

Spacify

Sponsor - Prof. Sharad Mehrotra

Version: 4.0

Team Name: TechTippers

Members and Authors:

[Sanchi Gupta](#)

[Himanshu Thallam](#)

[Samruddhi Dharankar](#)

[Viraj Hiren Desai](#)

Table of Contents

| | |
|------------------------------------|-----------|
| Requirements | 4 |
| UseCase Diagram | 4 |
| UseCase Status Update | 4 |
| Update Incentives: | 4 |
| Create and Update a Room: | 4 |
| Check User Profile (update): | 5 |
| Create Reservations (update): | 5 |
| Subscribe/Unsubscribe (update): | 6 |
| Notification (new): | 6 |
| Modify Reservations Module (new): | 7 |
| Sign In and Sign Up Module: | 7 |
| Functional Requirements (update) | 8 |
| Non functional requirements | 15 |
| UX Design | 17 |
| User Persona 1: Sarah | 17 |
| User Persona 2: Dr.James | 19 |
| User Persona 3: John | 21 |
| Application Screens (update) | 23 |
| Sign Up screen (update) | 23 |
| Sign In screen (update) | 24 |
| Create Room Screen (update) | 24 |
| Make Reservation Screen (update) | 25 |
| Create rules screen (update) | 26 |
| User Profile screen (update) | 26 |
| Display My Rooms Screen (update) | 27 |
| Display incentives screen (update) | 28 |
| Subscribe Rooms screen (new) | 28 |
| Software Design | 29 |
| a. Use Case diagram | 29 |
| b. Basic architecture | 31 |
| c. System Flows | 32 |
| i. Rule creation flow | 32 |
| ii. Incentive Calculation flow | 32 |
| iii. Subscription flow | 34 |
| iv. Sign Up flow | 35 |
| v. Sign Up Verification flow | 35 |
| vi. Sign In flow | 36 |

| | |
|---|----|
| vii Subscription flow from user perspective (new) | 36 |
| viii Notification flow part 1 (new) | 37 |
| ix Notification flow part 2 (new) | 37 |
| d. Schema Design (ER diagram) | 38 |
| e. Module Descriptions (update) | 39 |
| f. Test Plan (update) | 39 |

Requirements

UseCase Diagram

[Diagram](#)

UseCase Status Update

Update Incentives:

Team member : [Sanchi Gupta](#)

Contribution:

- Analyzed and researched about the various rules engines present in the market today.
- Chose a rules engine that fit the project requirements and was compatible with the project tech stack - Drools
- Learned about the working of drools and integrated drools with the application.
- Gave a knowledge transfer session on the working of Drools.
- Designed the low-level implementation of the working of the Spacify engine (majorly used to calculate incentives.)
- Created http connections with TippersDB to fetch data for credit calculations

Team Member: [Viraj Hiren Desai](#)

- Created the tables for the application such as User, Incentive, Reservation, Room etc.
- Created mock data for the entire application to test out the developed features.
- Learnt about Drools from [Sanchi Gupta](#) and implemented the entire incentive calculation flow.
- Created Rule objects based on calculated and fetched data and provided them to drools to fire the rules and calculate the incentives for each satisfied rule.
- Implemented the configurable Occupancy and Duration Rules.
- Creates new entries in the Incentive table for the user who followed the rules which were present on the reserved rooms.
- Implemented the total incentive calculation for each user which is to get executed after each the incentive calculation flow is executed. This updates the total incentives in the user table for the particular user.

Create and Update a Room:

Team Member: [Himanshu Thallam](#)

Contribution:

- Researched and studied about ReactJS, Java and Spring Boot.
- Designed and implemented the create room API on the backend.
- The API creates a new record for the room in the Room table and a mapping in the Ownership table.

- Refactored tippers_space_id field to integer from Long, since TippersDB uses Integer ids.

Team Member: [Viraj Hiren Desai](#)

- Designed and created the UI screen with backend API connectivity for selecting the building and tippersDB space and creating a room on it. (claiming ownership)

Check User Profile (update):

Team Member: [Viraj Hiren Desai](#)

Contribution:

- Created the API for getting the user details from the backend.
- The API fetches the rooms subscribed by the user, the rooms owned by the user, the total incentives the user has, the user's incentive audit history, the user's personal details and the user's future room reservations.
- Created API for saving the update user information snapshot to the backend database once user clicks Save
- Integrated the user profile UI with backend APIs

Team Member: [Himanshu Thallam](#)

- Created basic UI for user profile with settings pane containing navigations to Rewards page and My Rooms page.
- Created Rewards UI page for the users which facilitates the users to view their current incentives along with the date.
- Created My Rooms UI page for the users to view all the rooms currently owned by them.
- (05/04/2023) Created Profile UI page which facilitates the user to update their First name, Last name, Email address and add the MAC addresses of up-to 3 devices. This page will also display the current user information which is editable.
- (05/25/2023) Integrated the User Profile UI with the back-end which will now render the user information when the user logs in. This information will be obtained via SessionStorage.
- (05/25/2023) Integrated the Rewards UI with the back-end which will render the current incentives of the user along with the points and the date.

Create Reservations (update):

Team Member: [Samruddhi Dharankar](#)

Contribution:

- Developed the API to take room type as path parameter and fetch all the available slots that can be reserved by the user
- Divided available time into slots available for users which allows them to reserve the rooms.

- Researched and integrated the several JPA functions to get the available slots according to various room types.
- Created table for Available time slots for the rooms to be reserved.
- Created an API for the entire Reservation process for reserving rooms.
- Designed and implemented the entire Reservation Backend Processing.

Team Member: [Himanshu Thallam](#)

- Created the UI page for room reservation.
- The UI page allows the user to select the room type, choose the date and time which will then display all the available rooms for the user to reserve.
- Redesign the front-end of the Reservation module. This includes removing the ability for the user to preselect the time but instead selecting only the date and the available rooms (which also include the time)
- Research about the front-end and back-end integration using React.
- Integration of the UI and back-end of the Reservation module.

Subscribe/Unsubscribe (update):

Team Member: [Sanchi Gupta](#)

- Designed the flow for subscription.
- Presented the subscription flow to Sponsor Professor.
- Created stories for the subscription flow.
- Connected with tippersDb to collect monitoring data and stored that data into the monitoring table

Team Member: [Sanchi Gupta](#)

Update:

- Created subscription panel in SpacifyUI.
- Displayed Rules on the subscription panel for each room.
- Created dummy data to calculate occupancy trends.
- Calculated and displayed occupancy trends for each room.

Team Member: Samruddhi Dharankar

- Developed a robust functionality that enables users to effortlessly access and manage their subscribed rooms.
- Introduced a feature that enables users to conveniently subscribe to WhatsApp with a single click to enhance their engagement with subscribed rooms.

Notification (new):

Team Member: [Sanchi Gupta](#)

- Designed the flow for notification.

- Presented the notification flow to Sponsor Professor.
- Created stories for notification flow.
- Researched about the various tools available for notification eg: telegram, sms, email, whatsapp.
- Learned about the connectivity with whatsapp and the terms and conditions to follow for the free trial version.
- Implemented a small Proof of concept to show the connectivity with whatsapp.
- Implemented webhook on whatsapp to trigger activity when a user responds to a message.
- Implemented interactive messages with buttons to allow unsubscription.

Team Member: [Himanshu Thallam](#)

- (05/25/2023) Learned about the whatsapp notification and how to work with their Cloud API.
- (05/25/2023) Tested their API via Postman to trigger a whatsapp notification with the “Unsubscribe” button.
- (05/25/2023) Implemented the back-end trigger API which will trigger a whatsapp message along with the “Unsubscribe” button to the registered number.

Modify Reservations Module (new):

Team Member: [Samruddhi Dharankar](#)

Contribution:

- Solved issues occurring in the backend of reservation module: In order to ensure the Reservation module functions properly, I conducted a thorough analysis of the backend code and identified and resolved any technical issues or bugs that were impacting the module's functionality.
- Integrated the front-end and back-end of Reservation
- Created a complete and efficient end-to-end flow for the Reservation module. This flow includes all the steps involved in making a reservation, from selecting a date and time, to selecting the rooms and confirming the reservation.

Sign In and Sign Up Module:

Team Member: [Viraj Hiren Desai](#)

Contribution:

- Created the user interface for the user to sign up, showing the required fields.
- Implemented the backend flow for creating the new user, storing their information and for storing the hashed password into the backend. Also checks whether the user has already signed up and shows the appropriate message on the UI.
- Designed and implemented the Sign Up Verification flow to authenticate the email provided by the new user.

- Designed and implemented the Sign In flow covered in the System Design session to satisfy this use case.
- Built UI screen where user enters their username and password to sign in. If the credentials are valid and correct, the user is signed in to the application.
- Implemented the backend APIs with various credential checks for verifying the credentials and returning the appropriate response message to the user on the screen,
- Sign Out button functionality , which deletes the session and local storage information for the user and signs them out of the application.
- Integrated Email sender module
- Implemented Resend Verification feature for Sign In which sends a new user verification email to the registered email address for verification.
- Built a Forget Password feature which generates a new password and sends it to the user's registered email address. This password can then be used to sign in the application and be changed from the User Profile screen.
- Added pop up modals for displaying and accepting 1. Terms and conditions and 2.Privacy Disclosure for the user when they sign up/ register themselves to the application.

Functional Requirements (update)

Must Have:

1. As a user I should be able to search rooms that are available for booking in the Spacify database so that I can reserve or subscribe to them

Completed By: [Himanshu Thallam](#)

- Created a Reservation User Interface page.
- This facilitates the user to select some information which will then be used to display all the rooms that are available for the user to reserve based on the given information.
- This includes the room type such as study or office, date and time for which the user wants to reserve the room.
- The option to select time has been removed from the UI page. The user can only select the date and there will be a list of rooms available for the user to reserve which will also include the timings of the availability.
- The UI has been integrated with the backend which takes in input from the user regarding room type and date and fetches all the rooms available of that type and for that date.

2. As a user I should be able to see the time slots available for booking a valid Spacify room and also learn about the rules that govern the room so that I can make an informed decision while booking the room.

Completed By : [Samruddhi Dharankar](#)

Update:

- Designed the low-level implementation for booking the reservation.

- Created the Spacify-Engine portal
 - When a user selects a room type on the frontend of the portal, the reservation function takes that information and uses it to query the backend database to find all available rooms of that type.
 - Used the reservations that have already been made and stored in the database to determine which time slots are available for the selected date and room type. This involves querying the database for the relevant data, understanding the reservation schema, and filtering out the correct time slots based on the user's selection.
 - Implemented the logic to take the available time slots and divide them into one-hour increments so that they can be displayed on the frontend of the portal. This involves writing code that can parse the time data, calculate the correct intervals, and present the information in a user-friendly format.
3. As a user I should be able to see my incentive for the reservation credited to my account within the 24 hours of the completion of my reservation so that I can redeem the incentives for my benefit.
- Completed By: [Viraj Hiren Desai](#)
- This was achieved in the incentive calculation flow mentioned below (Points 13).
 - However a separate function for doing this was also implemented in case we want to run the incentive calculation and incentive addition separately.
4. As a user I want to register my profile with Spacify so that I can use the application features.
5. As a user I should be able to login to my profile once it has been created in Spacify so that I can proceed to create or reserve a room
6. As a user I should be able to search and choose a room that is available in the database so that I can take ownership of that room and create rules for it.

Completed By : [Sanchi Gupta](#), [Himanshu Thallam](#) and [Viraj Hiren Desai](#)

Update:

Team member: [Sanchi Gupta](#)

- Collaborated with Professor Sharad's team to understand the data stored in TippersDb
- Created http connections with Tippers Db to fetch the details of the rooms and the buildings available within TipperDb
- Created RestControllers in the backend that could be called from the frontend to display the data on the screen.

Team member: [Viraj Hiren Desai](#)

- Designed and created the UI screen with backend API connectivity for selecting the building and tippersDB space and creating a room on it. (claiming ownership)

Team member: [Himanshu Thallam](#)

- Designed and implemented the create room API on the backend.
- Whenever the user creates the room, the API creates a new record for the room in the Room table and a mapping in the Ownership table.
- Refactored tippers_space_id field to integer from Long, since TippersDB uses Integer ids.

7. As a user I should be notified if the room is pre-owned by some other user as soon as I select the room so that I cannot create rules on it.
8. As a user I should be able to see all the rules available for a room and provide my desired values for the rule so that I can create rules of my choices for the room.

Frontend Completed By: [Himanshu Thallam](#)

- Created a UI page called Edit Rules.
- This will allow the users to edit the rules associated with the room that they have created.

9. As a user I should be able to subscribe to a room so that I can be notified when the room is empty.

Team member: [Sanchi Gupta](#)

- Designed the flow for subscription.
- Presented the subscription flow to Sponsor Professor.
- Created stories for the subscription flow.
- Designed the flow for notification.
- Presented the notification flow to Sponsor Professor.
- Created stories for notification flow.
- Researched about the various tools available for notification eg: telegram, sms, email, whatsapp.
- Learned about the connectivity with whatsapp and the terms and conditions to follow for the free trial version.
- Implemented a small Proof of concept to show the connectivity with whatsapp.
- Created subscription panel in SpacifyUI.
- Displayed Rules on the subscription panel for each room.
- Created dummy data to calculate occupancy trends.
- Calculated and displayed occupancy trends for each room.
- Created notification if the room is already subscribed.

10. As a user I should be able to unsubscribe to a room so that I do not receive any further notifications about the room

Team member: [Sanchi Gupta](#)

- Designed the flow for unsubscription.
- Implemented webhook on whatsapp to trigger activity when a user responds to a message POC.
- Implemented interactive messages with buttons to allow unsubscription POC.

11. As a user I want to see the summary of my incentives so that I have the audit history of all the transactions of my incentives.

Backend Completed By: [Viraj Hiren Desai](#)

- Implemented the REST API for fetching all the incentives received by the user, the incentive history and the total incentive the user currently has. The API also fetches other user data such as subscribed rooms, owned rooms and other personal user details.

Frontend and Integration Completed By: [Himanshu Thallam](#)

- Created My Rewards UI page which is available in the User Profile page.
- This page will display the current incentives obtained by the user along with the date.
- (05/25/2023) This page is fully integrated with the backend API which will now display the current incentives along with the points and the date.

12. As a user I want to be shown all the necessary privacy disclaimers I should agree to before registering with the application so that there is no breach of privacy.

Team Member: [Viraj Hiren Desai](#)

- Designed the screen for displaying the Privacy disclaimer and accept checkbox.
- Currently drafting the disclaimer contents to be shown to the user.

13. As a user I want to implement a rule of the following format on the room I own:

"If the visitor utilizes the room for X% time of the booking then credit Y amount as incentive to the user's account" and name it as Duration rule. Here the X and Y values are to be taken from me via the UI. This is implemented so that I can create a rule where the visitor is credited the incentive based on the usage of the room and the rule created by me.

Completed by: [Sanchi Gupta](#) and [Viraj Hiren Desai](#)

Update:

Team member: [Sanchi Gupta](#)

- Designed the low-level implementation of incentive calculation
- Created http connections with Tippers Db to fetch the mac address details of the given tippers spaceld.
- Created an abstraction level that fetches data from TippersDb, parses it into the required format and provides the necessary details to the caller to help in the calculation of incentives for the rules.

Team member: [Viraj Hiren Desai](#)

- Tweaked the low level design implementation a bit for the calculation flow.
- Created logic for taking up the uncalculated reservations, the occupancy of the room during those reserved time slots, fetching rules for the reserved rooms and processing this data to calculate the actual rule thresholds and firing the rule engine, thereby completing the incentive calculation flow.
- Also adds new incentives to the incentive audit table if new incentives were provided to the user along with calculating and saving the total incentives for the user who received the points.
- Implemented the actual timestamp duration calculation logic for the Duration rule to get the expected reservation duration and the actual time duration for which the user who reserved the room was present in the room.

14. As a user I want to implement a rule of the following format on the room I own:

"A room can only accommodate X number of people, if the room is utilized by X or lesser number of people then credit the visitor with Y amount". This is the Occupancy Rule. Here the X and Y values are to be taken from me via the UI. This is implemented so that I can create a rule where the room is occupied only by a limited number of people

Completed by: [Sanchi Gupta](#) and [Viraj Hiren Desai](#)

Team member: [Sanchi Gupta](#)

- Designed the low-level implementation of incentive calculation
- Created http connections with Tippers Db to fetch the occupancy details of the given tippers spaceid.
- Created an abstraction level that fetches data from TippersDb, parses it into the required format and provides the necessary details to the caller to help in the calculation of incentives for the rules.

Team member: [Viraj Hiren Desai](#)

- Tweaked the low level design implementation a bit for the calculation flow.
- Created logic for taking up the uncalculated reservations, the occupancy of the room during those reserved time slots, fetching rules for the reserved rooms and processing this data to calculate the actual rule thresholds and firing the rule engine, thereby completing the incentive calculation flow.
- Also adds new incentives to the incentive audit table if new incentives were provided to the user along with calculating and saving the total incentives for the user who received the points.
- Implemented the actual occupancy calculation logic for the Occupancy rule to get the expected number of guests and the actual guests present in the reserved room..

15. As a user, I want to implement a rule of the following format on the room I own:

"X must always be present with Y when X visits the room." Here the X and Y values are to be taken from me via the UI. This is implemented so that I can control who visits the room.

16. As a user, I should be able to navigate between various screens on the user interface.

Completed By: [Viraj Hiren Desai](#)

- Setup ReactJS application and the React-bootstrap module for starting the frontend work.
- Created the project structure for the UI module.
- Implemented the Navigation bar to allow the user to switch between different screens.

17. As a user, I want to be able to view and edit my personal information. This includes my First Name, Last Name, Email address, etc.

Completed and Integrated by: [Himanshu Thallam](#)

- Created UI for the profile page. This will essentially render the user information such as First Name, Last Name, Email Address and Mac Address.

- These fields are editable. This allows the users to update their personal information.
- They can also add the Mac address of up to 3 devices.
- (05/25/2023) Integrated the Profile page with the back-end API which will render all the required user information.

Completed by: [Viraj Hiren Desai](#)

- Implemented the fetch API for getting user details
- Implemented a save API which takes in the user edited information and updates the backend database.

18. As a user, I should be able to register/ sign up to be able to use the application. I should be able to enter my details such as first name, last name, email, mac address and setup my password.

(04/04/2023)

Completed by: [Viraj Hiren Desai](#)

- Created the user interface for the user to sign up, showing the required fields.
- Created terms and conditions, and privacy disclaimer which the user has to accept before signing up for the application.
- Implemented the backend flow for creating the new user, storing their information and for storing the hashed password into the backend. Also checks whether the user has already signed up and show the appropriate message on the UI.

19. As a user, upon signing up for the application, I should receive an email verification email to authenticate myself.

(04/04/2024)

Completed by: [Viraj Hiren Desai](#)

- Designed and implemented the Sign Up Verification flow covered in the System Design session to satisfy this use case.

20. As a user, I should be able to sign in to the application for using various functionalities.

(04/04/2024)

Completed by: [Viraj Hiren Desai](#)

- Designed and implemented the Sign In flow covered in the System Design session to satisfy this use case.
- Built UI screen where user enters their username and password to sign in. If the credentials are valid and correct, the user is signed in to the application.
- Implemented the backend APIs with various credential checks for verifying the credentials and returning the appropriate response message to the user on the screen,
- Once signed in, the user remains signed in across all the application screens and is able to use all the functionalities they have access to.

21. As a user, I should not sign in each time I use the application if I have selected the Remember Me checkbox during the previous sign in.

(04/04/2024)

Completed by: [Viraj Hiren Desai](#)

- Implemented the Remember Me checkbox functionality to keep the user always signed in until they log out.
- Used a combination of browser's local storage and session storage functionalities for building this feature. The user remains signed in even after reopening the browser.

22. As a user, I should be able to log out/ sign out when I want.

(04/04/2024)

Completed by: [Viraj Hiren Desai](#)

- Built a Sign Out button, which deletes the session and local storage information for the user and signs them out of the application.

23. As a user, I should have the ability to sign in the application even though I have forgotten my password.

Completed by: [Viraj Hiren Desai](#)

- Implemented the Forget Password feature which generates a new password and sends it to the user's registered email address. This password can then be used to sign in the application and be changed from the User Profile screen.

24. As a user, I should have the ability to re verify myself in case I have not received the previous email verification mail.

- Integrated Email sender module
- Implemented Resend Verification feature for Sign In which sends a new user verification email to the registered email address for verification.

25. As a user, I should receive a whatsapp notification whenever the room I subscribe to becomes available.

(05/25/2023)

Team Member: [Himanshu Thallam](#)

- Learned about the whatsapp notification and how to work with their Cloud API.
- Tested their API via Postman to trigger a whatsapp notification with the "Unsubscribe" button.
- Implemented the back-end trigger API which will trigger a whatsapp message along with the "Unsubscribe" button to the registered number.

Nice to have:

1. As a room owner I should be able to edit the room rules so that I can add or delete the rules of the room
2. As an admin I should be able to approve the request of room ownership so that a user can own the room and rules on it.
3. As a room owner I should be able to add and delete users from my room queue so that the notifications are only sent to the required users.

4. As a room owner I should be able to add or delete other users as room owners so that I can control who changes the properties of the room.
5. As a user I should be able to raise a request to the admin so that I can raise concerns and doubts.
6. As an admin I should be able to access all the concerns and doubt of my user and be notified when a request has been raised
7. As an admin I should be able to add FAQs that contain common concerns and doubts of the users so that the basic doubts of the users can be cleared without raising a request.

Great if we can get to it:

1. As a room owner I want to be able to advertise about my room so that I can increase the usage of my room.
2. As a room owner I want to penalize the visitors when the visitor breaks a rule so that the conundrum of the room is maintained.
3. As a user I want to be able to see recommendations on where I can use my incentives so that I am aware of all the options available for me.
4. As a user I want to see the recommendations of bookings that I can make so that I can make the purchase of my choice.
5. As a user I should be able to choose the access levels for each room so that only users of certain access levels can perform activities on rooms within Spacify
6. As a user I want to have the ability to credit partial amounts of incentives based on my choices so that the users are credited incentives even if they follow a rule partially.
7. As an admin I should be able to add more rooms to the Tippers DB map so that more rooms are available to the Spacify application.

Non functional requirements

1. As a developer I want to connect to the TippersDB API so that we can show the rooms available and also fetch information related to their occupancy.
2. As a developer I want to ensure that my application is deployed in different regions so that my application is available 24*7
3. As a developer I want to configure Jenkins with my code repository so that deployments are regular and fast
4. As a developer I want to ensure that all the passwords used by my application are stored securely so that my application data is safe.
5. As a developer I want to enable kerberized connectivity with the database so that the developer is rested from the duty to rotate passwords and securely store them.
6. As a developer I want to create a common database that can be used by all the developers so that the data used to test the application is the same.
7. As a developer I want to build a script that automatically deploys the UI, portal and engine and brings them up so that the developer does not have to deploy each component separately.
8. As a developer I want to create a use-case diagram of the application so that the flow of the application is clear
9. As a developer I want to create an ER diagram so that we can use it to create and set up PostgreSQL database
10. As a developer I want to create an architecture diagram so that the application is scalable and available always.
11. As a developer I want to create flow charts for three major flows of the application, namely, create room, incentive flow and subscription flow so that the flow of control is clear.
12. As a developer I want to create UI/UX templates to design the look of the application so that we can create similar GUI for the application.

13. As a developer I want to create an API that accepts all the data for creating a room, parses it and stores the data in the database and then sends a successful response to the UI so that the user knows that the room has been created successfully.
14. As a developer I want to create a backend setup so that I can start development in an orderly manner.
15. As a developer I want to create a frontend setup so that I can start development in an orderly manner.
16. As a developer I want to choose an appropriate rules engine and understand its working and set it up in the engine of Spacify so that we can start implementing rules.
17. As a developer I want to implement asynchronous multi threaded calculation of incentives so that the calculations can be done in parallel.
18. As a developer I want create the Spacify-engine and configure postgreSQL in it so that we can implement incentive calculation program flow

UX Design

User Persona 1: Sarah

Basic Information



Sarah

"Student"

- 20
- Bachelor's degree
- Irvine, CA

Bio

Sarah is a driven and ambitious student who wants to excel in her academics. She is currently preparing for her final exams and needs a quiet study space to focus. Sarah lives in a dorm room on campus, but it is often noisy and distracting, making it difficult for her to concentrate.

More about Sarah

Interests



interested in biology and enjoys reading about new scientific discoveries and advancements

Influences

Her professors and academic advisors influence her academic and career decisions, providing guidance and mentorship

influenced by the resources and facilities available on campus, such as the study rooms and library resources

Needs and expectations

| | | | |
|--|--|--|---|
| She needs a system to book study rooms in advance to ensure she has a dedicated space for her study sessions during finals week. | She expects the study room to be available during the time slot she has reserved it for. | Sarah expects the booking process to be easy and seamless. | She expects to receive notifications when the study room becomes available. |
| She wants a comfortable and quiet environment where she can focus on her studies and collaborate with others. | She also expects the room to be available when she needs it and to be free from any distractions or interruptions. | Sarah needs the room to be conveniently located on campus and easily accessible so that she can make the most of her study time. | Sarah needs the room for combined studies or meetups with the teachers and professors and other students. |

Motivations

Sarah is motivated by her desire to do well in her academics and succeed in her future career

She is also motivated by her passion for learning and expanding her knowledge in the field of biology

Sarah is motivated by the opportunity to earn incentives for following up on her bookings. She understands that these incentives are a reward for her responsible behavior and encourage her to stay on track with her studies.

Additionally, Sarah is motivated by the idea of improving her academic performance and achieving her goals, and she sees following up on her bookings as a way to achieve this.

Goals

Wants to study in a quiet room with no disturbance

She wants to be notified when the study room becomes available so that she can go there right away.

Wants to gain more incentives

Pain points and frustrations

Sarah often struggles to find a quiet and comfortable study space on campus.

She finds it frustrating when she has to wait for a study room to become available, especially during finals week when demand is high.

frustrated when she cannot find a study space that meets her needs

She gets easily distracted by noise and other people, which affects her concentration and productivity.

frustrated by the lack of maintenance and cleanliness in some study rooms

User Persona 2: Dr.James



Dr. James Smith

"Professor"

- 45
- PhD in Computer Science
- Irvine, CA

Bio

Dr. James Smith is a highly qualified professor with over 15 years of teaching experience in the field of computer science. He is a dedicated and passionate educator who is committed to providing his students with the best possible education. Dr. Smith is currently preparing to teach a course with 140 students and wants to ensure that all of his students attend his classes.

[More about Dr. James Smith](#)

Interests

interested in the latest developments in the field of computer science and technologies



passionate about teaching and wants to ensure that all his students have the opportunity to learn and succeed in their studies.

Influences

biggest influence is his passion for teaching and his desire to provide his students with the best possible education

influenced by his colleagues in the field of computer science, who provide him with valuable insights and ideas for improving his teaching methods

Needs and expectations

Finding a classroom that can accommodate all of his 140 students and has the necessary equipment to facilitate his lectures

Needs a classroom that is conveniently located on campus to ensure that his students can easily attend his classes

Expects the classroom to be available when he needs it and to be conveniently located on campus.

Expects the classroom to be free from any distractions or interruptions, so that his students can focus on their studies and participate actively in the class

Needs slots of his availability to be booked & followed properly using incentives, as this encourages his students to attend his classes

Motivations

Dr. Smith is motivated by his passion for teaching and his desire to make a positive impact on his students' lives

He is also motivated by the opportunity to inspire his students and help them achieve their academic and career goals

hopes that incentives and rewards will help encourage them to prioritize their studies.

Pain points and frustrations

limited availability of classrooms on campus, especially during peak periods such as the start of the academic year

Dr. Smith is also frustrated by students who do not attend his classes regularly, as this affects their academic performance and his ability to provide them with the best possible education

Dr. Smith is concerned about his students' ability to learn effectively in a virtual or hybrid learning environment, and is worried about the impact of technology on their learning outcomes.

administrative tasks such as booking classrooms and tracking attendance, as these tasks can take time away from his teaching and research activities.

Goals

Ensure that all of his 140 students attend his classes regularly

Wants to book a classroom that can accommodate all of his students comfortably and has the necessary equipment to facilitate his lectures

He may also use a point system or reward program to track and incentivize attendance and participation, and may offer rewards such as gift cards, free textbooks, or other prizes to students who accumulate a certain number of points.

Dr. Smith may offer incentives such as extra credit, bonus points, or exemptions from certain assignments or exams to students who attend his classes regularly and participate actively in class.

User Persona 3: John



John
"University Administrator"

- 37
- Irvine, CA

Bio

John has been working in university administration for over 10 years, and is responsible for overseeing the use of campus facilities, including classrooms and study spaces. He is passionate about creating a safe and organized learning environment for students, and believes that efficient and systematic use of campus resources is key to achieving this goal.

[View user info](#)

Interests



John is interested in promoting a positive and inclusive campus culture that fosters learning and personal growth.

He is passionate about creating opportunities for students to connect with each other and with faculty members outside of the classroom

Influences

John is influenced by the needs and expectations of students, faculty members, and other stakeholders on campus.

He is also influenced by broader trends in higher education, including changes in technology, pedagogy, and student demographics.

Needs and expectations

A system that allows for real-time monitoring and tracking of facility usage, including occupancy rates and frequency of usage

User-friendly interface that can be easily accessed and navigated by both students and faculty

Expects this system to be user-friendly and accessible to all students and faculty members, and to provide real-time availability information to prevent double bookings and overcrowding.

Access to data and analytics that can inform decisions about resource allocation and facility upgrades

Integration with existing campus systems and software, including student information systems and event management platforms

Pain points and frustrations

- He is concerned about overcrowding and the potential for conflict between students who need access to the same resources.
- Limited resources available for upgrading or expanding campus facilities
- Frustrated by the lack of discipline and responsibility exhibited by some students, who may leave classrooms and study spaces in disarray or fail to show up for scheduled bookings.
- Concerned about the financial sustainability of the university, and wants to ensure that resources are used efficiently and effectively to promote long-term growth and success.
- Difficulty enforcing policies and regulations related to facility usage and student behavior

Goals

- Primary goal is to implement a system for booking and using campus facilities that promotes discipline and prevents overcrowding.
- Improve communication and collaboration between different departments and stakeholders on campus
- Wants to ensure that all students have access to the resources they need to succeed academically, while also ensuring that these resources are used efficiently and effectively.
- Wants to promote school culture and spirit by allowing students to use incentives at college goods stores
- Increase efficiency and effectiveness of facility usage through centralized booking and management
- Ensure long-term financial sustainability of the university by using resources efficiently and effectively
- Promote a positive and inclusive campus culture through extracurricular activities and events

Motivations

- John believes that incentives and rewards can be an effective way to promote discipline and responsibility among students, as well as to encourage them to participate in campus events and activities.
- He may use a variety of incentives, including discounts or free items at the college goods store, priority booking for popular spaces or events, and recognition or awards for students who demonstrate outstanding behavior or performance.
- John may also partner with local businesses or organizations to offer additional incentives or rewards to students who participate in community service or other initiatives that promote the university's values and goals.

Application Screens (update)

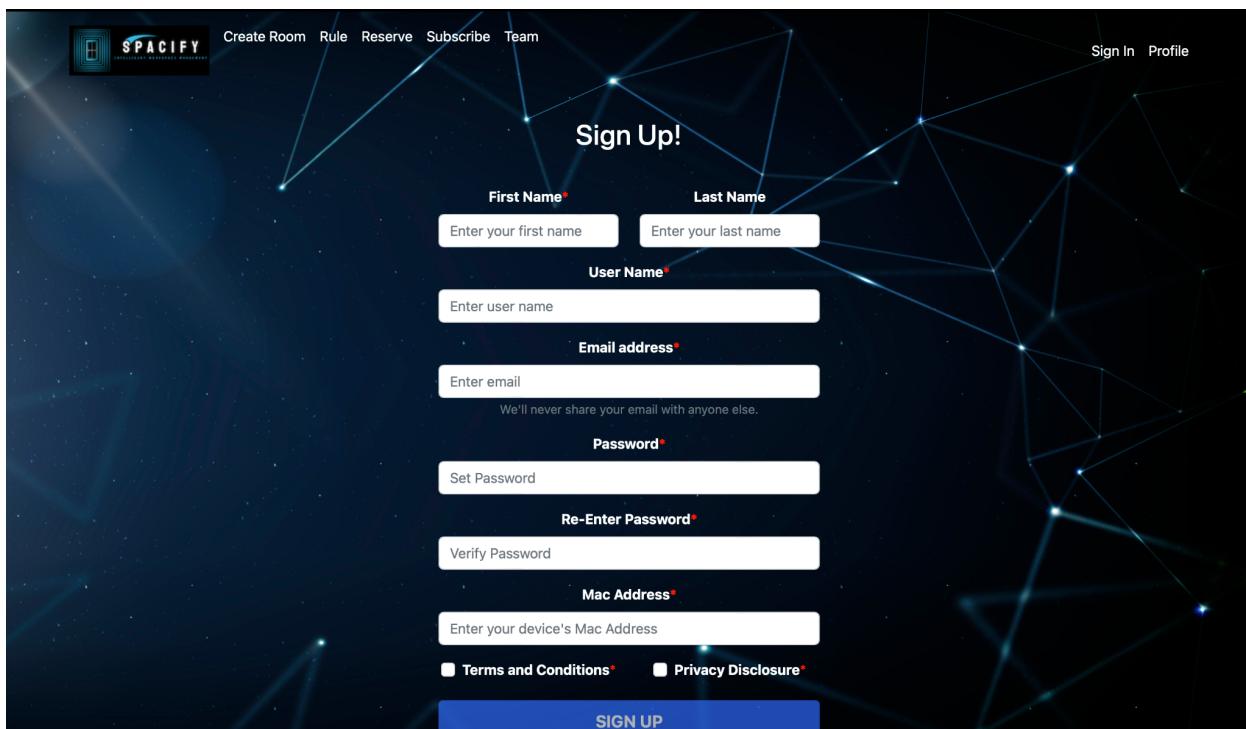
As reported in the previous sprint meeting, we are focusing on building the core of our application first - the rules engine, reservation and incentive calculation part. However, we have started developing the basic UI screens required for the MVP.

Update by: [Viraj Hiren Desai](#)

Revamped the UI feel by improving the theme of the application and also developed a new logo for Spacify!

Sign Up screen (update)

This is the screen where the user creates a new account. The user fills in their details such as name, email, uci net id, mac address and sets up their password. The user also needs to agree with the Terms and Conditions, and Privacy Disclaimer while signing up. This can be done by clicking the checkboxes provided. On clicking Sign Up, the system goes through the verification flow and a new user is created in the application.



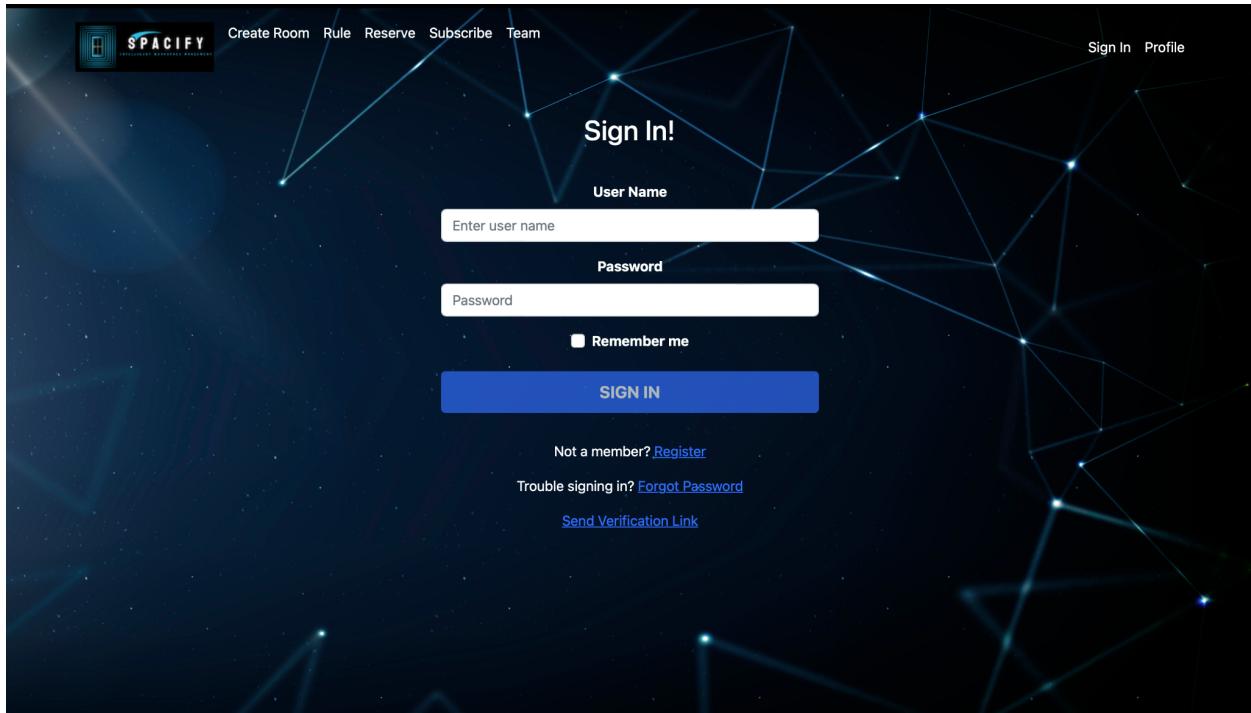
The image shows the Spacify Sign Up screen. The background is dark blue with a network of glowing blue lines and dots, giving it a futuristic, interconnected feel. At the top left is the Spacify logo. Along the top navigation bar are links for 'Create Room', 'Rule', 'Reserve', 'Subscribe', and 'Team'. On the right side of the header are 'Sign In' and 'Profile' links. The main title 'Sign Up!' is centered at the top. Below it are several input fields with validation requirements indicated by red asterisks:

- First Name***: Enter your first name.
- Last Name**: Enter your last name.
- User Name***: Enter user name.
- Email address***: Enter email. A note below says "We'll never share your email with anyone else."
- Password***: Set Password.
- Re-Enter Password***: Verify Password.
- Mac Address***: Enter your device's Mac Address.

At the bottom of the form are two checkboxes: Terms and Conditions and Privacy Disclosure. A large blue 'SIGN UP' button is positioned at the very bottom center.

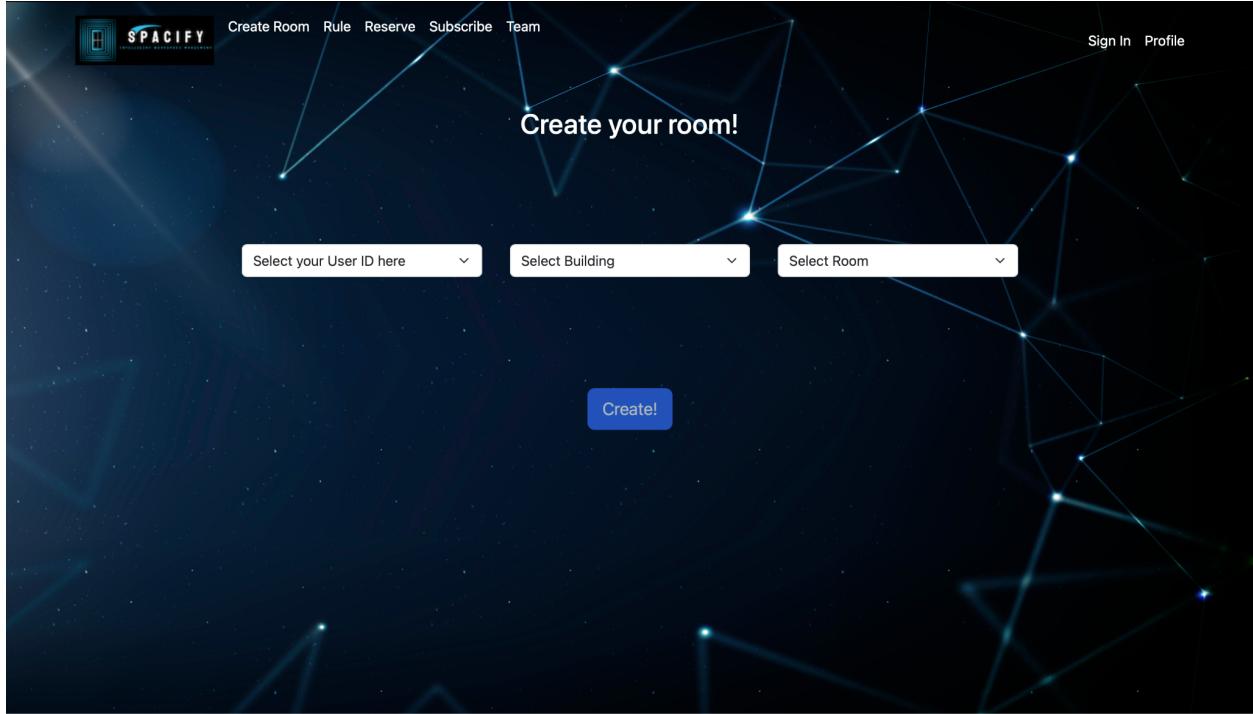
Sign In screen (update)

In order to use the application to its full capacity, the user needs to sign themselves in. The user enters their Username and password. Then they click the sign in button, on which the application verifies their credentials and signs them in. If the remember me option is selected, the application keeps the user signed in on the machine till they logout. Else if the option is not selected, the user is logged out when they close the application tab.



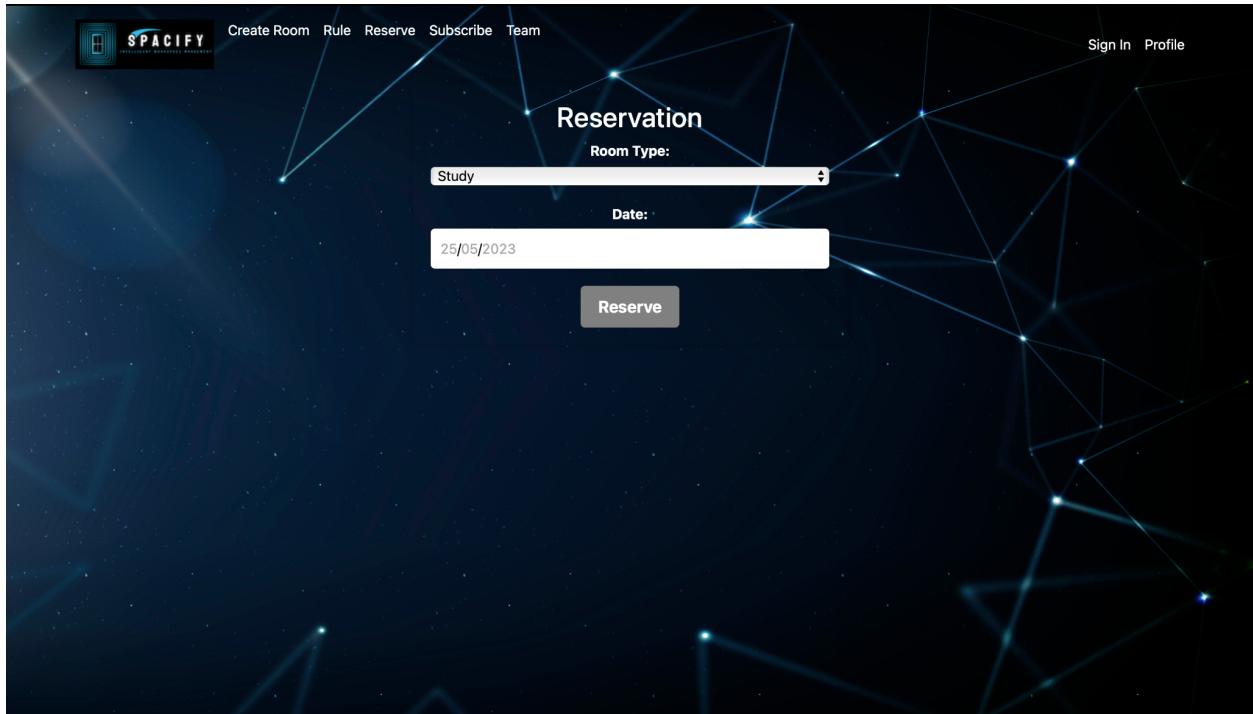
Create Room Screen (update)

In the Create Room Screen the authorized and signed in user can create a room from all the available spaces present in the system. This has been completed and the following is the current screen. If the room has already been created on the selected space, the user will be notified of the same and given the contact of the owner of the room for clarity. If possible, the room will be created and the user becomes the owner of that particular room.



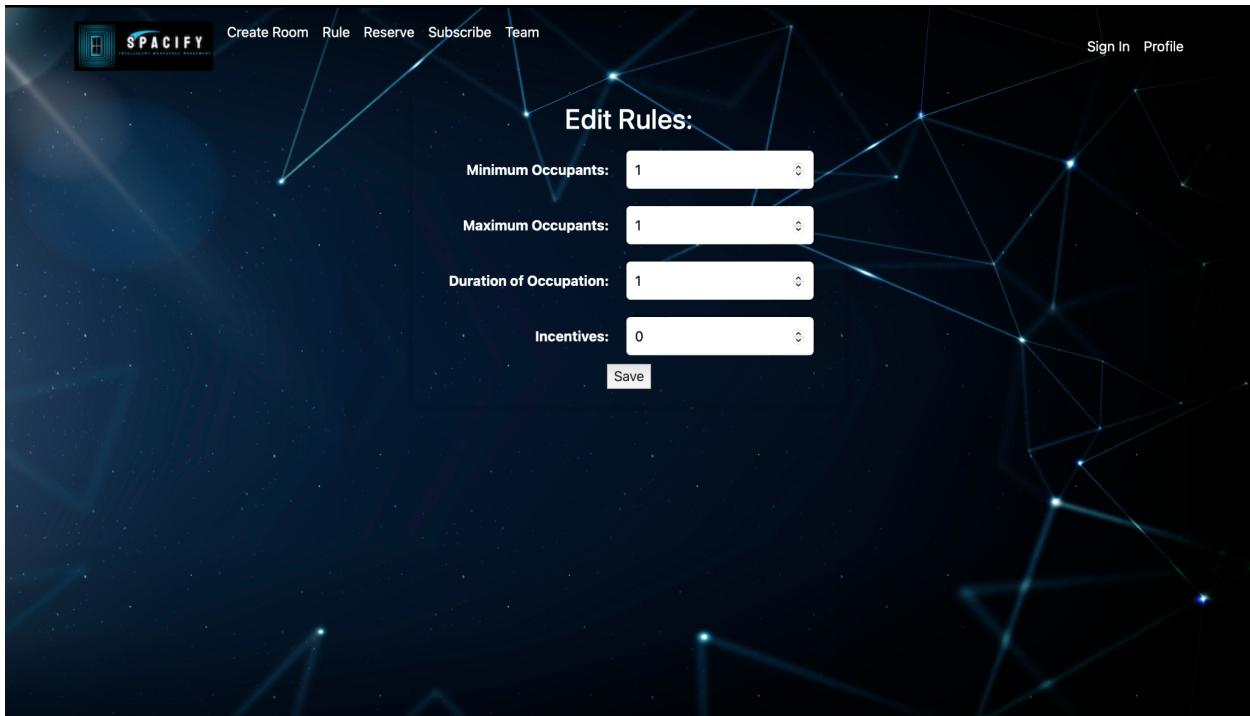
Make Reservation Screen (update)

This page displays all the available rooms to the user based on the room type and the date selected by the user. The rooms are displayed on hourly basis of availability.



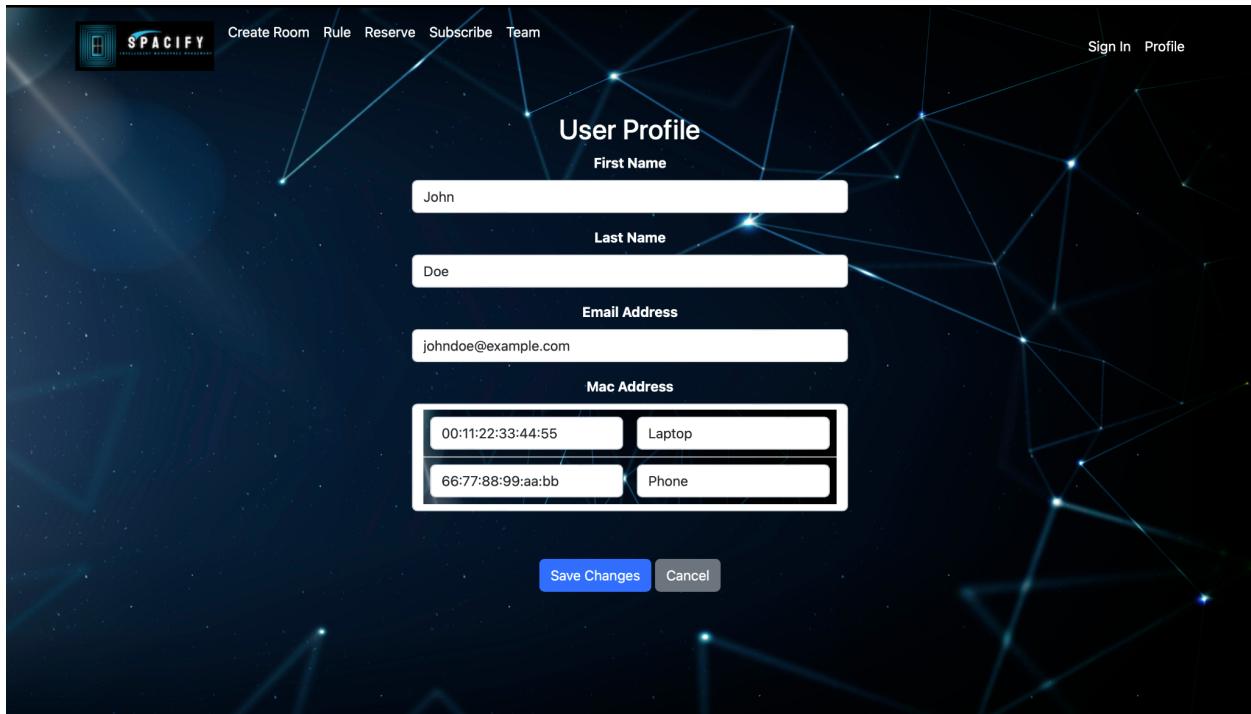
Create rules screen (update)

Here we create rules for a particular room, like from which time to which time are people allowed in the room and what is the maximum occupancy for the room. The rules are set by the room owner using the following screen.



User Profile screen (update)

This screen is used for maintaining the profile of the user.



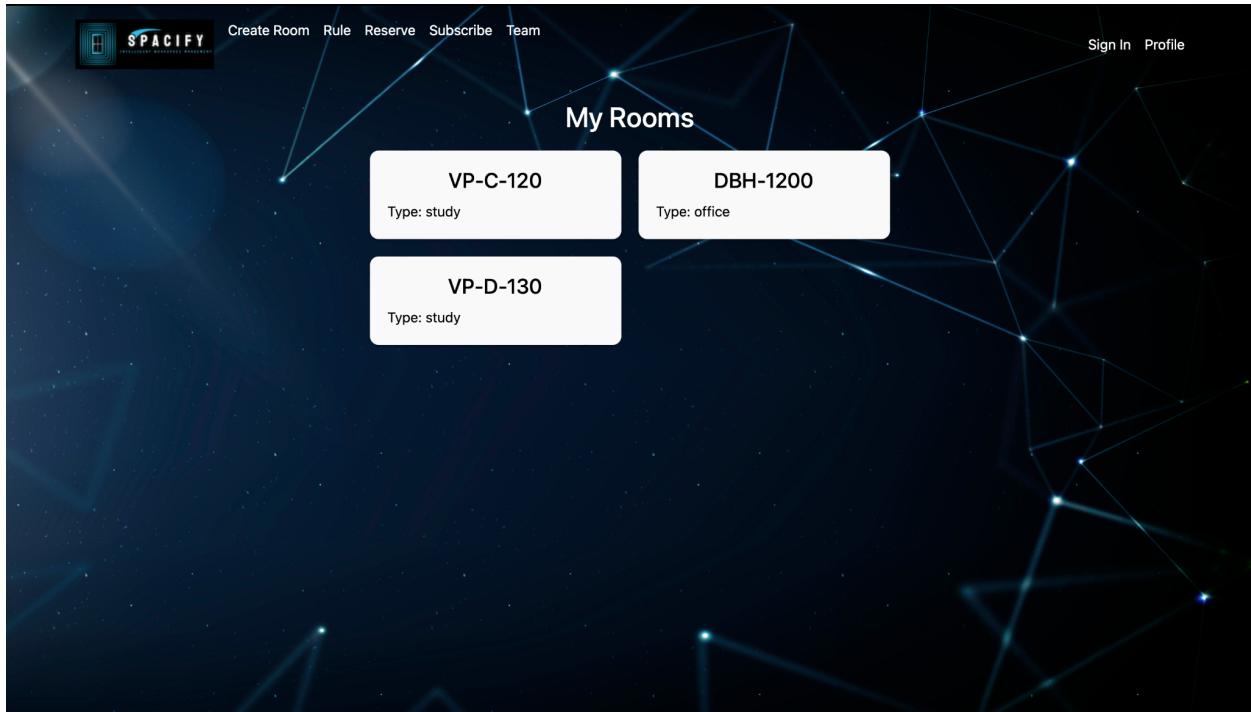
The screenshot shows the User Profile form on a dark-themed interface. At the top, there are navigation links: Create Room, Rule, Reserve, Subscribe, Team, Sign In, and Profile. The main title is "User Profile". The form fields include:

- First Name: John
- Last Name: Doe
- Email Address: john.doe@example.com
- Mac Address:
 - 00:11:22:33:44:55 (Laptop)
 - 66:77:88:99:aa:bb (Phone)

At the bottom are "Save Changes" and "Cancel" buttons.

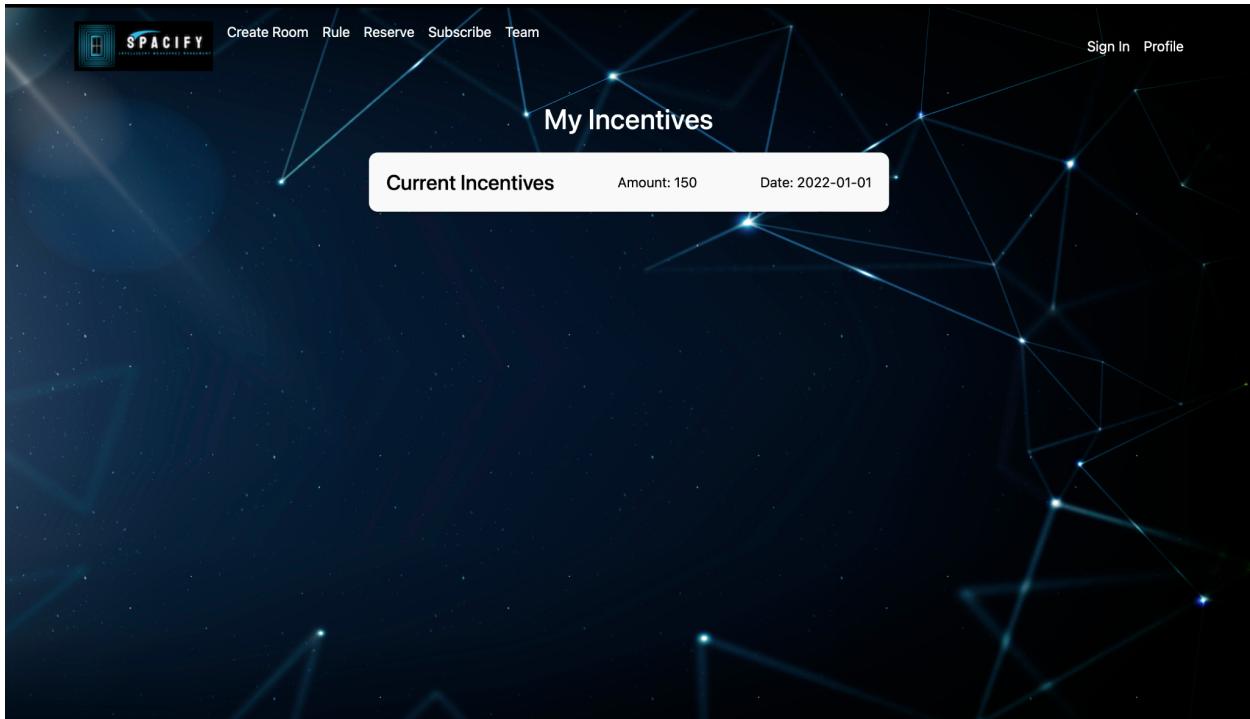
Display My Rooms Screen (update)

This screen shows all the rooms the user has subscribed to.

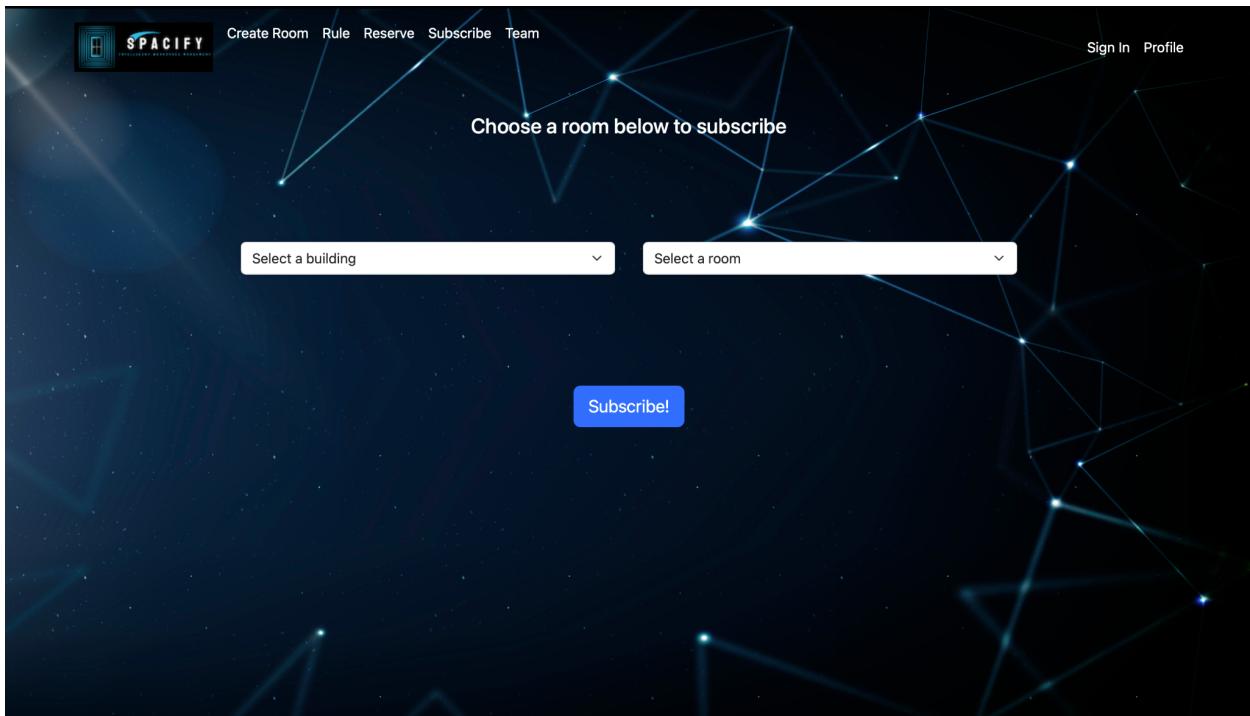


Display incentives screen (update)

This screen displays the users current incentives which also includes the amount (points) and the date the incentive was received.



Subscribe Rooms screen (new)



Software Design

a. Use Case diagram

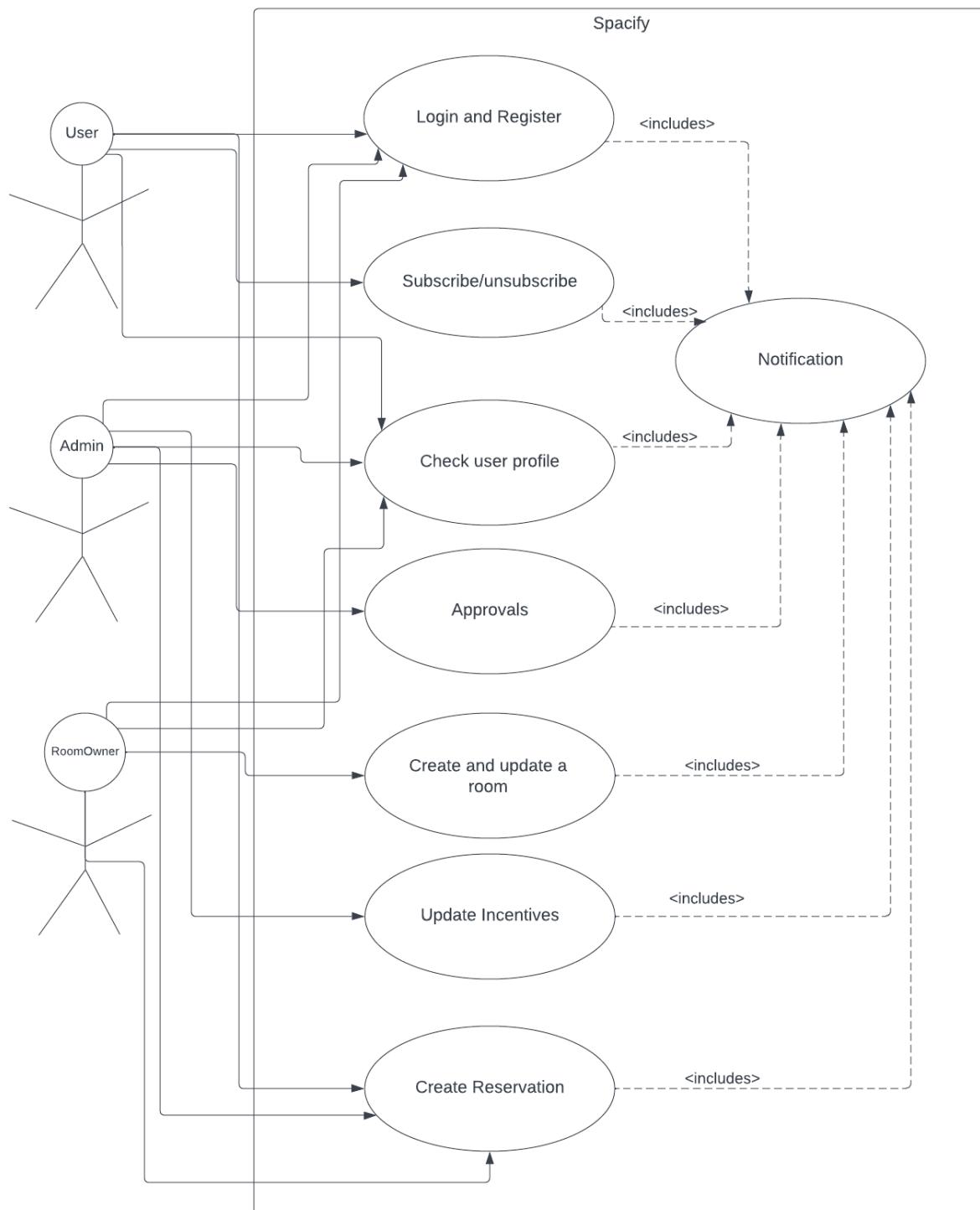
The below diagram is the Use Case diagram for the application. There are a total of three different actors that interact with the system.

User: The one who reserves rooms, subscribes to rooms and get incentives for following the rules of the room. User is a student most of the times.

Admin: This is a super user, which can do anything in the system. An admin can create rooms, own rooms, give access to other users and can make approvals for room ownership requests.

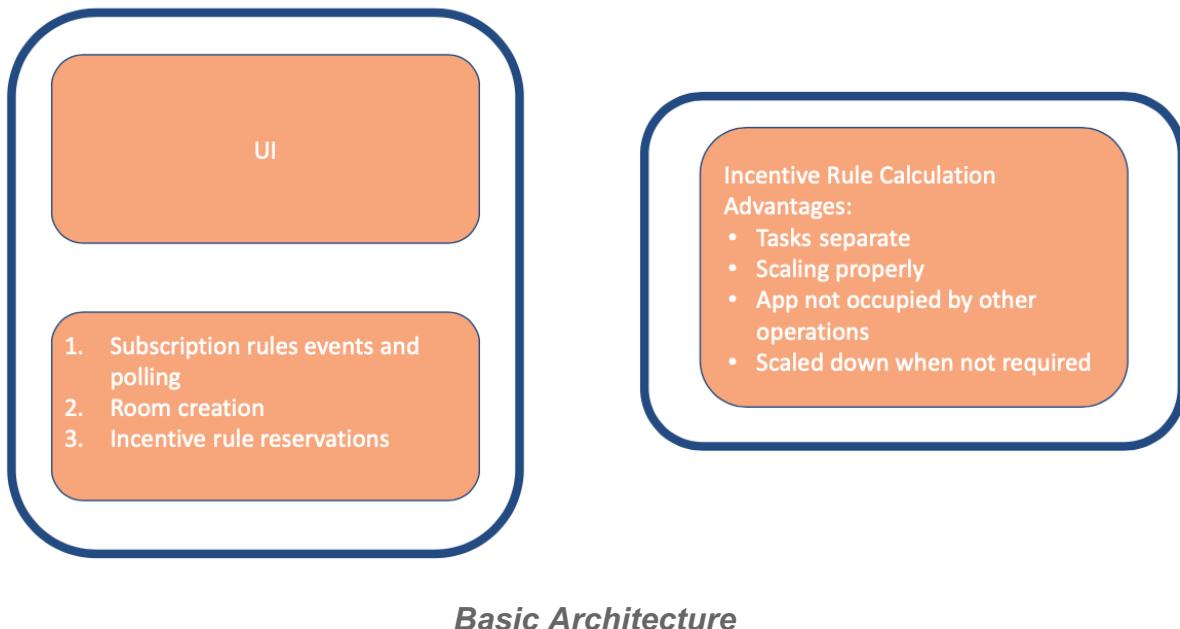
Room Owner: These actors are usually Faculty who can claim ownership of rooms. Like a professor can own their office. These have more access than normal users but have less access than an admin. They can create rules for their rooms and define the incentive points that need to be given to users who follow the rules.

Each actor can interact with the system via a set of tasks as depicted in the diagram below, and each actor will be notified via an email by the notification system.

*Use Case Diagram*

b. Basic architecture

Our application is primarily divided into two separate components. As seen in the diagram below, the left component houses the application UI, the backend server which exposes the REST APIs for the UI and the room reservation/subscription piece. The right component houses the Incentives calculation along with the rules engine, which handles the distribution of incentives to users who have followed the room rules in the previous time window. It is a scheduled job which we plan to run every night when the load on the backend server from user activity is minimal. Since the running the rule engine is a resource incentive process, we kept it separate to allow parallel processing and scalability. It will be able to use the application and UI component resources at night. Since it will run at night when there is low user activity, it will not slow down the regular use of the application for the users during the day.

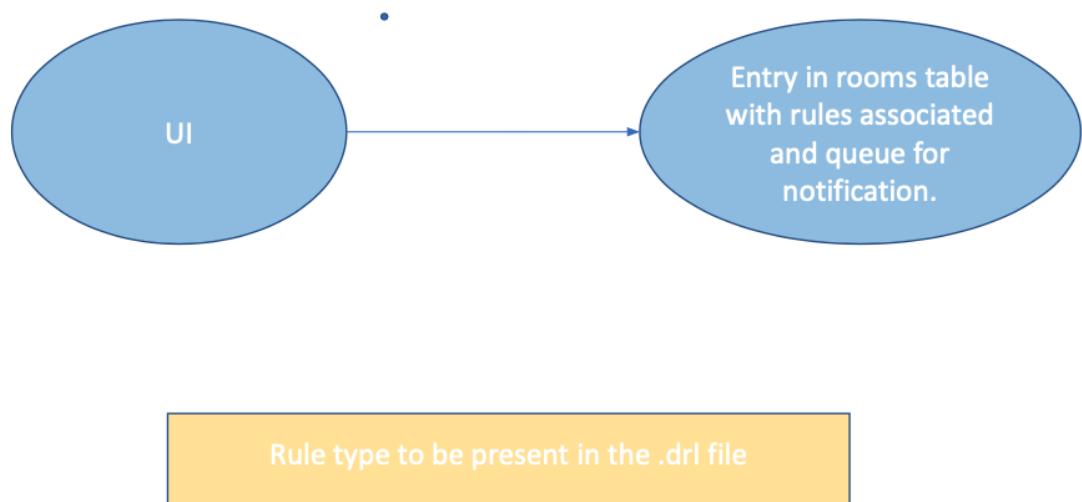


C. System Flows

i. Rule creation flow

This flow encompasses the process of the user going to the application UI

and creates a room for one of the available tippersDB spaces. Each user has an associated access level which can be one of these: Admin, Student or Faculty. The user can only create a room they have either the Admin or Faculty access. The user can add rules for the room which they are creating by selecting the parameters from the UI. Once the user clicks the Create! Button on the UI, the system creates an application-level room with a unique id and connects it to the TippersDB space ID. It also generates the rules for the room and saves the mapping as a JSON dump in the room table in the database. Further an entry is made in the Owner table containing the userId and the newly generated room ID to maintain the ownership of the creator.

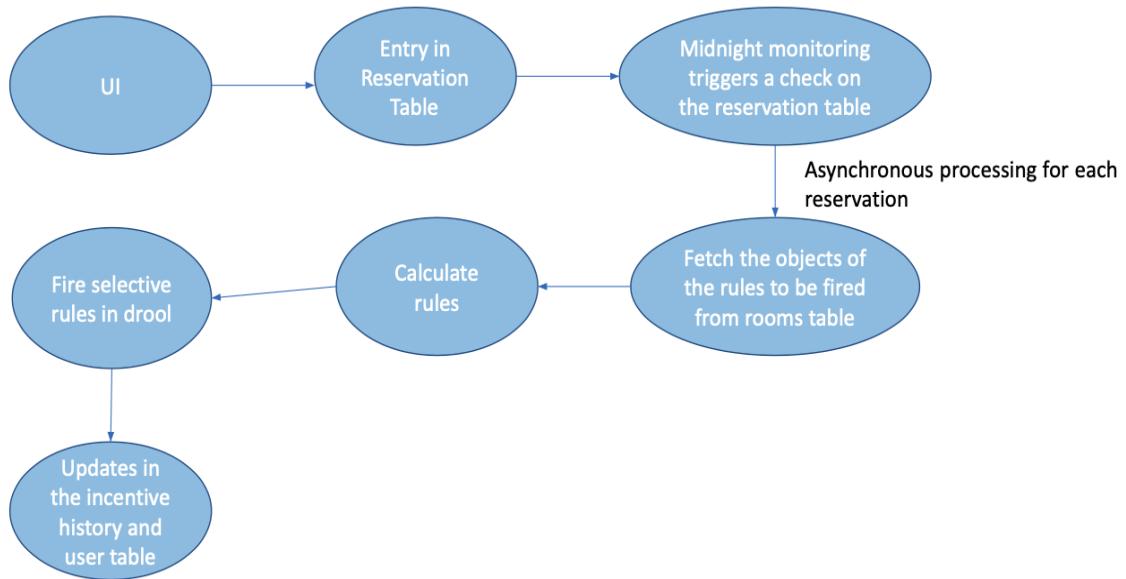


Flow Diagram 1

ii. Incentive Calculation flow

The incentive calculator flow deals with how the incentives are calculated by the application and provided to the relevant user. The user reserves a room from the application UI. At night, the regular incentives calculation job runs which checks where the user followed the reservation or not. If the user was present at the reserved room for the reserved slot the job then provides the user with a predefined number of incentive points. The rooms for which the reservation was made, would have rules associated with it, so the rules engine will calculate the

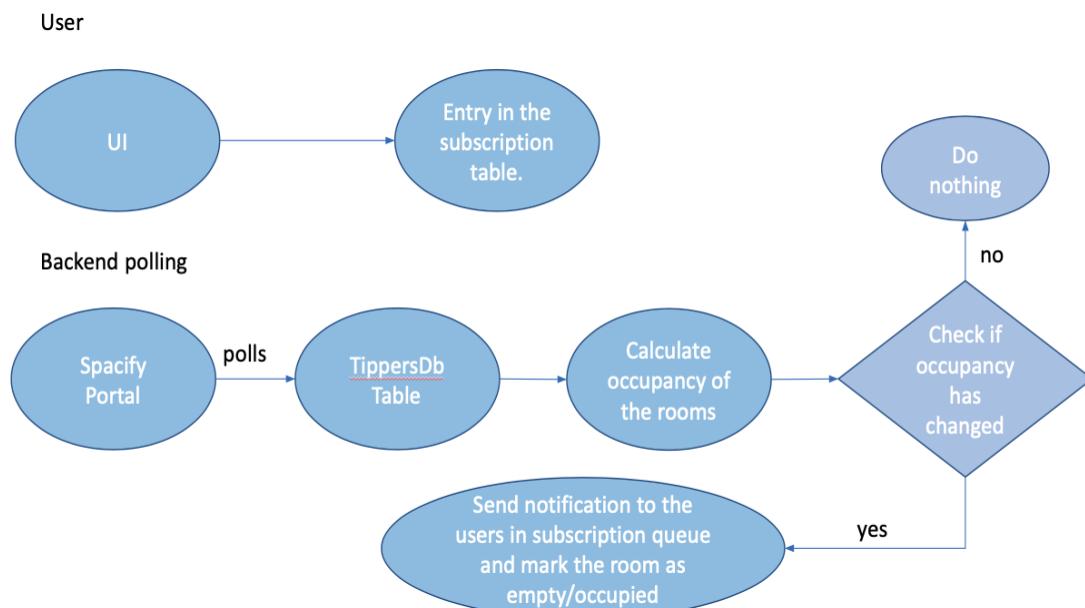
rules and then fire the selected rules from the drool file. After this is done, we update the incentive table for the user with the calculated incentive scores generated by each fired rule. And lastly, we update the total incentive points in the user table for that user.



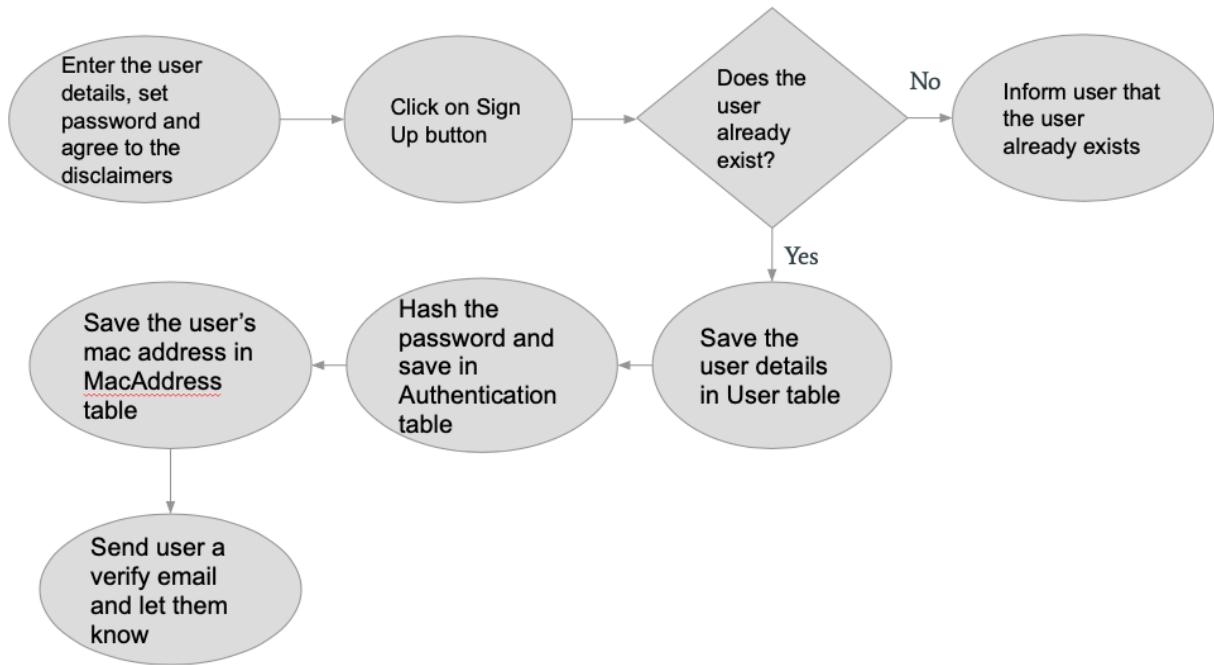
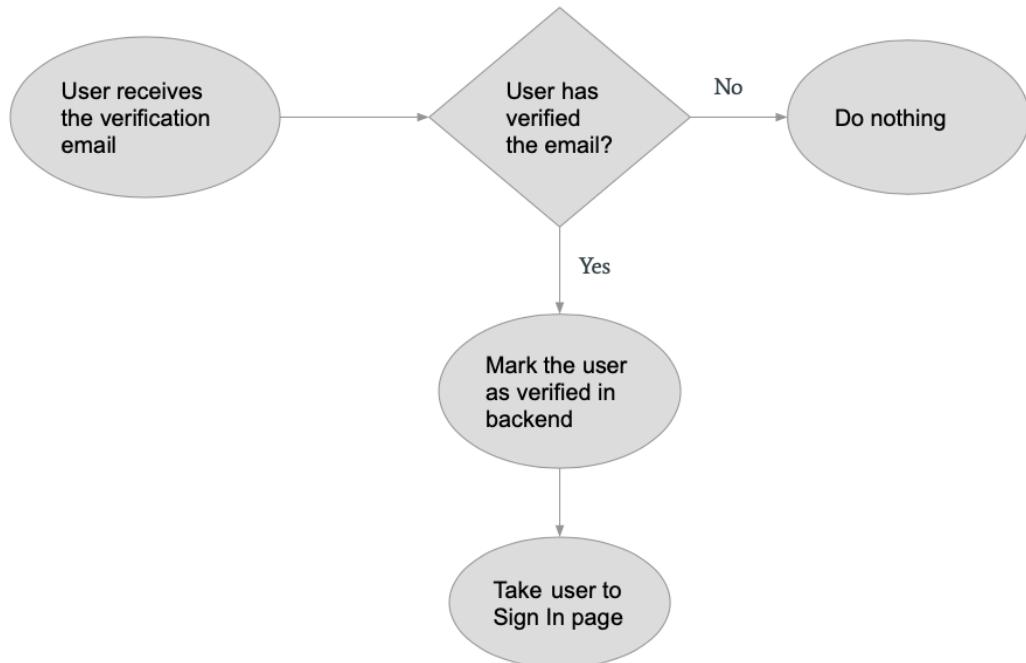
Flow Diagram 2

iii. Subscription flow

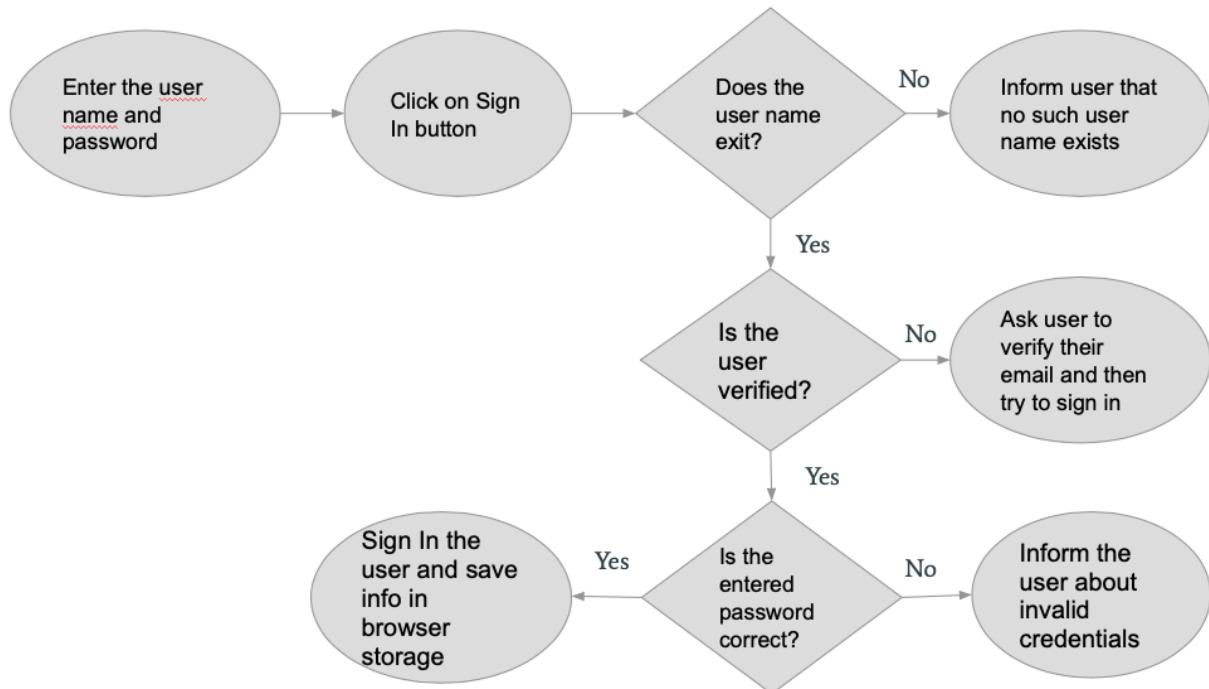
The final flow is the subscription flow. Firstly, the user subscribes to a particular room. This is done so that the user will get to know when the room becomes empty. Since the user has now subscribed to a particular room, the user is added to a subscription queue for that room. The application now will regularly poll for the occupancy of the room. Once the system identifies a change in the room occupancy, the system notifies the user in the front of the subscription queue via an automated mail and we mark the room as occupied. If the user does not enter the room in the coming half an hour, we again mark the room as unoccupied and notify the next user in line.



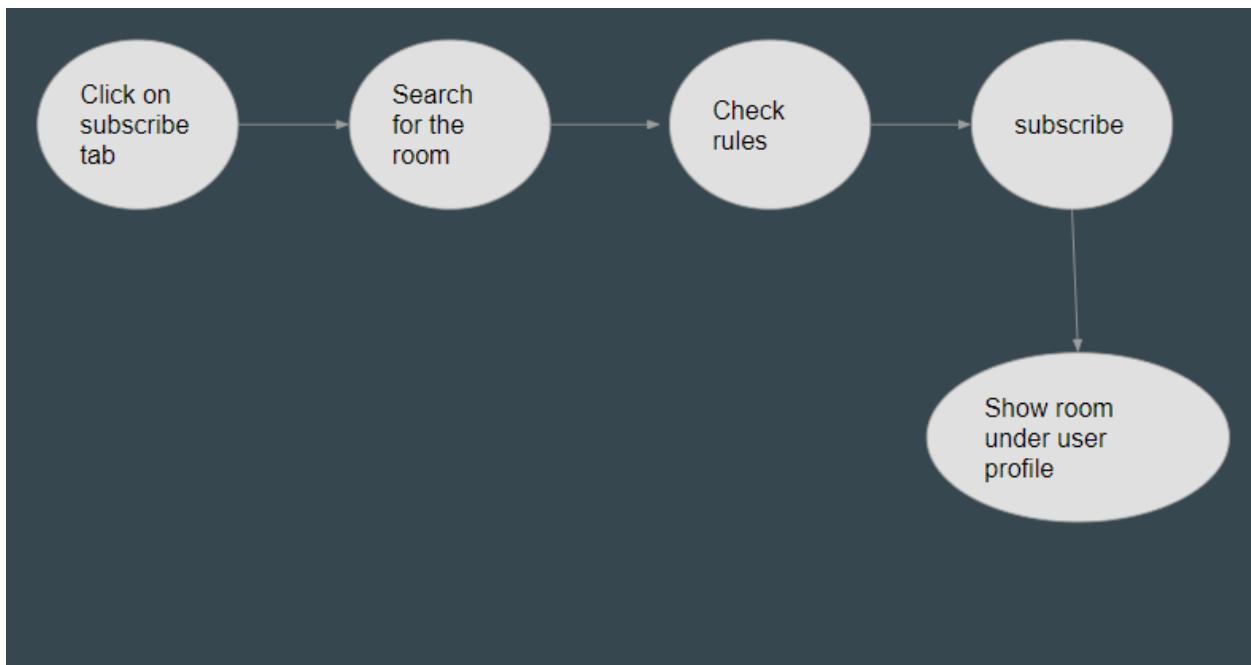
Flow Diagram 3

iv. Sign Up flowv. Sign Up Verification flow

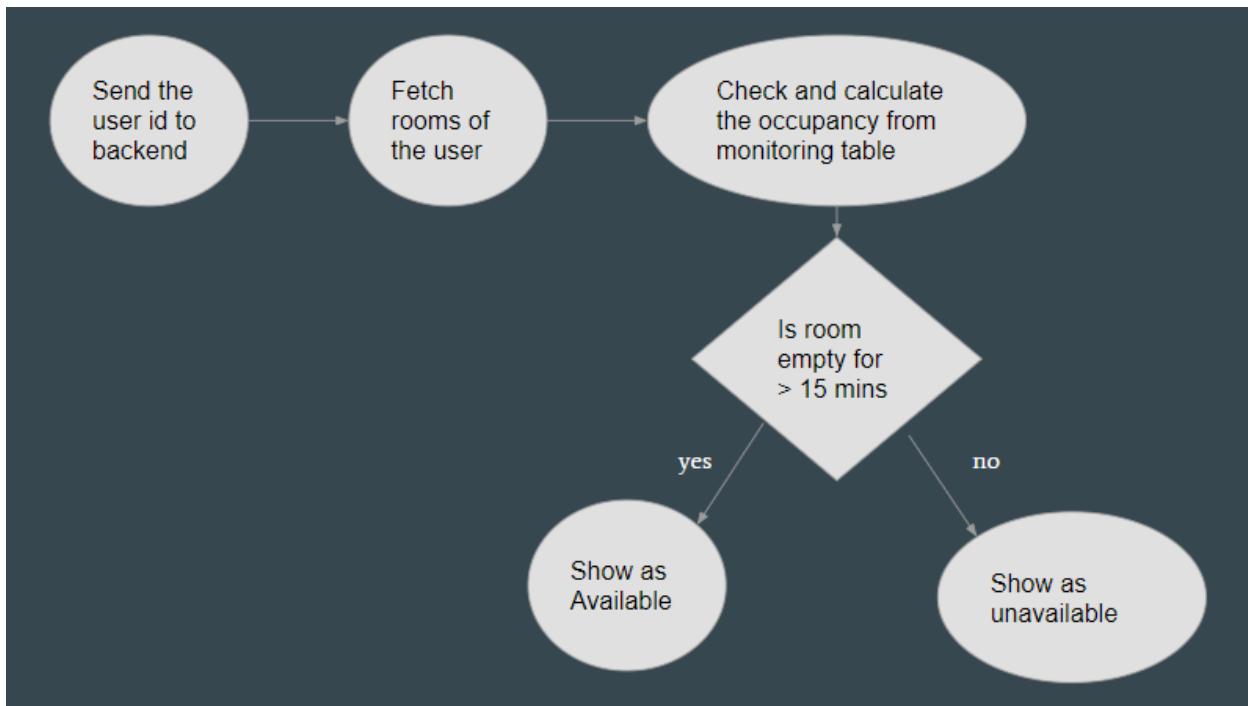
vi. Sign In flow



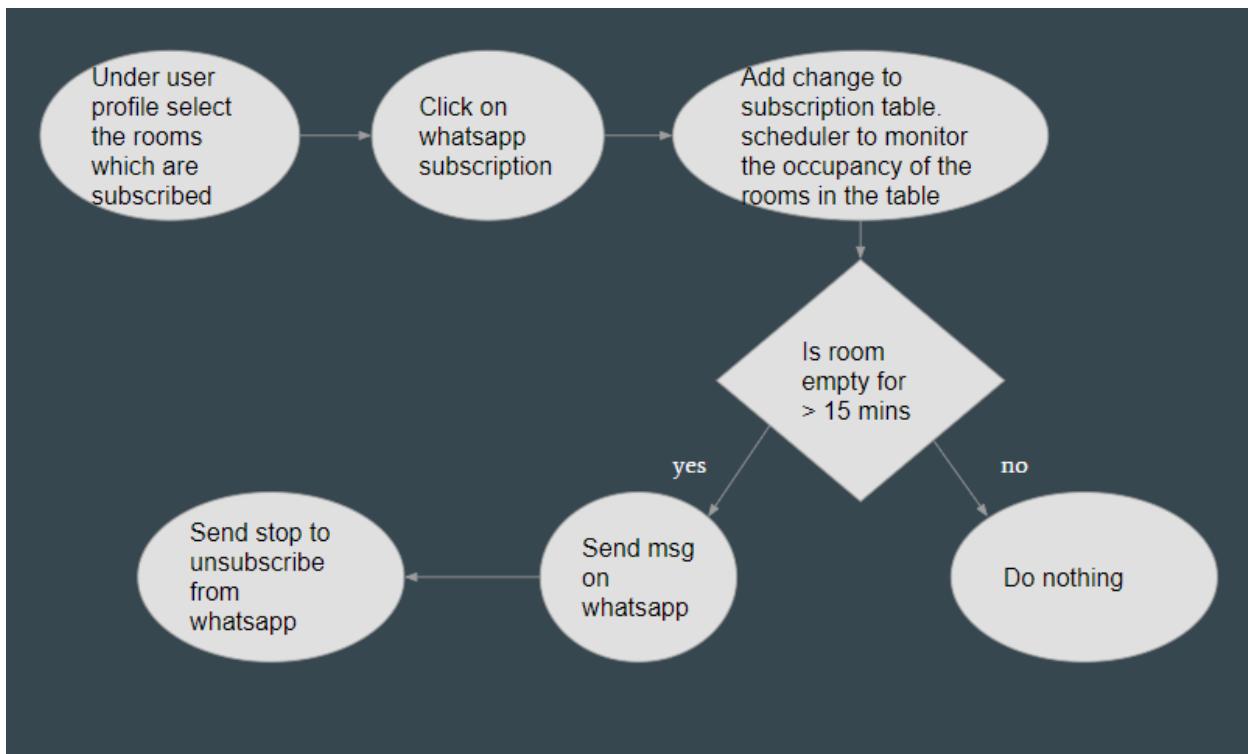
vii. Subscription flow from user perspective (new)



viii Notification flow part 1 (new)



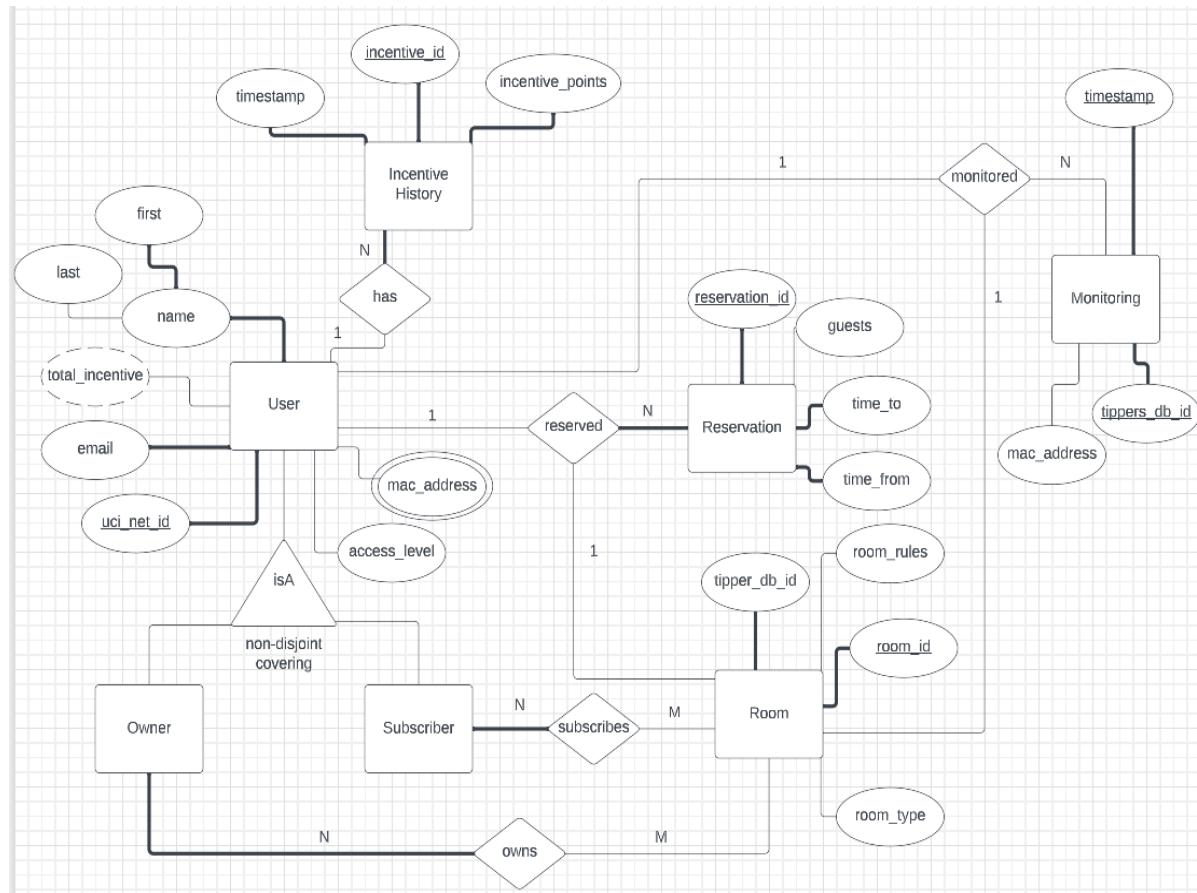
ix Notification flow part 2 (new)



d. Schema Design (ER diagram)

There are a total of 7 main entities and they are connected to each other with the relationships shown in the below diagram:

1. User
2. Owner
3. Subscriber
4. Room
5. Reservation
6. Monitoring
7. Incentive



e. Module Descriptions (update)

The project has three separate modules:

1. Spacify-engine: This module contains the rules engine and incentive calculation flow for the application.
2. Spacify-portal: This module contains the application server, authentication flow, the reservation system and the subscriber flow.
3. Specify-lib: Houses the common components and flow like entities, repositories which are used for both Spacify-engine and Spacify-portal
4. Spacify-ui: This module contains the UI for the application and various components used by it.

f. Test Plan (update)

Once the flow pipeline is implemented, we are planning to test by creating two rules and by using a few initial users. The two rules will be: 1. To be able to set the maximum occupancy of the room and 2. To be able to control who is not allowed in the room. Further we have created around 10 users out of which few have student access level, few have faculty access level and few have admin access level. These users will reserve rooms, create rooms with either of the two available rules or both rules and the rules engine will then calculate the incentives for other users who follow these rules. The incentive calculation flow will kick in and the incentives will be stored in the application database. Test the subscription flow once fully developed for a few users and for different rooms. Test the signIn/ singUp feature for various users on different browsers such as Chrome, Safari and Mozilla. Finally, we plan to test the modify reservation and create new reservation modules.

Relevant Links

Github Repository: <https://github.com/virajdesaiofficial/Spacify1.0>

Trello Link: <https://trello.com/b/y2lRqvJO/scrum-board>

Shared Google Drive:

https://drive.google.com/drive/u/1/folders/1k4_xDzWaNnSIXNC6ApkhHt4OZTzKiy4q