

Roommate Finder Web Application

Arpita Shekhar
ashekha@ncsu.edu
Computer Science

Aishwarya Sundararajan
asundar2@ncsu.edu
Computer Science

Amulya Varote
avarote@ncsu.edu
Computer Science

Bhavik Patel
bcpatel@ncsu.edu
Computer Science

ABSTRACT

North Carolina University is one of the top research universities in the United States of America. It is known for its diversity. Every year, 2600+ international graduate students join the university. Along with acceptance comes the stress of looking for roommates who are compatible with most of their preferences. In most of the cases, students end up selecting a house, however the search for roommate is inefficient. In this work, we have surveyed and examined the problems of communication gap and trust issues faced by current graduate students despite having social media groups to bond with each other for finding potential roommates. We have developed a web application to address this issue using the technologies like NodeJS, MongoDB, HTML, CSS and Bootstrap. The application has been reviewed by almost twenty people and the results have been drafted. The feedback of the testing is analyzed and put forward in application's future scope.

Index Terms - Roommate Rapport Scale (RRS), NodeJS, Express, MongoDB, HTML, CSS and Bootstrap.

1. INTRODUCTION

The use of Software Engineering to solve problems is ubiquitous. This paper focuses on solved problem of finding compatible roommates for incoming non-resident NC State students. Our goal was to derive meaningful results from the survey conducted, browse through relevant journals, and develop a robust, user friendly, and scalable web application to solve the problem. Our survey strongly emphasized the need for a separate web application for roommate finding. Though there are several social media platforms such as Facebook or WhatsApp that allow people to create 'roommate finder' groups, posts accumulate one after another and reading through all the posts becomes almost impossible. Drawing conclusions from the survey results, we infer that poor roommate finding methods necessitate a user friendly, reliable web application that attests a user as a good roommate through the virtue of recommendations. The web application developed by us caters all the requirements to find a compatible roommate.

2. PREVIOUS WORK

Traditional roommate finder applications, such as Yocket, manually ask the users to enter too many preferences, some

of which may be unnecessary [5]. Though there are success stories of users finding their potential roommates through Yocket, many people aren't aware of the application. A customized application for NC State students would help overcome the disadvantages of traditional roommate finder applications. Another issue with such roommate finder applications is reliability. The user could be a fraudster and may not be someone he claims to be. This calls into question the reliability of the application and compromises the safety of other users interacting with the fraudster. There are several stories of regret and roommate misunderstandings through the use of such applications.

2.1 BACKGROUND AND MOTIVATION

There are many research papers which show how interpersonal compatibility, food preferences and other attributes contribute towards satisfaction of sharing the living space with someone who has a personality similar to theirs. A literature review of roommate understanding and preferences depicted that the RRS discriminated between those who selected their roommates and those who did not. If a student makes good friends with his roommates, it would have a positive impact on his psychological well-being, personal growth, purpose in life, and self-acceptance[1].

A student with a roommate conflict due to difficulty in communicating with the roommate or advocating for their needs may have a negative impact on their academics [3]. Studies suggest that happiness and depression may be highly contagious across social ties [4]. This means an incompatible roommate could be a potential cause for unhappiness and could be detrimental to the social circle. Most students who live with roommates for the first time often find it difficult to communicate and adjust with their roommates[6]. So, it is essential for the student to pair up with like-minded people as their roommates. [2] In a nationwide survey, 5.6% of students reported that roommate difficulties hindered their academic performance. Roommate conflict is a widespread experience among college students.

Through our literature survey of how to test software system we concluded that the testing of the application is one of the important steps of software development life cycle. There are many research papers which show the

importance of testing the developed applications in users' perspective. Pre-acceptance testing has been performed mainly known as alpha and beta testing to ensure the customers are able to perform intended functionality and feedback is taken to further enhance quality of software [9].

In order to further understand the issues related to roommate finding, we conducted a survey through Google Forms. The survey was sent out to 112 current graduate students from NCSU. Some questions were multiple choice and others were open-ended. The survey results are presented in the figure 1.

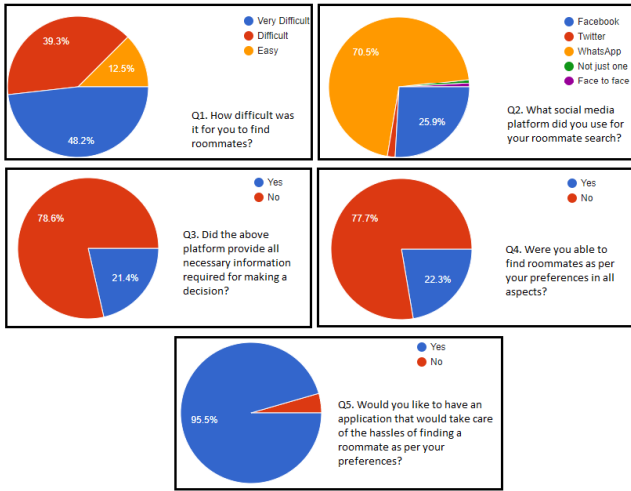


Figure 1: Initial Survey Results

The students were asked how difficult it was for them to find roommates. Out of all the responses, 48.2% of the people found it "very difficult" to find roommates and 39.3% of them indicated they found it "difficult" to find roommates.

The students were asked about the most commonly used application to search for roommates. WhatsApp was found to be the most common platform, used by 70.5% of students. In spite of having popular social media for roommate finding, the problem of finding compatible roommates persists.

The students were asked if the social media platform provided all the necessary information required to make the decision. Out of all the responses, 78.6% of the people thought that they did not have access to enough resources and only 21.4% of them were satisfied.

We can make out that the posts by the people who are looking for roommates on social media website are wordy and descriptive. The user may not go through each and every detail. The audience was asked if they found roommate as per their preferences in all aspects using the social media platform. Out of the recorded responses, 22.3% of the people found roommates as per their preferences and 77.7% of them did not find roommates as per their preferences in all aspects.

In the end, surveyed students were asked if they would like to have an application that would take care of the hassles of finding a roommate as per their preferences. 95.5% of the people said that they would like to have such an application.

We developed a web application which addressed all the issues identified by the above survey. The web application developed has information of all the enrolled university students who are looking for roommates. The application provides a friendly interface for people to input their characteristics and connect with compatible roommates. This application also takes all the required preferences from the other users in the form of tags and recommends compatible roommates to the user based on their preferences. The web application focuses on only the roommates finding feature unlike the social media sites like Facebook and WhatsApp. This prevents the wastage of time to search for the roommate finding related posts on social media sites.

3. PROBLEM STATEMENT

People from different countries face difficulties to find compatible roommates while attending graduate school. People will have issues like budget, food preferences and personal habits. Many people look for off-campus housing because it is cheaper compared to on-campus housing. A software application can help many students to find roommates of their choice. The development of a web application to help students find the people who have similar interests.

4. SOFTWARE REQUIREMENT SPECIFICATION

A software requirements specification describes the web application to be developed. It has functional and non-functional requirements of the Roommate Finder application, including a set of use cases that describe user interactions that the web application must provide.

4.1 FUNCTIONAL REQUIREMENTS

1. Signup and Email Verification:
The user shall sign up to the application by entering the email id and password. The user shall verify his email address by clicking the link which has been sent to the email id entered during the signup.
2. Login and Authentication:
The user shall login to the system using email id and password provided at the time of sign up.
3. Registration:
After successful sign up, upon first login, user shall register to use the application by entering the basic information about the himself. These characteristics are used to match different users with each other.
4. Update User Profile:
Upon successful login, user shall access his profile by clicking on "Your profile" on the navigation bar and

shall see the basic information entered during the registration. The user shall update his profile at any time using these options.

5. Update password:
Upon successful login, user shall access the Settings tab to update his password. To do so, he may enter the old password followed by the new password and confirmation.
6. Search:
The user shall search people by using the tags provided on the search page.
7. View and Connect with Users:
The user shall see the basic information by clicking the image of a person searched. The user shall send an email to the searched user by clicking the connect button.
8. Logout:
On clicking the Logout option, the user can successfully log-out of his session.

4.2 NON-FUNCTIONAL REQUIREMENTS

1. Security:
The security feature has been implemented by sending email to the user to verify the email id. The password of the user is encrypted using "Bcrypt" and is then stored in the database.
2. Usability:
The application provides better User Interface to the user to search the roommates based on his preferences by using the tagging system.
3. Scalability:
The application has been developed by using MongoDB which is very scalable. On an average, at any point in time, there are only ten thousands of students looking for apartments near NCSU. Our application can easily support the same. Additionally, the application uses AWS Sandbox Database-as-a-Service, whose Database connection is "mongo ds155268.mlab.com:55268/users-dev -u asundar2 -p Qwerty123"
4. Modifiability:
The application is designed in such a way that adding additional functionalities and editing the existing ones is easy and less time consuming.

5. Availability:
The application has been deployed on Heroku. The web application is available every time the system is running on heroku supported by the AWS Sandbox Database server.

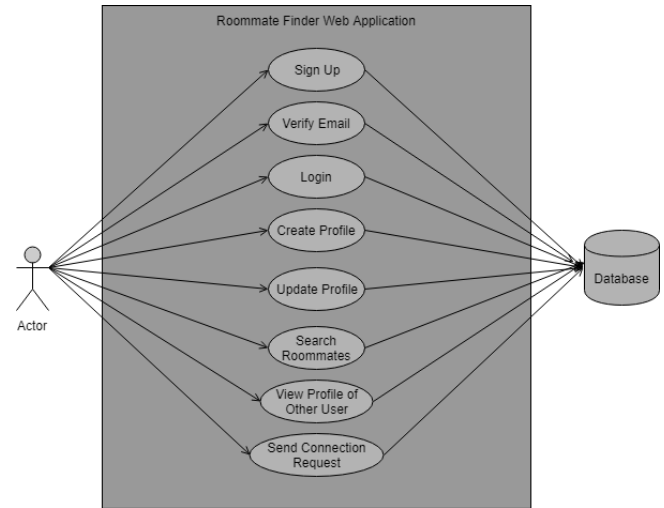


Figure 2: Use case diagram of the application

4.3 USE CASE DESCRIPTION

The use case diagram is the description of the set of functionality that the user can perform on the system. The Figure 2 shows that the user interacts with the web application. The set of functions that the user performs on the system are signing up, verification of the email, logging in with the credentials, viewing the basic information and characteristics, updating password and characteristics, searching people based on preference using tagging system, viewing the profile of the user whom the user is interested in and sending an email to connect with the searched user.

4.4 ARCHITECTURE OF THE APPLICATION

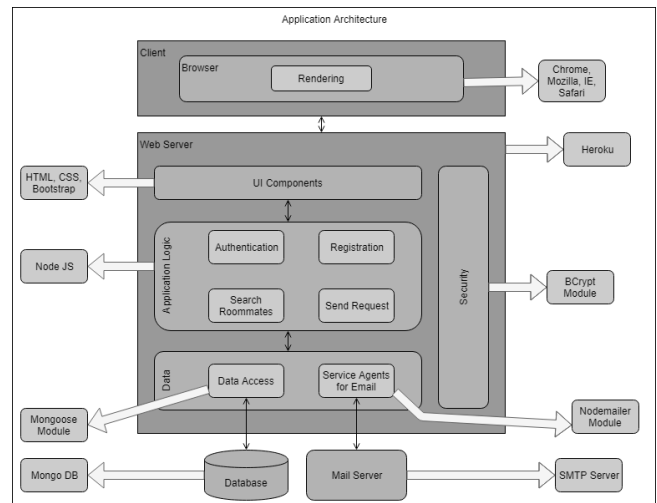


Figure 3: Use case diagram of the application

Figure 3 shows the detailed architecture of the application. The development of the application focuses on the user experience.

The client browses the application on Chrome, Mozilla, IE or Safari which is hosted on Heroku. The web server has UI Components, Application Logic, Security and Data unit. The application logic which includes authentication, registration, searching and sending the request (basically email) and has been implemented using NodeJS. The UI components have been implemented in HTML, CSS, and Bootstrap. The security feature has been included using BCrypt module. The module which is used to access the stored data available on external server is Data Access unit. The database used for the application is MongoDB at AWS Sandbox Database-as-a-Service. The link to the external server has been hardcoded in the application. The mail server is the SMTP server. The service agent Nodemailer module has been used for the mailing services.

5. APPLICATION FLOW AND FEATURES

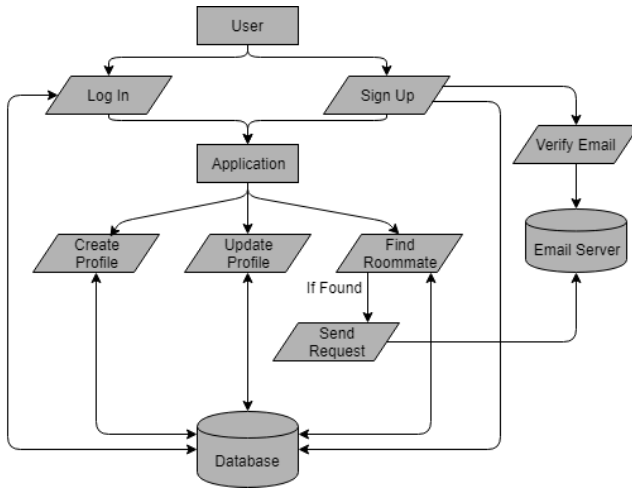


Figure 4: Use case diagram of the application

The Roommate Finder web application flow is shown in Figure 4. The user signs up with the system by creating a profile using NC State ID. Once the registered email ID is verified, the user can successfully login to the system. After successful login, the user can find options to 'Edit profile', 'Search roommates' and 'Update Preferences'. Clicking on the 'Edit profile' button enables the user to change his preferences. By default, the home page of the application shows the search button with Carousel. The user clicks on the search button on the carousel page to search for the roommates. Clicking on any of the displayed user profiles will navigate to the corresponding user's information. The 'Update Preferences' link will allow the users to edit his preferences.

The application provides the user a nice and intuitive user interface. The user should signup using the NC State University email id. The verification mail will be sent to the user's NC State gmail account. The user should verify it logging in to the email account. The link takes to the login page of the application. The user should login with the credentials to use the application. The user has to set up his profile after logging in to the application. The user should enter all the basic details and the characteristics.

The application has the landing page which has the search button to search the roommates. The user can search the roommates by the preferences using the tagging system. The information of the matched users will be displayed on the search page. The user can click on any of the profiles to see the detailed information. If the user finds a match, he/she can click on the "Connect" button and the email having the contact details can be sent to the searched user. If the searched user is interested, then he/she will contact to become roommates.

6. IMPLEMENTATION

The backend technology used to develop the web application is NodeJS. NodeJS is an open-source, cross platform JavaScript runtime environment. JavaScript was used only for client side scripting, in which the JavaScript code is embedded in the html code. Node.js enables JavaScript to be used for server side. The code is run on the server side to produce the dynamic output. Now a days, NodeJS has become one of the most important technologies to build a web application allowing web application development to unify around a single programming language, rather than relying on a different language for writing server side scripts.

The database used for the development is MongoDB. MongoDB is free and open source platform for document-oriented database development. It is classified as NoSQL database. MongoDB uses JSON-like documents with schemas.

For the front-end design, HTML, CSS, JavaScript and Bootstrap CDN has been used.

The web application opens up a login page as shown in the Figure 5. The user has to signup by clicking the Sign Up tab. The Sign Up page opens for the user. The user has to enter the user id and password. The password is verified against the confirm password entered by the user. The email ids are restricted to the NC State domain. The domain name "@ncsu.edu" is pre-populated. Hence the user has to enter only the unity id. The Sign Up is as shown in the Figure 6.

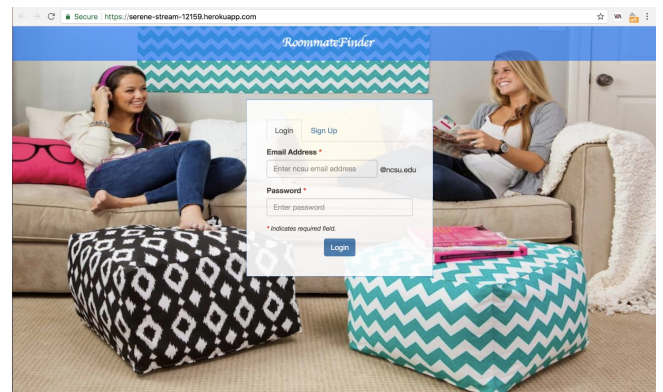


Figure 5: Application Login

The email will be sent to the users NC State Gmail

account to verify the email id.

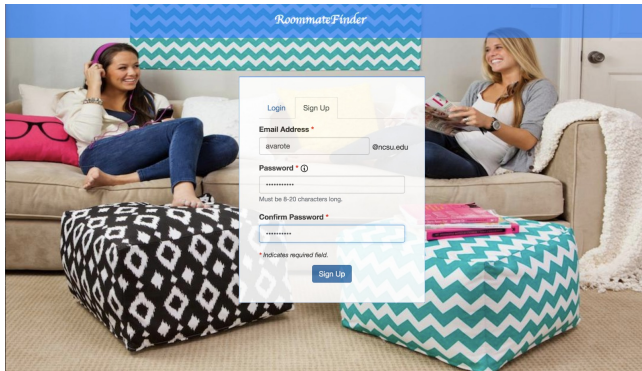


Figure 6: Application Sign Up

The Figure 7 shows that the user has received the link to verify the email id. The user has to click on the link. And the login page of the user opens.

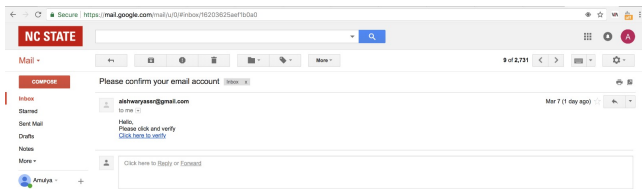


Figure 7: Verification email sent to the NCSU account

The user has to login using his credentials. The username and the password get validated and the user is authenticated. The registration page opens as shown in the Figure 8. The user has to enter all his basic details like First name, last name, Gender, Email address and Phone number.

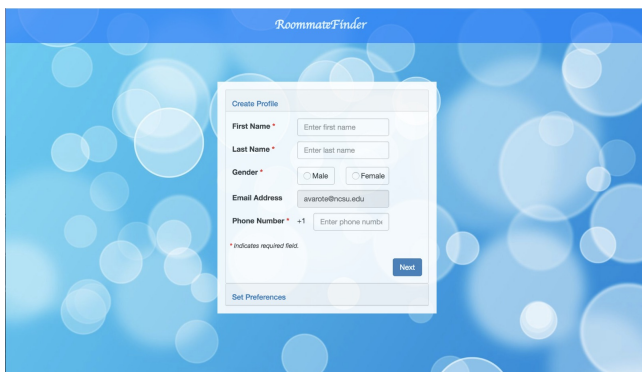


Figure 8: Registration page

The user has to enter his characteristics like budget, location, Dietary habits, Smoking habits, alcoholic habits, room sharing earliest move in date and latest move in date. All the fields are made mandatory. And all the text fields are validated. After setting up the profile, the user is taken

to "My Profile" page.

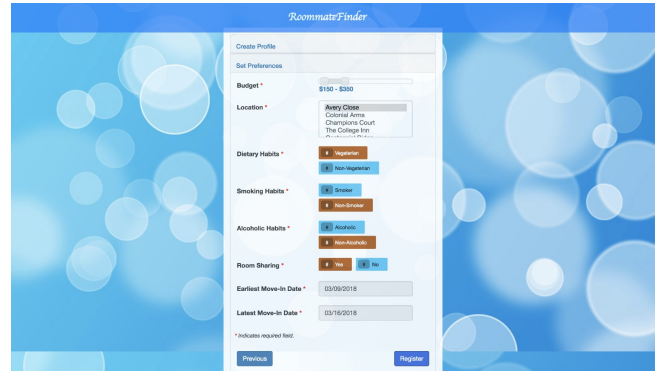


Figure 9: Setting up 'My Profile' page to enter the characteristics

The user can use the services. The user can revisit the website by logging in to the application. After logging in, the landing page opens. The landing page has been implemented as Carousel page with search button on it. The landing page is shown in the Figure 10.

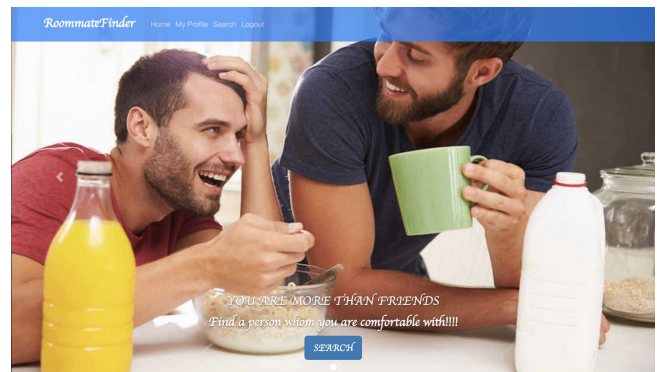


Figure 10: Landing page of the application

The user can check his profile details by clicking on the tab "My Profile", which is on the navigation bar. The profile page is shown in the Figure 11.

The user can view the characteristics which are entered during the registration. All the fields filled by the user are displayed. The view characteristics page is shown in Figure 12.

The user can update the characteristics by clicking on UPDATE option which is on the left panel. The update page is shown in the Figure 13. The row of the user in the database gets updated.

The settings option provides the user a functionality to change the password. The SETTINGS page is shown in the Figure 14. Again the passwords are validated accordingly and get updated in the database for that particular user.

The user can search for the people based on the preferences. The search functionality is based on tagging system. The user inputs the data by using the tags provided in the search page. The tags are provided for the better user

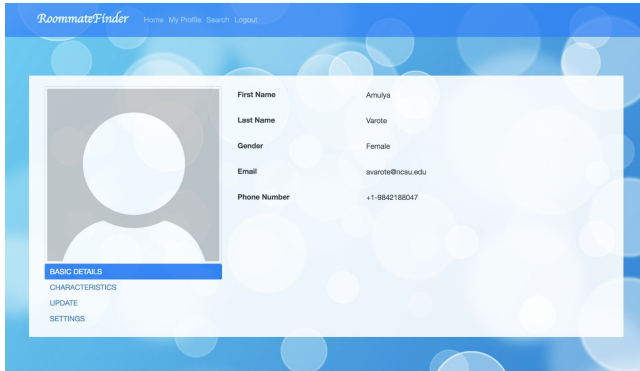


Figure 11: User profile page with basic details

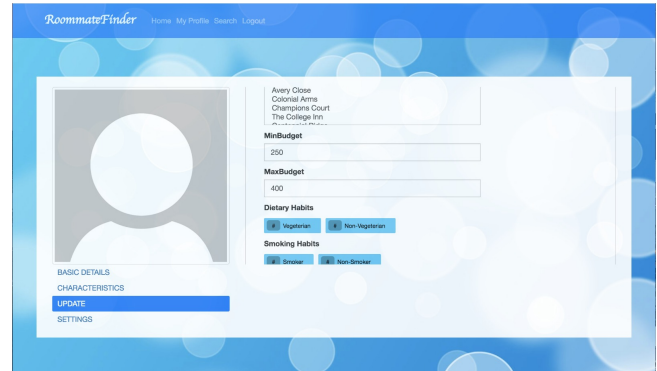


Figure 13: Updating the characteristics

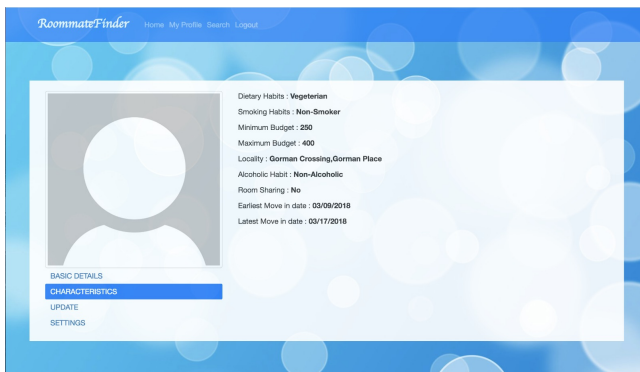


Figure 12: User profile page with characteristics

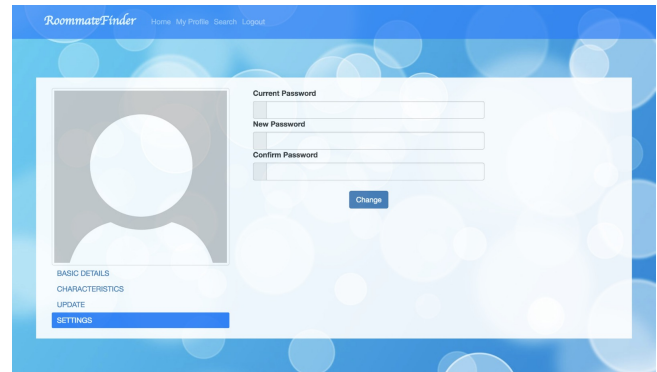


Figure 14: Updating the password

experience. The search page is shown in the Figure 15.

The previously searched parameters are also displayed on the page for the user to refer. The user can click on the button provided on the user card which is displayed on the page. The user profile opens with all the details. The user profile is shown in the Figure 16.

If the user is fine with the characteristics of the viewed person, then the user can connect with him by clicking on the connect button which is displayed. The email including the contact details will be sent to the person. The email template is shown in the Figure 16.

The intended person can contact the user if interested.
Search Functionality:

Personalized search has been extensively researched in the field of information retrieval and modeling user preferences. In our application, we have implemented tag clustering as a method to better describe the personalized search criteria. Users can search by location and is matched to a list of compatible users looking for a roommate in the vicinity. We have used tagging to identify potential matches rather than debating whether opposites attract or birds of a feather flock together.

7. EVALUATION

The methods which will be followed to evaluate our

proposed software are system Population with known data and beta testing. Both method targets level of satisfaction of the user as a metric for evaluation.

1. System Population with known data

In this evaluation method, we have studied some of the social media that are currently used to find roommates. From the result of this study, we extracted the user profiles who agreed to be roommates of each other. We fed these user profiles to our system and checked whether our system recommends the same roommates to users or not. By doing this analysis, we made concrete conclusions on our system's accuracy of recommendations. Even if our system did not produce desired results, using this analysis we made deductions regarding flaws of our system and we found mistakes in our algorithm or system design.

2. Beta Testing

We released our web application to some of the real-world users. We asked them to create their user profiles and feed their preferences of roommates. After recording their preferences, we created a dummy profile, which exactly matches the preferences set by a user. After creating this dummy profile, we checked whether this profile is recommended to the user whose preferences we recorded. We asked user to give feedback on ease of use and user experience aspects of our system. From this evaluation, we concluded accuracy and user satisfaction score of our



Figure 15: Search page of the application

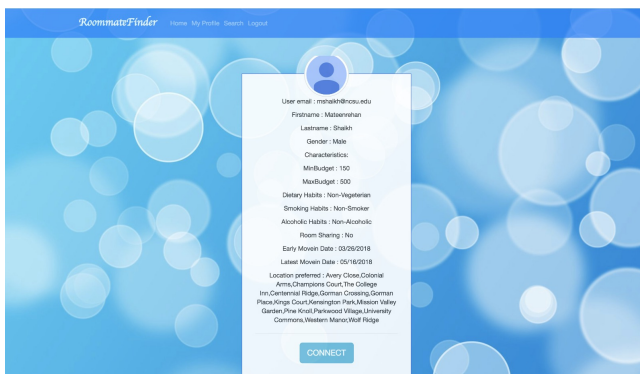


Figure 16: Profile of the searched user

system. From user feedback we also learned about the shortcomings of our system. This method is also useful in assessing the overall user experience of our system.

7.1 EVALUATION RESULTS

We evaluated our software with the help of 20 students manually experimenting and testing the web application. As the development of the application focuses on the user experience, the tagging system has been implemented to make the search functionality as simple as possible and interaction between the user and the system is kept as minimum as possible. The application flow is such that the user is not required to go through readme or help file to use the application. We also requested the evaluators to provide valuable inputs. The evaluators were also asked to think aloud about the bugs in the application.

We anticipated that the users will not face any issues while using the application. So we decided to deploy the application on Heroku. The users were provided with the Google form to provide their feedback. The Google form had the application link for the users to use and test our application.

A. Coolness of the idea:

The coolness factor of the application is the User Interface and the tagging system. From the literature survey, we learned that the people are looking for the system which is easy to use without any complicated flows. 60 percent of the people said that the idea of the project is very cool and

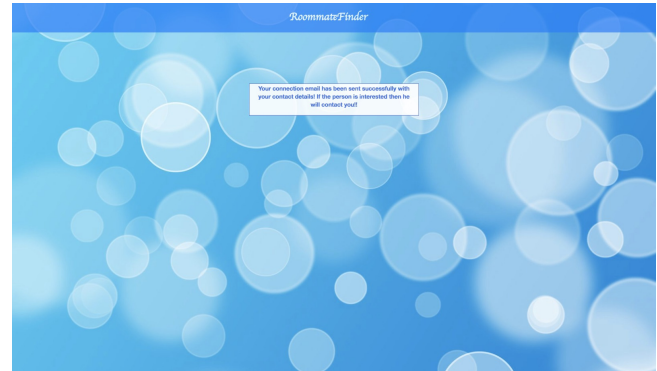


Figure 17: Connect email is sent

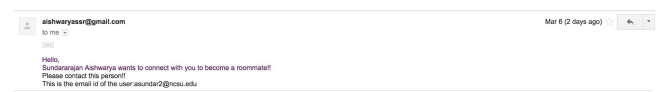


Figure 18: Email template of the connect request

remaining people's feedback was it is cool. Overall, all the evaluators liked the application idea. We can conclude that the project idea is cool based on the results shown in the Figure 19.

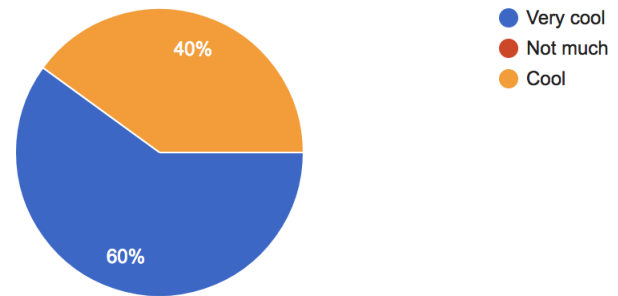


Figure 19: How cool is the idea to you

B. Ease of usage of the application:

As the main part of the project is to provide the users with the best user experience, we asked people to know if we have developed the application meeting all of their expectations. We can see that most of the people said that they could use the application without going through the readme file. From the Figure 20, we can come to the conclusion that the application has provided best user experience to the users.

C. Overall rating of the application:

We asked evaluators to rate the application overall. From the initial literature survey, we analyzed that the application should be a blend of good functionalities and user experience. We tried our level best to include both of them in the project. From the Figure 21, we can conclude that the we successfully delivered what was expected by the users. Most the evaluators rated the maximum that is 4 and 5.

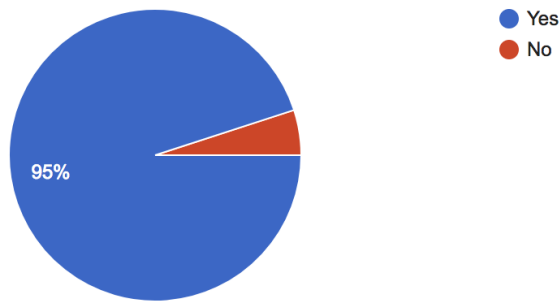


Figure 20: Are you able to navigate to the pages without reading the readme file

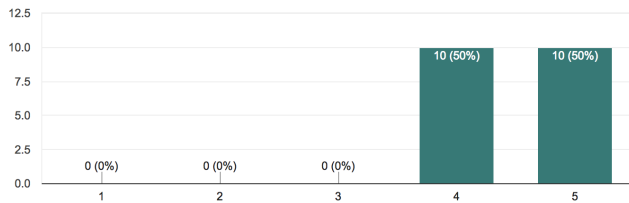


Figure 21: How do you rate the User Interface of the application

D. Inclusion of the features in the application:

We wanted the feedback to see if we have considered all the features in the search functionality. We asked evaluators whether the basic categories of preferences have been included in the search functionality. We found that the 90 percent of the evaluators were able to search for the roommates based on the preferences included in the application. We received the feedback to include some other features from 10 percent of the evaluators which is shown in the Figure 22.

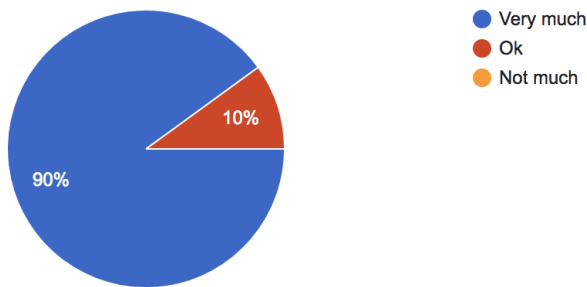


Figure 22: Did you feel that basic categories of preferences have been included in the search functionality

E. Rating of the tagging system:

We asked the evaluators to use mainly the search functionality. We asked the evaluators to rate the tagging system of the application. From the Figure 23, We found that most of the evaluators liked the system and they have given the maximum points to the feature which was focused much.

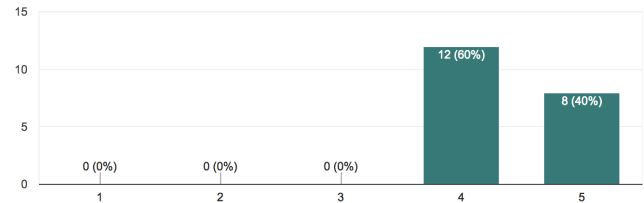


Figure 23: Rate the tagging system on scale of 5

F. Major Expectations of the project:

From the literature, we came to know that there are applications to find the roommates. But these applications do not fulfill the requirements of the user. Hence, the evaluators were asked if the major expectations of the project have been fulfilled. We found that majority of the people felt that major features have been included. The results have been shown in the Figure 24.

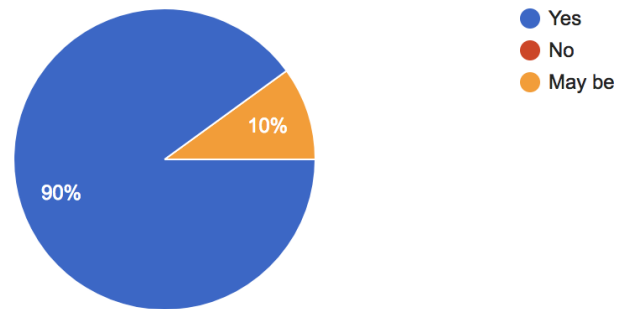


Figure 24: Did the application fulfill your major expectations

G. Finding roommates using the web application:

As we have included the tagging system, we asked evaluators if they found it easy to find roommates. We understood that 95 percent of the people found roommates easily. The results have been shown in the Figure 25.

H. Usefulness of the system:

We understood that most of the international students find it very difficult to find roommates from the literature survey. Hence, we asked the application evaluators regarding the usefulness of the system. From the Figure 26, we can observe that 80 percent of the people said that they find the application very useful to find the roommates.

I. Overall rating of the application:

The evaluators were asked to overall rate the application based on the User Interface, User Experience and func-

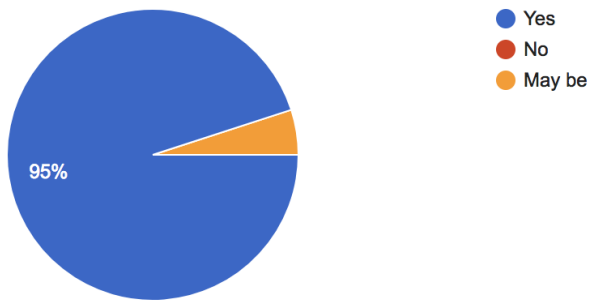


Figure 25: Did you find it easy to search roommates

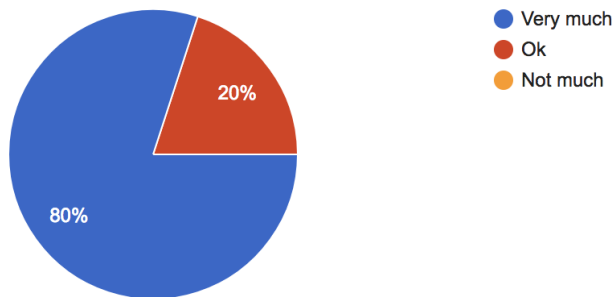


Figure 26: How useful do you think this system will be to find roommates

tionalities implemented. Most of the evaluators rated the application 4 out of 5. The average rating of the application based on all the features is 4.2 out of 5. The ratings are shown in the Figure 27.

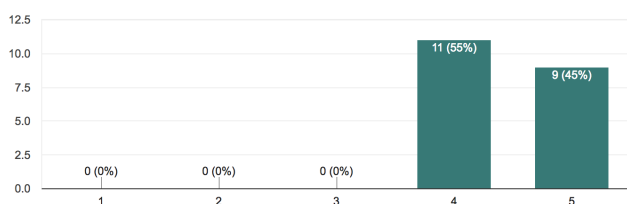


Figure 27: How do you rate our application overall

J. Usage if the application in future:

The evaluators were asked if they will be using the application in future. We found that 90 percent of them said that they want to use this web application in future.

We asked the evaluators regarding the strong points of the application. The evaluators said that they liked the application overall and they liked the User Interface much. Some of the answers of the evaluators are, "Tagging system is innovative. Also UI is aesthetic to the application theme", "The website is very well designed, with good interactive features. The application has potential to be very useful for people everywhere while moving into a new

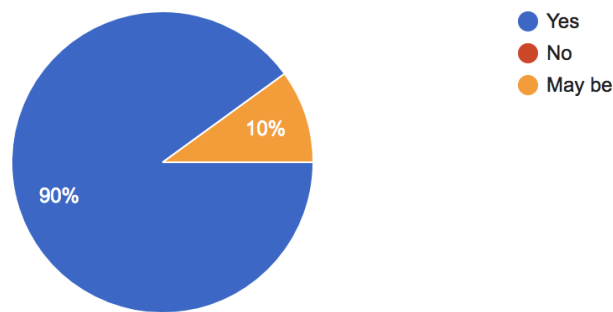


Figure 28: Will you use this web application in future

city" and "Very elegant and simple to navigate".

We also asked the evaluators, how the application can be improved. The best recommendations we got from the evaluators are, "Polishing edge scenarios" and "One important enhancement can be adding review system to the application. People can provide reviews for other people with whom they have stayed in the past so that it helps the future prospects to understand the person more clearly. Let us say, a person's nature can not be scaled by themselves and the review might help in those cases".

We got the best feedback for the web application developed by us. Most of the expectations have been met. Overall, the users enjoyed using the application.

8. TIMELINE

Timeline	Task	Description
2/2/2018	Report 1 Submission	Gathering requirements, surveying and determining the feasibility.
2/8/2018	Database Design	Plan and Design the database.
2/15/2018	User Interface Design	Plan and Decide the User Interface.
2/28/2018	Working Prototype	Develop a working prototype of the system to find the roommates
3/05/2018	Acceptance testing/Evaluation	Draft the testing report.
3/10/2018	Documentation and Readme File	Report of the project including the readme file on GitHub
3/12/2018	Report 2: Submission	Project report consisting of implementation and evaluation details
3/13/2018	5-minute presentation	Give a presentation on cool factor of the project

9. CHALLENGES

The challenges faced during the development of the application are as follows.

1. Learning a new technology:

Everybody in the team was new to Node JS and MongoDB. Learning about the new technology in a short span of time was challenging. It took almost half of the allocated time to learn about Node JS and MongoDB.

2. Designing User Interface:

The main focus of the application is to provide good user experience. Implementing the user interface to take the minimum inputs from the user was challenging. Designing each and every page to provide aesthetic look also was a challenging task. The minute details of the UI has been taken care while designing UI of the application.

3. Following Standard Practices:

The team was completely new to the technology. It was challenging to follow the good practices of coding. As team was unaware about industry standard of project structure and coding practices, we had to do lots of rework and maintenance tasks.

10. CONCLUSION

Our method of finding the most compatible roommate is very elementary and takes advantage of the fact that a person's liking might be different from their preference. For example, a person who is vegan/vegetarian can be open to having a non-vegetarian person as a partner. Widespread use of social media can be used to mine relevant data-sets for our application. However, it would be challenging to conclude very specific information regarding eating, smoking preferences for all the users. It would be relevant to take feedback from the users after a year to better understand if our perception of better roommate matching is persistent with actual experience of our target audience. We can then revisit and improve the metrics used to define compatibility.

11. FUTURE SCOPE

Integration with Social Media - According to one of our survey questions, it was evident that the most common platforms for roommate search is WhatsApp and Facebook. It would be accurate or insightful to gather data regarding user preferences from an individual's social media interaction. Tracking user's most contacted friends would help in predicting matches who might be friends with a common friend. It would enhance the competency of our application along with providing a better user experience. One of the other compelling reason to integrate with social media is ease of access. Almost everyone has an account on Facebook and twitter. It would be convenient for the users to Login using Facebook or Google + credentials, without having to sign up explicitly. It would also boost our applications popularity.

Predicting frequently occurring combinations of

preferences - Once the application becomes successful, we can gather data on combinations of preferences that occur together frequently. Using FP-tree growth algorithm, it would be possible to successfully come up with several interesting frequent patterns that can help re-design our database for faster and efficient access. For example, if we observe the trend that most of the people who have vegetarian as food preference also look for a non-alcoholic person as a roommate, then the tags <non-alcoholic> and <vegetarian> can be displayed as suggestions to the users to speed up the process of inputting tags.

Sending the request through application- Once the user finds the perfect match, request can be sent to the matched profile on email in current implementation. In future we can add social media type connect feature in application. The user who is searching roommates can send connection request to potential roommate match. The matched profile user can accept the request if he/she is fine with the preferences of the user. After matching the user, the state of users can be changed to Not Available, so that those users are not displayed to other users who are searching for users.

Integration with Chat- In order to make it easier for potential roommates to communicate with each other through our web application, a chat feature could be integrated. This avoids the need to use a third party application such as an email application. After users connect with each other on the web app, they can interact with each other using the chat box provided by our web application.

Combined Roommate Search- After two people have agreed to be roommates of each other, they both can search other roommates as combined user. In such cases, preferences of both the users can be taken into consideration to find the potential match. Additionally, group chat feature could also be introduced for a user to communicate multiple people who may be planning to stay together.

Review and Recommend Roommates- A user can review and recommend his roommate on web application. This review can be anonymous or identified. Such reviews can be useful for other users to make better informed decisions.

12. REFERENCES

- [1] Erb, S.E., Renshaw, K.D., Short, J. L., Pollard, J. W. (2014), The importance of college roommate relationships: A review and systemic conceptualization, Journal of Student Affairs Research and Practice.
- [2] M. Shekhawat, S. Deshmukh, G. Monroy, A. Tiwari, X. He, H. Shin, Y. Hong H. Lu, Usability Test of Personality Type within a Roommate Matching. Journal of International Technology and Information Management, Volume 25 | Issue 1
- [3] Ashley M. Payne, The challenges of living with a roommate: the impact on students with disabilities' residential experience, Rowan University
- [4] Ezra Golberstein, Janis L. Whitlock, and Marilyn F. Downs, Social contagion of mental health: Evidence

from college roommates, Health Econ. 2013 Aug; 22(8): 965-986

- [5] <https://yocket.in>
- [6] Kimberly Halpin, Roommate Rants: Understanding Roommate Conflicts among MSU Students, Journal of Undergraduate Research at Minnesota State University, Mankato. Volume 9 | Article 3
- [7] <https://github.com/googlemaps/google-maps-services-js>
- [8] <https://www.mongodb.com/blog/post/the-modern-application-stack-part-2-using-mongodb-with-nodejs>
- [9] Itti Hooda, Rajendra Singh Chhillar, "Software Test Process, Testing Types and Techniques", International Journal of Computer Applications (0975 - 8887) Volume 111 - No 13, February 2015.

13. EVALUATORS TOKEN NUMBERS

- 1. DVS
- 2. JMC
- 3. VCI
- 4. UOJ
- 5. EKY
- 6. XQJ
- 7. CTD
- 8. WGE
- 9. FYI
- 10. WTE
- 11. KSP
- 12. BIQ
- 13. XZY
- 14. DBC
- 15. GBF
- 16. ANE
- 17. BRB
- 18. UDL
- 19. YXL
- 20. JDK