

Sydney Wertz

Trevor Redding

Ethan Greene

Madison Dabbs



Project Proposal Report

Table of Contents

Summary of Work Completed	2
Overview of Project	3
Functional Requirements	4
Non-Functional Requirements	5
Activity Diagrams	6
Use Case Diagram	7
Use Case Specifications	8
Scenarios	11
System Design	12
Database Specifications	15
User Interface	17
Challenges Faced/Backup Plan	20

Summary of Work Completed

Frontend

The following pages were created: Home/Login, About, Contact, Register, House Listings, and Property Details pages, as well as the relevant style sheet to properly format each page. A template was created to quickly add the navbar to all pages made. An AI Chatbot overlay and relevant script was applied to the contact page for testing. Each page was created with ‘empty sockets’ to ensure smooth connection to backend database retrievals.

Backend

The databases for the user, user country, and houses were created in PHP, along with methods to access this information. The user database features methods to get user information from the website and create an account in the database. Methods were also added in order to allow users to login and for that login information to be validated in the database. The model view control paradigm was used to allow seamless transitions within the Dukedom website.

Other

Throughout the development process, all of the necessary reports, including the Software Requirement Specification Document, and diagrams, such as the Activity and Use Case diagrams, have been generated in order to facilitate a smooth development process.

Overview of Project

Homeownership is becoming an increasingly harder goal to achieve. The median U.S. household income does not match the inflation rate of housing costs, barring most Americans from being able to afford the milestone of purchasing a home.

Dukedom will help assuage this by getting more people into the real estate market, establishing themselves, and growing their investment portfolios in a diversifying direction. As opposed to a hedge fund, users can purchase properties directly, sharing the burden of cost with other users to break into the market without the full amount of money needed to outright purchase. It will boost their credit for house-buying, allow for dividends from possible rental agreements, and allow for consensual buyout of other purchasers over time.

Dukedom aims to provide users with the software necessary to piecemeal purchase properties to add to a portfolio as well as post properties to the software for other users to purchase. All of this will be achieved through a streamlined and professional user interface.

Functional Requirements

- Users shall be able to search for posted listings and sort them by area and price range.
- Users will be able to post a detailed listing, with a photo and all other necessary information, for others to purchase shares from.
- All users shall be able to view and manage all shares currently owned.
- Users who post shares must be able to view and edit their posts.
- Users must be able to "follow" a listing through the interface to track changes in the property, either by the original poster or Dukedom.
- Users must log in on the website using their Dukedom account in order to access the portfolio attached to the account and to manage assets.
- Users shall register for an account on the Dukedom website using their email and a strong password.
- Users should be able to contact in-house support for all questions regarding Dukedom's software. This support should be facilitated by Artificial Intelligence learning, with the option to contact a Dukedom employee.
- Users should be able to delete their account if there are no assets in their portfolio.

Non-Functional Requirements

- The system shall be published under the Dukedom domain name. Users should be able to access the application from all modern Internet browsers.
- The user interface shall be easy-to-understand and follow, especially for users of other real estate systems. The user interface will create a feeling of familiarity to new users coming to Dukedom from similar systems to better allow for useability.
- When the system is publicly-accessible, all required legal documentation relating to both real estate and the stock market will be present and easy to navigate. All users of Dukedom must agree to the appropriate terms and conditions provided in order to use the system.

Activity Diagrams

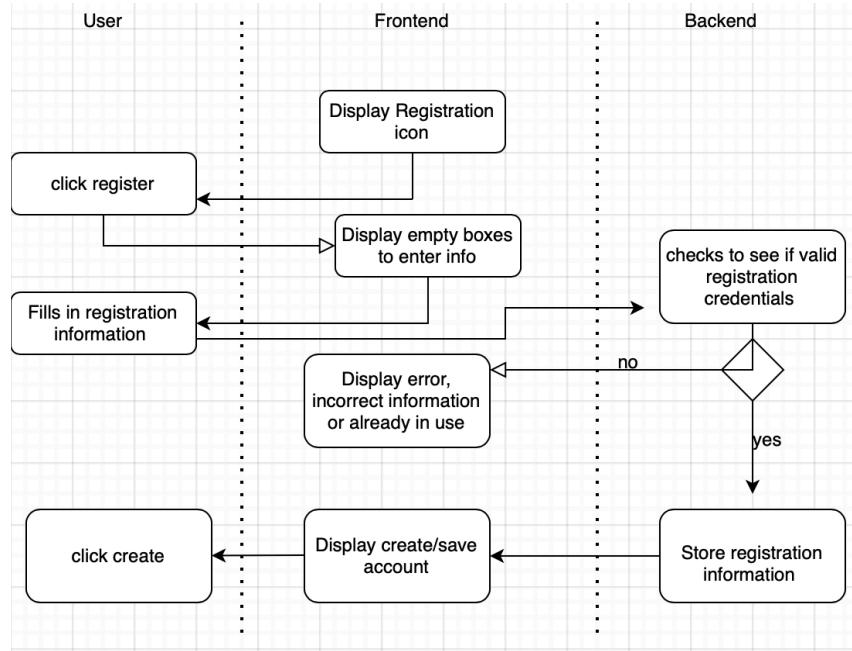


Figure 1: Activity Diagram for User Registration

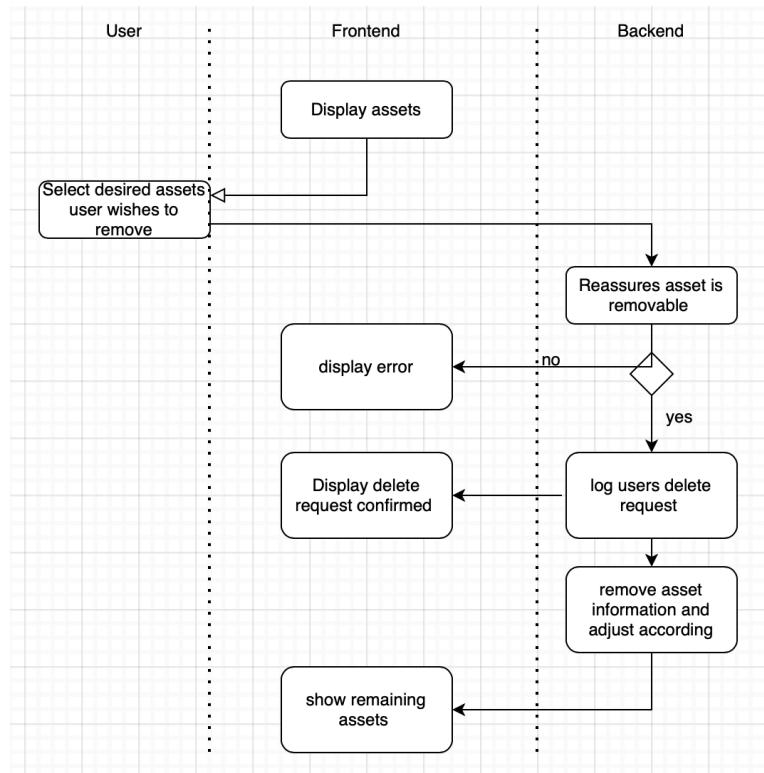


Figure 2: Activity Diagram for Removing A Home from the Database

Use Case Diagram

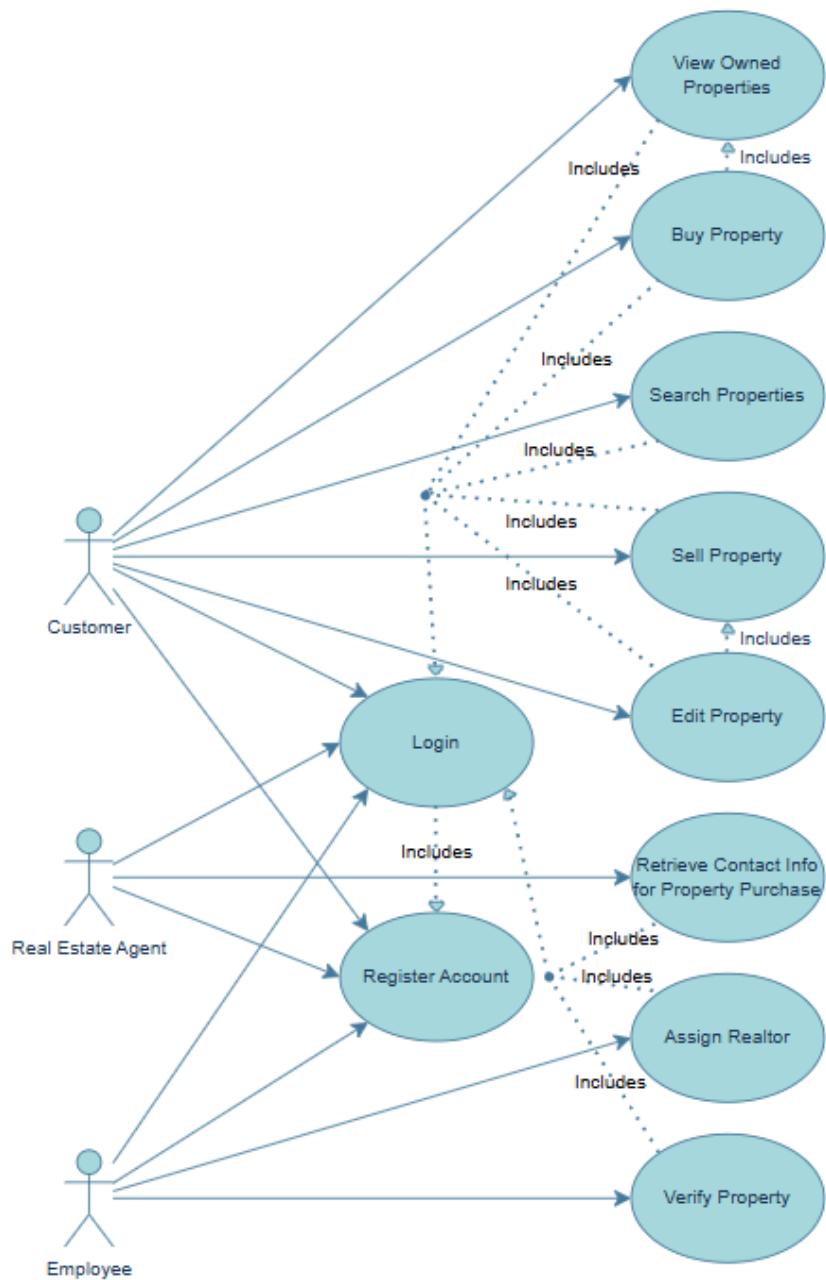


Figure 3: Use Case Diagram for Dukedom

Use Case Specifications

Login

Summary: User can input credentials in order to access Dukedom Account

Preconditions: Dukedom website must be opened on an internet-connected device

User must already have an account registered with Dukedom

Trigger: User opens the Dukedom website

Course of Events:

1. User enters login and password in prompted fields, which are always on the front page upon opening the website
2. User clicks the 'Submit' button
3. Dukedom checks the credentials against the database of registered users
4. Information matches and user is able to access account and associated information

Alternate Paths:

- 3a. User information provided is not stored in database, display error code and return to step 1

Show Available Properties

Summary: User can click a button on the Navigation Bar that will display all of the currently available properties in the database.

Preconditions: Dukedom website must be opened on an internet-connected device

Trigger: User clicks on the 'Buy' button on the Navigation Bar of the website.

Course of Events:

1. User opens the website, which displays the Home Page
2. User clicks on the 'Buy' button on the Navigation Bar
3. Dukedom moves from the Home Page to the Buy Page, which displays all currently available properties in the database

Alternate Paths:

- 1a. User logs in using valid Dukedom account and then continues to step 2
- 3a. User can also search for properties using the Search Bar on the Navigation Bar

View Property Details

Summary: User can view specific details relating to the chosen property clicked on

Preconditions: Dukedom website must be opened on an internet-connected device

Trigger: User clicks on the box that the desired property is displayed in

Course of Events:

1. The user will be presented with all of the available properties on the database from the 'Buy' page
2. The user clicks anywhere on the box containing the desired property
3. Dukedom moves from the 'Buy' page to display all of the selected property's details, including overall price, shares available, and buy-in amount
4. The user clicks the 'Purchase Shares' button to acquire shares from the property

Alternate Paths:

- 3a. User clicks on the 'Back' button to return to the 'Buy' page and view all other properties

Scenarios

Scenario 1 - New User Wants to Register

Individual: A new user with no account currently registered with Dukedom

Equipment: A device that is connected to the Internet

Scenario

1. The user enters the Dukedom URL into an internet browser to arrive at the website
2. The user clicks the button stating 'Sign Up Now' under the login boxes on the Home Page
3. The user enters in a valid email address and creates a password
4. The user is returned to the Home Page and is able to access account features

Scenario 2 - User Wants to Find Property

Individual: A Dukedom user who wishes to acquire property

Equipment: A device that is connected to the Internet

Scenario

1. The user navigates to the Dukedom website and logs in
2. The user clicks the 'Buy' button to be taken to the list of all properties in the database
3. The user selects the desired property in order to view more details
4. The user adjusts how many shares of the property will be purchased, aligning with the set minimum buy in
5. The user clicks the 'Purchase Shares' button to proceed to the payment and add the shares to the account portfolio
6. If the user is not logged in, the user must log in now or Dukedom will not proceed with the purchase

System Design

Figure 4, shown below, outlines our system design based on the requirements, including the optional employee interface requirement, as listed above.

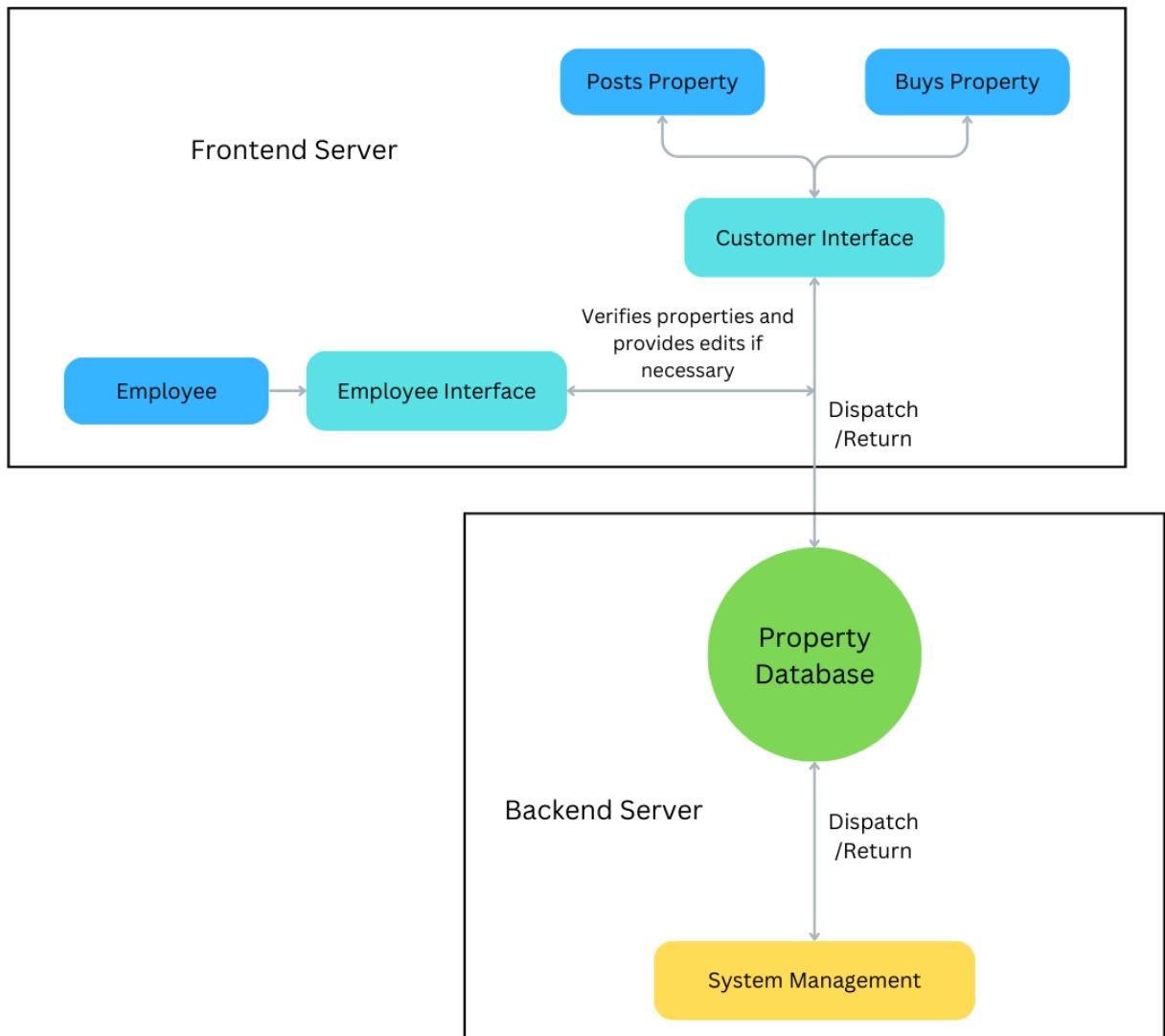


Figure 4: An abstract depiction of Dukedom's proposed system design. Each of the interfaces can be accessed from any modern internet browser on multiple computers at the same time from separate accounts. Every interface and thus account feeds into a central database that a system administrator can directly access to perform maintenance and other necessary tasks.

The figure below showcases our system design along with the optional employee-side interface, wherein all of the property data required to successfully operate Dukedom's services are hosted in a separate server than the interfaces that users, both customers and potential employees, will access. This server contains backups of the property database and allows for edits to the database, such as a change to what information is required for a property to be posted, by a system administrator.

The frontend server contains the web-based graphical interface that all users will interact with. Customers will be able to post and or purchase property remotely through the customer interface, which sends the necessary information to the property database in the backend server. Copies of information on the database can also be dispatched from the database to the frontend server to be viewed and interacted with through this interface (indicated via the black lines with arrows on the diagram).

The employee interface, if implemented, will operate in a similar way in terms of interacting with the database. The only difference in interface is that employees will have the capacity to correct property postings if necessary before they are submitted to the database. The customer and employee interfaces are independent of each other and are also independent of multiple instances of the same interface.

This system requires the property database, server-side scripting to interact with this database, and a web server software to host both the interfaces and this database. One possible system configuration would utilize PHP, mySQL, HTML, and DreamHost.

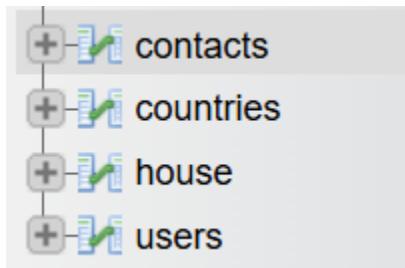
- The server-side programming may be implemented with PHP, which would allow for database retrieval, multiple levels of database access, and all required customer, employee, and system administration interfaces.

- The property database itself can be created using mySQL in order to facilitate property data storage and editing.
- HTML can be used to implement the required interfaces for displaying and editing the property database.
- The web server may be hosted on DreamHost, which can support multiple users accessing the system at the same time.

The combination of all of the above products (PHP, mySQL, HTML, DreamHost), and the team's own coding, the requirements outlined above will be satisfied.

Database Specifications

Our company utilizes the seamless integration of MySQL and PHP. Through doing so we have been able to create many databases to allow users to register, login, and view available houses. The current databases are pictured below:



Contacts:

- Form: Comments left under our contact page will go here
- Email: Allows Dukedom to return to customers that have contacted us

form	email
Form Test	
Hi database! example@email.com	
Test	email
Test	Email
Hi	Ethan@email.com

Countries: A list of countries allowing us to populate a drop-down list during registration

	<input type="button" value="←"/> <input type="button" value="→"/>	countryCode	countryName
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	AD	Andorra
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	AE	United Arab Emirates
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	AF	Afghanistan
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	AG	Antigua and Barbuda
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	AI	Anguilla
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	AL	Albania
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	AM	Armenia
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	AN	Netherlands Antilles
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	AO	Angola
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	AQ	Antarctica
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	AR	Argentina
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	AS	American Samoa
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	AT	Austria
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	AU	Australia
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	AW	Aruba
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	AX	Aland Islands
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	AZ	Azerbaijan
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	BA	Bosnia and Herzegovina
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	BB	Barbados
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	BD	Bangladesh

House:

- Address: address of a particular house.
- Path: allows us to show a picture of a particular house.
- Percent: shows the available percentage of a particular house.
- Price: shows the available price of a particular house.
- Buy_in: Shows the minimum price to buy stock in a particular house.

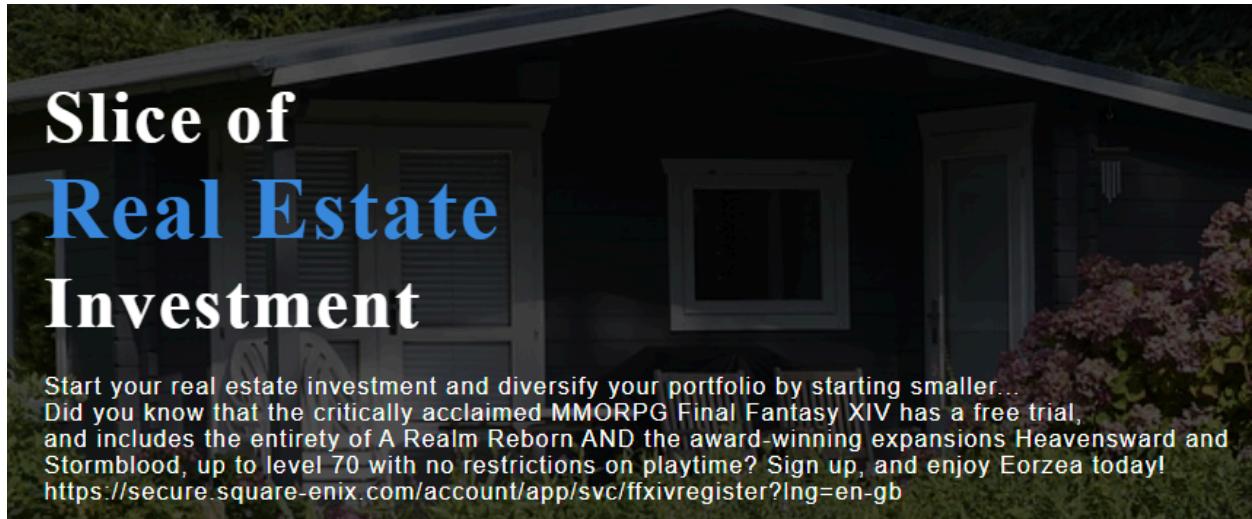
address	path	percent	price	buy_in
5 Sapphire Street, Astera, TNW, 00326	house_3.jpeg	86	86840	1000

User: typical database for a registered user.

username	password	first_name	last_name	address	city	state	postal	country	phone	email
Test1	Pass	First	Last	123 Sesame Street	Charleston	SC	29403	Andorra	1234567890	email@example.com
Ethan	a	Ethan	Greene	123 sesame street	Chs	Sc	29403	Gibraltar	1234567890	ethan@example

User Interface

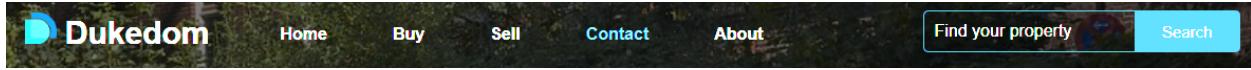
Dukedom's user interface is designed with an aim towards usability, readability and a pleasing aesthetic. The readability is mostly comprised of the contrast using a black or dimmed background with white writing overlaid on top of it:



Starting with the usability, the buttons need to be functional, and large enough for a user to press easily. Additionally, it needs to be clearly visible from the background of the page. For additional indication that it is a button, the cursor must become a pointer when hovered over, and ideally have some additional indicator that it is a button. In Dukedom's case, this is a color change on the buttons themselves.

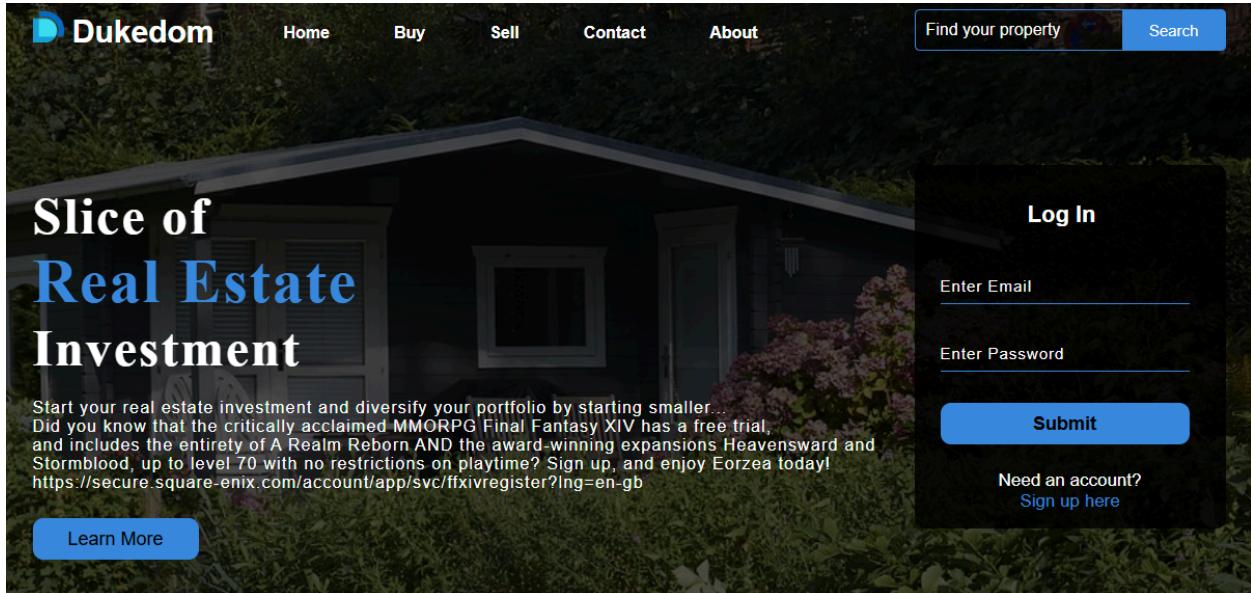


Another example of the usability is in the navigation bar, with each option similarly changing colors for the option of them being pressed. The navigation also follows standard conventions in that pressing the logo or brand name also returns you to the home page, as well as the locations of each element.

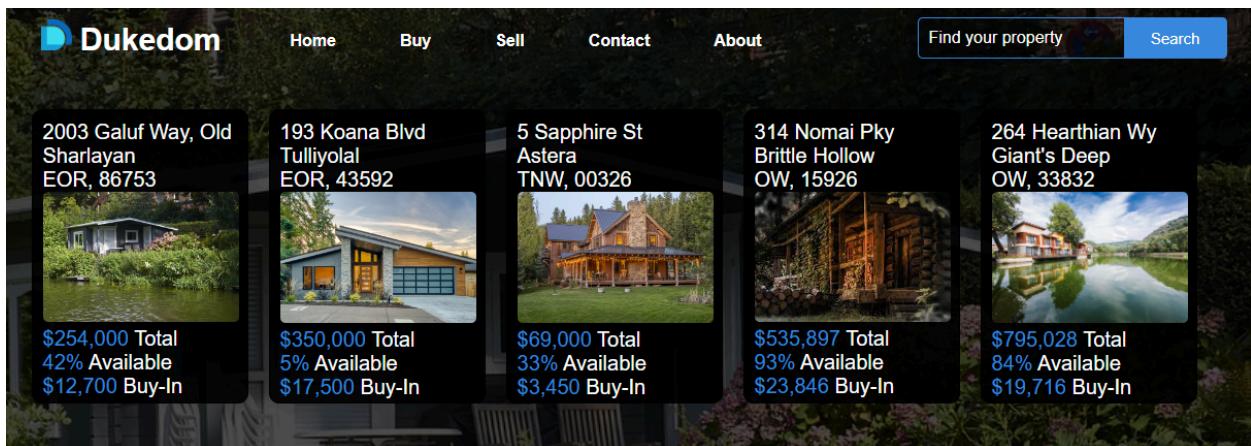


The header features the Dukedom logo with a blue 'D' icon, followed by the word 'Dukedom'. To the right are navigation links: Home, Buy, Sell, Contact, and About. A search bar with the placeholder 'Find your property' and a blue 'Search' button are on the far right.

Below is a compilation of each page to fully showcase the aesthetic:

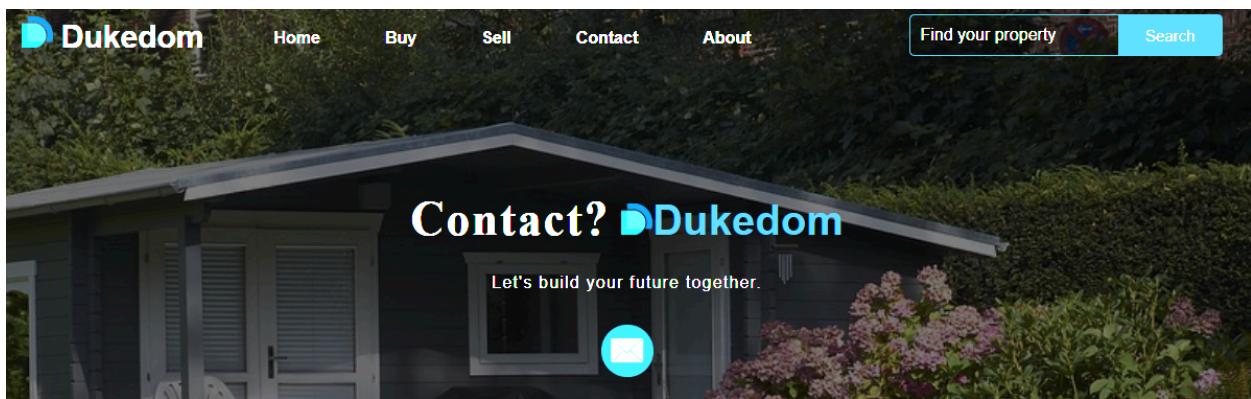


This page features a large image of a modern house with a dark exterior and light-colored trim. Overlaid on the left is a large blue text block containing the title 'Slice of Real Estate Investment'. Below the title is a promotional message about Final Fantasy XIV. At the bottom left is a blue 'Learn More' button. On the right side, there's a 'Log In' form with fields for 'Enter Email' and 'Enter Password', a blue 'Submit' button, and links for 'Need an account?' and 'Sign up here'.

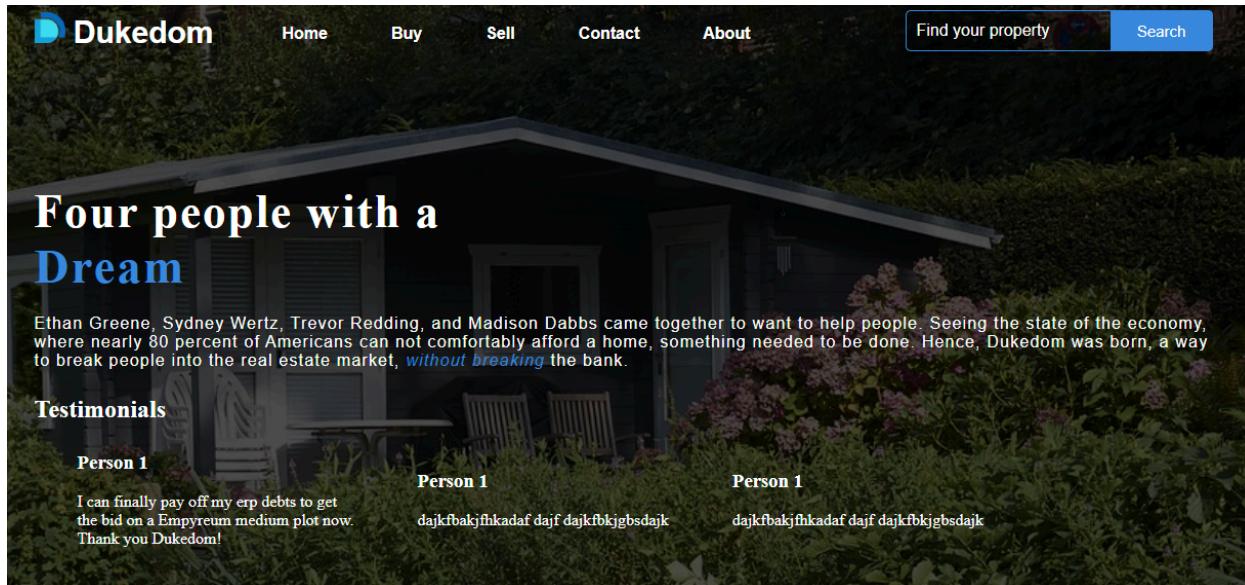


This page displays five property cards against a background of a lake and mountains. Each card includes a thumbnail image, address, total value, availability percentage, and buy-in amount.

Property Address	Total Value	Availability	Buy-In
2003 Galuf Way, Old Sharlayan EOR, 86753	\$254,000 Total	42% Available	\$12,700 Buy-In
193 Koana Blvd Tulliyolal EOR, 43592	\$350,000 Total	5% Available	\$17,500 Buy-In
5 Sapphire St Astera TNW, 00326	\$69,000 Total	33% Available	\$3,450 Buy-In
314 Nomai Pky Brittle Hollow OW, 15926	\$535,897 Total	93% Available	\$23,846 Buy-In
264 Hearthian Wy Giant's Deep OW, 33832	\$795,028 Total	84% Available	\$19,716 Buy-In



This page features a large image of a modern house with a dark exterior and light-colored trim. Overlaid in the center is the text 'Contact?  Dukedom' and the tagline 'Let's build your future together.' Below the tagline is a blue envelope icon.



Dukedom

Home Buy Sell Contact About

Find your property Search

Four people with a Dream

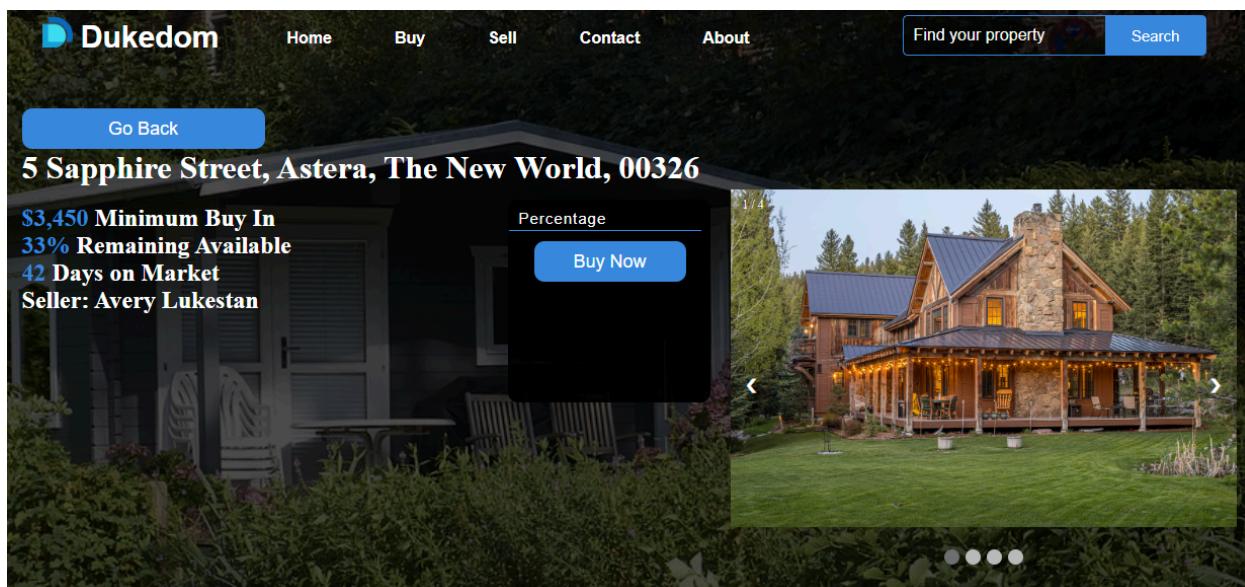
Ethan Greene, Sydney Wertz, Trevor Redding, and Madison Dabbs came together to want to help people. Seeing the state of the economy, where nearly 80 percent of Americans can not comfortably afford a home, something needed to be done. Hence, Dukedom was born, a way to break people into the real estate market, *without breaking* the bank.

Testimonials

Person 1
I can finally pay off my erp debts to get the bid on a Empyreum medium plot now. Thank you Dukedom!

Person 1
dajkfbakjfhkadaf dajf dajkfbkjgbsdajk

Person 1
dajkfbakjfhkadaf dajf dajkfbkjgbsdajk



Dukedom

Home Buy Sell Contact About

Find your property Search

Go Back

5 Sapphire Street, Astera, The New World, 00326

\$3,450 Minimum Buy In
33% Remaining Available
42 Days on Market
Seller: Avery Lukestan

Percentage

Buy Now



1 / 4

< >

Challenges Faced/Backup Plan

In order to make the creation of Dukedom go well we have divided the front and backend development among two of our group's members. When the frontend developer had created a new page we faced many issues getting it to work properly with the backend. This is because the frontend developer's system used a file management different from that of our backend. Every time a new page had been created, our backend developer had to go through every created page and update certain fields in order to integrate the updated frontend. With the use of a centralized header, though, our backend developer was able to call an include statement on each view, so that with every update all that needed to change was the single header file. This change is visualized below:

The diagram illustrates the structure of a PHP header file. It shows a hierarchical tree of HTML tags with specific code snippets highlighted in green. A large downward arrow points from the header code to a separate box containing the PHP include statement. The code includes elements like the DOCTYPE declaration, head, body, main, navbar, menu, and search sections.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Dukedom</title>
    <link rel="stylesheet" type="text/css" href="/Php362Project/style.css">
  <body>
    <div class="main">
      <div class="navbar">
        <div class="icon">
          <h2 class="logo"><a href="../index.php"><img src= "../Dukedom_Logo.png" alt="Dukedom brand logo, a blue D with a blue crown on top."></a></h2><!--ends icon-->
        </div>
        <div class="menu">
          <ul>
            <li><a href="../index.php">Home</a></li>
            <li><a href="../manage_buy">Buy</a></li>
            <li><a href="#">Sell</a></li>
            <li><a href="../manage_contact">Contact</a></li>
            <li><a href="../manage_about">About</a></li><!--THIS IS HOW YOU LINK TO OTHER WEBSITES/PAGES-->
          </ul>
        </div><!--ends menu-->
        <div class="search">
          <input class="srch" type="search" name="" placeholder="Find your property">
          <a href="#"><button class="btn">Search</button></a>
        </div><!--ends search-->
      </div><!--ends navbar-->
```

↓

```
<?PHP include (' ../view./header.php' ) ?>
```

Other challenges faced finding common times that everyone was available to meet as we were all busy college students. We overcame this by listing all of our typical available times and consistently meeting at those times so that we did not fall behind.

Our backup was that if we weren't able to develop certain features to scale back. For our first iteration, users are unable to list their own houses, rather users can only buy from Dukedom affiliated properties. This was a smart choice as we were able to get a system that ran well by the date requested, and the team successfully adapted to unwanted outcomes, such as time constraints. Though we know we may have to scale back certain features, we look forward to bringing our customers the company we envisioned and will continue to make good on these promises.