MSc in Computer Science

Interim Report

Team 2 – H.M.E (Housing Made Easy)

Charlotte Martin	C17502023
Fenna Kadir	D21125162
Nishant Kapur	D22124271
Stephen Anokye	D22124341
Sean Lynch	C18357081
Oisín Fitzpatrick	C17448992

1 TABLE OF CONTENTS

2 FIGURES

3 INTRODUCTION

The purpose of this project is to create an easy-to-use, aesthetically pleasing rental listings application, thus the name “Housing Made Easy” (HME). The application will provide a range of functionalities, including but not limited to, a comprehensive search experience, applying and managing rent applications, and property browsing.

The resulting application aims to reduce the anxiety of prospective tenants when searching for accommodation by streamlining the process. By analysing competing rental listing websites, the initial key functionalities can be obtained from user reviews.

Beyond the property listing information provided from Zillow, additional information from different data sources, such as coordinate data from Google Maps, and various crime data from Open Baltimore will be integrated into the project. This data will be used to create new attributes, consisting of nearby property service information (entertainment, transportation, retail, fitness, entertainment, and emergency service information) and nearby crime report count information. Alongside this, user profiles will be constructed by retaining user IDs, saved searches, and favourited property data.

The combination of this data will enable users to filter properties, getting a measure of lifestyle suitability in different regions of Baltimore, Maryland. Each property will also feature property scores designed to provide users with information about the available services and the crime safety levels in their respective areas. Additionally, a recommendation system will be incorporated into the project, and will use the collected data to make property recommendations to users via content-based filtering.

The MVP of HME is to create an intuitive and easy to use property rental website, that displays standard and novel property details to users more clearly than other existing websites, and to provide users with property suggestions using a built-in and thoroughly evaluated recommendation system.

4 USER SCENARIO

4.1 IDENTIFYING TARGET USERS

One of the most pervasive issues within Ireland is the housing crisis. The pressure on people to find affordable and suitable homes is overwhelming.

According to Tom Clonan in a Seanad debate in September 2022, "*... the housing crisis and the cost-of-living crisis affects everyone, it has a more significant impact on young people from socially disadvantaged backgrounds and people who are impaired by medical issues, disabilities or mental health issues; caregivers and young parents are also amongst those seriously affected...*" [Houses of the Oireachtas. (2022, September 28)].

This crisis is causing pressure and stress to accommodation seekers, with the average rent in Dublin reaching €2,000+ [*The RTB Rent index Q1 2023*].

According to the Central Statistics Office (CSO), most 20–30-year-olds rent (Figure 1) with a large proportion being students (Figure 2).

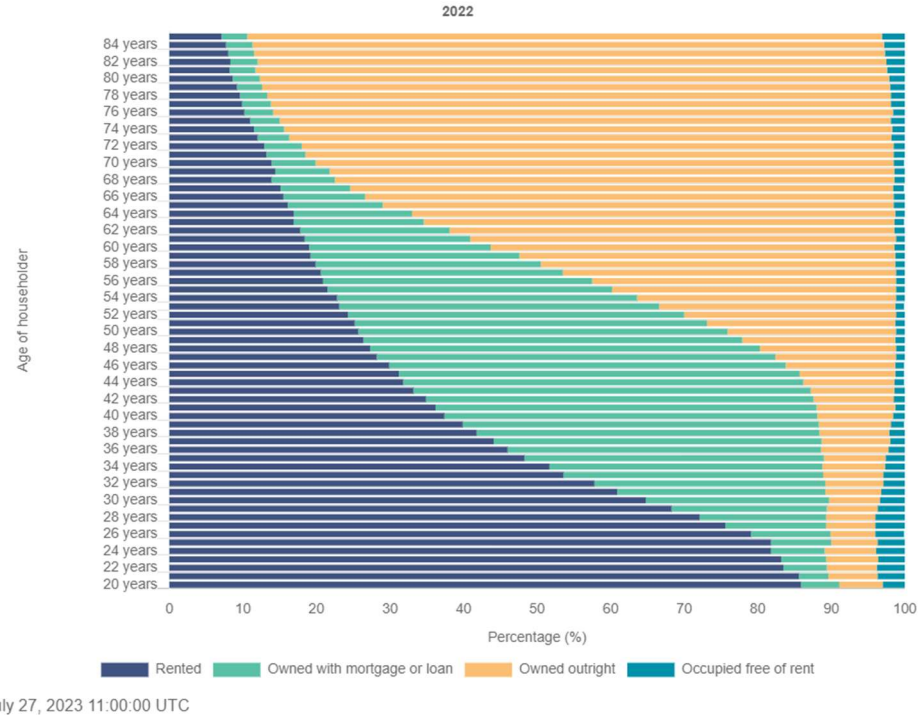


Figure 1 Age demographic vs renting/owning status. [Home ownership and rent - CSO - central statistics office. CSO. (2023, October 20)]

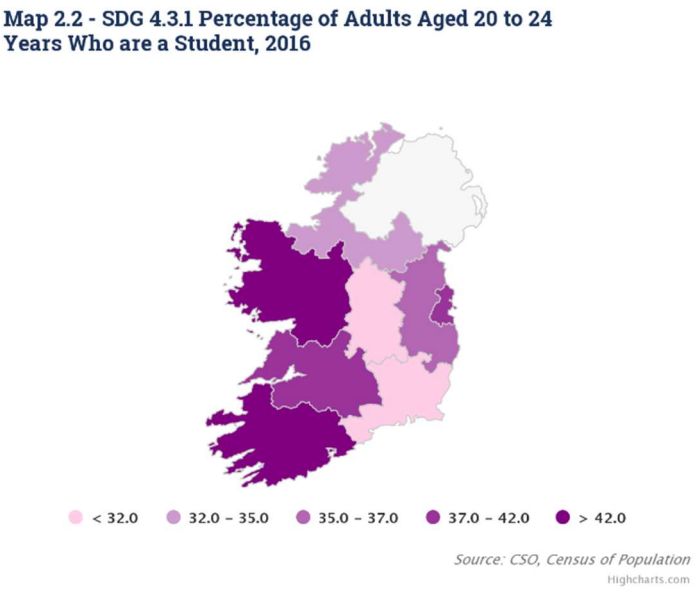


Figure 2 Young adult student population of Ireland [Adult education - CSO - central statistics office. CSO. (2020, August 19)]

In Ireland, of the approximately 10 % of the 250,000 students in third-level education are international, half of which are over the age of 24 [S T U D E N T S - higher education authority. (n.d.)].

According to Eric Konadu's Master Thesis (2021), in which he studied 100 first-year international students, the most important factors for choosing accommodation, ranked in order of preference are:

1. Cleanliness and Sanitation.
2. Cost.
3. Privacy.
4. Security.
5. Space for studying.
6. Bathroom space.
7. Room space.
8. Distance to lectures/ campus.
9. Distance to recreation.

Security is a crucial factor for international students when considering a place to live, ranking 4th overall [Konadu, E. (2021)].

To verify Konadu's findings, a demographics research survey was performed, it demonstrated that security assessment is a significant factor for 18–24-year-olds when browsing properties on listing websites (figure 3).

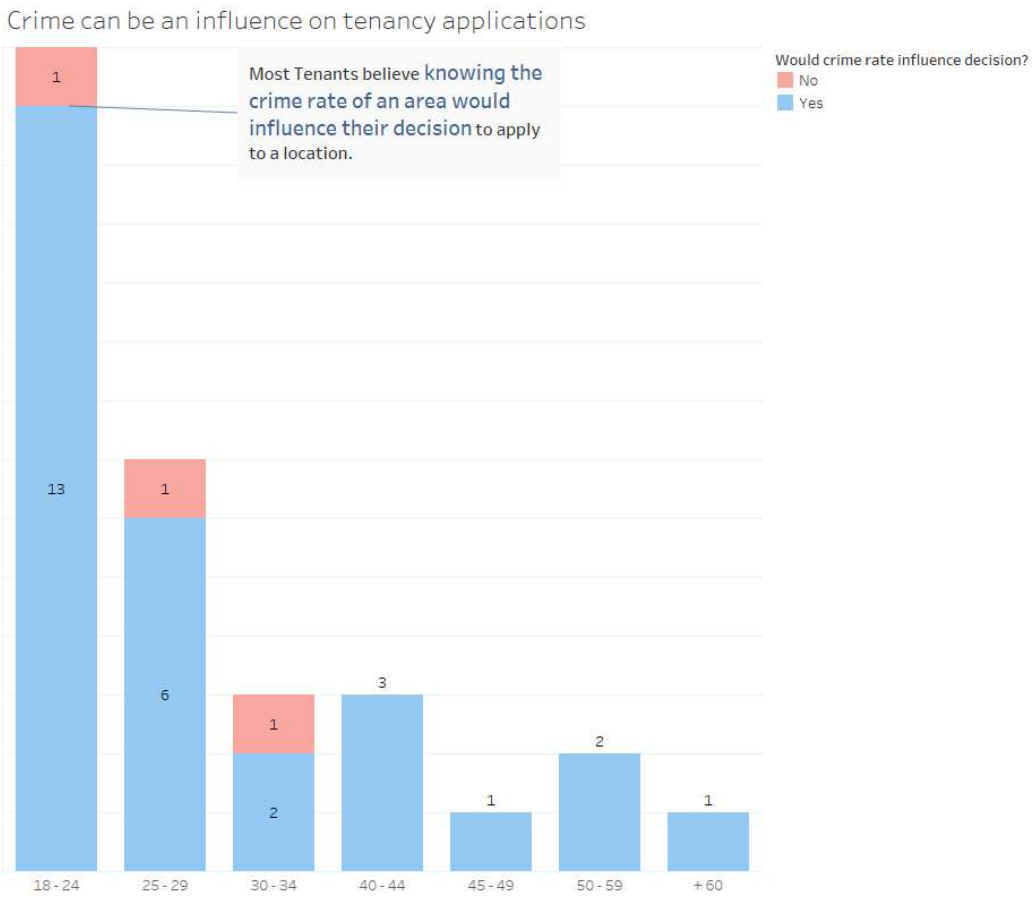


Figure 3 Effect of crime rate on location choice

Respondents were asked what methods they use to assess the safety/security of an area. Previous area knowledge appeared to be the most popular strategy, coinciding with internet to research current local crime data as the second most popular strategy (figure 4).

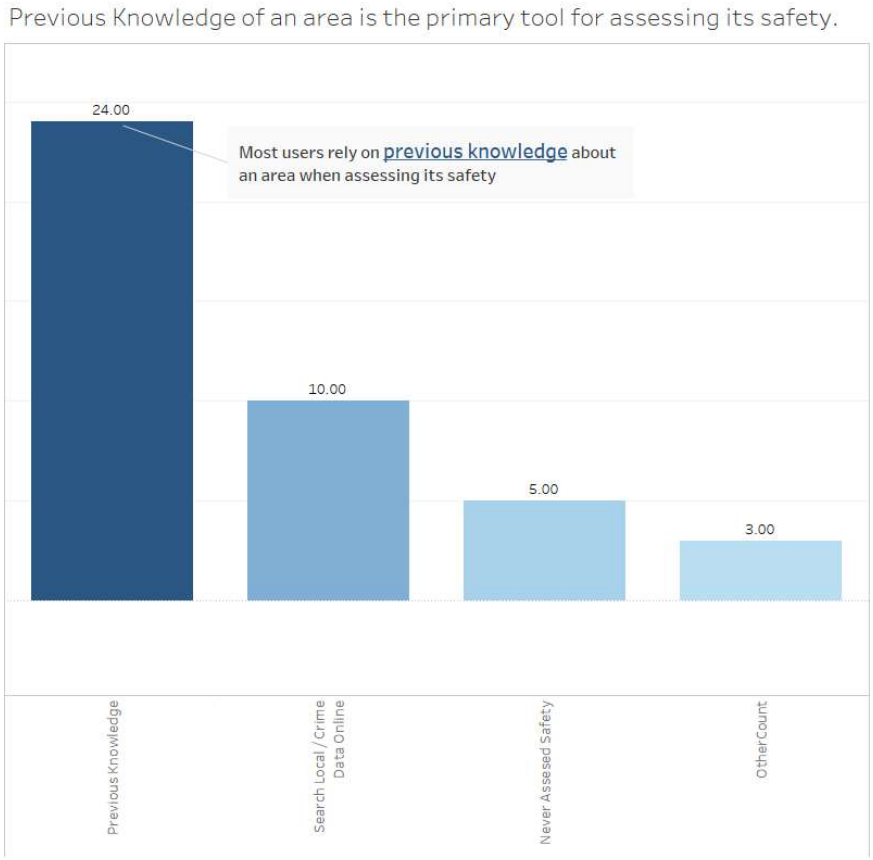


Figure 4 Strategies for Assessing Safety of Listings

4.2 Importance of Target Users

Identifying the primary target/user group is crucial to the project's success. Ensuring system usability to as wide-a-range of people is the core belief behind the philosophy of Universal Design. Equitable use across demographics and ability levels is not the same as designing for universal motivations in a system. The system should be usable and accessible to as many people as possible. By specifying the function and purpose a realistic and achievable project scope can be defined [*What is Universal Design*].

Group Visual's methodology for UX design uses 'Role' as a superset of personas. A user's role is defined as their objective within a specific use context. For HME, the persona's take on the role of prospective tenants, sharing a similar objective and problem, differentiated through their individual use context, motivations, and traits [GroupVisual.io (2018, June 19)].

User Role: Prospective Tenant

The prospective tenant role is concerned with identifying and applying for suitable accommodation. This role involves searching through the properties on a listings website to determine whether they fit the prospective tenants' requirements. This requires assessing a property's price, location, available amenities, and security.

Personas

From the user research, key user personas and motivations can be identified. A focus is put on younger demographics, as they are the primary group of renters. A demographic edge case user is also necessary to account for the outlying user ages. Below there are two personas, Jenni, the primary target, and Andre the edge case user.

These personas combine aspects of the research gathered, i.e., from the demographics survey, and the analysis of similar systems.

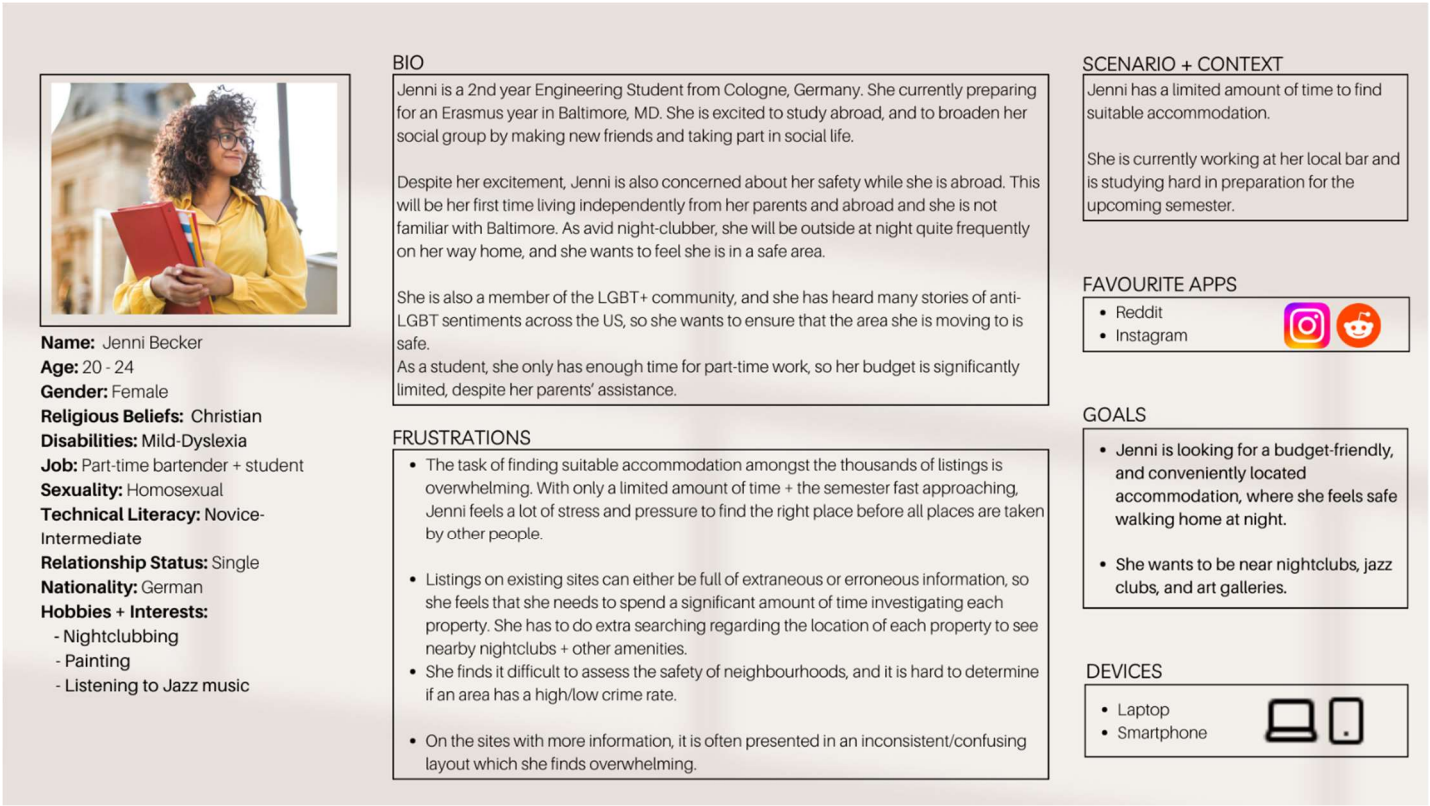


Figure 5 Personna 1, Jenni Becker

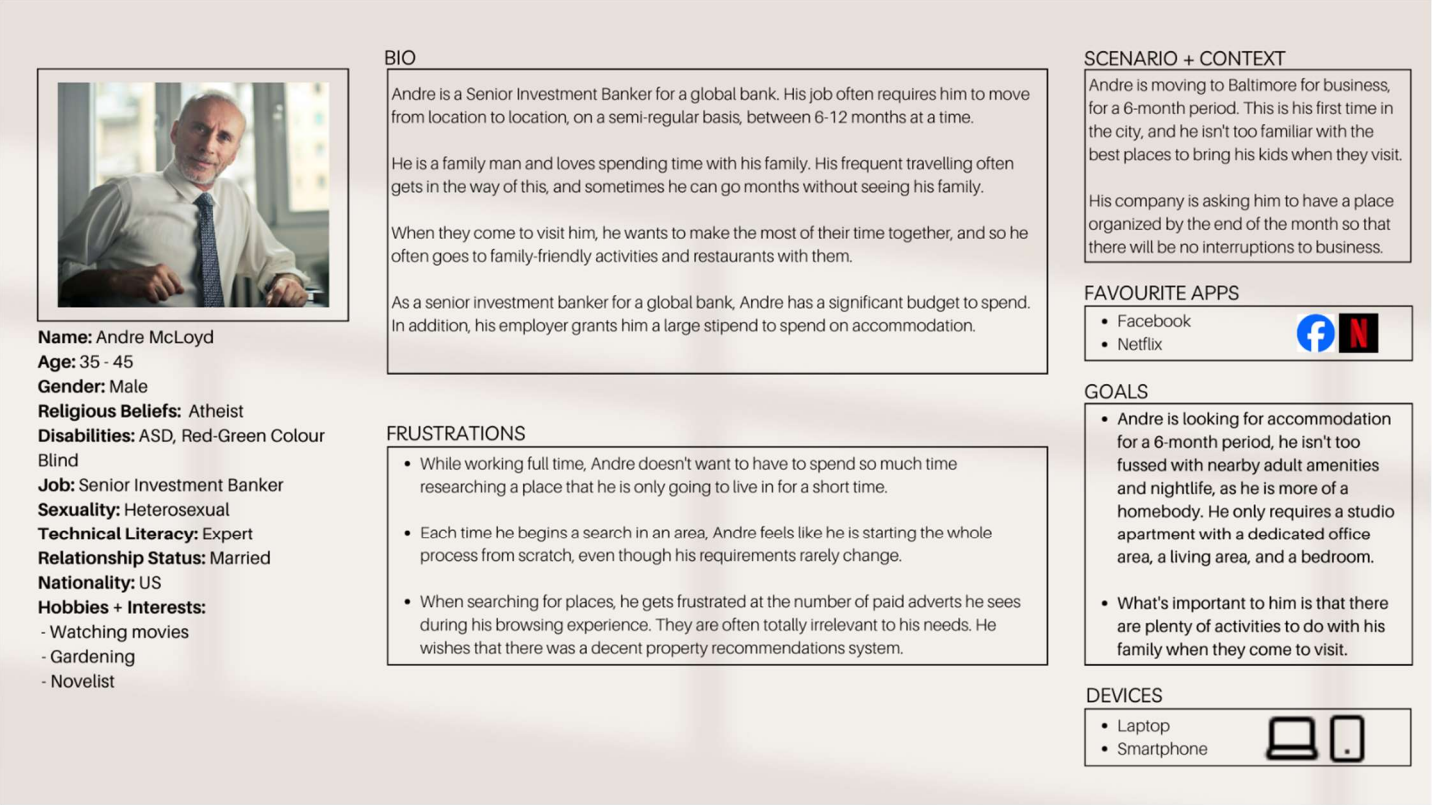


Figure 6 Personna 2, Andre McLoyd

4.3 User Problems

The personas are used to gather an understanding of issues users may encounter.

The demographics survey enabled the team to visualise problems that users face when utilising listing websites:

- Ugly / non-responsive user interfaces.
- Erroneous/ misleading property information.
- Duplicate/ no longer available advertisements.
- Poor search and filtering functionality.
- Irrelevant paid promotions shown before relevant search results.
- Contact and messaging not going anywhere.

An additional objective of the demographics survey was to gauge interest in the implementation of a recommendation system. Most respondents indicated an interest in implementing a property recommendation system.

Of those who were interested in recommendation systems, some specified that it would save them time trawling through search results and they would be more personalized.

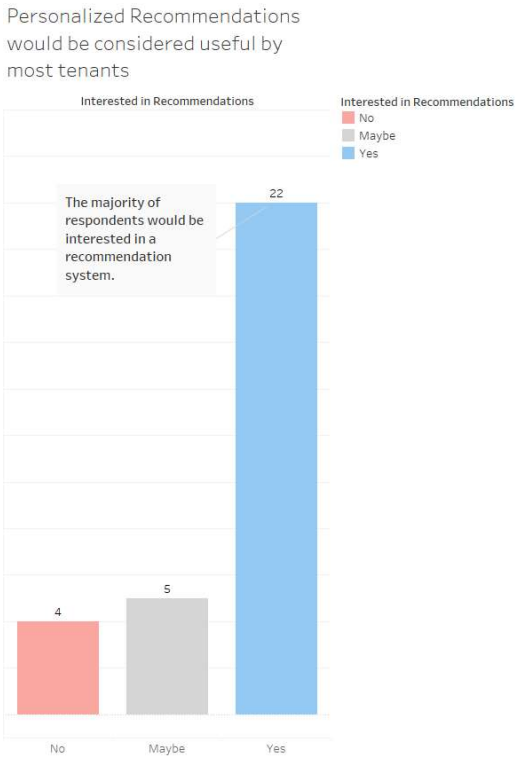


Figure 7 User interest in recommendation system

One disinterested respondent indicated that unless there was a significant number of properties it may be more effective to just filter manually. Another participant stated that recommendation systems typically devolve into paid property advertisements, rather than showing relevant properties to users.

HME aims to solve the following issues: ugly/ confusing user interfaces, personalized and relevant property recommendations, and comprehensive and simplified research processes.

5 TECHNICAL PROBLEM

5.1 CORE TECHNICAL PROBLEM

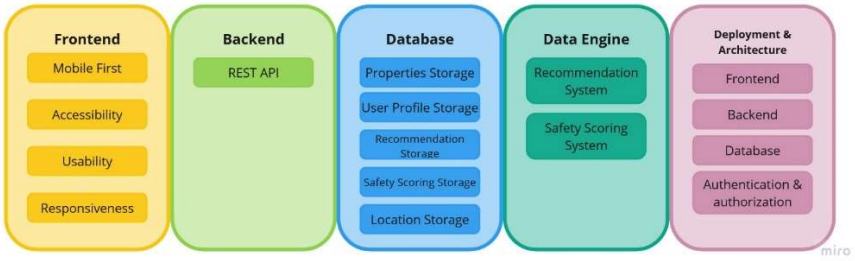


Figure 8: Technical Component Diagram

5.2 USER INTERFACE AND FRONTEND EXPERIENCE

This project centres around improving the user experience offered by existing online listings websites such as Daft.ie, Rent.ie, MyHome.ie, and Zillow.com. The demographics survey conducted suggests that user satisfaction of existing websites is poor, for example, the average rating for Daft.ie being 1.7/5 on Trustpilot.

As mentioned previously, the survey found that the following were cited by respondents as problems they had when using these sites:

- Erroneous and/or irrelevant property information.
- Duplicate and/or no longer available listings.
- Poor search and/or filtering functionality.
- Irrelevant paid promotions over accurate results.
- Poor communication and/or response time in general.

Beyond these complaints, the most cited problems are related to poor user interface design and responsiveness. Interfaces were considered ugly, unresponsive, slow, and confusing with their layouts. This is an area in which HME can improve upon existing solutions. The following subsections detail problems in detail.

Responsiveness

The results of the demographics survey suggest that mobile and desktop are the most frequently used devices when browsing listings on existing apps (Figure 9). Despite this, responsiveness remains a common issue on some sites such as Rent.ie. Therefore, the project is a mobile first web application. Designing from a mobile first perspective places the necessary restriction of how available screen space is used, by forcing the design to prioritise information and to streamline what is conveyed on screen. (Xia, 2022). While Mobile first design is important, it must be used with caution, as using a mobile layout on desktop can often impede usability with oversized elements. (Experience, n.d.). Therefore, multiple layouts based on device are created.

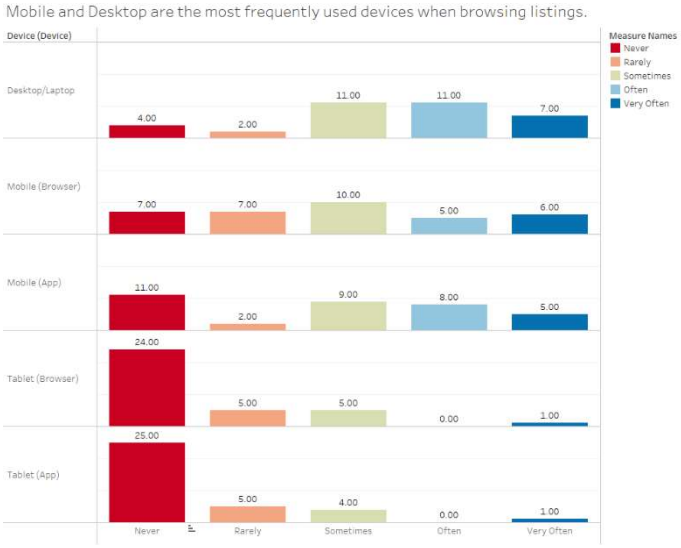


Figure 9 Devices used for browsing

Usability and Accessibility

Another core technical challenge for Frontend and UX is usability and accessibility. While this system focuses on the tenant role/experience, it must account for the diverse abilities of the population.

The principles of Universal Design are a key foundation for designing an accessible and usable system. It can be broken down into seven core principles, consisting of sub principles. This philosophy accounts for multiple types of design, with some irrelevant to software development. The following principles are a metric to design and evaluate a systems UI (*The 7 Principles | Centre for Excellence in Universal Design*, n.d.).

1. Equitable Use:
   - Design should be appealing and usable by all users.
   - Provide equitable controls for all users.
2. Flexible Use:
   - Provide multiple modes of interaction.
   - Account for disabilities, handedness, etc.
3. Intuitive Use:
   - Reduce complexity of interaction.
   - Maintain consistency.
   - Implement user feedback.
4. Perceptible Information:
   - Have alternative modes of communication for key information.
   - Sufficient contrast to differentiate key information from supplementary information.
   - Provide support for assistive technologies such as screen readers.
5. Error tolerance
   - Design layout in such a way to minimize user errors.
   - Provide recovery methods.
   - Communicate system and/or error states to the user.
6. Low physical effort:
   - Reasonable effort for interaction.
   - Reduce repetitive actions.
7. Size and space for appropriate use (irrelevant for software design)

Clarity of Communication

There will be a lot of information for the user to consume and analyse from a variety of sources. As such, users will need to have a quick and easy way to assess a property, that is shared amongst categories of information. For example, a single rating for each category can simplify the problem of disproportionate text descriptions for them.

Beyond the clarity of content communication, communication of the system state is also of fundamental concern. Improper and/or lack of feedback mechanisms can seriously impede the effectiveness of the system from the user's perspective. Toasts, alerts, or animations are great ways to communicate success and error states to the user.

5.3 ARCHITECTURE AND DEPLOYMENT

The architecture and deployment of the application was a core technical problem, specifically the various factors when considering the architecture and deployment technologies.

Delivering a high-performance application is crucial. Poor performance negatively affects user retention, a badly functioning application is an unused application. To ensure a high-performance application, the following was considered: scalability, security, monitoring, testing, database system management, resource allocation, llambda functions, and data migration:

- Scalability: Ensuring the architecture enables dynamic resource scaling and upkeep thus meeting changing user demands.
- Security: Integrated systems for robust authentication and authorization thereby guarding user data and preventing unauthorised access.
- Monitoring: The use of AWS CloudWatch to track performance in real-time thus enabling proactive problem detection and resolving.

- Testing: To find and fix possible bottlenecks before they have an impact on the end user, thereby adopting continuous testing approaches.
- Managing a responsive and reliable database system: Dealing with fluctuating loads and guaranteeing data consistency is a core technical problem in this project. Leading into the next point.
- Resource Allocation: Using the cost-effective MySQL t2.micro instance, which required optimisation to perform database operations well.
- Lambda functions: Leveraging Lambda functions for database auto-shutdown during extended idleness helping reduce unnecessary costs through automated management.
- Data Migration: To enable data uploads from CSV files, AWS's S3 buckets are used to guarantee quick and safe data transfers.

In addition, the cost associated with deployment whilst considering the limitations of an MSc project is a challenge. For architecture and deployment, the student free tier from AWS was utilised as it suits HME's needs and scope. Cost monitoring is enabled with AWS billing alerts and AWS budgeting. Despite budgeting efforts, the team unintentionally racked up a $500 fee for Amazon Aurora, which wasn't included in the free tier. Upon the discover of the unexpected charge, AWS customer support was contacted. AWS acknowledged the predicament and reimbursed the mistake. To mitigate additional mishaps, AWS assisted the team in setting up CloudWatch alarms and other system alerts. The implementation of these systems ensure that the team won't exceed the self-imposed restrictions.

Research and experience are required to manage deployment technologies. Due to the team's differing technological fluency, advanced technologies and their training processes could impede progress. AWS Amplify and Lambda were selected based on ease-of-use thus expediting the development and deployment. Research, tutorials, and experimentation was key to instilling team confidence. Shortcuts were also employed, with AWS Cognito being used to implement authentication and authorisation, mitigating the need to design and implement a solution from scratch.

Problems arise when interconnecting the components of an application. All the components must be contained within the AWS ecosystem to maximise compatibility and reduce integration issues. Additionally, it is cost-effective to maximise the use of AWS free tier, thus accomplishing the project's objectives without sacrificing quality.

5.4 DATA

Database Creation

As part of the ongoing efforts to optimize and modernize the infrastructure, the team decided to implement a cloud based RDS MySQL database using AWS's free tier offerings. This report highlights the steps that were taken to set up the database and elaborates on the challenges faced during the process, along with their corresponding solutions.

1. VPC Connectivity Issues:

During the initial stages of setting up the database on AWS, there were issues related to VPC connectivity. The default VPC configurations provided by AWS did not align with the stringent organisational security policies, potentially exposing the database to vulnerabilities.

Recognizing the gravity of the situation, the team was consulted, the VPC settings was tailored to bridge the gap between AWS's default configurations and the organizational needs. By doing so, the robustness of the security was ensured while ensuring seamless connectivity to the database.

2. Lambda Function Triggers:

The mission was to drive operational efficiency. Part of this was ensuring that the RDS instance was active only when required, thereby optimising costs. While this was theoretically sound, the practical implementation, especially the integration of Lambda functions with the API Gateway, proved to be a complex task.

The initial attempts were met with partial success, prompting a deeper search into AWS's documentation. With a few iterations and optimisations, the Lambda functions (StartRDSInstance and StopRDSInstance) were successfully set up. Linking them to the API Gateway became a key turning point, as developers now had the power to dynamically start the RDS instance, ensuring were only consumed resources when necessary.

3. API Deployment:

When deploying the API on the API Gateway, inconsistencies emerged. The Lambda function calls were not behaving as anticipated, leading to functional bottlenecks.

Root cause analysis revealed that the issue lay in the IAM roles and permissions. The initial permissions assigned to the API Gateway were insufficient to invoke the Lambda functions. These permissions were re-evaluated and adjusted, ensuring that the API Gateway had the explicit rights to trigger the Lambda functions. This rectification eliminated the discrepancies, making the system more responsive and efficient.

4. Exploration with Aurora:

In the quest to explore the best database solutions, an Aurora instance was set up. While Aurora offered promising features, there was an unexpected AWS billing issue as mentioned earlier.

This unexpected expense underscored the importance of vigilant resource monitoring. RDS MySQL free tier offering was used alternatively to curtail costs. Additionally, alerts and budget limits were setup on the AWS account, ensuring that there are prompt notifications about any abnormal spikes in usage or costs, preventing such financial surprises in the future.

Steps to Create the Database

Initial Setup: this was done on the Amazon RDS console to initiate the database creation process.

Configuration: Opting for the 'Standard Create' method, MySQL was chosen as the engine type with the free tier template.

Instance Details: The database was defined with the identifier myhmedb and the master username and password were also set up. The "Burstable classes (includes t2)" under the "DB instance size" section was chosen to ensure cost-efficiency.

Architecture Initial vs Current

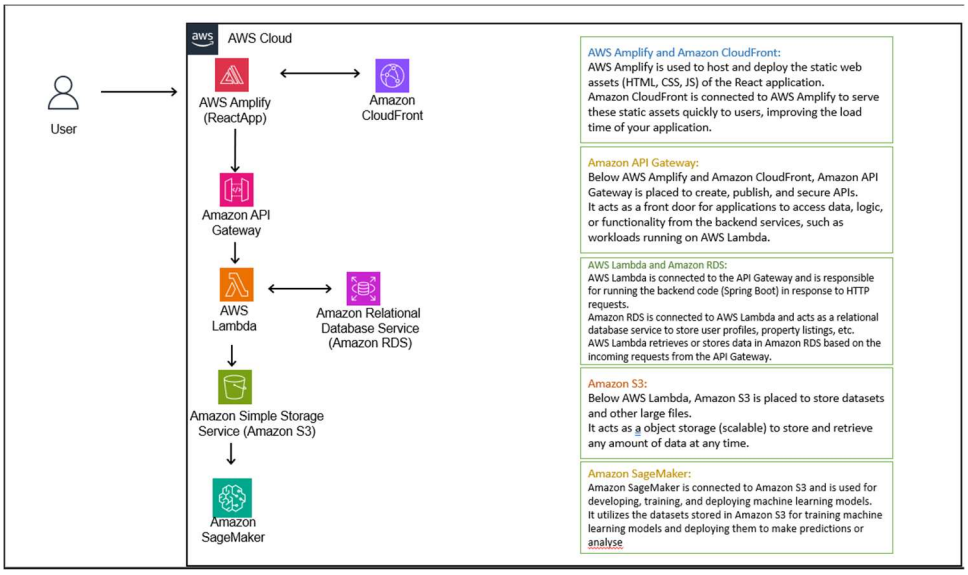


Figure 10 initial system architecture

Initial Architecture:

The original system was designed with the following workflow:

- User Interaction: Users would interact directly with the Amplify frontend.
- Amplify to API: The Amplify platform would relay user requests to an API endpoint.
- API to Lambda: The API would trigger a Lambda function to handle specific database operations.
- Lambda Interaction with DB: This function would communicate with the database to execute CRUD (Create, Read, Update, Delete) operations.
- Data Storage in S3: Processed or retrieved data would be stored in an S3 bucket.
- S3 to SageMaker: Data in S3 would then be used by SageMaker for machine learning or analytics tasks.

Revised Architecture:

The updated design has incorporated additional AWS services to optimise the operation, particularly focusing on cost-efficiency and improved automation:

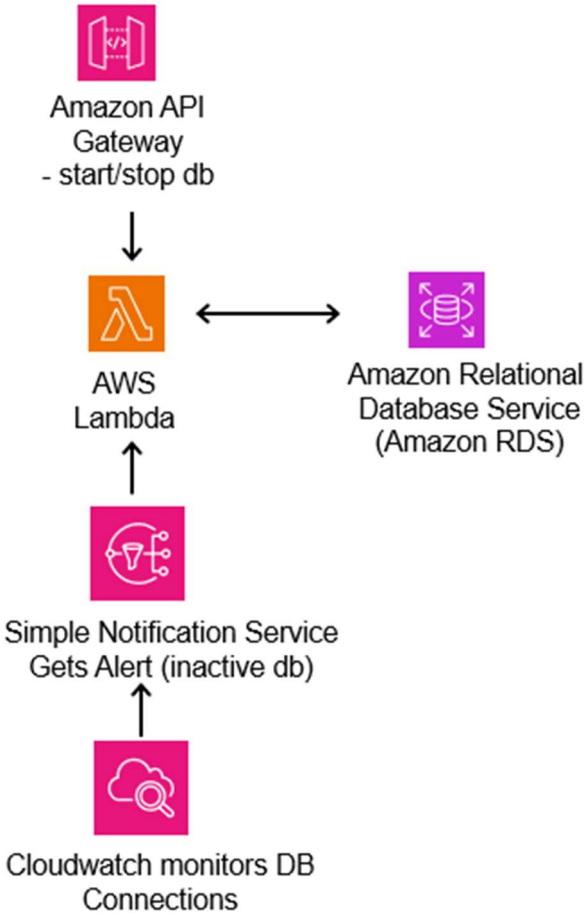


Figure 11 Database cloud architecture

- API to Lambda: This remains consistent, where the API triggers a Lambda function. However, the primary function of this Lambda now includes controlling the RDS database's state - turning it on or off based on requirements.
- Lambda to DB: When the database is needed, the Lambda function starts it, allowing for operations. Post operations, the Lambda function can trigger a stop command.
- CloudWatch Integration: CloudWatch monitors the database's activity. If there is a period of inactivity or low activity (1 hour), CloudWatch generates an event.
- SNS Topic: Upon detecting the CloudWatch event, an SNS topic is triggered. This topic effectively sends a command to stop the database, ensuring that the RDS is not running (and incurring costs) during periods of inactivity.

- VPC & Security Groups: To bolster security and streamline network communication, all these components (API, Lambda, DB, CloudWatch, and SNS) are encapsulated within the same Virtual Private Cloud (VPC) and are governed by the same security group. This ensures that the components can efficiently communicate internally while being shielded from potential external threats.

Data Ingress

Data Extraction: The initial step was to pull data from these public sources. The Zillow API and Google Maps provided direct API endpoints for this purpose, whereas crime information from Open Baltimore was downloaded in structured CSV format. The crime dataset is spanned over a 7-year period, from 2016 to the present date.

- Data Transformation: Given the diverse nature of the sources, the extracted data came in different formats. Using an ETL (Extract, Transform, Load) process, all incoming data was converted into a consistent format, making it compatible with the database schema.
- Loading: Once the data was transformed into a suitable format, it was loaded into the RDS MySQL database. To ensure efficiency and reduce potential strains on the system, batch processing techniques were used to do this. While the process was meticulously planned, challenges were still encountered:
- API Limitations: APIs, especially the Zillow API and Google Maps, had rate limits which constrained the speed at which data could be fetched.
- Data Consistency: Since data was being taken from various sources, there were occasional discrepancies in the data, such as different naming conventions or units of measurement.

Cleaning and Preprocessing Data

Multiple sources were used to get data for this project. Prefetched data was collected online, as well as data collected by extraction from available sources. They all still needed a level of cleaning. Some property listings had been removed from the Zillow database between the time frame of retrieving property ID information, and property attribute information; this resulted in some missing data when pulling the properties dataset. The businesses datasets from Google had a policy on data that can be cached and so only data that is allowed to be kept was extracted.

- The geometry column of the business datasets from the Google Places API contained the location coordinates, but also contained bounding box coordinates which were removed because they were not needed.
- Different rows in the datasets from Zillow had missing values (because some listings had been removed from Zillow). Each row had to be linked to a property ID in each dataset, so to have no missing information, approximately 300 rows had to be removed from the dataset.
- Apartment units needed to be linked to their respective apartment buildings. This linking was done by creating a building ID.
- There were three different datasets with property details (amenities, appliances, etc.). A lot of duplicate information was shared amongst the three datasets. However, the information was conveyed in different ways, so keyword searches were done in different strings to reduce the data.
- Some date and price data were missing on the Zillow dataset. This was imputed using the data from the different columns in the price history dataset to deal with this.
- Neighbourhood information had to be extracted from google API and linked to each address in geo coordinate formatting to retrieve business information during runtime, and to be able to group crime counts by neighbourhood.
- A lot of the address information for the apartment units had different names. A new unit's column had to be created using end values of street address. A pattern expression had to be defined with the different unit string names contained inside of it.
- All the datetime columns had to be converted into the same YYYY-MM-DD HH:MM: SS format.
- In the crime dataset, some of the neighbourhoods in the address's dataset are not contained in it. Geocoordinates of the neighbourhood address dataset were taken and the Haversine distance formula was used to find the closest neighbourhood in the crime dataset, to prevent a lot of properties containing null information on crime counts.
- Exploratory data analysis was done on crime records to show crime frequency in the neighbourhoods and hours recorded.

- Data management practices of only collecting necessary data, in this case businesses.

5.5 Review of Existing Solutions

In this section, Daft.ie and Zillow will be examined, giving brief overviews, feature descriptions, and critiques of each. Many of the existing sites cater to different user roles, i.e., landlords and estate or letting agents, but these are beyond the scope of the project and therefore will not be included in the analysis.

Common Functionality Across Systems

There are several features that are shared across the different existing sites. These make up the MVP features of the system.

- Browse Rental Listings
    - Search by location
    - Filter by:
        - Distance
        - Price
        - Bed count
        - House Type
        - Facilities/Amenities
        - Date Available
        - Date Posted
    - View property details
        - View description
        - View on map
- Save or favourite listings for later viewing.
- Save or favourite search configuration for later use.
- Apply online or contact agent.

Daft.ie

Overview

Daft.ie is Irelands leading rental listings web app. Beyond basic rental functionality, Daft.ie is quite barebone, with little unique functionality to significantly differentiate it from other competitors.

Screenshots

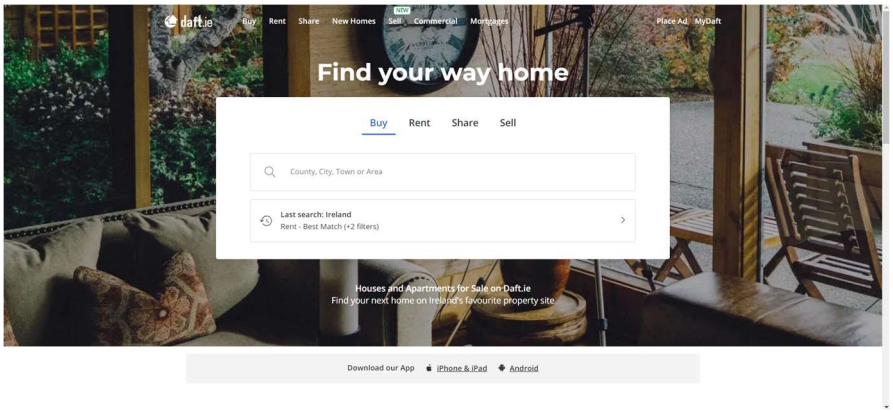


Figure 12 Daft landing page

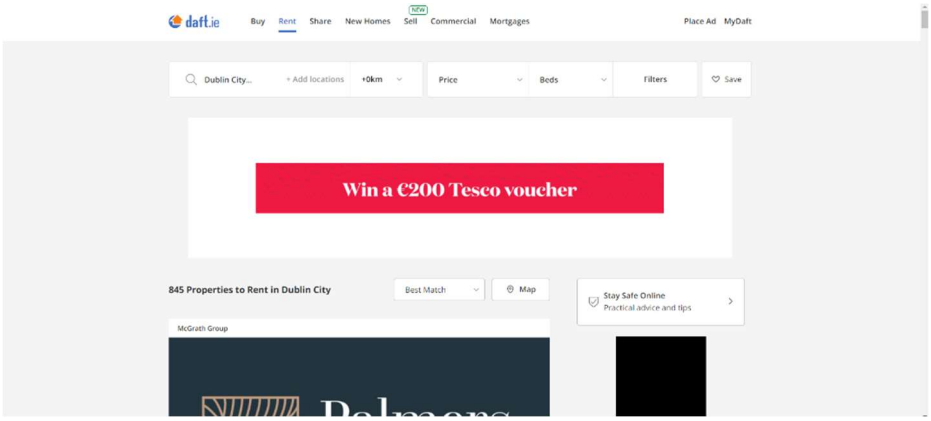


Figure 13 Daft browsing page (upper)

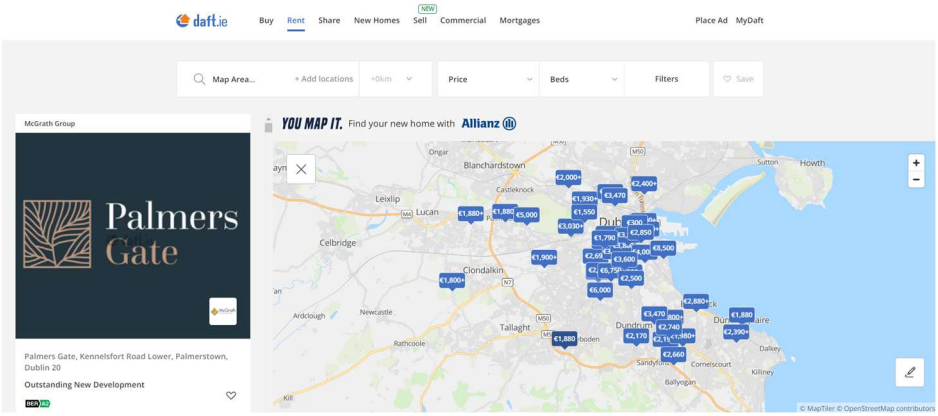


Figure 14 Daft browsing page (map view)

Critique

Daft.ie is a pretty good example of the standard functionality of existing listings sites, see appendix A for demo video. It lacks a recommendation system but does feature a map view in which users can draw a shape outline to filter results. The site follows a minimalist UI and is clearly designed from a mobile first perspective. However, it falls into the error noted by *Experience, n.d.* in that its desktop site uses oversized elements which increases the distance needed to scroll. Combined with the lack of a floating header bar, this means users must scroll a lot down a page, and then scroll all the way back to the top manually to navigate to another page. This is troublesome based on the 6th principle of universal design.

There are additional issues introduced by the amount of screen space dedicated to advertisements (see screenshots above). When browsing, there is a banner advertisement between the filter inputs and the results, and there is additional advertisement to the right of the page. The site generally feels overly spaced-out.

In terms of UX laws and principles, the website makes use of Gestalt Laws such as proximity, similarity, and continuity. Because it chooses to only have a plain white background with minimal visible line dividers, it is harder to visually discern categories of information. This is most pronounced on the property details page.

The aforementioned needs of the personas reflect user profile requirements for a recommendation system that Daft.ie doesn't provide. It fails to incorporate as many of the additional details (i.e., Bus routes/schools) in a visual way as well. It lacks a highlighting system that could allow for quick assessment of the suitability of a property from the results page.

Zillow.com

Overview

Zillow.com is a North American rental listings application, with numerous additional services including home loans. Zillow has features most like the proposed system, including its scoring system and recommendation system.

Zillow features a great scoring system for Walkability, Transit, and Biking. It also features information on nearby schools, local legal protections, and price history. All in all, it is a fully featured system, only lacking in real-world crime rate scorings. Another useful feature it has is a tool to directly compare properties. Since it has more features that daft.ie and other known sites such as rent.ie, its feature set could be considered state-of-the-art.

Screenshots

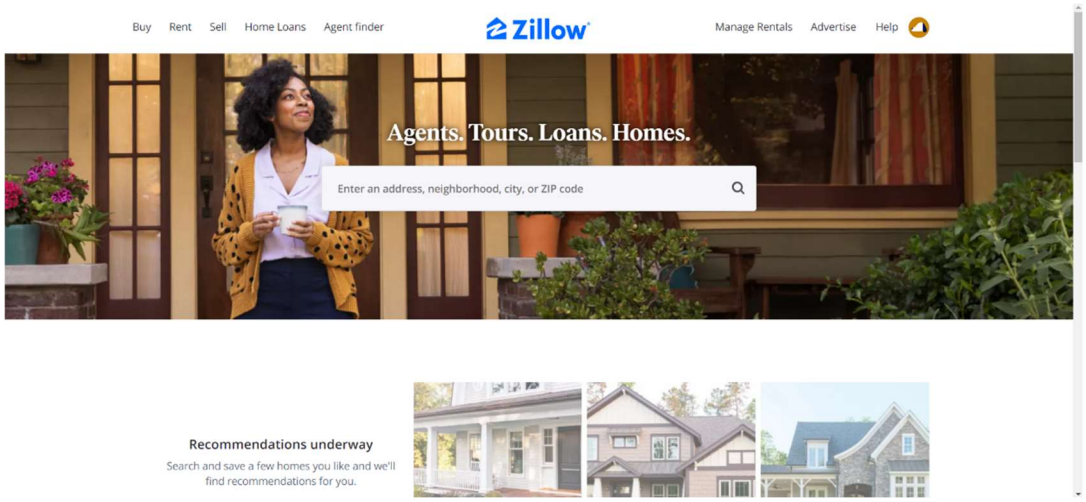


Figure 15 Zillow landing page featuring recommendation carousel

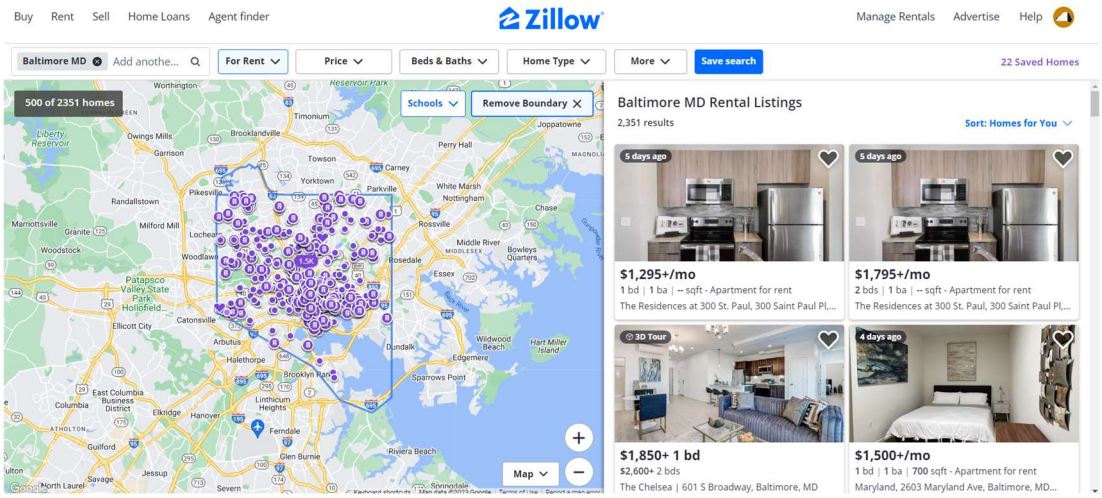


Figure 16 Zillow browsing page

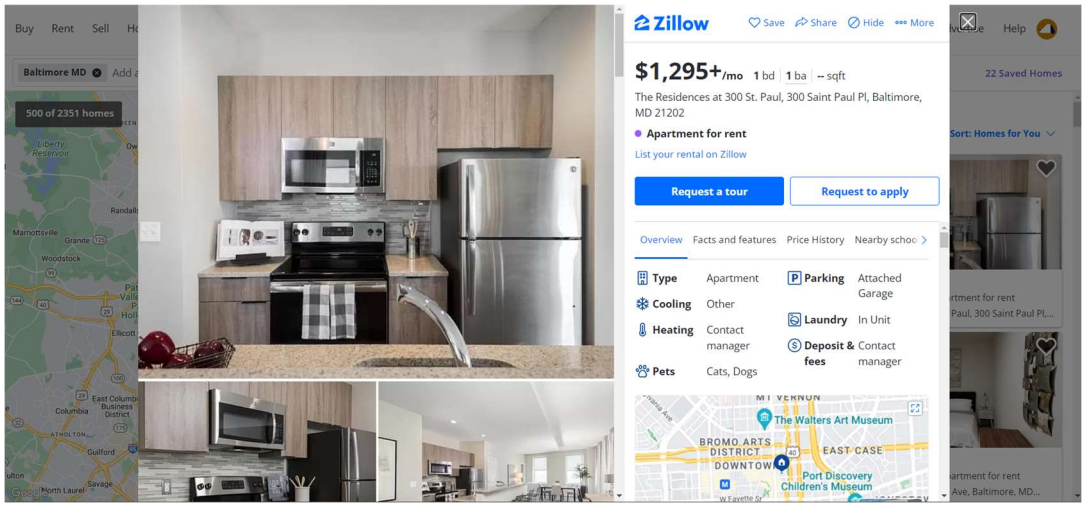


Figure 17 Zillow property details modal

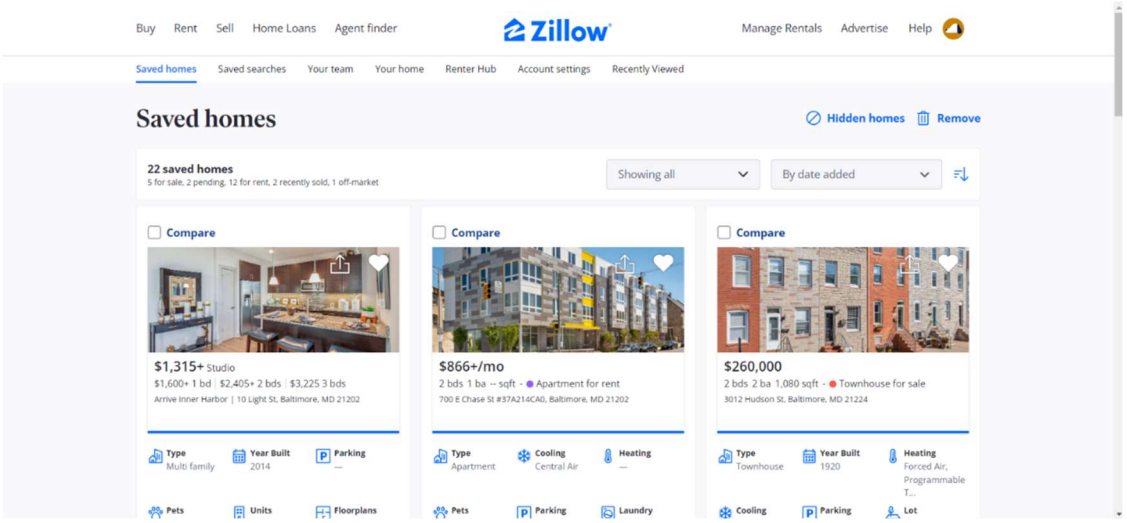


Figure 18 Zillow saved properties page

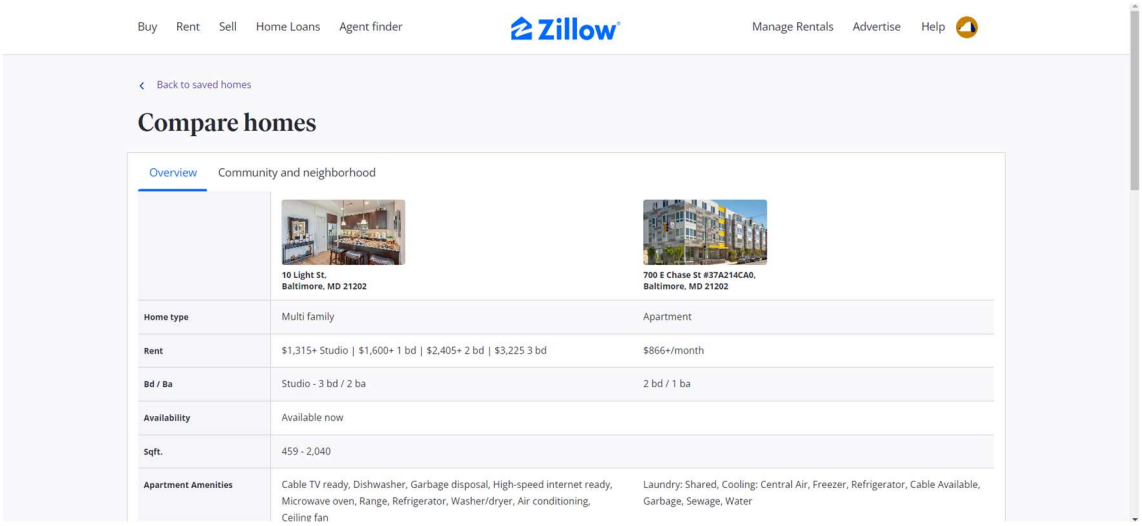


Figure 19 zillow property comparison page

Zillow Recommendation System

Zillow's recommendation system is designed to provide prospective homebuyers with personalized and diverse recommendations to help them explore the real estate market effectively. It addresses some unique challenges in the real estate domain, particularly the "new-listing cold-start problem," where newly listed homes lack sufficient user interactions for traditional recommendation approaches. Zillow employs a combination of content-based and diversification algorithms to enhance the diversity of home recommendations.

In this case, content-based filtering is used to address the cold-start problem. A content-based model is used to compute a relevance score for each home listing based on its attributes and the user's preference profile. The aim is to predict the user's click probability for each home independently, considering factors like price, square footage, and house type. A content-based model is trained to understand a user's preferences and could be particularly valuable for newly listed homes with limited user interactions.

To improve the diversity of recommendations, Zillow applies a diversification algorithm. This algorithm re-sorts the list of candidate homes to balance both relevance and diversity. It's an essential step to ensure that the recommended homes are not overly similar. The diversification algorithm considers the user's preferences across different categories (e.g., property type, price, size) and adjust

ts the weight for diversity (λ) as needed. The result is a list of top diversified items that provide users with a more varied set of options.

Zillow's recommendation system improves upon traditional content-based filtering by focusing on diversity and personalized recommendations. By addressing the cold-start problem and employing diversification techniques, it helps prospective buyers discover a wider range of properties that align with their preferences and include newly listed homes. These strategies produce in an effective recommendation system.

Critique

Zillow has the most comprehensive information available on its listings, see appendix B for demo video. It makes use of several Gestalt laws throughout: common region, similarity, proximity etc., and generally follows the design principles of Flat 2.0, with some depth indication using box shadow to group and divide elements.

However, Zillow has a few problems, Zillow is cramped compared Daft.ie. Its browsing page does not allow for the map view to be toggled on/off, meaning that it permanently takes up ~60% of the screen's width. From the user's perspective, it is more intuitive for the map to be toggleable, so that they have finished coarse filtering using the map, they can disable the view to focus more on the results themselves.

This overcrowding is worse on the property details modal, where the property images take up ~50% of the modal width. Primarily on desktop, this reduces the area that can be used for conveying textual information about the property. In addition, the price, address, and action buttons are sticky elements. This results in roughly 25% of the modal area is used for textual information (Figure 19).

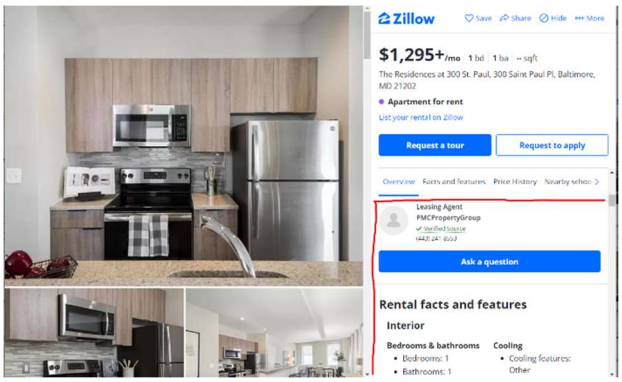


Figure 20 Zillow property modal with scrollable area highlighted

The lack of space may overwhelm the user and make it harder to accurately comprehend the information being conveyed. Additionally, it is understandable that the apply and request tour buttons are given a significant amount of dedicated space where they can be quickly accessed. However, it is unintuitive that those buttons are above the property details, which is before the user has made the decision to apply or not. It makes more logical sense for the buttons to be at the very end or be sticky elements at the bottom of the page, like the mobile version of the site.

The site makes a poor distinction between individual listings, and listings for buildings with multiple units. The only way to differentiate the two from the results is a small '+' symbol beside the price on the listing card. This is significant because individual units and buildings have different layout schemas. This is potentially difficult for users such as Jenni, novices to the system who are still building their mental model of the site.

Many of these issues can be addressed by using a dedicated property page rather than a modal.

Summary

Table 1 – Competitor comparison

Daft	Zillow
Difficult to exit the map view	Good property details information
Too much space dedicated to advertisements	Good filtering functionality
Oversized pictures on desktop, requires too much scrolling	Poor use of screen space on desktop
Results show sponsored listings (from larger agencies) first	Unclear differentiation between individual and multiple unit listing
Minimalist UI	Nested modals
Simple plain white background generally makes text easy to distinguish	Lack of feature highlighting for user-relevant property features
Some issues when using a colour contrast analyser on subtext and issues for grouping sections of related text due to lack of borders/shapes.	Recommendation system using content-based filtering and diversification algorithm
Lack of floating navigation bar results in manual scrolling to the top of the screen	

6 TECHNICAL SOLUTION

HME is aimed at improving the aforementioned websites (i.e., Daft, Zillow) and presenting improved solutions to the user identified problems. HME needs to incorporate the common tenant functionality of pre-existing systems (e.g., browsing listings, saving searches, applying/ contacting landlords etc.). As mentioned in previous sections, HME will also introduce additional functionality by combining external information sources and adding a recommendation system.

The project is purposefully limited in scope as the team's work is unlikely to rival competing websites. HME is proof-of-concept, this system can be expanded by future work.

6.1 FUNCTIONALITY SPECIFICATION

The following use-case diagram is a high-level description of the system's MVP functionality and describes the functionality available to a logged-in tenant.

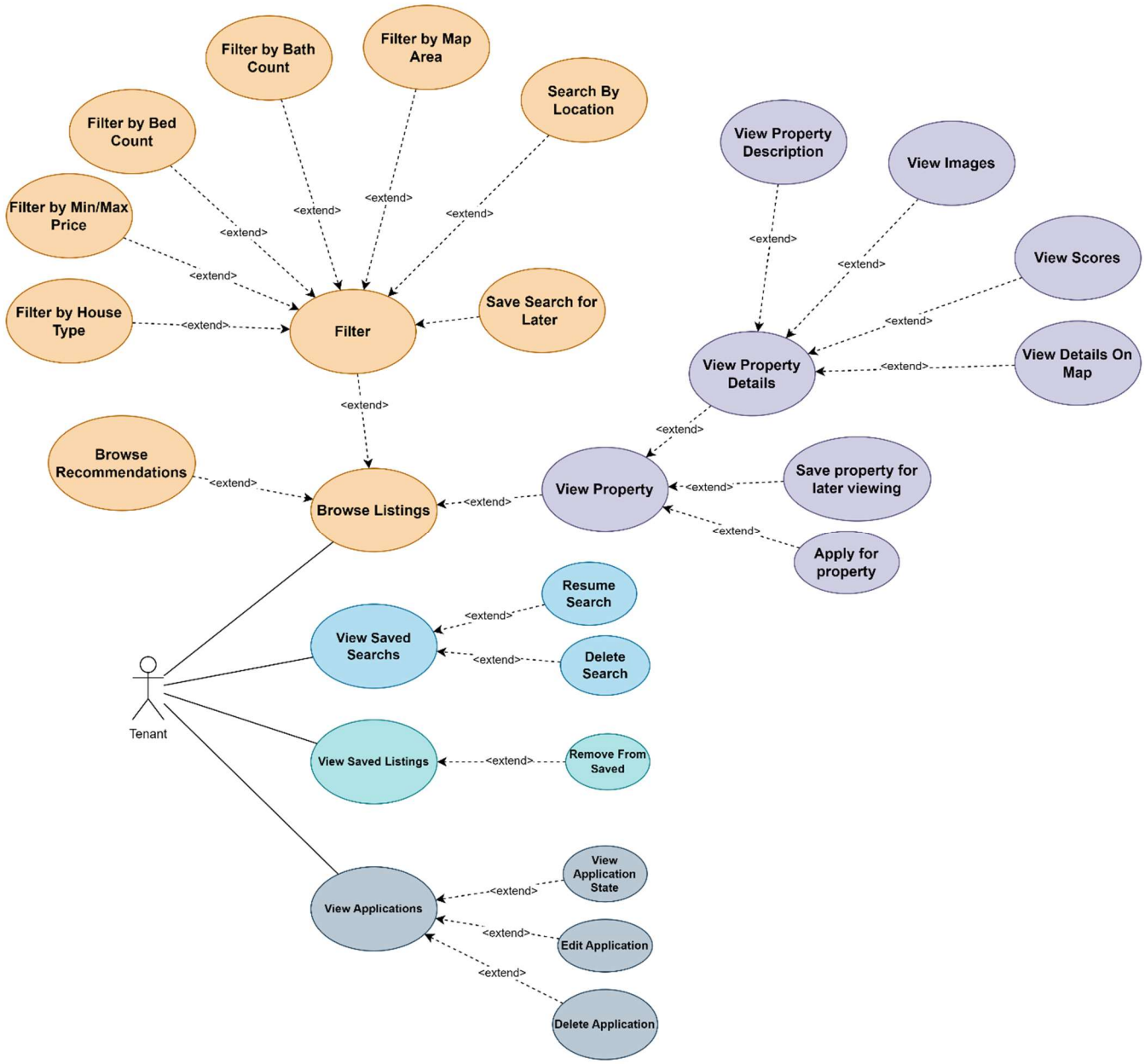


Figure 21 High-level use-case diagram of MVP functionality

The following functionality will be included:

- Additional Information Sources for Property Details: Existing systems (e.g., Zillow) use third-party APIs to provide scorings on walkability, transportation etc., HME will expand upon this by adding crime-rate and security information, and provide different ratings based on Google Maps API data. This information will be presented as scorings with additional explanatory information. This will help Jenni in her search for properties, as she will only be looking at properties in secure areas.
- Personalized Recommendations Carousel: Whenever Andre relocates for work, he repeats the accommodation finding process. This repetitive process can be alleviated by introducing a recommendations system based on Andre's personal profile. By incorporating user profile information, more relevant properties can be suggested (See **Error! Reference source not found.**)
- Tag Highlights: To improve the legibility of the system, a mechanism for highlighting each property's best scores, and the scores which are the most relevant to the user will be implemented. For example, one of Jenni's main concerns is safety, so a listing with a high safety score will have a 'Secure' badge to indicate its score. (See **Error! Reference source not found.**)

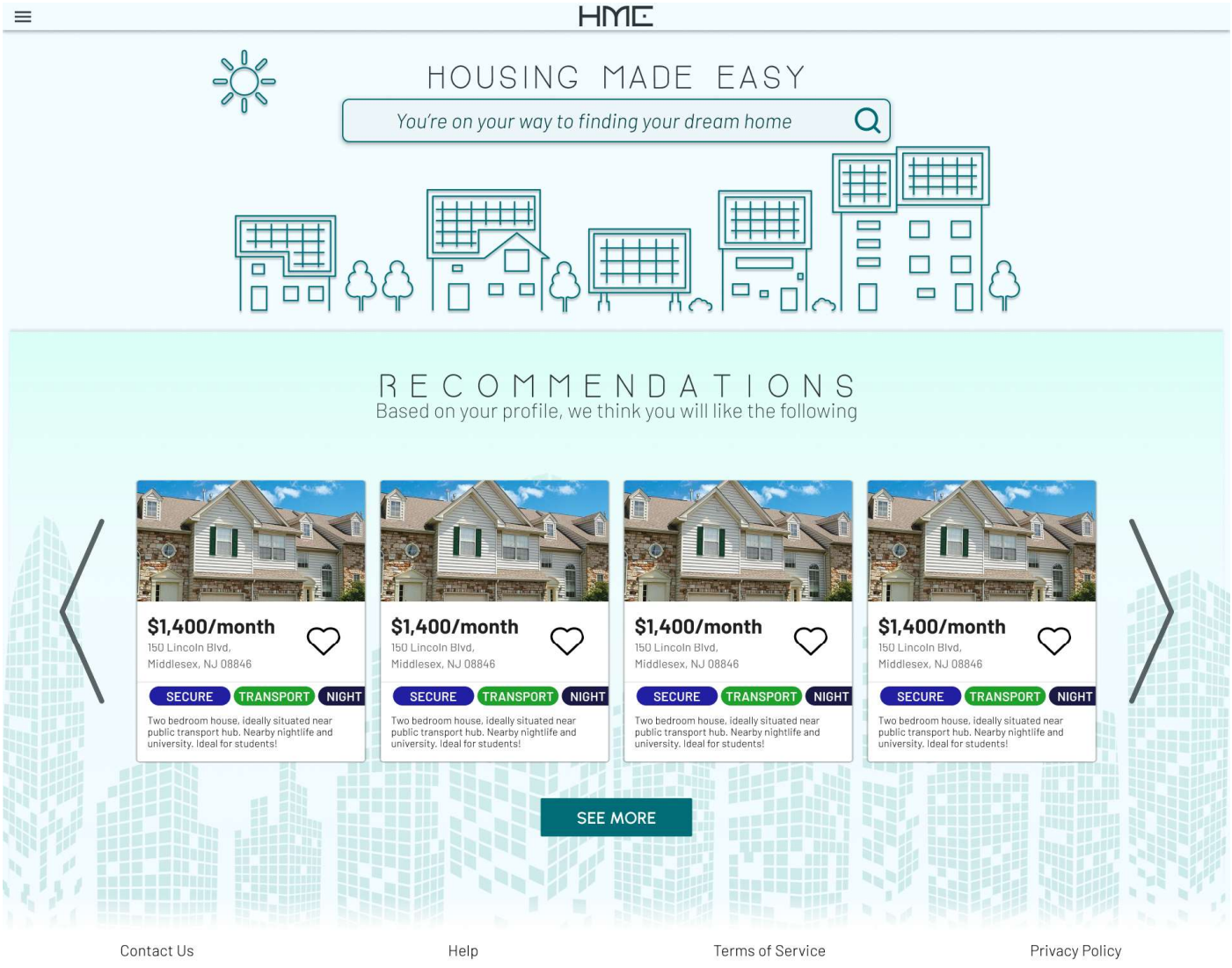


Figure 22 Mock-up of homepage recommendations carousel

6.2 MOCK-UPS AND PROTOTYPES

Before developing the frontend for HME, mock-ups were required for performing preliminary user testing. The following screenshots are taken from the early Figma designs and were used during Low Fidelity Prototype (LFP) user testing.

Link to the Figma Page with all device layouts: https://www.figma.com/file/eXNndfTVIhnTQqWkJdBKhu/UX-Design-File?type=design&node-id=188-13927&mode=design

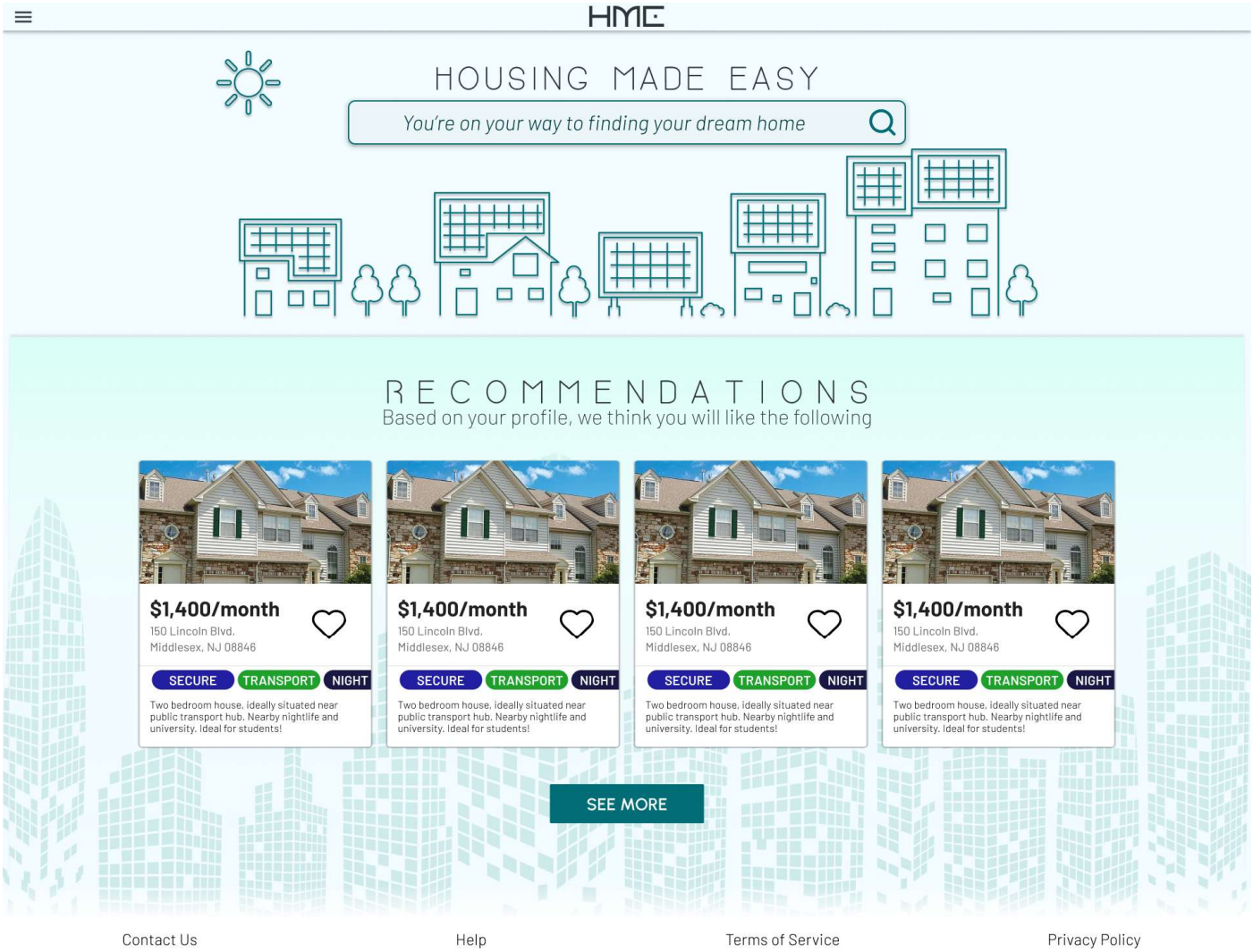


Figure 23 Early mock-up of landing page

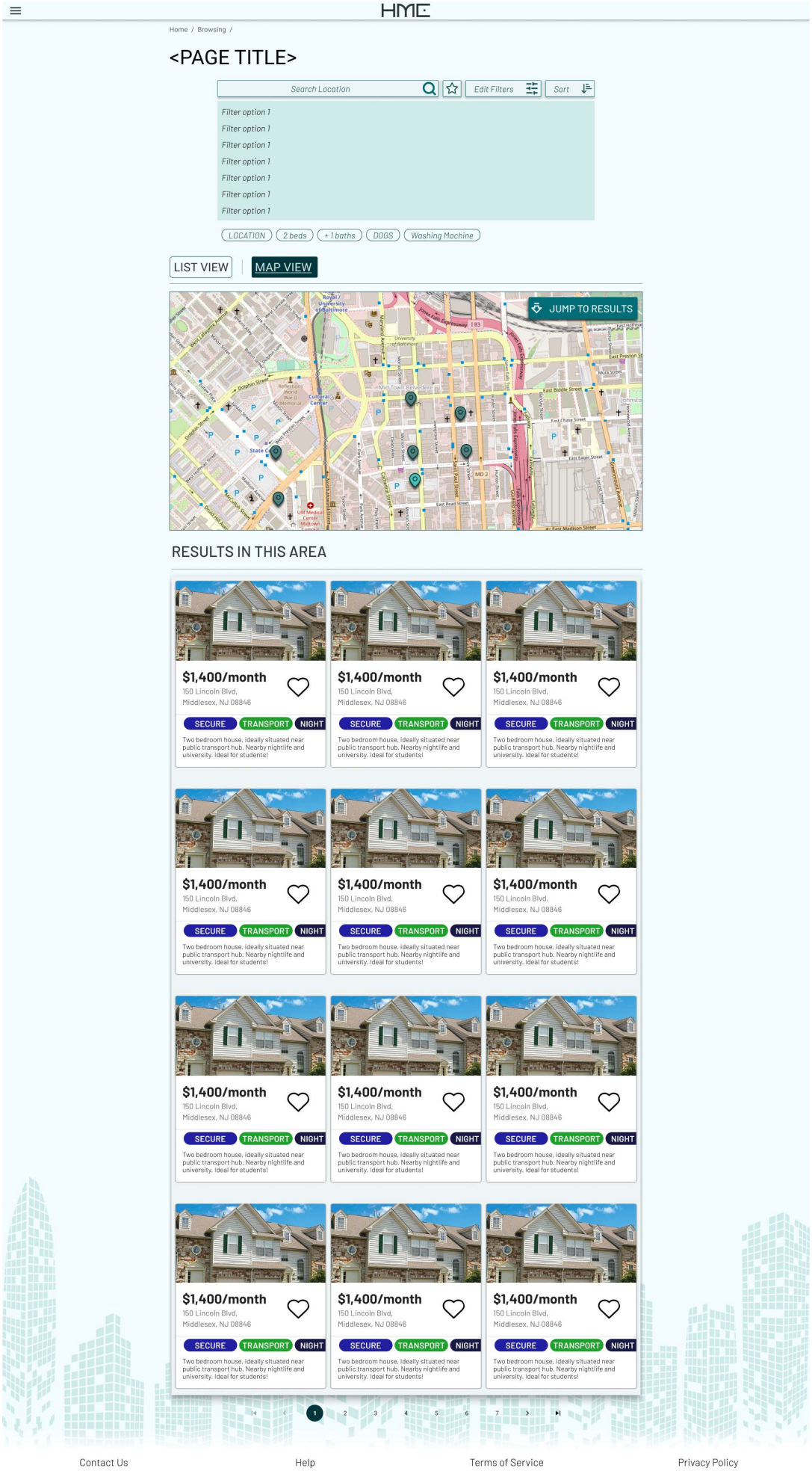


Figure 24 Early mock-up of browsing page with map view enabled

Figure 24 Early Mockup of Browsing Page with Map View Enabled

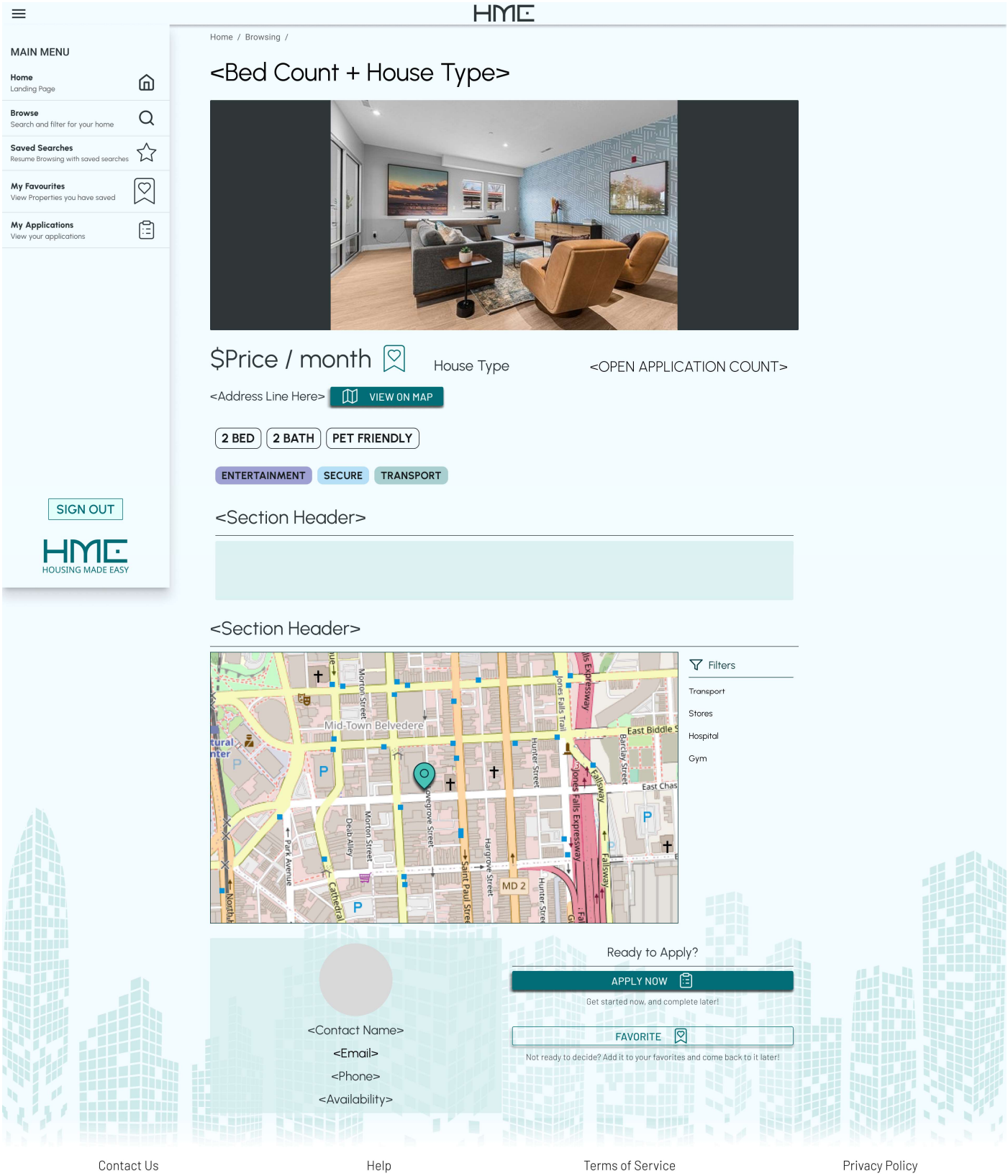


Figure 25 Early mock-up of property details page with site menu open

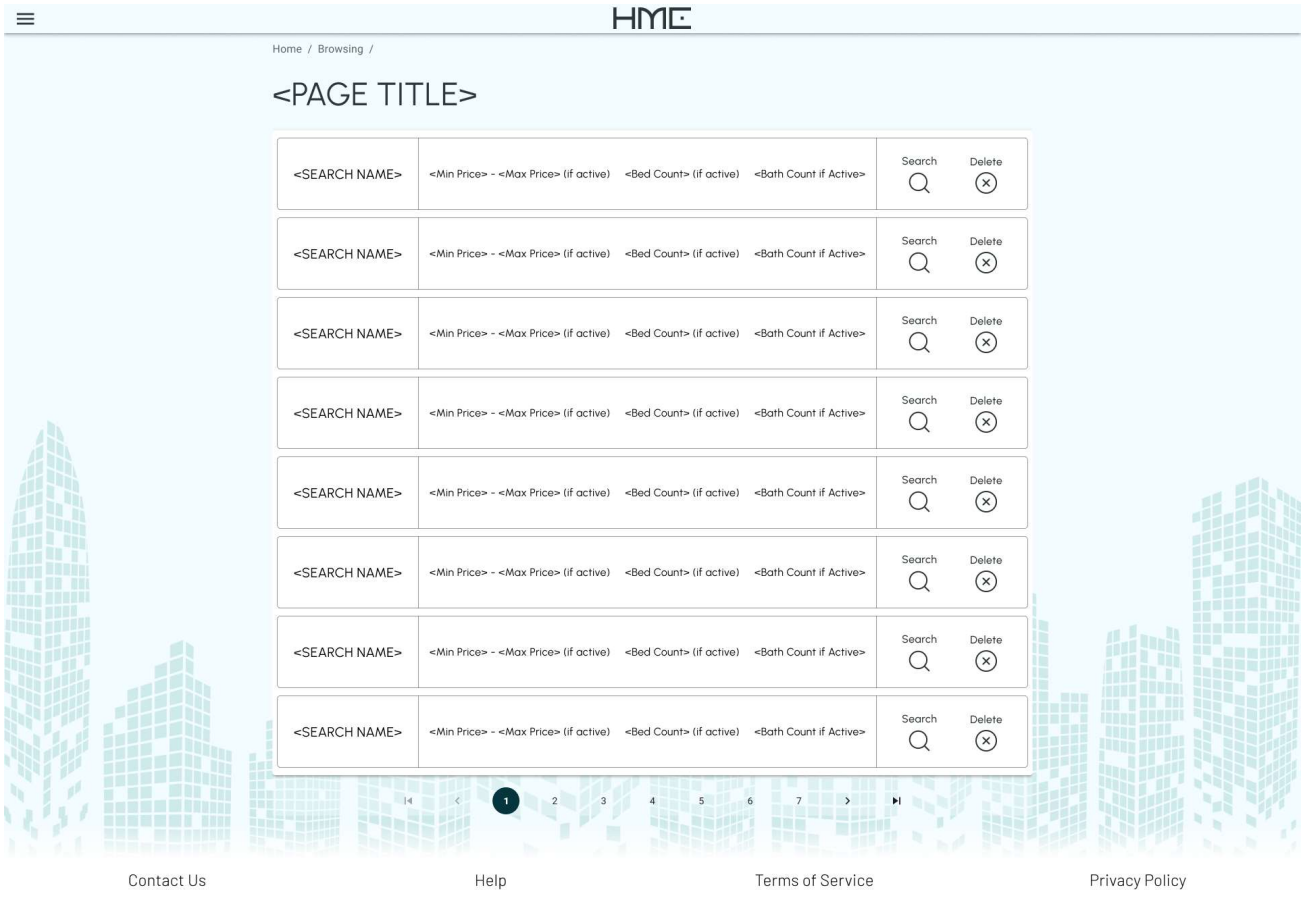


Figure 26 Early mock-up of saved searches page

Figure 27 Early mock-up of my applications page

6.3 System Architecture

The architecture of the system relies heavily on the ingress of data from external API sources. Due to legal limitations on storing data from Google's Map/Places APIs, a temporary caching strategy is required. This data is then combined with the properties dataset and processed using the content-based filtering and scoring systems. See Figure 29 and appendix C for details on this process. In addition, the deployment and system architecture of H.M.E is shown below in Figure 28.

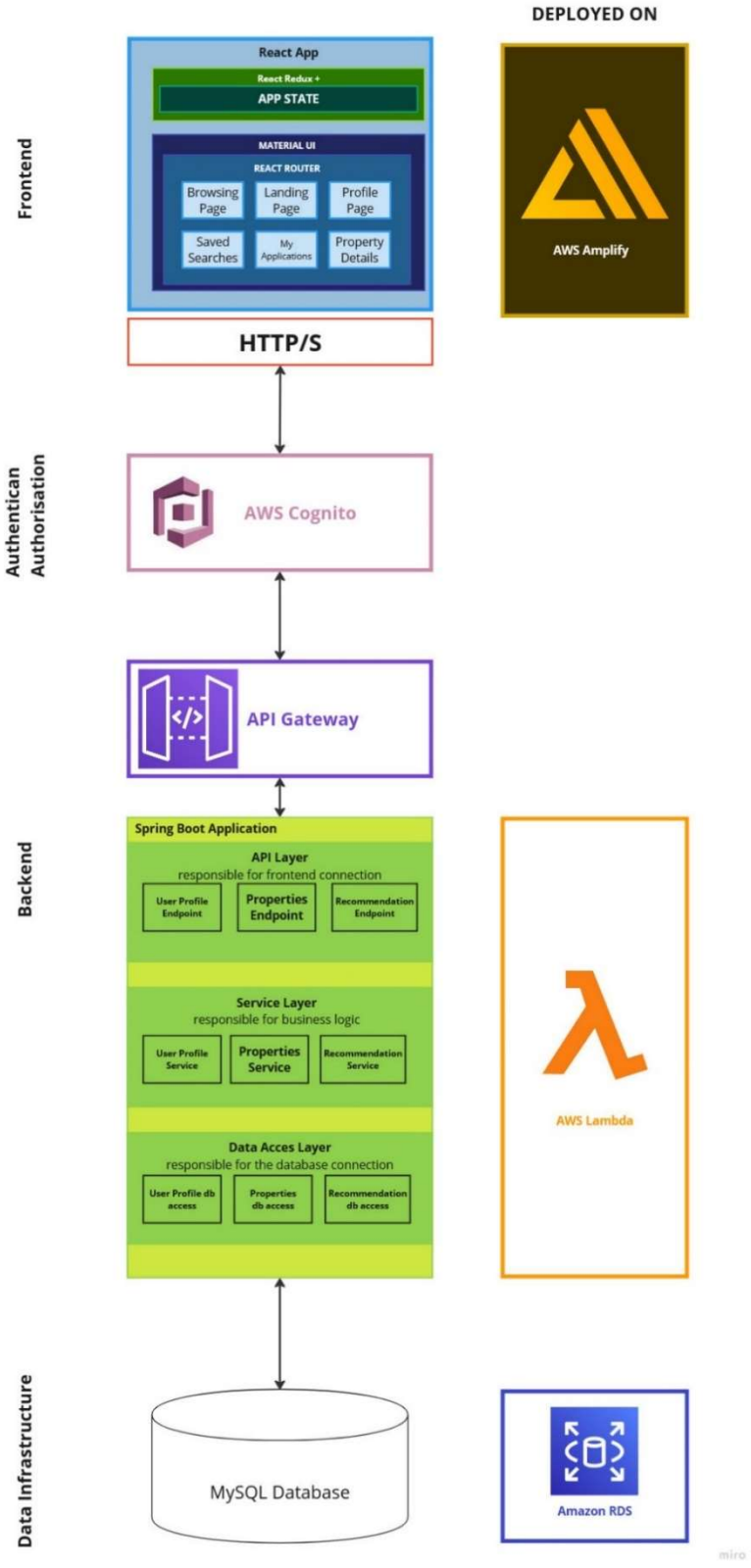


Figure 28 Current system architecture diagram

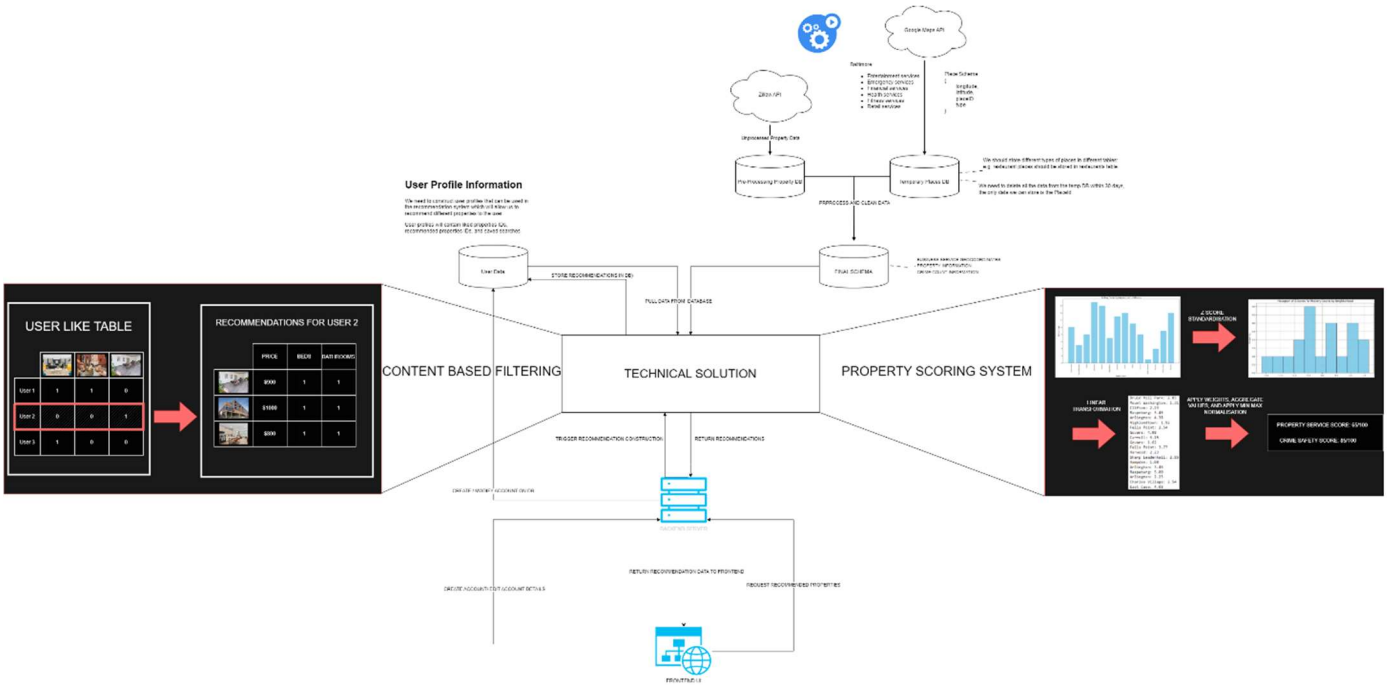


Figure 29 Data flow and scoring architecture diagram (see appendix C for larger image)

6.4 TECHNOLOGIES

Front-end

- Frontend Architecture: React allows the team to reuse components to build a frontend application as part of a Component Based architecture.
- Authorisation: AWS Cognito authentication allows for authentication and authorisation state management on the frontend using a Context-like API. Cognito also provides a pre-build sign-in form to speed up development.
- Deployment: AWS Amplify, allows for quick and continuous deployment. Integrates well with Cognito and API Gateway. A webhook was created to allow for manual deployment to prevent automatic builds from committing to GitHub to reduce costs.
- Form State Management: React Hook Form provides an easy-to-use form state management solution which will ease the development of search filters.
- Component Library: Material UI (MUI) is a semi-opensource component library, with pre-styled components that are responsive and accessible. It also features a custom props-based styling system, which enables easier state-dependent styling than traditional CSS.

Backend Technologies

- Authentication and authorisation: rather than manually implement this, AWS Cognito user pool was discovered as part of the AWS free tier. This decision was based on various arguments.
    - AWS Cognito is compatible with the rest of system, specifically React.js.
    - Cognito lightens the workload for backend development by avoiding the implementation of an authentication and authorisation system from scratch.
    - The Cognito framework is quite straightforward to implement. Which is crucial considering the timeframe of the project.
    - The project fully utilizes AWS student free tier.

- API framework: Both Node.js and spring boot were considered as the backend framework. After initial research, both appeared to suit the needs and scope of the project. Despite more team members having experience with Node.js and/or JavaScript, Spring Boot was chosen as the framework for the following reasons.
    - The backend development team was interested in gaining significant experience with Java and Spring Boot.
    - The API was designed with the REST architecture and sends data through HTTP requests to the frontend using JSON. This architecture style was known to most of the team and provided the application with the most straightforward solution for information exchange.
    - Apache Maven is used for dependency management, testing and compiling, packaging and deploying source code. There is a serverless Maven archetype that allows for easy implementation of the AWS serverless model using spring boot. This resulted in a spring boot application configuration for AWS Lambda.
- Deployment:
    - For an expedited development and deployment, AWS Lambda is used to deploy the spring boot REST API. In this way, the application is not always online, rather the lambda function is triggered by different events. The lambda function receives resources automatically and will use the necessary memory for the duration of the task (Heddings, 2020).
    - API Gateway serves one purpose, to expose the lambda function as a backend endpoint and route requests to the lambda function. It converts HTTP requests into event objects and redirects them to the lambda function (Perera, 2018).
    - SAM CLI is used for configuring, packaging, and deploying the backend to AWS lambda and API Gateway.

Data Technologies

- Data scraping: The property information was extracted from Zillow using a Zillow API from rapidapi.com.
    - This included core attribute information such as price, date, and bed and bathroom data.
    - An unofficial Zillow API was being used so many challenges were faced. The API was unclear to use, and contained many different types of API calls, each retrieving various kinds of attribute information.
    - The requests could not be made in any order and each request had specific input parameters that could only be attained from previous requests, so a trial-and-error approach was incorporated as a solution. This involved going through each request, looking through its contents, and seeing if it contained the required input parameter information for the other calls. Some requests would only return a proportion of the information needed, and the remains had to be found through other requests.
    - Another part of the problem was the planning of what data was required. Some requests would return hundreds of different keys with different and duplicated information, knowing what was needed became overwhelming. A list containing the necessary data that was needed was created as a solution.
    - Every key in each request was explored manually, and if it was felt that the information in each key met the list criteria, it was extracted.
    - Only 11,000 requests to the API could be made before a billing fee would incur, for this reason the data had to be pulled strategically, the solution was to find the available API requests that provided the largest amounts of information, so that others would not have to be used. Each API request returned information for one property.

- Google maps**:** A business location dataset was needed to help inform what types of businesses are around each property. Google Maps platform was the most convenient for this task. The Baltimore city website was explored but had poor usability and it was unclear how to extract data from it.
    - The google maps platform made available the same database used by Google Maps, and so it had enough location data for Baltimore MD, USA.
    - They had different APIs for different functionalities such as Places API which searches for businesses around a geocoordinate and Distance Matrix to show how far one place was from another.
    - The Google Maps platform requires users to have an account on Google Cloud to access an API key.
    - This API key is used with the "googlemaps" library in Python to extract location data.
    - The nearby search request is used where parameters such as location(coordinates), radius (to determine circle search area around each property), and type (which type of business, for example, restaurants).
    - The queries are limited to one location and one business type at a time and only give 60 results after going through 3 pages with 20 results per page.
    - Google maps does not allow the storage of the names of businesses, their geocoordinate information was retrieved instead (*Policies for Places API*, n.d.).
    - Due to cost issues, rather than making active API calls to access this information, it was decided to gather and store all geolocation information by neighbourhood in a separate dataset beforehand. Then using the haversine formula (Dauni et al., 2019), nearby services for each property will be found within a 2-kilometer radius.
- Open Baltimore**:** The crime dataset was downloaded from a publicly available dataset on the Open Baltimore website.
- Database schema**:** After retrieving the data, a lot of duplicate information was found, particularly amongst the address and property detail data.
    - As a resolution, the property detail information was broken down into two datasets, and the address information was separated into three datasets. This prevented tables from populated with a lot of duplicate data.
    - Each table has its own primary key to enforce data integrity and to allow for a solid foundation that can easily be built upon if tables needed various kinds of information added in the future.
- The tables in the schema are as follows:
    - Main information table
    - Address table
    - Unit address table
    - Apartment building information table
    - Property fees table
    - Property price history table
    - Crime information table
    - Nearby services table
    - User profile table
    - School table
    - Property details table
    - Property details extra information table

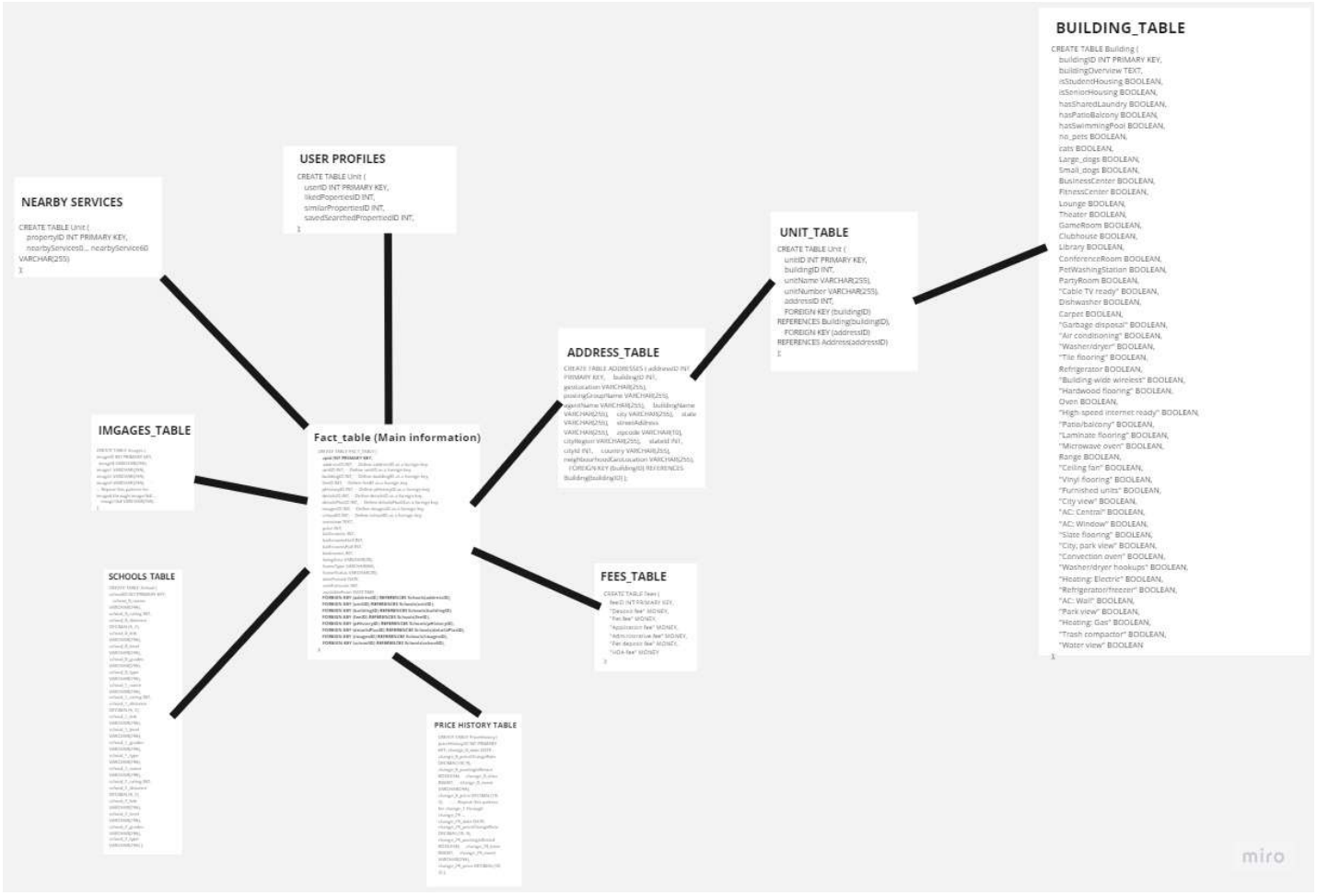


Figure 28 Data schema (see appendix D for larger image)

Property Scoring System:

- Two property scores will be generated for each property and will be displayed as values out of 100 to users. These scores will provide a general overview of the area each property is in. This first score will display crime safety rating based on the neighbourhood of the property. The different types of crimes can be viewed in the figure below.

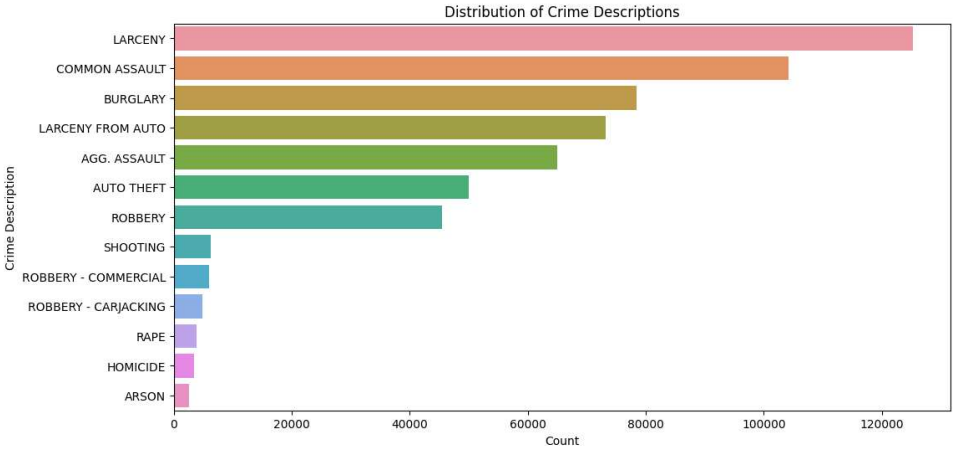


Figure 29 Crime Categories + Frequency

- To generate the ratings for the crime safety score, there is a data balancing issue. Some counts of different crime categories will inevitably have significantly higher values than other categories due to crime commonality. This introduces a bias issue that needs to be addressed, so that when generating an overall crime score, certain categories don't overpower the result. To solve this problem z score standardisation will be implemented. This will generate a different score for each category in each neighbourhood in Baltimore. These scores will be described in terms of their relationship to the mean. Where their values are measured in

terms of standard deviations from the mean.

- In the two figures below, the first figure below displays robbery count by neighbourhood in Baltimore. The second figure demonstrates the new z scores for each count, which are represented in standard deviations from the mean. Note, there are less bins in the second diagram as certain bins represent z scores for neighbourhoods with the same robbery counts.

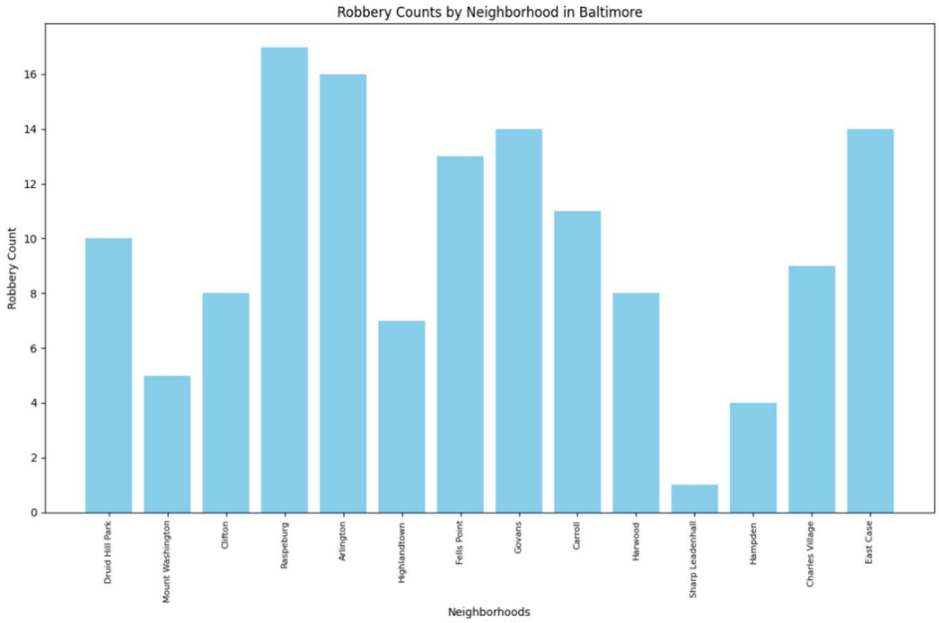


Figure 30 Robbery Counts by Neighbourhood

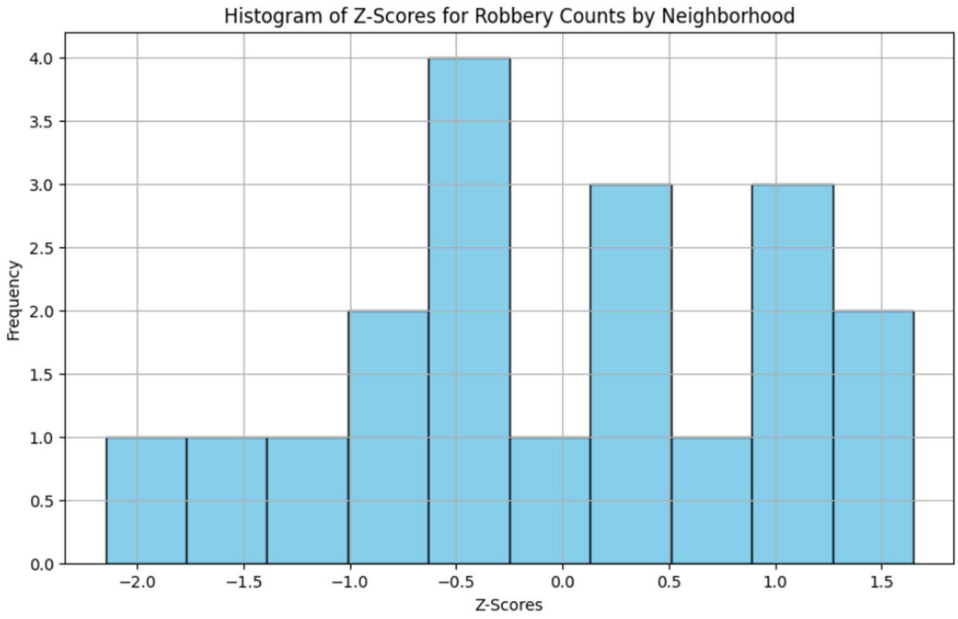


Figure 31 Robbery Count Z-Score Conversion

- After this, the z scores will be mapped between values of 1 and 5 using linear transformation, this will be done for clarity purposes, making it easier for the user to observe the information. The mapped outputs can be seen in the figure below. For clarification purposes, the information represents robbery scores only, not an overall crime safety score.

```
Druid Hill Park: 2.85
Mount Washington: 1.31
Clifton: 2.23
Raspeburg: 3.46
Arlington: 4.38
Highlandtown: 1.92
Fells Point: 2.54
Govans: 4.08
Carroll: 3.15
Govans: 1.62
Fells Point: 3.77
Harwood: 2.23
Sharp Leadenhall: 2.85
Hampden: 1.00
Arlington: 3.46
Raspeburg: 5.00
Arlington: 2.23
Charles Village: 2.54
East Case: 4.08
```

Figure 32 Linear Transformation of Z-Scores

- In addition to the z score data, users will be able to see an overall crime safety score as well. To calculate an overall score for each property, each value between 1 and 5 will be multiplied by a weight. The reason this is done is because certain crimes are more severe than others. For example, a neighbourhood may have a low crime count of 10 for each of its crime categories, however this does not reflect the safety of the area as one of those categories is homicide. The weights are determined using research from the following papers: (Adriaenssen et al., 2018), (Kwan et al., 2000), and (Borg et al., 2022). These studies involved surveying participants to assess their rankings of crime information in terms of severity. Subsequently, statistical tests were employed to generate a ranked output based on the survey results.

- Once each value has been multiplied by its according weight, the final values will be aggregated to get an overall score. Since, this score needs to reflect crime safety based on the other neighbourhood data in Baltimore, it will be normalised using Min-Max normalisation, putting its value between 0 and 1. The value will then be given out of 100 so it is clearer to the user, and so that they understand it easily.

- The second score will provide property nearby service information. The different service categories used for the project can be viewed in the figure below.

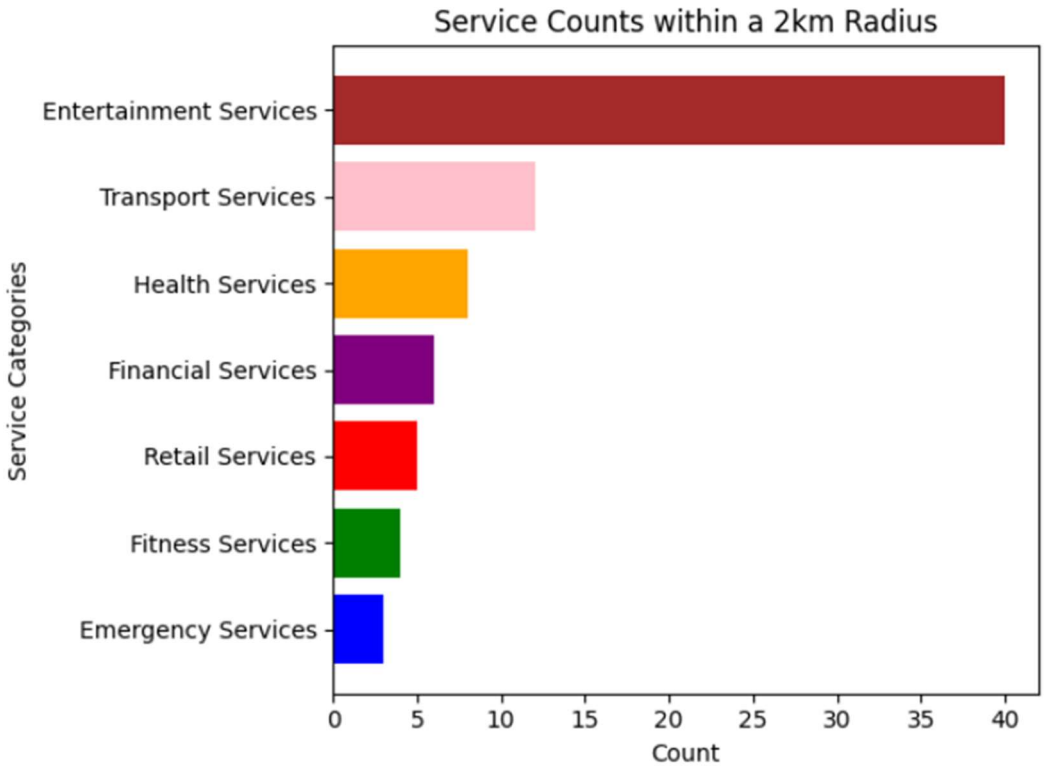


Figure 31 Service count with 2km radius

- The same approach will be used to calculate an overall score; however, the weight values will be determined differently. Instead, they will be determined by the user. They will be asked to rank service categories in terms of their interest, with the most valued service having a score of 7, and the least valued service having a score of one.

Property Recommendation system:

- The recommendation will use a content based filtering approach, incorporating a KNN algorithm.
- Instead of using property click information that Zillow implements, this project will use user like information instead. This allows us to ensure that the user is interested in a property.
- Collaborative filtering approaches that would use just like information were considered. This would involve the use of the cosine similarity formula and the matrix factorization algorithm, which would use latent factors to reveal the similarity of properties. However, these systems face the cold start problem and rely on pattern data, that would be generated from users. As a future product they can be implemented but for the scope of this project, content filtering will be used only.
- The advantage of using a content-based approach instead is that as little as one like from a user to make recommendations, addressing the cold start problem.
- Using property attribute information and applying it to their respective likes, properties with similar attributes can be found using k nearest neighbours. These properties can then be recommended to the user.

Figure 3233 content based filtering

7 EVALUATION

7.1 DEFINITION OF SUCCESS

The project is composed of three major components, each with their own requirements, and thus success criteria. In general, the project objective is to provide the MVP features and the custom features defined previously.

Frontend Success Criteria

- Improved usability over existing sites
- Improved aesthetic rating over existing sites
- Minimal accessibility errors
- Good site browser performance

Backend Success Criteria

- Fully implemented API endpoints for properties, user profile, scoring and recommendation systems.
- Securely implemented authentication and authorization.
- Implemented security measures to protect sensitive data.
- Implemented high test coverage for the code base.
- Implemented monitoring and logging for debugging and application health and performance.
- Provided comprehensive documentation.

Data Success Criteria

- Cleaned, formatted, and pre-processed data available to be used for project.
- Adhered to data privacy and data management best practices and policies.
- Provided comprehensive documentation.
- Created a recommendation system that uses the features from a user's selected properties as input.
- Implemented a database that is easy to query and provides quick responses.

7.2 EVALUATION STRATEGIES

Usability

Usability is a core concern for the system. The main issues highlighted by the personas described previously are centred around problems affecting the usability/user-friendliness of existing systems. Therefore, H.M.E aims to improve upon this. To evaluate usability, a comparative Task Journey Analysis will be conducted between the proposed system and Zillow. This process will generate quantitative results from which to draw a quantitative comparison. Preliminary Usability has already been assessed through a LFP testing procedure to identify major issues early on. Results of that evaluation have been implemented in the MFP (medium fidelity prototype). The competitive analysis will be conducted using the MFP, following the same testing procedure against the two systems. The procedure involves the following steps:

- The test participant is presented with a slideshow of screenshots of the UI of each website. They are required to perform a sequence of instructions. They do not progress the slides until instructed to.
- The user is asked to complete a task. E.g., "Add this item to your favourites". They will then hover their mouse cursor over the relevant element within the screenshot of the current slide.

- After indicating the correct element, the user is asked how difficult it was to complete the task on a scale from 1-5 (easy-difficult). They are then asked to progress to the next slide.
- This process is repeated for multiple instructions. Each instruction represents a step in a core user sequence.
- Each participant's session is recorded for further analysis.

This procedure is a valid application of the Single Ease Question (SEQ) post-task questionnaire strategy for user testing, although it has been modified slightly to use a scale of 1-5 rather than 1-7. The suggested number of test subjects should be ~8. Success will be indicated by a higher average usability rating across all sequences on the proposed system. (Experience, n.d.).

In addition to the above methodology, to ensure that the icons used throughout the interface align with user's expectations and mental models, a survey has been conducted to validate the choice of iconography used throughout the site. If the iconography is more in line with user's expectations, it should go some way to supporting a higher usability rating.

Aesthetics

Aesthetic evaluation is an important aspect of a good user experience, and it has been demonstrated to also contribute to higher self-reported usability amongst testers (Sonderegger & Sauer, 2010). This is commonly referred to as the 'Aesthetic Usability Effect' (Experience, n.d.).

To assess the aesthetics of the app, a single Likert scale questionnaire will be utilised. Users will be asked to rate 'how pretty and/or nice does the site look' for both the proposed system and Zillow. Success is determined if the average score for the proposed system is higher than Zillow.

Accessibility

Due to limitations on time, a focus group for accessibility testing for different disabilities is unfeasible. However, automated accessibility testing tools exist for evaluating interfaces in the Chrome browser as extensions, notable WAVE (*WAVE Web Accessibility Evaluation Tools*, n.d.).

By using this tool, accessibility issues can be identified. There should be no serious errors/contrast errors on the site, and features and ARIA element counts should be maximized as much as possible.

Browser Performance

In addition to the accessibility and usability evaluations, the Lighthouse Chrome extension will be used to evaluate browser performance (*Lighthouse*, n.d.). This is an automated tool that gives ratings on Performance (Network and Code), accessibility, Best Practices, and Search Engine Optimization. Search Engine Optimization is important; however, it is not a primary concern for this project. Performance will be the main metric this tool will be used for.

Test Coverage

Both front- and backend are using the Test Later Development (TLD) framework. This is when as the name implies, tests are implemented after the code is written. This is allowing for better edge case testing, fewer trivial tests and shorter development time (Hu, 2022). Using Istanbul for react and JaCoCo for spring boot, the project aims to have ~80% test coverage for the codebase.

In addition, the following testing strategies will be utilized:

- Automated scanning tools that can identify common vulnerabilities such as Postman for API testing and/or Sonarqube.
- Test authentication and authorization to ensure users can only access permitted data and functionality.
- Regularly check dependency vulnerabilities.
- Regularly monitor system resources such as memory usage, database execution times and CPU utilisation.
- API benchmarking and external API evaluation by monitoring response time, memory usage, availability, and rate limits.

Recommendation system

The recommendation system using content-based filtering will be evaluated using Accuracy, Precision, Recall, and F1 score classification metrics:

- Accuracy: the measure of the proportion of correctly classified instances out of the total instances in the dataset.
- Precision: the ratio of true positive predictions (predictions correctly identified as positive) to the total number instances predicted as positive (which could contain incorrect predictions), positive meaning 1 while 0 would be negative, each representing a category. In this case there will be more than 2 categories.
- Recall: a measure of the ratio of true positive predictions to the total number of positive instances (actual positive instances).
- F1-Score: a harmonic average of precision and recall.

While these metrics in theory could provide accurate measurements for the recommendation system. The results may not be consistent with reality. To get a better understanding of the success rate of recommendation system, it will be manually observed if recommended properties are similar to each other. In addition, , a prompt asking users how satisfied they are with their recommendations will appear to give user feedback.

8 CONCLUSION:

8.1 PROJECT MANAGEMENT STRATEGY

Throughout the development of HME, a SCRUM methodology has been implemented. This strategy will be continued for the remainder of the project. The project manager alternates each week, and a variety of technologies are used for team management.

Discord: Daily standups on discord serve multiple purposes such as, monitor progress, problems, and project submissions. Discord also allows for impromptu discussions for additional clarification or minor issues.

GitHub: The project repository is located here for version control, and adhere to the four-eyes principle for pull requests and issues etc.

Trello: Is used for splitting tasks and used to visualise what is in progress, completed, pending etc.

8.2 FUTURE CHALLENGES

The core challenges can be listed as follows.

- Query the database from the backend.
- Further develop REST API.
- Connect the frontend to the backend.
- Implement a recommendation and scoring system.
- Continuously adapt UX to reflect user feedback.

8.3 SUCCESSFUL OUTCOME

Section 5.1 discusses the success criteria for each sub-section of the project. These goals can be achieved by ensuring the following timeline (table 2) thus creating a successful project.

Table 2 – Project timeline (blue = completed, white = pending)

Week No.	Dates	Activity	Comments
2	25 Sept – 29 Sept	Project plan Frontend: react and UX research Backend: framework and deployment research and experimentation DS: Datasets and Database research and exploration	
3	2 Oct – 6 Oct	Frontend: Initial frontend layout, set up routing, nav bar etc. Backend: Authentication and authorization. DS: Data extraction research, exploratory data analysis and database design	
4	9 Oct – 13 Oct	Frontend: Continuation with page layouts mock property data created. Backend: Spring Boot deployment and properties API endpoint DS: Data extraction and Database creation	Getting Google Places API to work took longer than expected
5	16 Oct – 20 Oct	Frontend: Frontend deployed and connected to Cognito. Backend: Properties endpoint and unit testing, connect to database DS: Data cleaning, preprocessing and database testing	Delays because API properties endpoint was broken.
6	23 Oct – 27 Oct	Interim Report Frontend: Minor layout changes.	

		Backend: Connect to database DS: Recommendation system research and report documentation. Collaboration to the interim report.	
7	30 Oct – 3 Nov	Frontend: Connect to backend Backend: Database queries, API Gateway authorization, implement AWS parameter store DS: Recommendation system planning and support backend team in terms of connection and access. Provide optimized queries to back end and test spatial data implementation and search.	
8	6 Nov – 10 Nov	Frontend: Implement initial user feedback and start recommendation page Backend: User profile endpoint, JUnit testing DS: Recommendation system creation and loading datasets into database and making dashboards for better insights.	
9	13 Nov – 17 Nov	Frontend: Additional user testing and refactoring. Backend: Scoring endpoint, Junit testing DS: Recommendation system evaluation and comparison. Verify insights with team.	
10	20 Nov – 24 Nov	Frontend: Refactoring Backend: Recommendation endpoint, Junit testing DS: Refine Algorithms if needed, verify insights, and create more detailed graphs.	
11	27 Nov – 1 Dec	Frontend: Final user testing Backend: Testing, Refactoring DS: Review of the recommendation system	
12	4 Dec – 8 Dec	Frontend: Final touch-ups. Backend: Refactoring DS: Bug fix if needed. Document research and approaches for final presentation.	

9 REFERENCES AND KEY RESOURCES

Adriaenssen, A., Paoli, L., Karstedt, S., Visschers, J., Greenfield, V. A., & Pleysier, S. (2018). Public perceptions of the seriousness of crime: Weighing the harm and the wrong. *European Journal of Criminology*, *17*(2), 127–150. https://doi.org/10.1177/1477370818772768

Adult education - CSO - central statistics office. CSO. (2020, August 19). https://www.cso.ie/en/releasesandpublications/ep/p-sdg4/irelandsunsdgs2019-reportonindicatorsforgoal4qualityeducation/adulteducation/#:~:text=The%20proportion%20of%2015%20year,of%20the%20population%20were%20students

Borg, I., Hermann, D., & Bilsky, W. (2022). The perceived seriousness of crimes: Inter-individual commonalities and differences. *Quality & Quantity*, *57*(1), 765–784. https://doi.org/10.1007/s11135-022-01379-9

Daft.ie is rated "bad" with 1.7 / 5 on Trustpilot. Trustpilot. (1969, December 29). https://ie.trustpilot.com/review/daft.ie

Dauni, P., Firdaus, M. D., Asfariani, R., Saputra, M. I., Hidayat, A. A., & Zulfikar, W. B. (2019). Implementation of haversine formula for school location tracking. *Journal of Physics: Conference Series*, *1402*(7), 077028. https://doi.org/10.1088/1742-6596/1402/7/077028

Experience, W. L. in R.-B. U. (n.d.). *Beyond the NPS: Measuring Perceived Usability with the SUS, NASA-TLX, and the Single Ease Question After Tasks and Usability Tests*. Nielsen Norman Group. Retrieved 26 October 2023, from https://www.nngroup.com/articles/measuring-perceived-usability/

Experience, W. L. in R.-B. U. (n.d.). *The Aesthetic-Usability Effect*. Nielsen Norman Group. Retrieved 26 October 2023, from https://www.nngroup.com/articles/aesthetic-usability-effect/

Experience, W. L. in R.-B. U. (n.d.). *The Negative Impact of Mobile-First Web Design on Desktop*. Nielsen Norman Group. Retrieved 26 October 2023, from https://www.nngroup.com/articles/content-dispersion/

Google. (n.d.). Google maps platform | google for developers. https://developers.google.com/maps

GroupVisual.io. (2018, June 19). *Personas vs. roles (how we research our users)*. Medium. https://medium.com/@groupvisualio/user-roles-vs-personas-246cc0a0a93d

Heddings, A. (2020, April 3). *What Are Lambda Functions, And How Do You Use Them?* How-To Geek. https://www.howtogeek.com/devops/what-are-lambda-functions-and-how-do-you-use-them/

Home ownership and rent - CSO - central statistics office. CSO. (2023, October 20). https://www.cso.ie/en/releasesandpublications/ep/p-cpp2/censusofpopulation2022profile2-housinginireland/homeownershipandrent/

Houses of the Oireachtas. (2022, September 28). *Seanad éireann debate - Wednesday, 28 Sep 2022*. House of the Oireachtas. https://www.oireachtas.ie/en/debates/debate/seanad/2022-09-28/12/

Hu, T. (2022, September 19). *Test-driven vs. test-later development: When should you write your tests?* Codecov. https://about.codecov.io/blog/test-driven-vs-test-later-development-when-should-you-write-your-tests/

Introduction to recommendations at Zillow - Zillow Tech Hub. (n.d.). https://www.zillow.com/tech/introduction-recommendations-zillow/

Konadu, E. (2021). *Determinants of international students' choice of accommodation in Ireland: A study of National College of Ireland.* [PDF *Determinants of international students' choice of accommodation in Ireland: A study of National College of Ireland.*]. https://norma.ncirl.ie/5500/1/erickonadu.pdf

Kwan, Y. K., Ip, W. C., & Kwan, P. (2000). A crime index with Thurstone's scaling of crime severity. *Journal of Criminal Justice*, *28*(3), 237–244. https://doi.org/10.1016/s0047-2352(00)00039-8

Lighthouse. (n.d.). Retrieved 26 October 2023, from https://chrome.google.com/webstore/detail/lighthouse/blipmdconlkpinefehnmjammfjpmpbjk

Perera, A. (2018, April 18). Exposing AWS Lambda functions with the API Gateway. *Medium*. https://asankha.medium.com/exposing-aws-lambda-functions-with-the-api-gateway-fbdc15c19dce

Part 1 crime data. Open Baltimore. (n.d.). https://data.baltimorecity.gov/datasets/baltimore::part-1-crime-data/explore

Policies for Places API. (n.d.). Google for Developers. Retrieved 26 October 2023, from https://developers.google.com/maps/documentation/places/web-service/policies

Sonderegger, A., & Sauer, J. (2010). The influence of design aesthetics in usability testing: Effects on user performance and perceived usability. *Applied Ergonomics*, *41*(3), 403–410. https://doi.org/10.1016/j.apergo.2009.09.002

S T U D E N T S - higher education authority. (n.d.). https://hea.ie/assets/uploads/2022/03/2022-Info-Byte-International-Students-Final.pdf

The 7 Principles | Centre for Excellence in Universal Design. (n.d.). Retrieved 26 October 2023, from https://universaldesign.ie/what-is-universal-design/the-7-principles/

The RTB Rent index Q1 2023. The RTB Rent Index Q1 2023 | Residential Tenancies Board. (n.d.). https://www.rtb.ie/about-rtb/data-insights/data-hub/the-rtb-rent-index-q3-2022

WAVE Web Accessibility Evaluation Tools. (n.d.). Retrieved 26 October 2023, from https://wave.webaim.org/

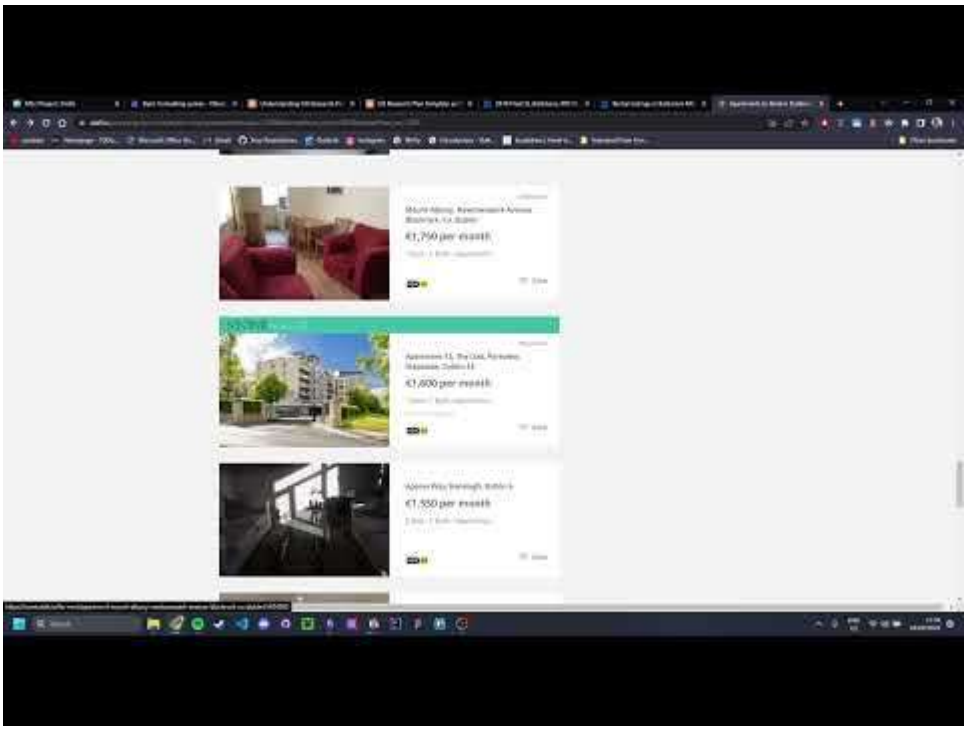
What is Universal Design. Centre for Excellence in Universal Design. (n.d.). https://universaldesign.ie/what-is-universal-design/

Xia, V. (2022, March 24). What is Mobile First Design? Why It's Important & How To Make It? *Medium*. https://medium.com/@Vincentxia77/what-is-mobile-first-design-why-its-important-how-to-make-it-7d3cf2e29d00

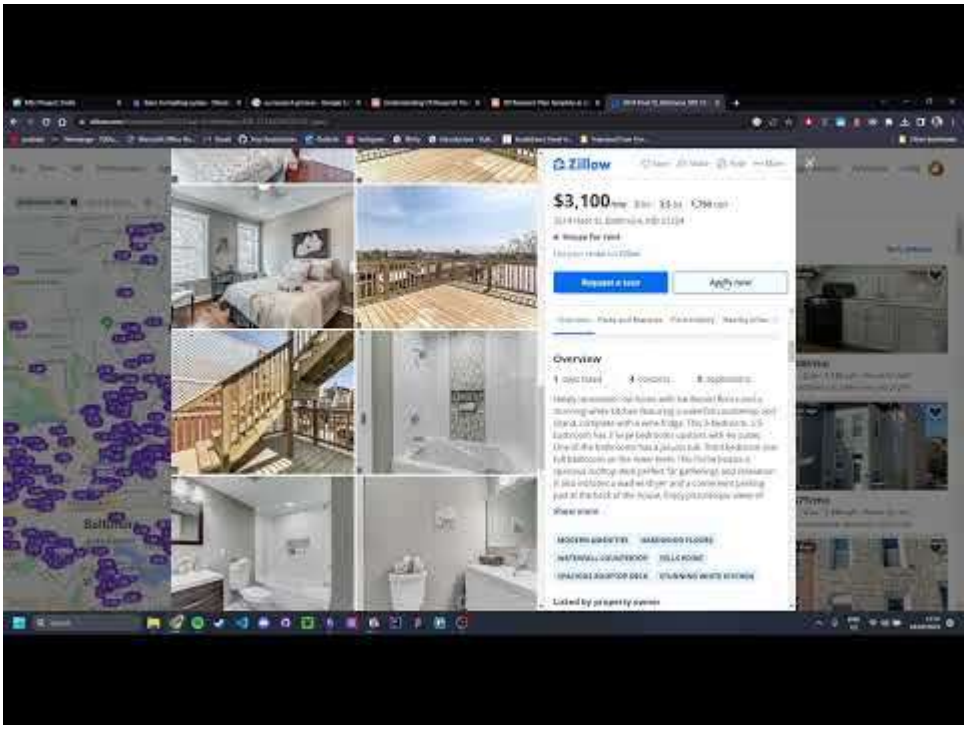
Zillow.com API documentation (apimaker): Rapidapi. Rapid. (n.d.). https://rapidapi.com/apimaker/api/zillow-com1/

10 APPENDIX

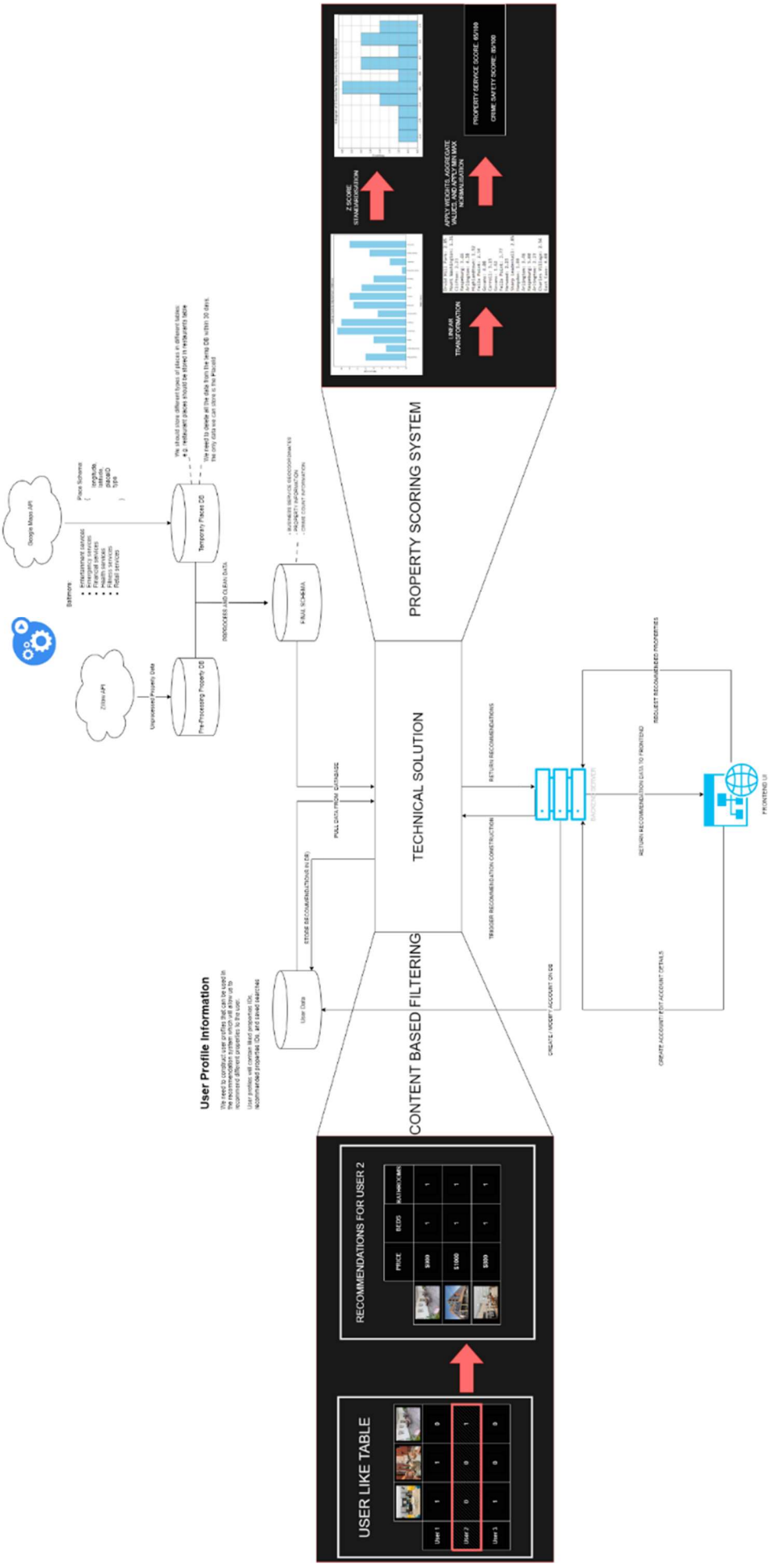
A. Daft.ie Demo Video



B. Zillow.com Demo Video



C. Data Engine Architecture Diagram



D. Databse Scehma