers home buyers and investors with the information needed to make well-informed decisions.

1.1 Scope

The design of RealEase is intended to provide a seamless user experience, leveraging real-time data from multiple sources to offer users insights into properties and neighborhoods. This document focuses on the design architecture for Milestone 1, which includes the core website structure, the layout of the homepage, and the features mentioned above. The scope of this document includes a detailed design of the platform's user interface, backend architecture, and data flow.

1.2 Purpose

The purpose of this document is to describe the design of the RealEase platform for Milestone 1. This includes defining the user interface elements, the system's backend architecture, and how the data will be managed. The document also outlines the core modules and functionalities of the website, explaining how each feature works within the platform. By documenting this design, the project team ensures clarity and direction for the development process.

1.3 Intended Audience

This document is intended for the development team, project stakeholders, and faculty advisors who will be involved in building and reviewing the RealEase platform. It serves as a guide for understanding the structure of the platform, ensuring that all members of the team are aligned with the project goals and design.

1.4 Conformance

The design outlined in this document conforms to the industry-standard practices for web development and user experience (UX) design. It also ensures that the platform adheres to security and data management protocols required for handling sensitive information like property data, financial details, and user activity.

2. Definitions

For the purpose of this document, the following terms are defined:

- **User:** The individual who interacts with the platform to search for properties, access insights, or calculate financial metrics.
 - **API:** Application Programming Interface, used to retrieve data from external real estate, financial, or location-based services.
 - **MERN Stack:** A combination of MongoDB, Express.js, React, and Node.js that will be used to build and power the platform.
-

3. Conceptual Model for Software Design Descriptions

3.1 Software Design in Context

The RealEase platform's architecture is built with scalability and user experience in mind. The **MERN Stack** provides a flexible framework, enabling rapid development and real-time data retrieval. RealEase's design is centered on offering three core features:

1. **Homepage and Search Bar:**

- Users can input a ZIP code or address to find property details and local insights.

- The homepage will showcase a clean layout, with navigation links to other key features like the **Neighborhood Insights Dashboard**, **Home Comparison Tool**, and **ROI Calculator**.

2. **Neighborhood Insights Dashboard:**

- Once a user inputs a location, the system pulls data from external APIs to provide information on schools, crime rates, demographics, and amenities within that neighborhood.
- This information will be displayed in a combination of map views, charts, and lists.

3. **Home Comparison Tool:**

- Users can select multiple properties to compare side-by-side.
- The comparison includes detailed information like price, square footage, number of bedrooms, and unique property features.
- Users will be able to see the data in a comparison table with easy-to-read metrics.

4. **ROI Calculator:**

- Allows users to enter mortgage details, property prices, rental income, and expenses.
- The tool will calculate ROI metrics such as cash-on-cash return, net profit, and internal rate of return (IRR).

3.2 Software Design Descriptions within the Life Cycle

In **Milestone 1**, we are focusing on laying out the basic functionalities of the **Search Bar**, **Neighborhood Insights Dashboard**, and **Home Comparison Tool**. This milestone will provide a foundation for understanding the direction we want to go and aid feature development, integration, and performance optimization in future phases.

4. Design Description Information Content

4.1 Introduction

The design of RealEase is broken down into three main layers: frontend, backend, and database. The frontend will handle user interactions and data visualization, while the backend manages data retrieval and processing. The database stores user-specific information, such as search history and favorite properties. The integration of external APIs is key to providing real-time property data, neighborhood insights, and financial metrics.

4.2 SDD Identification

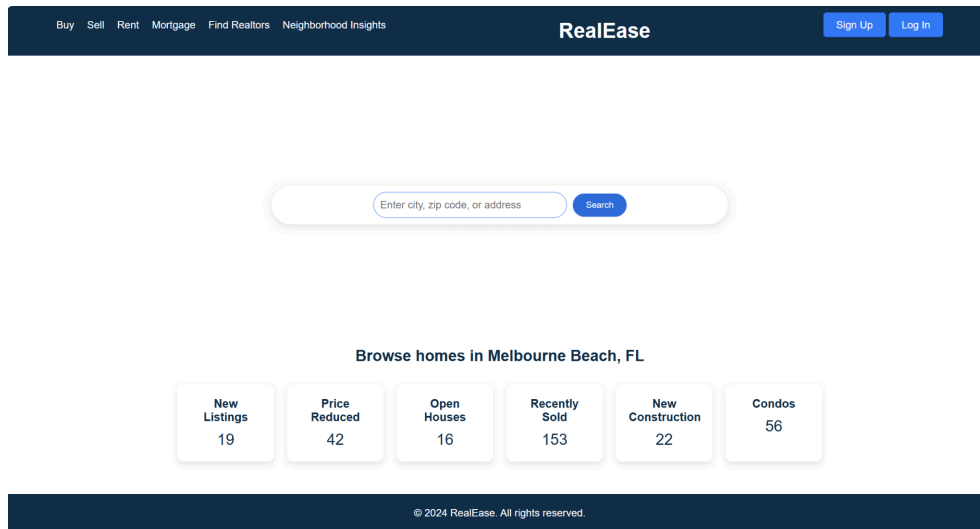
This Software Design Description (SDD) identifies the core components and interactions within the system. It outlines the architecture, modules, and user flows that make up the RealEase platform for Milestone 1. Each component is designed with scalability in mind, allowing for easy integration of future features.

4.3 Design Stakeholders and Their Concerns

The primary stakeholders include:

- End-users: Looking for a smooth experience in finding neighborhood information, comparing homes, and calculating ROI.
- Developers: Focusing on ease of implementation, modularity, and ensuring a reliable data flow between frontend and backend.
- Project Stakeholders: Concerned with the platform's functionality, user engagement, and the potential for future growth.
- Faculty Advisors: Ensuring the platform meets academic and technical standards while solving a real-world problem.

4.4 Design Views



4.4.1 User Interaction View

Users interact with RealEase through a clean, easy-to-use interface. On the homepage, the search bar is prominently placed, allowing users to quickly input a ZIP code or address. Once they search, results are displayed with an option to navigate to the Neighborhood Insights Dashboard, Home Comparison Tool, or ROI Calculator. Navigation links to the "Buy", "Sell", "Rent", and "Finance" sections are also available at the top of the page.

4.4.2 Backend Data Processing View

The backend retrieves data from various APIs, processes it, and sends it to the frontend. Data is requested based on user inputs in the Search Bar, and results are displayed in real-time. The Neighborhood Insights Dashboard pulls from external APIs to provide data on schools, crime, and amenities, while the Home Comparison Tool uses MongoDB to retrieve saved properties for comparison.

4.4.3 Database View

The database stores user-specific data such as favorite properties, search history, and home comparison selections. MongoDB is chosen for its scalability and flexibility, allowing us to store complex property data and user preferences.

4.5 Design Viewpoints

4.5.1 Frontend Viewpoint

The frontend is developed using React, which communicates with the backend through RESTful APIs to retrieve and display property data efficiently. We aim to make a simple front end where users can enter an address or zip code or select one of your five features from the taskbar at the top of the screen.

4.5.2 Backend Viewpoint

The backend, powered by Node.js and Express, handles requests from the frontend and communicates with external APIs to fetch property data, neighborhood insights, and financial metrics. It processes user inputs and returns relevant information for display on the frontend.

4.5.3 Database Viewpoint

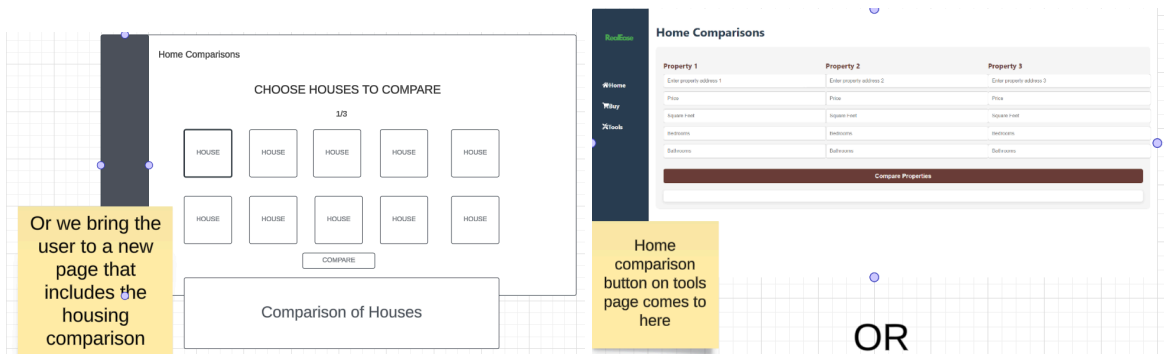
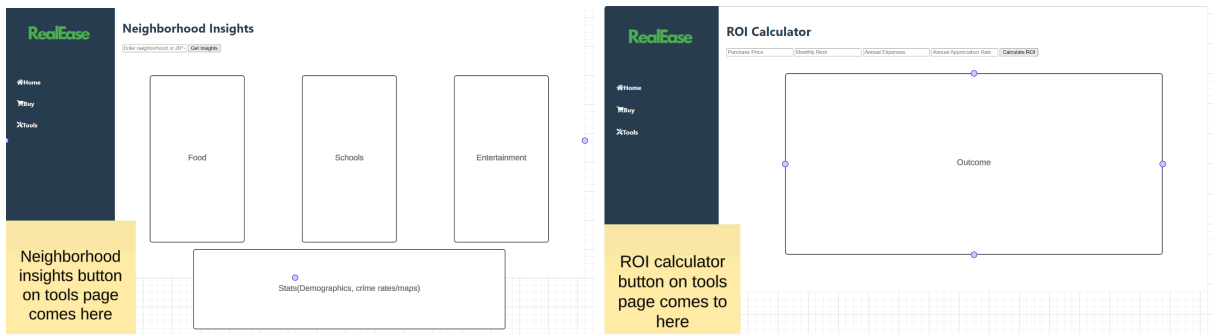
MongoDB is the primary database used to store user data, property details, and comparison history. Its document-based storage model allows for efficient querying and fast data retrieval, ensuring that users can quickly access their saved data and comparisons.

4.6 Design Elements

The platform's design elements are broken down into three major components:

1. Search Bar: Allows users to enter a ZIP code or address and submit a request to retrieve data.
2. Dashboard and Tools: Includes the Neighborhood Insights Dashboard, Home Comparison Tool, and ROI Calculator, each visualizing complex data in a simple format.
3. API Integration: Handles external data sources and manages API requests to ensure the most accurate and up-to-date information is displayed.

4.6.1 WireFrame Layout



Color Scheme:	6:3:1 White, Dark Blue/Gray, Brown	Logo work in progress, will not be green
---------------	---------------------------------------	--

4.7 Design Overlays

The design overlays focus on how different modules interact. When a user inputs a search query, the system processes the data in real-time, fetching results from the API and rendering them on the page. These overlays ensure the design remains modular and scalable. Our early expectation for this is to have our small database that stores some queries so that we don't have to make an API call every time a user loads a web page, as we'll have some data caught.

4.8 Design Rationale

Using MERN stack allows for a fast, scalable platform that can handle real-time data retrieval and visualization. The modular nature of React makes it easy to add features, while Node.js and Express ensure the backend remains efficient and flexible.

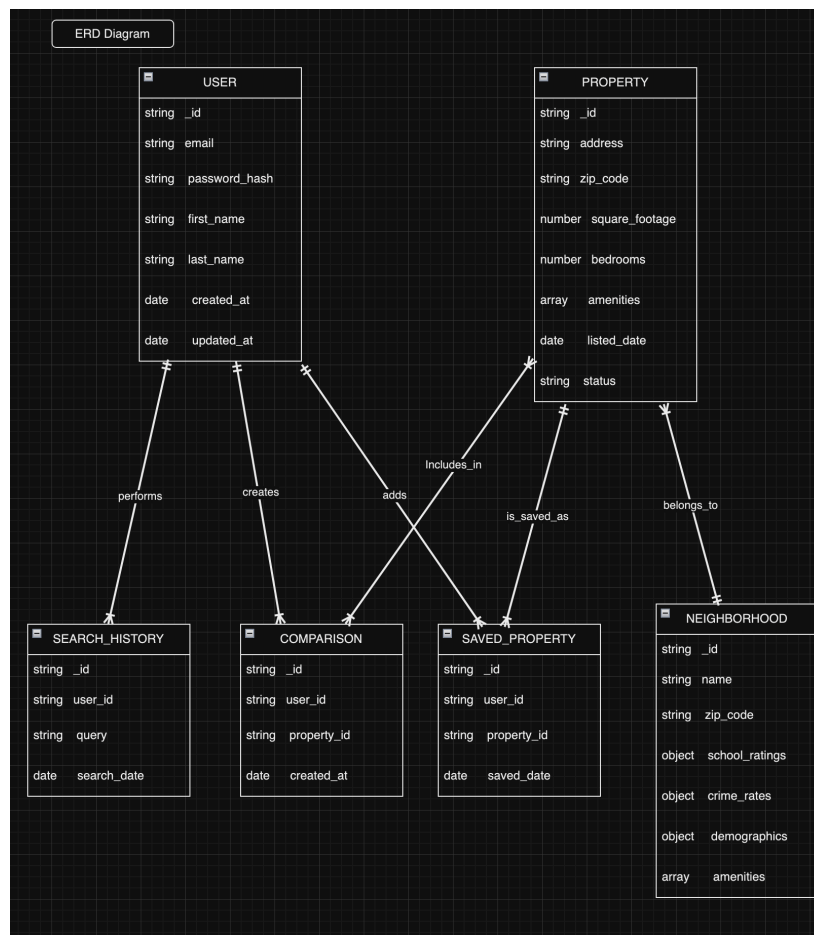
4.9 Design Languages

The frontend uses React and JavaScript for interactivity, while the backend leverages Node.js and Express for data processing. MongoDB is the preferred choice for the database due to its scalability and ability to handle complex queries.

5. Design Diagrams

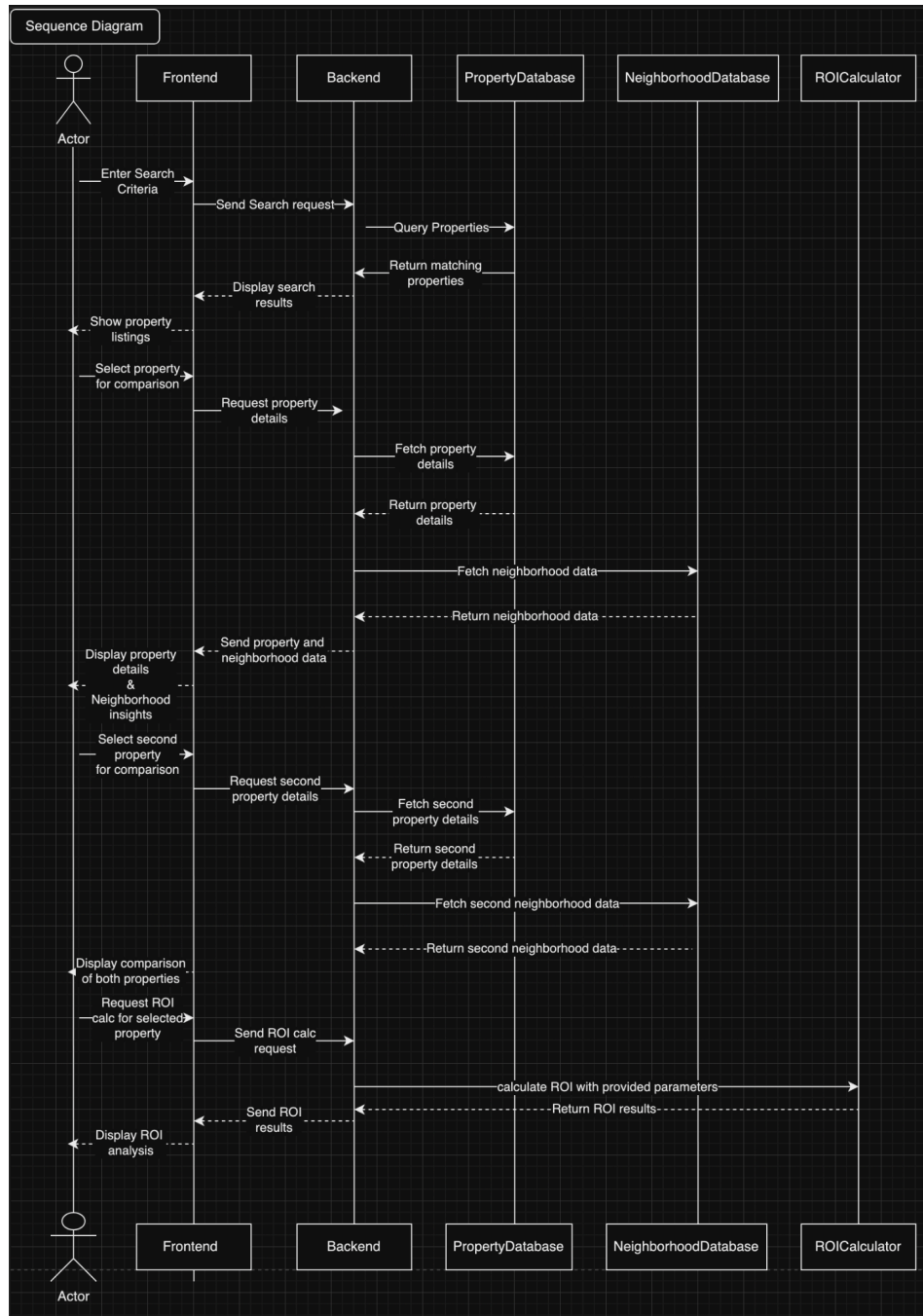
5.1 UML Diagram

The UML diagram represents the interactions between the different components of the RealEase platform. It showcases how the Search Bar, Neighborhood Insights Dashboard, Home Comparison Tool, and ROI Calculator are interconnected. The frontend communicates with the backend, which then retrieves and processes data from MongoDB and external APIs.



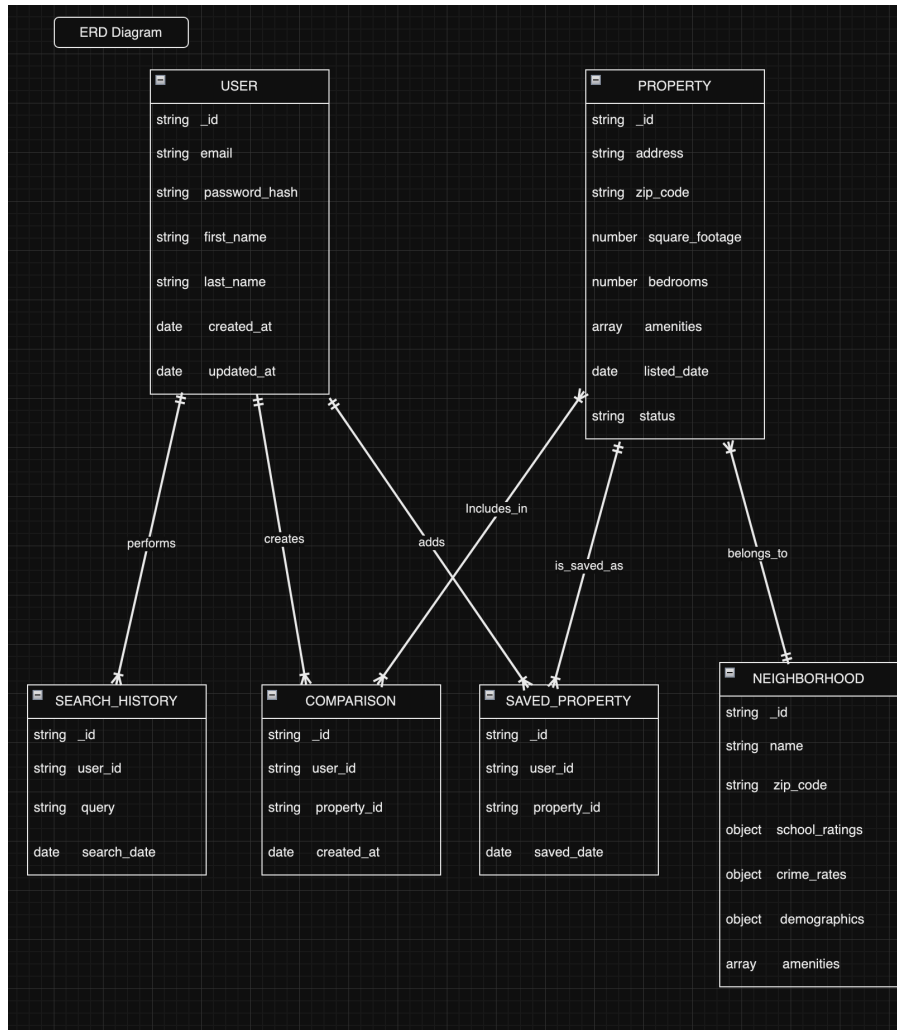
5.2 Sequence Diagram

The sequence diagram illustrates how the user interacts with the platform. For example, in the Search Feature, the user inputs a ZIP code, and the backend processes the request by retrieving data from MongoDB or external APIs, returning it to the frontend for display.



5.3 Entity-Relationship Diagram (ERD)

The ERD depicts how data is structured within the MongoDB database. The entities include User, Property, and Neighborhood, and show the relationships between these entities, such as which properties a user has saved or compared.

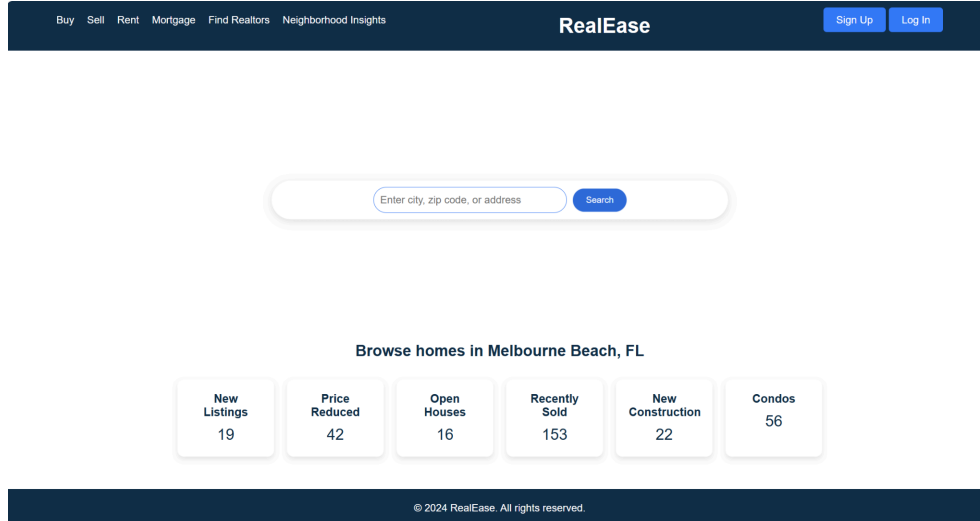


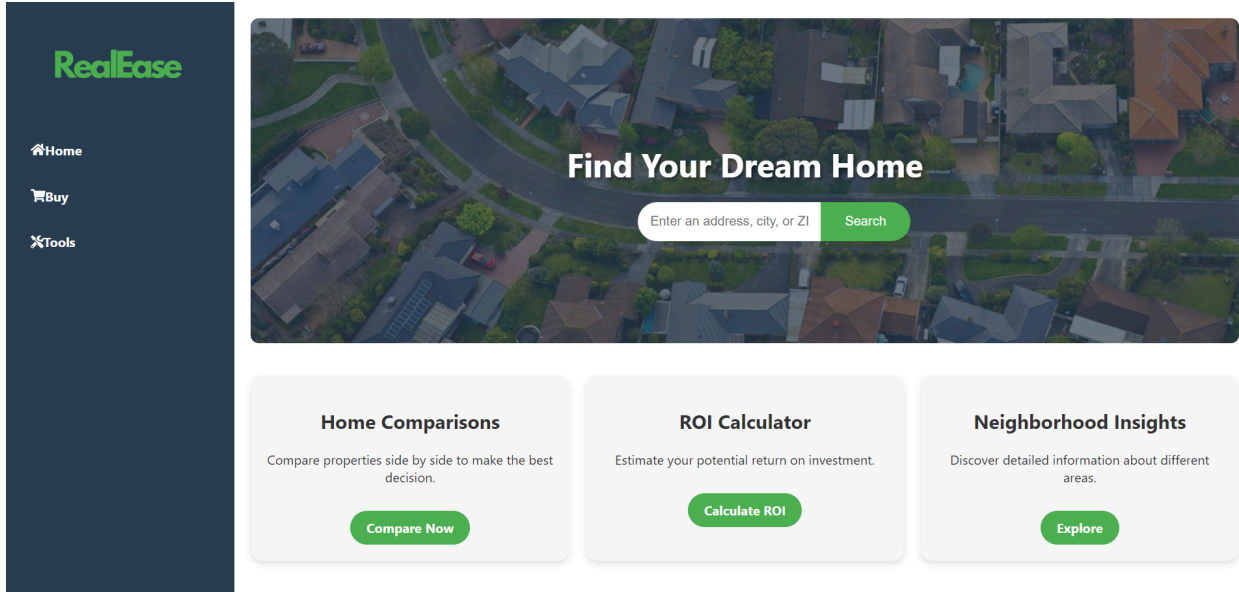
6. Human Interface Design

6.1 Overview of User Interface

When users open the RealEase web application, they can enter an address or ZIP code to search for a specific house or area. Additionally, users can access any of the three main features—Neighborhood Insights Dashboard, Home Comparison Tool, and ROI Calculator—from the top toolbar. Users will be directed to a dedicated page for that specific tool by selecting one of these features. On these pages, users can input relevant information: for the Neighborhood Insights Dashboard, they can enter a neighborhood or ZIP code; for the Home Comparison Tool, they can select a house to compare; and for the ROI Calculator, they can input details for a specific house along with the necessary financial information to calculate return on investment (ROI). Each feature is designed to be intuitive, guiding users through the process with easy navigation and data input fields.

6.2 Screen Images





This mockup features a dark blue sidebar on the left with the RealEase logo in green. The main content area has a background image of a residential neighborhood. The search bar and call-to-action buttons are green.

RealEase

- Home
- Buy
- Tools

Find Your Dream Home

Enter an address, city, or ZI

Home Comparisons

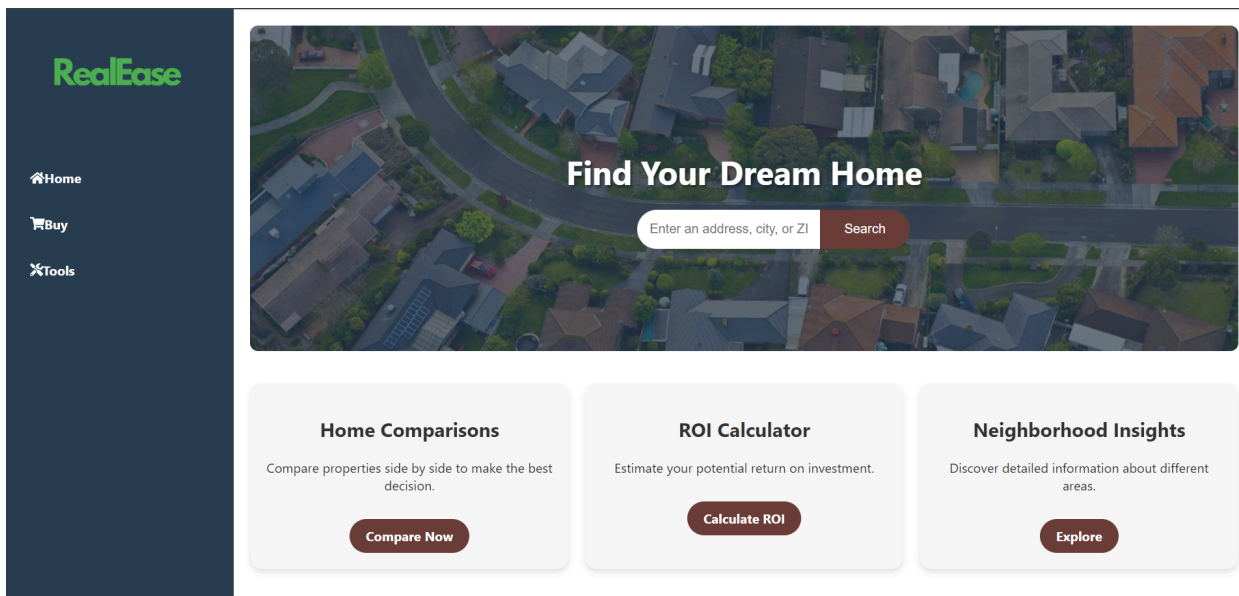
Compare properties side by side to make the best decision.

ROI Calculator

Estimate your potential return on investment.

Neighborhood Insights

Discover detailed information about different areas.



This mockup features a dark blue sidebar on the left with the RealEase logo in green. The main content area has a background image of a residential neighborhood. The search bar and call-to-action buttons are brown.

RealEase

- Home
- Buy
- Tools

Find Your Dream Home

Enter an address, city, or ZI

Home Comparisons

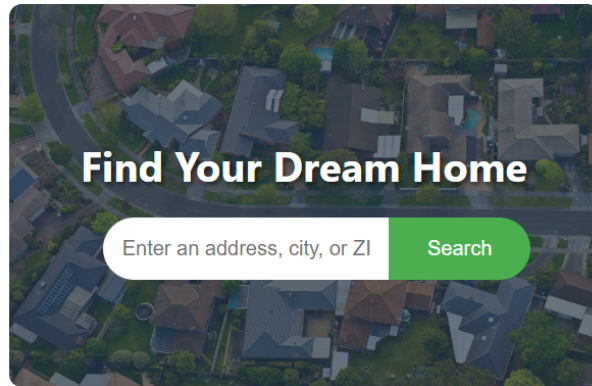
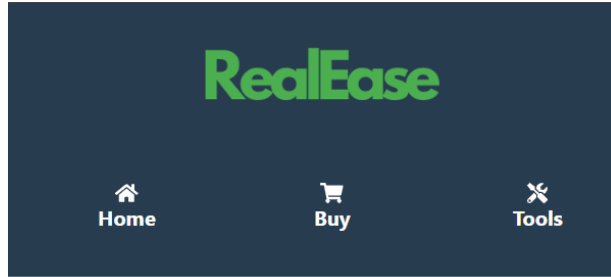
Compare properties side by side to make the best decision.

ROI Calculator

Estimate your potential return on investment.

Neighborhood Insights

Discover detailed information about different areas.



7. Requirements Matrix

Milestone 1 Requirements Matrix					
Requirement ID	Requirement Description	Category	Priority	Status	Notes/Comments
REQ-001	Create a Detailed Home Comparison Tool for comparing multiple homes based on price, features, size, etc.	Functional	High	In Progress	Dashboard needs to support side-by-side comparison for at least 2 homes.
REQ-002	Implement a Real-Time ROI Calculator that takes user inputs and provides ROI metrics in real time.	Functional	High	Not Started	Tool must include purchase price, rental income, and other parameters for accurate ROI calculation.
REQ-003	Develop the Neighborhood Insights Dashboard for providing information on schools, crime, demographics, etc.	Functional	High	In Progress	API integrations are being tested for retrieving local amenities and neighborhood data.

REQ-004	Ensure the User Interface (UI) is responsive and adaptive to different devices and screen sizes.	Interface Requirement	High	In Progress	Design for mobile responsiveness is in progress; mockups completed.
REQ-005	Integrate external APIs such as Realty in US, Crime Data, SchoolDigger, etc., for real estate and neighborhood data.	Interface Requirement	High	In Progress	Testing on external APIs is ongoing, including handling rate limits and data accuracy.
REQ-006	Provide a Secure API Gateway to manage all external API connections and ensure data security.	Interface Requirement	Medium	Not Started	API gateway setup needed to ensure safe and secure communication between system and third-party APIs.
REQ-007	Ensure system Performance supports multiple concurrent users and fast data retrieval.	Performance	High	Not Started	Need to implement caching and efficient API queries to maintain performance.

REQ-008	System must have 99.9% availability excluding maintenance periods.	Performance	Medium	Not Started	Uptime monitoring and logging mechanisms required.
REQ-009	Complete the Requirements Document for the real estate platform.	Documentation	High	Completed	Document reviewed by advisor, approved.
REQ-010	Complete the Design Document, including system architecture and detailed diagrams.	Documentation	High	Completed	Includes front-end wireframes and API integration diagrams.
REQ-011	Complete the Test Plan Document, outlining test cases for key features such as search, comparison, and ROI tools.	Documentation	High	Completed	Includes basic test cases for the main features; will add more in future iterations.
REQ-012	Implement basic error handling for failed or slow API requests to ensure smooth user experience.	Functional	Medium	In Progress	Basic error handling in place; needs further refinement for various API failure scenarios.

REQ-013	Ensure the system adheres to real estate regulations for data privacy and security, as well as legal requirements.	Functional	Medium	Not Started	Regulatory compliance needed for handling user data and transactions.
REQ-014	Ensure that all user data is stored and managed securely, adhering to data privacy laws and best practices.	Functional	High	Not Started	Secure authentication and encryption of sensitive data (user login, property searches) must be implemented.
REQ-015	Implement an intuitive and easy-to-use search function for properties based on ZIP code or address.	Functional	High	In Progress	Search function to be linked to the external API (Realty in US); initial implementation tested.