

Student Housing System

CSC640 Fall 2023

Team Members

- Varsha Annamalai
- Sony Annem
- Mostafa Abdelmegeed
- Keerthi Devireddy

Introduction

When entering a new college town, students often find themselves grappling with many housing-related dilemmas. One of the foremost issues is the trade-off between proximity to the campus and the cost of rent. Many students wish to be as close as possible to their academic institutions for convenience, yet they are often confronted with exorbitant rental rates in such prime locations. This financial strain can significantly affect a student's budget and overall college experience. Conversely, opting for more affordable housing options on the outskirts of town may lead to different complications. Safety becomes a paramount concern, as these areas may not offer the same security and infrastructure as those closer to campus. Students may find themselves navigating unfamiliar neighborhoods with uncertain reputations, raising apprehensions about personal safety and well-being.

Furthermore, the process of searching for housing can be overwhelming due to the sheer volume of available options, each with its unique set of pros and cons. Without a reliable source of information, students may struggle to differentiate between housing alternatives, often relying on word-of-mouth recommendations or unverified online reviews, which can be unreliable and biased. Hygiene and cleanliness are aspects that can become problematic. Students may find housing that appears suitable on the surface but lacks proper maintenance and cleanliness standards, leading to unpleasant living conditions that can affect their academic performance and overall well-being.

Considering these challenges, our software solution appears to be a valuable resource for students. It empowers them with a comprehensive and unbiased overview of available housing options, considering factors like cost, proximity, safety, and cleanliness. By harnessing the collective wisdom of Google reviews and the ability of specialized assessors, our platform aims to alleviate the housing-related stress that students face, ensuring they can make well-informed decisions that align with their unique needs and circumstances.

This document will supply a comprehensive understanding of the system's functionalities, user roles, and the specific requirements necessary to bring the Student Housing System to life. It includes detailed descriptions of each module, use case scenarios, and system architecture to ensure successful project planning and development.

In the following sections, we will delve into the specific requirements for each module, detailing user interactions, system behaviors, and the underlying data structures. The goal is to create a robust and user-centric platform that simplifies the student housing experience, making it easier for students to find suitable accommodations and landlords to manage their properties efficiently.

Glossary

Hereby after in this document **Admin** and **Reviewer** terms will be relating to the same actor, the justification for using these two terms for the same actor was that in some use cases “Reviewer” was descriptive of the nature of the use case.

The terms **Property**, **Listing**, and **Unit**, all relate to the same thing.

Scope

The Student Housing System encompasses four key modules:

Module	Assignee
Admin/Reviewer	Keerthi Devireddy
User	Sony Annem
Landlord	Varsha Annamalai
Google API	Mostafa Abdelmegeed

Use Cases

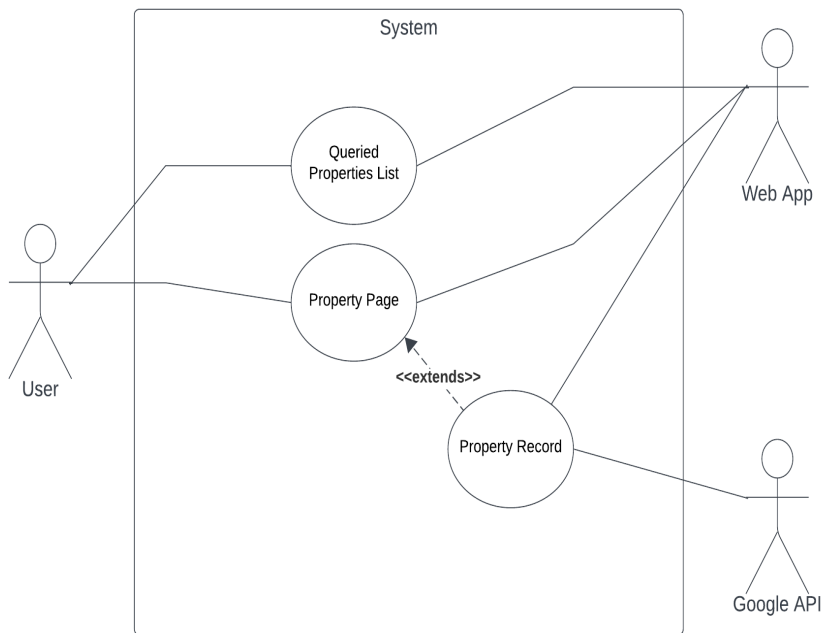


Figure 1 Use Case Diagram by Mostafa A.

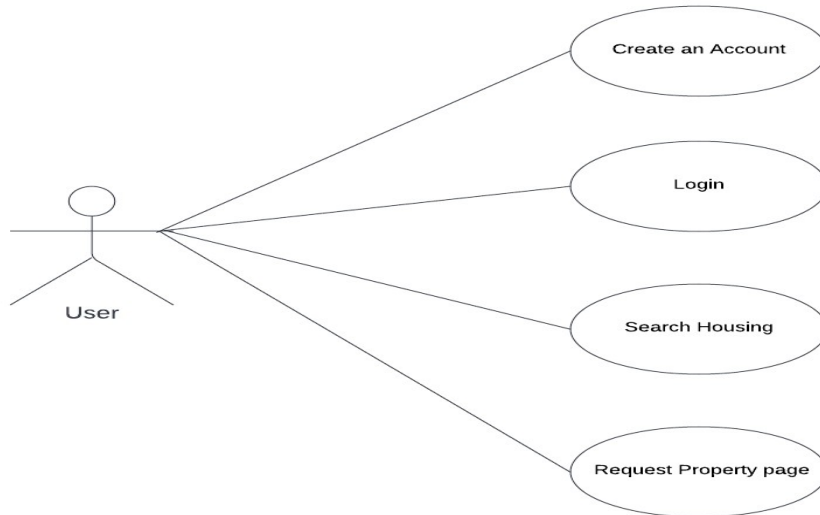


Figure 2: Use Case Diagram by Sony Annem

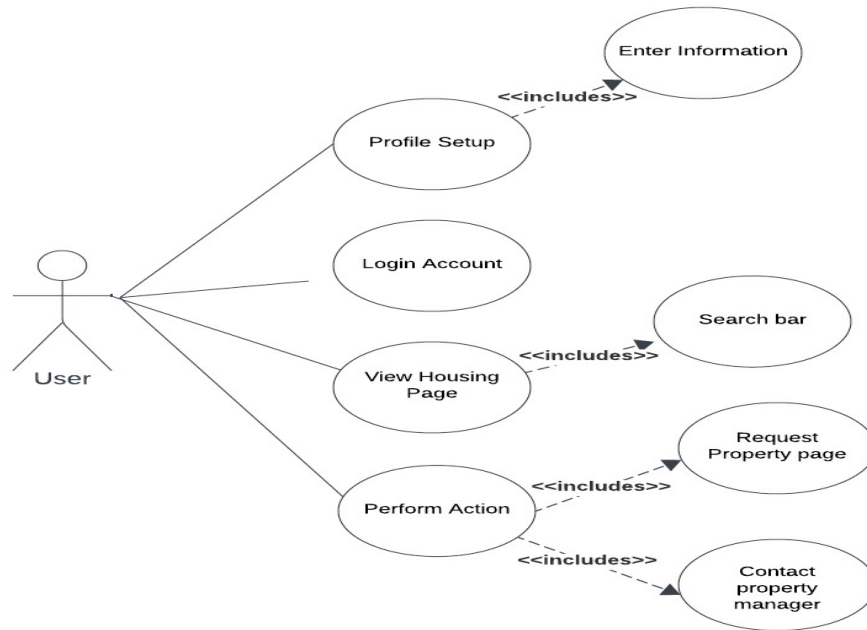


Figure 3: Use Case Diagram by Sony Annem

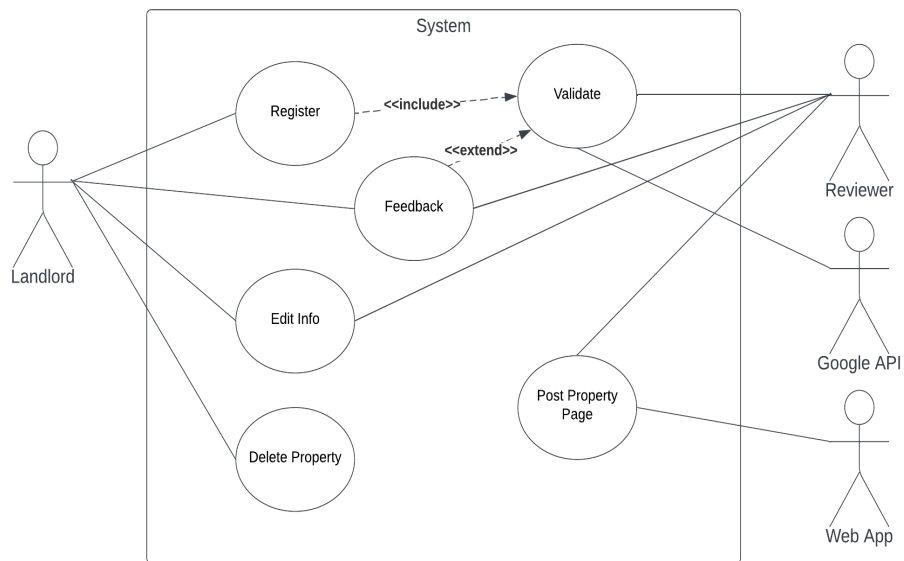


Figure 4: Use Case Diagram by Mostafa A.

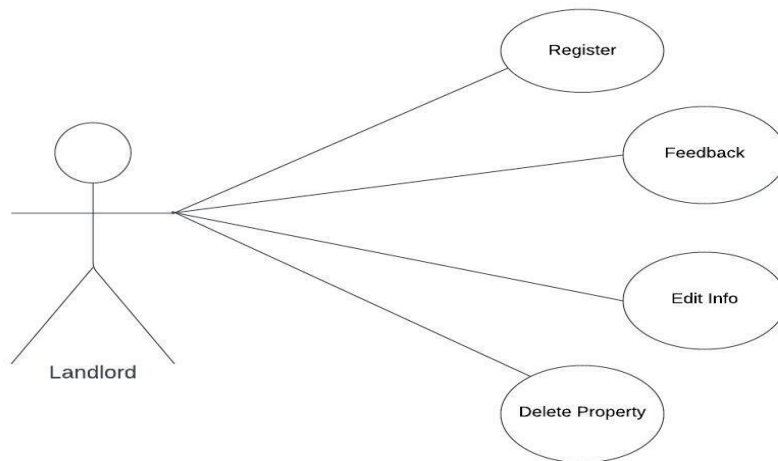


Figure 5: Use Case Diagram by Varsha Annamalai

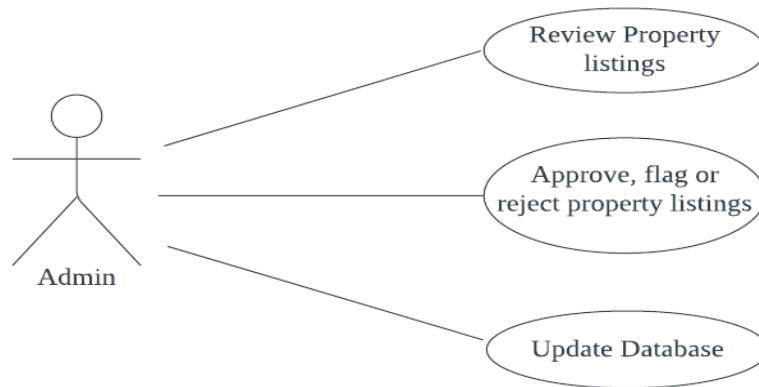


Figure 6: Use Case Diagram by Keerthi Devireddy

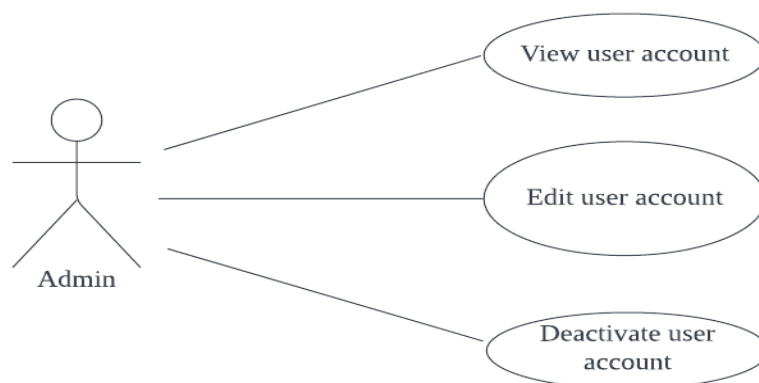


Figure 7: Use Case Diagram by Keerthi Devireddy

Descriptions

Use Case 1

A user starts using the system by triggering a search query, the web application handles retrieving and showing the search results as a list of records for the user. The user can choose one of the search results records to get a detailed description of the property selected, at the user's request, the web application starts retrieving the necessary details from the database, and communicate with Google API, Google API compliments the retrieved data with any additional information about the property (e.g., location, reviews, ratings), the final page is sent to the user by the web application.

Use Case 2

Users can securely log in to the portal and supply their profile details, including contact number, student ID, date of birth, and gender. They can then start housing searches based on specific criteria, such as area or university names, using advanced filters like proximity to universities, amenities, and housing types. The portal offers a map-based interface for visualizing housing options, allows users to save their search criteria for future notifications, and supplies sorting and comparison options. Users can click on housing listings to view comprehensive property details, high-quality images, and contact information for property owners or managers. Additionally, they can access user reviews and ratings to make informed housing decisions.

Use Case 3

In this use case, the user begins by completing their profile setup, supplying personal information and optionally a profile picture. Once logged in, they gain access to the user's dashboard or homepage of the student housing portal, where they can start a housing search. Utilizing search criteria such as location, price range, and property type, the user triggers the "Search Housing" functionality. The system responds by presenting a list of housing options that match the specified criteria, including property names, locations, prices, and brief descriptions. The user can then click on a specific housing listing to access the "Request Property Page." On this page, they find comprehensive details about the chosen housing property, encompassing property specifics, high-quality images, contact information for the property owner or manager, and user reviews and ratings if available. From here, the user can take various actions, such as reaching out to the owner, saving the property for future reference, or reporting any issues, aiding them in making informed decisions about their housing needs.

Use Case 4

In this use case, landlords have the capability to manage their properties within the system. It begins with the "Property Registration" process, where landlords can send property information, including property name, address, and description. This information is then subject to review and acceptance by the admin or reviewer. If the registration is rejected, the landlord receives detailed feedback through the "Property Feedback" mechanism. In response, landlords can make necessary updates to their property pages based on the provided feedback, and once again, send

the revised information for admin review. Landlords also have the option to start "Property Deletion" if they wish to remove a property from the system. This suite of functionalities allows landlords to effectively manage their property listings and keep up-to-date and correct information within the housing portal.

Use Case 5

The admin has the highest level of access and control. This use case outlines the actions and responsibilities of the admin in keeping a high-quality and correct database of listed properties. The admin looks after and manages property listings within the system. Admin's role in managing property listings is critical to ensure that the Student Housing System keeps reliable, and up-to-date property listings in the backend database for the benefit of both students(users) and property owners. This process helps create a trustworthy and efficient platform for all users.

Use Case 6

In this use case, the admin acts as a reviewer. The admin who has complete access to manage the user related information, oversees and handles the user account data. He/she can manage the data by performing view, edit and deactivate functionalities. It is a critical step to perform the feasible operation on user data and save the changes in order to maintain up-to-date and reliable user account information. This process helps in ensuring proper and accurate user information within the system.

Use Case 7

This use case outlines the actions in ensuring proper user information within the system. The admin handles overseeing and managing User Accounts within the Student Housing System. Admin's role is critical in keeping up-to-date user information. This process helps create a trustworthy and efficient platform.

Scenarios

Scenario 1

Initial Assumption

User knows the region of interest to search for housing information in.

Normal

- Users enter their region of interest for housing search (city, town, or college).
- The web application queries the database using the user's input.
- It retrieves a list of all possible properties from the database.
- Optionally, the web application may retrieve data from Google API to supplement the property information.
- Once the retrieval process is complete, a detailed property page is displayed to the user, presenting comprehensive information about the selected property.

What can go wrong?

- If the search input keywords are not found in the database or are not recognized as a known region, a message is displayed to inform the user.
- The message notifies the user that there isn't enough data about the region they specified.
- The user is then prompted with options to either continue navigating to other regions or confirm if there was a typo in their input.
- If the search input keywords are not found in the database or are not recognized as a known region, a message is displayed to inform the user.
- The message notifies the user that there isn't enough data about the region they specified.
- The user is then prompted with options to either continue navigating to other regions or confirm if there was a typo in their input.

System State on Completion

A retrieved informative property page is shown to the user.

Scenario 2

Initial Assumption

The user is logged into the housing portal.

Normal

- User specifies housing preferences, refines with advanced filters, and visualizes options on a map.
- User saves the search and sets up notifications for new listings.
- They use sorting and comparison features to evaluate properties.
- Users access detailed property information with high-quality images.
- User contacts property owner/manager and checks reviews for decision-making.

What Can Go Wrong

- Slow image loading due to poor internet.
- Delayed notifications for new listings.

State After Completion

User has explored, filtered, compared, and contacted a potential housing option, ready to make an informed decision.

Scenario 3

Initial Assumption

Landlord is a registered user on the system and has the required information to send a form to the reviewer.

Normal

- Landlord initiates property registration request by filling a form with all required information (e.g., address, images, Google Maps URL, amenities).
- The submitted form is sent to a reviewer for data validation and cross-checking with Google Maps.
- If the data is confirmed, the property can be posted on the system.
- In case of missing or incorrect information, feedback is sent to the landlord.
- Waitlisting occurs until the landlord supplies missing information.
- Landlords with registered properties can edit information but need reviewer confirmation before updates are posted.
- Landlords with registered properties have the choice to remove any posted property as needed.

What can go wrong?

- An overwhelming influx of property registration requests and an inadequate number of reviewers may lead to approval bottlenecks.
- If multiple requests are sent for the same property, reviewers prioritize and approve only one of them while ignoring replicated submissions.
- Repeated submissions with false or inaccurate information could result in the reviewer blacklisting the landlord.

System Stage on Completion

The admin is logged into the Student Housing System as an authorized administrator. The system is running without any technical issues or errors. The admin has access to all the existing property listings (the list of the properties registered by the property owners/landlords) and user account information in the system.

Scenario 4

Initial Assumption

If a property listing is correct and follows policies, the admin approves the unit, making it visible to users. If a listing requires modification, the admin can edit the listing information, if necessary. However, if there are any missing, or incorrect information, feedback is sent back to the landlord, and the property is waitlisted until the landlord return with the missing information.

Normal

- The admin reviews the property listings. The admin checks the accuracy of the property details, such as property name, description, location, amenities, and availability. The admin assesses whether the property adheres to the system's policies and guidelines.
- If a property listing is correct and follows policies, the admin approves the listing, making it visible to users. If a listing requires modification, the admin can edit the listing information, if necessary. However, if there are any missing, or incorrect information, feedback is sent back to the landlord, and the property is waitlisted until the landlord return with the missing information.
- If a listing has inaccurate information, the admin communicates with the property owner about the issues. The admin may reject the listing if the issues are not resolved.
- After approving or making edits to the existing listings, the admin updates the database to reflect the changes.

What can go wrong?

- Property owners may not respond promptly to admin communications, potentially causing delays in resolving issues with listings.
- Technical issues, such as database errors or server downtime, may occur, hindering the admin's ability to access or update property listings.

System Stage on Completion

The admin module assumes that there are existing user accounts in the system. The user who has administrative access logs into the system with the necessary credentials. He/she has complete access to user account data and management functionalities.

Scenario 5

Initial Assumption

The admin module assumes that there are existing user accounts in the system. Admin user has logged in with the necessary credentials. Admin has access to user account data and management functionalities.

Normal

- Admin starts the 'User Account Management' use case.
- System displays a list of user accounts with options: view, edit, or deactivate.

- Admin selects a specific user account to manage.
- Deactivate the account if the information is inappropriate or user data is no longer required.
 - View user's details.
 - The account information is updated in the backend database by the system.
 - Deactivate the account.
- If editing, admin makes necessary changes and saves them.
- System updates user account information in the database.
- If deactivated, the system sets the user account to an inactive state, preventing further access.

What can go wrong?

- Admin may accidentally select the wrong user account for editing or deactivation.
- Issues, such as connection errors, may occur while saving edited information.
- Deactivating an account may raise concerns and inquiries from the affected user.

System Stage on Completion

On completion, the system will have updated user information or deactivated accounts, ensuring proper management and security within the system.

User Requirements

1. User Registration and Authentication
 - 1.1. Users must be able to register using their personal information.
 - 1.2. Users should be able to log in securely to access the application's features.
2. Search and Filtering
 - 2.1. Users can search for housing units based on criteria like price range, proximity to campus, number of bedrooms, and amenities.
 - 2.2. With a pre-existing database of housings around campuses in the US, a correct query should be sufficient to retrieve multiple entries.
 - 2.3. The list of records presented to the user should include simple and clean information about the property (e.g., name, address, price)
 - 2.4. Results should be displayed with relevant property details.
 - 2.5. Fully verified properties with functional Google API link should be shown on the top of the list.
3. View Property Details
 - 3.1. Users can view comprehensive information about a selected property, including photos, contact details for the landlord, and a map showing the property's location.
4. Property Submission
 - 4.1. Any registered user can access the platform and send a request to post their property on the system.
 - 4.2. Landlords can send their property details, including price, location, number of rooms, and amenities.
 - 4.3. The portal should supply clear instructions on how to fill the required fields, and the specifications required for images to be registered (size < 2 MB) per image.
 - 4.4. The landlord should be able to add multiple amenities in a separate part of the portal, each amenity is selected out of a list.
 - 4.5. Landlord should enter a proximity to different campuses around their property in miles.
 - 4.6. Prices are presented per month, and any other fees should be included in the overall price.
 - 4.7. Submitted properties must be reviewed and approved by an admin or reviewer before being added to the database.
5. Location and Mapping Integration
 - 5.1. The application should integrate with Google Maps to display the location of each property on a map.
 - 5.2. Users should be able to see the proximity of a property to the campus and other landmarks.
 - 5.3. Users can use map-based features like zooming, panning, and street view to explore the neighborhood.
6. Property Reviews and Location Information

- 6.1. Property listings should display Google Maps' location data, including nearby restaurants, grocery stores, public transportation, and educational institutions.
- 6.2. Users can access reviews and ratings for the neighborhood and nearby services using Google Maps data.
- 6.3. In the property detailed page, a couple of reviews from google should be shown to the user, preferably 2 of the highest reviews, and another 2 from the worst reviews.
- 6.4. Use Google API to retrieve average rating of properties and display it next to every property record.
- 6.5. In case Google API does not offer enough information, or no information at all, a notification and an email shall be sent to the landlord to update the record on Google, notifying a landlord give them a month to update their record, or their property will be unverified and will not show as much in queries.
7. Landlord Notification System
 - 7.1. Landlord must be informed once their property was removed or no longer accessible via google maps.
 - 7.2. A weekly warning will be sent to the landlord in case the property still is inaccessible via google maps.
8. Up to Date System
 - 8.1. If a listing stays inaccessible via google maps for more than a month, it should be removed from the system and reported to the landlord.

Non-functional Requirements

Product Requirements

- The Student Housing System should be available to all users 24/7, a notice of a week at least should be put on the landing page in case a planned maintenance was coming.
- In case of failures and unplanned downtime, the system should not be down for more than 30 minutes at a time, with an overall total of 5 hours downtime per year.
- Queried results should not take more than 2 seconds to present to the user.
- In case of failures in communication with Google, a suitable reason should be shown to the user wherever possible and visible.
- The system should be designed to handle a growing database of properties and users.
- The user interface should be intuitive and user-friendly.

Organizational Requirements

- Reviewers base should be diverse when it comes to demographics, trained, and locals to their region whenever possible.
- Landlords will need to authenticate themselves as property owners in cases of conflicts, or when multiple requests are done for the same property.
- Database shall be hosted on a cloud-based service.
- Maintain comprehensive technical documentation for the application's architecture, APIs, and data schema.

External Requirements

- The system shall implement users' data privacy provisions, never exposing users' data to landlords without their consent.
- Ensure compliance with local housing regulations and privacy laws.