

Realest

Realest

Dinesh Balakrishnan, Eshwar Reddy Gadi, Daniel Pandyan, Arjun Som

CSCE 482 Senior Capstone Design, Spring 2025

Department of Computer Science and Engineering

Texas A&M University

Abstract	4
1 Introduction	5
1.1 Case Scenarios	6
1.2 Goals and Constraints	7
1.3 Solution Summary	8
1.4 Evaluation Summary	9
2 Related Work	10
3 Requirements	13
3.1 User Stories	14
3.2 Definition of Success	16
4 Engineering Standards, Regulations, and Considerations	18
4.1 Engineering Standards	19
4.2 Applicable Regulations	20
4.3 Environmental and Health/Safety Considerations	21
4.4 Ethical, Social, and Political Considerations	22
5 Design Exploration	24
5.1 Comparison of Potential Solutions	25
5.2 Lo-fi Prototyping	29
5.3 Pilot Studies	33
5.4 Access for All Considerations	35
6 System Design	37
6.1 Functional Design	38
7. Evaluation	44
7.1 Functionality Evaluation	45
7.2 Usability Evaluation	49
8 Discussion	52
9 Future Work	54
10 Conclusion	56
References	57
Appendix A: Project Management	59

A.1 Team Agreement	60
A.2 Software Development Methodology	63
A.3 Implementation Schedule	65
A.5 Budget	94
Appendix B: Implementation Details	96
Appendix C: User's Manual	103

Abstract

Investment focused realtors, a rapidly growing minority in the real estate industry, are riddled with inefficiencies stemming from the lack of specialized tools designed for property analysis and client management. Current platforms require real estate agents to manually calculate cash flow for each property they process, and because many clients may require different cash flow estimates this workload often grows exponentially. This results in poor prioritization, missed deals, and ultimately client dissatisfaction. As the real-estate industry becomes increasingly data-driven, the absence of a consolidated solution leaves both agents and their investing clients underserved.

We propose our platform, Realest, a cross-platform real-estate application designed to bridge the gap between agents and their detail-oriented clients. Through automated cash flow projections, swiping-based property selections and filtering, as well as Customer Relationship Management (CRM) dashboards, our applications bundle the entire agent-client real estate experience. CRM refers to the practices, strategies and technologies that companies use to manage and analyze customer interactions and data throughout the customer lifecycle. To evaluate our solution, we conducted both functional and usability testing. **Unit and integration tests confirmed 80% accuracy in cash flow calculations and sub-second system responsiveness on average.** A structured user study with realtors and investors **revealed that agents experienced a 40% reduction in time spent on analysis per client, while investors reported that almost 90% prefer the swipe interface over regular home searches.** Our solution to the evolving data-driven real estate environment is a scalable platform that empowers realtors with analytics and communication tools, and equips investors with clear, actionable insights, streamlining a traditionally complex process into a user-friendly experience.

1 Introduction

Investment real estate agents, who make up about 10% of all realtors, face an enormous challenge in managing their day-to-day operations efficiently. The problem at the core is the lack of a consolidated platform that makes investment property searching and analysis simple. Agents must manage numerous time-consuming tasks, such as:

- Manually analyzing cash flow and financials on numerous properties
- Securing buyer approvals and financing
- Finding listings that match each client's unique investment criteria

Not having a powerful, all-in-one solution for investment real estate agents has important consequences:

- **Lost time:** Agents tend to spend approximately 65% of their workday on routine and administrative work instead of on income-generating activities such as client acquisition and deal closure (National Association of Realtors, 2022).
- **Missed opportunities:** Inefficient property matching procedures result in lost investment opportunities, which amount to lost income for the investor and agent. Over 40% of agents attribute property matching inefficiencies as one of the major reasons for lost deals (Colibri Real Estate, 2021).
- **Client dissatisfaction:** The long and complex process of investment property search and analysis can lead to frustration and dissatisfaction for the clients, which can damage the reputation of the agent and his/her future business prospects.

No popular platform exists today that addresses these very significant issues faced by investment-real estate agents. By creating a solution that streamlines property search, calculates cash flow automatically, and provides investment real estate agents with strong capabilities, we can significantly improve their productivity and efficiency. Not only would this release agents from valuable time, but also make the process of purchasing easier for clients, leading to increased client satisfaction and more deals closing.

1.1 Case Scenarios

Case 1: Rory is a realtor working at a mid-sized real estate firm with 10 years of experience. Despite previous success with collecting information manually and delivering it to their clients, Rory finds that the vast amount of data online from platforms such as Airbnb and Zillow has become overwhelming to manage effectively. Rory decides they prefer the interpersonal and sales aspects of working in the real estate industry, rather than the technical and “spreadsheet heavy” side.

- Challenge:
 - Each listing requires a large amount of work to give a comprehensive analysis of the investment.
 - The realtor does not have a platform to effectively market their listings.
 - Finding potential listings and aggregating them by category of client is cumbersome.
- Solution:
 - The realtor utilizes the platform to browse listings.
 - When a favorable listing is encountered, the realtor can assign the listing to a preferred group. This preferred group system allows for the realtor to pick multiple listings for different clients.
 - The platform generates market analysis, cash flow projections, and investment potential.
 - The realtor gains market exposure and saves time from avoiding the manual calculations they would otherwise have to provide.

Case 2: David is a new real estate investor looking to purchase his first rental property. He is hoping to turn a profit by renting out a property. Ideally this would be a property that requires minimal time and effort to reach profitability. After connecting with a realtor they met, they receive an invitation to view the realtor’s offerings through Realest’s application. David is told that there are several promising properties that match his criteria and budget.

- Challenge:
 - The investor requires a platform with a thorough breakdown of cash flow projections, market analysis, and investment potential.
 - Lacks experience evaluating real estate investment opportunities
- Solution:
 - The investor utilizes our platform to quickly gain insights on the listings provided by their realtor.
 - As investments inspired by the platform pay off, the investor creates a stronger relationship with the realtors on the platform.

1.2 Goals and Constraints

The primary goal is to create a platform that bridges the gap between realtors and investors. It will provide:

1. Enhanced Listing Visibility: Realtors can manage their listings and create more attractive offerings.
2. Strengthen Investor Confidence: Investors obtain access to highly detailed, data-driven insights about the listings and avoid manual market research.
3. Streamlines the Real Estate Investment Process: Rather than provide materials through email or other means, realtors can deliver information about their listings directly to clients.

High Level Deliverables:

- Property Listings Dashboard: Interface for realtors to manage their listings and provide clients with updated information if needed.
- Investor UI Interface: Interface for potential investors to interact with listings data. This interface would also allow for the management of the listings that the investor is interested in.
- Market Analysis & Cash Flow Analysis Tools: Provides comparable property data, market trend analysis as well as automated cash flow projects (including cap rate and ROI).
- Reporting and Visualization: Interactive charts, graphs, and reports to present digestible insights for both investors and realtors.

Exclusions:

- Does not provide legal or financial advice.
- Does not provide direct transaction handling.
- Only provides insights for listings provided by realtors.

Constraints:

- Concerning data quality and real-time updates, there are constraints based on how much money is spent..
- User privacy for both realtors and investors limits the information that can be obtained.

1.3 Solution Summary

We built a preliminary solution based on discussions in meetings held with the client. The following highlights the key features of our proposed preliminary solution as approved by the client.

1. Setup & Customization

- The realtor invites their buyer to the app.
- During setup, the buyer selects their realtor, linking them within the app.
- The app automatically applies the realtor's branding (logo, colors, etc).

2. Smart Property Filtering & Recommendations

- Realtors can browse available listings, applying filters to find properties that match the buyer's investment goals (e.g., rental yield, cash flow potential, location preferences).
- The app integrates MLS data and automated investment-focused research (e.g., estimated rental income, property taxes, neighborhood appreciation trends).
- Realtors handpick properties and send them to buyers for review.

3. Interactive Property Selection (Tinder-Style Swiping for Buyers)

- Buyers receive recommended properties in a simple swipe-based interface:
- Each listing includes a **cash flow analysis, detailed expense breakdown, and investment potential insights** tailored to the buyer's strategy.

4. Decision & Next Steps

- If a buyer is interested in a property, the realtor gets notified instantly.

The solution helps address the need of the agent by showing them a cash flow preview that allows them to easily show the buyer properties. The solution also helps the buyer because it provides a simple way to view the agent's recommended properties instead of conversing to emails and other manual forms of communication.

1.4 Evaluation Summary

The primary method for our system to be functionally evaluated will be to do unit, integration, and system testing.

Additionally, In order for the system to be functionally evaluated the following metrics will be evaluated:

1. Accuracy: Ensure that property listings, cash flow analysis, and investment insights are correctly presented.
2. Responsiveness: Ensure that the application is speedy and that filtering and looking through the listings takes a minimal amount of time that does not cause frustration to the user
3. Usability: Ensure that the application is easy to use and intuitive so that buyers and agents won't spend time struggling with the app to accomplish tasks.
4. Integration: Examine how well the system is able to pull in data from the MLS and also make sure that metrics and communication between buyer and agent is secure and fault free.

For the behavioral evaluation of the system the following metrics will be evaluated:

1. User satisfaction: While the user (buyers and agents) are using the solution, ensure that the application is actually improving their work.
2. Efficiency: Track whether the application is able to reduce the amount of back and forth communication between both parties and is also able to speed up the process of finding a potential investment property.

2 Related Work

Many real estate professionals today struggle to manage tasks, analyze investment properties, and communicate effectively with clients. Issues such as disorganized workflows and weak client engagement often arise from fragmented communication methods like text, phone, and email, while evaluating investment properties has become more complex due to evolving financing structures and varying factors such as vacancy and management fees that differ significantly between markets (National Association of Realtors, 2023; Zillow Economic Research, 2023). These challenges directly motivate our project's goal of building a unified platform that simplifies realtor operations and strengthens client communication.

2.1 Analysis of Existing Platforms

Several platforms try to tackle parts of these problems, but none fully cover what realtors need. DealCheck, for example, is strong at running financial numbers like cash flow and ROI (DealCheck, 2023). However, it mainly targets solo investors and wholesalers, not realtors managing client interactions.

Another example is Roofstock, which connects investors to rental properties and helps with buying and managing them (Johnson, Williams, & Roberts, 2022). But it's designed for direct-to-investor deals and does not facilitate personalized realtor-client collaboration.

Zilculator automates property analysis and pulls data from sources like MLS and Zillow (Zilculator, 2023). It offers strong calculations, but lacks built-in tools for client engagement and relationship management.

CRM systems like PropertyBase help manage leads and deals (PropertyBase, 2024; Garcia & Lee, 2024). While they cover basic client management, they do not integrate investment analysis or support interactive property selection, both essential for realtor-investor workflows.

Investor-focused platforms like Arrived and Fundrise offer fractional ownership and accessible investing (Arrived, 2023; Chen, 2023; Fundrise, 2022; Martinez, 2024), but they bypass realtors entirely and focus only on direct-to-investor models.

Another key platform that has come into existence during the development of our project that is certainly worth a mention is Bigger Pockets (specifically the BiggerPockets Deal Finder).

BiggerPockets Deal Finder provides a home search with complex cash flow analysis for each property. (BiggerPockets, 2024) However, like other investor-focused platforms, it emphasizes direct access for individual buyers rather than supporting realtor-client collaboration.

2.2 Novelty and Justification

These solutions, while effective in their niches, leave a gap in providing a fully integrated tool for realtors that combines financial analysis, client management, and interactive property engagement. Our platform addresses this gap directly.

Recommender systems in real estate often have trouble suggesting good options when there's little user history (Al-Bashrawi, Ahmad, Ismail, & Anadale, 2021; Wilson, Martinez, & Chang, 2022). Our platform solves this by letting realtors guide and create the first recommendations, following case-based reasoning approaches (Rodriguez & Kim, 2023; Wang, Guo, Wang, & Zhang, 2016), allowing immediate relevance even with new users.

Our platform's interface design follows mobile-first and accessible design principles. Research shows over 85% of agents rely on mobile devices to interact with clients (Statista, 2023; Thompson & Davis, 2023). Swipe-based designs boost engagement and decision-making efficiency (Thompson, 2023; Jackson, 2024). We also align with Google's Material Design Guidelines (Google, 2023) to maintain a modern, intuitive experience as many of the similar options on the market fail to account for the fact that many real estate investors tend to be older in age and therefore may struggle to use complicated modern apps.

3 Requirements

Core Requirements:

1. Dual user interfaces - separate UIs optimized for realtors and investors
2. Property listings dashboard for realtors to manage listings and provide updates
3. Investor UI with simplified interface to view and interact with recommended listings
4. Market analysis and cash flow analysis tools to provide detailed financial insights on properties
5. Reporting and data visualization of key metrics for both realtors and investors

Realtor-Specific Requirements:

1. Ability to send property listings to specific investors
2. View investor decisions/feedback on recommended properties
3. Adjust cash flow projections before sharing with investors
4. Filter listings by tags and send to groups of investors
5. Client management dashboard to track investor activity and follow-ups

Investor-Specific Requirements:

1. Receive property recommendations from realtors
2. Swipe-based interface to quickly review and decide on properties
3. View detailed cash flow and market analysis for each listing
4. Provide feedback to realtors on recommended properties
5. Save favorite properties and access previously swiped listings

3.1 User Stories

Our two personas, realtor and investor, drive our user stories. Realtors are the primary users who handle property listings, engage with clients, and leverage the tools offered by the platform to streamline their workflow. Investors are the final customers who use the app to review property recommendations presented by their realtor and provide feedback. The user stories that follow are written around the two personas and their main requirements.

Blue is the user/persona. **Green** is the “what”. **Purple** is the “why”.

1. As a **realtor**, I want to **be able to send my buyers property listings** that I think they might want **so that I can see if they approve of it or not**.
2. As a **realtor**, I want to **see what properties that my clients like**, so that **I can look into the property myself and discuss moving forward with it**.
3. As a **realtor**, I want to **see automatically calculated cash flow insights for every single house that comes from the MLS feed** so that **I don't have to spend time manually calculating it**.
4. As a **realtor**, If I come across a property on the app that I like, I want to be able to search up users by a tag (ie, if there are multiple buyers that want a duplex, I can send it to temp by just looking up the tag.) and send, so that **I don't have to individually send it to multiple people by myself**.
5. As a **realtor**, I want to **be able to look at the most recently changed listings on the MLS** (last 24hrs, 1 week, etc.) so that **I am not going through properties I have already seen**.
6. As a **realtor**, I want to **see a dashboard of all my clients on one page**, so that **I can see what properties I need to review and follow up on**.
7. As a **realtor**, I want to **be able to adjust the cash flow insights myself before sending the property (with insights) to the buyer**. (For example, if a property is 30 min out of town, it might have a higher vacancy rate than a property in the middle of town which will affect the cash flow.) so that **the buyer can know what a realistic cash flow is based on my perspective**.
8. As a **realtor**, I want to **be able to see what houses my clients did not like** that I sent them so that **I can learn why and maybe follow up to learn their style**.
9. As a **realtor**, I want to **be able to generate and send a link to my client and make their setup easy** so that **I don't have to waste too much time with them not looking at houses**.
10. As a **realtor**, when I am not using my phone to look at the app, I want to be able to **view it on my desktop** so that **I can see more features and interact with the interface on a bigger screen**.
11. As a **buyer**, I want to **be able to view recommended listing or listing that fit what I am looking for** so that **I can send them to my realtor**
12. As a **buyer**, I want to **be able to send listings to my realtor** so that **they can look into whether the property would be a good fit for what I am looking for** and maybe move forward with a tour

13. As a **buyer**, I want a **highly simplified interface on my mobile phone** so that **I don't have to spend too much time learning how to use the app**
14. As a **buyer**, I want to **be able to view my realtor's recommended houses** and be able to let **him know if I like or dislike it** so that **we can move forward with the ones I do like.**
15. As a **buyer**, I would **like to make notes on each property** so that **my realtor can see why I liked or disliked a certain property**
16. As a **buyer**, I want to **be able to view cash flow insights and adjust some numbers** so I can **make sure the house is going to give me a return on my investment that I expect.**
17. As a **buyer**, I want to **be able to see the listing** that I previously swiped left on so that I can **find it in case I decided later that "maybe I actually like that property"**
18. As a **buyer**, I want **to have a minimal setup experience and not have to worry about passwords** because **I will probably forget it and dont want to waste time giving information by realtor should already know**
19. As a **buyer**, I would **like to view information about a house like when it was made and how many bedrooms it has** so that **I can make an informed decision on whether it will fit what I am looking for.**
20. As a **buyer**, I would also like to **receive an email when my realtor sends me a house to look at** so that **I won't miss it since I might not be looking at the app all the time.**

3.2 Definition of Success

1. **Sending Listings to Buyers** – Realtors can select and send property listings to specific buyers. Buyers receive notifications (in-app and email) about new listings shared with them.
2. **Viewing Buyer Preferences** – Realtors can see a list of properties that each buyer has liked or disliked. Realtors can view buyer comments if any exist on properties for better insight.
3. **Automatic Cash Flow Insights** – Each MLS listing includes an automatically generated cash flow analysis (as specified by Greg). The insights update dynamically with MLS data.
4. **Tag-Based Buyer Search & Bulk Sending** – Realtors can search buyers by tags (e.g., "duplex") and send listings to all matching buyers. The system suggests potential buyers based on their saved preferences.
5. **Recently Updated MLS Listings** – Realtors can filter MLS listings based on recent updates (last 24 hours, 1 week, etc.). They can exclude properties they've already reviewed.
6. **Client Dashboard** – A centralized page displays all clients, their saved properties, and follow-up actions. Actionable insights (e.g., pending homes to review) are highlighted.
7. **Adjusting Cash Flow Insights** – Realtors can modify cash flow calculation variables and see the cash flow calculations change in real time. Changes made by realtors are visible to buyers by default.
8. **Easy Client Onboarding** – Realtors can generate a unique link for buyers to set up their account quickly. Buyers don't need to enter unnecessary information if the realtor already has it.
9. **Desktop Compatibility** – The platform is accessible via desktop with additional features optimized for larger screens.
10. **Viewing Recommended Listings** – Buyers can browse listings recommended by their realtor.
11. **Sending Listings to Realtors** – Buyers can send listings they find to their realtor for review..
12. **Mobile-Friendly Interface** – The mobile app provides a simple, intuitive experience with minimal learning required. This is evaluated by having over 90% of users in testing agreeing that the interface is friendly.
13. **Liking or Disliking Listings** – Buyers can indicate whether they like or dislike a property by swiping left or right.
14. **Adding Notes to Listings** – Buyers can add comments to listings to explain why they like or dislike a property. Realtors can view buyer notes within their dashboard.
15. **Cash Flow Insights for Buyers** – Buyers can view automatically generated cash flow insights. They can adjust certain inputs (e.g., vacancy rate, maintenance costs) to see how it affects cash flow.
16. **Recovering Disliked Listings** – Buyers can revisit previously rejected properties.
17. **Minimal Setup & No Password Hassle** – Buyers can sign up easily using a link from their realtor. Login is simplified (e.g., no password requirement or magic link-based authentication).
18. **Viewing Essential Property Information** – Buyers can see key details (e.g., build year, bedrooms, square footage) for every listing.

19. **Accuracy** - Property listings, cash flow analysis, and investment insights are correctly presented over 95% of the time.
20. **Responsiveness** - Filtering and browsing listings takes under 2 seconds on average. App interactions feel speedy with no noticeable lag.
21. **Latency** - Notifications about new listings or status updates are delivered within 1 minute.
22. **Reliability** - The platform has over 99.9% uptime with no major outages.
23. **Ease of Use** - Over 90% of users rate the app as "easy" or "very easy" to use in post-study surveys.
24. **Efficiency** - Realtors report spending 50% less time on manual property analysis and communication compared to their previous workflow. Buyers find properties they like in under 10 minutes on average.
25. **Clarity** - More than 85% of users say the cash flow insights and property details are "clear" or "very clear".
26. **Overall Satisfaction** - The platform maintains an average rating of 4.5 stars or higher. At least 80% of users say they would recommend it to others.

4 Engineering Standards, Regulations, and Considerations

4.1 Engineering Standards

Our application complies with several key software engineering standards to ensure security, reliability, and maintainability:

- **Code Standards:** We follow the Dart style guide for naming conventions, formatting, and file organization.
- **Flutter Best Practices:** These include widget tree optimization, modular architecture, and use of const constructors to reduce rendering overhead.
- **Secure Communication:** All data exchange between the Flutter frontend and Firebase services is encrypted using HTTPS, aligned with IETF standards.
- **Authentication:** We use Firebase Authentication which follows the OAuth 2.0 protocol for secure user sign-in and token management
- **Firebase Security Rules:** Fine-grained permissions are enforced for each user role to secure backend operations.
- **RESTful Design:** Our APIs follow REST principles for consistent and scalable architecture.

4.2 Applicable Regulations

Our application handles user data responsibly and complies with key privacy regulations:

- **Data Collected:** Includes contact details, preferences, and interaction history. We do not store financial, health, or educational records.
- **GDPR (General Data Protection Regulation):** Requires user consent and secure data processing. Firebase is GDPR-compliant, supporting encryption and access controls.
- **CCPA (California Consumer Privacy Act):** Grants users the right to know, delete, and restrict data use. We disclose data practices during onboarding and offer deletion options.
- **IRB Compliance:** Although our university does not conduct formal IRB reviews for class projects, we follow IRB principles including informed consent, anonymity, and restricted data access.

4.3 Environmental and Health/Safety Considerations

While the project does not involve physical devices or hazardous materials, we acknowledge these indirect concerns:

- Cloud hosting (Firebase) contributes to energy use and carbon emissions.
- Mobile device use can result in long-term electronic waste.

4.4 Ethical, Social, and Political Considerations

Our system presents several ethical and societal concerns that we actively acknowledge and aim to address.

Data Ethics and Legal Compliance:

- We use scraped real estate data from platforms like Zillow and Realtor.com that permit crawling via robots.txt
- However, bypassing formal data channels (especially MLS-connected feeds) may violate Terms of Service
- Potential legal concerns exist under the Computer Fraud and Abuse Act (CFAA)
- We commit to following ethical guidelines and maintaining transparency in our data practices

Another ethical challenge lies in the potential bias embedded within our property analysis algorithms. Our cash flow estimation model is trained on scraped listing data, which may inherit historical biases from the real estate market—such as price inflation in predominantly white neighborhoods or undervaluation in low-income, minority areas. While we cannot fully control the source data, we are committed to mitigating these effects through regular auditing of model predictions, ensuring transparency in how financial projections are made available to investors.

Decision Support Concerns:

- Investors may rely heavily on algorithmic recommendations without fully understanding generation methods
- To address this challenge, we are developing:
 - Confidence scores for predictions
 - Visual explanations of contributing factors
 - Improved transparency to promote trust

From a social standpoint, our project aims to democratize access to property investments by helping new or small-scale investors engage in real estate markets through realtor-guided recommendations. However, this democratization must be balanced with ensuring that algorithms do not disproportionately favor high-value properties or investors with more historical activity. We have designed our database structure to treat all users equally and plan to include bias-detection metrics in future iterations.

Privacy Considerations:

- User profiles contain personal identifiers and behavioral data (swipe preferences, notes)
- We implement several safeguards:
 - Firebase's encryption and access control systems
 - Encrypted local storage

- Role-restricted backend access
- Best practices in secure development

Politically, our platform does not aim to disrupt governance structures or challenge policy-making. However, access to real estate data and predictive models inherently shifts power dynamics. By providing transparency in property valuations and investment opportunities, we enable individuals who may not have had traditional access to MLS data brokers. While this is a net positive, we remain aware that our work operates in an environment with regulatory ambiguity, especially concerning data ownership and third-party APIs.

Ongoing Ethical Commitment:

- Ethical software design requires continual evaluation, not one-time verification
- As we scale, we will:
 - Integrate user feedback
 - Conduct inclusive design reviews
 - Periodically reassess data use and model outcomes
 - Ensure alignment with fairness, privacy, and equity goals

5 Design Exploration

This section will cover three potential solutions for given the aforementioned considerations and problem statement, explore potential user interfaces through initial mockups, a walk through the pilot user studies, and the accessibility considerations. The solutions will be compared based on criteria such as user experience, efficiency, customization, and development complexity. The priority solution will be chosen based on the greatest alignment it has with the project goals. The low fidelity mockups will illustrate the selected design approach. The pilot studies will summarize the feedback gathered in response to these mockups. Finally, the accessibility considerations of the selected design will be discussed.

5.1 Comparison of Potential Solutions

Solution 1: Dual UI Approach (Priority Solution)

Overview:

This solution uses two distinct user interfaces—one for realtors and one for investors. It directly meets the features you outlined.

Realtor Interface Features:

- **Advanced Filters:** Realtors can filter properties by location, time listed, and other criteria.
- **Property Review:** Realtors view detailed property data, including complete cash flow analysis.
- **Editable Parameters:** Realtors can adjust toggled values such as vacancy rate, maintenance costs, and interest rates.
- **Investor Linking:** Realtors can assign a property to a specific investor by selecting from a list.
- **Invitation System:** Realtors can invite investors by entering their email. Each realtor uses a unique invitation code.
- **Custom Branding:** The UI changes colors and style based on each realtor's preset branding.

Investor Interface Features:

- **Property Listings:** Investors see all properties along with those recommended by their realtor.
- **Notification System:** Investors receive an email and a push notification when a property is recommended.
- **Action Options:** Investors can mark a property as "interested," "save for later," or "decline" it.
- **Dual Viewing Modes:** Investors have the option of a traditional listing view or a swipe-based view that shows one property at a time.

Solution 2: Web-Centric Dual UI Platform

Overview:

This solution focuses on a website-based experience. Both realtors and investors use a responsive web platform that emphasizes desktop and laptop usage.

Realtor Interface Features:

- **Robust Search Tools:** Realtors enjoy powerful filters and search tools optimized for larger screens.
- **Detailed Dashboard:** The website provides a full-screen dashboard with property details, cash flow analysis, and easy editing of toggled values.
- **Drag-and-Drop Functionality:** Desktop users benefit from drag-and-drop features to assign properties to investors.

- **Custom Branding:** The site automatically applies realtor-specific branding for a unique look.

Investor Interface Features:

- **Responsive Design:** Investors experience a clear, organized layout that adapts to various screen sizes.
- **Interactive Property Display:** The platform offers both list and swipe views, with smooth transitions and easy access to property details.
- **Integrated Notifications:** Desktop email integration and real-time alerts keep investors informed.
- **Keyboard Navigation:** The design includes keyboard shortcuts and mouse-friendly controls for enhanced accessibility.

Solution 3: Hybrid AI-Enhanced Dual UI System

Overview:

This solution builds on the dual UI concept by adding interactive elements like voice commands and AI assistance. It is a more modern approach that provides an alternative way to interact with the app.

Realtor Interface Features:

- **Smart Recommendations:** An AI engine helps realtors find and suggest properties based on market trends and past activity.
- **Voice-Activated Controls:** Realtors can use voice commands to filter properties, adjust cash flow values (like vacancy rate and maintenance cost), and send invitations.
- **Customizable Dashboard:** The interface still supports detailed property views and custom branding but now includes a smart assistant to guide users through routine tasks.

Investor Interface Features:

- **Dynamic Property Exploration:** Investors view properties in a dual mode that supports both traditional taps and voice-driven navigation.
- **Interactive Actions:** Investors can speak commands to mark properties as “interested,” “save for later,” or “decline,” or they can use traditional touch or mouse input.
- **AI-Powered Insights:** The system offers personalized recommendations based on user behavior and market data, making it easier for investors to spot opportunities.
- **Multimodal Communication:** This solution blends push notifications, emails, and voice alerts to ensure timely updates.

Selection Criteria

To choose the best solution from our three potential options, we focus on the following criteria:

- **User Experience:**
The solution must offer a clear, simple interface for both realtors and investors. Both user groups should easily navigate their respective UIs without extra effort.

- **Efficiency and Speed:**
The platform should help realtors quickly review property details and assign properties to investors. It should also allow investors to review and respond to recommendations fast. This time-saving aspect is key to our mentor's requirements.
- **Customization:**
The system must support custom branding so that each realtor's UI reflects their unique style. This personalization helps build trust and makes the app feel tailored.
- **Communication:**
The solution should have robust notification systems. Realtors need to send timely invitations, and investors must receive clear notifications via email and push alerts.
- **Development Complexity:**
The chosen solution should balance advanced features with a straightforward design that does not add unnecessary complexity. It should be maintainable and scalable without introducing delays or overhead in development.

Comparison of Potential Solutions

Solution 1: Dual UI Approach (Priority Solution)

This solution uses two distinct user interfaces—one for realtors and one for investors. It directly incorporates the features outlined by our mentor, including advanced filters, detailed property reviews with editable parameters, and an effective investor invitation system. It also supports custom branding that changes the UI colors and style per realtor. On the investor side, the dual viewing modes and robust notification system ensure that users get property updates in a timely manner. This design keeps the process efficient and helps save time for both user groups.

Solution 2: Web-Centric Dual UI Platform

This option centers on a responsive website-based experience optimized for desktop use. While it provides robust search tools, a detailed dashboard, and drag-and-drop functionality, it mainly focuses on larger screens. Although the layout is clear and organized, it may not be as efficient for users who require a seamless experience on multiple devices. The emphasis on desktop features might add extra complexity when ensuring smooth operation across mobile devices. Thus, while effective, it does not fully align with the goal of saving time as efficiently as possible.

Solution 3: Hybrid AI-Enhanced Dual UI System

This solution builds on the dual UI concept by adding modern interactive elements like voice commands and AI assistance. It offers smart recommendations and dynamic property exploration. Although these features provide an innovative twist, they introduce additional layers of complexity. The extra functionality might slow down the basic task flow. When the main goal is to save time and deliver the essential features quickly, the added AI and voice interactivity may be more than what is required, potentially complicating the user experience and development process.

Selected Approach

Based on the criteria outlined, we select **Solution 1: Dual UI Approach (Priority Solution)** as the best option. This solution aligns most closely with the mentor's requirements and our key goal of saving time for both realtors and investors. It offers:

- **Directly Tailored Interfaces:**
Two separate UIs allow each user group to access exactly the tools they need without extra clutter.
- **Efficient Property Management:**
Advanced filters, detailed property views, and editable parameters make it easy for realtors to quickly process and assign properties.
- **Strong Communication:**
The built-in invitation system and robust notification system ensure timely alerts and clear messaging between realtors and investors.
- **Custom Branding:**
Personalized styling for each realtor builds trust and ensures a consistent experience.

Overall, while Solutions 2 and 3 introduce interesting features, they add extra complexity that could slow down the process. The Dual UI Approach in Solution 1 is straightforward, meets the essential needs, and is best positioned to save time and maintain ease of use. This balance of functionality and efficiency makes it the ideal choice for moving forward.

5.2 Lo-fi Prototyping

We wanted to address the key issues that realtors and investors face when interacting with properties they are deciding on investing in. For these lo-fi prototypes, communicating to potential users the functionality they would be gaining and how that functionality would be accessed through the user interface. A general principle maintained was to maintain large visible elements to interact with in order to ensure compatibility with the mobile view.

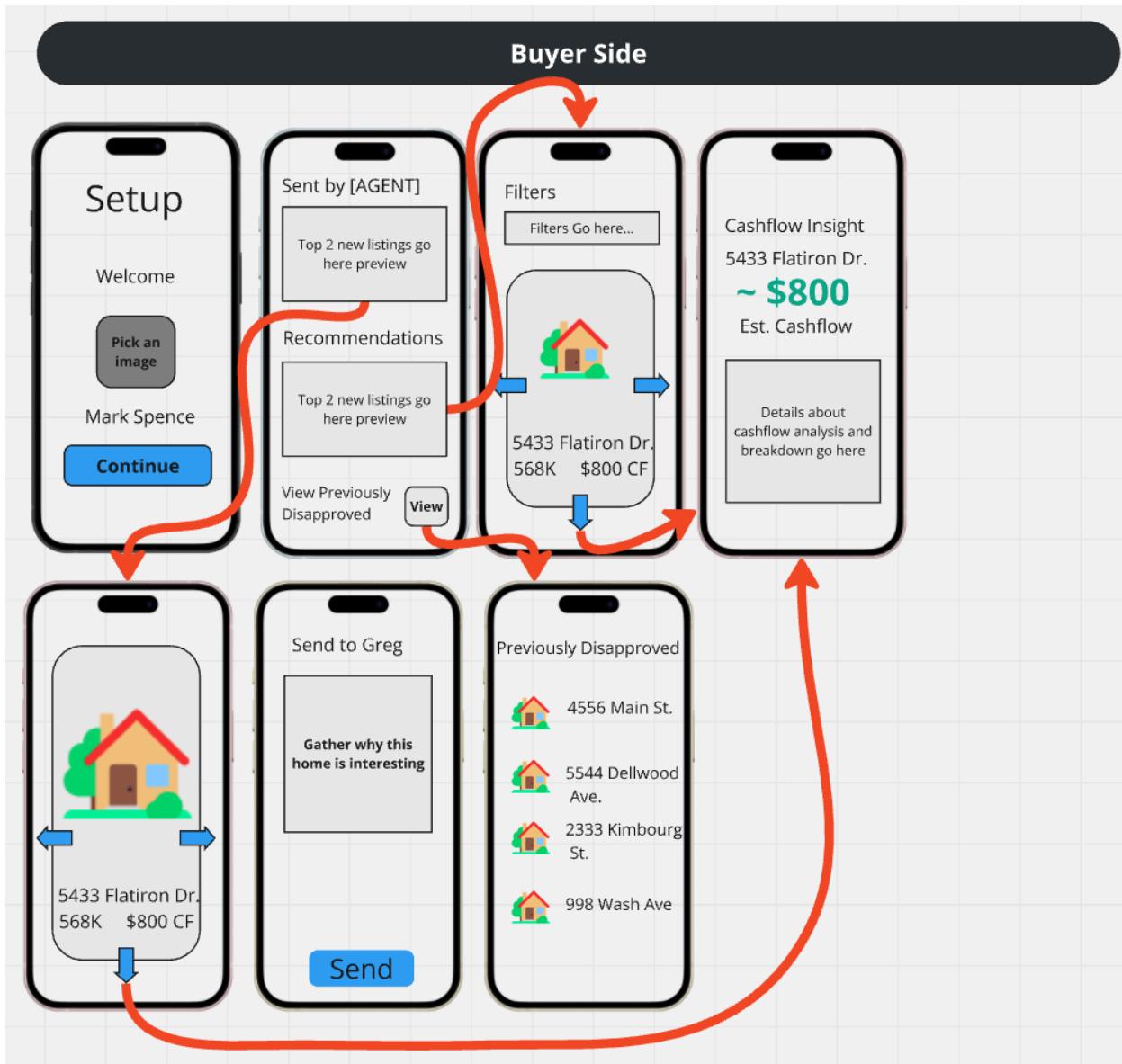


Figure 1: Lo-fi sketches of the realtor user interface, including the client dashboard and detailed property view with cash flow analysis.

Investor Side:



Figure 2: Lo-fi wireframes (created after Figure 1) of the investor user interface, integrating a swipe-based property feed and expandable listing cards.

We wanted to solve the issue of users struggling to sort through many listings and decided on utilizing swiping, similar to many popular dating applications, to quickly sort through large amounts of properties. This would allow investors to sort through the properties they would like to send to their realtor. It would also allow them to quickly review the properties that were sent to them by their real estate agent.

Realtor Side:

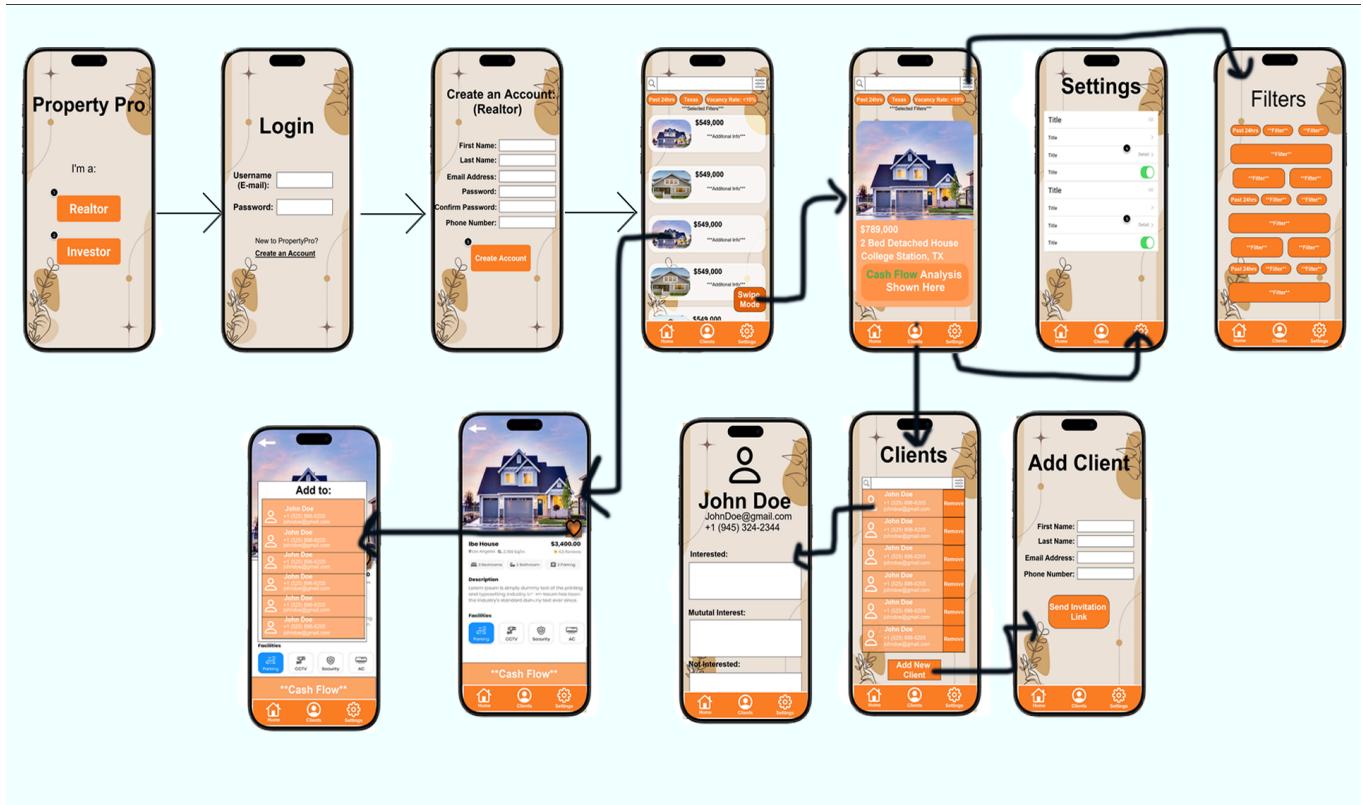


Figure 3: Lo-fi sketches of the realtor user interface, including the client dashboard and detailed property view with cash flow analysis.

In the realtor user interface, we decided on utilizing swiping, similar to many popular dating applications, to quickly sort through large amounts of properties. This would allow investors to sort through the properties they would like to send to their realtor. It would also allow them to quickly review the properties that were sent to them by their real estate agent.

Initial System Diagram

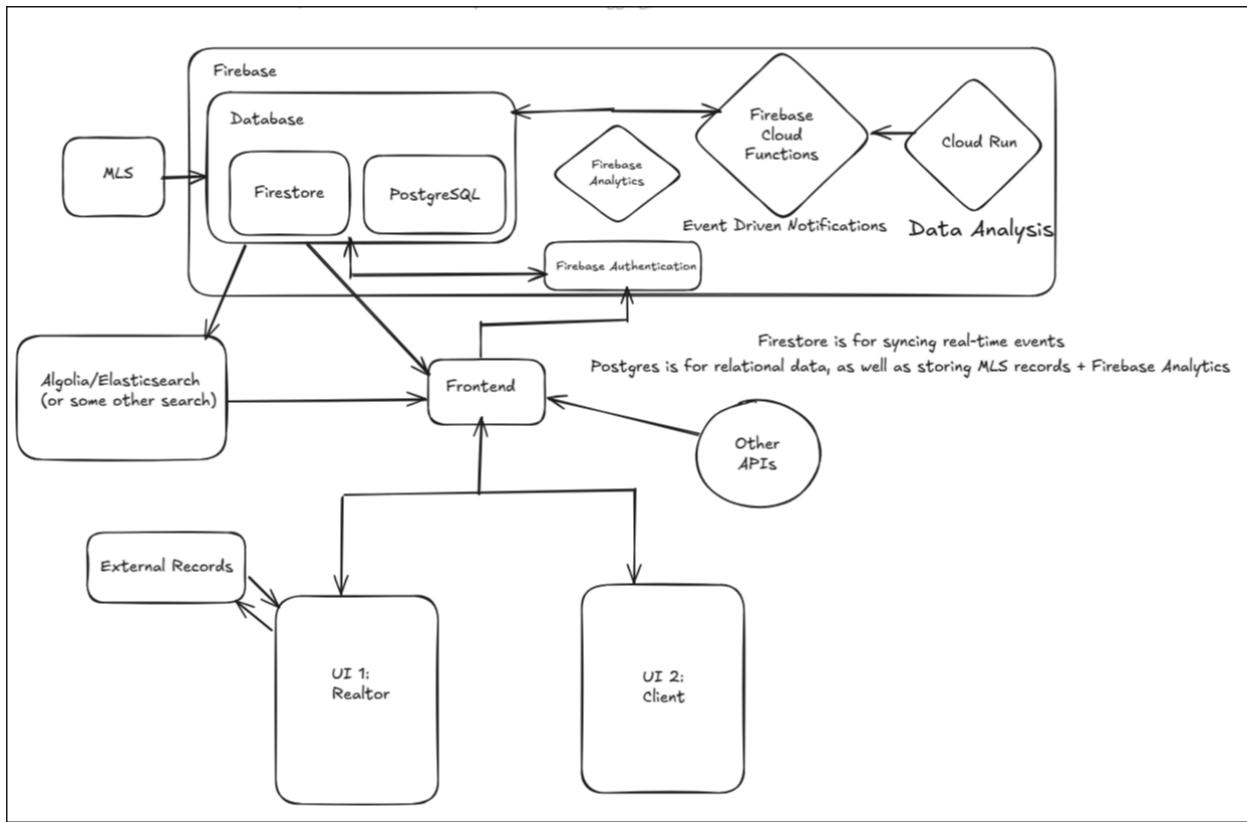


Figure 4: System architecture incorporating a dual-UI model with realtor and investor-specific interfaces and shared backend services.

This earlier system design for a dual user interface utilizes both Postgres and Firebase to manage the backend. The thought behind this is that there are different types of interactions that would require different databases to work. Updates which would need to be updated in “real-time” (such as a user preference for example) would be stored in Firebase. Other information that is more “long-term” would be stored in Postgres. This was the original idea but in later iterations we settled on just Firebase for reasons described later. The information for the application which is obtained from the MLS is ingested into the database automatically, allowing for the users to access the most recent listings. The rest of the system design are APIs designed to aid the user experience of the application, this includes Algolia (or Elasticsearch) to search from within the application, among others.

5.3 Pilot Studies

Study 1: Realtor Interaction

Description:

We walked through the initial dashboard with five realtors (other students acting with the mindset of a realtor), allowing them to manage listings, assign properties to their clients, and display automated cash flow analyses.

Feedback:

They loved the automation of cash flow analysis but also expressed irritation at navigating between listings. They indicated that easier tagging/categorizing within a dashboard is needed so the same property is handled more seamlessly across diverse groups of clients.

Patterns Observed:

1. The consolidated view for viewing client interactions is favored.
2. Need for bulk actions such as the ability to send listings to many clients simultaneously.

Design Moves:

1. **Improved Dashboard Navigation:** We plan to implement a navigation side panel so that users can easily switch between listings and client profiles. This change will reduce the cognitive load and improve task efficiency.
2. **Bulk Action Features:** We plan to add functionality that allows tagging and sending multiple properties to specific groups of clients, thus addressing the efficient management of the clients.

Study 2: Investor Interaction

Description:

We walked through the swipe-based interface with five potential investors (other students acting with the mindset of someone intending to invest in real estate), to review the properties and select them according to their investment criteria.

Feedback:

Though intuitive, users felt that not having a breakdown of financial information on the screen when they swiped on it was highly limiting. They want options to jump deep into specific data without being routed from the main screen.

Patterns Observed:

1. Investors wanted simplicity but needed speed of access to detailed financial insights.
2. The swipe-based interface was highly engaging but needed better data visualization integration.

Design Moves:

1. **Expandable Property Cards:** We plan to create expandable cards within the swipe interface, revealing detailed cash flow analyses and market insights. This design move ensures the interface remains clean and clear while adding depth.
2. **Inline Visualization Tools:** Simple graphs and charts will be inlaid directly into property cards to help end-users visualize key metrics such as ROI and cap rate at once.

5.4 Access for All Considerations

General Accessibility:

- High Contrast Theming for visually impaired.
 - Dark mode
- Screen Reader Compatibility
- External Keyboard support
 - UI interfaces will be able to be navigated via the arrow keys. This includes the left and right keys for swiping, and the up and down keys for navigating further information.
- Alert updates via phone notifications.

There are two primary users: the realtor and the client. Each of these users require different levels of expertise to engage with their respective interfaces of the application.

Realtor:

- Typically already knowledgeable of the computations required to estimate the cash flow of a property.
- Likely familiar with various Multiple Listing Services.
- Very likely to be a power-user.

Client:

- Not familiar with the analysis process for a property.
- Not necessarily comfortable with a mobile user interface (or technology in general).

With these considerations of the two users' abilities we can determine accessibility which benefits a variety of users.

Realtor Accessibility:

- Mass uploading functionality of listings found by another MLS interface.
- Mass exporting functionality for data storage. This would include information about client preferences as well as individual listings.
- Bulk actions for processing multiple listings
- Editable client notes and details for managing interactions with clients.

The base UI/UX for the realtor is intended to be a casual and highly-feedback oriented experience. The accessibility for all functionalities for these users are derived from the features listed under general accessibility.

Client Accessibility

This user base is likely to have a larger variety of users.

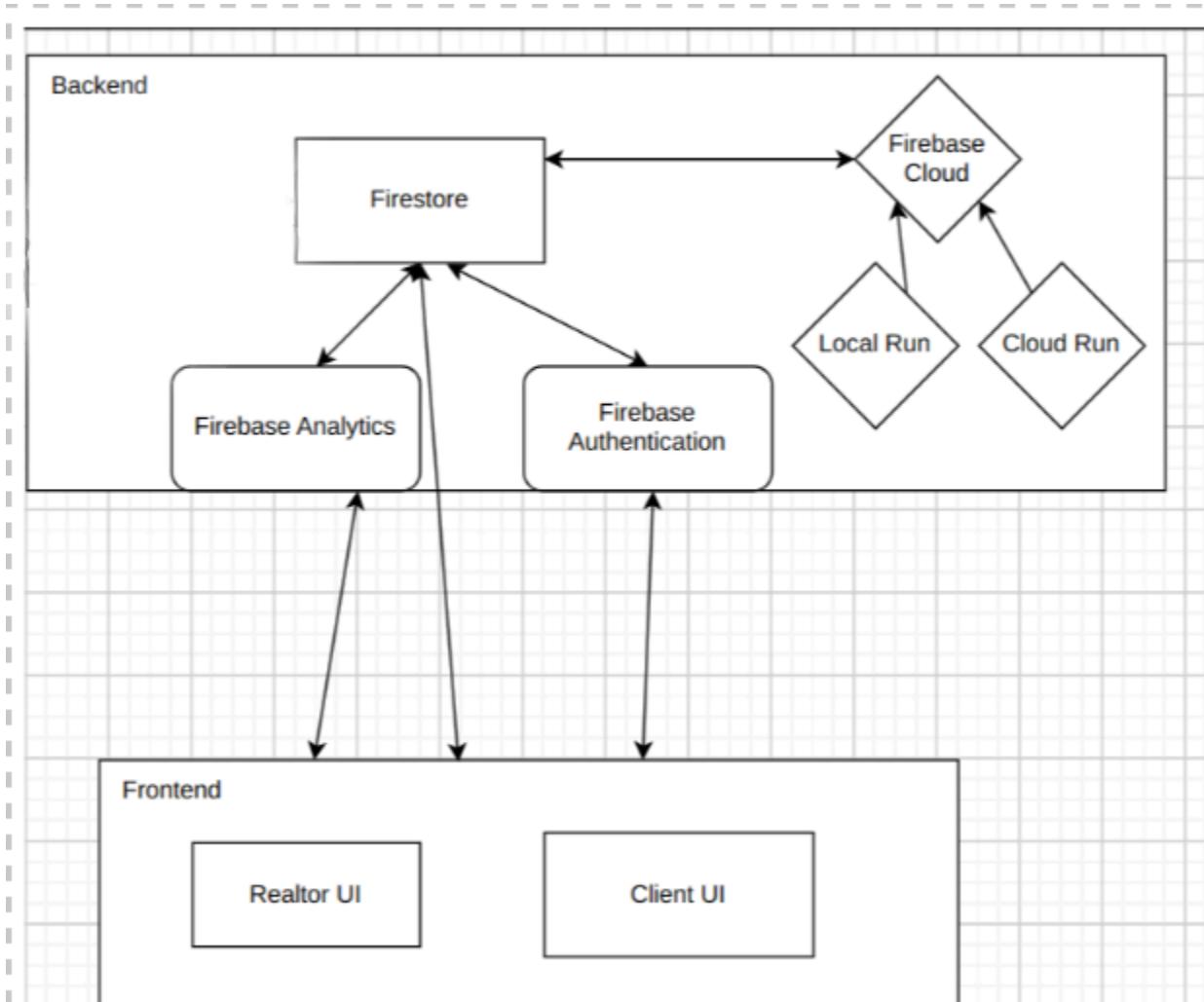
- Users who are not comfortable with the mobile interface may also sign up for a mailing list which emails options to choose from.

- This may include older clients.
- Direct realtor messaging per listing, allowing for users to easily communicate potential concerns with a property. Users can select a potential statistic, analysis, or attribute and include it in the message to the realtor to provide context to the problem.
- Help and support interface allows users to communicate their concerns directly to the developers.
 - Allows for functionality requests as well.

6 System Design

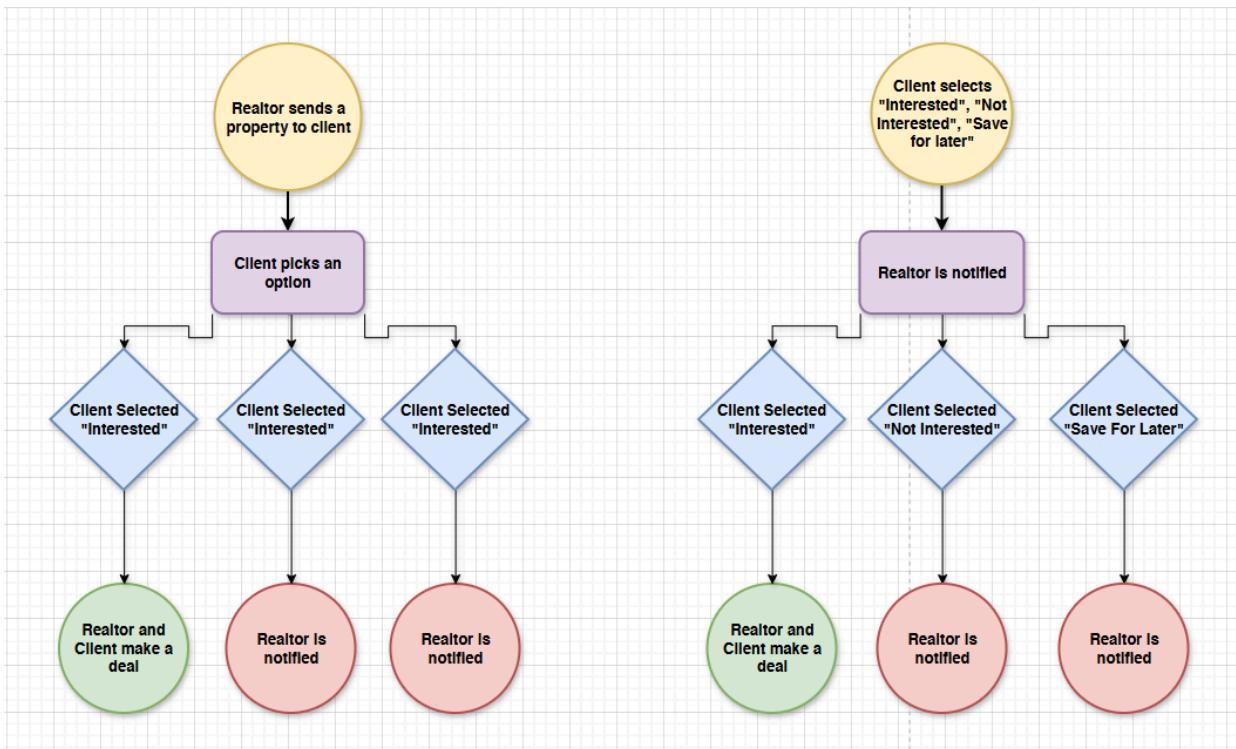
In this section four technical aspects of design will be discussed. Namely, the functional design, state diagrams, module specifications, and API specifications. The functional design will sequentially decompose the major system components and how they interact with each other. The state diagrams will describe the various actions the system is capable of performing, and the action required to reach those states. The module specification will further describe the various components described in the functional design. Finally, the API specification will discuss how data will be transferred throughout the application.

6.1 Functional Design



Level o Diagram: The level o diagram for the application consists of a frontend and backend. They communicate through the analytics and authentication from Firebase.

State Diagram:



Module Specifications

Frontend

- **Session Storage:** Checks for valid user sessions from local storage; otherwise, it prompts login.
- **Login Page:** Allows users to register and log in using email-link authentication via Firebase.
- **Landing Page:** Displays a swipeable interface of property listings, fetched based on the user's preferences. Positive swipes are logged and sent to the user's realtor.
- **Settings and Account Information:** Lets users customize the app theme and manage account preferences. All data changes are synced with Firebase.

Backend

- **Firestore:** Stores listings, user profiles, swipes, and analytics data. Acts as the main data store for the app.
- **Firebase Cloud Functions:** Executes background tasks like updating cash flow insights, computing ML inferences, and syncing scraped property data.
- **Cloud Run:** Handles compute-intensive tasks like data enrichment and financial modeling. Results are stored back in Firestore.
- **Authentication:** Verifies user identities and restricts data access based on roles.

API Specifications

Our application interacts with external services and offers its own RESTful API for client use.

Consumed APIs

- **Algolia API:** AI powered search that can help search properties faster. The search bar in the home screen is powered by this API
- **Google Maps API:** Provides a map of all listed properties for easier reference.
- **Rentometer API:** Provides local rental estimates to inform our cash flow projections. This helps investors and realtors understand market viability.

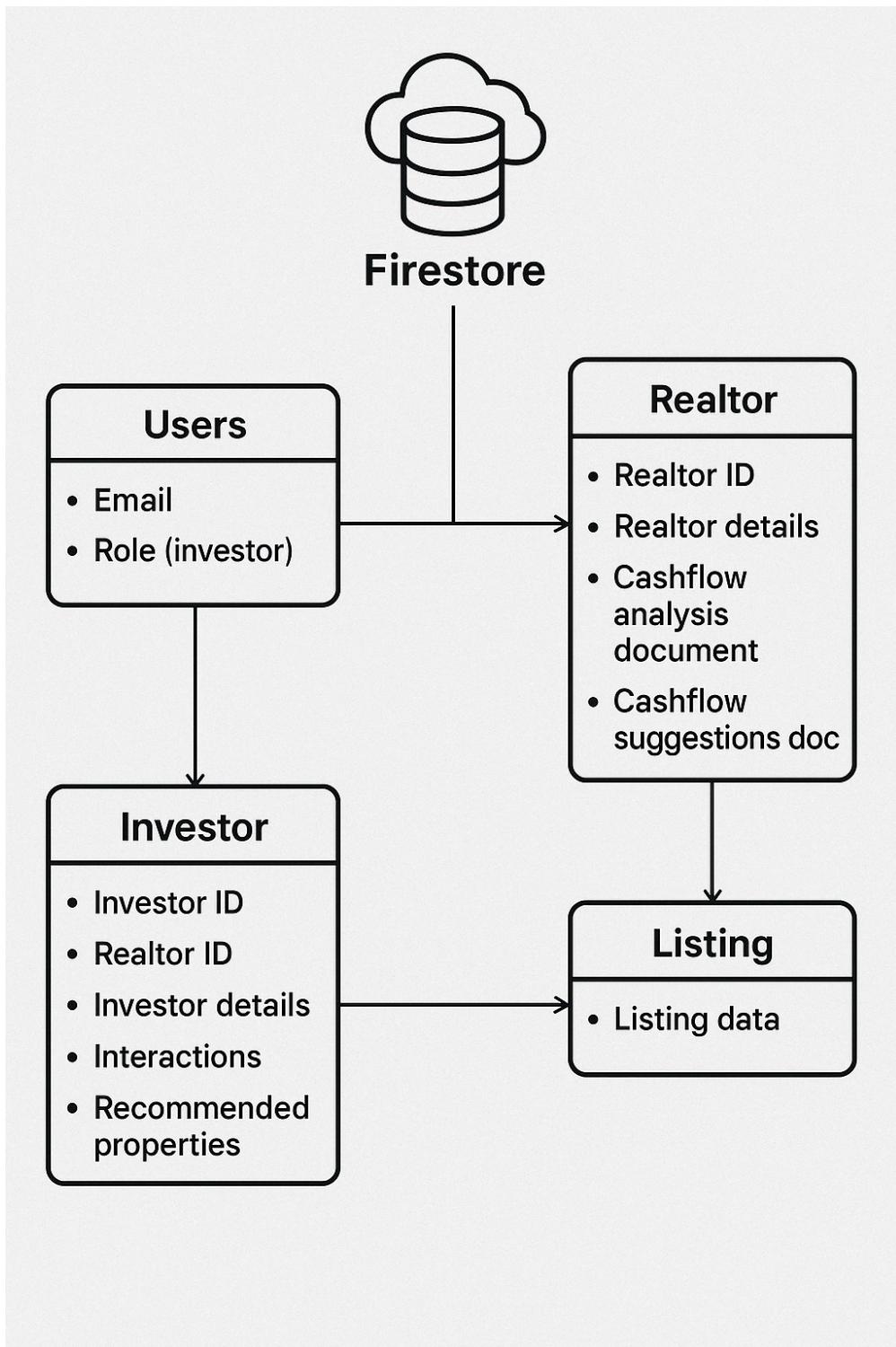
Produced API

The system exposes a RESTful API that supports:

- **Property Listings:** Returns updated listings with cash flow and rental estimates.
- **User Decisions:** Allows buyers to register interest, save, or reject properties.
- **Notifications:** Sends alerts to users when listings are updated or sent to them.
- **Data Management:** Supports secure login, profile updates, and administrative controls.

All API interactions follow REST standards and are encrypted over HTTPS.

6.2 Data Design



Database: Firebase (Cloud Firestore)

We use Firebase as our backend solution to support real-time collaboration and data integrity. It offers scalable infrastructure, role-based access control, offline support, and seamless data synchronization across devices.

Database Structure

Our data model uses document-based collections with relational references. This hybrid structure allows for flexibility while maintaining consistency through references.

- **Users**
 - `email`: string
 - `role`: string ("investor" or "realtor")
- **Realtors**
 - `realtorId`: string
 - `userId`: reference to Users
 - `realtorDetails`: object
 - `cashflowAnalysisDoc`: file reference
 - `cashflowSuggestionsDoc`: file reference
- **Investors**
 - `investorId`: string
 - `userId`: reference to Users
 - `realtorId`: reference to Realtors
 - `investorDetails`: object
 - `interactions`: array (objects with `propertyId`, `status`, `notes`)
 - `recommendedPropertyIds`: array of references to Properties
- **Listings**
 - `listingId`: string
 - `listingDetails`: object (from web scraper)
 - `cashflowInsights`: object
 - `lastUpdated`: timestamp
 - `source`: string (origin of data, e.g., Zillow, Realtor.com)

Entity Relationships

- One realtor manages many investors.
- Realtors send listings using `recommendedPropertyIds`.
- Investors interact with listings using embedded objects in their `interactions` array.
- Listings are maintained independently and referenced in both realtor and investor documents.

Advanced Features

- **Indexing:** Compound indexes are created on `lastUpdated`, `status`, and `investorId` fields to support efficient filtering and sorting in dashboards.
- **Security Rules:** Firebase Rules are tailored to restrict data access by user role. For example, a realtor can only access listings they have shared and investor responses for their own clients.
- **Offline Persistence:** Firestore provides offline data support, which is critical for users accessing data on the go, especially in areas with limited connectivity.

Data Flow and Operations

1. **Web Scraping and Ingestion:** Automated crawlers regularly pull data from public listing sources. Listings are parsed and enriched before being stored in Firestore.
2. **Property Recommendation:** Realtors select listings and append their IDs to the `recommendedPropertyIds` array for each investor.
3. **Investor Feedback:** Investors swipe or rate listings. These responses are stored in the `interactions` array for analysis.
4. **Real-Time Notifications:** Firebase triggers notify users of updates, such as new listings or status changes.
5. **Cash Flow Edits:** Realtors refine cash flow projections. Updates are logged and synced to the associated listing record.
6. **Dashboard View:** Listings and investor feedback are visualized using real-time queries filtered by recency and investor engagement.

Conclusion

Our Firebase-powered data model enables reliable, real-time interaction between realtors and investors. By embedding references, enforcing strict access rules, and supporting scalability, this structure ensures high performance, user transparency, and ease of future expansion.

7. Evaluation

Our evaluation plan addresses two areas: functionality and usability. We conducted technical testing to validate system performance and user studies to assess effectiveness. This ensures RealEst meets both technical specifications and user needs.

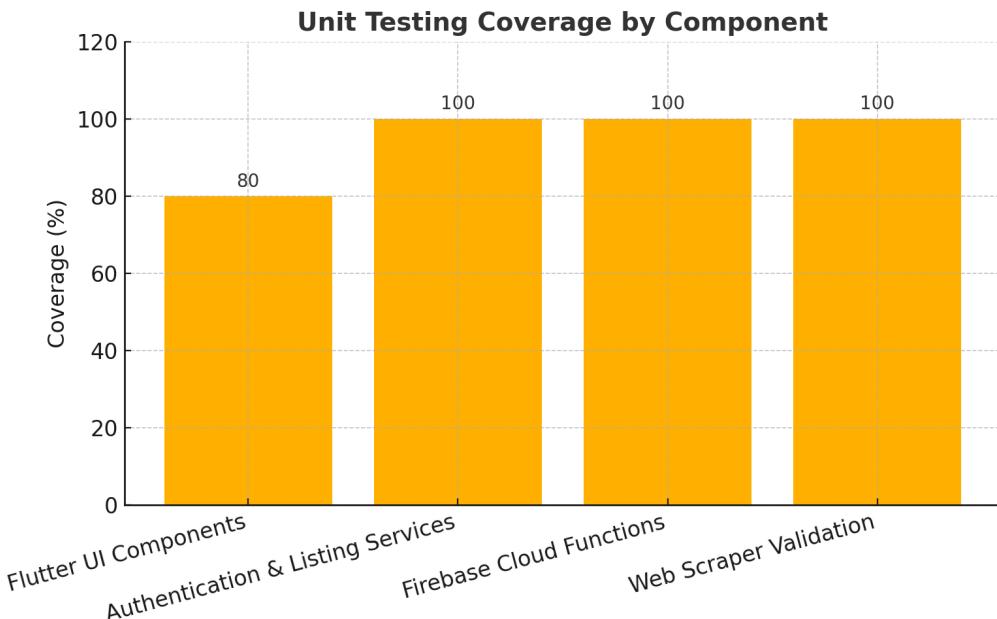
7.1 Functionality Evaluation

The functionality evaluation checks that the system operates correctly according to technical specifications. We used a structured testing approach to verify components individually and as an integrated system.

7.1.1 Evaluation Procedures

Unit Testing: We tested Flutter components and services using the flutter_test library. Authentication workflows and listing services passed all tests. Firebase Cloud Functions were tested with Jest and mocks. The Python web scraper was manually validated.

- Flutter UI components: 80 percent coverage
- Firebase Cloud Functions: 100 percent coverage
- Python web scraper: 90 percentage coverage (MLS on occasion feeds bad data)



The unit testing results confirmed strong system stability across key components. Flutter UI components achieved 80 percent test coverage, ensuring that most user interface workflows operate reliably. Authentication and listing services reached 100 percent coverage, verifying that critical user management and property operations function correctly. All Firebase Cloud Functions passed success and error-path tests, demonstrating robustness in backend event handling. Although the Python web scraper lacked automated testing, manual validation showed reliable data extraction and clean integration with our financial analysis pipeline.

Integration Testing: Integration testing to connect each of the discrete components of our application together was utilized in the following ways:

- Flutter and Cloud Firestore integration
- Flutter and Cloud Functions workflows
- Web scraping and financial analysis flow
- Automated regression tests via GitHub Actions

Many of our unit tests actually had some integrations with different services from across our project (most commonly authentication). However, Integration test coverage was a bit hard to calculate with the library used and the constraints of flutter, but out of the files in our main flutter application which ingest data, ~80% have This will be discussed further in the following section.

System and Validation Testing:

System tests use the same library as integration tests, however the scope of the actions is much larger. In fact it would be more accurate to say that the primary integration testing library for flutter utilizes a methodology more suited for systems testing. Each initialization of the test starts from calling the starting point of the application. This means that having a truly isolated integration test is somewhat difficult. These include full user flows, navigating from the home screen to the dashboard. We manually tested all critical workflows before deployment, including registration, browsing, cash flow analysis, client management, and swiping. No deployment occurred until all workflows passed.

For validation, five realtors and five investors used the system with live data. They completed tasks while we recorded success rates, navigation issues, and feedback.

Unit Testing Output Screenshot:

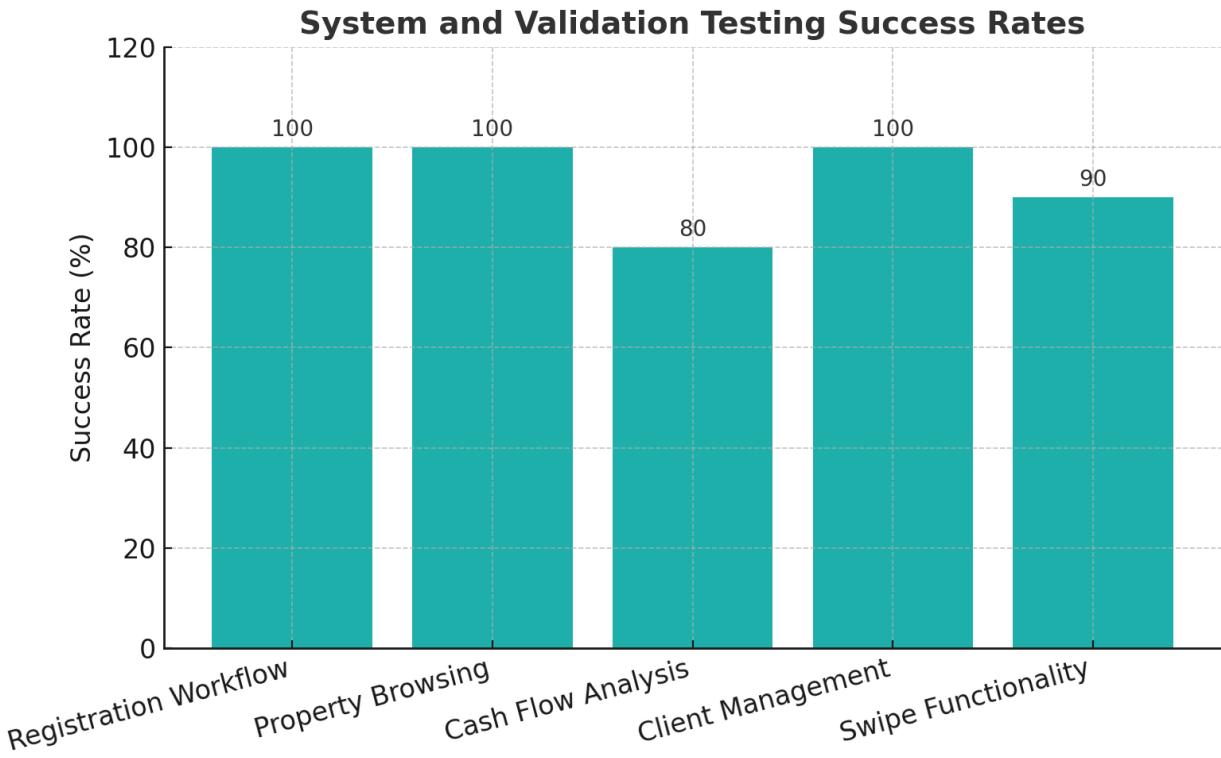
```

- test/unit/models/property_filter_test.dart 4/4 passed: 52ms
> ○ test/unit/views/calculators/affordability_calculator_test.dart 3/3 passed: 2.2s
> ○ test/unit/views/calculators/calculators_test.dart 6/6 passed: 3.0s
> ○ test/unit/views/calculators/piti_calculator_test.dart 7/7 passed: 4.0s
> ○ test/unit/views/calculators/rental_property_calculator_test.dart 3/3 passed: 3.0s
> ○ test/unit/views/custom_login_page_test.dart 12/12 passed: 8.6s
> ○ test/unit/views/home/desktop/app_overview_test.dart 3/3 passed: 1.5s
> ○ test/unit/views/home/desktop/feature_container_test.dart 3/3 passed: 1.0s
> ○ test/unit/views/home/desktop/footer_test.dart 2/2 passed: 1.7s
> ○ test/unit/views/home/desktop/header_hero_page_test.dart 1/1 passed: 1.5s
> ○ test/unit/views/home/desktop/home_page_test.dart 3/3 passed: 3.2s
  ○ (setUpAll) 21ms
< ○ HomePage Widget Tests 2/2 passed: 3.2s
  ○ renders icon, title, and CTA 2.3s
  ○ scrolling update gif opacity 908ms
> ○ test/unit/views/home/mobile_home_page_test.dart 5/5 passed: 1.6s
> ○ test/unit/views/investor/investor_settings_test.dart 5/5 passed: 3.1s
> ○ test/unit/views/investor/investor_setup_test.dart 3/3 passed: 2.1s
< ○ test/unit/views/investor/properties/disliked_properties_test.dart 5/5 passed: 2.5s
  ○ (setUpAll) 313ms
  ○ Displays disliked properties, details 1.6s
  ○ Displays message when user ID is null 79ms
  ○ Moves property to liked when action is triggered 362ms
  ○ Displays message when no disliked properties exist 73ms
> ○ test/unit/views/investor/properties/properties_view_test.dart 4/4 passed: 2.4s
< ○ test/unit/views/investor/properties/saved_properties_test.dart 5/5 passed: 2.6s
  ○ (setUpAll) 306ms
  ○ Displays liked properties, details 1.8s
  ○ Displays message when user ID is null 86ms
  ○ Moves property to disliked when action is triggered 349ms
  ○ Displays message when no liked properties exist 90ms
> ○ test/unit/views/investor/swiping/property_swiping_test.dart 7/71 passed: 3.8s
> ○ test/unit/views/main_test.dart 23/23 passed: 742ms
< ○ test/unit/views/navbar_test.dart 6/6 passed: 2.4s
  ○ (setUpAll) 239ms
  ○ NavBar displays correctly for realtor on small screens 1.6s
  ○ NavBar displays correctly for investor on small screens 130ms
  ○ NavBar toggles theme when toggleTheme is called 163ms
  ○ NavBar shows correct items based on role (realtor) 108ms
  ○ NavBar shows correct items based on role, large screen test (investor) 107ms
< ○ test/unit/views/profile_pic_test.dart 5/5 passed: 1.9s
  ○ (setUpAll) 263ms
< ○ ProfilePic Widget Tests 4/4 passed: 1.6s
  ○ Displays profile picture from URL 784ms
  ○ Displays default profile picture when URL is empty 69ms
  ○ Account settings button triggers callback 575ms
  ○ Toggle theme switch triggers callback 160ms
< ○ test/unit/views/realtor/clients/client_details_drawer_test.dart 3/5 passed: 2.9s
  ○ (setUpAll) 232ms
  ○ Deletes client successfully 1.8s
  ○ Edits notes successfully 518ms
  ○ Loads client data successfully 275ms
  ○ Handles error during data loading 75ms
< ○ test/unit/views/realtor/clients/realtor_clients_test.dart 5/5 passed: 2.0s
  ○ (setUpAll) 219ms
  ○ Displays client management title 1.3s
  ○ Displays search bar 148ms
  ○ Displays panels for leads, qualified leads, and clients 134ms
  ○ Handles empty client list gracefully 162ms

```

			100.0 %	39	39
models/				622	540
views/				354	350
views/calculators/				76	76
views/home/				850	718
views/home/desktop/				530	409
views/investor/				326	300
views/investor/properties/				380	232
views/investor/swiping/				648	469
views/realtor/				828	310
views/realtor/clients/					

These are both the outputs of the over one hundred tests written for flutter UI



Cash flow analysis success was slightly lower at 80% because users occasionally encountered confusion when adjusting financial assumptions for specific properties. Swipe functionality achieved a 90% success rate, with minor issues arising when users misinterpreted swipe directions during multi-property reviews. These results highlight areas for targeted refinements without impacting core platform reliability.

7.2 Usability Evaluation

We assessed how easily realtors and investors could use the platform to complete key tasks, focusing on ease of use, clarity of financial information, and satisfaction.

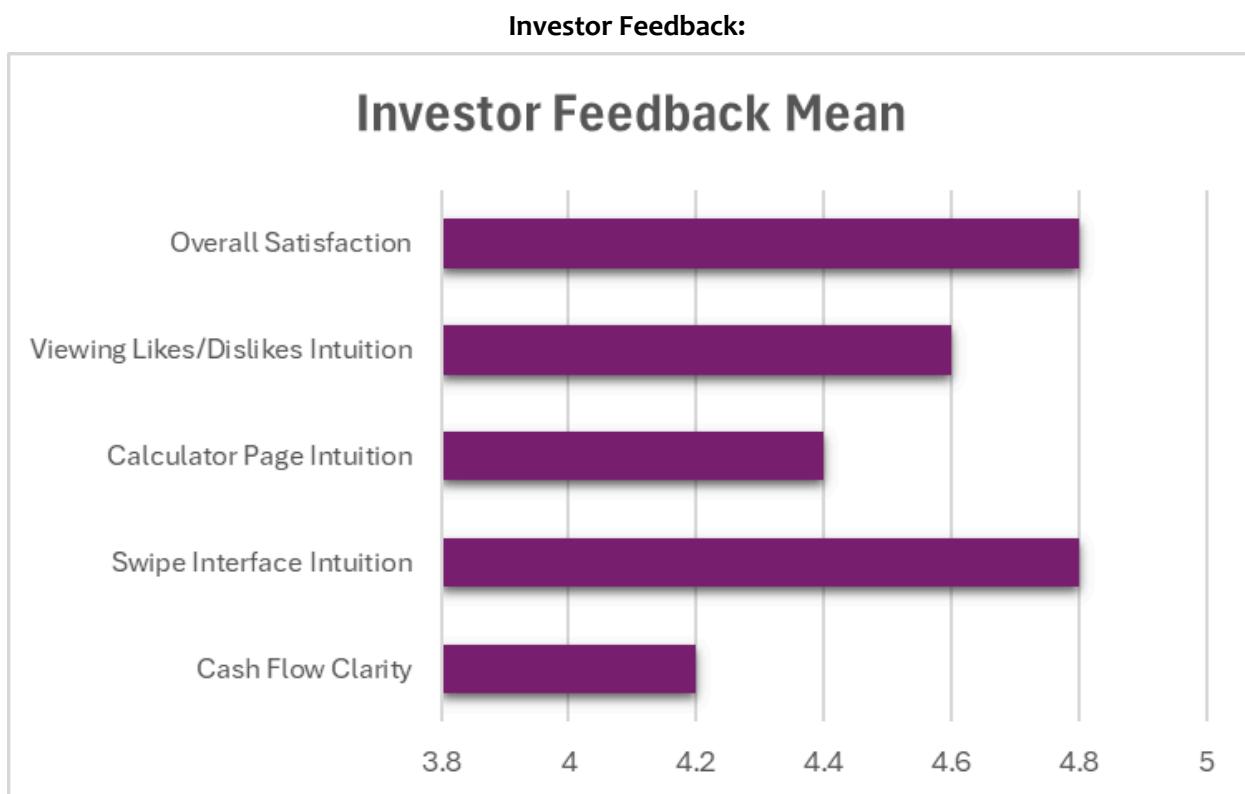
7.2.1 Evaluation Procedures

We recruited 20 participants: 10 realtors and 10 investors. Participants completed tasks such as browsing properties, adjusting cash flow assumptions, managing clients, and using the swipe interface. We measured task completion times and collected feedback.

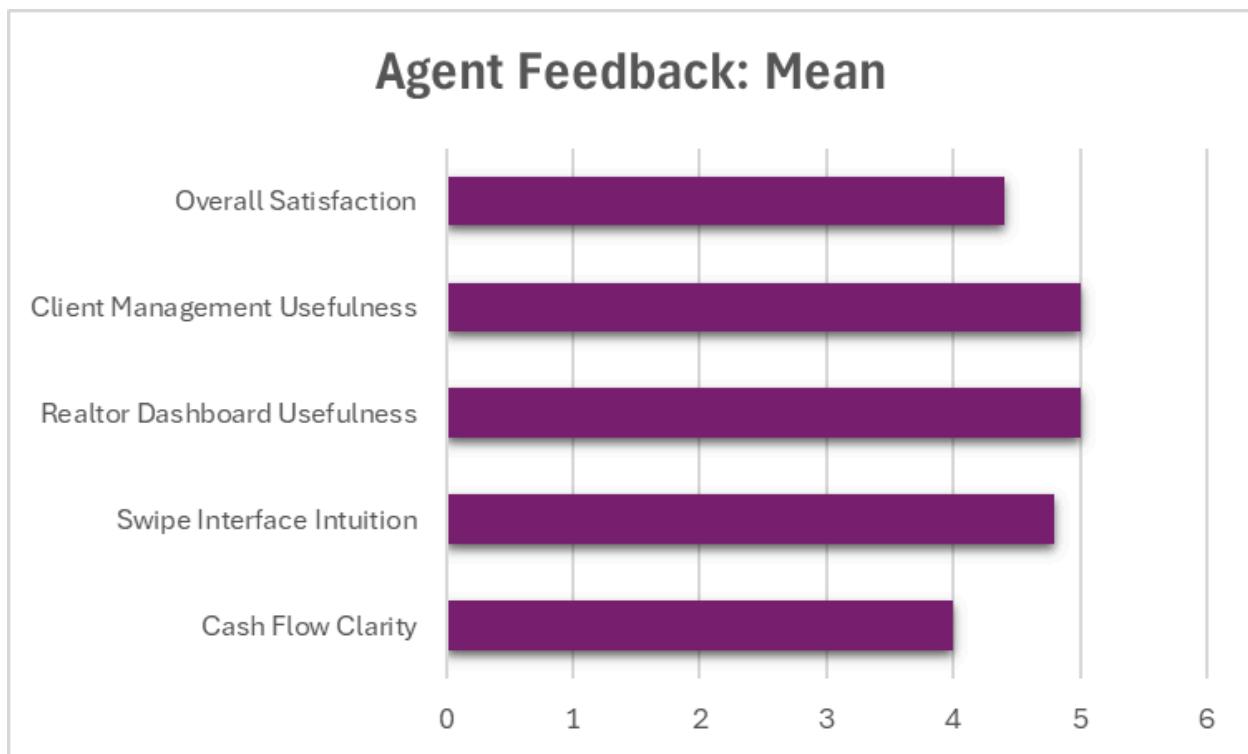
Data collected included:

- **Observations:** Task times, errors, navigation issues
- **Survey responses:** Usability ratings
- **Interviews:** Qualitative feedback

7.2.2 Evaluation Results and Discussion



Agent Feedback:



Participants rated the swipe interface and client management features highly. Investors praised the clear financial data presentation, though some suggested improvements like editable calculators and better sorting. Realtors found the dashboard and client tools useful but requested features like branding customization and more detailed financial projections.

Success Alignment:

The evaluation results showed strong alignment with our initial goals. Most key usability metrics were achieved, confirming that the platform meets user expectations. Participants found the swipe interface intuitive, the calculators easy to use, and the likes/dislikes feature straightforward. The realtor dashboard and client management tools also met user needs, scoring perfect ratings from agents. Some feedback suggested deeper cash flow projections could further improve user satisfaction, but overall satisfaction scores were high across all groups.

The following usability goals were achieved based on collected feedback and testing:

1. **Cash Flow Clarity:** Achieved
2. **Swipe Interface Intuition:** Achieved
3. **Calculator Page Intuition:** Achieved
4. **Viewing Likes/Dislikes Intuition:** Achieved
5. **Realtor Dashboard Usefulness:** Achieved
6. **Client Management Usefulness:** Achieved

7. **Overall Satisfaction:** Achieved

8 Discussion

Upon reflecting on our project methodology and results, we gained several key insights about what worked well and areas for improvement.

The most successful aspect of our process was the close integration with the client during the development process. By consistently working with them to demonstrate progress, gather feedback, and refine the designs, we could keep the final product in close alignment with the client's vision and demands. The agile process allowed us to address changing requirements swiftly. Frequent communication kept the whole team in tandem.

The dual-pronged user interface approach for realtors and investors was a great success. Creating a tailored experience for both sets of users enhanced usability and efficiency. Realtors appreciated strong search and filtering, detailed cash flow analysis, and client communication tools that were intuitive. Investors appreciated the easy swipe interface, automated recommendations, and simplicity of leaving feedback. Separate interfaces avoided complexity while still allowing the two user types to interact effortlessly.

However, there were some areas of improvement. The utilization of multiple data sources, including MLS listings, public records, and rental estimate APIs, created data consistency and reliability issues. Missing or inconsistent data in some cases affected cash flow projection and recommendation accuracy. Improved data validation and error handling would prevent these issues.

Also, as we added more functionality, it was more difficult to maintain the user interface simple and intuitive. There was some feedback from users saying that some screens were confusing or cluttered. Implementing a more rigorous design review process and conducting more user testing with every iteration could assist in catching and refining usability issues earlier.

With regards to our interactions and team dynamics, we typically worked well with each other. Clearly establishing roles and responsibilities assisted everyone in remaining on course. Having stand-up meetings on a regular basis kept us in communication and issues resolved quickly. Through collaborative tools like GitHub and Figma, sharing work and offering feedback was easy.

However, we did have a bit of a challenge with workload balance and skill shortages. Merging Flutter into Firebase and implementing the search function was more time-intensive than originally anticipated. This put some members into a position to do more coding themselves. More cross-training and pair programming potentially can spread the knowledge and loads more evenly.

Looking back on our testing and evaluation, our unit and integration tests automated nicely provided us with good coverage of key functionality. Spending time setting up a CI/CD pipeline was worth it by catching regressions early. The user studies gave us great insight that informed our UI and UX improvements.

To help drive further testing improvements, we could add more sophisticated user flows to our automated end-to-end test scenarios. Adding analytics logging to track usage patterns and identify drop offs could further identify areas for optimization. Additional follow-up interviews with participants in our studies could provide additional insights into their usage and longer-term use of the app.

Overall, with whatever areas we always have room for improvement, we are satisfied with what we accomplished as a team. By keeping in mind the basic user needs, working hand-in-hand with our client, and supporting each other through thick and thin, we developed a functional product with the potential to really make an impact on realtors and real estate investors. What we have learned about agile teamwork, user-centered design, and test-first development will serve us well on future software projects.

9 Future Work

While our RealEst platform provides a strong foundation for streamlining realtor-investor collaboration, there are several exciting opportunities to expand and enhance the system going forward.

One of the largest opportunities for future expansion is increasing lead capture and CRM capabilities for realtors. Integration with leading CRM platforms like Salesforce would allow realtors to track their leads, automate follow-up emails, and keep track of deal progress all within RealEst. AI-powered analytics could identify top-priority clients and predict likely conversions. Automated drip campaigns and targeted property alerts would allow realtors to stay top-of-mind with hot leads and grow their client list organically.

Improving the investment intelligence capabilities is another promising area. Adding historical appreciation of property and projected rent growth would provide investors with a better understanding of a listing's long-term value. Collaborating with real estate data providers such as CoreLogic or ATTOM Data Solutions would allow us to develop strong machine learning models that forecast future asset value based on local market conditions, economic data, and property attributes. Presented in an intuitive, visual way, these findings would cement RealEst as a valuable utility for savvy investors.

Personalizing the investor experience through intelligent property matching is another major opportunity. Leveraging a learning-based recommendation engine that can draw upon the swiping behavior, saved searches, and profile data of every user could generate highly relevant listing recommendations. Collaborative filtering algorithms would identify underlying preferences and surface appropriate properties that may have otherwise gone unnoticed. Over time, the app would be a smart advisor, automatically alerting users to high-potential deals that meet their particular specifications.

Looking into the distant future, RealEst can be a beginning point for PropTech disruptors. Some of the concepts are a platform for fractional rental property ownership and cash flow sharing, enabling mass market investors to purchase shares readily in income-generating real estate. It will make acquisitions, financing, management and distributions for them, lowering barriers of entry. The second concept is a salary disclosure platform for property professionals, an idea borrowed from Levels.fyi. This would match anonymized commission and compensation data, allowing agents to compare their payouts and negotiate better splits.

Finally, taking RealEst from a basic app to a full ecosystem is ambitious but a possible idea. Imagine a browser extension where users can instantly analyze any property they find on the web. With a single click, they could access RealEst's cash flow estimates, local market information, and smart value estimations without ever leaving the listing page. MLS connections would pre-populate important fields, removing time from data entry for investors. Realtors could view which of their

clients are searching where, tailor recommendations, and initiate highest-priority contact, all within their RealEst dashboard. This "RealEst everywhere" strategy would embed the platform into the fabric of the new real estate investment process.

As property technology continues to evolve at a breakneck speed, RealEst is well-positioned to be a leader in the industry for facilitating data-driven decision-making, forming successful partnerships, and broadening access. By relentlessly focusing on our users' needs, constantly iterating, and thoughtfully integrating new technologies, we can build a platform that truly makes realtors' and retail investors' lives better. The prospect of property investment is bright, and RealEst will be at the forefront.

10 Conclusion

Within this project, we developed RealEst, an innovative platform that makes real estate investment easier by connecting realtors and investors through a data-driven, easy-to-use interface. Our solution addresses critical pain points within the industry, such as poor property analysis, tedious client communication, and expectations misalignment.

By virtue of its dual-UI approach, RealEst provides customized experiences for both investors and realtors. Realtors benefit from enhanced property search and filtering, automatic cash flow forecasting, and easy client management functionality. Investors, on the other hand, benefit from a carefully curated pipeline of properties that meet their criteria, view detailed financial analysis, and provide instant feedback through an easy swipe interface.

To guarantee the quality of our solution, we performed thorough functionality and usability testing. Unit and integration tests confirmed the accuracy of cash flow calculations and responsiveness of main features. User testing with realtors and investors indicated high satisfaction levels, with users praising the improved efficiency, clarity of information, and seamless collaboration enabled by RealEst.

At its core, RealEst is a tremendous step forward in the democratization of real estate investing by empowering realtors with amazing tools and enabling investors to make informed decisions with confidence. By using technology to bridge the gap between these important stakeholders, our platform has the potential to transform the way that real estate investments are discovered, analyzed, and transacted. Looking back over the journey, we are proud of the collaboration, technical innovation, and user-centric thinking that brought RealEst into being. The learning in agile development, evidence-based design, and stakeholder management will certainly serve us well in our future endeavors.

References

1. Al-Bashrawi, M., Ahmad, M. N., Ismail, A., & Anadale, F. (2021). Recommender systems in the real estate market—a survey. *Applied Sciences*, 11(16), 7502. <https://www.mdpi.com/2076-3417/11/16/7502>
2. Arrived. (2023). Fractional real estate investing platform. <https://arrived.com>
3. Chen, T. (2023). Democratizing real estate investment through fractional ownership models. *Journal of Property Investment & Finance*, 41(2), 112-128.
4. DealCheck. (2023). Real estate analysis software for investors. <https://dealcheck.io>
5. Fundrise. (2022). Real estate investment platform. <https://fundrise.com>
6. Garcia, M., & Lee, S. (2024). CRM systems in modern real estate practice. *Real Estate Technology Review*, 8(1), 45-62.
7. Google. (2023). Material design guidelines. <https://material.io/design/>
8. Inspiration Supply. (2022). Design mobile swiping that enhances user experience. Medium. <https://medium.com/inspiration-supply/design-ideas-for-mobile-menus-df307db05853>
9. Jackson, P. (2024). Accessible swipe design: Inclusive practices for mobile interfaces. *Accessibility Today*, 6(2), 73-88.
10. Johnson, R., Williams, T., & Roberts, A. (2022). Digital marketplaces in rental property investment. *Journal of Real Estate Technology*, 15(3), 219-237.
11. LogRocket. (2023). Designing swipe-to-delete and swipe-to-reveal interactions. <https://blog.logrocket.com/ux-design/accessible-swipe-contextual-action-triggers/>
12. Martinez, E. (2024). Alternative investment platforms and their impact on real estate markets. *Journal of Alternative Investments*, 26(4), 102-117.
13. Meier, C., Zhang, H., & Wu, D. (2022). Application of data mining technology in evaluating real estate investment plan based on GRA-AHP. *Data Science in Real Estate*, 4(2), 156-173.
14. National Association of REALTORS. (2023). Real estate agent productivity report. <https://www.nar.realtor>
15. PropertyBase. (2024). CRM platform for real estate professionals. <https://www.propertybase.com>
16. Realtor Magazine. (2024). Trends and analysis in real estate. <https://www.realtormag.realtor.org/>
17. Rodriguez, S., & Kim, J. (2023). Case-based reasoning approaches in property recommendation systems. *Artificial Intelligence in Real Estate*, 7(3), 284-301.
18. Roofstock. (2022). Investment property marketplace. <https://www.roofstock.com>
19. Smith, J. (2023). Digital tools for real estate investment analysis. *Property Technology Journal*, 12(4), 328-345.
20. Statista. (2023). Real estate agents' use of mobile devices in the United States. <https://www.statista.com/statistics/1051114/real-estate-agents-use-of-mobile-devices-us/>
21. Taylor, M., & Brown, N. (2022). Crowdfunding platforms in real estate: Opportunities and challenges. *Real Estate Finance Journal*, 37(2), 91-105.

22. Thompson, D. (2023). Swipe interfaces and user engagement metrics in mobile applications. *User Experience Design Quarterly*, 18(1), 42-56.
23. Thompson, R., & Davis, K. (2023). Mobile technology adoption among real estate professionals. *Journal of Real Estate Practice*, 25(2), 173-189.
24. Wang, X., Guo, L., Wang, J., & Zhang, Z. (2016). Real estate recommender system using case-based reasoning approach. *International Conference on Knowledge Science, Engineering and Management*, 418-429.
https://www.researchgate.net/publication/308052818_Real_estate_recommender_system_using_case-based_reasoning_approach
25. Wilson, T., Martinez, L., & Chang, K. (2022). Comprehensive survey of recommender system approaches and challenges. *Journal of Information Systems*, 46(3), 219-248.
26. Williams, A. (2023). Automated property analysis tools and their accuracy. *Real Estate Technology Review*, 7(4), 213-229.
27. YieldStreet. (2024). Alternative investments platform. <https://www.yieldstreet.com>
28. Zilculator. (2023). Real estate analysis software. <https://zilculator.com/>
29. Zillow Economic Research. (2024). Housing market trends and economic analyses.
<https://www.zillow.com/research/>
30. BiggerPockets. (2024). Real estate investment listings.
<https://www.biggerpockets.com/listings>

Appendix A: Project Management

A.1 Team Agreement

Team Agreement

PropertyPro - 02/03/2025

Team Goals and Objectives

Team-Driven Approach:

Our team strives to build software that is intuitive and scalable. We emphasize clean code, modular design, and comprehensive testing to create a platform that can evolve with user needs. We believe that maintainability is key to long-term success, so our development practices include regular code reviews and agile methodologies.

Platform Goals:

- **User-Friendly Experience:** We provide an interface that is easy to navigate for both realtors and buyers. This includes a swipe-based property selection experience and clear notifications to streamline communication.
- **Seamless Integration:** The system integrates MLS data and offers smart property filtering. This ensures that realtors can quickly provide detailed property recommendations, including cash flow analysis, to potential buyers.
- **Efficient Communication:** Our design enables realtors to send recommended listings to buyers effortlessly, while buyers can easily respond with their interest, saving for later, or declining a property. This back-and-forth communication is handled through real-time notifications, keeping both parties informed.

Roles and Responsibilities

This section outlines the roles assigned to each member and the related responsibilities those roles carry. It should be noted that while we are listing the specialists for each role, each team member will still make an effort to work on some frontend, backend, and CI/CD tasks. However the specialist will have control of any major decisions that need to be made relating to the organization and implementation of features.

Project Manager (Dinesh Balakrishnan):

- Manages the relationship with key stakeholders (Client, Professor, Users, etc.)
- Ensures that the product is on track to be complete within the expected timeframe
- Assigns work to team members and manages the weekly standup meetings.

Front-End Lead (Eshwar Gadi):

- Responsible for designing and implementing the user interface using modern front-end frameworks.
- Ensures responsiveness and usability for realtors and buyers.
- Collaborates with the back-end team for API integration.

Back-End Lead (Daniel Pandyan):

- Develops and maintains the server-side logic and database structure.
- Ensures seamless integration with MLS data and cash flow analysis tools.
- Implements security measures to protect user data.

CI/CD Lead (Arjun Som):

- Sets up and maintains automated testing, deployment pipelines, and version control best practices.
- Ensures continuous integration to streamline development and deployment.
- Manages staging and production environments.

Communication Expectations

All primary team communication will occur through our team's Discord server. Discord is where the team can propose questions for the client, ask for a review of a pull request, collaborate on assignments, and clarify general questions related to the project. Weekly updates and coordination will happen through a dedicated discord channel, with urgent matters being addressed via direct messages. Any significant project decisions will be documented to maintain transparency. Open and respectful communication is encouraged, ensuring that everyone stays informed and aligned with project goals. All team members are expected to be responsive within 6 hours of being pinged. When not pinged, all team members are expected to confirm their viewership of notifications addressing the project with a thumbs up within 12 hours.

Meeting Structure and Availability Expectations

Given that we are college students working on a one-semester project, meetings will be flexible to accommodate academic schedules. Meetings will primarily take place on Discord, with additional discussions occurring asynchronously as needed. The structure is as follows:

- **Weekly Check-Ins:** Held on Discord at a time agreed upon by the team to discuss progress, blockers, and next steps.
- **Bi-Weekly/Monthly Mentor Meetings:** Regular meetings with our mentor to review progress, seek feedback, and adjust project direction as needed.

- **Documentation:** All major decisions and action points will be recorded in Discord for future reference.

Availability: Team members should communicate their availability and respond within a reasonable timeframe.

Change Requests

All change requests must be submitted via Discord and assigned to the relevant lead. Requests should include a clear description, rationale, and expected impact. Minor changes will be reviewed within a day, while major changes require team discussion and approval.

Team Agreement Maintenance

This agreement will be reviewed at the beginning of each sprint. Modifications require consensus from all team members. Any unresolved disputes will be escalated to the team lead or the mentor.

Team Member Signatures

Dinesh Balakrishnan, Eshwar Reddy Gadi, Daniel Pandyan, Arjun Som

A.2 Software Development Methodology

Our software development method will be Agile. Agile, with its focus on flexibility, collaboration, and ongoing improvement, best aligns our project goals. This alignment guarantees the PropertyPro platform is developed economically, user-friendly, and responsive to changing needs, with a degree of certainty as to the project direction.

How Agile Aligns with Our Project Goals:

Flexibility and Adaptability

Agile allows us to rapidly respond to changing requirements and user feedback. This will allow us to focus on delivering the most important features to investors and realtors in a dynamic way.

Collaboration and Communication

Agile focuses on tight collaboration among team members and stakeholders. Frequent meetings and demos will make sure that everyone is in sync on the project development and objectives.

Continuous Improvement

By implementing frequent retrospectives and iterative development, Agile ensures ongoing improvement in the quality and functionality of the platform.

Implementation Strategy:

Agile Sprints

Our 8-week development cycle consists of four 2-week sprints. Planning, development, testing, and review will be the components of each sprint.

- **Sprint 1:** Setup of initial development environment, development of core features, and prioritization of user stories.
- **Sprint 2:** Scale core functionality, user test, and implement feedback.
- **Sprint 3:** Enhance user interface elements, include user feedback improvements in both mobile and desktop versions.
- **Sprint 4:** “Final” testing, bug fixing and deployment of the “platform” for end-user testing.

User Stories and Backlog Management

We will create user stories to capture requirements and maintain a prioritized backlog. The team will collaboratively estimate and plan each sprint based on the highest priority items.

Regular Meetings and Demos

Stand-up meetings will be held every day to discuss progress, problems, and plans. Sprint demos and reviews will showcase finished work to stakeholders for feedback.

We are confident that applying Agile in developing PropertyPro will result in a strong, efficient, and feature-packed platform that will satisfy the needs of realtors and investors and yet remain adaptable and open to continuous improvement throughout the development phase.

A.3 Implementation Schedule

Below outlines the planned implementation schedule for the project development. Keep in mind that this plan could change over time and that the scope of the project will be limited given the time and resources limited compared to an ideal fully featured implementation. The goal is to implement all basic core functionality within the first two sprints and start implementing features during the last two sprints.

Sprint 1: Week 4-5

Initialize application and set up application with firebase, github, CI/CD (codemagic), and create any databases. This is a **must-have** since without any of these services or setup, the development of the application cannot start. This is in sprint 1 because all other work depends on this being completed first.

Activate API Keys for all APIs and start subscriptions for services we will be using for the development. This is a **must-have** since without this, our application will not get data, and therefore it will be useless. This is in sprint 1 because it is very important to figure out what APIs we will be using and get access before implementing it into our project.

Implement authentication for Realtors and allow them to set up accounts. This is a **must-have** since it is basic core functionality that is a crucial part of the flow for the user. This is in sprint 1 because it should be fairly simple and gives us some progress app development wise.

Finalize ER diagram for database and system design. This is a **must-have** because we need to have an understanding of all the entities and database design before moving forward with development since large changes later will cause significant extra work. This is in sprint 1 because it is required work before further development.

Sprint 2: Week 6-7

Implement authentication for the buyer via link from realtor. Complete authentication and authgate implementation. This is a **must-have** because it is a core functionality and makes sure that the app we build is secure. This is in sprint 2 because we need this working before implementing further core functionality.

Implement imap email protocol for sending homes and emails to users. This is a **nice-to-have** because we could just expect the user to select the realtor in the app. But this would likely be a much

easier implementation and be easier for the user so it makes sense to implement this now so that is why it is included in the sprint.

Implement a home search page for realtors and buyers with basic filters. This is a **must-have** and is a major portion of the core functionality for the application. Most of the effort this sprint will likely be directed towards this. It is included in this sprint because many features we will implement next are based on this.

Implement cash flow insights calculation given address. This is a **must-have** because it is what makes our application a novel idea. This will require some time learning about cash flow generation with greg. It is included in this sprint since it is an important part of the application idea and we want to make sure it is implemented before other things.

Sprint 3: Week 8-9

Refine home search page and Implement tinder-style swiping mechanism. This is a **must-have** because it is how realtors and buyers actually see the houses and send them to each other. It is included in this sprint because it ties into other features that will be implemented alongside.

Implement ability to view previously disapproved homes for both realtor and buyer. This is a **nice-to-have**. However it should be fairly hassle free to implement. It is included in this sprint because it can be done simply after the above task is implemented.

Implement agent dashboard for client management. This is a **must-have**. Without this the agent will not be able to see what their clients are up to or review what their clients have sent. This is important core-functionality. It is included in this sprint because it ties into other features that will be implemented alongside.

Sprint 4: Week 10-11

Implement a notification system for push notifications and email. This is **nice to have**. We will implement it if time permits. This would bolster the multi-platform appeal of our application.

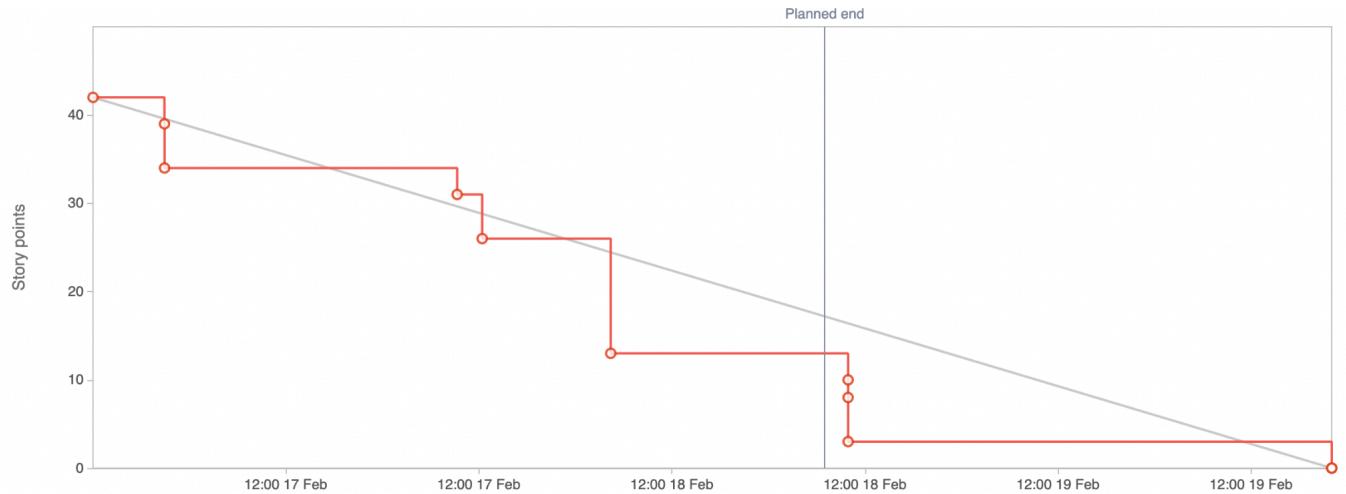
Implement more financial calculators as suggested by Greg. This is **nice to have**. We will implement it if time permits.

Refine user interface and improve responsiveness. This is a **must-have**. It is included in this sprint because we want to make sure that before development concludes, the user interface is implemented correctly and is beautiful.

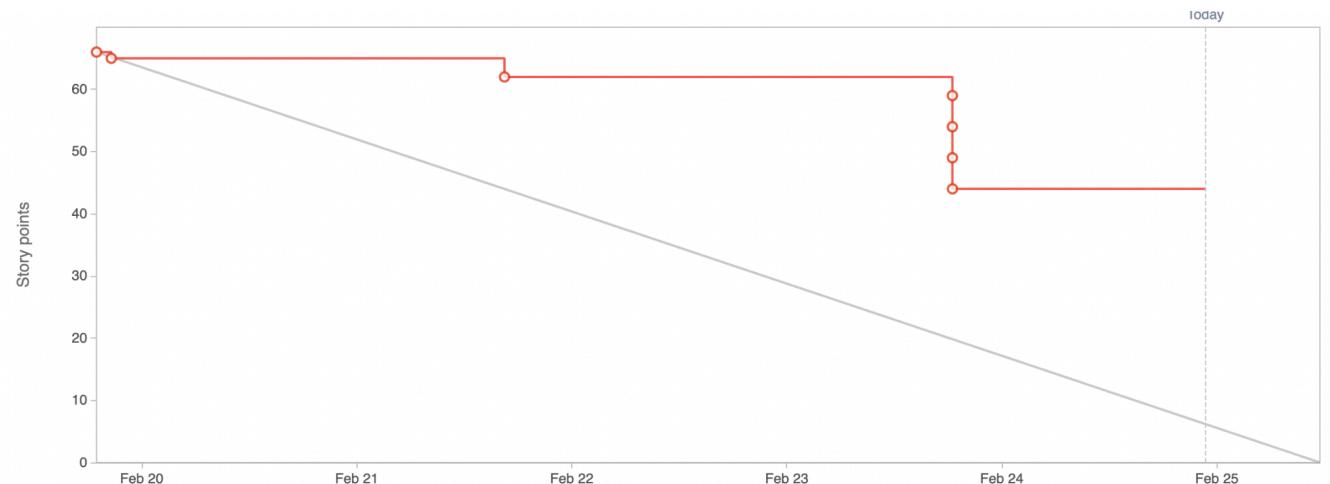
Implement a web compatible view for agents/realtors. This is **nice to have**. We will implement it if time permits.

A.4 Software Development Artifacts

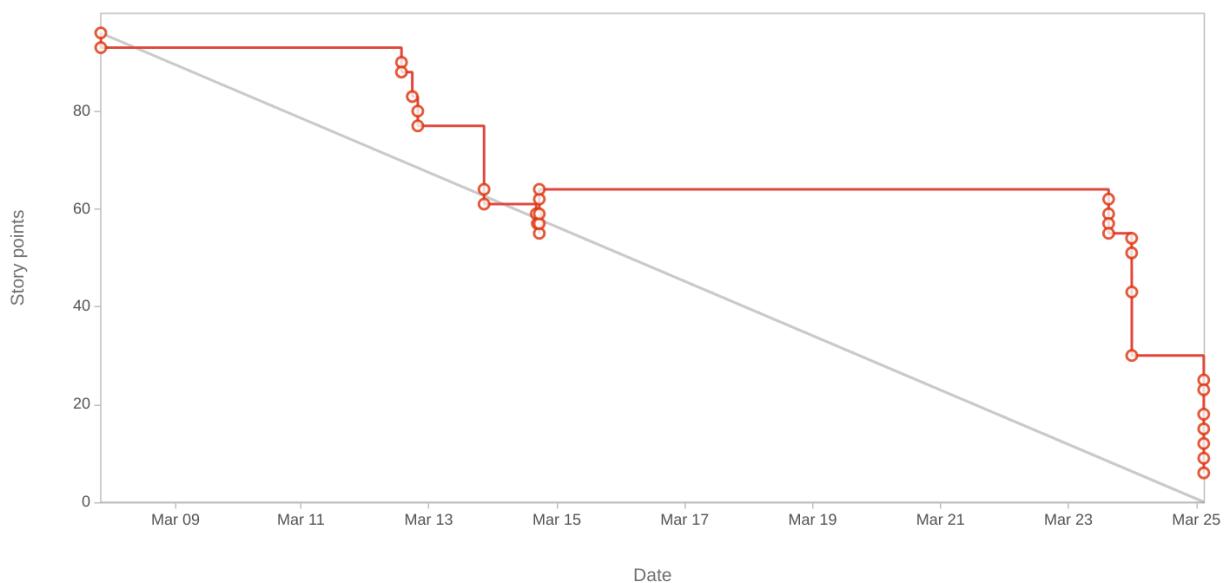
Burndown Chart (Sprint 1)



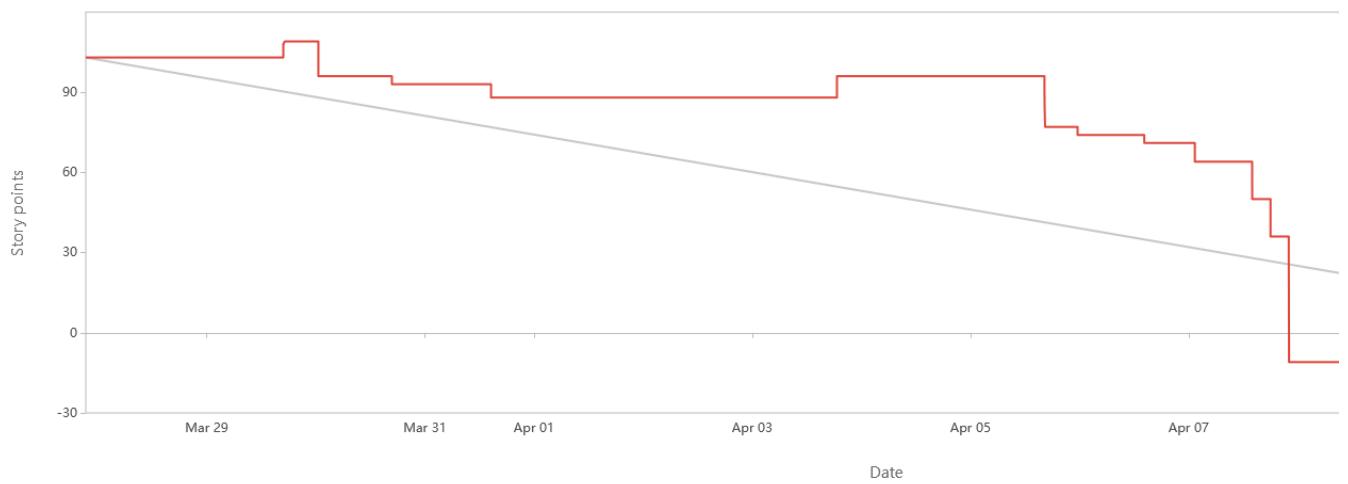
Burndown Chart (Sprint 2)



Burndown Chart (Sprint 3)



Burndown Chart (Sprint 4)



Sprint 1 Backlog

Key	Summary	Issue type	Epic	Status	Assignee	Story points
REAL-16	add firebase to the app	<input checked="" type="checkbox"/> Task		DONE	DB	5
REAL-17	create and finalize ERD	<input checked="" type="checkbox"/> Task		DONE	DB	13
REAL-18	figuring out what secrets we need for the GH repo and loca...	<input checked="" type="checkbox"/> Task		DONE	DB	2
REAL-19	create databases/rules	<input checked="" type="checkbox"/> Task		DONE	D	3
REAL-14	each person initialize application	<input checked="" type="checkbox"/> Task		DONE		3
REAL-15	setup CI/CD (code magic)	<input checked="" type="checkbox"/> Task		DONE	DB	5
REAL-20	do some exploration on the relationship between firebase ...	<input checked="" type="checkbox"/> Task		DONE	D	3
REAL-21	activate api keys and start subscriptions	<input checked="" type="checkbox"/> Task		DONE		3
REAL-22	implement authentication for realtors	<input checked="" type="checkbox"/> Task		DONE	D	5

Sprint 2 Backlog

KAN Sprint 2 19 Feb – 25 Feb (17 issues)		28	16	22	Complete sprint	...
<input checked="" type="checkbox"/>	REAL-39 fix git history	DONE	1	D		
<input checked="" type="checkbox"/>	REAL-38 Integrate the Firebase Auth template with our UI.	DONE	3	E		
<input type="checkbox"/>	<input checked="" type="checkbox"/> REAL-13 add settings	DONE	3	E	...	
<input checked="" type="checkbox"/>	REAL-28 refine home search page UI	DONE	5	E		
<input checked="" type="checkbox"/>	REAL-23 different landing page for buyers vs realtors	DONE	5	E		
<input checked="" type="checkbox"/>	REAL-43 implement agent dashboard ui skeleton	DONE	5	E		
<input checked="" type="checkbox"/>	REAL-5 Figure out what goes in the "Cash Flow" section	TO DO	3	DB		
<input checked="" type="checkbox"/>	REAL-34 login with google	IN PROGRESS...	2	D		
<input checked="" type="checkbox"/>	REAL-40 CI/CD setup (CircleCI)	IN PROGRESS...	5	A		
<input checked="" type="checkbox"/>	REAL-11 refine swipe card implementation to look better	IN PROGRESS...	3	E		
<input checked="" type="checkbox"/>	REAL-36 figure out .gitignore issue	VALIDATION	1	D		
<input checked="" type="checkbox"/>	REAL-41 Setup the automatic android APK creation with GH actions	IN PROGRESS...	5	A		
<input checked="" type="checkbox"/>	REAL-42 Run all flutter tests with CI/CD	TO DO	3	A		
<input checked="" type="checkbox"/>	REAL-44 setup data job for listings	TO DO	13	D		
<input checked="" type="checkbox"/>	REAL-46 add first time user in backend	TO DO	3	D		
<input checked="" type="checkbox"/>	REAL-47 add first time user intro for home page	TO DO	3	👤		
<input checked="" type="checkbox"/>	REAL-48 dark mode implementation	TO DO	3	DB		

Sprint 3 Backlog

Completed issues							View in issue
Key	Summary	Issue type	Epic	Status	Assignee	Story points	
REAL-29	Implement tinder style swiping	<input checked="" type="checkbox"/> Task		<div style="width: 100%;">DONE</div>		3	
REAL-50	Visual indicator that left is bad + right good	<input checked="" type="checkbox"/> Task		<div style="width: 100%;">DONE</div>		2	
REAL-54	Complete ML model for rent forecast	<input checked="" type="checkbox"/> Task		<div style="width: 100%;">DONE</div>		13	
REAL-60	Figure out a more efficient method to load data and manage local state	<input checked="" type="checkbox"/> Task		<div style="width: 100%;">DONE</div>		8	
REAL-61	Look into recommendation algo given filters	<input checked="" type="checkbox"/> Task		<div style="width: 100%;">DONE</div>		3	
REAL-62	Filters for investor side	<input checked="" type="checkbox"/> Task		<div style="width: 100%;">DONE</div>		3	
REAL-63	Realtor should be able to send invite link	<input checked="" type="checkbox"/> Task		<div style="width: 100%;">DONE</div>		3	
REAL-65	Implement realtor dashboard page	<input checked="" type="checkbox"/> Task		<div style="width: 100%;">DONE</div>		5	
REAL-67	Implement rental property calc calculations	<input checked="" type="checkbox"/> Task		<div style="width: 100%;">DONE</div>		3	
REAL-68	Implement PITI calc	<input checked="" type="checkbox"/> Task		<div style="width: 100%;">DONE</div>		3	
REAL-69	Implement affordability calc	<input checked="" type="checkbox"/> Task		<div style="width: 100%;">DONE</div>		3	
REAL-56	Define a universal theme for UI	<input checked="" type="checkbox"/> Task		<div style="width: 100%;">DONE</div>		1	
REAL-57	Go through and replace the theming and elements	<input checked="" type="checkbox"/> Task		<div style="width: 100%;">DONE</div>		3	
REAL-58	Implement a property details section for realtors	<input checked="" type="checkbox"/> Task		<div style="width: 100%;">DONE</div>		3	
REAL-25	Implement IMAP/email protocol for emailing user	<input checked="" type="checkbox"/> Task		<div style="width: 100%;">DONE</div>		3	
REAL-30	Implement ability for realtors to view previously disapproved and approved homes	<input checked="" type="checkbox"/> Task		<div style="width: 100%;">DONE</div>		5	
REAL-27	Implement cash flow insights calculation given address	<input checked="" type="checkbox"/> Task		<div style="width: 100%;">DONE</div>		3	
REAL-70	Implement the clients page	<input checked="" type="checkbox"/> Task		<div style="width: 100%;">DONE</div>		5	
REAL-72	code refactor/guidelines	<input checked="" type="checkbox"/> Task		<div style="width: 100%;">DONE</div>		13	
REAL-73	implement testing coverage	<input checked="" type="checkbox"/> Task		<div style="width: 100%;">DONE</div>		2	
REAL-74	add unit tests for realtor/investor auth	<input checked="" type="checkbox"/> Task		<div style="width: 100%;">DONE</div>		3	
REAL-75	add unit tests for realtor settings page	<input checked="" type="checkbox"/> Task		<div style="width: 100%;">DONE</div>		2	
REAL-76	add unit tests for realtor calculators page	<input checked="" type="checkbox"/> Task		<div style="width: 100%;">DONE</div>		2	
REAL-78	Implement ability for buyers to view previously disapproved and approved homes	<input checked="" type="checkbox"/> Task		<div style="width: 100%;">DONE</div>		2	

Issues completed outside of sprint							View in issue
Key	Summary	Issue type	Epic	Status	Assignee	Story points	
REAL-59	Make Navbar corner square and implement expand/collapse	<input checked="" type="checkbox"/> Task		<div style="width: 100%;">DONE</div>		5	
REAL-64	Fix user state information loading	<input checked="" type="checkbox"/> Task		<div style="width: 100%;">DONE</div>		2	
REAL-66	Add profile icon and replace settings button in navbar	<input checked="" type="checkbox"/> Task		<div style="width: 100%;">DONE</div>		3	
REAL-43	implement agent dashboard ui skeleton	<input checked="" type="checkbox"/> Task		<div style="width: 100%;">DONE</div>		5	

Sprint 4 Backlog

Completed issues							View in is
Key	Summary	Issue type	Epic	Status	Assignee	Story points	
REAL-80	allow buyers to send notes to realtor/comment on home	<input checked="" type="checkbox"/> Task		<div style="background-color: #e0f2e0; width: 100px; height: 10px;"></div>	D	3	
REAL-82	implement search in home search	<input checked="" type="checkbox"/> Task		<div style="background-color: #e0f2e0; width: 100px; height: 10px;"></div>	DB	5	
REAL-83	implement filter by home type, size, and more	<input checked="" type="checkbox"/> Task		<div style="background-color: #e0f2e0; width: 100px; height: 10px;"></div>	DB	5	
REAL-84	implement home details on investor side	<input checked="" type="checkbox"/> Task		<div style="background-color: #e0f2e0; width: 100px; height: 10px;"></div>	DB	3	
REAL-85	create debug mode for local development (make sure images can be seen)	<input checked="" type="checkbox"/> Task		<div style="background-color: #e0f2e0; width: 100px; height: 10px;"></div>	DB	1	
REAL-87	generate link for realtors to invite clients and show it in app and allow realtor to ...	<input checked="" type="checkbox"/> Task		<div style="background-color: #e0f2e0; width: 100px; height: 10px;"></div>	DB	5	
REAL-86	allow realtor to tag their clients	<input checked="" type="checkbox"/> Task		<div style="background-color: #e0f2e0; width: 100px; height: 10px;"></div>	DB	3	
REAL-95	allow realtor to be able to send homes to clients based on tags and name lookup	<input checked="" type="checkbox"/> Task		<div style="background-color: #e0f2e0; width: 100px; height: 10px;"></div>	E	5	
REAL-88	be able to generate information about how many interactions their clients have	<input checked="" type="checkbox"/> Task		<div style="background-color: #e0f2e0; width: 100px; height: 10px;"></div>	D	3	
REAL-97	export client data as a .csv file	<input checked="" type="checkbox"/> Task		<div style="background-color: #e0f2e0; width: 100px; height: 10px;"></div>	DB	3	
REAL-92	be able to verify realtors license through scraping TREC	<input checked="" type="checkbox"/> Task		<div style="background-color: #e0f2e0; width: 100px; height: 10px;"></div>	A	8	
REAL-89	be able to download full report for any specific property listed	<input checked="" type="checkbox"/> Task		<div style="background-color: #e0f2e0; width: 100px; height: 10px;"></div>	DB	5	
REAL-96	store interactions about what investor does in app	<input checked="" type="checkbox"/> Task		<div style="background-color: #e0f2e0; width: 100px; height: 10px;"></div>	D	5	
REAL-101	clients page: add leads column, active leads, and clients	<input checked="" type="checkbox"/> Task		<div style="background-color: #e0f2e0; width: 100px; height: 10px;"></div>	DB	5	
REAL-102	add buyer net sheet calculator	<input checked="" type="checkbox"/> Task		<div style="background-color: #e0f2e0; width: 100px; height: 10px;"></div>	A	3	
REAL-103	add pagination to home search listings	<input checked="" type="checkbox"/> Task		<div style="background-color: #e0f2e0; width: 100px; height: 10px;"></div>	DB	3	
REAL-107	make searching clients in clients page work	<input checked="" type="checkbox"/> Task		<div style="background-color: #e0f2e0; width: 100px; height: 10px;"></div>	D	5	
REAL-106	implement rentometer	<input checked="" type="checkbox"/> Task		<div style="background-color: #e0f2e0; width: 100px; height: 10px;"></div>	DB	3	
REAL-108	make filters in clients page work	<input checked="" type="checkbox"/> Task		<div style="background-color: #e0f2e0; width: 100px; height: 10px;"></div>	D	3	
REAL-105	show cashflow quick view in property listing card	<input checked="" type="checkbox"/> Task		<div style="background-color: #e0f2e0; width: 100px; height: 10px;"></div>	DB	3	
REAL-98	Add a dashboard column that shows if a property that customer likes has changed	<input checked="" type="checkbox"/> Task		<div style="background-color: #e0f2e0; width: 100px; height: 10px;"></div>	D	5	
REAL-104	add more details to home details sheet	<input checked="" type="checkbox"/> Task		<div style="background-color: #e0f2e0; width: 100px; height: 10px;"></div>	DB	3	
REAL-99	Add a dashboard column for realtor to see pinned/top clients	<input checked="" type="checkbox"/> Task		<div style="background-color: #e0f2e0; width: 100px; height: 10px;"></div>	D	3	
REAL-100	Make notifications column work (client updates/changes)	<input checked="" type="checkbox"/> Task		<div style="background-color: #e0f2e0; width: 100px; height: 10px;"></div>	D	3	
REAL-109	Add a dashboard column to show most active clients	<input checked="" type="checkbox"/> Task		<div style="background-color: #e0f2e0; width: 100px; height: 10px;"></div>	D	5	
REAL-110	Add a home screen for both realtor and investor, rather than just having the login...	<input checked="" type="checkbox"/> Task		<div style="background-color: #e0f2e0; width: 100px; height: 10px;"></div>	E	5	
REAL-111	Create a backend server on vercel to run the IMAP protocol	<input checked="" type="checkbox"/> Task		<div style="background-color: #e0f2e0; width: 100px; height: 10px;"></div>	E	1	
REAL-91	testing coverage	<input checked="" type="checkbox"/> Task		<div style="background-color: #e0f2e0; width: 100px; height: 10px;"></div>	A	8	

Product Backlog

Key	↑	Summary	Status	Sprint	Assignee	Updated
REAL-5		Figure out what goes in the "Cash Flow" section	DONE	✓ KAN Sprint 2	DB Dinesh Balakrishnan	Mar 2, 2025
REAL-11		refine swipe card implementation to look better	DONE	✓ KAN Sprint 2	E egadi2004	Feb 25, 2025
REAL-13		add settings	DONE	✓ KAN Sprint 2	E egadi2004	Feb 23, 2025
REAL-14		each person initialize application	DONE	✓ Sprint 1		Feb 16, 2025
REAL-15		setup CI/CD (code magic)	DONE	✓ Sprint 1	DB Dinesh Balakrishnan	Feb 17, 2025
REAL-16		add firebase to the app	DONE	✓ Sprint 1	DB Dinesh Balakrishnan	Feb 16, 2025
REAL-17		create and finalize ERD	DONE	✓ Sprint 1	DB Dinesh Balakrishnan	Feb 17, 2025
REAL-18		figuring out what secrets we need for the GH repo and L...	DONE	✓ Sprint 1	DB Dinesh Balakrishnan	Feb 18, 2025
REAL-19		create databases/rules	DONE	✓ Sprint 1	D d_pandyan	Feb 18, 2025
REAL-20		do some exploration on the relationship between firebas...	DONE	✓ Sprint 1	D d_pandyan	Feb 19, 2025
REAL-21		activate api keys and start subscriptions	DONE	✓ Sprint 1		Feb 17, 2025
REAL-22		implement authentication for realtors	DONE	✓ Sprint 1	D d_pandyan	Feb 18, 2025
REAL-23		different landing page for buyers vs realtors	DONE	✓ KAN Sprint 2	E egadi2004	Feb 23, 2025

REAL-24	implement authgates verification	TO DO		DB Dinesh Balakrishnan	Apr 1, 2025
REAL-25	Implement IMAP/email protocol for emailing user	DONE	✓ KAN Sprint 3	E egadi2004	Mar 12, 2025
REAL-27	Implement cash flow insights calculation given address	DONE	✓ KAN Sprint 3	DB Dinesh Balakrishnan	Mar 25, 2025
REAL-28	refine home search page UI	DONE	✓ KAN Sprint 2	E egadi2004	Feb 23, 2025
REAL-29	Implement tinder style swiping	DONE	✓ KAN Sprint 3	D d_pandyan	Mar 7, 2025
REAL-30	Implement ability for realtors to view previously disapp...	DONE	✓ KAN Sprint 3	D d_pandyan	Mar 25, 2025
REAL-33	login with apple	TO DO		A arjunsom	Apr 1, 2025
REAL-34	login with google	TO DO		D d_pandyan	Mar 4, 2025
REAL-36	figure out .gitignore issue	DONE	✓ KAN Sprint 2	D d_pandyan	Feb 27, 2025
REAL-37	implement custom theming for the realtor to customize	TO DO	KAN Sprint 5		Apr 11, 2025
REAL-38	Integrate the Firebase Auth template with our UI.	DONE	✓ KAN Sprint 2	E egadi2004	Feb 21, 2025
REAL-39	fix git history	DONE	✓ KAN Sprint 2	D d_pandyan	Feb 19, 2025
REAL-40	CI/CD setup (Github)	DONE	✓ KAN Sprint 2	DB Dinesh Balakrishnan	Mar 3, 2025

REAL-41	Setup the automatic android APK creation with GH action	DONE	(✓) KAN Sprint 2	DB Dinesh Balakrishnan	Mar 2, 2025
REAL-42	Run all flutter tests with CI/CD	DONE	(✓) KAN Sprint 2	DB Dinesh Balakrishnan	Mar 3, 2025
REAL-43	implement agent dashboard ui skeleton	DONE	(✓) KAN Sprint 3	DB Dinesh Balakrishnan	Mar 7, 2025
REAL-44	setup data job for listings	DONE	(✓) KAN Sprint 2	D d_pandyan	Mar 3, 2025
REAL-46	add first time user in backend	DONE	(✓) KAN Sprint 2	DB Dinesh Balakrishnan	Mar 2, 2025
REAL-47	add first time user intro for home page	DONE	(✓) KAN Sprint 2		Mar 2, 2025
REAL-48	dark mode implementation	DONE	(✓) KAN Sprint 2	DB Dinesh Balakrishnan	Mar 2, 2025
REAL-49	support left or right on a card in the backend	DONE	(✓) KAN Sprint 2		Mar 4, 2025
REAL-50	Visual indicator that left is bad + right good	DONE	(✓) KAN Sprint 3	D d_pandyan	Mar 14, 2025
REAL-54	Complete ML model for rent forecast	DONE	(✓) KAN Sprint 3	DB Dinesh Balakrishnan	Mar 13, 2025
REAL-56	Define a universal theme for UI	DONE	(✓) KAN Sprint 3	DB Dinesh Balakrishnan	Mar 23, 2025
REAL-57	Go through and replace the theming and elements	DONE	(✓) KAN Sprint 3	A arjunson	Mar 23, 2025
REAL-58	Implement a property details section for realtors	DONE	(✓) KAN Sprint 3	DB Dinesh Balakrishnan	Mar 13, 2025

REAL-59	Make Navbar corner square and implement expand/collapse	DONE	(✓) KAN Sprint 3	E egadi2004	Mar 7, 2025
REAL-60	Figure out a more efficient method to load data and make it faster	DONE	(✓) KAN Sprint 3	D d_pandyan	Mar 23, 2025
REAL-61	Look into recommendation algo given filters	DONE	(✓) KAN Sprint 3	A arjunson	Mar 25, 2025
REAL-62	Filters for investor side	DONE	(✓) KAN Sprint 3	E egadi2004	Mar 27, 2025
REAL-63	Realtor should be able to send invite link	DONE	(✓) KAN Sprint 3	E egadi2004	Mar 12, 2025
REAL-64	Fix user state information loading	DONE	(✓) KAN Sprint 3	E egadi2004	Mar 7, 2025
REAL-65	Implement realtor dashboard page	DONE	(✓) KAN Sprint 3	E egadi2004	Mar 25, 2025
REAL-66	Add profile icon and replace settings button in navbar	DONE	(✓) KAN Sprint 3	E egadi2004	Mar 7, 2025
REAL-67	Implement rental property calc calculations	DONE	(✓) KAN Sprint 3	DB Dinesh Balakrishnan	Mar 25, 2025
REAL-68	Implement PITI calc	DONE	(✓) KAN Sprint 3	A arjunson	Mar 25, 2025
REAL-69	Implement affordability calc	DONE	(✓) KAN Sprint 3	A arjunson	Mar 25, 2025
REAL-70	Implement the clients page	DONE	(✓) KAN Sprint 3	E egadi2004	Mar 12, 2025
REAL-72	code refactor/guidelines	DONE	(✓) KAN Sprint 3	D d_pandyan	Mar 23, 2025

REAL-73	implement testing coverage	DONE	(✓) KAN Sprint 3	A arjunsom	Mar 23, 2025
REAL-74	add unit tests for realtor/investor auth	DONE	(✓) KAN Sprint 3	A arjunsom	Mar 23, 2025
REAL-75	add unit tests for realtor settings page	DONE	(✓) KAN Sprint 3	A arjunsom	Mar 23, 2025
REAL-76	add unit tests for realtor calculators page	DONE	(✓) KAN Sprint 3	A arjunsom	Mar 23, 2025
REAL-78	Implement ability for buyers to view previously disappro...	DONE	(✓) KAN Sprint 3	D d_pandyan	Mar 25, 2025
REAL-79	refactor code and ensure modular structure	TO DO	KAN Sprint 5		Apr 11, 2025
REAL-80	allow buyers to send notes to realtor/comment on home	DONE	(✓) KAN Sprint 4	D d_pandyan	Apr 7, 2025
REAL-81	implement IOS build	TO DO			Mar 25, 2025
REAL-82	implement search in home search	DONE	(✓) KAN Sprint 4	DB Dinesh Balakrishnan	Apr 1, 2025
REAL-83	implement filter buy home type, size, and more	DONE	(✓) KAN Sprint 4	DB Dinesh Balakrishnan	Apr 1, 2025
REAL-84	implement home details on investor side	DONE	(✓) KAN Sprint 4	DB Dinesh Balakrishnan	Apr 7, 2025
REAL-85	create debug mode for local development (make sure i...	DONE	(✓) KAN Sprint 4	DB Dinesh Balakrishnan	Apr 7, 2025
REAL-86	allow realtor to tag their clients	DONE	(✓) KAN Sprint 4	DB Dinesh Balakrishnan	Apr 5, 2025

REAL-87	generate link for realtors to invite clients and show it in ...	DONE	(✓) KAN Sprint 4	DB Dinesh Balakrishnan	Apr 5, 2025
REAL-88	be able to generate information about how many interact...	DONE	(✓) KAN Sprint 4	D d_pandyan	Apr 7, 2025
REAL-89	be able to download full report for any specific property...	DONE	(✓) KAN Sprint 4	DB Dinesh Balakrishnan	Apr 7, 2025
REAL-91	testing coverage	DONE	(✓) KAN Sprint 4	A arjunsom	Apr 7, 2025
REAL-92	be able to verify realtors license through scraping TREC	DONE	(✓) KAN Sprint 4	A arjunsom	Apr 7, 2025
REAL-94	implement a demo mode so users can validate app with...	IN PROGRESS	KAN Sprint 5	DB Dinesh Balakrishnan	Apr 11, 2025
REAL-95	allow realtor to be able to send homes to clients based ...	DONE	(✓) KAN Sprint 4	E egadi2004	Apr 5, 2025
REAL-96	store interactions about what investor does in app	DONE	(✓) KAN Sprint 4	D d_pandyan	Apr 7, 2025
REAL-97	export client data as a .csv file	DONE	(✓) KAN Sprint 4	DB Dinesh Balakrishnan	Apr 7, 2025
REAL-98	Add a dashboard column that shows if a property that c...	DONE	(✓) KAN Sprint 4	D d_pandyan	Apr 7, 2025
REAL-99	Add a dashboard column for realtor to see pinned/top cl...	DONE	(✓) KAN Sprint 4	D d_pandyan	Apr 7, 2025
REAL-100	Make notifications column work (client updates/changes)	DONE	(✓) KAN Sprint 4	D d_pandyan	Apr 7, 2025
REAL-101	clients page: add leads column, active leads, and clients	DONE	(✓) KAN Sprint 4	DB Dinesh Balakrishnan	Apr 1, 2025

REAL-102	add buyer net sheet calculator	DONE	KAN Sprint 4	A arjunsom	Apr 7, 2025
REAL-103	add pagination to home search listings	DONE	KAN Sprint 4	DB Dinesh Balakrishnan	Apr 1, 2025
REAL-104	add more details to home details sheet	DONE	KAN Sprint 4	DB Dinesh Balakrishnan	Apr 5, 2025
REAL-105	show cashflow quick view in property listing card	DONE	KAN Sprint 4	DB Dinesh Balakrishnan	Apr 1, 2025
REAL-106	implement rentometer	DONE	KAN Sprint 4	DB Dinesh Balakrishnan	Apr 6, 2025
REAL-107	make searching clients in clients page work	DONE	KAN Sprint 4	D d_pandyan	Apr 11, 2025
REAL-108	make filters in clients page work	DONE	KAN Sprint 4	D d_pandyan	Apr 7, 2025
REAL-109	Add a dashboard column to show most active clients	DONE	KAN Sprint 4	D d_pandyan	Apr 7, 2025
REAL-110	Add a home screen for both realtor and investor, rather t...	DONE	KAN Sprint 4	E egadi2004	Apr 5, 2025
REAL-111	Create a backend server on vercel to run the IMAP prot...	DONE	KAN Sprint 4	E egadi2004	Apr 5, 2025
REAL-112	Add HOA or not filter to home search	TO DO	KAN Sprint 5		Apr 11, 2025
REAL-113	Fix HOA showing as 0 on cashflow calculations	TO DO	KAN Sprint 5		Apr 11, 2025
REAL-114	Investors can suggest edits about cashflow	TO DO	KAN Sprint 5	DB Dinesh Balakrishnan	Apr 11, 2025

REAL-115	Pull interest rate from FRED api and store in firestore	TO DO	KAN Sprint 5		Apr 11, 2025
REAL-116	Implement investor cashflow defaults for financing	TO DO	KAN Sprint 5	DB Dinesh Balakrishnan	Apr 11, 2025
REAL-117	Get rid of agent/office info on home details	TO DO	KAN Sprint 5	DB Dinesh Balakrishnan	Apr 11, 2025
REAL-118	Add data MLS source and MLS ID to home details	TO DO	KAN Sprint 5	DB Dinesh Balakrishnan	Apr 11, 2025
REAL-119	Make sure buyer netsheet is fully implemented	TO DO	KAN Sprint 5	A arjunsom	Apr 11, 2025
REAL-120	add description of each calculator and field	TO DO	KAN Sprint 5	A arjunsom	Apr 11, 2025
REAL-121	Add filters and most recency to investor recommended f...	TO DO	KAN Sprint 5		Apr 11, 2025
REAL-122	Last 48 hours most active to dashboard widget	TO DO			Apr 11, 2025
REAL-123	make investor account after qualified lead not client	TO DO	KAN Sprint 5	DB Dinesh Balakrishnan	Apr 11, 2025
REAL-124	tag should include all (and join)	TO DO	KAN Sprint 5		Apr 11, 2025
REAL-125	make temp password random	TO DO	KAN Sprint 5	DB Dinesh Balakrishnan	Apr 11, 2025
REAL-126	add number of properties sent badge to investor side an...	TO DO	KAN Sprint 5		Apr 11, 2025
REAL-127	pdf report create report, save to report page, share with...	TO DO	KAN Sprint 5		Apr 11, 2025

REAL-128	Testing 1	TO DO	KAN Sprint 5	A arjunsom	Apr 11, 2025
REAL-129	Testing 2	TO DO	KAN Sprint 5	DB Dinesh Balakrishnan	Apr 11, 2025
REAL-130	Testing 3	TO DO	KAN Sprint 5	D d_pandyan	Apr 11, 2025
REAL-131	Testing 4	TO DO	KAN Sprint 5	E egadi2004	Apr 11, 2025
REAL-132	Make an extension to show cashflow analysis and inter...	TO DO	KAN Sprint 5		Apr 11, 2025
REAL-133	add form validation through out the app	TO DO	KAN Sprint 5		Apr 11, 2025
REAL-134	Authgate setup page	TO DO	KAN Sprint 5		Apr 11, 2025

Weekly Status Reports

Week 1:

Status at end of week

Task	Task Lead	Status	Notes
Got our Project Idea	Dinesh	Done	
We met with the client	Dinesh, Eshwar, Daniel	Done	Client is very interested and want to continue meeting
Made lofi prototype	Daniel, Arjun, Dinesh	Done	Basic
Determined the problem	Dinesh, Eshwar, Daniel, arjun	Done	Spent lots of time figuring this out

Feedback

From Whom	Feedback	Next Steps
Greg, our client	What problems in real estate exist and what type of product is needed	Understand what the minor details and asks are for our project.
Classmates	Need to communicate more effectively the purpose of the project. Some of the general recommendations we got: the app would benefit more as a “instagram-clone” rather than a “tinder-clone”. Bulk filters for realtors allow them to more effectively narrow down properties without swiping.	Focus on communicating concerns with our client.

Week 2:

We have a more solid foundation on what our application is going to look like. The big issue we needed to iron out was understanding the relationship between the client and the realtor on the platform. Realtor's will essentially act as a curator for the information that their clients interact with. They will be able to select categories of clients that they want to add a list to. This will reduce the overall amount of work that is required from the realtor. On the client side, they can either do exploration themselves, which will send the realtor the listing or (the more likely option) they browse a feed that is created by the curated listings that the realtor chose.

Going forward we are going to enhance PropertyPro by enabling buyers to access all property listings, not just those recommended by realtors. Additionally, realtors should also be able to receive notifications when clients mark a property as "Not interested." These updates aim to improve user experience and communication between realtors and clients.

We are going to be working on building out the main frontend and backend capabilities. We expect this to be a relatively easy process due to our tech stack.

Status at end of week

Task	Task Lead	Status	Notes
Meeting #2 with Greg	Dinesh	Complete	
Lo-Fi Prototyping	Eshwar	Complete	May need to reiterate
System Design	Daniel	Complete	May need to reiterate
Set up flutter and Github	Dinesh	WIP	
Pilot studies + software methodology	Arjun	Complete	DevSecOps will work in our favor

Feedback

From Whom	Feedback	Next Steps
Greg, our client	Recommended us to show all the listings to buyers rather than just the one recommended by realtors. The realtor should also get notified when the client says "Not interested".	Make sure all the listings are available and notify the realtor when their client says "not interested" too.
TA + Professor	Make sure app is compatible with iOS since we are developing using	Use an iOS device to test the app on XCode

	Android Studio	
Friends (user stories)	We realized that buyers appreciate receiving more and more listings, but they would like to have more detailed information on them.	We will try to enhance the property listing to include not only details like price, location, etc but also key information like average rent in the area, vacancy rate, etc.

Week 3:

Last week, our team built a basic UI that serves as a solid base for future work. We created a simple layout that users can navigate easily. We faced challenges with Flutter and Dart as we learned how to use these tools. We spent extra time on debugging and understanding the new coding patterns. This process helped us gain important skills that will serve the project well. Our team met regularly to share updates and discuss issues, which helped us work through problems quickly.

We also linked the front end to the backend to enable user authentication. Our work ensured that the authentication module works as expected. This integration marked an important milestone in our progress. We learned how to manage code dependencies and connect different parts of our project. Overall, despite the initial challenges, our project remains on track, and our clear communication kept everyone aligned.

This week, we plan to integrate a basic Firebase authentication template into our UI. This step will secure user access and provide a solid base for future features. We will also focus on polishing the UI. Our goal is to create a minimalistic design that is easy to navigate. This work will improve the user experience and prepare us for the next phase of integrating the MLS listings API. Our progress is very steady, and our communication is very strong. We tend to solve problems together and try to solve them as soon as possible rather than delaying the progress. The team is able to meet its deadline and stay on course for the sprint.

Status at end of week

Task	Task Lead	Status	Notes
Authentication	Daniel	WIP	Basic Email Auth Done
CI/CD	Arjun	WIP	Tests need to be completed
Front-end Development	Eshwar	WIP	Basic UI for realtors and Investors. The

			Login page for both the users.
--	--	--	--------------------------------

Feedback

From Whom	Feedback	Next Steps
Ourselves	Needs more output	Implement performance improvement plan (we made one on discord)

Week 4:

Status Report

The team made significant progress on the project. We completed the entire front end UI skeleton, which lays a solid foundation for upcoming features. The implementation of the swipe card mechanism began, aiming to improve user interaction in a manner similar to popular swipe-based apps. In parallel, we connected the front end to Firebase for user authentication, ensuring a secure and efficient login process. We also integrated GitHub with Code Magic, establishing a continuous integration and delivery pipeline that will streamline our development process and reduce deployment time. Additionally, we successfully figured out how to web scrape property listings, which will be a key feature for our application.

Despite these advances, we encountered several challenges. Integrating various systems required careful coordination, and resolving layout issues—such as oversized buttons and pixel overflow—took additional time and testing. These hurdles highlighted areas where further refinement is needed, particularly in ensuring the UI adapts smoothly across different devices.

Our goals for this week include refining the swipe card functionality and further optimizing the user interface. We plan to conduct extensive testing across multiple devices to guarantee a consistent experience. In addition, the team will focus on integrating the property listings into the front end and enhancing our CI/CD pipeline to maintain our development momentum. Overall, the team remains focused and united as we work toward delivering a stable and user-friendly application.

Current Status

- What did the team work on this past week?

Task	Task Lead	Status	Notes
------	-----------	--------	-------

Implemented CI/CD pipeline	Arjun	In progress	
Completed the front end UI skeleton	Eshwar	Completed	
Started implementing card swipe functionality	Eshwar	In progress	
MLS data collection	Dinesh	In progress	
AUTH implementation	Daniel	In Progress	

2. What feedback has the team received?

From Whom	Feedback	Next Steps
No feedback this week, did not meet with client		

3. Are any resources needed? If so, what?

None needed

Plans for Next Week

What are your plans for this next week?

Task	Task Lead	Notes
Implement card swipe functionality	Eshwar	
Web scrape property listings	Dinesh	
Build an LLM for rent prediction	Daniel	
Enhance CI/CD pipeline	Arjun	

Week 5:

Status Report

Last week, our team made steady progress on the real estate platform, developing two interfaces: a web application for realtors and a mobile app for investors. Realtors can now view listings and locate them on maps using the Google Maps API, and the dark mode setting works well. For investors, we implemented a swipe card feature, letting them swipe left for “no” and right for “yes” on listings, with a saved listings section to review their choices.

We also have a Python script that scrapes the web to find real-time listing data from various real estate websites like realtor.com and zillow.com. The script successfully pulled in fresh data on prices, locations, and property details. All this data is currently stored in the firebase database and these listings are used by our apps to display them to the realtors and the investors. The real-time updates will help realtors and investors make informed decisions quickly. Additionally, user authentication is working smoothly, allowing realtors and investors to securely log in and access their personalized dashboards in both applications.

This week, our goal is to add a cash flow section for investors to analyze potential returns on listings, including estimated rental income and expenses. We also plan to refine the UI based on user testing, focusing on clarity and ease of use, and improve the mobile app’s performance by optimizing data loading. The team will test both applications thoroughly to catch any remaining issues, ensure the Google Maps API handles high traffic smoothly, and explore adding filters for realtors to narrow down listings by price or location.

Current Status

1. What did the team work on this past week?

Task	Task Lead	Status	Notes
CI/CD Complete	Arjun	Done	Took a very long time and had help from Dinesh to find a solution
Project Report - Related Work	Eshwar	Done	
Firestore data collection	Daniel	Done	Will run every two days, and will get more volume in the future.
Realtor Dashboard	Dinesh	Done	

2. What feedback has the team received?

From Whom	Feedback	Next Steps
Client	Greg gave us an overview of how cash flow works.	Will implement in the following sprint.

3. Are any resources needed? If so, what?

N/A

Plans for Next Week

What are your plans for this next week?

Task	Task Lead	Notes
Adding the listings to the realtor UI	Dinesh	
Adding the Clients section	Dinesh	
Adding the Reports section	Arjun	
Refining the overall UI	Eshwar	
Adding realtor recommendations section for investors	Daniel	

Week 6:

Status Report

This past week we worked on bringing in the segmented parts of our application. This primarily included the different views for the realtor and client apps as well as combining the different views that we developed separately for mobile and web. There are different user interface and user experience demands for each platform and since we developed them somewhat separately we are working on integrating them together. For this week we plan to make all aspects of the application fully functional and then save the minor polishing for the last sprint. One of the main things we need to do is refactor some of our code to be compatible with other parts. This should also aid us in creating a unified user interface throughout all platforms. There is one aspect of our software that lies outside of any of the aforementioned platforms. We want to have a system that allows for

realtors to invite users to the platforms via a code. This system needs to be managed by our database and be shared via either the realtor sending it manually or an automated email system. One thing we may save for the last sprint is generating links that can be clicked and then allowing the users to connect in that manner.

Current Status

- What did the team work on this past week?

Task	Task Lead	Status	Notes
Visual Swiping Indicator	Daniel	Complete	Open to feedback
Home search and details page	Dinesh	Almost complete	None
Realtor mobile UI implementation	Dinesh	Complete	Implemented framework (sidebar, drawer, etc), still need to make the pages mobile compatible
Develop ML model for rent price approximation	Dinesh	Complete	Within 10% range of error
Implemented unit tests for multiple functions	Arjun	Complete	
Client Management UI	Eshwar	Complete	Needs to be ported to mobile

- What feedback has the team received?

From Whom	Feedback	Next Steps
Greg (Client)	Thanks for the update. Looks like you're putting in some hard work!	Send Greg a demo link by the end of this week.

- Are any resources needed? If so, what?

No resources needed at the moment.

Plans for Next Week

What are your plans for this next week?

Task	Task Lead	Notes
Finish calculators core function	Arjun	
Finish Refactors	Daniel	
Finish home search core function	Dinesh	
Port Client Management to Mobile	Eshwar	

Week 7:

Status Report

This sprint, the team completed key features for the app. We implemented a machine learning model that predicts house rent prices with an accuracy of plus or minus 10 percent. The team replaced placeholder pages from the last sprint, including the dashboard, clients, and home search pages, with fully functional versions. The client page now displays a realtor's client types, while the dashboard shows notifications, property updates, and sold houses to track monthly quotas. We added the IMAP protocol to send emails to investors, including app invitations and invitation code to link the realtor with the investor. The home search page uses the Google Maps API to show all houses in College Station, with clickable images linking to property details. For investors, we built a signup process using the realtor's invitation code to connect both parties. We also added emojis, like a cross for a left swipe to reject a property and a heart for a right swipe to approve, plus a view of realtor-sent properties.

Challenges included tuning the ML model, which required extra time to reach the target accuracy. We also had a lot of CORS issues while displaying the property images that we scraped from the web. These issues happened because of different urls used for development and actual debugging. We also ran into problems with the IMAP protocol which we were able to solve by the end of the sprint.

For this next sprint, the team plans to resolve all CORS issues to ensure smooth cross-origin requests. We will implement a feature letting realtors send properties to clients, with visibility into which properties clients liked or disliked. Realtors will also gain the ability to customize cash flow analysis when sharing properties with investors. Our goal is to stabilize these features and test them thoroughly by the sprint's end.

Current Status

- What did the team work on this past week?

Task	Task Lead	Status	Notes

Finished all calculators	Arjun	Done	
Completed the Clients page	Eshwar	Done	
Implemented the Dashboard page	Eshwar	Done	
Swipe Gestures	Daniel	Done	
Tor Scraping	Daniel	Done	
Realtor send flow	Daniel	WIP	
Cashflow	Dinesh	Done	

2. What feedback has the team received?

From Whom	Feedback	Next Steps
Friend at Home	Needs more simplified UI	Simplify UI

3. Are any resources needed? If so, what?

N/A

Plans for Next Week

What are your plans for this next week?

Task	Task Lead	Notes
Implement more filters	Arjun	
Fix UI + State Issues	Daniel	
Make the dashboard better	Eshwar	
Implement various searching	Dinesh	

Week 8:

Status Report

Last week, our team focused on redesigning the home page and updating the login page to align better with our theme. We also made improvements to the home search screen, enhancing the user experience. Additionally, we started integrating the property details screen from the web version into the app. To maintain consistency across the project, we worked on defining a central theme that will guide the design of the entire app and server.

However, we faced a challenge with screen sizing. While the design worked well on the web, it did not display correctly on mobile devices. To address this, we decided to allocate more time to develop separate layouts for web and mobile to ensure proper functionality on both platforms.

Looking ahead to this week, our primary goal is to enable the feature that allows realtors to send property listings to investors. We will also work on adding investor tags, allowing realtors to apply custom filters. These features will enhance the app's usability and cater to the specific needs of our users.

Current Status

- What did the team work on this past week?

Task	Task Lead	Status	Notes
Implement the Landing Page	Eshwar	In progress	
Host the IMAP server on Vercel	Eshwar	Done	
Adding widget with average rent price graph	Arjun	In Progress	
CDN + Dashboard refactor	Daniel	In Progress	
Fix performance issues in home search page and revamp UI to make efficient use of space	Dinesh	Done	
Reimplement Client management UI to track flow from lead to client	Dinesh	In progress	

- What feedback has the team received?

From Whom	Feedback	Next Steps
Greg	Need to iron out value propositions for realtors. Fix realtor -> investor flow.	UI refactor and some feature implementation

Greg	“That's great. Looks good” after sending him status update with how home search and client management is looking	Follow up with him friday with a demo link

3. Are any resources needed? If so, what?

Plans for Next Week

What are your plans for this next week?

Task	Task Lead	Notes
Redefining the Login Page	Eshwar	We have to change the theme of the login page to better match our theme
Make custom invitation messages	Eshwar	The realtor should be able to send custom invitations to their clients
Finish the average rent price widget	Arjun	
Setup CDN + Refactor Dashboard	Daniel	Need cdn to avoid massive firestore calls
Finish reimplementing client page and fix minor UI issues around the app	Dinesh	

Week 9:

Status Report

We have nearly gotten an entirely functional user flow. This past sprint we implemented many features for managing clients as well as sending properties to clients from the realtor home search. We implemented the ability to filter properties on home search and have persisting filters and data throughout. We also added detailed cash flow analysis along with quick view cashflow badges which helps our app be more tuned to investors. There are also a significant amount of various UI touch ups and small features implemented throughout the application.

One reach goal that we have is having a more robust notification system. This would allow realtors to know when clients have matched with a property they sent. It would be somewhat trivial to include this information on the realtor dashboard. However the “reach” aspect of this goal would be to somehow implement push notifications into the mobile application. While we have full cross platform functionality between mobile and web, as well as responsive design, it would be nice to fully lean into the UX of the mobile application. In terms of realistic improvements it all comes down to UI improvements and performance improvements for retrieving data.

Current Status

1. What did the team work on this past week?

Task	Task Lead	Status	Notes
Landing Page + UI for Login	Eshwar	Complete	
Realtor Dashboard + Three Widgets	Daniel	Complete	
Investor Tags	Dinesh	Complete	
Added Buyer netsheet calculator	Arjun	Complete	May be edited to add more features
Revamped investor UI with major UI fixes to look better and make more sense to user	Dinesh	Complete	
Implemented cash flow analysis for each property with breakdown and ability to edit realtor defaults	Dinesh	Complete	
Added a modal for viewing and managing client details	Dinesh	Complete	

2. What feedback has the team received?

From Whom	Feedback	Next Steps
Ourselves	Needs more notifications	Add a widget to the dashboard which includes the matched properties.

3. Are any resources needed? If so, what?

No resources needed.

Plans for Next Week

What are your plans for this next week?

Task	Task Lead	Notes
Testing coverage	Daniel	
Search + Refresh Bugs	Daniel	
Unit Testing	Arjun	
UI touch-ups	Eshwar	
Testing for home search client page	Dinesh	
Fix minor UI issues as I see them	Dinesh	

Week 10:

Status Report

Last week, we completed all core functionalities for the app. Realtors can now view and edit cash flow details, invite clients, and send property listings. Realtors can also create tags for their investors. These tags can be used to send properties to multiple investors at once making it easier for the realtor to send properties to each of his clients. They also have the ability to see which properties their clients liked or disliked. Investors can view listings sent by their realtor and swipe to like or dislike properties. We have also added the additional ability to change the theme of the whole app. The user should be able to change into dark mode or light mode, whichever one they prefer. We also completed a lot of small requests like calculators or widgets that the mentor wanted.

We also began developing a browser extension that adds a cash flow widget to Zillow's property pages. When a user clicks on a listing, they will see a small embedded tool that shows monthly cash flow projections based on our model.

One of the main challenges we faced was related to data reads. Each time the home search page reloads, the app pulls 200 to 300 properties, which creates high read activity on Firebase. We are currently working on a way to reduce these reads and improve performance.

This week, our focus is on refining this extension, optimizing data reads, and testing additional features that could enhance user experience. Overall, we have completed all the core functionality that we wanted to finish from day one. From now on, we are mainly focusing on just getting any additional features that our mentor mentioned would be useful.

Current Status

1. What did the team work on this past week?

Task	Task Lead	Status	Notes
Extension	Daniel	WIP	Likely won't be done
Login and Register validation	Eshwar	Done	
Setup page route protection	Eshwar	WIP	
Added dark mode	Dinesh	Done	
Investor personalized cash flow estimate	Dinesh	Done	
Revamped client management	Dinesh	Done	
Buyer net sheet calculator	Arjun	Complete	
Capstone Poster Draft	Arjun	Complete	
Capstone Report Resubmissions	Arjun	WIP	Will be ongoing till all are at 20/20
Descriptions for Calculators	Arjun	WIP	

2. What feedback has the team received?

From Whom	Feedback	Next Steps
Investors	Looks good, but be able to see more data about cash flow, add a leverage slider.	We won't add any new features. It's too late in development.

3. Are any resources needed? If so, what?

N/A

Plans for Next Week

What are your plans for this next week?

Task	Task Lead	Notes
Testing + Extension	Daniel	
Creating a Welcome page for mobile apk	Eshwar	

Testing	Dinesh	
Testing	Arjun	
Finish Calculator description	Arjun	

Week 11:

Status Report

Write a one-third to one-half page status report describing the team's overall progress and challenges faced last week. Also discuss the team's goals and plan for this week.

Last week, our team focused on testing and debugging the application. We wrote tests for several core files, including components tied to the cash flow analysis, property filtering, and user interaction features. These tests helped us catch logic errors and fix bugs that would have affected the user experience. We ran these tests regularly to make sure changes in one part of the code didn't break another. We also checked for edge cases that could cause problems during real use. By the end of the week, the app was more stable, and many of the main features performed as expected during test runs.

One of our biggest challenges was setting up tests that reflected how users interact with the platform. This was important for both the realtor and investor dashboards. We had to make sure the test data matched what users would actually see. Another challenge was tracking down bugs that only appeared during integration. These issues were sometimes hard to find because they didn't come from a single file, but from how files worked together. Debugging these problems took time and coordination between team members.

Our main goal for this week is to reach full test coverage across the app. We will continue writing and refining unit and integration tests for all remaining files. We also plan to review the current tests to make sure they are reliable and easy to maintain. In addition, we will begin testing full user flows to catch any issues that don't show up in isolated cases. By the end of the week, we expect the app to be fully tested, with each feature checked for accuracy and consistency. This will put us in a good position to move into the final testing and review phases.

Current Status

- What did the team work on this past week?

Task	Task Lead	Status	Notes
Report Revisions	Arjun	WIP	
Added better error handling for login, signup and password	Eshwar	Completed	
Unit testing coverage	Daniel	WIP	

Implemented Final Agents proposed by user studies	Dinesh	WIP	

2. What feedback has the team received?

From Whom	Feedback	Next Steps
Investors	Need more options for leverage and appreciation	
Realtors	Add appreciation and add more cash flow details like return on investment, etc.	

3. Are any resources needed? If so, what?

N/A

Plans for Next Week

What are your plans for this next week?

Task	Task Lead	Notes
Report revisions	Arjun	
Test cases for each file	Daniel	
Adding comments to the code	Eshwar	

A.5 Budget

In order to ensure proper development and deployment of our platform, we require various software tools and services. Below is a table outlining the required software and hardware components, specifying which ones are in place and which ones are to be purchased.

Item	Purpose	Status	Cost	Purchase Link	Need By
Firebase	Database, Authentication, Hosting	Accessible / Needs Purchase	Free (Spark Plan) / Pay-as-you-go (Blaze Plan) 50k reads per day is sufficient About \$0.12/month	Firebase Pricing	3/4/2025 (Depends if we go over)
Cloud Computing (GCP)	Hosting backend services and data storage	Needs Purchase	50k Reads per day sufficient Cloud run is \$0.00002160 / vCPU-second beyond free tier We get \$300 in free credits	GCP Pricing	3/18/2025
Rentometer	Rental Estimates	Acquired	\$99/1000 credits	Rentometer API Access	3/25/2025

Previously we needed to pay massive amounts to various MLS systems to obtain the MLS data required. However, the zillow robots.txt actually specifies that this data can be obtained from them for free.

```
# Access to and use of Zillow.com is governed by our Terms of Use. See http://www.zillow.com/corp/Terms.htm

User-agent: *

Allow: /homes/for_sale/$
Allow: /homes/for_sale/fore_lt/$
Allow: /homes/fsbo/$
Allow: /homes/for_sale/1_open/$
Allow: /homes/new_homes/$
Allow: /homes/coming_soon/$
Allow: /homes/for_rent/$
Allow: /homes/for_rent/condo,apartment_duplex_type/$
Allow: /homes/for_rent/house,townhouse_type/$
Allow: /homes/recently_sold/$
Allow: /homes/apartment-communities/$

Allow: /homes/for_sale/*_p/$
Allow: /homes/for_sale/fore_lt/*_p/$
Allow: /homes/fsbo/*_p/$
Allow: /homes/for_sale/1_open/*_p/$
Allow: /homes/new_homes/*_p/$
Allow: /homes/coming_soon/*_p/$
Allow: /homes/for_rent/*_p/$
Allow: /homes/for_rent/condo,apartment_duplex_type/*_p/$
Allow: /homes/for_rent/house,townhouse_type/*_p/$
Allow: /homes/recently_sold/*_p/$
Allow: /homes/apartment-communities/*_p/$
```

Zillow.com Robots.txt

Appendix B: Implementation Details

1. System Architecture

Our application follows a modular client-server architecture using Flutter for the frontend and Firebase as the backend. The hierarchy consists of three major layers:

A. Frontend Layer (Flutter)

- Built using **Flutter 3.24.0**, enabling a responsive UI for both mobile and web.
- Navigation is handled by **GoRouter**, and the state management follows the **Provider + ChangeNotifier** pattern
- Widget structure is broken into reusable components for scalability (e.g., **PropertyCard**, **ClientDetailDrawer**, **FilterDrawer**, etc.).

B. Backend Layer (Firebase)

- **Cloud Firestore** is the primary NoSQL database storing all dynamic data.
- **Firebase Auth** (Email Link Authentication) handles user login for realtors and clients.
- **Firebase Functions** Used for sending invites or sending alerts. Handles all of our applications mailing features

C. Search Layer (Algolia)

- **Algolia InstantSearch** provides fast address-based lookup. This is connected to the Algolia firebase extension to index our listings and clients so we can provide lookup.
- Integrated using **algolia_helper_flutter** and configured with filters to allow partial address matching and swiping-to-detail transitions.

2. Component Overview

A. Realtor Dashboard

- Allows the realtor to quickly access information about pinned clients. This widget includes a search bar to add clients and uses Algolia instant search.
- **ClientDetailsDrawer** opens when selecting a client and includes property interactions grouped by status (liked, disliked, sent, matched). It also includes all the notes that the investor sent.
- Has a widget for tracking client activity so that the realtor can see how long clients are spending on the app and what properties they are looking at.

B. Realtor Client Management

- Allows realtors to see all the leads, qualified leads (contact was made and info was gathered), and clients.
- Realtor is able to add new lead with just name and email and is able to take notes
- Realtor is able to move a lead to qualified lead
- Qualified leads can be promoted to a client which creates an account for the client and sends them the credentials.
- Realtor is able to create some predefined tags to assign to clients and is also able to assign those tags to the client.
- Realtor is able to search a client up by name

C. Calculators

- Clients and Realtors are able to see 4 different calculators: PITI calc, Affordability calc, Rental Property Calc, and Buyer Net Sheet calc
- All pages are made mobile friendly and the navbar switches to a dropdown menu on smaller screens

D. Home search

- Realtors and clients are able to view a map with all the properties marked on it with their price.
- Users are able to search the properties using address, MLS ID, or neighborhood
- Users can see all the listings with property cards that give a quick intro to the property including its cashflow, status, address, price, beds, baths, and square feet.
- Users are able to filter the properties using quick selectors and a filter drawer.
- Realtors are able to send properties to investors by looking up tags to bulk send or searching up names.
- Investors are able to send a note about the property to the realtor
- Users are able to see a cash flow analysis for each property including its breakdown, they are also able to edit the cash flow

E. Client Swiping Feed

- Investors are able to swipe left or right on the property or click like/dislike to indicate to the realtor if they like or dislike a property.
- Investors can click a button to view more information about the property and its cash flow analysis
- Investors can toggle between a feed of properties that fit their filter and a feed of properties that were sent to them by the realtor.
- Utilized a widget from Flutter called **CardSwiper** to allow for swiping to occur. Simply loads in the information from Firebase.

F. Liked and Disliked properties

- User can see the properties that they previously swiped left on and disliked and are able to move it to liked
- Users can see the properties that they swiped right on and liked and are able to move it to disliked.

G. MLS Scraper and Rent Price

- Utilized the various free endpoints provided by *zillow.com*, *realtor.com*, and *redfin.com* to scrape the home details of properties newly added to the MLS.
 - Automated a Github actions script to run every other day (the script itself searches for properties added to the MLS in the past three days).
 - Basic HTTP request sent from Python.
 - Data directly inserted into the Firestore after being cleaned and filtered.
 - Data is not included if it is missing key fields such as address or estimation.
 - Data is not reinserted if matching or similar listing already exists (two properties of the same address for example).
- Using the property data, the rental estimate data is obtained from *Rentometer*. This information is then stored with the matching record in Firestore.

3. Tools and Libraries Used

Tool / Library	Purpose
Flutter	Cross-platform UI development
Firebase Auth	Handle user passwords and identity
Firestore	NoSQL backend for listings, clients, notes, user information, and other data our app needs to store in the backend
Algolia	Instant search for looking up listings and clients
Provider	State management
GoRouter	Declarative routing
flutter_card_swiper	Swiping interface
flutter_cache_manager	Picture and image caching
intl	Date/time formatting
Material	Certain UI components
google_maps_flutter	Showing maps on the application
shared_preferences	Enabling persistence on refresh

4. Screenshots

The screenshot shows the Firebase Functions dashboard for the project "realest". It lists five functions:

Function	Trigger	Version	Requests (24 hrs)	Min / Max Instances	Timeout
sendInviteEmail us-central1	[Unknown trigger]	v2	0	0 / 100	1m
ext-firebase-al... us-central1	HTTP Request https://us-central1-realest-3a0d2.cloudfunctions.net/...	v1	0	0 / -	9m
ext-firebase-al... us-central1	document.write investors/{documentID}	v1	32	0 / 3000	1m
ext-firebase-al... us-central1	HTTP Request https://us-central1-realest-3a0d2.cloudfunctions.net/...	v1	0	0 / -	9m
ext-firebase-al... us-central1	document.write listings/{documentID}	v1	7	0 / 3000	1m

Items per page: 25 1 – 5 of 5

The screenshot shows the Cloud Firestore console for the project "realest". It displays a document in the "cashflow_analysis" collection with the ID "7010957506". The document contains the following fields:

Field	Type	Value
averageAnnualReturn	number	-4150.983219818841
cashInvested	number	74000
cashOnCashReturnAnnualized	number	-0.2040346341898444
closingCost	number	6000
costToSell	number	0.08
defaultHOA	number	50
downPayment	number	0.2
holdingLength	number	20
insurance	number	0.005
interestRate	number	0.06
loanTerm	number	30
maintenance	number	0.01

SEARCH

- CONFIGURE**
 - Index**
 - Query Suggestions
 - Dictionaries
- OBSERVE**
 - Analytics
 - A/B Testing
- ENHANCE**
 - Rules
 - AI Synonyms
 - Re-Ranking
 - Query Categorization
 - Personalization

Application Index

RealEst investors

Index

records 44 # events 0 avg. record size 198.75B

Enhance search relevance! Send event data to unlock more value from your application.

New... Add records Manage index

Browse Configuration Replicas Search API Logs Stats UI Demos

Search: minecraft

Last 7 days:

Facets	objectID	contactEmail	firstName	lastName	realtorId
1	"bZD2SxpmgxeSJ2vTkFn7"	"selyspyrqyxhkuzrs@poplk.com"	"minecraft"	"steve"	"CYiNkGX0lNaKJkRgMncUvnOxZTC2"

Show more attributes (2)

Summary

Jobs

pull-data (succeeded yesterday in 30s)

Run details

Usage

Workflow file

pull-data

Validate Service Account

Run realest data collection

```

1 • Run python main.py --debug --days 3
12 Processing city: College Station, TX
13 Scraping listings for location: College Station, TX, status: FOR_SALE
14
15 ▲ Invalid listing at index 20:
16 Missing fields: beds, sqft
17 Available data: property_id: 9504330784, status: FOR_SALE, full_street_line: 15420 Arhopulos Rd, street: 15420 Arhopulos Rd, city: College Station, state: TX, zip_code: 77845, list_price: 1195000
18
19 ✅ Successfully fetched 27 FOR_SALE valid listings (1 invalid listings skipped) for College Station, TX
20 Scraping listings for location: College Station, TX, status: SOLD
21
22 🔞 Critical error fetching listings for College Station, TX, status: SOLD: Missing required columns: property_id, full_street_line, street, city, state, zip_code, status, list_price, beds, sqft
23
24 Scraping listings for location: College Station, TX, status: PENDING
25
26 🚀 Successfully fetched 1 PENDING valid listings (0 invalid listings skipped) for College Station, TX
27 🚀 STARTING PROCESSING FOR College Station, TX
28 🔞 Loading existing listings...
29 🌐 Found 397 existing listings in College Station, TX
30 🔞 Processing 24 new listings...
31
32 🔙 Processing Property ID: 7458642542 (1/24)
33 🌐 NEW LISTING DETECTED
34 City: College Station
35 State: TX
36 Initial values: {'status': 'FOR SALE', 'text': 'This is a beautiful 4 bedroom 2 bath home. It is in a wonderful school district. Recently renovated the entire house including granite counter tops, all new cabinets, new stained glass light fixtures. NEW ROOF, new paint inside and out, new insulated garage doors, huge yard...you will love it! Come by and see it today! being sold AS IS.'}
37
38 🔙 Processing Property ID: 9841525075 (2/24)
39 🌐 NEW LISTING DETECTED
40 City: College Station
41 State: TX
42 Initial values: {'status': 'FOR SALE', 'text': 'Like new one story single-family home located in Midtown Reserve. Built in 2022, this Cypress plan offers 1849 square feet of living space, featuring 3 bedrooms and 2 bathrooms, 2 car garage. The property includes a bonus room that can serve as an office/study/hobby room. The interior boasts a spacious living room that seamlessly connects to the kitchen and dining areas. The kitchen is equipped with granite countertops, an eating bar, pantry, gas range, and stainless steel appliances. A full laundry room with a washer and'}

```


Appendix C: User's Manual

This user guide is designed to help you get started with RealEst, a property swiping and client matching application that bridges the gap between property seekers and real estate professionals.

Our Users

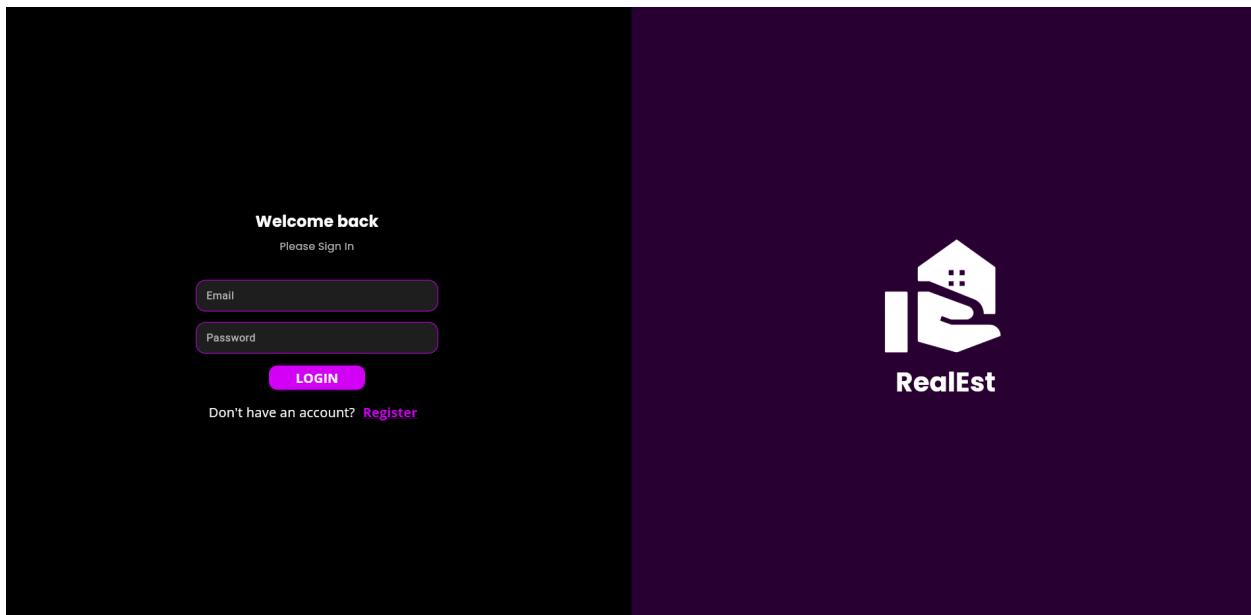
RealEst serves two primary user types:

Property Seekers

Property Seekers are individuals looking to buy or rent properties. They may have specific preferences but struggle to communicate them effectively. Property seekers want to browse properties in an intuitive, engaging way and need to connect with realtors who understand their needs.

Real Estate Professionals

Real Estate Professionals are realtors and agents seeking an understanding of client preferences. They want to showcase properties to the right audience and are looking to build their client base efficiently.



For Property Seekers

Getting Started

Download and Install

Not available on Apple devices, requires APK installation on an Android platform which supports the most recent version of Flutter. Search for "RealEst" and download the application. Alternatively, access the web version at real-est.me (until the hosting service is taken down).

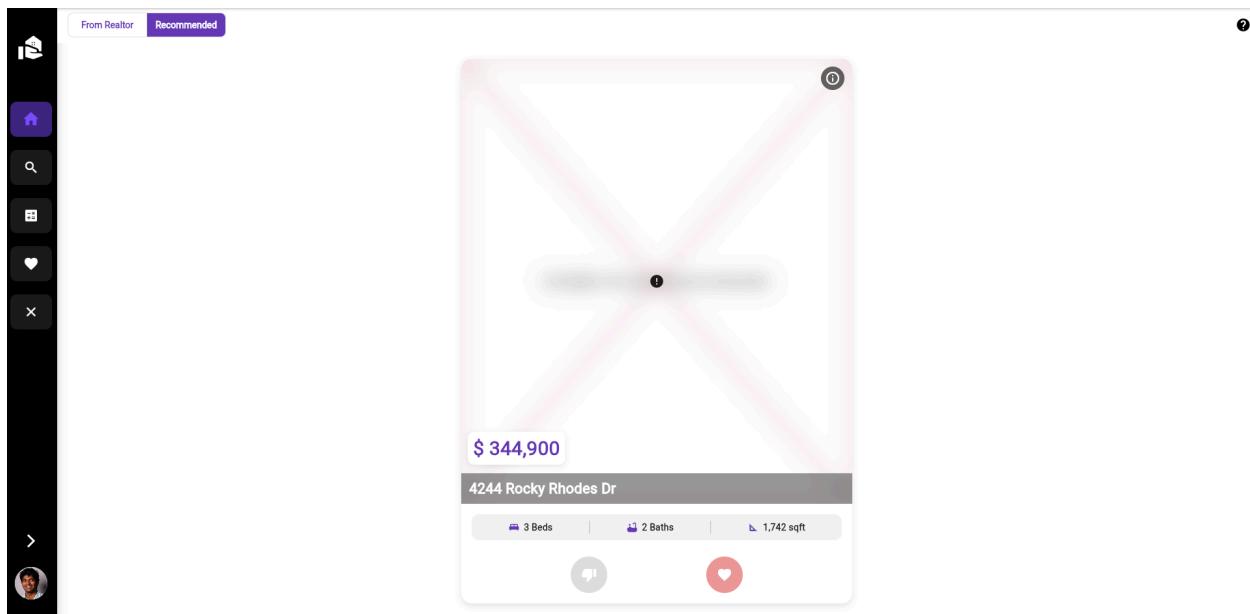
Account Creation

Your realtor should have reached out to you via email. This email should have supplied you with a temporary password. The username is your email.

Using the App

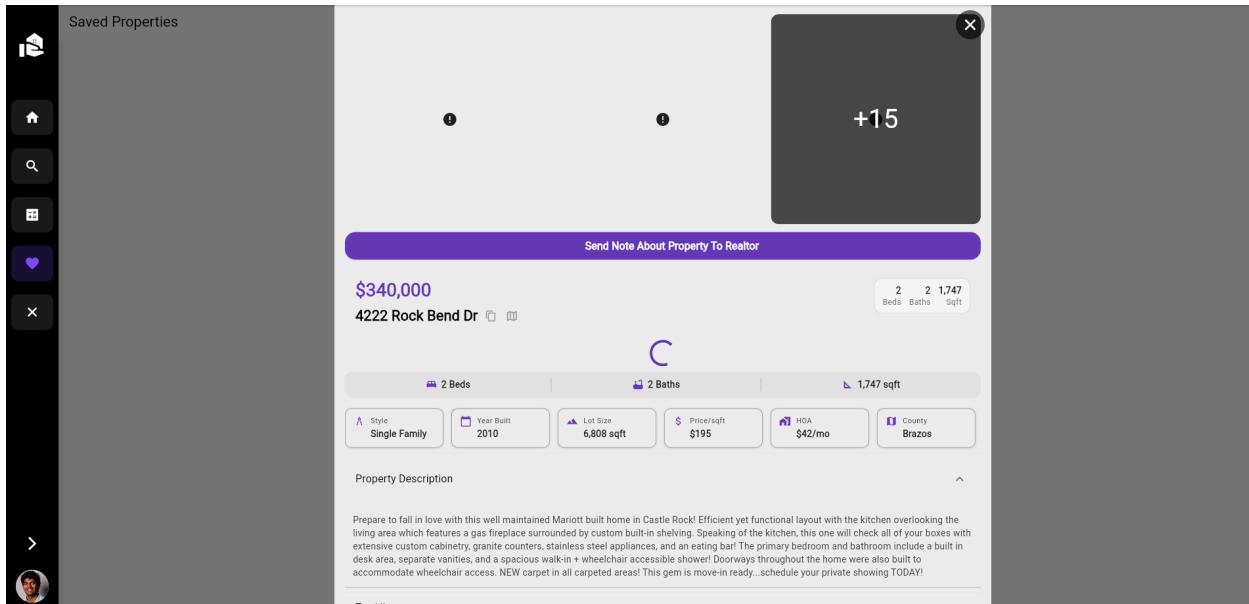
Property Swiping

The main screen displays properties one at a time. Swipe right on properties you like, left on those you don't. Tap on a property card to see detailed information including photos, specifications, and location.



Saved Properties

You can view saved properties on the saved properties tab. This also allows you to send notes about the property to your realtor.



Setting Preferences

Access the preferences menu from your profile.

Matching with Realtors

Your realtor can view which properties you have swiped left or right on. This allows for them to reach out and follow up on any questions you might have. Initiate conversations through the in-app notes system.

Saving Favorites

Properties you swipe right on are saved to your "Saved" list.

For Real Estate Professionals

Getting Started

Download and Install

Same installation process as for investors. Same web options exist as well.

Account Verification

Sign up as a realtor and add relevant realtor details on the setup page.

Using the App

Managing Your Properties

View currently for sale properties on the home search tab. You can send individual clients these properties, or even groups of clients.

The screenshot displays a mobile application interface for managing properties. On the left is a map of a residential area with numerous pins indicating property locations, each labeled with a price. To the right of the map is a grid of nine property cards, each showing a thumbnail image, the address, price, and some basic details like square footage and bedrooms/bathrooms. The top of the screen features a search bar and several filters: '\$ 100000 - \$100...', '2+bd / 2+ba', '2 selected', and a 'More' button. The bottom of the screen shows standard navigation icons.

Address	Price	Type	Bedrooms	Bathrooms	Sqft
1725 Harvey Mitchell Pkwy S Apt 2124	\$209,900	For Sale	2	2	1,052 sqft
1725 Harvey Mitchell Pkwy S Apt 2111	\$255,000	For Sale	2	2	1,080 sqft
2400 Longmire Dr Apt 403	\$240,000	For Sale	3	2	1,512 sqft
906 Hollerman Dr	\$284,900	For Sale	2	2	1,408 sqft
905 Coffee Mill Ln	\$292,505	For Sale	3	2	1,390 sqft
914 Coffee Mill Ln	\$302,075	For Sale	3	2	1,489 sqft
6203 Trado Dr	\$309,280	For Sale	3	2	1,508 sqft
2509 Warkworth Ln	\$449,900	For Sale	3	2	2,100 sqft
1110 Chesapeake Ln	\$352,500	For Sale	4	2	1,760 sqft

Client Management

Add potential clients to the management dashboard. Review client preference notes to understand their needs by viewing their profile on the dashboard. Initiate contact when you find a promising match. Create tags to group clients together for sending properties.

\$340,000
4222 Rock Bend Dr

2 Beds 2 Baths 1,747 sqft

Style: Single Family Year Built: 2010 Lot Size: 6,808 sqft Price/sqft: \$195 HOA: \$42/mo County: Brazos

Property Description:

Prepare to fall in love with this well maintained Mariott built home in Castle Rock! Efficient yet functional layout with the kitchen overlooking the living area which features a gas fireplace surrounded by custom built-in shelving. Speaking of the kitchen, this one will check all of your boxes with extensive custom cabinetry, granite counters, stainless steel appliances, and an eating bar! The primary bedroom and bathroom include a built in desk area, separate vanities, and a spacious walk-in + wheelchair accessible shower! Doorways throughout the home were also built to accommodate wheelchair access. NEW carpet in all carpeted areas! This gem is move-in ready... schedule your private showing TODAY!

Tax History

Additional Property Details

Agent & Office Information

Map showing the location of 4222 Rock Bend Dr. in College Station, Texas.

For Both

Calculators

Cashflow calculators can be accessed to generate insights about a property or location.

Calculators

PITI Calculator

Affordability Calculator
Rental Property Calculator

PITI Calculator

Home Price (\$)

Down Payment (\$)

Interest Rate (%)

Loan Term (Years)

Annual Property Tax (\$)

Monthly Insurance (\$)

Calculate

Please enter valid values to calculate your payment.