Room Mate

Cameron Davis, Jaden Feldman, Grace Harrell, Mei Hecht and Brendan Lancaster

Project Description:

Room Mate is a dating app style application in which students of participating colleges will be able to find and match with potential roommates. Room Mate aims to ease the process for the average college student to find a compatible roommate. Using user-entered data, we will be able to match students to compatible roommates based on personality metrics. Each student will enter a housing id that indicates where they would like to live, what year they are graduating, the minimum rent amount they are willing to pay as well as the maximum and a brief summary about themselves.Once the students are matched, they will be provided with each other's contact information to talk further, and provide to their school which will complete the assignment of the dorm rooms.

Project Tracker: Link to Project Board Github

https://github.com/users/grace-harrell/projects/2/views/1

Video: 5 minute or less video demonstrating your project. Your audience is a potential customer or person interested in using your product.

VCS: Link to your git Repository. Instructor/TAs will check, weekly, to ensure the following are stored in your VCS repository:

https://github.com/grace-harrell/14-02-csci3308.git

- **Source Code**
- **Test Cases**
- **Video demo**
- **README.md in GitHub**
- **Project documentation**
- **Project Board**

- **Contributions:**
  - **A brief (not more than 100 words) from each team member about their contributions.**
    - **This should include the technologies worked on**
    - **Features that have contributed to**
  - **You can also include:**
    - **A screenshot of the project Board**
    - **A screenshot of the contributions on GitHub**

Cameron Davis:

- I mostly worked on the backend design also with a focus on integration between frontend and backend. I designed many of the APIs in the index.js file and jumped between ejs and NodeJS frequently to get everything playing together

nicely. I was mostly responsible for designing the backend systems for messaging, discovering potential roommates, and the database table layouts.

Jaden Feldman:

Most of my time during the project was spent on working on the frontend and database. I created data sets of users and their corresponding relations such as year, price, matches etc. and further integrated this data into our front end. I also created the presentation for the project and helped manage the github and project boards.

Grace Harrell:

The majority of my time was spent working, and touching up APIs. My main focus was the match api, which stored the information about user matches in the database. It prevents duplicate matches and sends a message to a user when someone requests to match with them. I also spend a lot of time managing the github and helping my teammates with their pull requests.

Mei Hecht:

I mostly worked on the basic initial front end design of the website for the pages like login, register, and a little bit of the roommates page. I also did a little bit of backend design pulling information from the users to display on the roommates matching page.

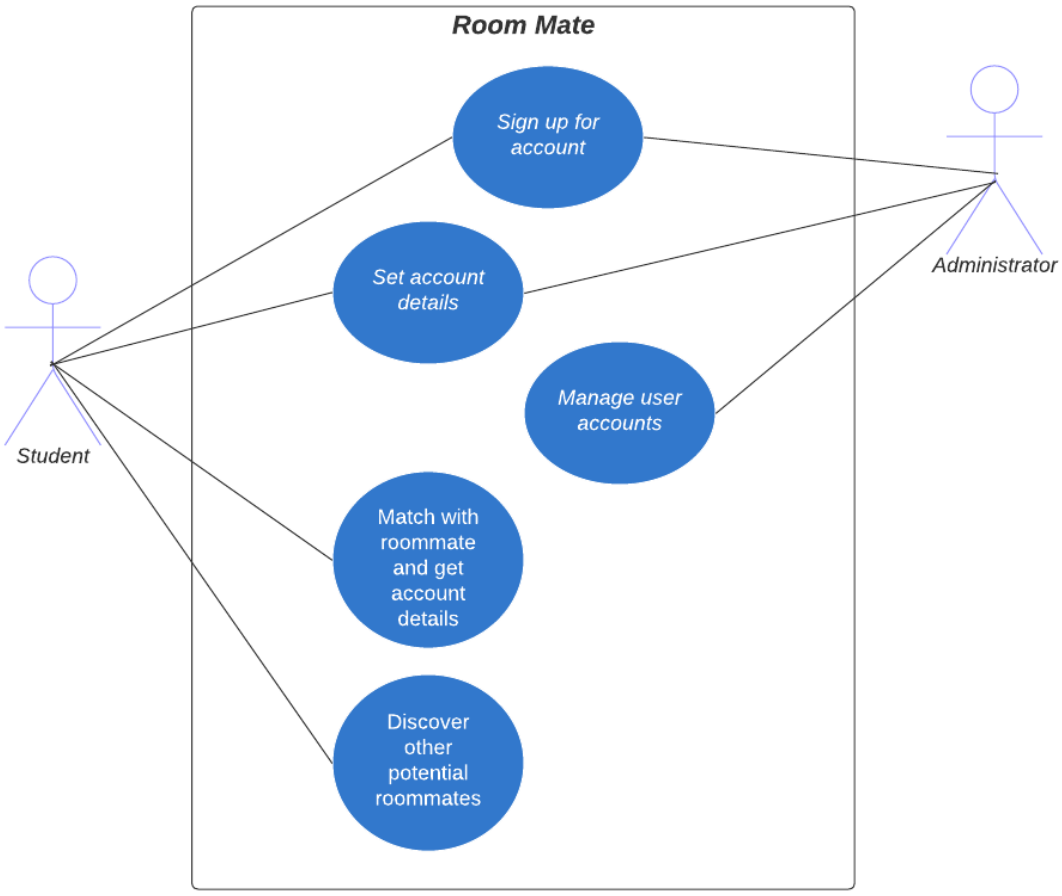
Brendan Lancaster:

The majority of my time working on our website was spent towards making it pretty on the front end. I created the initial figma prototype for us to go off of and shaped our pages to look like them as much as possible (mostly finalized versions of the roommates, inbox and profile pages, and touched up the others). I also wrote a script to pump out a bunch of dummy data to use on the roommates page.

Use Case Diagram:

Use case diagram

Cameron Davis | October 28, 2022



Test results: In Lab 11, you created a Test Plan. You need to include the test results and observations in the project report. Refer to this for more information

Requirements:
1. Create 1 document per team that describes how at least three features within your
finished product will be tested.
2. The test plans should include specific test cases (user acceptance test cases) that
describe the data and the user activity that will be executed in order to verify proper
functionality of the feature.
3. The test plans should include a description of the test data that will be used to test the
feature.
4. The test plans should include a description of the test environment that will be used to
test the feature.
5. The test plans should include a description of the test results that will be used to test the
feature.
6. The test plan should include information about the user acceptance testers. Please note
that the deployed application can only be tested on the CU Boulder campus. So make
sure to select your testers accordingly
1) Login page
Test Environment: QA environment
UA Testers: CU students looking to find a roommate on campus
a) Features
i) Authentication
(1) Test cases
(a) Logging in with a non exist user
(i) Expected Behavior:There should be a pop up
saying “user does not exist” and will stay on the
login page to re enter an existing user.
(b) Logging in with an incorrect password
(i) Expected Behavior: There should be a pop up
saying “ Incorrect password”

(c) Logging in with correct username and password:
(i) Expected behavior: When logging in with the correct information the user should be redirected to the home page
ii) Buttons / Links
(1) Register Button
(a) Expected Behavior: Redirects to the Registration page
(2) Login Button
(a) Expected Behavior: Redirects to the Homepage if the test cases are met
2) Carousel cards
a) Test cases:
i) The site should display a potential roommate's name and information on the screen when initially opening the page. There should be a button at the bottom of the screen which you select when you are interested in matching with the displayed user.
ii) If the user selects the the match button, then they should get a notification: "Added *Username*", and the account they selected should get a message in their inbox indicating that a user matched with them.
iii) If the user swipes left, the previous account they viewed should be displayed.
iv) If a user swipes right, a new account should be available for them to use.
v) If the user selected a dorm, all of the accounts displayed should also be assigned to that dorm.
vi) If the user has not selected a dorm, all of the accounts displayed should not be assigned a dorm.
vii) The accounts displayed should match the user's preferences.
viii) If the user has swiped through all of the current potential accounts, a message will be displayed "End of current selection".
b) Test environment
i) Local QA docker image with dummy data for database and admin

account access in the db.
3) User search
a) Test environment
i) Local QA docker image with dummy data for database and admin account access in the db
b) UA testers
i) CU students who may or may not actually be looking for a roommate.
c) Search bar
i) Should appear at the top of each page in the nav bar
(1) Should follow our styling guidelines
ii) Shouldn't be broken by updating html, static at the top of the page.
d) Text input
i) Numbers, special characters, very long inputs, as well as other unicode
that could break the text input. Should also test against common sql injection vulnerabilities.
(1) Server should sanitize special inputs and return an error, or not process the data at all when error-prone inputs are entered. As all searches should be names, special characters should not return.
ii) Empty input
(1) Should return an error for empty input
iii) Regular inputs with results
(1) Should post the search API and return to the results page with a list of users that were found.
iv) Regular inputs without results
(1) Should post the search API and return to the results page with an error stating the search returned no results

Deployment: Link to deployment environment or a written description of how the app was deployed and how one might access/run the app. The app must be live, working, and accessible to your TA.

- The app can be run through a simple docker compose file. It can be accessed through http://localhost:3000/ on the browser, and is in a fully live and functional state at the time of the presentation.