Table Of Contents

1.1) ScholarHomes Overview

We've crafted the ScholarHomes system as a cutting-edge online platform that aims to bring together Students, simplifying the tiresome process of searching for student accommodation and creating leases. Our feature-rich portal is designed to not only facilitate a meaningful connection between students, but also to empower students in making informed decisions about their accommodation choices. The platform is separated into two distinct sides, each catering to the specific needs and functionalities of both Listers and Seekers. Listers can effortlessly post and manage leasing through one interface, while Seekers are provided with an intuitive portal to explore property listings and make well-informed selections that align with their preferences and requirements.

This application is designed to increase the number of accessible student accommodation by opening the market to private sectors by allowing students who currently have accommodation to rent to other students. For example, a student living in a two bedroom apartment looking for a flatmate, to post listings that describes not only the accommodation type they have available e.g. a shared room or a single room; But also displays the type of tenant they are seeking. Strict academic students may seek a more introverted tenant to avoid disruptions in their accommodation, whereas a shared student accommodation consisting of several different students may be seeking a more extroverted tenant who can better integrate into a communal living environment. This also benefits students as it provides them with more context about the accommodation before committing to a lease. International students may seek to stay with a more traditional accommodation option to further immerse themself in Irish culture during their visit.

We recognize the importance of appealing to students and conform to their standards of quality digital landscape. By incorporating familiar features reminiscent of popular social media platforms, such as user profiles, the ability to follow other users, and track followers, we create an engaging and intuitive environment tailored to resonate with our target demographic. Leveraging these elements not only enhances user experience but also cultivates a sense of community and connectivity, ultimately enriching the overall platform experience for students seeking community and accommodation. This combined with our sleek and ease to use User Interface has granted ScholarHomes a competitive edge of our competition.

1.2) Organisational Structure

Current Team Composition: During the development of our first prototype we have operated as a team of two. While it was taxing to plan, develop and document the first prototype with such a small team, it was an ideal composition for this stage. As a small team we were able to quickly come to a consensus on what type of application we wanted to create, how to go about creating the product we envisioned as well as who it is targeted towards.

Given that the initial idea and requirements and ideation of the application is pivotal in this phase, it was beneficial to work as a pair rather than a large team. This did make the development of the prototype arduous as we had limited manpower to create several key features that were fundamental to establishing the foundation of our prototype. With hard work and perseverance, we managed to overcome this struggle and can now seek to employ more resources to our development team going forward to ensure a more rapid development schedule.

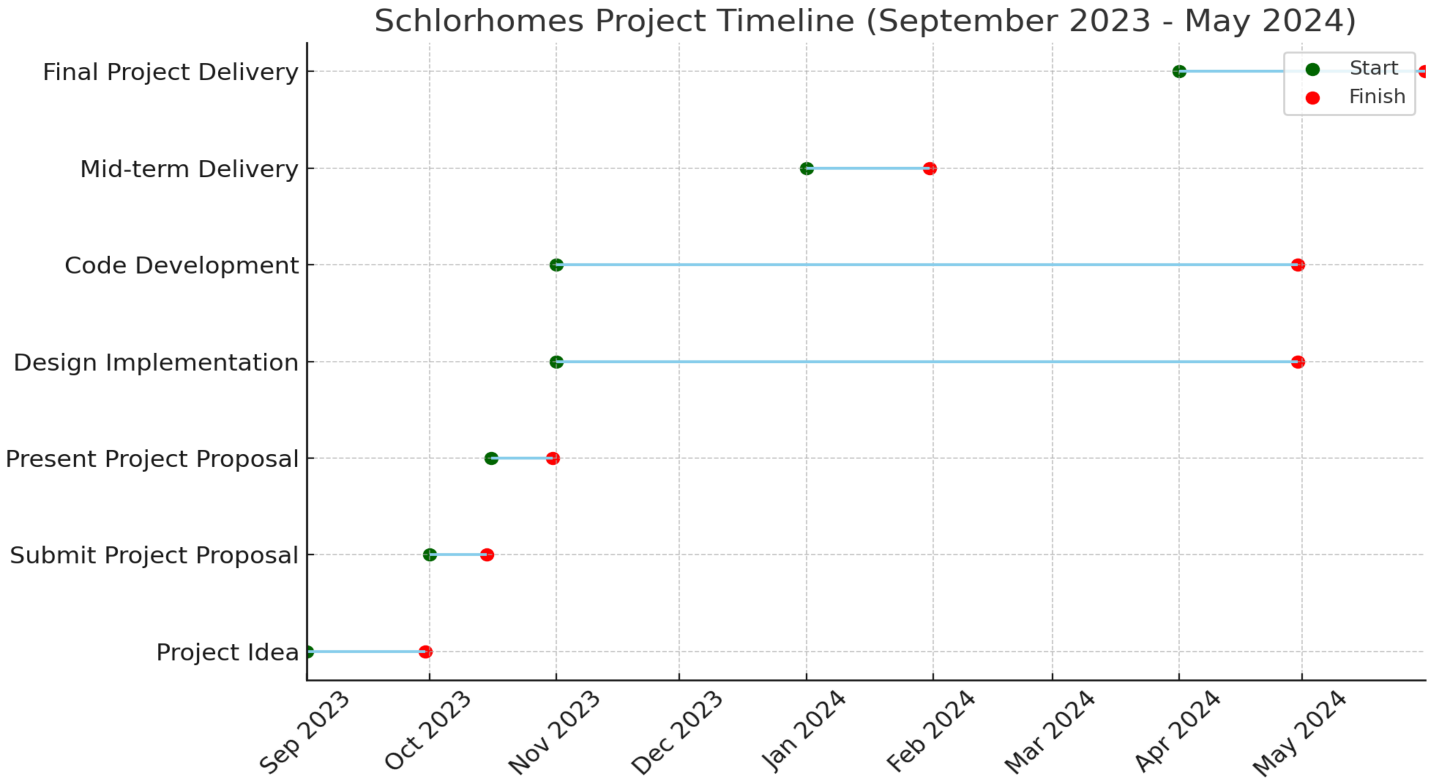
Desired Team Composition for Second Prototype: Going forward we wish to expand our project team to contain 5-10 members conforming to the standard of agile teams. During the iteration of our second prototype we hope to focus on development, refining features already in place and developing new features that we lacked the resources to develop in our first prototype.[1]

As we have already firmly established our requirements and have clearly defined the direction and features we intend to include, we feel there is a low risk associated with proceeding to focus on development in the next phase of this project. Normally there is a considerable risk in taking this approach as it is significantly easier to fix a requirement in the requirement planning phase as opposed to after development has begun. Both members from the first phase will be active participants in the second phase to ensure there is no ambiguity in the requirements and should the development team have any inquiries, they can provide clarity. We will continue to practise agile methodologies to ensure rapid development without compromising the integrity of our requirements. Agile methodologies will be beneficial going forward as its flexibility will allow us to adjust our design and features based on feedback we have received from the first prototype. Daily standups will ensure any issues members of the team encounter are addressed and to monitor and adjust the project timeline as needed.[2]

[1] *Agile teams (2024) Scaled Agile Framework.*

[2] *Development of a human error taxonomy for software requirements: A systematic literature review*

1.3) Project Timeline



Project Idea:

- Came up with a project idea (ScholarHomes) and made a plan for the project's theme and goals.
- Wrote down the most important wants, goals, and possible outcomes.
- Made a rough plan for the project's goals and outputs.

Submit Project Proposal:

- We wrote a detailed proposal that described the project's goals, methods, and boundaries.
- We described the effects that are anticipated and the tools it will need.
- Send the plan to the project advisors so they can review it and give it feedback.

Present Project Proposal:

- Prepared a presentation to communicate the project's vision to project advisors.
- Addressed questions and concerns to gain support or necessary modifications.
- Finalised the project details based on feedback received.

Design Implementation:

- We made thorough drafts and plans for the parts of the project.
- We made an architectural or functional plan to help with future development.
- We made sure the plans were good by talking to the project advisor.
- Implemented new designs and plans as the project is ongoing.

Code Development:

- We turned our ideas into software that works.
- Incorporated feedback from users and advisors into the design process.
- Focused on both front-end and back-end development to make sure the site is stable and easy to use.

Mid-term Delivery:

- Presented the interim results to project advisors.
- Collected detailed feedback to refine the project further before the final delivery.

Final Project Delivery:

- Completed all development, design and refinements based on the mid-term feedback.
- Prepared comprehensive documentation and user manuals.
- Officially submitted the project and handed it over to the project advisors accompanied by a final project report.

1.4) Ryan - Developer Log

Initial Build (Separate Repository From Final Year Project Git Repository):
https://gitlab.computing.dcu.ie/shannor8/2024-ca4094-ScholarHomes/

- **Nov 12th**, **2023**: Implement dual user authentication (login), URL routing and user creation forms
- **Nov 13th**, **2023**: Apply to property form added, Property creation form and View properties page
- **Nov 14th**, **2023**: Property images added
- **Nov 18/19th**, **2023**: CSS styling implemented, frontend design
- **Nov 20th**, **2023**: Student and Landlord dashboards added, Navigation bar implemented
- **Nov 22-30th**, **2023**: Content added to website pages, frontend CSS styling/polishing formatted
- **Dec 07, 2023**: Code Commented

Final Year Project Build:
https://gitlab.computing.dcu.ie/shannor8/2024-ca472-shannor8-kafij2/

- **Jan 21**, **2024**: Implemented previous build onto new repository
- **Jan 23**, **2024**: Responsive message patch added using Javascript
- **Jan 28**, **2024**: User Profiles added, Profile Models and degree added and user Model adjusted
- **Feb 05**, **2024**: Lister and Seeker Profiles completed and additional profile functionalities added, footer implemented
- **Feb 08**, **2024**: Student oriented update added, content altered to be student specific, Newly added code commented, frontend css added
- **Feb 25**, **2024**: property view count added, date created added, filter bar added for property list
- **Mar 10**, **2024**: New Pages and features styled
- **Mar 11**, **2024**: Google map API implemented
- **Mar 13**, **2024**: Google map API patched, Follow functionality added

- **Mar 19** , **2024**: View followers/following added
- **Mar 20**, **2024**: Additional patched made to filter bar, View following/followers fully patched and styled
- **Mar 21**, **2024**: Product page UI updated
- **Mar 29**, **2024**: View profile routing error resolved
- **Apr 17**, **2024**: Fronted edit profile form styling, following/unfollow property bug patch
- **Apr 19-20**, **2024**: Styling/bug fixes
- **Apr 21, 2024**: Delete button added to My Applications and My Properties, Accept/Reject applicants functionality added
- **Apr 21, 2024**: Project finalised with comments
- **Apr 22, 2024**: Additional comments added

1.5) Jomi - Developer Log

Initial Build (Separate Repository From Final Year Project Git Repository):
https://gitlab.computing.dcu.ie/shannor8/2024-ca4094-ScholarHomes/

- **Nov 17, 2023**: Communication between both dashboards
- **Nov 20, 2023** : View Applications page
- **Nov 20, 2023**: Fixed errors on view application page
- **Nov 21, 2023**: View Applicants page
- **Nov 24, 2023**: Changes to lister dashboard template and property list template
- **Nov 27, 2023**: Adding applications feature to lister dashboard and small error handle

Final Year Project Build:
https://gitlab.computing.dcu.ie/shannor8/2024-ca472-shannor8-kafij2/

- **Feb 11, 2024**: Changes to the student profile page
- **Feb 24, 2024**: Error handled for student registration page and update to student profile page
- **Feb 27, 2024**: Creation of application status for student and Accept and decline feature for listers
- **Mar 13, 2024**: Created new degrees and universities for the student and lister registration
- **Mar 14, 2024**: Worked on lister profile adding profiles for test data
- **Mar 21, 2024**: Data implementation for the project 11 new lister profiles added and 11 new properties all added to the site.
- **Mar 29, 2024**: Created edit profile feature for all users
- **Apr 01, 2024**: changes to edit profile, fix non-displaying property details
- **Apr 02, 2024**: bug fix for edit profile and view profile
- **Apr 18, 2024**: receipt added to accepted applications
- **Apr 21-22, 2024**: Code commented

2.1) Product Description

ScholarHomes is a creative app for finding and renting student housing that is meant to make the process easier. This all-in-one tool not only makes it easy for landlords (students who have listed the property) and student seekers to talk to each other, but it also makes sure that the transaction process is clear and quick.

<u>The Landlord's Listing Process</u>: On ScholarHomes, the journey starts when a landlord posts a listing for a property. The process is meant to be simple and easy to use; This allows landlords to put all the important details about the accommodation, such as its location, price, amenities, and photos in one convenient location for viewing. Our application encourages landlords to give the full details so that they can find suitable tenants that suit their interests. After the listing is complete, it goes through a quick validation check to make sure the information is correct and honest according to ScholarHomes' standards. This first step is very important because it sets the stage for finding compatible student renters and makes sure that the listings are both transparent and appealing.

<u>Student Viewing and Application Process</u>: Once the post goes live, students who are looking for housing can view these detailed listings through an easy-to-use search interface that lets them filter properties based on things like location, price, and amenities. If a student finds a post that fits their needs, they can use the app to directly apply for it. As part of the application process, students are asked to provide a previous reference that verifies them as an acceptable tenant. ScholarHomes uses this information to make sure that the student and landlord can trust each other. This step is very important because it's when the possible tenant and the property owner start talking to each other in a formal way.

<u>Landlord's Application Review and Response</u>: When the landlord gets an application, he or she can look at the student's information and past references (if given). ScholarHomes makes this review process easier by giving the landlord all the information they need to make an informed choice. The owner can then decide whether to accept or reject the application based on their own rules. This process for making decisions is very important because it decides if the talks and deals move on to the next step.

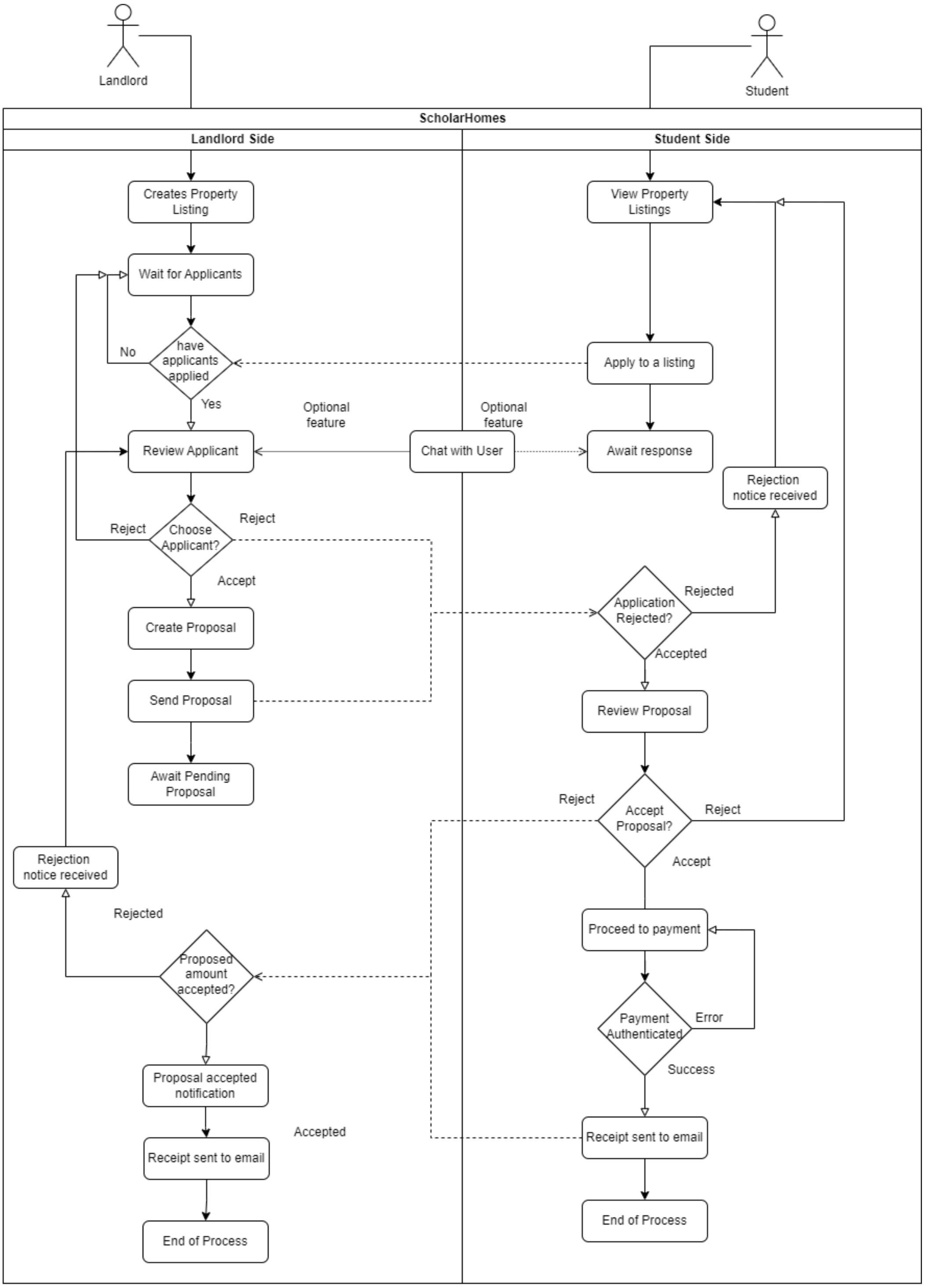
<u>Proposal by the Landlord</u>: If the landlord agrees to rent to a student, they can then make an official offer through the app. Usually, this plan spells out the details of the lease, like the rent, the amount of the security deposit, how long the lease will last, and any other conditions that are important. ScholarHomes makes sure that this proposal is written in a way that is legal and easy to understand by giving renters.

<u>Property Viewing and Final Agreement</u>: If the student accepts the offer, they should go see the property to make sure it meets their needs and to make sure the terms in the offer are still valid. This is something that ScholarHomes suggests be done in person to avoid any confusion or disagreements in the future. After a good viewing of the property, if everyone is still okay with it, the landlord makes a final payment proposal through the app.

In conclusion, ScholarHomes not only simplifies the process of renting student accommodation but also ensures that every step—from listing creation to move-in—is handled with the utmost efficiency and security. By providing an all-encompassing, easy-to-navigate platform, ScholarHomes stands as a pivotal solution for landlords and students, streamlining the rental process and enhancing the overall experience for users across the board.

2.2) User Flowchart

This visualisation offers some insight into the business process that takes place between landlords and students. This diagram also provides some context into what happens when scenarios do not go entirely to plan, e.g. a student does not agree to the proposal the Landlord has made. The optional chat feature was initially intended to be a built in chat function, but was downsized to chatting through email for the initial prototype. This will allow us to focus on the chat feature when implementing our second prototype.



2.3) Validating the Market

We understand that there is a clear market for students who are seeking accommodation. This is evident by the current lack of available student accommodation in Ireland. There have been numerous attempts by students to get the government to invest in accommodation for students. This can be seen in a protest held in Dublin city centre on the 4th of October 2023 which approximately 300 students attended. These students demanded the government lower the student contribution fee and build student accommodation. This is a clear sign that there is a current need in the market for additional accommodation specifically for students. We believe our application fits this niche.[3]

We are confident that students will be open to sharing accommodation. Currently, the average rent in Ireland surpasses the peak of the Celtic Tiger era by more than 50%. This presents a lucrative opportunity for landlords, as they can profit significantly by renting a single room to a student for the duration of an academic year, spanning 7 to 8 months. This is particularly noteworthy given that Ireland is currently one of the most expensive countries to reside in within Europe. To illustrate, Irish goods are 40% pricier than the average cost of goods in the EU. This places a tremendous financial strain on students, however this burden can be alleviated somewhat should they choose to live in a shared space or rent a room in their current accommodation.[4]

2.4) Market size

Through rigorous research, we have determined the target market for students to be 202,100. This calculation involved subtracting the existing student accommodations from the total number of students enrolled in college annually. To uphold the accuracy and relevance of our findings, we relied on verified sources specialising in Irish student data. Additionally, we ensured that the statistics utilised were gathered within the last 3 years, emphasising the reliability of our analysis.

We used statistics directly taken from the higher education authority website. The higher education authority (HEA) is a government run organisation that works on researching and developing higher level education in Ireland. In 2021, the HEA recorded the number of students enrolled in college to be 245,600. This figure has shown itself to be increasing over time based on HEA statistics, meaning our target market may be even larger than predicted. The HEA also estimated that the number of student accommodation available for the year 2024 to be under 43,500. As this is an overestimation of the amount of student accommodation available, it can be presumed that our target market will be larger than the figure we predict. Through a methodical approach, we have calculated our target market by subtracting the available accommodation from the total number of enrolled students. This calculation indicates a substantial market size of at least 202,100 students.[5]

[3] *RTE.ie, Students protest in Dublin over accommodation crisis*
[4] *Joe.ie, Average rent in Ireland is now 50% higher than Celtic Tiger levels*
[5] *Higher education – key facts and figures 2020/2021*

2.5) Primary Research Carried Out

Pre Research Criteria:

We conducted research specifically targeting students through a hybrid interview system employing Google Forms and in-person interviews. Focusing on the student demographic, we reached out to various contacts including international students we engaged with in classes and clubs/societies, as well as Irish students who relocated to Dublin due to distance from their universities. Our primary method of interaction with students involved brief, informal in-person discussions lasting about five minutes. Key discussion points included:

- "What were the major challenges you faced while searching for student accommodation?"
- "What factors influenced your choice of accommodation?"
- "How did you go about booking your accommodation?"

Participants were also given opportunities to share personal experiences and insights. For those we couldn't meet in person, such as international students who had returned home, we provided Google Forms with similar questions and additional sections for insights.

Primary Research Findings:

Our findings reinforced the challenges surrounding student accommodation in Ireland. Six out of nine students cited difficulty in finding accommodation as the most challenging aspect of their application process. Many international students struggled to secure accommodation outside of university-provided services, raising concerns about accommodation scarcity. Some students, faced with limited options, paid premium rates months in advance to secure accommodation.

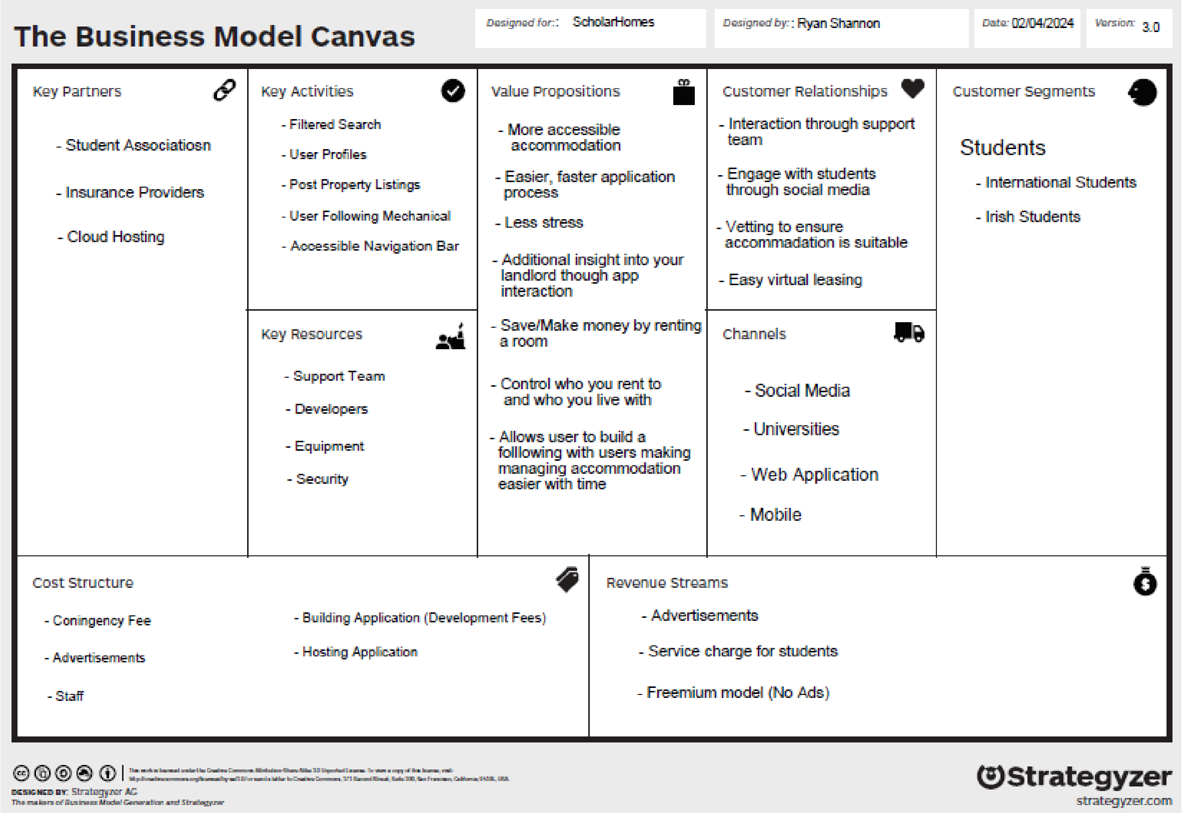
Interestingly, a majority of international students expressed a preference for living with Irish residents, citing benefits such as language practice and local insights. The students who said this had positive experiences staying with Irish residents and one mentioned it made them feel more confident during their stay.

Interviews with people who leased their accommodation revealed reluctance to pay service charges for listing accommodations, citing alternative methods that do not incur a fee. Additionally, many landlords highlighted the challenge of compatibility between tenants, emphasising the importance of genuine connections.

Our platform, ScholarHomes, addresses these concerns by facilitating authentic connections between students listing accommodation and students seeking accommodation, easing anxieties about leasing to a new tenant.

2.6) Business Model Canvas

From a business perspective there are several income channels we intend to tap into in order to generate a profit. Firstly, we intend to allow third parties to pay for advertisements to be placed on the site. These ads will be designed to not be overly intrusive for the user. If we create advertisements that are too invasive, it may discourage users from using the application. We also intend to implement a freemium model, allowing users to pay a small fee to avoid seeing these ads. By paying this fee, users will also unlock additional features that are exclusive to premium users. Some potential premium features we're considering implementing after launching our first prototype are, priority listings or applications, being able to upload videos on a listing and being able to apply to have your accommodation or student account verified.



The Business Model Canvas

Designed for:: ScholarHomes

Designed by:: Ryan Shannon

Date: 02/04/2024

Version: 3.0

Key Partners

- Student Associatiosn
- Insurance Providers
- Cloud Hosting

Key Activities

- Filtered Search
- User Profiles
- Post Property Listings
- User Following Mechanical
- Accessible Navigation Bar

Key Resources

- Support Team
- Developers
- Equipment
- Security

Value Propositions

- More accessible accommodation
- Easier, faster application process
- Less stress
- Additional insight into your landlord though app interaction
- Save/Make money by renting a room
- Control who you rent to and who you live with
- Allows user to build a folllowing with users making managing accommodation easier with time

Customer Relationships

- Interaction through support team
- Engage with students through social media
- Vetting to ensure accommadation is suitable
- Easy virtual leasing

Channels

- Social Media
- Universities
- Web Application
- Mobile

Customer Segments

Students

- International Students
- Irish Students

Cost Structure

- Coningency Fee
- Advertisements
- Staff
- Building Application (Development Fees)
- Hosting Application

Revenue Streams

- Advertisements
- Service charge for students
- Freemium model (No Ads)

DESIGNED BY: Strategyzer AG
The makers of Business Model Generation and Strategyzer

Strategyzer
strategyzer.com

In our initial BMCs we planned to have both Students and Landlords as customer segments, however we decided to narrow our focus solely to students in order to corner the market and give us an edge from our competition. In order to ensure we were appealing to this demographic, we altered our features to be similar to a social media platform. We emphasised the social element of our application by adding features like user profiles and a follow mechanic and kept our UI slick and consistent. This aligns with the basic features and principles used by social media applications today such as Instagram and Snapchat. Social media platforms are highly sought after by our target demographic, University Students, as shown in a study conducted by Campus Sonar where over 70% of students use Instagram and 65% of students use Snapchat.[6]

[6] *Social media demographics for higher ed—2021*.

2.6.1) Value Side

Customer Segments:

Students: The customer segment comprising university students seeking to study in Ireland is diverse and encompasses a range of needs and preferences. These students are typically divided into two main groups: international students coming from various countries to pursue their education in Ireland and local Irish students in search of suitable accommodation near their colleges or universities.

International Students: This subset comprises individuals from different corners of the world aspiring to undertake their studies in Ireland. They face unique challenges, such as navigating a new country, understanding cultural nuances, and finding suitable housing for the duration of their academic pursuits. Their accommodation needs often include proximity to their university, safety, affordability, and comfort. Providing resources, guidance, and tailored housing options becomes crucial in assisting these students to settle in and focus on their studies.

Irish Students: On the other hand, Irish students also form a significant part of this customer segment. While they might not face the challenges of relocating to a new country, they still seek appropriate accommodation near their colleges. Their needs may differ from international students but often include factors like convenience, proximity to campus, cost-effectiveness, and sometimes specific preferences for shared housing with fellow students. Irish students may also have already secured accommodation and are seeking a roommate to cut the expense of rent.

Grasping the distinct requirements of both these factions within the broader community of university students seeking housing in Ireland is crucial. Tailoring services to cater to their diverse needs, whether it's providing information on local culture, offering flexible housing options, ensuring safety measures, or creating community-focused living spaces can drastically enhance their overall experience and satisfaction during their academic journey in Ireland.

Customer Relationships:

Students: ScholarHomes will have a dedicated support team to assist students with any queries. This team is also equipped to assist users in resolving issues that may arise. The communication between the support team and students will primarily take place through emails, however calls can be organised when necessary. We will also engage with students through social media, promoting our application by interacting with the student community. We surveyed most students that regularly interact with communities they are a part of do so through social media. Students will also undergo a brief vetting process by our team to ensure they are qualified to use ScholarHomes. Should the student be listing accommodation we will also need to vet their accommodation and ensure it is suitable.

Channels:

Social media: We intend to utilise social media as an efficient way to increase our applications exposure to the public. It has been surveyed that both of our customer segments regularly interact with social media making this an ideal platform.

Universities: Our mission aligns with universities in aiding students throughout their academic journey. In this spirit, we seek partnerships wherein universities can refer students to us when their on-campus accommodation reaches capacity, ensuring every student has access to suitable housing.

Web Application: Users have the option to directly engage with the web application by entering its URL into their preferred search engine. This can be done through a search browser such as Google or Firefox.

Mobile: We're also planning to launch our app on mobile devices, enabling users to conveniently access it from their smartphones. This move will significantly expand our user base. As per the survey findings, every participant reported frequent interaction with smartphones, highlighting the vast potential of this market.

Revenue Streams:

Advertisements: We have decided to incorporate advertisements as a revenue stream by allowing third-party ads on our site. Through this arrangement, we will generate a modest profit from both student listers and seekers. It has been surveyed that advertisements do not detract users from using applications so long as they are not overly intrusive.

Service Charge for Students: After interviewing students facing limited housing choices, it's clear that accessibility takes precedence over affordability for them. There's a widespread acknowledgment among students that in today's housing market, paying a premium is expected. This is an evolving trend shaping their housing expectations.

Freemium Model: We will also offer users the opportunity to pay a small fee to gain access to the premium version of our application. Premium users enjoy an ad-free experience and gain access to exclusive features unavailable to regular users. We incorporated this model based on survey feedback that revealed landlords were reluctant to cover a service charge. However, they showed interest in using an application that offered a freemium model.

2.6.2) Cost Side

Key Partners:
Student Associations: The DCU Student Association is an important partner for our student accommodation app because they will help to improve the general student experience by giving us useful information, working with us, and providing support. This association is a well-known voice for Dublin City University (DCU) students, so it has a deep understanding of the unique needs and preferences of students looking for housing. Working with the DCU Student Association lets us add features and functions to our app that are focused on students. This makes sure that it perfectly fits the needs and tastes of DCU students. The association's involvement also gives us a direct way to get feedback, which lets us keep tweaking and improving the app based on real-time input from students. Along with making the app more useful, this partnership also helps the app developers get along better with the rest of the university community. This makes for a better, more comprehensive housing option for students.

Insurance Provider: An insurance company is a key partner for our student housing app because they provide an extra layer of safety and protection for both students and landlords. Working with an insurance company lets us add insurance choices to our app without any problems. This gives students peace of mind that their things and accommodation are safe. The insurance company becomes an important part of the app's ecosystem, making sure that coverage choices are made to fit the needs of both landlords and students who are looking for a roommate. This partnership makes our app more valuable overall, turning it into a complete solution that not only makes it easier to find and handle accommodations but also helps with the important task of reducing risk.

Cloud Hosting: Cloud hosting is an important partner for our student accommodation app because it provides a scalable and resilient system that makes sure the app works well and is easy for everyone to access. By using cloud services, our app can handle different amounts of user traffic quickly and easily, changing to meet the needs of more users during busy times like registration times. Cloud hosting is reliable and has multiple backups, so there is less chance of losing data or having your server go down. It's also a cost-effective option because we can adjust the resources based on how much they are used. The app is responsive because cloud hosting is flexible and scalable. This means that students, landlords, and managers can safely access and manage information about housing from anywhere. This relationship with cloud hosting not only makes the app run faster, but it also makes it more reliable and improves the user experience. Upon researching, we concluded that AWS would be the best cloud hosting provider as you only have to pay for the resources that you use, it is scalable and highly compatible with the django framework.[7]

Key Activities:
Filtered Search: Filtered searches is one of the most important features of our student accommodation app because it allows users to quickly and accurately find housing choices that meet their specific needs and preferences. The filtered search feature of the app lets students narrow down their options based on area, rental prices, amenities, and other important factors. The process of looking for acceptable accommodation can be very difficult, but this feature makes it easier to do, which saves time and improves efficiency.

[7] *Ryabtsev, A. (2024) Top 6 django compatible hosting services, Software Development Blog & IT Tech Insights | Django Stars.*

User Profiles: User profiles are one of the key activities of our student accommodation app because they keep all of the information and opinions about each user in one place. By letting student listers and seekers make and maintain profiles, the app is creating a personalised experience where people can enter information like preferred housing, rental history, and contact information. The process of matching students with suitable housing is sped up by user profiles, which also make it easy for listers to get to know possible tenants. This feature also allows users to see all property listings made by a user and go through them. In summation, focusing on user profiles not only makes the app easier to use, but it also builds community and openness, leading to a more user-centred and interesting overall student housing experience.

Post Property Listing: Posting property listings is a crucial function of our student accommodation software, enabling property listers to effectively display available accommodations to the student seeker community. This feature functions as a conduit between property providers with prospective tenants, enabling them to upload comprehensive information, top-notch photos, and unique amenities pertaining to their properties. This activity enables landlords to efficiently promote their lodgings, expanding their reach to a wider demographic of students in search of appropriate living arrangements.

User Following Mechanic: The user following mechanic is the crux of our social media aesthetic and instils in users the ability to connect with other users. Users can follow one another and keep track of the users who follow them. This feature is interchangeable with the following mechanics on other social platforms and has been shown to be widely used by the student demographic we are aiming to target.

Accessible Navigation Bar: The navigation bar is an essential function for any application. It ensures seamless navigation throughout the entire application. Should a navigation bar be overly intrusive or lack access to core pages, it may be off-putting to users and deter them from using the application entirely. Thoughtful UI design will be needed to ensure our navigation bar is both accessible and visually appealing without being intrusive to the processes in the application.

Key Resources:

Support Team: The support team is a vital resource for our student accommodation app, being very important in making sure that both student seekers and listers have a good time and can easily use the service. As the first line of help, the support team answers questions quickly, fixes technical problems, and walks users through how to use the app's features. Their knowledge makes it easy for new users to get started, and they're always there to help with any problems that come up during the accommodation search or ad management. By being present and quick to respond, the support team builds trust among users, which makes them more likely to trust the app.

Developers: Developers are a cornerstone resource for our student accommodation app, driving its innovation, functionality, and overall success. Their knowledge of programming, software architecture, and user experience design is very important for making a tool that works well and is easy for people to use. Developers are very important to make sure that the app meets the changing needs of both students and landlords. They do this by adding complicated backend features and making frontend interfaces that are easy to use. Additionally, their ongoing work to fix bugs, improve speed, and keep up with new technologies helps make the app reliable and flexible.

Equipment: Equipment constitutes a fundamental and indispensable resource for our student accommodation app, including all the gear and software tools that are needed to build, maintain, and support it. High-performance servers, computers, and networking technology make up the app's backbone and make sure it works well and quickly. Our team of developers can make and improve the app's features with the help of development tools that come with high-tech software. Cloud services also offer flexible storage and computing power, which lets the app change based on what users want.

Security: Security is a paramount and non-negotiable resource for our student accommodation app, making sure that sensitive user data, financial transactions, and the platform's general security are kept safe. Strong security measures, like encryption protocols, secure login procedures, and frequent vulnerability assessments, are needed to keep user data safe and create a trustworthy environment. A dedicated security team constantly looks out for and reacts to possible threats, lowering risks and making sure that industry standards are met. The focus on security is not only an important resource, but also a fundamental part that builds trust among users, which makes them more likely to make purchases, speak safely, and give the app their personal information.

Cost Structure:

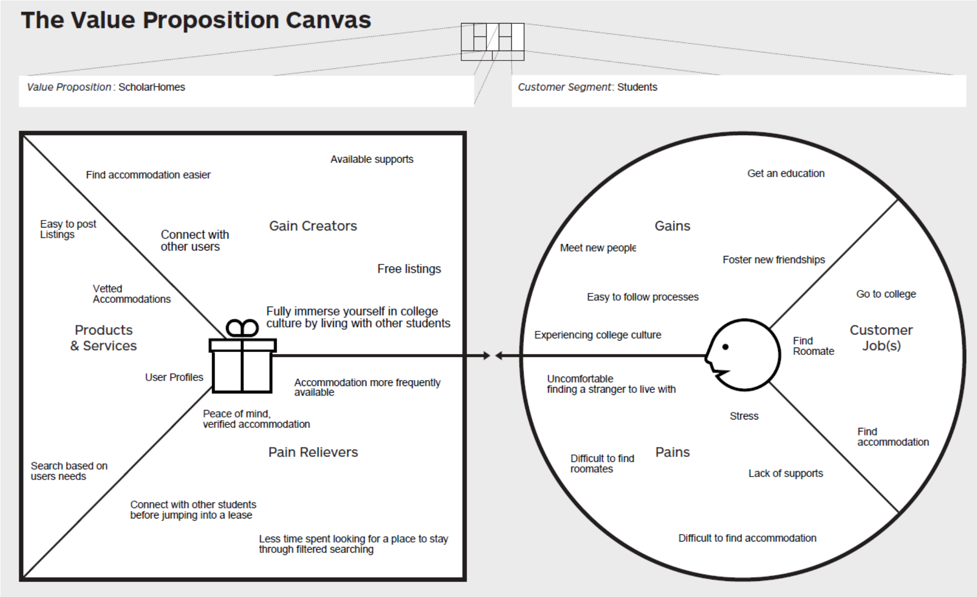
Contingency Fee: A contingency fee is a key cost for our student accommodation app, representing a proactive investment in dealing with problems or uncertainties that came up out of the blue during the creation and use of the app. This fee covers any extra costs that might come up, like sudden technology problems, changes in what users want, or changes in the way the market works. By setting aside money for a contingency fee, we create a financial safety net that lets us be adaptable and flexible throughout the duration of the app. This proactive method not only lessens the effect of unplanned events, but it also makes the app more resilient and better able to adapt to the changing landscape of student housing services.

Staff: Staff is a crucial and integral cost for our student accommodation app, including the trained workers who are in charge of different parts of making the app, keeping it running, and helping customers. Developers, artists, project managers, and support staff all use their skills to make sure the app works, gives users a good experience, and is a success overall. Their work includes coding, creating, testing, updating, and answering user questions. This is what the app does for a living. It is important to hire skilled and committed employees to encourage new ideas, keep the app relevant in the competitive student housing market, and give users a smooth and dependable experience.

Building application: Building the application is a pivotal and substantial cost for our student accommodation app, the costs of making software, including writing, designing, testing, and setting up the infrastructure. This important funding will make sure that a strong, easy-to-use platform is built, complete with filtered search, user profiles, and safe communication tools. A group of skilled developers, artists, and project managers work together to create a new, scalable app that meets the specific needs of both students and landlords. The costs of technology, licensing, and development tools are a big part of making a reliable and feature-packed app for student housing.

Hosting Application: Hosting the app is a fundamental and significant cost for our student accommodation platform the costs of setting up a hosting system that is reliable and scalable. This price covers maintenance, server space, data storage, bandwidth, and making sure the app can handle different amounts of user traffic and data storage needs. Choosing good server services can help the app run faster, be more reliable, and be safer. This investment is necessary to keep user experiences smooth, support features like filtered search and user profiles, and make sure the app works well during busy times, whether you use cloud hosting or private servers. Hosting costs are a big part of making sure that users have a strong and always-available platform. This shows how committed we are to making a reliable and effective student housing app.

2.7) Value Proposition



The Value Proposition Canvas
Value Proposition: ScholarHomes
Customer Segment: Students
Find accommodation easier
Available supports
Easy to post Listings
Connect with other users
Gain Creators
Free listings
Vetted Accommodations
Products & Services
Fully immerse yourself in college culture by living with other students
User Profiles
Accommodation more frequently available
Peace of mind, verified accommodation
Pain Relievers
Search based on users needs
Connect with other students before jumping into a lease
Less time spent looking for a place to stay through filtered searching
Get an education
Gains
Meet new people
Foster new friendships
Easy to follow processes
Go to college
Experiencing college culture
Find Roomate
Customer Job(s)
Uncomfortable finding a stranger to live with
Stress
Find accommodation
Difficult to find roomates
Pains
Lack of supports
Difficult to find accommodation

2.7.1) Customer Side

Customer Job(s):
This customer segment comprises university students in pursuit of suitable accommodation or seeking a roommate for their current accommodation. Their primary focus revolves around attending classes and excelling in their academic pursuits. As a result, their key priority lies in finding housing options conveniently located near their university campus or finding roommates whose goals align with theirs.

Gains:
For students, the benefits extend beyond education; it's about pursuing passions and crafting careers. They aspire to expand their social network and forge connections during their college years. International students, in particular, highlighted the allure of exploring new cultures and travelling as a significant factor motivating them to study abroad.

Pains:
The endeavour to find accommodation or suitable roommates presents numerous daunting challenges for students. The entire search process can be mentally draining, causing significant stress to those involved. Those who've encountered struggles in finding housing expressed dissatisfaction with the support available, feeling it fell short in addressing their queries. In Ireland, the scarcity of accommodation amplifies these difficulties further. Students have also highlighted the issue of trust, feeling uneasy about staying with landlords or roommates they haven't interacted with personally. For international students living alone, the absence of a local contact for guidance magnified their concerns. For students seeking roommates they find it hard to commit to a long term leasing with someone they haven't interacted with.

2.7.2) Value Side

Products & Services:
ScholarHomes provides a range of services dedicated to helping students discover their ideal accommodation. Our app boasts robust search and filter features, enabling users to tailor their search to their specific needs. ScholarHomes enables students to create personalised profiles and showcase their personalities, creating transparency among the user base. Users can also follow each other similar to other social media platforms. This encourages users to reach out and connect with one another and could potentially flourish friendships among students that would last throughout their college years.

Gain Creators:
ScholarHomes offers students several key advantages to improve their experience. Firstly, they gain access to a diverse range of accessible accommodation options, allowing them to choose what best fits their individual preferences. Whether they are seeking a single bed in a shared room in order to maximise savings or value privacy and want to rent a private single bedroom; ScholarHomes is built to give students control over their living arrangements. Students can view the profiles of other students, enabling them to connect with those they are most compatible with. This feature allows them to assess compatibility based on factors such as shared interests, lifestyle preferences, and study habits. By connecting with like-minded peers, students can enhance their living experience and foster meaningful relationships within their accommodation community. Our simple and sleek User Interface ensures the process is seamlessly easy to follow.

Pain Relievers:
The ScholarHomes support team is committed to providing students with a reliable service and assistance throughout their accommodation search journey. Whether students encounter technical issues, have an inquiry about the listing process, or need guidance navigating the platform, our dedicated support team is readily available to address their concerns and provide swift solutions. To ensure transparency and ensure student safety, our team thoroughly vets property listings before they're posted. This reassurance leverages the trust of users and hopefully boosts the confidence of students using the platform. Our built-in search and filter functionalities empower students to save time during their accommodation. They can refine the properties listed by location, type, price, and other criteria, enabling the process to focus on the most suitable options. This will be a tremendous benefit to the user and dramatically enhance the efficiency of the overall process. No user wants to wade through dozens of property listings that do not match their search criteria.

2.8) Competitor Analysis

Rent.ie, Daft.ie, and DCUaccommodation.ie are established players in the student accommodation market, each offering unique features and services. However, ScholarHomes aims to revolutionise the industry by offering a feature-rich platform that caters specifically to the needs of both property listers and seekers, while leveraging social media-inspired functionalities to enhance user engagement and connectivity.

Rent.ie:
Rent.ie is a respected and well known platform for property listings, including student accommodation. It provides a diverse range of listings across Ireland, catering to various demographics. However, Rent.ie lacks the specialised features and user-focused approach that ScholarHomes provides. It fails to provide any features that have a social aspect that cater to the student demographic. While Rent.ie serves as a valuable resource for finding accommodation, it does not offer the tailored experience that students seek in their housing search.[8]

Daft.ie:
Daft.ie is one of the largest property portals in Ireland, offering extensive listings for both rental and sale properties. It includes a dedicated section for student accommodation, making it a popular choice among students. Its strength lies in its vast database and market reach. However, like Rent.ie, it lacks the student-centric features and community-building aspects that ScholarHomes emphasises. Its User Interface design can also be considered dated by modern standards compared to other applications like Daft.ie, DCUaccommodation and even ScholarHomes.[9]

DCUaccommodation.ie:
DCUaccommodation.ie caters specifically to students of Dublin City University, offering listings for both on-campus (e.g. Undergraduate dorms) and off-campus accommodation options (e.g. Aspen). While it provides a focused solution for DCU students, its scope is solely limited to students attending Dublin City University. Scholar Homes aims to serve students from various institutions across Ireland. Additionally, DCUaccommodation.ie does not offer the same level of user engagement features as ScholarHomes, such as user profiles and social media-inspired functionalities.[10]

ScholarHomes:
ScholarHomes sets itself apart by offering a comprehensive platform that addresses the specific needs of students seeking accommodation. By allowing students to not only search for properties but also to connect with potential roommates and landlords, ScholarHomes creates a dynamic and interactive environment tailored to the student experience. The incorporation of social media-inspired features like user profiles, following, and followers enhances user engagement and fosters a sense of community among users. Additionally, ScholarHomes' focus on providing context about accommodation options, such as the type of tenant sought by the lister, sets it apart as a student-oriented solution. With its user-friendly interface and innovative approach, ScholarHomes aims to revolutionise the student accommodation market and provide a seamless housing search experience for students across Ireland.

[8] *rent.ie, Rent Dublin, apartments and houses for rent in Ireland*.

[9] *Daft.ie, Search Ireland’s no. 1 property website*.

[10] *dcuaccommodation.ie, Dublin City University accommodation DCU Accommodation*.

2.9) Legal and Ethical considerations

What are the legal and ethical considerations relevant to our application that we must consider? The rules that our student housing app follows are set by local and national laws. It spells out the rights, duties, and safeguards that students and landlords have, making sure that everyone is treated fairly and legally. There are strict measures in place to keep user data safe and secure, which means that personal information stays private. We are committed to handling data in a way that is legal and moral, and that includes having clear privacy rules. Landlords are taught how important it is to follow fair housing rules, and any kind of discrimination is strictly forbidden. By following the rules of fair housing, the app helps make housing environments that are welcoming, diverse, and acceptable. It is the law and the right thing to do to communicate clearly and openly. The information on the app is correct and organised in a way that makes it easy to make smart choices. This includes ads for places to stay, rental terms, and costs. Contract law guides the writing of deals and contracts between students and landlords. This makes sure that everything is clear, fair, and that everyone's rights are protected. It is both the law and the right thing to do to give users the information they need to make good choices. Features like user profiles, limited search options, and reviews from past students give you a lot of information to help you make smart decisions. The app promotes openness and user-generated material, creating a community-driven system that puts the needs of both student listers and seekers first.

We understand and accept that we have different moral duties to student seekers and listers when we create and run our student accommodation app. For seekers, it is our moral duty to treat user privacy with the greatest care and respect, making sure that their personal information is handled carefully. We promise to support fair housing practices, fight all kinds of discrimination, and build a community that welcomes everyone and loves differences. Giving student seekers clear information and helping them make well-informed decisions is in line with our moral duty to improve their housing experiences. At the same time, we have a moral responsibility to be honest with listers about our contractual agreements, to be fair about the accommodations we make for them, and to build trust through honest actions. We believe in creating a moral framework that not only follows the law but also puts the rights, well-being, and principles of ethics of both seekers and listers in the student housing community first.

If you want to live in our student housing community, you're welcome to a place where strict legal compliance and unwavering moral ideals rule. It's what makes us stand out as an ethical platform: our dedication to protecting user privacy, pushing fair housing practices, and encouraging inclusion. Transparency in communication is important to us because it not only helps us follow the law, but it also builds trust among all users. You can be sure that your personal information is kept safe, and our dedication to fairness will make sure that you become a part of a community that values diversity and welcomes everyone. Our moral duties to both student seekers and listers show that we want to make the setting positive and respectful. This means that your search for housing is not only legal, but also moral. Help us make our community a better place by valuing respect, honesty, and moral duty.

2.10) Financial plan

Detailed financial plan attached in appendix

Yearly Growth Assumptions

Revenue Growth:
Subscription Fees: Assume a 10% annual growth in user base due to increased market penetration and brand recognition.
Ad Impressions: Project a 5% increase in impressions and rate due to growing user engagement and inflation adjustments on ad rates.
Investment Income: This will depend on further investments and bank loans taken in the subsequent years; assume a steady investment growth of 5% year over year if the same investment strategy is maintained.

Expense Growth:
Fixed Costs (Office Rent, Salaries, Insurance): These costs typically rise with inflation. Assume a 3% annual increase.
Variable Costs (Marketing, Legal, Development): Depending on strategy shifts and inflation, assume a 5-10% increase. Focus on efficiency gains might offset some of these increases.

Extended Financial Plan Overview

Year 1 (Current Plan)
Revenue: €22,750 from subscriptions, €1,820 from ad impressions, and additional revenue from other sources as outlined.

Expenses: €151,190 total, encompassing all operational, marketing, and development costs.
Net Income: Initially a loss as startups often spend heavily to gain market traction.

Year 2 Projections
Revenue:
Subscription: €22,750 + 10% growth = €25,025
Ad Impressions: €1,820 + 5% growth = €1,911
Assume stable other income sources or minor growth based on strategic investments.
Expenses:
Assume a 5% overall increase due to scale efficiencies balancing out inflation in some areas: €151,190 * 1.05 = €158,750
Net Income: Calculate based on adjusted revenues and expenses.

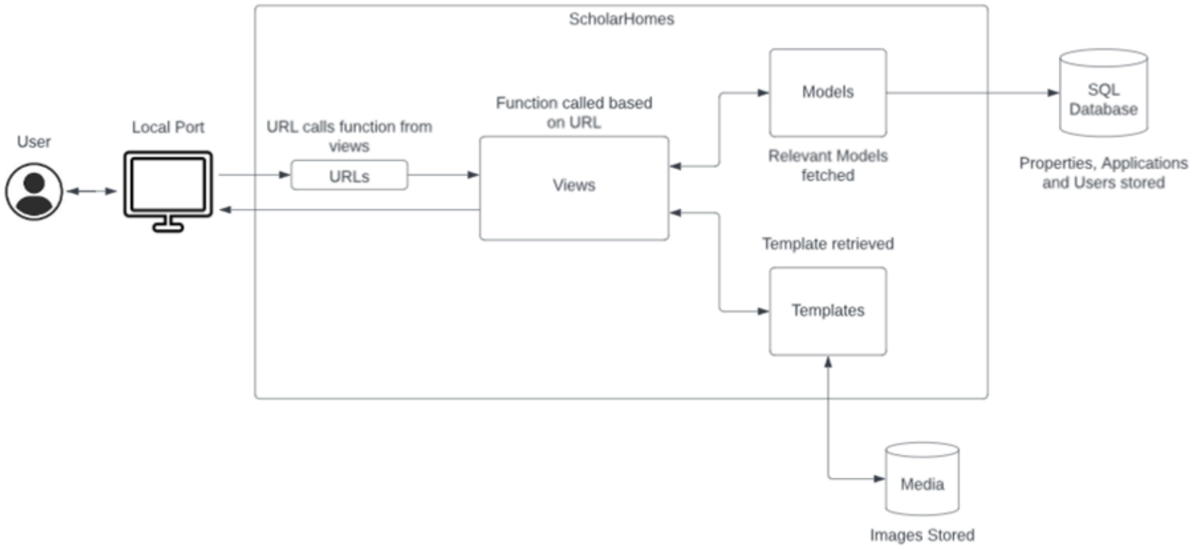
Year 3 Projections
Revenue:
Subscription: €25,025 + 10% growth = €27,528
Ad Impressions: €1,911 + 5% growth = €2,007
Continued stable or growing other income.
Expenses:
Further 5% increase if no major strategic changes: €158,750 * 1.05 = €166,688
Net Income: Calculate based on these projections.

Considerations for Expansion

Market Expansion: Add new features or services to grow the user base and income per user.
Cost management: involves minimising turnover and improving operational efficiencies to manage rising costs.

Investment Opportunities: Seek extra funds or loans to assist growth projects, particularly those involving growing the app's features and reach.

3.1) System Architecture:



Django Framework: Django is a high-level Python web framework that we have chosen to use to build our web application. It promotes rapid development and clean, pragmatic design which is extremely beneficial as we are a small team of 2 with a short time frame of 6 months to plan, research, develop and document this application. It offers several features that make it an excellent choice for developing web applications**.** Django offers built-in security features that safeguards from SQL injections, cross site scripting and other types of malicious attacks. Django is built to scale and handle both large databases (which is crucial for our application) and large amounts of traffic by distributing multiple requests. The framework is versatile and supports a wide array of features from user authentication to supporting 3rd party APIs such as google maps which we used for our application. Django has a wide range of documentation and support available should we encounter any issues. This mitigated many of the risks we feared may occur as a result of using this framework. Django's built-in admin panel serves as a powerful Human-Machine Interface (HMI), offering an intuitive platform for managing application data. It enables us to craft custom user scenarios effortlessly, ideal for showcasing features during our demonstrations. With the admin panel, we can manipulate database objects on-the-fly and swiftly create new ones, bypassing the typical user journey for rapid prototyping or testing.[11]

Browser to Urls.py: This link makes sure that when a user interacts with the app in the browser by typing in a URL or clicking on buttons and links, the Django framework can connect these actions to the functions set out in the urls.py file. The urls.py file is like a router; it has patterns that match the URLs that people type in. In the views.py file, each URL pattern is linked to a different view. The logic and material that should be shown to the user are determined by these views. This link makes sure that when a user does something in the browser, the right views and functions are activated. This makes it possible for a well-organised and adaptable web app.

[11] *Django documentation: Django documentation Django Project.*

Urls.py to the views.py: The relationship between the urls.py and views.py files in the student accommodation app are crucial for coordinating the navigation and operation of the program. The urls.py file serves as a router, encompassing a mapping of URL patterns to certain views within the application. The connection is formed by utilising Django's URL patterns, where each pattern is linked to a particular view that is described in the views.py file. The views.py file contains the views that encapsulate the logic and behaviour associated with various components of the student accommodation app. For instance, when a user wants to find out about the accommodations that are available, the URL pattern in urls.py will guide the request to the appropriate view in views.py that is specifically built to handle queries related to accommodations.

Views to the models.py and templates folder: In the student accommodation app, the views.py file, the models.py file, and the template folder must all be linked in order for the app to be well-structured and work properly. The three parts work together to keep different issues separate, keep data safe, and give people clear information. The models.py file sets up the data models that describe the app's most important entities, like Student, Accommodation, and Reservation. These models are used by the views in views.py to get, make, update, or delete data based on what the user wants. To show this answer to the user, the views use HTML files that are kept in the template folder. The templates set the structure and layout of the user interface and include information that changes based on what the views do. The related view in views.py gets the information from models.py and sends it to the right template if a user wants to see more information about a certain room. The template then displays the data in an easy-to-use way, making sure that the user has a smooth and uniform experience.

Views to the models.py and templates folder: In the student accommodation app, the views.py file,the models.py file, and the template folder must all be linked in order for the app to be well-structured and work properly. The three parts work together to keep different issues separate, keep data safe, and give people clear information. The models.py file sets up the data models that describe the app's most important entities, like Student, Accommodation, and Reservation. These models are used by the views in views.py to get, make, update, or delete data based on what the user wants. To show this answer to the user, the views use HTML files that are kept in the template folder. The templates set the structure and layout of the user interface and include information that changes based on what the views do. The related view in views.py gets the information from models.py and sends it to the right template if a user wants to see more information about a certain room. The template then displays the data in an easy-to-use way, making sure that the user has a smooth and uniform experience.

3.2) Key Features:

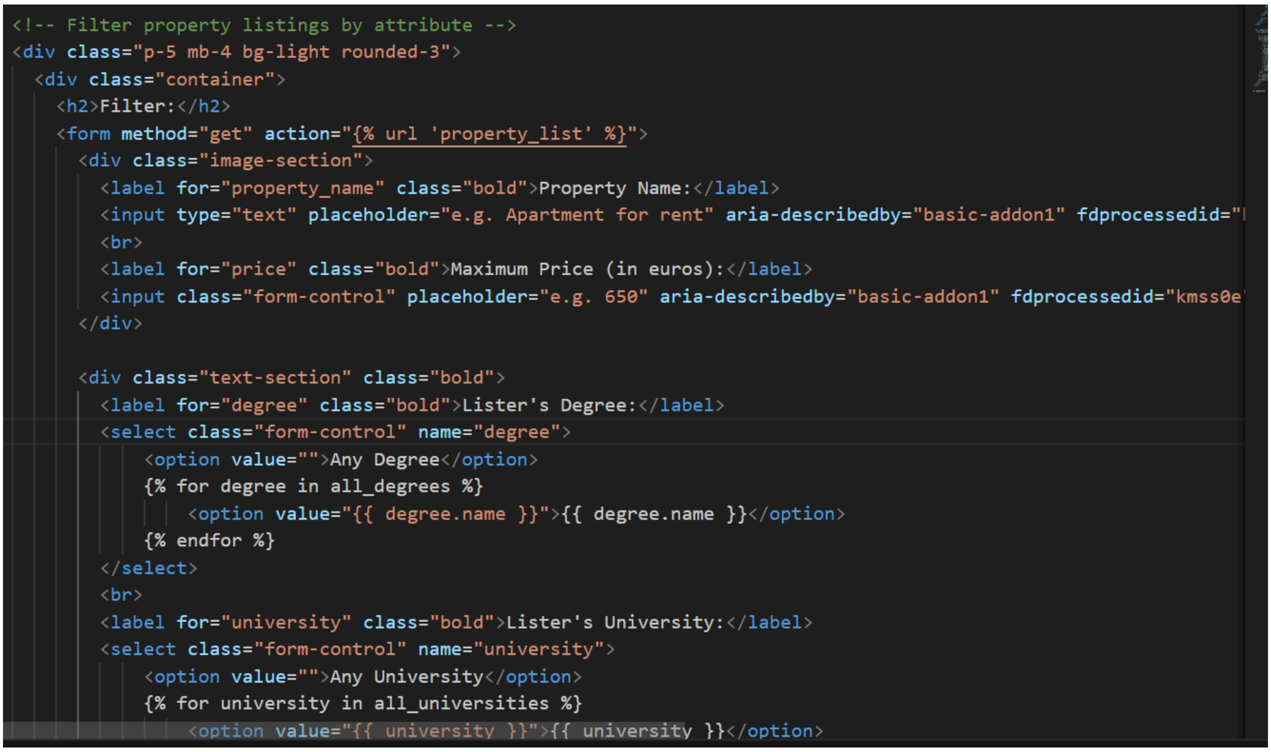
3.2.1.) Filter properties:

Description: The filter function is a form at the top of our property list that, once submitted, will allow users to filter properties based on property name, the Lister's degree, the Lister's college and property price. This is done by limiting the number of properties pulled by the property_list function in views.html to properties that match the attributes selected in the form. If a value on the form is empty, the conditional statements prevent the filter function from activating.

Criticality: This feature drastically enhances the property search process, omitting properties not relevant to the users search criteria.

Technical Issues: This function requires access to our SQL server in order to pull the relevant properties within the filter's criteria. Should an issue affect our database, this will result in this feature losing its functionality.

Dependencies: This function relies on properties existing with the attributes selected in the filter to function effectively.



```
<!-- Filter property listings by attribute -->
<div class="p-5 mb-4 bg-light rounded-3">
  <div class="container">
    <h2>Filter:</h2>
    <form method="get" action="{% url 'property_list' %}">
      <div class="image-section">
        <label for="property_name" class="bold">Property Name:</label>
        <input type="text" placeholder="e.g. Apartment for rent" aria-describedby="basic-addon1" fdprocessedid="
        <br>
        <label for="price" class="bold">Maximum Price (in euros):</label>
        <input class="form-control" placeholder="e.g. 650" aria-describedby="basic-addon1" fdprocessedid="kmss0e
      </div>

      <div class="text-section" class="bold">
        <label for="degree" class="bold">Lister's Degree:</label>
        <select class="form-control" name="degree">
            <option value="">Any Degree</option>
            {% for degree in all_degrees %}
                <option value="{{ degree.name }}">{{ degree.name }}</option>
            {% endfor %}
        </select>
        <br>
        <label for="university" class="bold">Lister's University:</label>
        <select class="form-control" name="university">
            <option value="">Any University</option>
            {% for university in all_universities %}
                <option value="{{ university }}">{{ university }}</option>
```

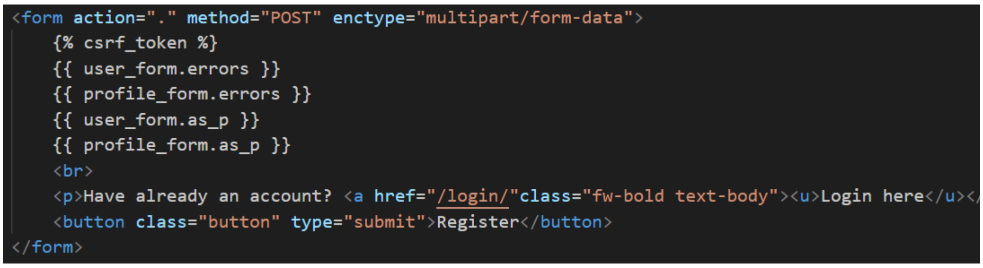
3.2.2.) Register:

Description:The user's first step towards becoming a member is to use the ‘Register’ function. Our site has two kinds of registered-users: seekers and listers. On the site's main page, there is a link that takes you to a registration form for either a property lister or property seeker. Any user who wants to use our site will have to sign up with their details. The user must enter their password, username, and email address when this form asks for them.

Criticality: Having separate user profiles for student seekers and listers in a student housing app makes the experience more personalised and quick, meeting the needs and expectations of everyone involved in the renting process.

Technical issues: The form for signing up will be made in HTML and will match the main style of the site. PHP functions will be used to handle the form entries and will connect to our MySQL database to store user data.

Dependencies: Not dependent on other features



```
<form action="." method="POST" enctype="multipart/form-data">
    {% csrf_token %}
    {{ user_form.errors }}
    {{ profile_form.errors }}
    {{ user_form.as_p }}
    {{ profile_form.as_p }}
    <br>
    <p>Have already an account? <a href="/login/"class="fw-bold text-body"><u>Login here</u></
    <button class="button" type="submit">Register</button>
</form>
```

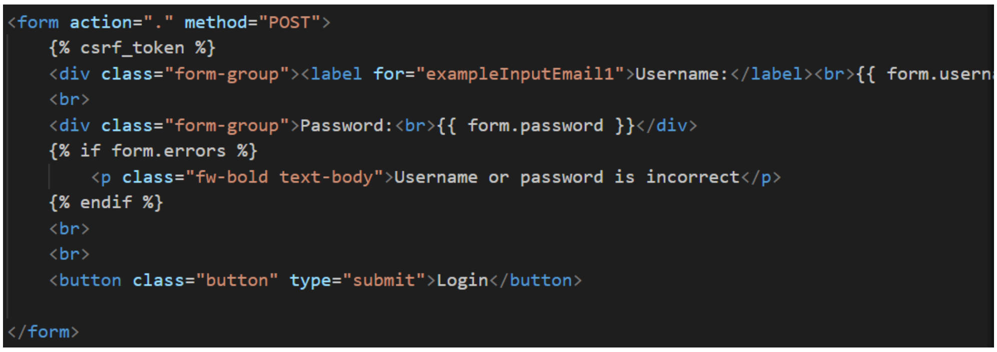
3.2.3.) Log In:

Description: This function is a simple script that asks the user for their username and password when they select to log in. This information is provided by the user when they register and stored in our database. When a user wishes to log in, they complete a form entering their username and password. This function then searches our database for the username and password entered by the user. If the user forgot their username or password, they can click the following link “I forgot my password” to receive an email containing the user’s username and steps to reset their password.

Criticality: The ability to log in to our app is very important for many reasons. Firstly, it enhances the security of our system by having user authentication. It does this by protecting sensitive data and ensures that only people who are allowed to can access the platform. This feature also lets users customise their experiences, so students can easily handle their profiles, preferences, and interactions. Having system authentication is also critical to our business processes that require multiple users, e.g. property leasing between student listers and seekers.

Technical Issues: The system will need some kind of protection to keep people from getting in without permission. We will also need to ensure our site’s urls are routed correctly and only users who are correctly authenticated with the correct permissions can access certain pages, functions and data.

Dependencies: For users to sign in, they will need to use the correct information they gave when they registered. Relies on the register function as an initial step.



```
<form action="." method="POST">
    {% csrf_token %}
    <div class="form-group"><label for="exampleInputEmail1">Username:</label><br>{{ form.usern
    <br>
    <div class="form-group">Password:<br>{{ form.password }}</div>
    {% if form.errors %}
        <p class="fw-bold text-body">Username or password is incorrect</p>
    {% endif %}
    <br>
    <br>
    <button class="button" type="submit">Login</button>

</form>
```

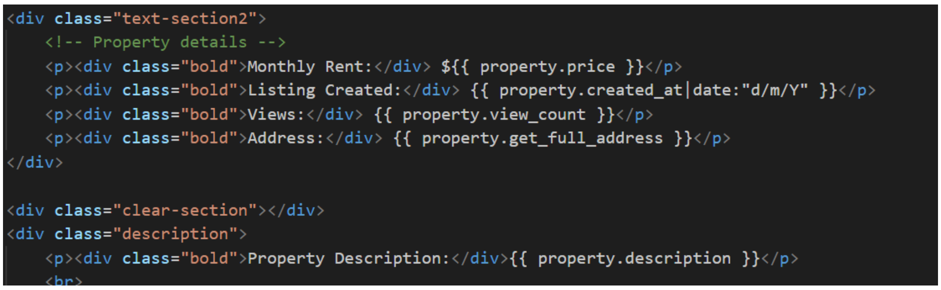
3.2.4.) View Properties function:

Description: The ‘View Properties’ function enables student listers to showcase and manage their rental properties. Student listers can create detailed listings that include property specifications, such as location, rental rates, amenities, and lease terms. This function allows for easy property management, enabling listers to update availability status, upload images, and communicate directly with prospective student seekers. Users do not require authentication to see these listings. The number of views a property listing has received can also be seen here

Criticality: The ‘View Properties’ function is critical in a student accommodation app as it serves as the primary avenue for listers to present their rental properties to potential roommates or tenants .This feature streamlines the property management process, allowing listers to provide comprehensive details about their listings which can be updated as needed. It is not only an efficient and direct way for student listers to communicate the details of the property to student seekers, but also ensures a simple user-friendly experience for both parties. This function is also pivotal for enticing unregistered users to register by showing them the accommodation available on our application.

Technical issues: Details of each property listing will be stored on our SQL database. This function may pose technical challenges related to data management and scalability. As our system evolves and the scale of our application expands, we may need to consider migrating our database to a different platform.

Dependencies: Depends on lister-users signing up to the app to post listings.



```
<div class="text-section2">
    <!-- Property details -->
    <p><div class="bold">Monthly Rent:</div> ${{ property.price }}</p>
    <p><div class="bold">Listing Created:</div> {{ property.created_at|date:"d/m/Y" }}</p>
    <p><div class="bold">Views:</div> {{ property.view_count }}</p>
    <p><div class="bold">Address:</div> {{ property.get_full_address }}</p>
</div>

<div class="clear-section"></div>
<div class="description">
    <p><div class="bold">Property Description:</div>{{ property.description }}</p>
    <br>
```

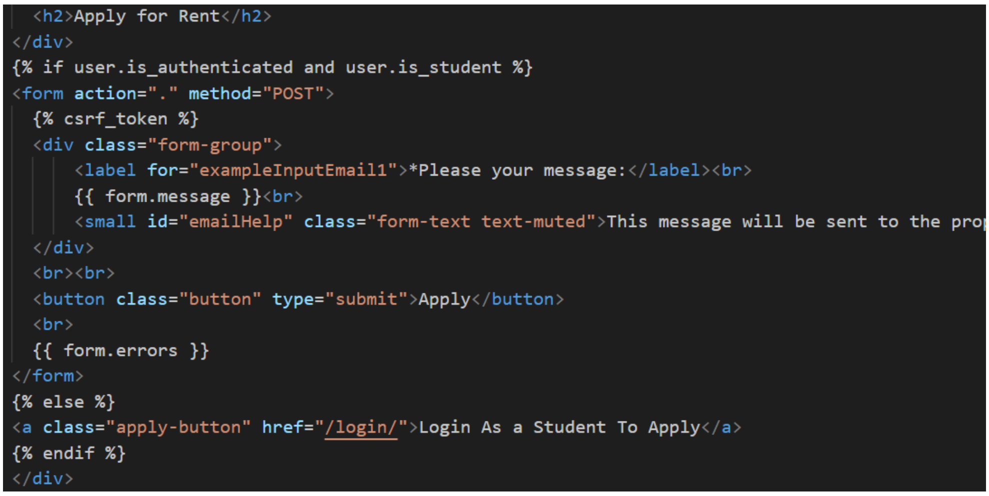
3.2.5.) Apply for Rent function:

Description: The ‘Apply for Rent’ function allows student seekers to submit applications to renting specific properties listed on the platform. This feature streamlines the application process by providing a user-friendly interface for students to enter necessary information, upload required documents, and communicate their interest to listers.

Criticality: The "Apply for Rent" function is of the utmost importance because it's where potential renters can show interest in a property and officially apply for it.

Technical issues: The form used to create the apply for rent function has to be fully functional or students will not be able to rent. These details require storage on our SQL database.

Dependencies: Depends properties being listed completely. Also dependent on registered student-users to apply for listings.



```
  <h2>Apply for Rent</h2>
</div>
{% if user.is_authenticated and user.is_student %}
<form action="." method="POST">
  {% csrf_token %}
  <div class="form-group">
      <label for="exampleInputEmail1">*Please your message:</label><br>
      {{ form.message }}<br>
      <small id="emailHelp" class="form-text text-muted">This message will be sent to the pro
  </div>
  <br><br>
  <button class="button" type="submit">Apply</button>
  <br>
  {{ form.errors }}
</form>
{% else %}
<a class="apply-button" href="/login/">Login As a Student To Apply</a>
{% endif %}
</div>
```

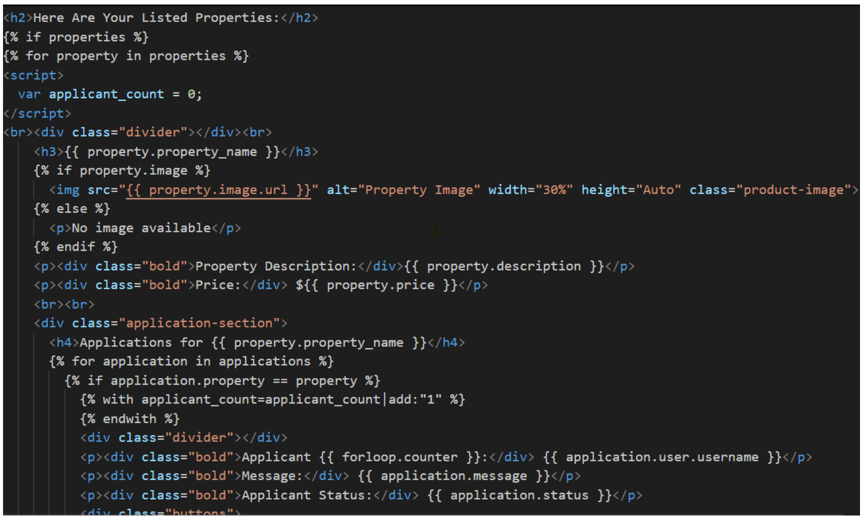
3.2.6.) View my Listings function:

Description: The ‘View my Listings’ function gives student listers a central, easy-to-use interface to govern their rental properties. This feature gives student listers the tools they need to oversee and make changes to property listings, keep track of tenant applications, talk to potential or current renters, and keep an eye on financial transactions like rent payments.

Criticality: The dashboard operates similarly to a command centre; it gives student listers the full scope of all of their properties and makes it easy to handle them, communicate with tenants, and keep track of payments all within the app.

Technical Issues: The issue we could face here is that there are a lot of steps that are regulated through this dashboard. We must ensure that all processes operate as intended and flow cohesively to ensure flawless functionality throughout the overall leasing process. This involves running multiple tests to find and patch exploits as soon as they arise.

Dependencies: Depends on property listings made by the student lister. Requires the student to be registered as a lister to view this interface.



```
<h2>Here Are Your Listed Properties:</h2>
{% if properties %}
{% for property in properties %}
<script>
  var applicant_count = 0;
</script>
<br><div class="divider"></div><br>
    <h3>{{ property.property_name }}</h3>
    {% if property.image %}
      <img src="{{ property.image.url }}" alt="Property Image" width="30%" height="Auto" class="product-image">
    {% else %}
      <p>No image available</p>
    {% endif %}
    <p><div class="bold">Property Description:</div>{{ property.description }}</p>
    <p><div class="bold">Price:</div> ${{ property.price }}</p>
    <br><br>
    <div class="application-section">
      <h4>Applications for {{ property.property_name }}</h4>
      {% for application in applications %}
        {% if application.property == property %}
          {% with applicant_count=applicant_count|add:"1" %}
          {% endwith %}
          <div class="divider"></div>
          <p><div class="bold">Applicant {{ forloop.counter }}:</div> {{ application.user.username }}</p>
          <p><div class="bold">Message:</div> {{ application.message }}</p>
          <p><div class="bold">Applicant Status:</div> {{ application.status }}</p>
          <div class="buttons">
```

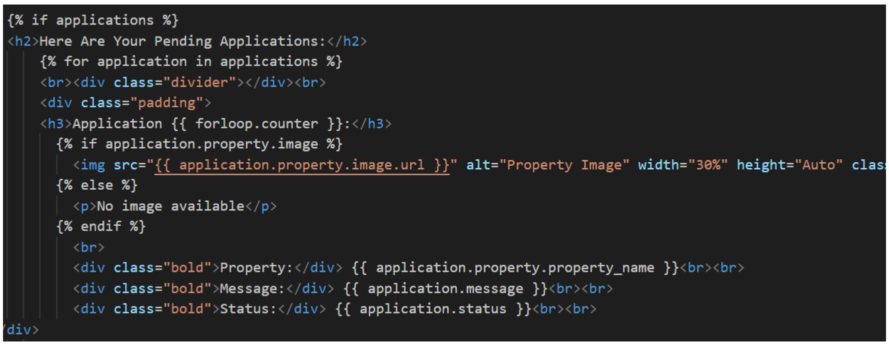
3.2.7.) My Applications function:

Description: This feature allows student seekers to access a personalised portal where they can view the properties they have applied for and check the status of their applications for each property.

Criticality: This is important because it provides students a hub to keep track of their applications and to be able to get direct feedback from landlands. It is also pivotal that students can view the history of applications they have made to comply with modern design practices.

Technical issues: There are a lot of potential bugs that can arise due to the constant change in application status. We must ensure that the flow of our functional processes are cohesive and do not conflict with each other. For example, an exploit in the flow of the process within our code may cause a student's application to be halted and unable to proceed. The logic for these processes will be written in Python and contain elements of HTML to design the user interface.

Dependencies: depend on applications being made by students and also properties being listed. Additionally requires a registered seeker-user.



```
{% if applications %}
<h2>Here Are Your Pending Applications:</h2>
    {% for application in applications %}
    <br><div class="divider"></div><br>
    <div class="padding">
    <h3>Application {{ forloop.counter }}:</h3>
      {% if application.property.image %}
        <img src="{{ application.property.image.url }}" alt="Property Image" width="30%" height="Auto" clas
      {% else %}
        <p>No image available</p>
      {% endif %}
        <br>
        <div class="bold">Property:</div> {{ application.property.property_name }}<br><br>
        <div class="bold">Message:</div> {{ application.message }}<br><br>
        <div class="bold">Status:</div> {{ application.status }}<br><br>
/div>
```

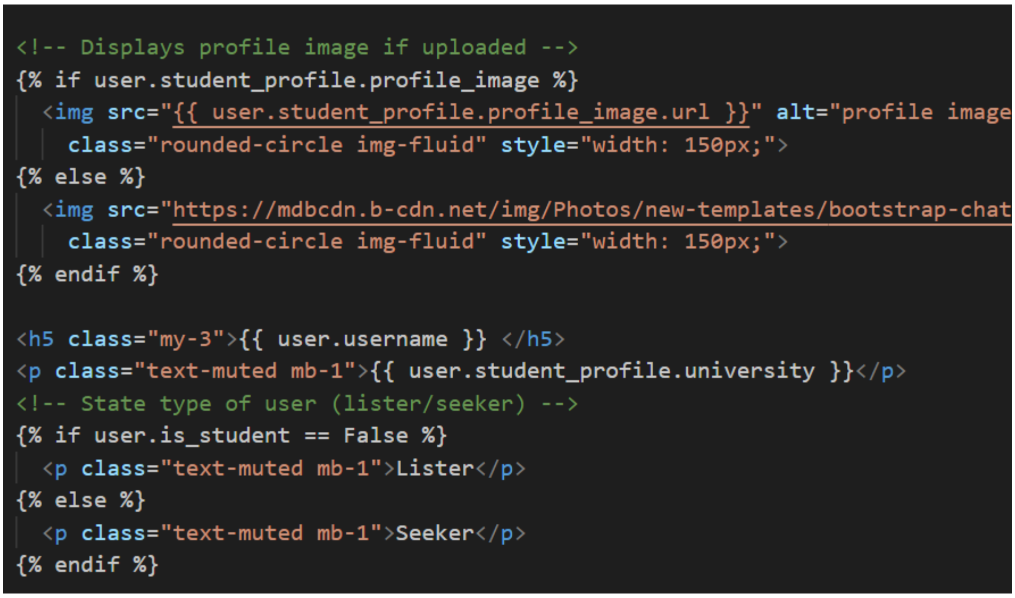
3.2.8.) View User Profile:

Description: This feature allows students to view each profile. Profiles contain details about the user such as their name, university, degree and profile image. If the user is a Lister, their ongoing property listings will also appear and can be viewed. The users current followers and following count can be seen here. Users can also view their own profile and update their information from here by filling out a form.

Criticality: This feature is pivotal as it allows users to gain insight into one another and demonstrate their personality. This feature emphasises the social aspect of our application and appeals to the student demographic. Profiles allow users to make informed decisions, and ease them into a relationship with other users before entering into a lease.

Technical issues: There is a functionality in the view profile function from viewing the users property listings, follower counts, social medias and the user details. This makes it highly reliant on our SQL database which could be a potential issue as it is a single point of failure. In addition to this the sheer amount of functionality combined with the update profile form leave room for the possibility of bugs.

Dependencies: Fundamentally, a user is required in order for this function to work as intended.



```
<!-- Displays profile image if uploaded -->
{% if user.student_profile.profile_image %}
  <img src="{{ user.student_profile.profile_image.url }}" alt="profile image
    class="rounded-circle img-fluid" style="width: 150px;">
{% else %}
  <img src="https://mdbcdn.b-cdn.net/img/Photos/new-templates/bootstrap-chat
    class="rounded-circle img-fluid" style="width: 150px;">
{% endif %}

<h5 class="my-3">{{ user.username }} </h5>
<p class="text-muted mb-1">{{ user.student_profile.university }}</p>
<!-- State type of user (lister/seeker) -->
{% if user.is_student == False %}
  <p class="text-muted mb-1">Lister</p>
{% else %}
  <p class="text-muted mb-1">Seeker</p>
{% endif %}
```

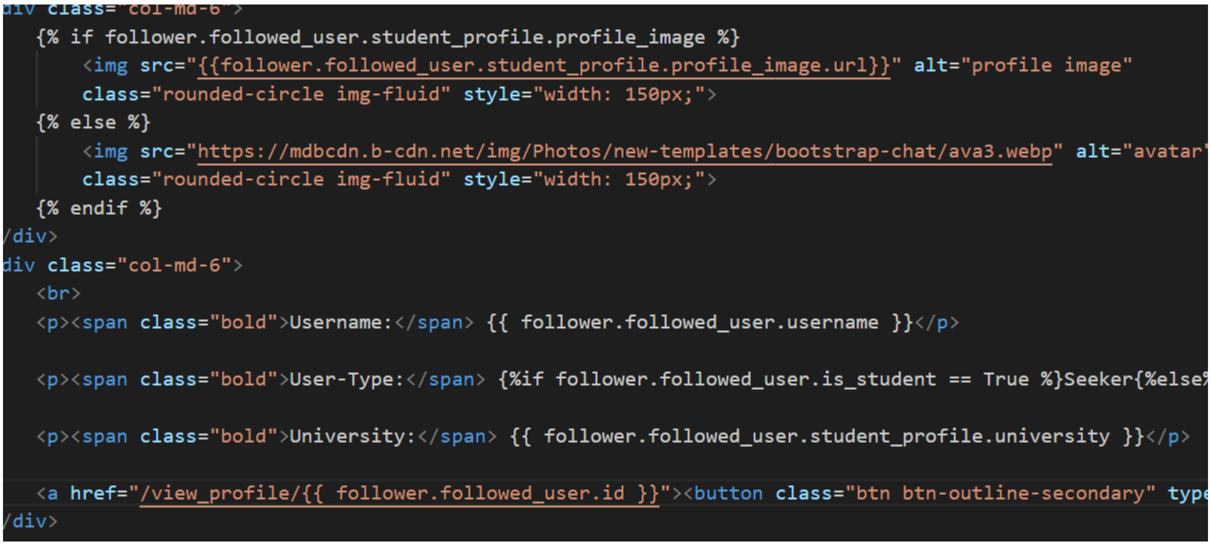
3.2.9.) Followers/Following:

Description: User A is given the option to follow a User B upon visiting their profile. Should they select follow, an entry will be created in our database noting User A as following User B. You can see the total number of followers a student has or the students they are following by clicking either followers or following in their profile.

Criticality: This feature further follows the base functionalities of a social media application, which we hope to emulate to some extent. It also allows users to bookmark students whose profiles they may wish to access later by following them.

Technical issues: It is possible bugs may be present within our code that could potentially let users follow a user more than once. No such bugs have been found yet however should such an occurrence arise, strict constraints in our code will need to be implemented.

Dependencies: Relies on a user to be followed and a user following. This feature is also heavily reliant on the view profile feature and would become obsolete without it.

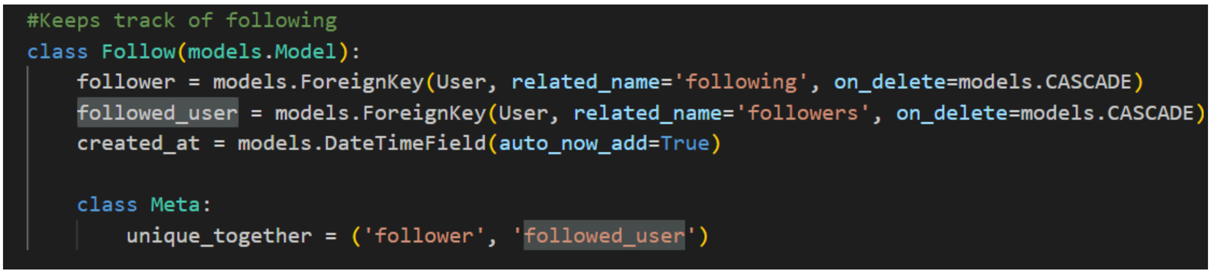


```
div class="col-md-6">
   {% if follower.followed_user.student_profile.profile_image %}
       <img src="{{follower.followed_user.student_profile.profile_image.url}}" alt="profile image"
       class="rounded-circle img-fluid" style="width: 150px;">
   {% else %}
       <img src="https://mdbcdn.b-cdn.net/img/Photos/new-templates/bootstrap-chat/ava3.webp" alt="avatar"
       class="rounded-circle img-fluid" style="width: 150px;">
   {% endif %}
/div>
div class="col-md-6">
   <br>
   <p><span class="bold">Username:</span> {{ follower.followed_user.username }}</p>

   <p><span class="bold">User-Type:</span> {%if follower.followed_user.is_student == True %}Seeker{%else%

   <p><span class="bold">University:</span> {{ follower.followed_user.student_profile.university }}</p>

   <a href="/view_profile/{{ follower.followed_user.id }}"><button class="btn btn-outline-secondary" type
/div>
```



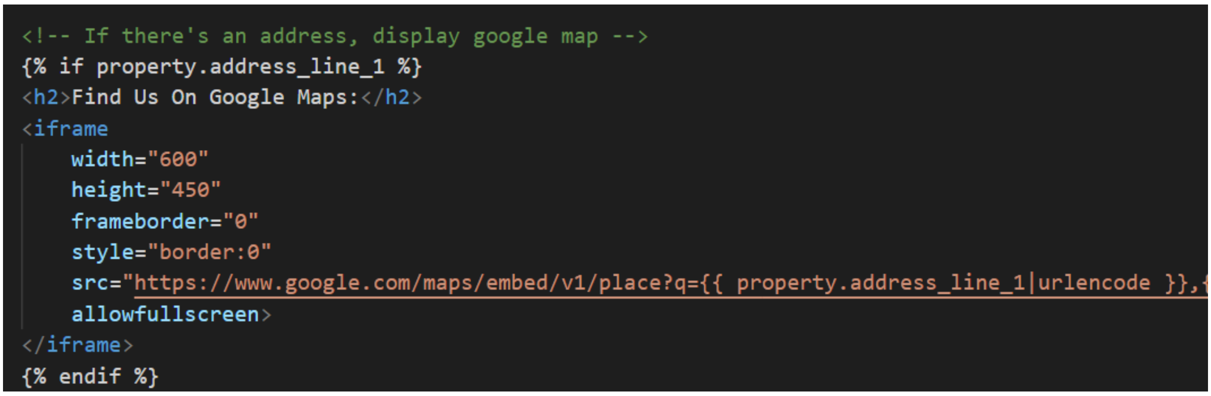
```
#Keeps track of following
class Follow(models.Model):
    follower = models.ForeignKey(User, related_name='following', on_delete=models.CASCADE)
    followed_user = models.ForeignKey(User, related_name='followers', on_delete=models.CASCADE)
    created_at = models.DateTimeField(auto_now_add=True)

    class Meta:
        unique_together = ('follower', 'followed_user')
```

3.3) External Interfaces:

Google Map API:
For the majority of our applications development we were able to create the majority of our functionalities through built in django features (e.g. User Authentication, Database management, Url configuration, e.t.c.); However the use of google maps which was essentially for our applications functionality was not supported.

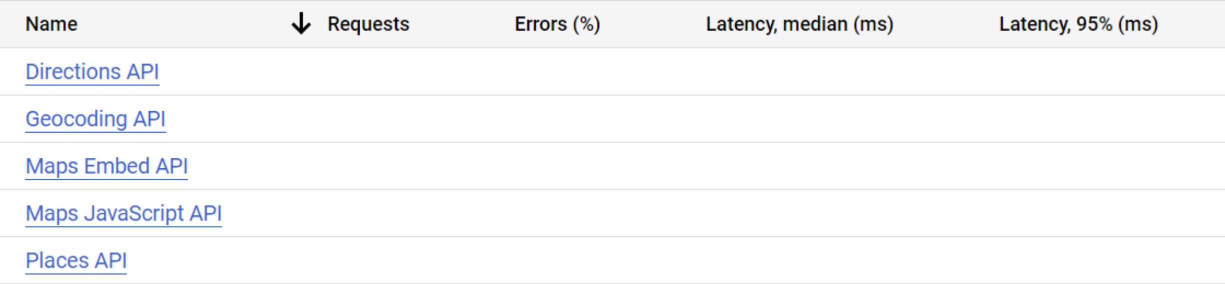
The below snippet of code below shows how we embedded a Google map API into our templates:



```
<!-- If there's an address, display google map -->
{% if property.address_line_1 %}
<h2>Find Us On Google Maps:</h2>
<iframe
    width="600"
    height="450"
    frameborder="0"
    style="border:0"
    src="https://www.google.com/maps/embed/v1/place?q={{ property.address_line_1|urlencode }},
    allowfullscreen>
</iframe>
{% endif %}
```

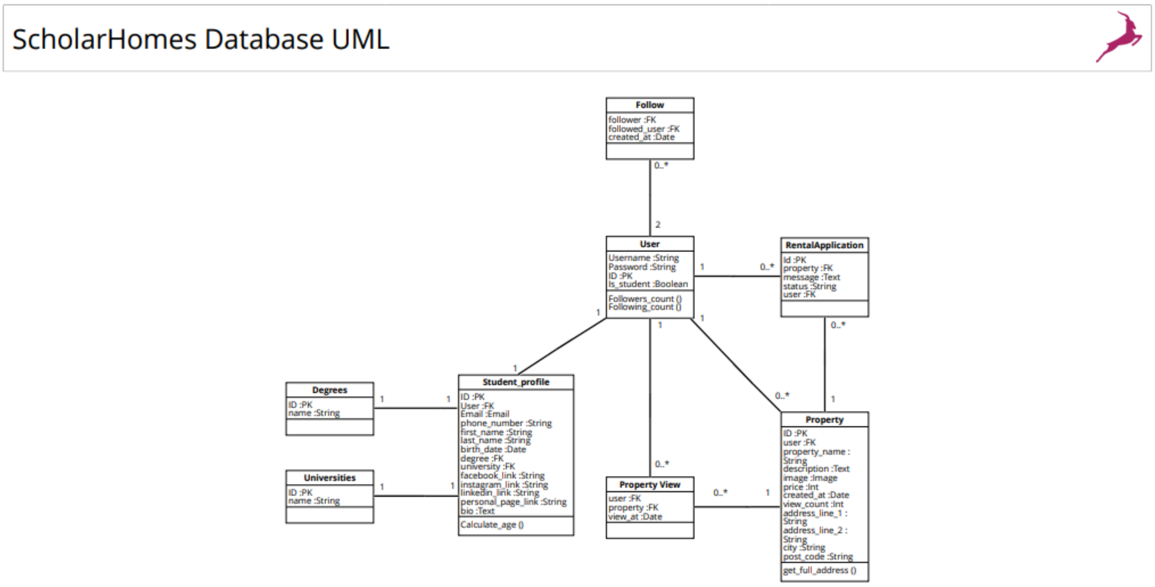
We have embedded this pivotal feature into our **individual_property.html**. We needed to first set up the API on Google Cloud, this involved enabling several different API's related to google maps that we believed would be beneficial to the user (e.g. directions, pins for locations). We then generated an API key that we could use to call our Google Maps API when running our application. We put some constraints on our API key so that it could only be called specifically from our website. If we did not add this constraint, our API key could be called from other sites which would result in us being charged whenever other sites call on our API key. Should we commercially release ScholarHomes we will need to consider the financial cost associated with using this API and incorporate it into our budget.

Below are the current APIs enabled we currently have enabled:



Name	↓ Requests	Errors (%)	Latency, median (ms)	Latency, 95% (ms)
Directions API				
Geocoding API				
Maps Embed API				
Maps JavaScript API				
Places API				

3.4) UML Database Design:



This diagram shows not only what the ScholarHomes platform can do, but also all of its complicated parts and the choices that went into making it. The main goal of this detailed schema is to organise and make it easier for users to interact with each other based on their profiles, academic connections, and activities related to listing properties and applying for rentals.

Primary Entities and Attributes:
The core of the ScholarHomes database revolves around several key entities including User, Student_profile, Degrees, Universities, Property, Follow, Property View, and RentalApplication. Each entity is designed to hold specific types of data essential for the functioning of the platform.

1. User: This entity captures basic authentication and identification details such as Username, Password, and ID (primary key). It also includes a Boolean attribute Is_student to distinguish between student seekers users and landlords (student listers). Furthermore, dynamic attributes like Followers_count() and Following_count() show off the presence of our social feature within the platform, allowing users to follow others and engage with a community.

2. Student_profile: This entity stores specific personal information about the student, such as their email address, phone number, first and last names, and date of birth. It is linked to the User entity through a foreign key. Degrees and Universities are linked to academic information through foreign keys, which show where the student went to school. Fields like facebook_link, instagram_link, linkedin_link, and personal_page_link show that social media is integrated, which makes it easier for users to interact with each other and make connections. The bio field lets you write your own history, and the method Calculate_age() probably uses the birthdate to figure out the user's age.

3. Universities and Degrees: These groups store basic but important details about the universities and degrees available on the site. Each one has a unique ID and name that are used to find academic information in student records.

4. Property: This entity shows listings for homes that are offered to rent. It has many attributes, like property_name, description, image, price, and specific address fields, that let you give a lot of information about each property. The view_count property and the get_full_address() method suggest ways to keep track of people who are interested in listings and give full address information.

5. Follow: This entity tracks relationships between users with characteristics like follower and followed_user, both of which are foreign keys that link back to the User entity. This is how ScholarHomes' social features work. The created_at attribute helps keep track of when these connections happened.

6. Property View: This object is very important for figuring out which properties users are interested in because it has a timestamp view_at that shows when a property was viewed. It has foreign keys to both the user and the property.

7. RentalApplication: Finally, this entity handles rental requests sent by users. It connects them to properties and shows the progress of each request with fields such as message, state, and foreign keys to both the user and the property.

The UML diagram certainly shows different kinds of links, like one-to-many or many-to-one, which shows how users can have many followers, property views, or rental applications. These connections are very important for keeping the data correct and making sure that the database can handle complicated searches well.

3.5) Database Requirements:

Our application will utilise a MySQL database that is connected within the Django framework. We aim to use version 8.3.0 which is the latest version of MySQL currently available; However should we run into complications and need to switch to an older version, we must revert no lower than 8.0.11 as this is the oldest version supported by django.

MySQL databases have robust performance and scale suitability in the context. It is an open source database with an easy to use UI. It also is suitable for holding large sets of data which will be crucial as we plan to hold images in our application which require larger storage space.

We estimate we will require at least a 500GB database. For reference, applications such as daft.ie with a large user base and numerous listings have a storage size estimated to be 1 TB. As our target demographic is significantly more niche (being university students) we estimate we will require this amount of space, which is significantly smaller.[12]

[12] Mitrofanskiy, K. (2024) *How to choose the right database for your needs: IntelliSoft Guide*, *Intellisoft*.

3.6) Performance Requirements:

When developing software for student accommodation, which serves both listers—those offering accommodation—and seekers—those looking for accommodation—it's crucial to prioritise and define key performance requirements. These requirements not only ensure that the platform functions effectively and efficiently for all users but also enhance the user experience by providing reliable and responsive service.

Responsiveness: Ensuring rapid page loads is crucial in order to prevent user dissatisfaction, especially during periods of high user activity, such as the start of the academic year when students are actively looking for housing. Page load durations should ideally not exceed two to three seconds. User attrition from a given platform may ensue as a consequence of unsatisfactory launch times. To achieve this we use django.

Django is built to handle requests efficiently using a shared-nothing architecture. Each request is handled separately, which means that the process handling the request loads its own copy of necessary data, improving reliability and fault isolation. Django has a strong caching system that can store results from actions that are expensive or often used, like database queries or template rendering. It is possible to serve later requests faster by caching these outputs. This is because the information is pulled from the cache instead of being calculated again. The Object-Relational Mapper (ORM) in Django lets us talk to the database in Python, which hides and improves SQL queries. This separation of concerns not only makes it easier to maintain and create code, but it also lets Django make improvements like query set caching, which can stop queries from being sent twice. Additionally, we can use Django's ORM to do bulk insertions and updates, as well as database-specific features (such as on-duplicate-key updates) that can greatly speed up processes that involve a lot of data.

Search Efficiency: When designing our student housing software, we realised right away that search efficiency would be a critical performance requirement. As developers, we aimed to match our users' high expectations—digital natives who want flawless and responsive online experiences. Delays or lags in retrieving search results may annoy users and drive them to speedier competitors. This knowledge prompted me to prioritise speedy delivery of search results, which is critical for user retention.

The changing nature of student needs also influenced our design strategy. Users of our programs often search for lodgings using specified criteria such as location, price, room type, and closeness to their educational institutions. Efficiently sorting through countless listings to swiftly deliver relevant results is critical. This capacity enables students to quickly understand their options and make fast selections, which is crucial given the often urgent need for accommodation. Scalability was another major challenge. At the beginning of each academic year, there is an expected increase in demand as students hurry to obtain housing. This surge in activity necessitates that our system handle high volumes of searches without compromising speed. Scalability aspect of our system not only ensures a pleasant user experience during these key periods, but it also protects against system overloads, which can cause crashes or slowdowns.

We took into account the accuracy and relevancy of the search results. It is critical for students to believe that the app will appropriately reflect their requirements and preferences with each search.

This trust is built on a highly filtered search algorithm capable of interpreting and prioritising a wide range of user-specified criteria. Implementing these algorithms entailed utilising filtered search strategies to guarantee that the most relevant listings were constantly at the forefront, hence increasing user happiness and engagement.

Reliability and Uptime: In developing a student accommodation app, we prioritised reliability and uptime as critical performance criteria, realising that students rely greatly on ongoing access to our services. Recognising our user base's tight deadlines and urgent needs, it was evident that any downtime, no matter how minor, may disturb their home search, potentially resulting in them missing out on preferred lodgings. The need for constant availability is crucial not only for user convenience, but also for planning and decision-making processes.

We focused on improving customer experience and happiness by reducing downtime and dependability concerns, which could result in frustrations, unfavourable reviews, and a damaged reputation, reducing user retention and discouraging new users. Ensuring good uptime is thus critical to maintaining a positive customer perception and overall happiness with our application. Technical consistency was another important component of our reliability approach. The software was built to manage increasing operations easily, particularly at peak times such as the start of academic sessions, when there is a rise in the number of users and activities. Maintaining reliability during these times guarantees that all users are serviced efficiently and without system failure.

Django also helped with the reliability and uptime of our application as Django supports scalability in several ways. It can handle large volumes of traffic and interactions smoothly with proper setup, using its built-in caching framework and other optimizations. Scalable applications can manage increased loads with minimal disruption, a key attribute for maintaining high uptime. Django strongly emphasises the DRY principle, which promotes the reusability of code. By reducing redundancy, Django helps ensure that there are fewer bugs and that the codebase is easier to maintain, both of which are important for the long-term reliability and uptime of an application.

Location Services Integration: Integrating location services into our student accommodation software was a critical performance requirement for several strong reasons. This feature improves the app's functionality and user experience by allowing for accurate, context-aware service delivery. By implementing location services, we were able to provide users with housing possibilities based on their current location or any specific area, increasing the relevance and efficiency of searches. This targeting according to location feature enabled students to find property near their colleges or in their preferred neighbourhoods, shortening the search process and making it more user friendly.

Location services allowed us to develop distance features, which showed consumers how distant potential rooms were from their educational institutions. This was especially useful for students new to the area, allowing them to make educated judgments about where to reside based on convenience and accessibility. The integration of mapping and directions significantly increased the user experience, allowing students to explore houses on a map and receive directions, which was helpful for arranging property visits or knowing the location of their accommodation. We used strong third-party APIs, such as Google Maps, to provide dependable mapping and routing functionality, as well as optimised data processing to reduce latency and improve app speed. Google Maps APIs are constantly updated with the most recent geographic information and backed by a massive data infrastructure, assuring great accuracy and up-to-date content. The familiar design and comprehensive functionality also contribute to user engagement, lowering the learning curve for new users and increasing satisfaction through consistent and intuitive user interactions.

The addition of location services to our app not only made it more responsive and beneficial for our customers, but it also improved the overall user experience, making it a top choice for students looking for housing. By focusing on these factors, we hoped to provide a deeper, more efficient, and personalised experience for each user, ensuring that our app was more than just a tool for locating the finest student housing options.

3.7) Technical Challenges:

Deploying our application: Deploying our student accommodation project with Django, especially as beginners, entails navigating various hurdles that we may lack technical skills to handle at first. One of the main challenges is correctly configuring the environment parameters for production. This includes upgrading from a development database like SQLite to a more robust system like MySQL, managing sensitive settings like API keys through environment variables with tools like django-environ, and separating settings into development and production files to avoid accidentally deploying unsafe settings.

Managing static and media files correctly is a common stumbling block; in development, Django handles these files automatically, but in production, you must configure settings such as STATIC_ROOT and MEDIA_ROOT and run the collectstatic command to collect these files in a single location that your web server can access. This is necessary to ensure that the app's stylesheets, JavaScript files, and user-uploaded content are available to users. Database migrations can pose a risk; transferring database structures from development to production without incurring downtime or data loss necessitates careful planning and execution. As beginners, we must always back up our databases before trying migrations, and we should test these migrations in a staging environment to iron out any bugs before they reach the live system. We will find it tough since we lack the necessary setting.

Django does have some helpful features which if the time was given we would be able to further expand our knowledge to work through some of these challenges when it comes to deployment. In summary, as beginners deploying a Django-based student accommodation app requires mastery of several technical and operational disciplines, ranging from environment and server configuration to handling static and media files, performing safe database migrations, securing and optimising the app, and complying with data privacy legal standards. Each sector brings unique obstacles, as well as opportunities for growth and learning. In the future we would hope to tackle these technical challenges more when it comes to the deployment of our application.

Data: One of the most fundamental technical issues was gathering reliable and trustworthy data. This problem is caused by a combination of circumstances that can hinder data collection and validation operations. To begin, acquiring data for properties requires gathering detailed and reliable information on rental costs, locations, amenities, and availability. This data is frequently provided by many landlords and property managers, each with their own set of criteria for data accuracy and timeliness, making it challenging to maintain consistency and trustworthiness throughout the platform. The difficulty is worsened by the dynamic nature of the housing market, where property data can change often, necessitating constant revisions to maintain the app's listings current and helpful to students.

Another challenge is to ensure the data's legitimacy. There is a potential of false listings or misrepresented properties, which might cause trust concerns with the app. To mitigate this risk, we had to carefully validate the data provided to guarantee that we did not include any false listings on our page. There are further issues that arise from the variety and breadth of data that is required for a comprehensive service. In order for a student accommodation app to genuinely be useful, it must not only include lists of available rentals, but also comprehensive information about the surrounding area. This information should include safety ratings, local facilities such as grocery stores and launderettes, the availability of public transportation, and maybe evaluations from former tenants.

In addition to these issues, there is the challenge of integrating this data in a smooth manner into the user interface of the application, making certain that it is presented in a manner that is both welcoming to users and easily accessible. One example of this is the development of user-friendly navigation systems and interactive components that make it easy for seekers to locate the information they seek. It is possible that the use of map-based interfaces to enhance the experience of browsing properties will require a great amount of technological skill as well as imagination. It is vital to do continual monitoring and updates of the data in order to ensure that the information that is presented is accurate and keeps its relevance. Due to the continual maintenance, it is necessary to have a staff that is solely responsible for data management and an integrated operational plan in order to manage the continuous flow of new data and changes to the entries that are already in place. Not only does this feature have an effect on the technical infrastructure, but it also has an effect on the overall business plan and how to allocate the resources.

Obtaining and managing trustworthy data for the application that assists students in finding accommodation is challenging for a variety of reasons, including concerns around privacy, integration, scalability, data collection, and validation. When it comes to maintaining the app's dependability, usefulness, and compliance with the law, each of these areas requires a technological and operational approach that has been carefully considered. The app's users will feel more satisfied and more confident as a result of this.

Testing: Testing a student accomodation app requires a complicated technique that includes unit testing, integration testing, user acceptance testing, and stress testing, among others. The goal is to ensure that the software is not just functional, but also user-friendly, secure, and capable of dealing with real-world usage circumstances. However, testing such an application presents its own set of issues.

One of the key issues is the wide range of user interactions that must be replicated. Students, landlords, and administrative users all engage with the app in unique ways, therefore the testing strategy must encompass a wide range of use cases and workflows. For example, students may be seeking housing near their institution, landlords may be uploading property information, and administrators may be monitoring listings and resolving conflicts. Each of these features necessitates extensive testing to guarantee that the software completes all duties correctly and efficiently.

Integration testing was especially difficult because the app must communicate with numerous external systems, including email services, social networking sites, and external APIs for maps. To ensure that the app interacts correctly with these external services, extensive testing is required, which often includes the creation of mock interfaces and staging environments. We only had one environment to test these external services in, thus it was difficult to comprehend how they would perform in multiple contexts. Given the sensitivity of the data involved, security testing is also a crucial factor. The purpose of these security tests is to ensure that no data leakage occurs from our application, ensuring that users are protected and their information is secure. This not only safeguards consumers' data, but also aids in the compliance of data protection legislation such as GDPR.

Testing the use of the product was another challenge. As part of this process, we asked actual users to engage with the application in order to discover usability concerns that we would not have noticed otherwise. During this process, the input that is received is extremely important for the purpose of refining the user interface and increasing the overall user experience. On the other hand, the challenge that we encountered was successfully recruiting a diverse set of test users who appropriately represented the app's intended demographic. This was a challenging and time-consuming endeavour.

In basic terms, testing the usability of our student accommodation app was a complicated challenge that required dealing with complex user needs, connecting many external systems, safeguarding sensitive data, and testing the app's usability. It was necessary to take a strategic approach and carefully design each of these areas in order to guarantee that our application not only satisfies the technical requirements, but also provides a user experience that is safe, effective, and entertaining.

3.8) Interface rationale Including personas and user story

Interface rationale: ScholarHomes' interface is based on simplicity and functionality, with the goal of giving people a smooth experience. Modern standards for usefulness are met by the design, which has clean lines, lots of white space, and a colour scheme that makes you feel safe and confident. Intended to reduce cognitive load, the app's minimalist design makes navigation easy for all users, even foreign students who may not speak the native language fluently.

User Personas
Persona 1: Jack, The International Student
Demographics: 22 years old, studying abroad, not fluent in the local language.
Needs: Needs to find accommodation close to the university, desires a simple process due to language barriers, and prefers visual content over textual explanations.

Persona 2: John, The Young Master's Student
Demographics: 26 years old, a master's student with an apartment near the university.
Needs: Seeks to efficiently find a compatible roommate, prefers a system that simplifies the screening process, and needs a clear dashboard for tracking applications and communications.

Layout and Structure
Home Screen:

- For Jack, the home screen features large, clickable areas with a student dashboard to view properties he has applied for and see his status for those properties.
- For John, the home screen includes quick access to his roommate's matching dashboard, notifications, and a messaging system to communicate with potential roommates.

Search and Filters:

- The app uses search columns and visual selectors allowing Jack to set preferences easily.
- John benefits from advanced filters to view roommate applications by compatibility scores and shared interests.

Listings Display:

- Each listing uses a card layout with high-quality images and essential information in bullet points to help Jack make quick, informed decisions.
- John can see summarized views of potential roommates with key information like shared interests, study habits, and preferred living conditions highlighted.

User Interaction
How to Get Feedback:

- Immediate feedback through color changes and animations reassures Jack that his actions are registered.
- John benefits from transactional feedback confirming when he has successfully contacted a potential roommate or received an application.

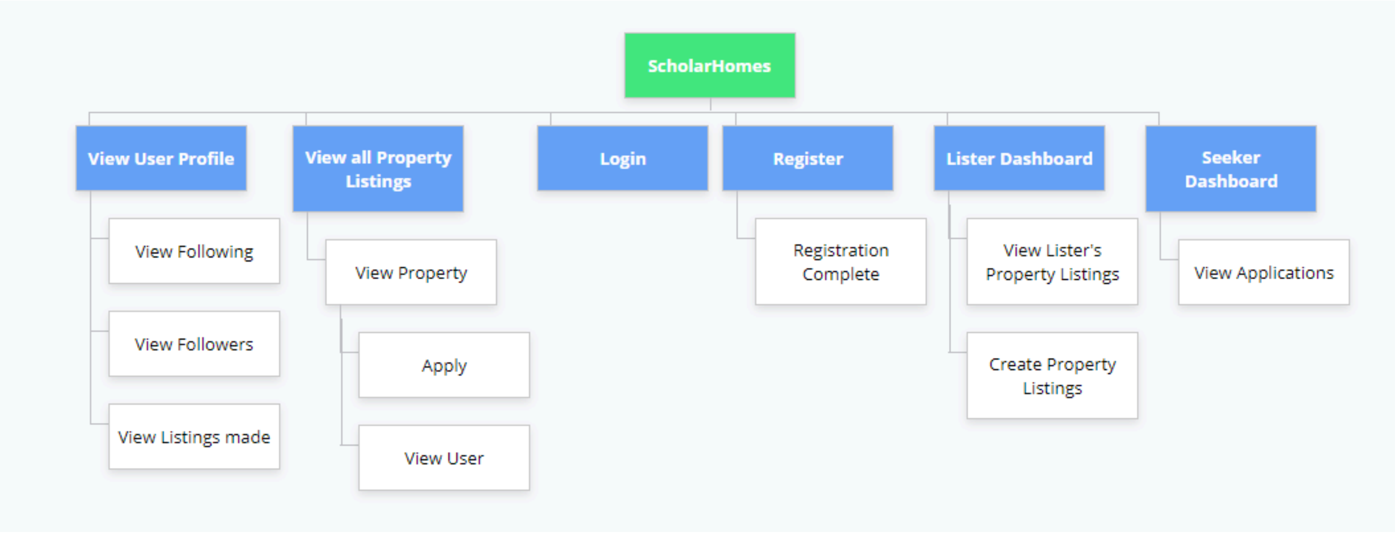
Progressive Disclosure:

- For Jack, detailed property information and booking options expand only when he selects a listing, keeping the initial search results uncluttered.
- John can expand details on roommate profiles and their compatibility information only when needed, maintaining a clean dashboard.

Technical Considerations

- Responsive Design: The app is fully responsive, providing a consistent experience for Jack and John on either of their respective devices.
- Security Features: Secure input fields, especially in payment and personal information forms, use standard encryption and visual indicators to reassure users like John and Jack of their data's safety.

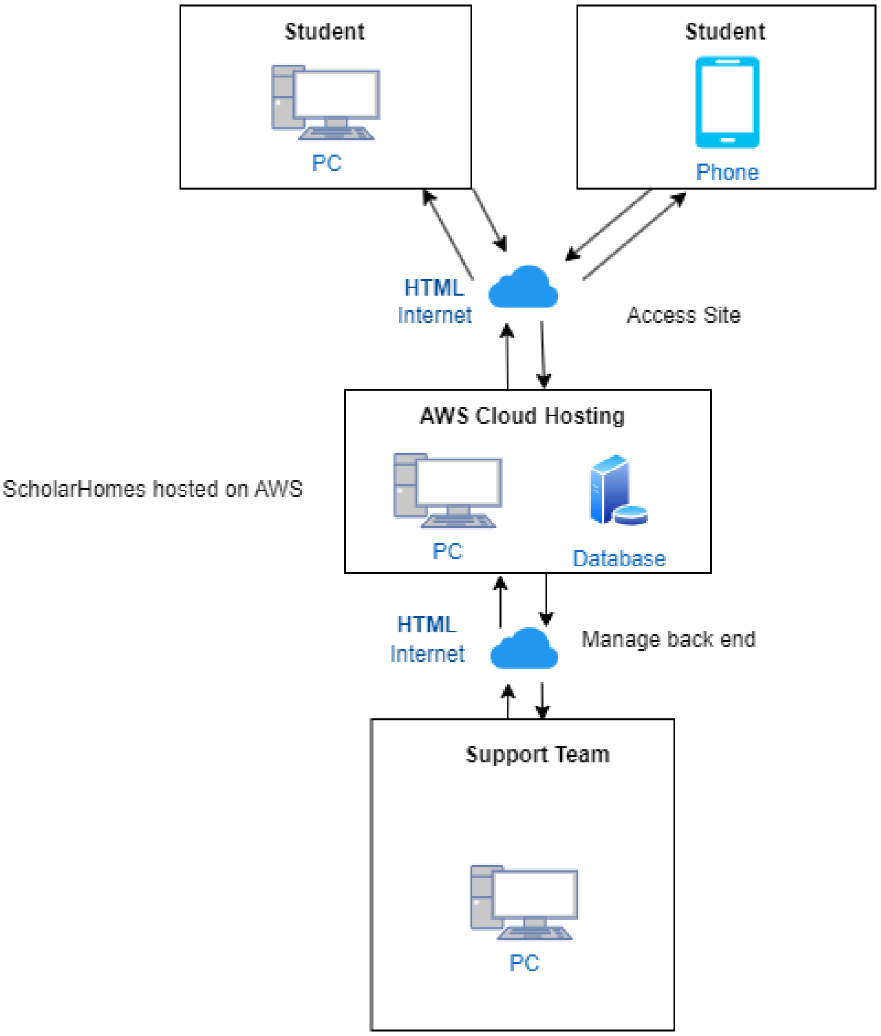
3.9) Site Map Architecture



This site map diagram demonstrates the different level of accessibility we wish each page in our application to have. At the peak of our application we have pages like login and register which are immediately accessible across all our pages through the navigation bar. We also have homepages such as the Lister and Seeker dashboards.

Below we have some tertiary pages that require the user to be on the aforementioned page to access it. For example, you can not enter the view property page from the seeker dashboard, you must be on the View Property Listings page and select the property which page you want to view first.

3.10) Distributed system architecture



The above visualisation shows the parties who we expect to directly interact with the ScholarHomes application.

- Support Team: We will have a dedicated support team available to call should users face any issues when using the application. If the support team needs to manually alter our database from the backend should this type of intervention be needed. They can access our application's admin panel to do this.

- AWS Cloud Hosting: The AWS team provides a wide array of support in order to help developers deploy and maintain their applications. These supports may incur a fee but are greatly beneficial should we encounter any issues beyond our expertise.

- Students: Students are our target demographic and make up the entirety of our user base. Students can access our application through hyper text transfer protocol (HTTP) by searching a URL in a search browser on their phone or computer. We intend to advertise our application's URL on social media and student platforms in order to increase publicity.

4.) Appendix

The appendix does not form part of the assessment, and is to be used for additional content that supports statements in the main body of the deliverable that may be referred to by the assessors to seek supporting evidence. The appendix has no page limit.

Ryan Shannon - **Current Product Configuration and Automation Engineer**

Current final year Enterprise Computing student studying at DCU. An aspiring software developer with a wealth of knowledge on software development and database management from studying at Dublin City University and working alongside Aryza as a Product Configuration and Automation Engineer. Played a key role in designing the front end of the ScholarHomes system and architecting the systems database.

Technical Skills

- Microsoft Office
  Educated on Microsoft Word, Excel and Powerpoint. Useful for documenting the ScholarHomes infrastructure.

- Database Management
  Learned to design database infrastructures in DCU and had practical experience managing databases in Aryza using SQL.This experience was useful when designing the ScholarHomes database within the django infrastructure.

- Object-Oriented Programming
  Learned in Dublin City University and utilised in Aryza to develop the Insolv system. Utilised to create functionality within the backend of the django infrastructure of ScholarHomes.

- Familiar With:
  html, css, Python, Java Script,Java, SQL, Django, Linux, Agile, React learned at DCU and used in Aryza during product development. Html and CSS were especially useful when designing the frontend.Python, Java Script and Django were essential for designing the backend.

Business Oriented

- Project Management
  Agile development and similar methodologies learned at DCU and used at Aryza. Incredibly useful in order to plan and follow the timeline for this project given the limited timeframe.

- Customer Relations
  Consistent customer interaction at Aryza in order to define request requirements. Useful during the initial pitching period and communicating my ideas to my partner for this project.

- Time Management
  Can be seen through simultaneously attending university and working at Aryza. Given the limit timeframe, this skill was pivotal to ensure we stuck to the deadlines we set and we carried out processes as efficiently as possible

- Documentation
  Frequently assembled robust reports detailing bug reports and required changes in the system. This was useful for detailing and formatting the report accompanying the ScholarHomes prototype.

Jomi Kafi - **Former Software development Intern & Former Data analyst Intern**

Current final year Enterprise Computing student studying at DCU. Aspiring IT Analyst having acquired extensive expertise in IT systems, software programming, and database management during my studies at DCU and previous internships at Goodbody Stockbrokers: Significantly contributed to the backend development of ScholarHomes and played a crucial role in implementing the project's data.

Technical Skills:

- Database management:
  I acquired expertise in designing database infrastructures during my studies at DCU, and gained real-world experience in database management during my internships at Goodbody, utilising SQL and PL/SQL.

- Troubleshooting:
  I gained the most of my troubleshooting expertise during my internship as an IT intern at Goodbody. During my experience, I encountered several problem-solving approaches in the fields of software development, IT operations, and service management.

- Software development using python:
  I acquired the majority of my skills in Python throughout my time at DCU through a diverse range of courses spanning from my first to final year. Throughout my internship, I successfully applied my acquired knowledge in a practical setting.

Business Oriented

- Teamwork**:**
  During my IT internship, I worked in a devops environment, which allowed me to enhance my skills in teamwork and collaboration. These skills proved to be valuable for the project I was working on.

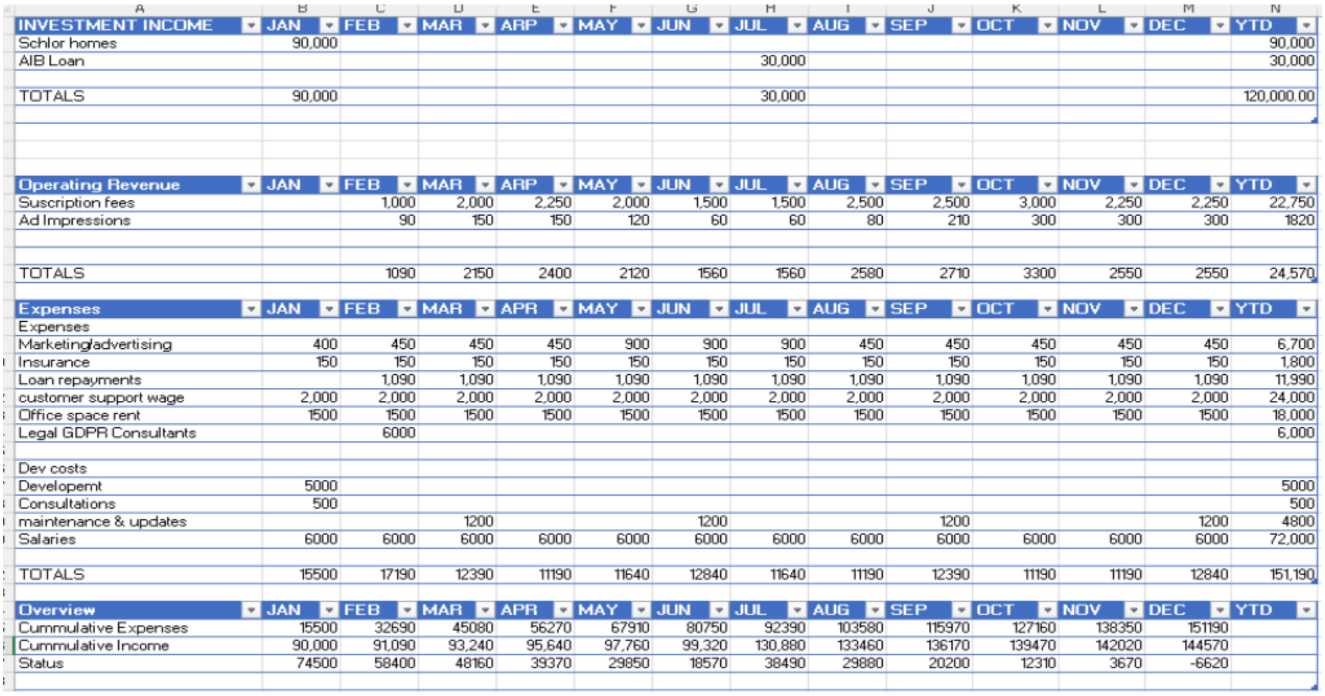
- Communication:
  I have consistently developed good communication skills through my active participation in sports and extensive involvement in group assignments at DCU. These factors contributed to the enhancement of my overall communication abilities.

- Innovation:
  My innovative skills were undoubtedly improved during my stay at DCU. In several modules during the years, we were required to employ our initiative to improve a given project idea or enhance a project that we had to develop ourselves.

- Service-management relations:
  During my time at Goodbody, I was employed on the service management desk, where I assisted colleagues across the firm with any system-related difficulties and IT inquiries they encountered. This experience enhanced my professionalism in handling interdepartmental employee interactions and effectively managing individual staff issues.

4.1) Financial Plan Additional Details

Financial plan for the first year below:



INVESTMENT INCOME	JAN	FEB	MAR	ARP	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	YTD
Schlor homes	90,000												90,000
AIB Loan							30,000						30,000
TOTALS	90,000						30,000						120,000.00

Operating Revenue	JAN	FEB	MAR	ARP	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	YTD
Suscription fees		1,000	2,000	2,250	2,000	1,500	1,500	2,500	2,500	3,000	2,250	2,250	22,750
Ad Impressions		90	150	150	120	60	60	80	210	300	300	300	1820
TOTALS		1090	2150	2400	2120	1560	1560	2580	2710	3300	2550	2550	24,570

Expenses	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	YTD
Expenses													
Marketing/advertising	400	450	450	450	900	900	900	450	450	450	450	450	6,700
Insurance	150	150	150	150	150	150	150	150	150	150	150	150	1,800
Loan repayments		1,090	1,090	1,090	1,090	1,090	1,090	1,090	1,090	1,090	1,090	1,090	11,990
customer support wage	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	24,000
Office space rent	1500	1500	1500	1500	1500	1500	1500	1500	1500	1500	1500	1500	18,000
Legal GDPR Consultants		6000											6,000
Dev costs													
Developemt	5000												5000
Consultations	500												500
maintenance & updates			1200			1200			1200			1200	4800
Salaries	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	72,000
TOTALS	15500	17190	12390	11190	11640	12840	11640	11190	12390	11190	11190	12840	151,190

Overview	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	YTD
Cummulative Expenses	15500	32690	45080	56270	67910	80750	92390	103580	115970	127160	138350	151190	
Cummulative Income	90,000	91,090	93,240	95,640	97,760	99,320	130,880	133460	136170	139470	142020	144570	
Status	74500	58400	48160	39370	29850	18570	38490	29880	20200	12310	3670	-6620	

Looking at Revenue:
We intend to explain our money-making and spending practices by developing a solid financial plan for our student housing app. Our program aims to make student housing booking easy for students and property owners. We want to make money to keep our service going and flourish from special feature subscription fees, property commissions, and finding more relevant products and advertisements. Our resources will be properly invested in adding users, creating apps, and servicing clients while minimising costs. This funding strategy reflects our commitment to a long-term, user-focused platform that improves college housing.

Investment Income:
Investment income is crucial for student housing app because it funds many app goals and services. The app can maintain financial stability by investment money, especially early when user fees or ads may be weak.
Our January investment is our first. Our 90,000-euro company decision will be a clever and crucial strategic step. As the workplace transforms, this investment will help growth, innovation, and competition. Our business can expand and add services. Use this money for services. It could boost our business's ability to adapt to market changes, improve finances, and strengthen business position.

Our business needs an AIB 30,000-euro loan in the second part of the year. This timeline may aid cash flow management and strategic goals. Many businesses must work harder in the second half of the year, especially before Christmas or end-of-year financial planning, so being ahead of this was one of our goals when seeking the loan. At this time, a cash inflow can fund fresh purchases to meet demand, sell items or hire temporary labour. The loan's arrival in the second half of the year boosts the company just in time to capitalise on growth possibilities, manage seasonal changes, and prepare for a successful year-end, satisfying its payment obligations.

AIB is one of Ireland's largest and oldest banks and has a superb reputation. They may be trusted by borrowers due to their extensive market presence. Our company has earned 120,000 euros in investment income this year, which will be distributed over the year. This year, we've spent wisely, which is beneficial for our finances and potential growth.

Operating Income:
Our operating income comes from several sources that aid the site's enduring wealth. User fees, advertising, and value-added services are typical revenue streams. Users like students and landlords pay for app features like listing or finding rooms. Targeted advertising partnerships can also make money by marketing to service providers and property owners who desire to reach the student audience.

Our initial and main revenue stream is app subscriptions at 10 euros monthly per user starting February. This made student life easier and more accessible. Housing software lets you sign up for 10 euros a month for those students. We understand that students struggle financially and believe that access to appropriate lodging information and services should be available to everyone. Based on this nominal fee, students can obtain updated property listings, roommate matching, and housing-related savings. We earned 5250 euros in subscription fees in the first three months, which is good as students are still in school. We anticipate 100 users in February and expect to grow between March and April. As summer approaches, user numbers plummet because it is when most students go home, therefore they don't require dorms at this time. Our financials reveal that summer sales produce less revenue. June and July are pleasant months, so we expect 1500 euros for both.

Once subscriptions grow after the summer when school resumes, fees since students will join our app to find housing. We estimate monthly fees from August to the end of the year will total around 10,000 euros. So far this year, our subscription cost is 22,750 euros.
Ad impressions are our second revenue source. Ad impressions, or "impressions," in internet advertising can track ad views on the internet or other digital channels. Site visitors view ads every time. Whether someone clicks the ad or not, it counts as one impression. Impressions are vital for advertisers and publishers because they display ad visibility and reach.

For every 1000 views on our app, brands will receive 30 euros. Since February, we expect to make 3000 impressions that month, which earned us 90 euros from impressions. We anticipate continued site use in March, April, and May will increase ad reach. We expected to make 14,000 impressions in those three months, earning 420 euros. In the summer, views drop due to fewer app users, so expect that we barely had 4000 views in June and July, earning us only 120 euros. We expect more app users from August to the year-end when we're busiest. We estimate 39,500 impressions on our app, which will net us 1,100 euros later this year. So far, ad views have brought in 1820 euros this year.

Expense Charges:
Marketing and Advertising: Making new housing software user-friendly is a crucial tool that alters how college students find and preserve their ideal living places. Our student rental app lists homes and uses innovative marketing to help students find the perfect home away from home. We tried to market our app all year. Our January marketing campaign started with online surveys and several advertisements. We spent 400 euros, then 450 euros, and it stayed that way until April. We spent 900 euros on marketing our company in May, June, and July because summer meant fewer applicants, so we needed to market more. Most students had gone home for the summer holiday. Thus, more ads are needed for the next school year so they may learn about our program. Our marketing costs would lower after summer to 450 euros since we won't require as much marketing help as before summer. So far this year, our marketing costs are 6,700 euros.

Insurance: Home rentals are risky, thus insurance is essential for a student housing app. Many first-time independent students may experience crashes, theft, or property damage. Getting insurance functions as a safety net, protecting landlords and tenants from financial issues caused by these sudden tragedies. Monthly fees of 150 euros are a compromise between cost-effectiveness and comprehensive coverage. At this price, students can spend some of their budget to secure their living arrangements without spending too much money. With such high costs, insurance frequently covers several potential risks, reassuring students and landlords and making the region a safe learning environment. We recommend spending 150 euros a month to reduce risks and maintain peace between students and property owners within the student housing app.

Loan Payments: Monthly 1090 euro loan payments support the financial security of students who buy homes on the site. These payments cover several expenses like rent and related fees. The unwavering promise to repay 1090 euros each month is both a financial commitment and a tool to keep students safe and stable through our app. The year-to-date number becomes more crucial as the year ends. It tracks the full amount paid back monthly, giving our student rental app a comprehensive corporate history. This total number reveals useful information about our app's financial stability and commitment to their financial obligations, maintaining long-term housing open during school. Actually, this number is 11,900 euros, and we intend to repay a 30,000-euro loan in the first year.

Support and Pay for Customers: Student living requires good customer service tools to provide a great user experience. Given how things change, living with students and managing properties require an immediate assistance system. Students may face technology issues or have questions about their lodging while adjusting to a new place. A specialist support staff is needed to quickly resolve issues and assist landlords and tenants. A monthly payment of 2,000 euros for the support team is fair given their hard work. This team's salary reflects their commitment to deliver exceptional service, resolve many concerns, answer questions, and preserve the app's reputation. Competitive pricing attracts skilled professionals, but it also motivates and engages the support personnel, improving overall student housing app performance and enjoyment.

Renting an Office Space: Renting an office space is crucial for the growth and efficiency of our college housing service. An office is where teams plan and collaborate on projects to fulfil the platform's varied operating needs. Renting an office for 1,500 euros per month acts ethically for numerous reasons. First, the position is crucial for client and staff accessibility. This sum gets you a decent workplace location, making things easier and faster. Next, a good work environment boosts team innovation and productivity, which benefits the app's growth and usability. The price includes all facilities, maintenance, and you get a well-equipped business workstation with extra services. In the highly competitive IT industry, renting an office for 1,500 euros a month demonstrates a commitment to business success, employee satisfaction, and growth of the college housing app. The rental of our office space would cost 18,000 euros per year for the season.

For our student housing software to succeed, we need Legal GDPR Consultants to navigate privacy and data protection laws swiftly. Student housing software always handles private data, thus GDPR (General Data Protection Regulation) must be observed. The right legal GDPR specialists can ensure the app's data management follows stringent legal requirements, reducing legal impacts and data security risks. A fair fee for these experts is 6,000 euros because they are prudent law skill investments. This price includes detailed evaluations, creating strong data security standards, regularly reviewing compliance, and acting fast to changing laws. The 6,000 euros spent on Legal GDPR Consultants were crucial in a highly regulated environment, ensuring app image, user trust, and longevity.

Development Costs:
Software development projects should always include developer consultations. Spending 500 euros on this help is wise. Decisions throughout developers' technical expertise and ideas are crucial. Their feedback improves the project, refines its demands, and addresses potential issues during the growing process. The advice price covers developers' time and skills, ensuring the project benefits from their expertise. The 500 euros reserved for getting developer advice improves the project and it also demonstrates your dedication to a successful result.

Developing an easy-to-use, smooth-running tool through the difficult process of making a student housing app is complicated and requires professional expertise. Hi people. Paying 5,000 euros for this is reasonable and necessary to guarantee app functionality. Important portions of the price include back-end infrastructure, security, software architecture, UI design, and quality assurance testing. For 5,000 euros, you may hire highly skilled developers with knowledge and skill. Well-designed student living software is more likely to succeed, attracting more users, streamlining tasks, and improving user satisfaction—all of which help make the initial investment pay off.

Software and Maintenance Updates: We hope to update our software every three months. Our developers will receive $1,200 per quarter to review certain software features that may need fixing. Keeping the app fresh is good and crucial for client service. Maintenance and updates thus far this year have cost approximately 4,800 euros annually. Developer salaries are crucial to the software industry and demonstrate specific skills and knowledge are in high demand. A $6,000 payment makes logical sense. Euros per month for several reasons.

The job market is full of engineers because software creation involves a profound understanding of complex coding languages, technologies, and systems. A 6,000-euro monthly salary indicates the ability and expertise of developers. Additionally, competitive salaries attract top-notch personnel, producing a skilled and dedicated staff that impacts the quality and efficiency of software development. We seek the best software developers, and they think this salary will get them. We've spent 72,000 euros on this salary this year. Our expenses since January are 151,190 euros. We are happy with this number for our first year and will try to lower it in the future.

4.2) References

1. *Agile teams (2024) Scaled Agile Framework.*
   Available at: https://scaledagileframework.com/agile-teams/
   (Accessed: 2 April 2024).

2. *Vaibhav Anu, Wenhua Hu, Jeffrey C Carver, Gursimran S Walia, Gary Bradshaw, Development of a human error taxonomy for software requirements: A systematic literature review, Volume 103, 2018,*
   Available at: https://www.sciencedirect.com/science/article/pii/S0950584916302373
   (Accessed: 2 April 2024).

3. *Sullivan, C.* (2023) *Students protest in Dublin over accommodation crisis, RTE.ie.*
   Available at: https://www.rte.ie/news/ireland/2023/1004/1408964-student-protest/
   (Accessed: 16 January 2024).

4. *Hughcarrhere* (2022) *Average rent in Ireland is now 50% higher than Celtic Tiger Levels, JOE.ie.*
   Available at: https://www.joe.ie/news/average-rent-ireland-2022-747992
   (Accessed: 16 January 2024).

5. *O'Shea, M.* (No date) *Higher education – key facts and figures 2020/2021, Higher Education Authority.*
   Available at: https://hea.ie/2021/10/11/higher-education-key-facts-and-figures-2020-2021/
   (Accessed: 17 January 2024).

6. *Social media demographics for higher ed—2021*.
   Available at: https://blog.campussonar.com/blog/social-media-demographics-for-higher-ed-2021
   (Accessed: 4 April 2024).

7. *Ryabtsev, A. (2024) Top 6 django compatible hosting services, Software Development Blog & IT Tech Insights | Django Stars.*
   Available at: https://djangostars.com/blog/top-django-compatible-hosting-services/
   (Accessed: 17 April 2024).

8. *Rent Dublin, apartments and houses for rent in Ireland - rent.ie*.
   Available at: https://www.rent.ie/
   (Accessed: 4 April 2024).

9. *Search Ireland's no. 1 property website Daft.ie*.
   Available at: https://www.daft.ie/
   (Accessed: 4 April 2024).

10. *Dublin City University accommodation DCU Accommodation*.
    Available at: https://www.dcuaccommodation.ie/
    (Accessed: 4 April 2024).

11. *Django documentation: Django documentation Django Project.*
    Available at: https://docs.djangoproject.com/en/5.0/
    (Accessed: 6 April 2024).

12. Mitrofanskiy, K. (2024) *How to choose the right database for your needs: IntelliSoft Guide*, *Intellisoft*.
    Available at: https://intellisoft.io/making-sense-of-databases-how-to-choose-the-right-one/
    (Accessed: 16 April 2024).