ROOMIEMATCH

NG ROOMMATES MADE EASY

Table of Contents

Executive Summary

Packing up and moving is hard, but finding compatible roommates makes this decision even harder. Especially for university students who move from one state to another or one country to another to attend their choice of university. Some individuals are entirely new to sharing a living space with someone else. Thus, finding a compatible roommate can be challenging, time-consuming, and stressful as students have multiple other things to work out when starting a new semester simultaneously. The RoomieMatch extension, which will be implemented on the current MySlice at Syracuse University, is a comprehensive solution for these problems. The application includes several features to ensure a positive user experience, such as user profile creation, roommate search, sending and receiving connection requests, and filters that the user can apply to filter out individuals based on preferences like location, budget, lifestyle habits, and interests. It will also help the system match them with potential roommates who meet their criteria. Once the algorithm matches students with potential roommates, they can send a connect request to the other person, and once this request has been accepted, they can interact with each other. This extension will not only provide students a convenient way to find compatible roommates but also fosters a sense of community among students, this would especially benefit the first-year students who will be new to the area and are looking to connect with their peers. RoomieMatch will comply with various data protection, security, and breach notification laws. This will be done by implementing appropriate security measures to protect user data and notifying users in the event of a data breach and will have procedures in place to handle such incidents.

Project Name

RoomieMatch

Project Sponsor

Syracuse University

Project Overview

RoomieMatch is a new extension of the current MySlice application to be used by students at Syracuse University. MySlice is a web portal for students & staff of Syracuse University; it acts as a central platform and provides access to academic and administrative information, which includes features like current class schedules, billing statements, registration for courses in the upcoming semester, and a range of other services like career service and access to various campus resources like health services and IT support. Currently, the current and incoming university students find housing using social media platforms like Facebook and WhatsApp groups which is tedious and chaotic and bears poor results as people are unable to find compatible roommates. The proposed RoomieMatch extension will make this task more manageable as it will be a web-based application design used to find compatible roommates more efficiently by providing filters that individuals can use to screen out others based on preferences like lifestyle habits, location, budget, and common interests, and provides external links to housing which they can use as an additional resource in finding a house. The system will also allow users to communicate with their compatible roommate once the other sends and accepts a communication request. This will enable them to communicate and get to know each other before living together. To ensure the safety and security of the users on the platform, currently, only students attending the university who have their SU credentials and are vetted by the university system can use this application. The platform will also have a rating system for users who can rate their experience, which will gather insights about the application and make way for improvements to make the whole experience more seamless. When new features and updates are made to the application, these will be pushed out to the users. This regular action will improve the functionality and performance of the application, adding to the user experience and thus leading to increased user satisfaction and customer loyalty.

Project Objectives

1. Develop a user-friendly interface that allows users to connect and match with potential roommates easily.
2. Put in place a robust algorithm for matching roommates.
3. Allows for customizing user profiles by adding summary, hobbies, stream, cleanliness requirement, budget, location, etc.
4. It has a chat interface for communicating with matched users.
5. Has a feedback system to capture more insights & improvement areas in the application.
6. It should integrate seamlessly with the existing MySlice portal.

Project Scope

1. Should include a basic user flow of accepting and rejecting communication requests
2. Currently, RoomieMatch will be under MySlice's authentication only.
3. Currently, RoomieMatch will not have any payment system for charging users for their usage, this will be later added based on the success criteria met.
4. The application will not have any major change to MySlice application and will only be limited to adding an option to navigate to the RoomieMatch application.

Assumptions

1. The existing technical infrastructure for mySlice is sufficient to support the development and integration of RoomieMatch on mySlice.
2. The technical stack will be shared for different operating systems and will have common implementation for most of the project with little maintenance for cross-operating system support.
3. Users will require minimal support in using RoomieMatch, and support will only be available via email.
4. Most mySlice users are the target demographic for RoomieMatch and require roommates in the future.

Dependencies

1. Might require additional resources and legal experts for any complaint with regulations related to housing and roommate-matching services
2. Availability of existing technical infrastructure is to be made, and any technical issues in infrastructure could impact the development & performance of RoomieMatch.
3. Engagement in the application is dependent upon successful marketing and promotion events.

Constraints

1. The application must be developed and deployed using the programming languages supported by MySlice.
2. Adding this feature to MySlice should maintain the function and performance of any other features provided on the platform.
3. This application should be built and deployed within the infrastructure provided by the MySlice platform.
4. The application should follow data privacy laws and comply with all other regulations.
5. As the service grows, the application should have the ability to accommodate a large number of users and data
6. The application should be developed and thoroughly tested to ensure it is reliable. This will reduce downtime due to bugs.
7. The application should adhere to the Syracuse University branding.

Criteria of Success

1. The number of users that establish profiles, connect with potential roommates, and participate in chat chats can be used to gauge the application's success. High user engagement shows that the software is filling a market demand and offering value to its users.

2. Ratings from users are significant measures of user satisfaction. The application satisfies user needs and delivers a favorable user experience if it has a high rating.

3. The application should have a high uptime and minimum downtime to guarantee that users can use the service when needed. User engagement and satisfaction will rise with high availability.

4. The application should be able to handle a large number of users and data with quick response times. A great user experience will be ensured and will help boost engagement and satisfaction with quick and dependable performance.

5. The effective implementation of data privacy and security safeguards will increase user engagement with the application and foster a sense of confidence among users.

Requirements

Data Collection Approach

Technique	Description	Data Type	People Involved
User Surveys for Filters to be added	Create survey forms to collect information about roommate services requirements about filters required and to be added to Roomiematch by Admin	Qualitative	Product Team
Data Analytics	Collect engagement data on MySlice for each touch/click and track timing metrics to evaluate existing engagement on MySlice to measure success criteria	Quantitative	Product Team
Development Infrastructure Requirements	Gather information about current MySlice hardware and software infrastructure to estimate if any more infrastructure is to be	Quantitative	Development Team

	added		
Housing Providers Team	Partner with housing providers such as student housing facilities, and apartments to gather data on residents and their preferences which will help in development of the algorithm	Qualitative	Executive Team

Table 1.0 : Data Collection Approach

Functional Requirements

1. **User Profile Creation**
   a. Users must create their profile & without profile features - roommate search, chat will not be accessible.
   b. Users can't change their registered mail address and will be tied to mySlice logged-in email address.
   c. The user's age will be auto-calculated and can't be modified. It will be pulled from mySlice information database.
   d. Users can add and change their preferences for the fields made available by the admin.
   e. Users must add mandatory fields configured by the admin, and can't be left empty.

2. Roommate Search

 a. Users request candidates to send connection requests to; results will be listed for pairs found by the algorithm.

 b. Users can add/remove filters to finetune the search.

 c. Search to limit by top 10 users.

 d. Currently, users can request unlimited searches, but implementation supports limitations for the future paid model.

3. Sending and Receiving Roommate Connection Requests

 a. Users can send a connection request to the received output from the search.

 b. Users will be notified upon receiving a connection request via registered email.

 c. The email will include a redirection to the application to view the connection request.

 d. The receiver can view the entire profile of the sender.

 e. The receiver can either approve or reject the request.

 f. The sender will be notified when the receiver accepts the request.

 g. The number of connection requests a user can send will be unlimited as of now; implementation to support limitations for the future paid model.

4. Chat

 a. Receivers, only once accepting, can chat with the sender.

 b. Chat can be done from both sides.

 c. Both ends will be notified by push notification on receiving a text.

 d. Text will be limited to 1000 characters.

5. User Feedback and Ratings

 a. Feedback popup will be shown to users every time a user has been engaging in chat for a day with another user.

 b. The feedback popup will contain a star-based rating system ranging from 1 to 5. This field will be mandatory

c. The feedback popup will also contain a text area to enter detailed feedback. This field will be mandatory.

Non-Functional Requirements

1. **Privacy**
   a. User data should be secured with industry-standard encryption.
   b. User consent before collecting engagement data
   c. Data deletion requests can be taken with the agreement of 30 days.
   d. Data sharing with third parties (for example - housing providers) will be disclosed before and requires explicit consent while registering.
   e. Users can opt out of data sharing with external third parties.

2. **Availability**
   a. Minimum uptime of 95% of the roommate services
   b. Application monitoring to identify failures and plans to mitigate them should be made.
   c. Maintenance/Downtime periods will be at night, from 12 am EST to 5 am EST, which can be of any duration.
   d. Application status is to be made available and integrated with mySlice status page.

3. **Performance**
   a. Maximum response time of search feature of 5 seconds
   b. Maximum response time of other requests of 2 seconds
   c. The application should be able to handle 100 parallel searches on the matching algorithm and a minimum of 1000 users load per second.

4. Usability

 a. The application should have a consistent design across all pages and functionalities.

 b. Error messages should be present in case of any failures in the system.

 c. Error popups are to be present throughout the application to minimize user errors.

 d. Application documentation will be present on MySlice website only, with a FAQ section to guide users.

5. Compatibility

 a. Support all browsers supported by MySlice, such as Google Chrome, Firefox, and Brave.

 b. Support all mobile devices of various screen sizes, which mySlice supports.

6. Maintainability

 a. Maintenance will be performed as per existing MySlice maintenance schedules.

 b. Support will only be provided by email with documentation and FAQ on MySlice website.

Use Case Analysis

Use Case Glossary

Use-Case Name	Description	Associated Actors
Enters Personal Preferences	The student can Enter their Preferences before continuing to search for Roommates.	Student
Search for Roommates as per Criteria	The student can search for roommates according to the Criteria they want to focus on.	Student
Apply for Available Filters	The student can then apply filters available to them along with the preferences.	Student
Find the Perfect match to Raise Roommate Request	Once the student finds a perfect match according to their preferences and filters, they can raise a request to the other student to become a roommate.	Student
View and accept Requests	The student can view and accept the requests they have received from Potential Roommates.	Student
Request for Matched Details	Once the Match is made, the students can request for their communication details.	Student

Send feedback	Students after each match get to fill a feedback form which then helps the admin know the system better	Student
Add New Student Details	Admin adds new student details to the system.	Admin
Add link to OFF/ON Campus Housing Selections	Admin could add the housing links for the students to use to lease a place	Admin
Add filters to the Search Criteria Section	The Admin could add search filters according to feedbacks and demands, to help students	Admin
Push Notifications about rules and changes	Admin could send updates and changes to students as announcements/notifications	Admin
Verify Changes made to Source Data	Admin needs to verify any changes made to existing student data and update accordingly	Admin
Request Feedback Reports	The Admin could request students to fill the feedback forms.	Admin

Table 2.0 : Use Case Glossary

Use Case Narrative:

Name:	**Add New Student Details**
Actor:	Admin
Description:	Admin adds new student details to the system.
Successful Completion:	1. The User enters their login credentials. 2. The system would know if it's an Admin or Student. 3. The system will then display the options to choose if it's a new student or old student. 4. User enters New Student Details. 5. The system will then output Student Details entered for verification before committing the changes. 6. The user will then Commit the changes.
Alternative:	Admin could use the SQL Interface to access the same, but not without login.
Pre-condition:	The User needs to be logged in First.
Postcondition:	Post successful login, the user will be able to select if they want to enter new details or update old ones.
Assumptions:	The user knows which Student details are new and which needs just an update.

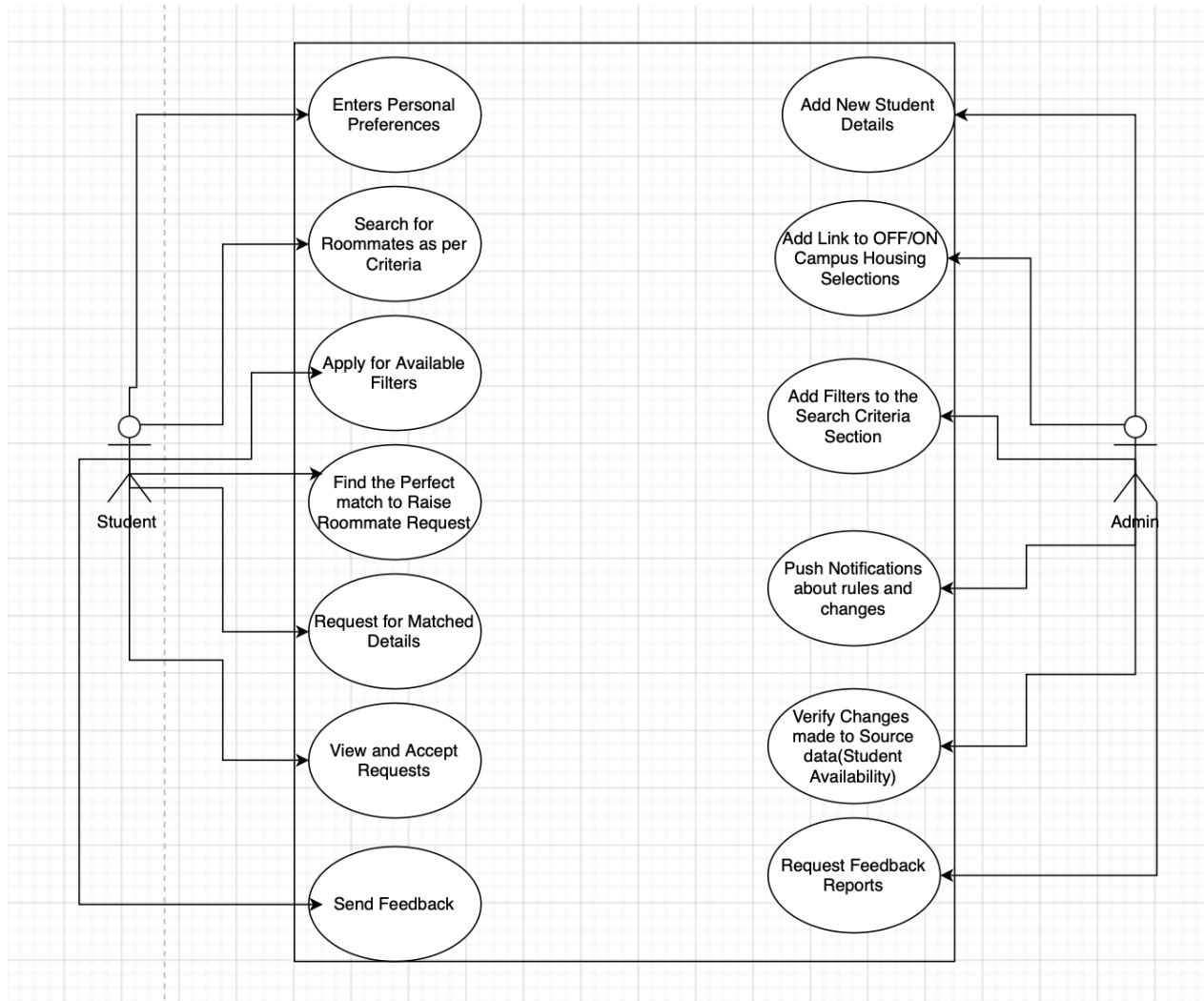Table 3.0 : Use Case Narrative

RoomieMatch Information System

Fig1.0 : Use Case Diagram For RoomieMatch

Data Flow Analysis
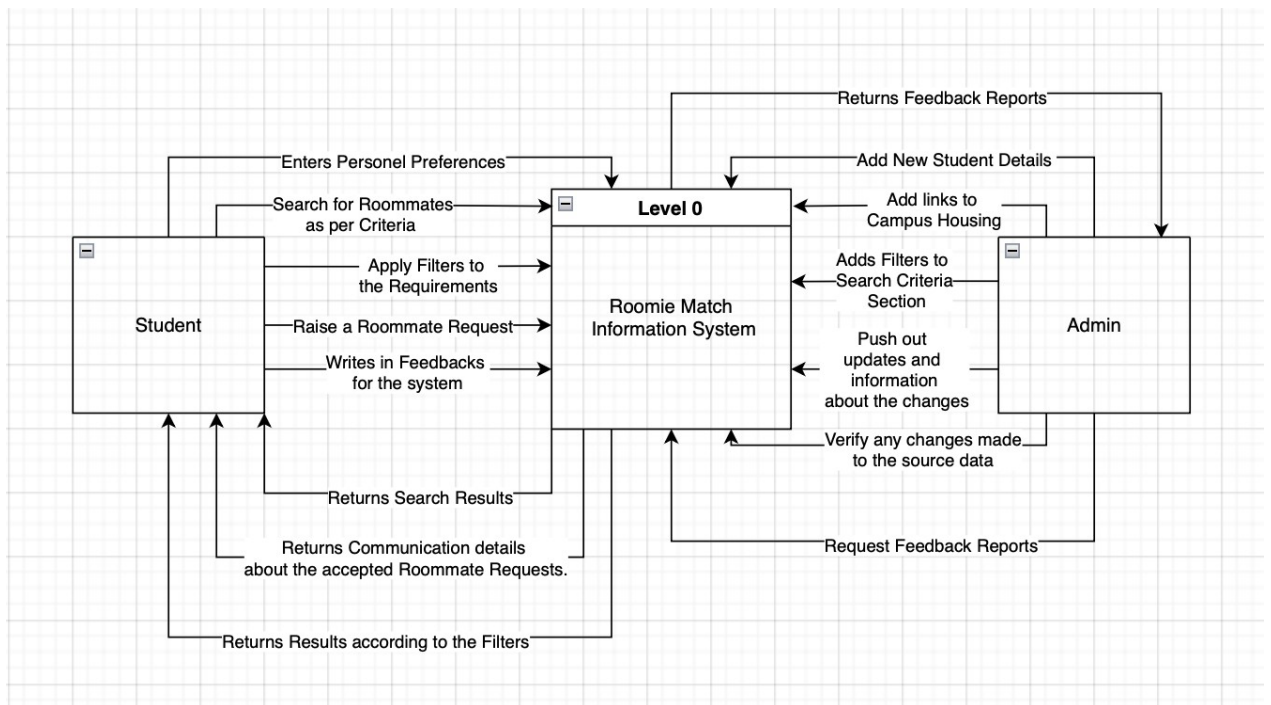
1. Context Diagram

Fig 2.0 : Context Diagram

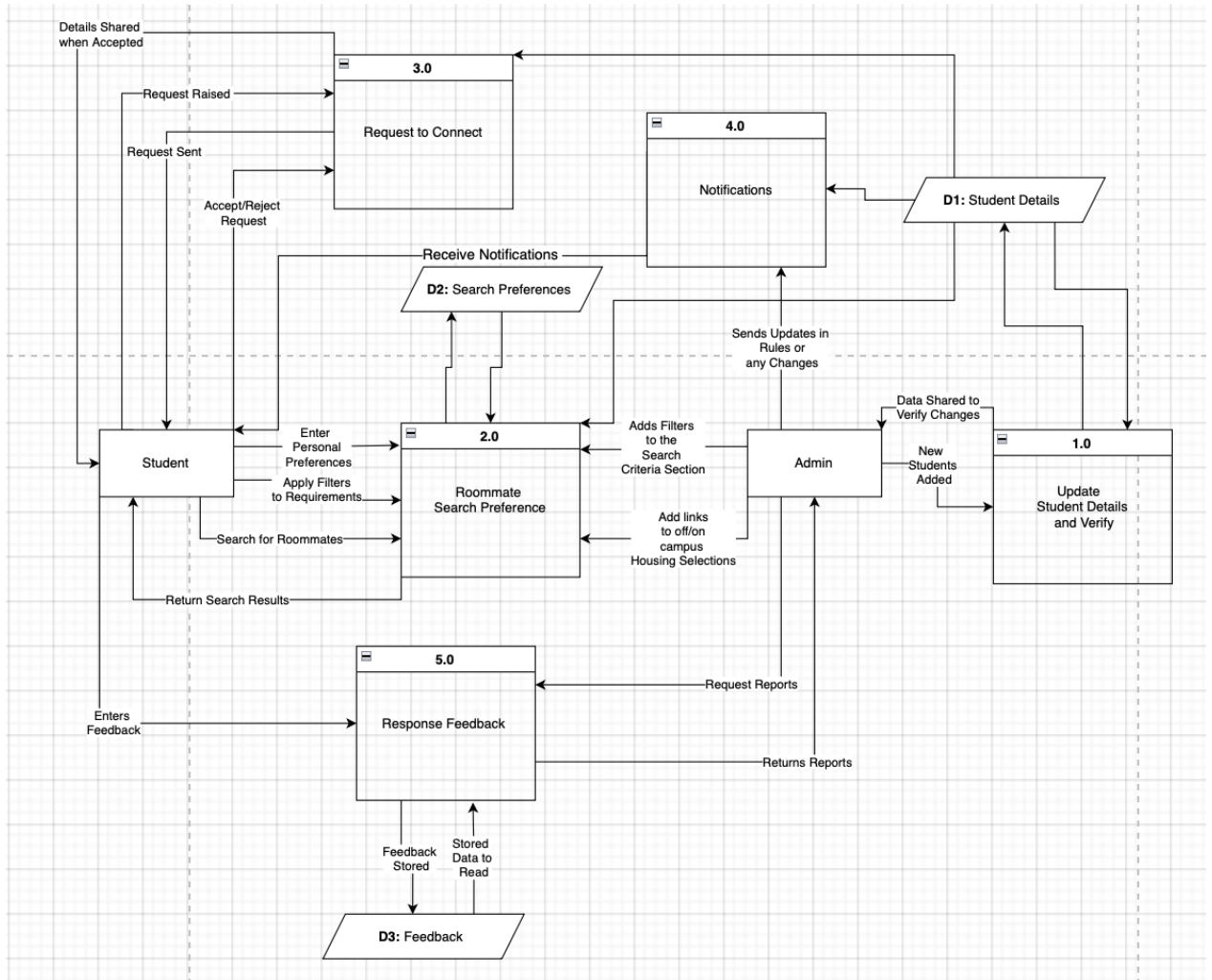## 2. Level 0 Data Flow Diagram

Fig 3.0 : Level 0 DFD

3. Level 1 Diagram

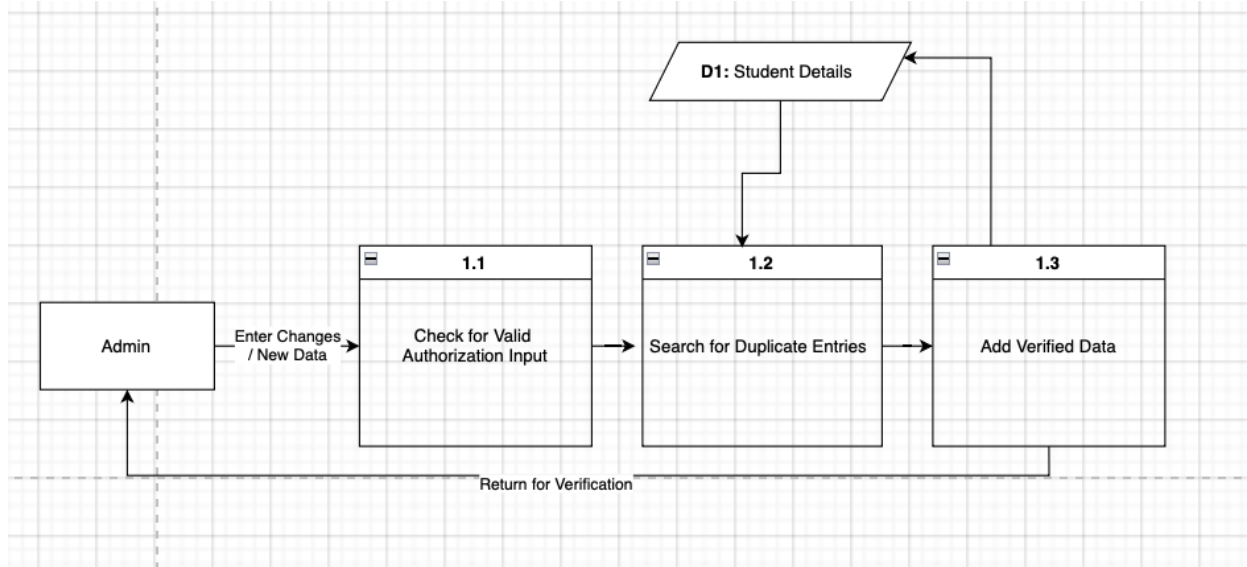## a. 1.0 Update Student Details and Verify

Fig 4.0 : Level 1 DFD: Update Student Details
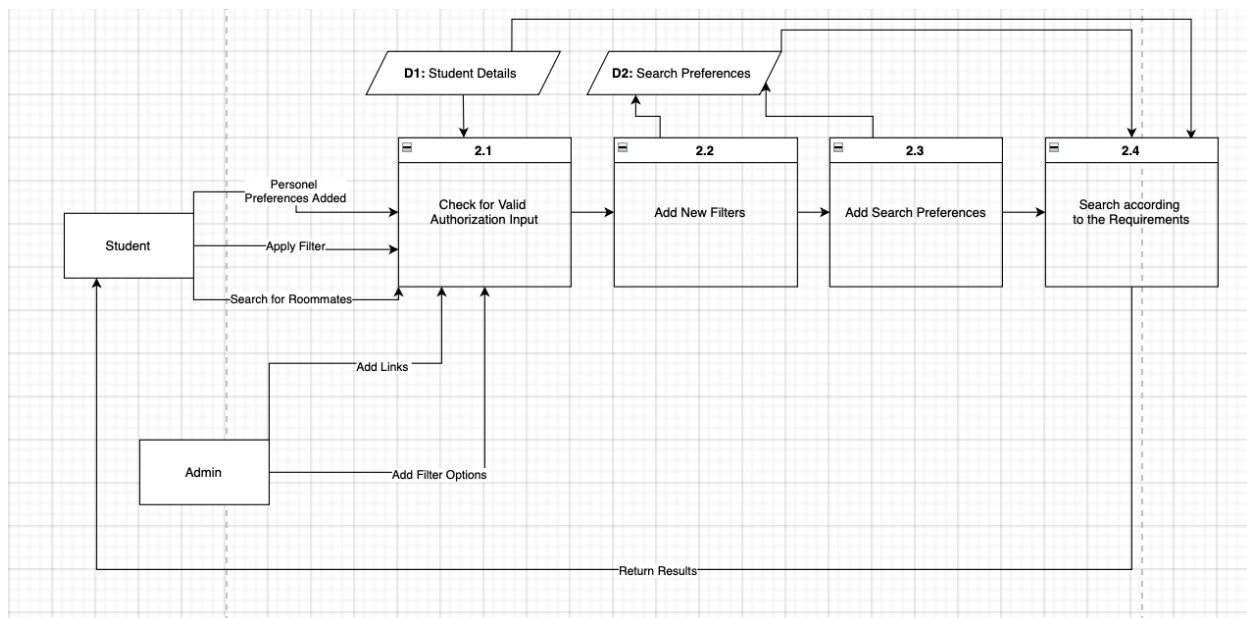
b. 2.0 Roommate Search Preference

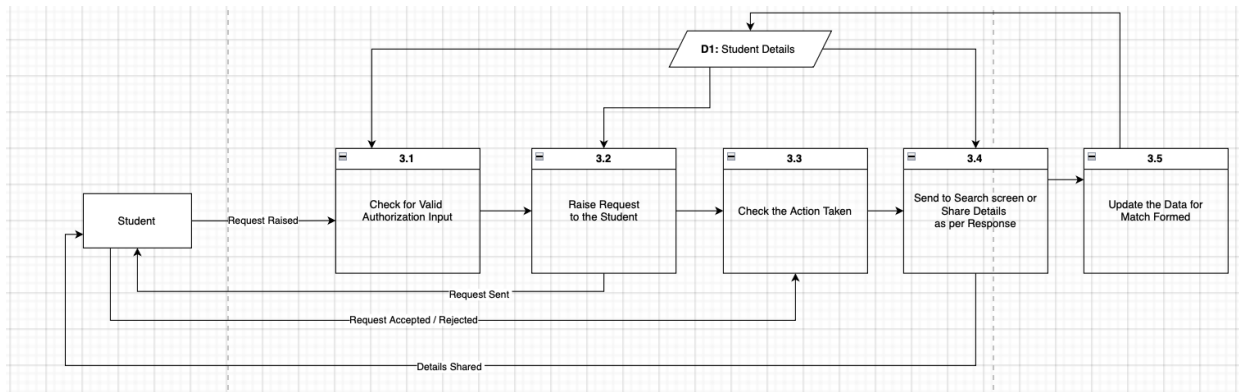Fig 5.0 : Level 1 DFD: Roommate Search Preference

c. 3.0 Request to Connect

Fig 6.0 : Level 1 DFD: Request to Connect

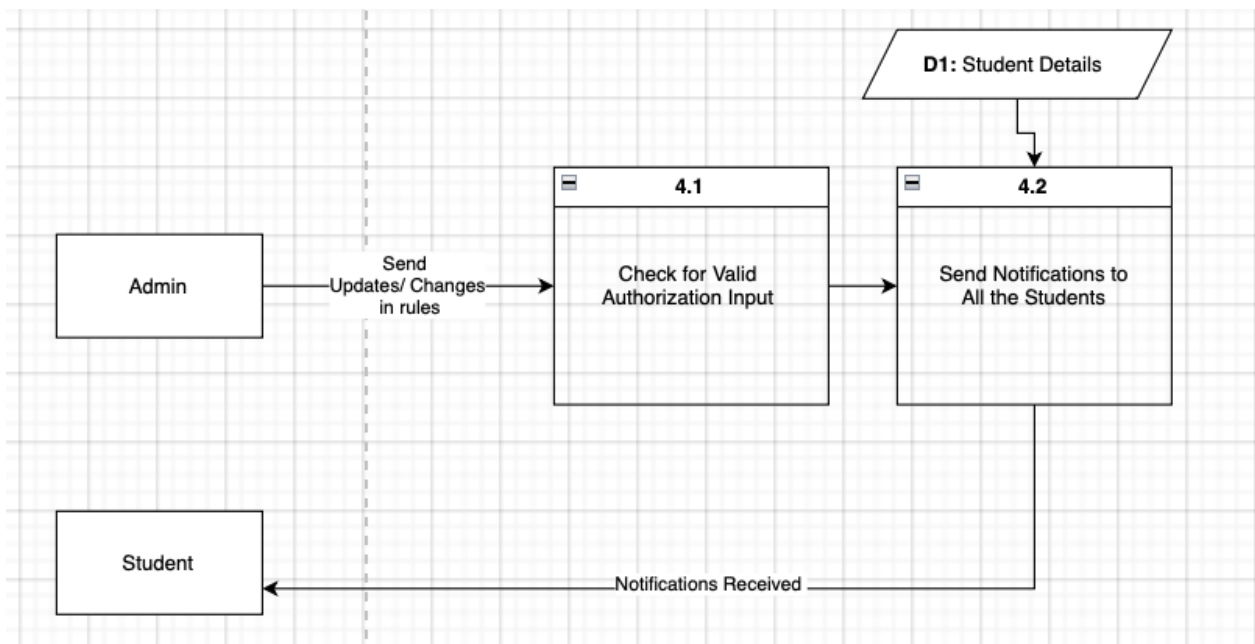### d. 4.0 Notifications

Fig 7.0 : Level 1 DFD : Notifications
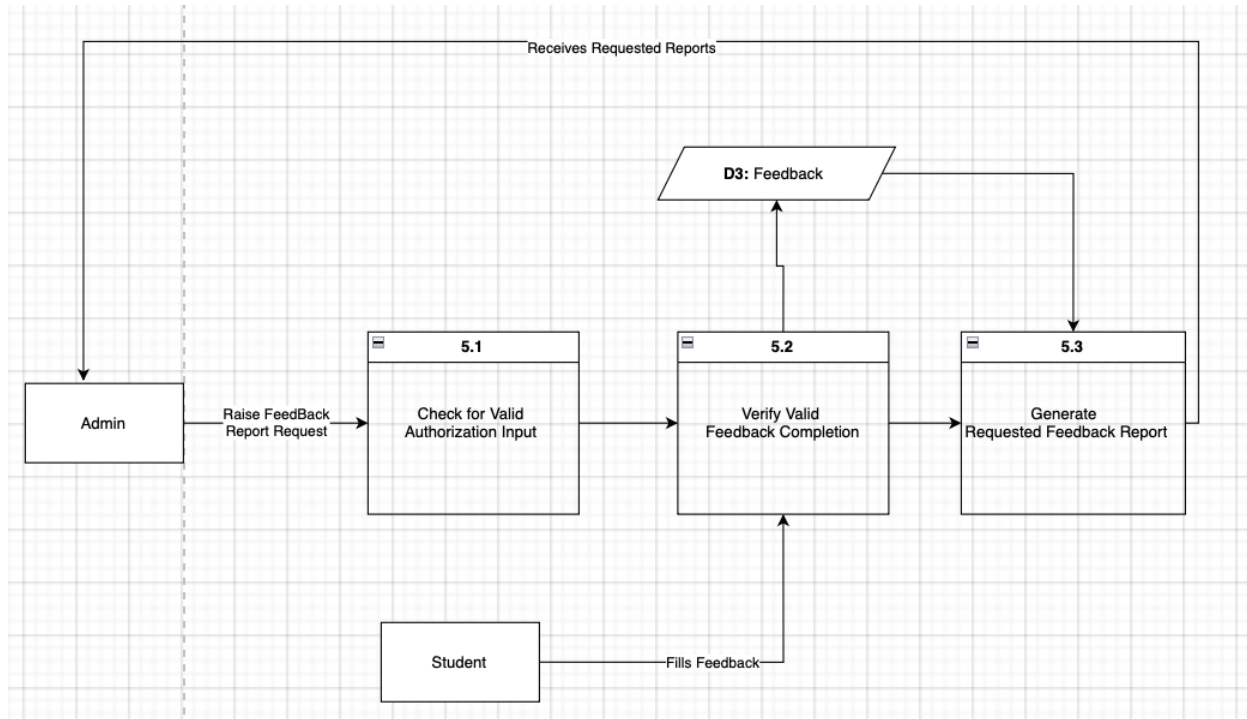
e. 5.0 Response Feedback

Fig 8.0 : Level 1 DFD : Response Feedback

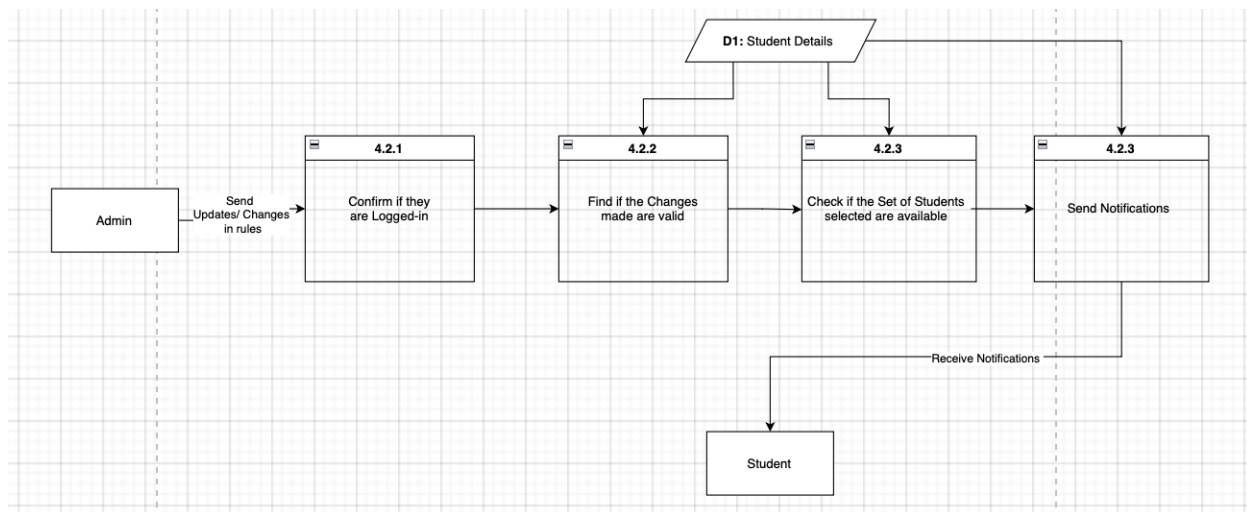## 4. Level 2 Diagram: 4.2 Send Notifications to all the Students

Fig 9.0 : Level 2 DFD : Send Notifications to all Students

Entity Relationship Diagram

Entity	Business Description
Student Details	Consists of Student Details (SUid, Name, Major, Address etc.)
Preferences	Contains Preferences Student would add before able to search for mates.
Admin	Contains the Admin Details (Admin ID, Name, Email ID, Current Address, etc.)
Notifications	Contains the Notifications sent by the Admin to the Students. Its details, time of release and description is stored.
Feedback	Contains the Feedback details shared by students.
Filters	Contains the Filter details added by the Admin.

Match Counter	Contains details of any potential match formed or requested.

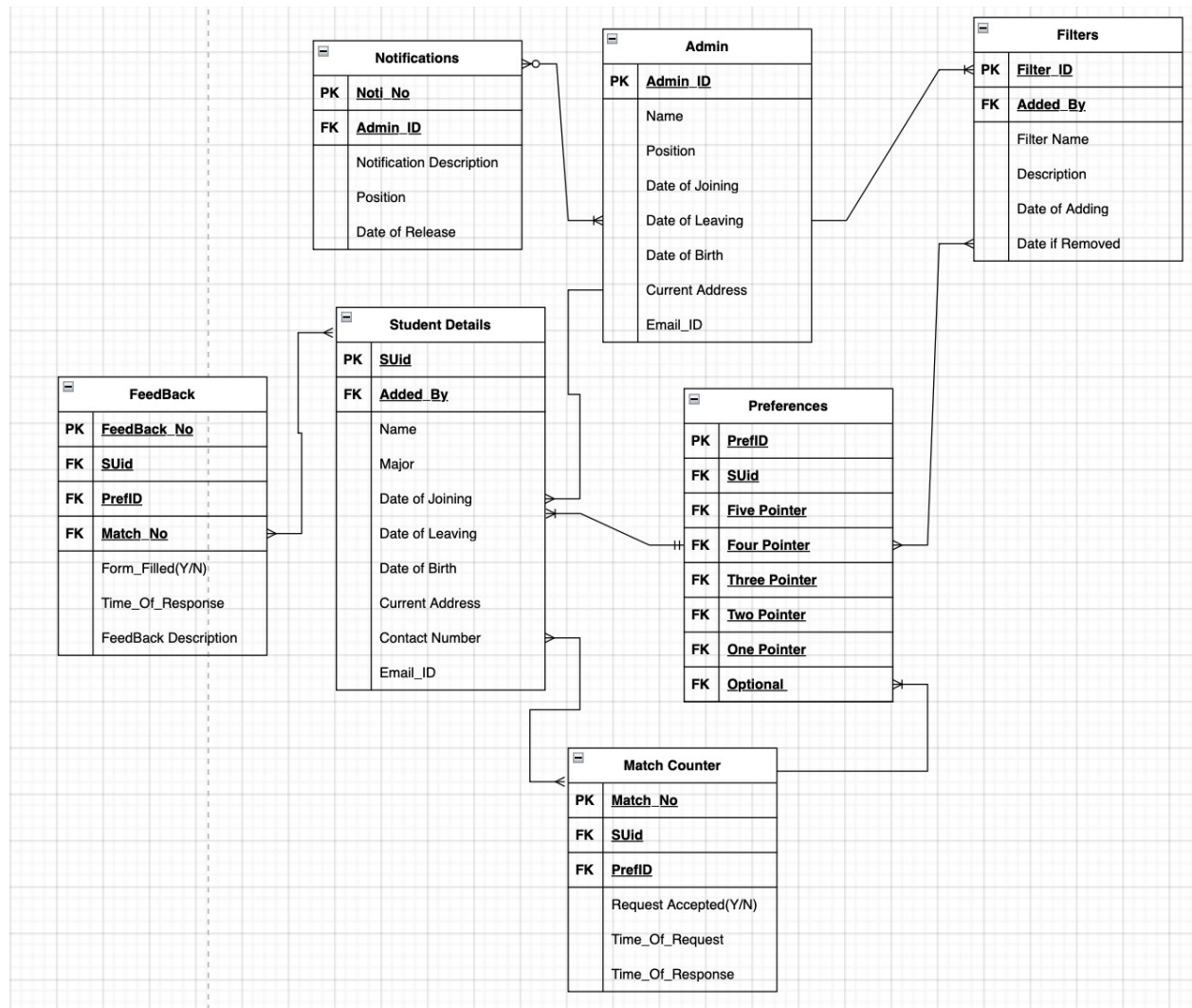Table 4.0 : ERD Description

Fig 10.0 : Entity Relationship Diagram
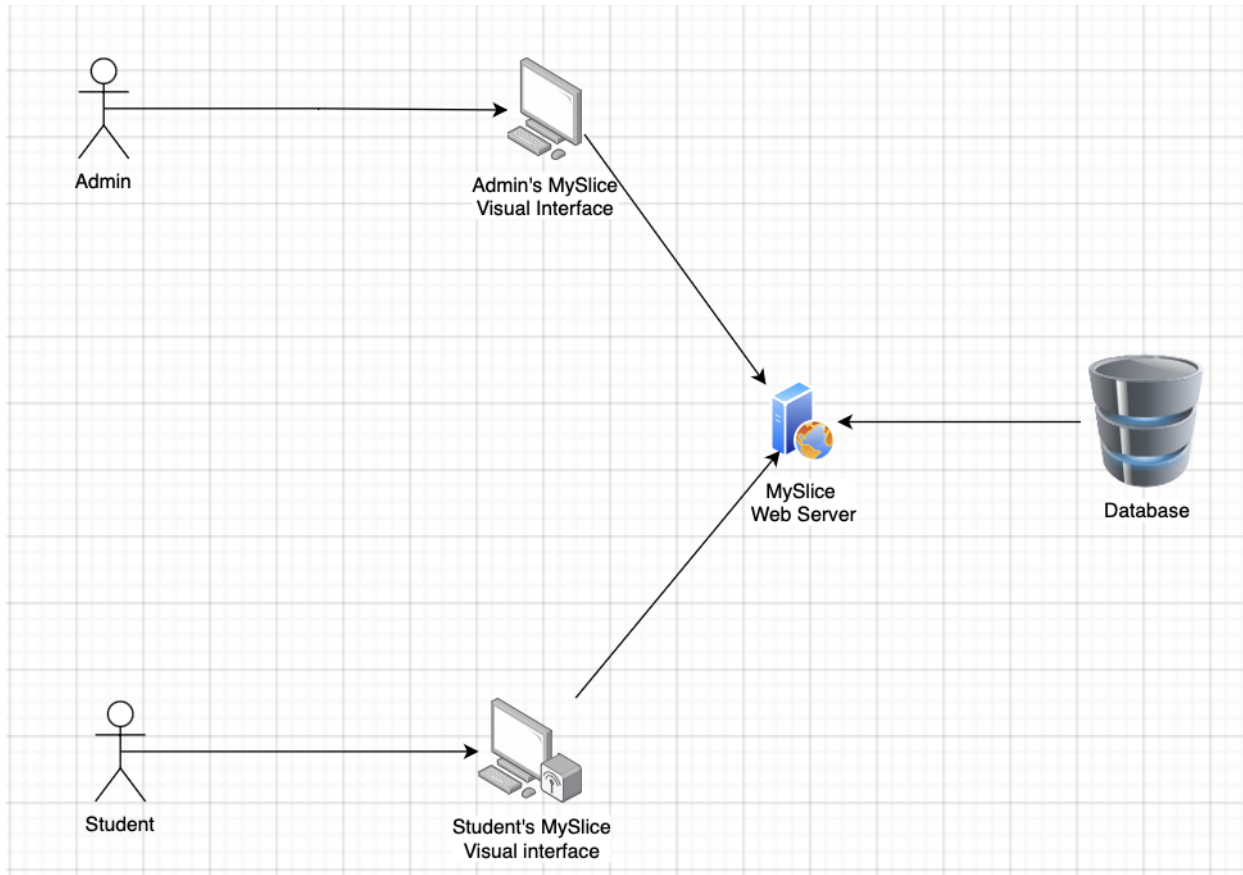
System Architecture

Fig 11.0 : System Architecture Diagram

The above System Architecture Diagram illustrates how we anticipate our system to function in general. We would utilize the MySlice platform, which would have its own security and safety architecture to safeguard data transactions. Administrators and students would each have their own systems, but share the same database on the server to prevent data loss. The System Architecture also validates the plan for Developers to work on the Student site independently from the Admin site, facilitating the division of labor.

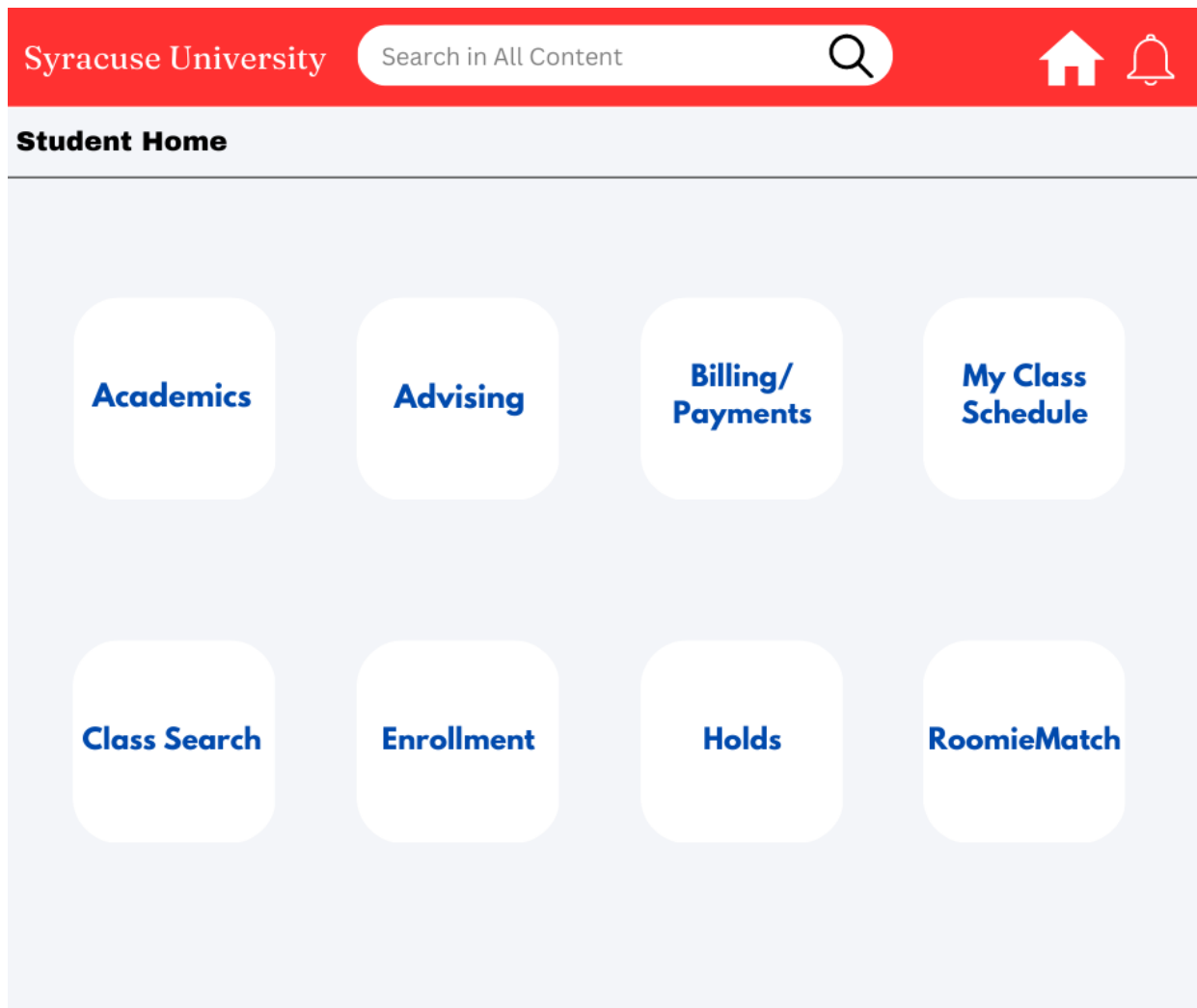# User Interface

Landing Page:

Fig 12.0 : Landing Page

- This interface is used by users authorized to use MySlice portal.
- The RoomieMatch extension can be accessed from this screen.
- When a user clicks on the RoomieMatch option, they are navigated to the search preferences interface for further steps.

Search Preferences Page:

Fig 13.0 : Search Preferences Page

- The user can search their desired outcome for roommates with the help of available filters.
- This can be done by entering the preferences relevant to them, for example location, distance in miles, food habits, cleanliness level, etc.
- Once the user has set the preferences according to their wish, they can hit enter and get the results.

Search Result Page:
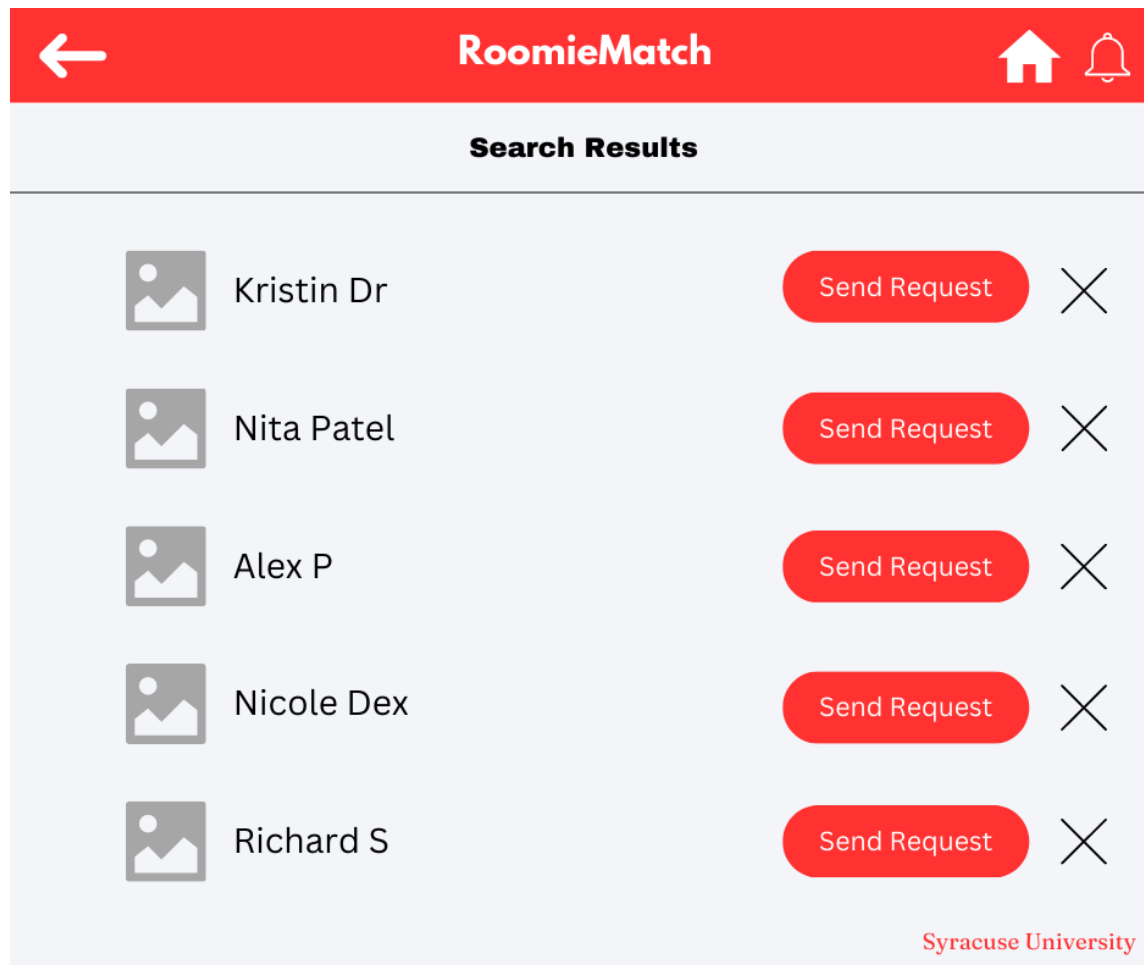
Fig 14.0 : Search Result Page

- This layout is obtained after the user has entered all the preferences and set the filters.
- The search result displays profiles of people who match with the filters and have successfully created a profile with Syracuse University email address.
- The user can go through the entire profile by clicking on it and get to know more about them.
- If the user is interested to connect, then they can click on the Send Request icon and wait for them to accept it or else just cancel by clicking on the X icon.
- The user can also navigate backwards to edit the search preferences to get better outcomes.

Match Page:
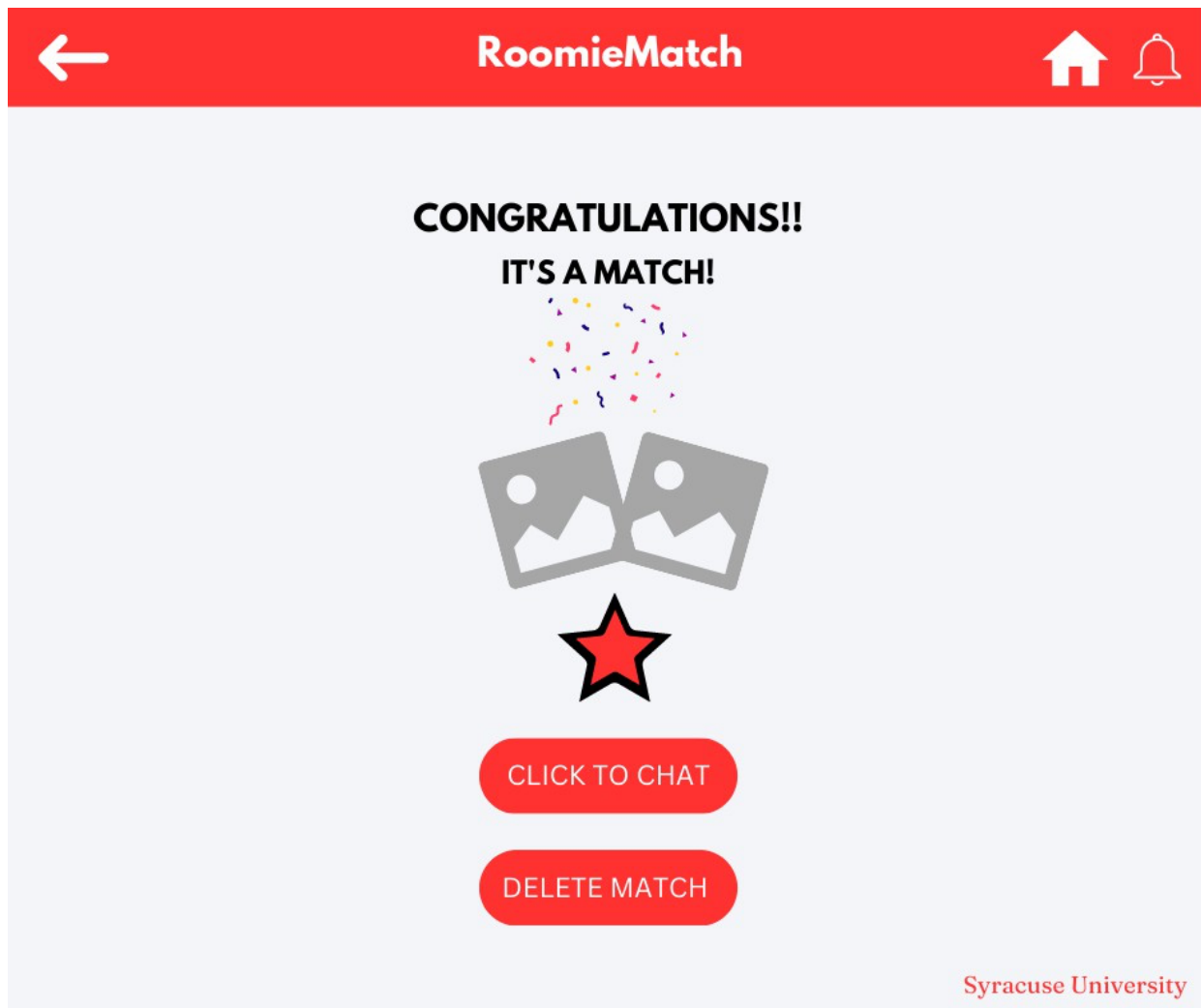
Fig 15.0 : Match Page

- After a user has sent a connection request to a potential roommate and he/she has successfully accepted it, this screen is displayed stating 'It's A Match'.
- The user can either start the conversation by clicking on the 'Click To Chat' option.
- The user also has the flexibility to change their mind later and delete the match if they think it is not the right fit for them.
- This layout has a home icon similar to all other above layouts, allowing users to reiterate to the home screen whenever desired.

Feasibility Analysis

Economic Feasibility

The economic feasibility of RoomieMatch depends on various factors such as the cost of development, maintenance, and promotion of the application, as well as the revenue generated from the platform. As the platform is an extension of the existing MySlice application, some of the costs of development and maintenance is already covered. However, additional costs such as server hosting and development of new features will still need to be taken into account. Another factor that may impact the economic feasibility of the platform is the level of demand for such a service among the university students. If there is significant demand for a roommate-finding service, the platform will attract a large user base, which may lead to increased revenue and profitability. If we create an effective revenue model and attract a significant user base, the platform has the potential to be economically feasible. The economic feasibility of RoomieMatch will depend on the following factors:

1. Development Cost: The cost of developing the extension will be a significant factor in its economic feasibility. This will include the cost of hiring developers, designers, and testers, as well as the cost of software and hardware needed to build and deploy it which is beared by Syracuse University in this case.

2. Marketing Cost: To attract a large user base, the extension will need to be marketed effectively. This will include the cost of advertising on social media platforms, search engines, and other channels.

3. Maintenance Cost: The cost of maintaining the extension will also be a crucial factor in its economic feasibility. This will include the cost of hosting the application on servers, bug fixes, security updates, and ongoing feature development.

4. Revenue Generation: The application will generate revenue by charging users a fee for using certain features, such as premium filters or advanced communication features in the future. It is currently free for the customers. Additionally, the platform may take a commission from external housing links provided on the platform.

5. Competition: The competition in the market will also impact the economic feasibility of the application. Other established platforms that offer similar services, will create challenges to attract users and generate revenue.

6. Scalability: If RoomieMatch proves successful at Syracuse University, it could be scaled up to other universities, potentially generating additional revenue streams and expanding the platform's user base.

Development Cost

Equipment Cost

The equipment cost is $0 as the complete cost of adding the RoomieMatch extension to MySlice portal is taken up by the university.

Personnel Cost

Personnel	Quantity	Duration	Cost	Total Cost
Developer	1	1 month	$ 100/hr	$ 16,000
System Admin	1	1 month	$ 70/hr	$ 11,200
Project Manager	1	1 month	$ 65/hr	$ 10,400
Data Analyst	1	1 month	$ 80/hr	$ 12,800
Executive Team	3	1 month	$ 50/hr	$ 24,000
Total				$ 74,400

Table 5.0 : Personnel Cost

Total Development Cost:

Name	Cost
Equipment	$ 0
Personnel	$ 74,400
Total Cost	$ 74,400

Table 6.0 : Total Development Cost

Operational Feasibility

The operational feasibility of RoomieMatch refers to the ease of implementing and using the platform within the existing systems and processes of Syracuse University.

1. A significant factor affecting the operational feasibility of the platform is the practicality of integration with MySlice. As the platform is an extension of existing MySlice, it should be designed to work seamlessly with it, without causing any disruptions, conflicts or problems.

2. The degree of technical expertise necessary to run and maintain the platform is another aspect that affects operational feasibility. With clear instructions and user-friendly interfaces, the application should be simple to use and navigate. Additionally, the developers must possess the technical knowledge necessary to address any arising problems and offer users timely assistance.

3. The security and privacy of user data must be taken into utmost consideration as personal details of users including their preferences and chats with their potential roommates is stored, so it is important for the platform to comply with all the laws and regulations and have proper data and security measures.

4. The RoomieMatch platform will need constant updates and enhancements to be relevant and meet the needs of the users.

5. The usage and adaptability of the platform will depend on the marketing and promotion campaigns deduced, it is mandatory to create and increase the user base.

6. If the RoomieMatch platform turns out to be a hit then it will result in a large user base which will need to be monitored and controlled.

Technical Feasibility

The technical feasibility of RoomieMatch refers to the ability of the platform to be developed, deployed, and maintained from a technological perspective within Syracuse University. Here are some aspects that are needed to be considered:

1. The technological stack considered to be used for RoomieMatch must be compatible with the existing MySlice portal. It will be beneficial for smooth integration and proper functioning of the extension.

2. The technical feasibility will extensively depend on the technical expertise required to develop and maintain the platform. The developers should have the relevant experience to provide prompt support and assistance to users in case of technical issues.

3. The platform needs to be tested comprehensively to ensure that it is free from technical errors and glitches before it is launched. End-to-end service validation of the product will result in a larger outreach and increase the likelihood of RoomieMatch becoming a household name.

4. For RoomieMatch to stay current and continue to satisfy the demands of its users, continual maintenance and support are necessary. The availability of a group of developers and technical support personnel is crucial for handling any potential problems.

Schedule Feasibility:

The schedule feasibility of RoomieMatch refers to the ability of the project to be completed within the desired timeline and budget.

1. The availability of resources, such as developers, designers, and project managers, is a crucial aspect influencing schedule feasibility. Any potential delays or bottlenecks should be identified and dealt with in a proactive manner, and the project team should have the resources to finish it in the required time frame.

2. The scope of the project is another element that influences schedule feasibility. To make sure that the project continues on track and within budget, it should have a well defined scope, along with definite objectives and deliverables. To guarantee that any scope

adjustments do not result in substantial delays or budget overruns, they should be thoroughly assessed.

3. The platform's complexity is yet another crucial factor in the feasibility of the program. To reduce the chance of delays and guarantee the project is completed on time, the platform should be created using a modular approach, with each module being developed and tested separately.

4. Last but not least, the degree of cooperation and communication between Syracuse University and the project team will determine if RoomieMatch can be completed on schedule. The platform should be developed in close collaboration with Syracuse University stakeholders to guarantee that it satisfies their needs and expectations and that any potential problems are found and fixed as early in the development process as possible.

Cultural Feasibility:

The cultural feasibility of RoomieMatch refers to the compatibility of the platform with the cultural values and practices of Syracuse University and its student community.

A significant factor affecting cultural feasibility is the diversity of the Syracuse University student community. The platform should be designed to accommodate the diverse cultural backgrounds, lifestyles, and preferences of the student community. The filters and preferences available on the platform should be comprehensive and inclusive, so that all students can use the platform effectively. This includes providing options for users to indicate their gender identity, sexual orientation, and cultural background, as well as ensuring that the matching algorithm is sensitive to these factors.

RoomieMatch should be created to take into account the diverse communication norms that may exist across cultures. For instance, certain cultures could place more value on formal communication as compared to informal style.

Implementation Plan

1. **Design**
   a. Designers will design the user interface, and user experience wireframes for RoomieMatch application. It'll also include user flows and user stories to depict the usage of the application. The design should match the existing mySlice application and reuse existing components wherever applicable. Post-creating, get the design reviewed and approved by all the stakeholders.
   b. Develop data schema for all the systems involved.
   c. Design system architecture
   d. Analyze the technology stack, including both front-end and back-end parts, to identify if any existing technology is unsuitable or insufficient for the RoomieMatch application.

2. **Development**
   a. The development plan will have three components with development in a 14 days sprint. A small team of 3 people is enough for the development, with two front-end developers and one back-end developer to complete the project within the deadlines.
   b. Components:
      i. Database development: This will involve commissioning the server on the existing cloud provider for mySlice, and creating up-down SQL scripts. It'll also include setting up alerts and monitoring for the server.
      ii. Front-end development: Involves implementing the front-end components and UI wireframes. Developers will also write unit tests for components developed. Peers will review the code written.
      iii. Back-end APIs development: Involves writing APIs and connecting to the database to store and retrieve the data. Also, developers will publish API schema for front-end and back-end integration.

3. Testing

a. Testers and developers, in collaboration, will develop a test plan to test all the functionalities of the RoomieMatch application with different testing focus areas. Testers will ensure the test cases cover all possible scenarios. Testers are also expected to prepare the test data to execute the test case.

b. Types of testing:

   i. Functional Testing: Testing of each feature to ensure all components work together seamlessly. Testers will also test error handling to ensure errors are gracefully handled. Testers will also conduct boundary testing to test how the features behave.

   ii. Performance Testing: Testers will conduct load testing to ensure the application handles parallel requests and high traffic.

   iii. Security Testing: The application will be tested by existing security frameworks for any possible vulnerabilities.

   iv. Regression Testing: Testing will be conducted on the existing mySlice application to ensure no impact on existing features of mySlice. All the defined mySlice test cases will be retested as well.

4. Deployment

a. Developers will deploy on the production servers of mySlice. A continuous integration and continuous deployment pipeline will also be created for future updates and fixes

5. Maintenance

a. Monitoring server performance and security will ensure the availability of non-functional requirements.

b. The team will push out regular updates and bug fixes to improve the application's functionality and user experience.

References

- *Systems Analysis and Design,* 12th Edition, Scott Tilley (2020), Cengage Learning.