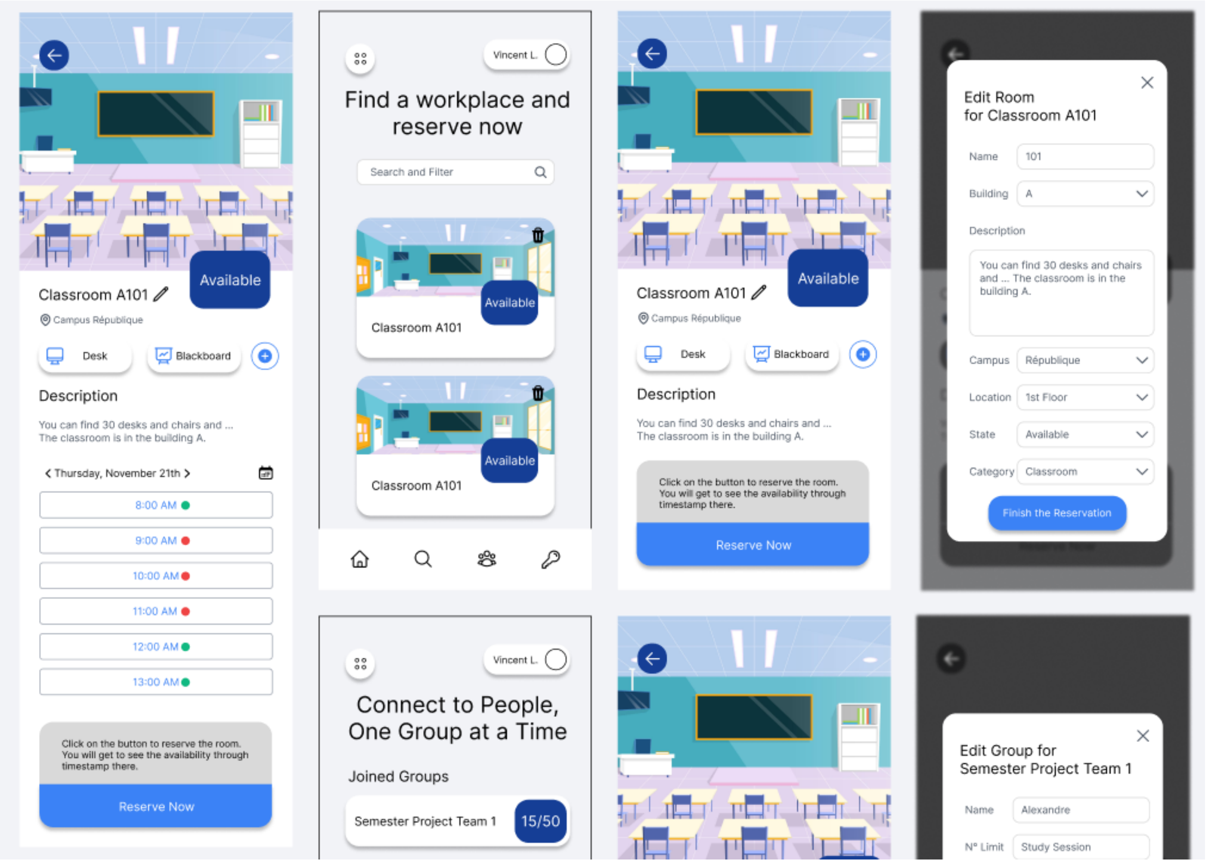
UML Project : EmptyClassroom



Find a workplace and reserve now

You can find 30 desks and chairs and ... The classroom is in the building A.

Click on the button to reserve the room. You will get to see the availability through timestamp there.

Connect to People, One Group at a Time

Edit Room for Classroom A101

Name: 101
Building: A
Description: You can find 30 desks and chairs and ... The classroom is in the building A.
Campus: République
Location: 1st Floor
State: Available
Category: Classroom

Edit Group for Semester Project Team 1

Name: Alexandre
N° Limit: Study Session

Alexandre CHRISTINA & Vincent LAM

2023-2024
To the attention of Mr. SZABÓ-RESCH Zsolt
EFREI, L3

Table of Content

Project Description

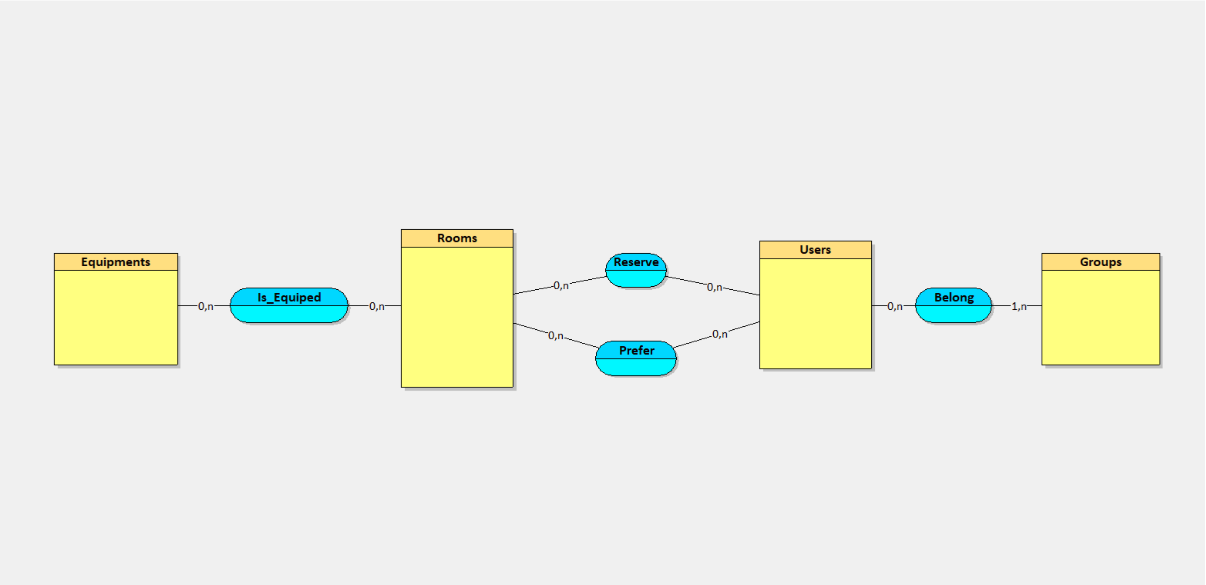
We're going to take the topic of the web development project. That said, we're going to add a few extra features to simulate a more complex project. Here's the description of the web project we wrote previously:

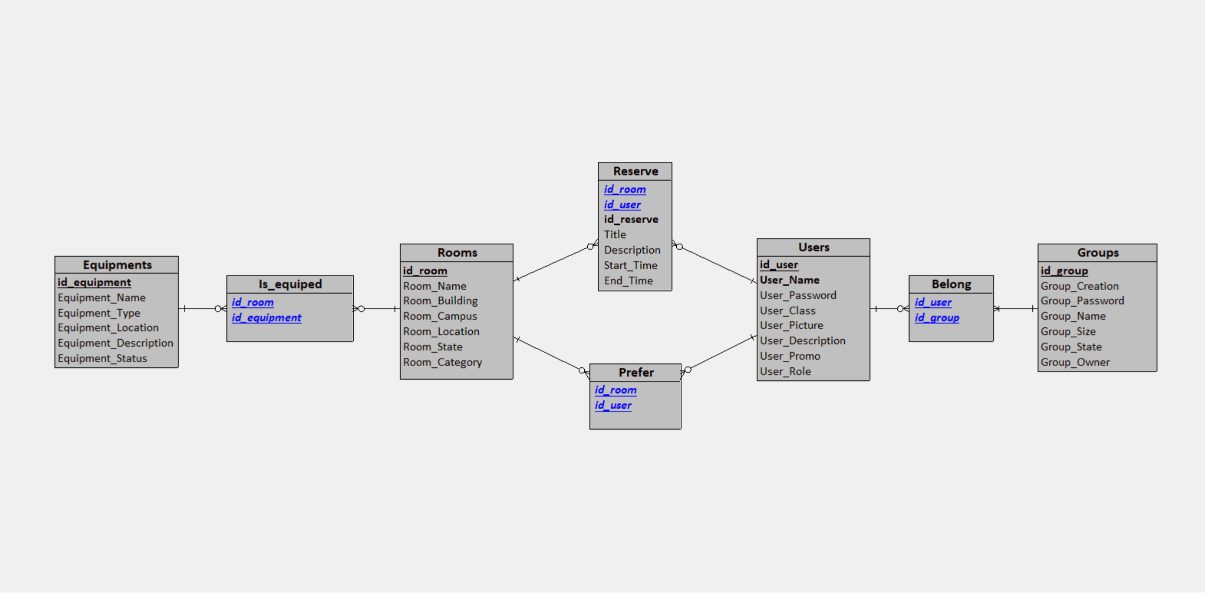
> “For this project, we made the choice to build a useful web application called “Empty Classroom”. The main goal of this project is to provide for students a means to easily find and reserve empty classrooms they can use for study or any kind of school related events. Our application will let each student create an account, login and access an interface from which they will be able to view a list of all available rooms of each campus of the school, reserve it for themselves or for a group they would have previously created from their dashboard.“

Required Diagrams at the beginning

ER & Table Structure Diagram for the database

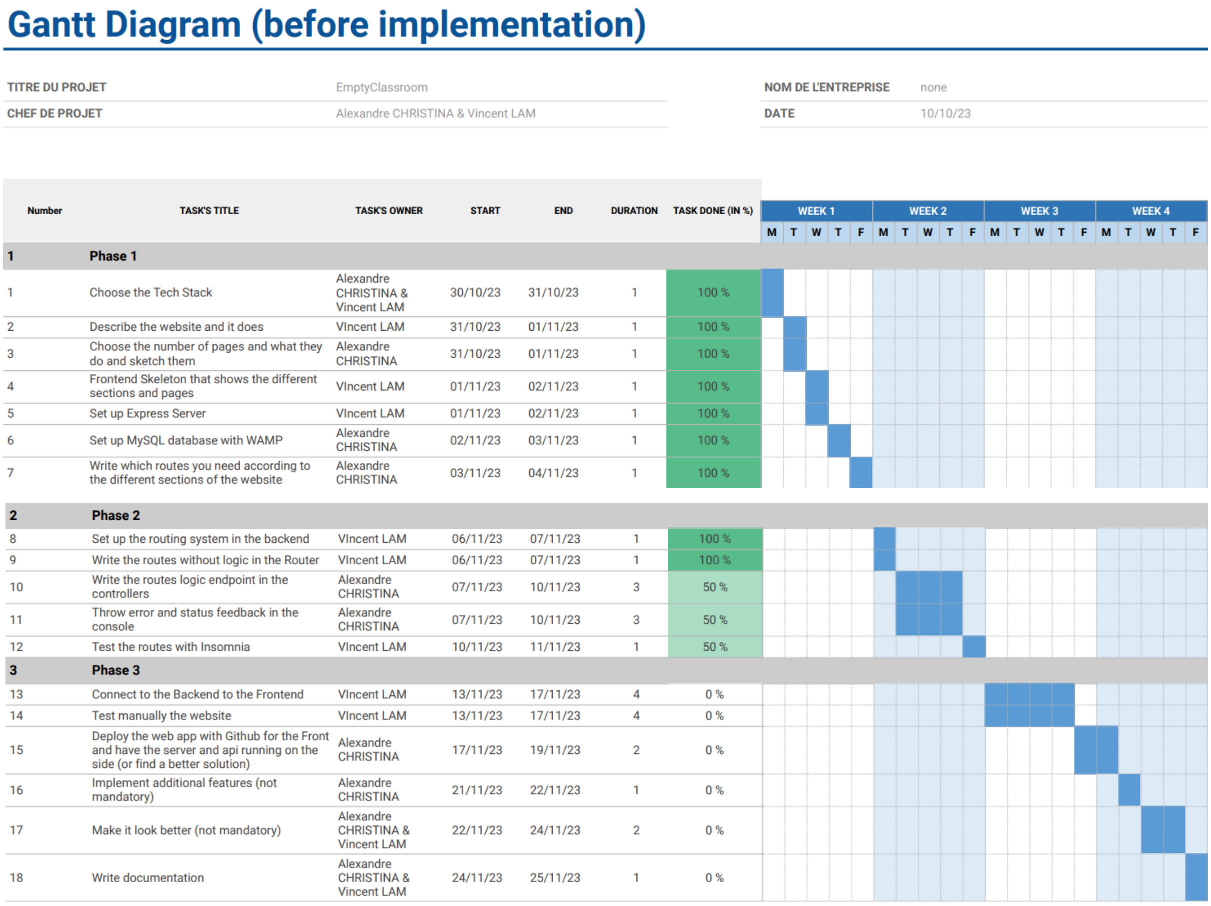
To make this project possible, we need the four tables that you can see below.





Gantt Diagram (planned)

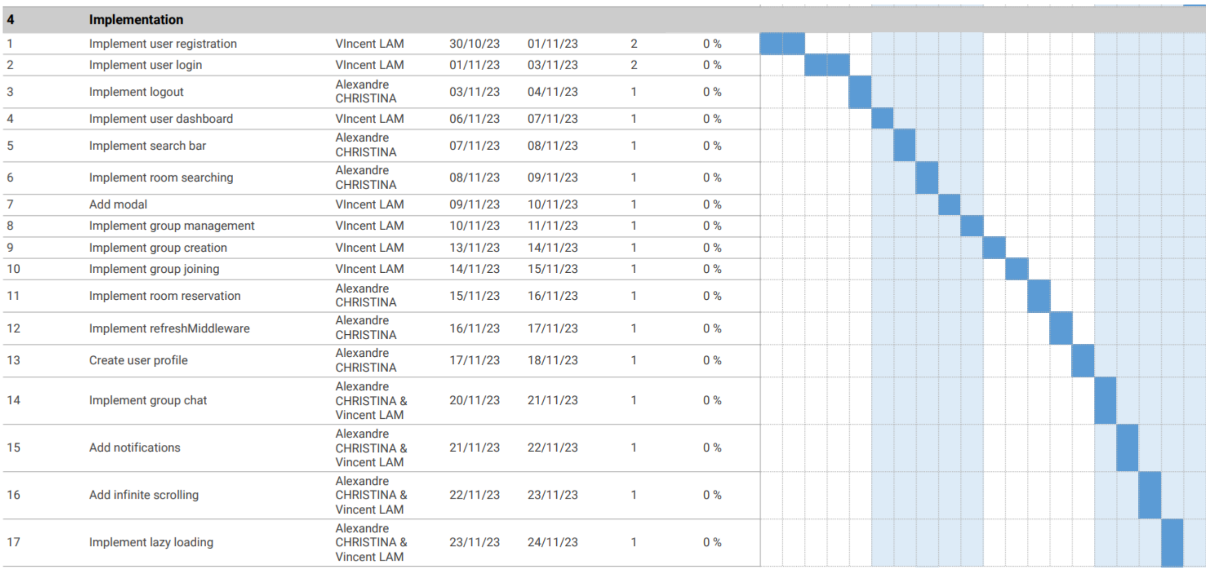
We have written three phases to complete the project and an Implementation part to track the progress of the different features. We put the number, title, task's owner and date in the diagram.



Gantt Diagram (before implementation)

TITRE DU PROJET	EmptyClassroom	NOM DE L'ENTREPRISE	none
CHEF DE PROJET	Alexandre CHRISTINA & Vincent LAM	DATE	10/10/23

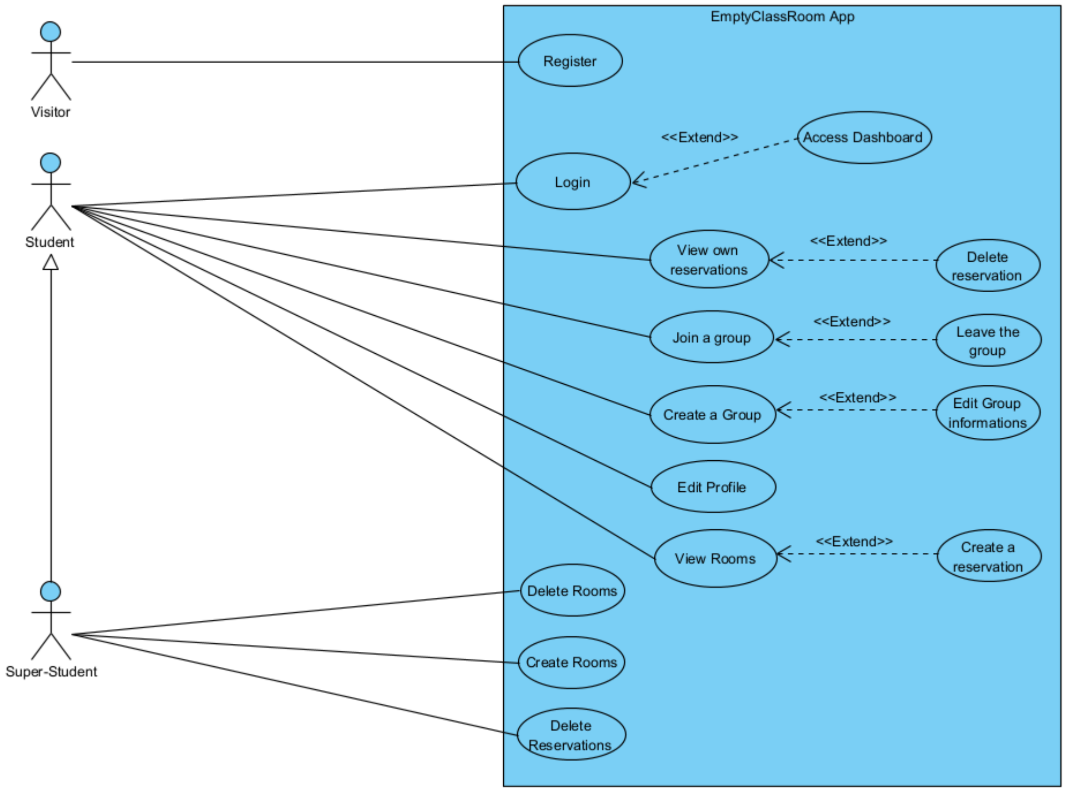
Number	TASK'S TITLE	TASK'S OWNER	START	END	DURATION	TASK DONE (IN %)	WEEK 1					WEEK 2					WEEK 3					WEEK 4				
							M	T	W	T	F	M	T	W	T	F	M	T	W	T	F	M	T	W	T	F
1	Phase 1																									
1	Choose the Tech Stack	Alexandre CHRISTINA & Vincent LAM	30/10/23	31/10/23	1	100 %	■																			
2	Describe the website and it does	VIncent LAM	31/10/23	01/11/23	1	100 %		■																		
3	Choose the number of pages and what they do and sketch them	Alexandre CHRISTINA	31/10/23	01/11/23	1	100 %		■																		
4	Frontend Skeleton that shows the different sections and pages	VIncent LAM	01/11/23	02/11/23	1	100 %			■																	
5	Set up Express Server	VIncent LAM	01/11/23	02/11/23	1	100 %			■																	
6	Set up MySQL database with WAMP	Alexandre CHRISTINA	02/11/23	03/11/23	1	100 %				■																
7	Write which routes you need according to the different sections of the website	Alexandre CHRISTINA	03/11/23	04/11/23	1	100 %					■															
2	Phase 2																									
8	Set up the routing system in the backend	VIncent LAM	06/11/23	07/11/23	1	100 %						■														
9	Write the routes without logic in the Router	VIncent LAM	06/11/23	07/11/23	1	100 %						■														
10	Write the routes logic endpoint in the controllers	Alexandre CHRISTINA	07/11/23	10/11/23	3	50 %							■	■	■											
11	Throw error and status feedback in the console	Alexandre CHRISTINA	07/11/23	10/11/23	3	50 %							■	■	■											
12	Test the routes with Insomnia	VIncent LAM	10/11/23	11/11/23	1	50 %										■										
3	Phase 3																									
13	Connect to the Backend to the Frontend	VIncent LAM	13/11/23	17/11/23	4	0 %											■	■	■	■						
14	Test manually the website	VIncent LAM	13/11/23	17/11/23	4	0 %											■	■	■	■						
15	Deploy the web app with Github for the Front and have the server and api running on the side (or find a better solution)	Alexandre CHRISTINA	17/11/23	19/11/23	2	0 %															■	■				
16	Implement additional features (not mandatory)	Alexandre CHRISTINA	21/11/23	22/11/23	1	0 %																	■			
17	Make it look better (not mandatory)	Alexandre CHRISTINA & Vincent LAM	22/11/23	24/11/23	2	0 %																		■	■	
18	Write documentation	Alexandre CHRISTINA & Vincent LAM	24/11/23	25/11/23	1	0 %																				■



4	**Implementation**					
1	Implement user registration	VIncent LAM	30/10/23	01/11/23	2	0 %
2	Implement user login	VIncent LAM	01/11/23	03/11/23	2	0 %
3	Implement logout	Alexandre CHRISTINA	03/11/23	04/11/23	1	0 %
4	Implement user dashboard	VIncent LAM	06/11/23	07/11/23	1	0 %
5	Implement search bar	Alexandre CHRISTINA	07/11/23	08/11/23	1	0 %
6	Implement room searching	Alexandre CHRISTINA	08/11/23	09/11/23	1	0 %
7	Add modal	VIncent LAM	09/11/23	10/11/23	1	0 %
8	Implement group management	VIncent LAM	10/11/23	11/11/23	1	0 %
9	Implement group creation	VIncent LAM	13/11/23	14/11/23	1	0 %
10	Implement group joining	VIncent LAM	14/11/23	15/11/23	1	0 %
11	Implement room reservation	Alexandre CHRISTINA	15/11/23	16/11/23	1	0 %
12	Implement refreshMiddleware	Alexandre CHRISTINA	16/11/23	17/11/23	1	0 %
13	Create user profile	Alexandre CHRISTINA	17/11/23	18/11/23	1	0 %
14	Implement group chat	Alexandre CHRISTINA & Vincent LAM	20/11/23	21/11/23	1	0 %
15	Add notifications	Alexandre CHRISTINA & Vincent LAM	21/11/23	22/11/23	1	0 %
16	Add infinite scrolling	Alexandre CHRISTINA & Vincent LAM	22/11/23	23/11/23	1	0 %
17	Implement lazy loading	Alexandre CHRISTINA & Vincent LAM	23/11/23	24/11/23	1	0 %

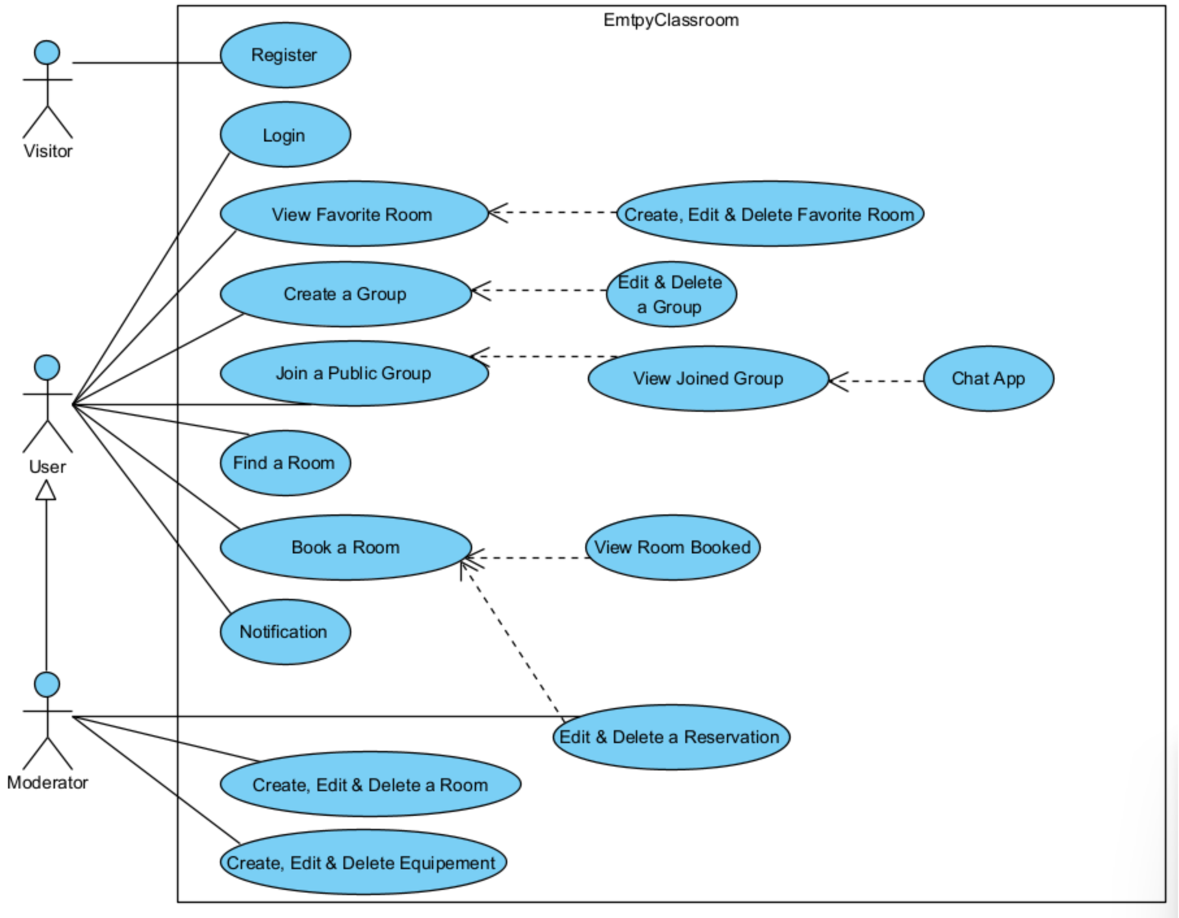
Main Diagrams (one per member)

Use Case Diagrams (first Alexandre & second Vincent)



This use case diagram was made to represent the main interactions of the different users of our application in its current state. In this diagram we took the cases of three different actors ; **the visitor** which represents anybody who visits the website without an account, **the student** which is for students already having an account, and **the super-student** which represents the site super-users with special accounts. We also added the main use cases and the interactions between them and the users. As an example, we can see that ; the super-student inherits from the student, because they can do everything a student does.

For the second **use case diagram**, It's basically the same idea except there is a chat app and a notification feature.



Activity Diagrams

Explanation for the General Activity Diagram (Alexandre)

Access User Dashboard:
The process starts with the user accessing the login page.
The system checks the login credentials.
If invalid, the user is directed back to the login process.
If valid, the user proceeds to the dashboard.

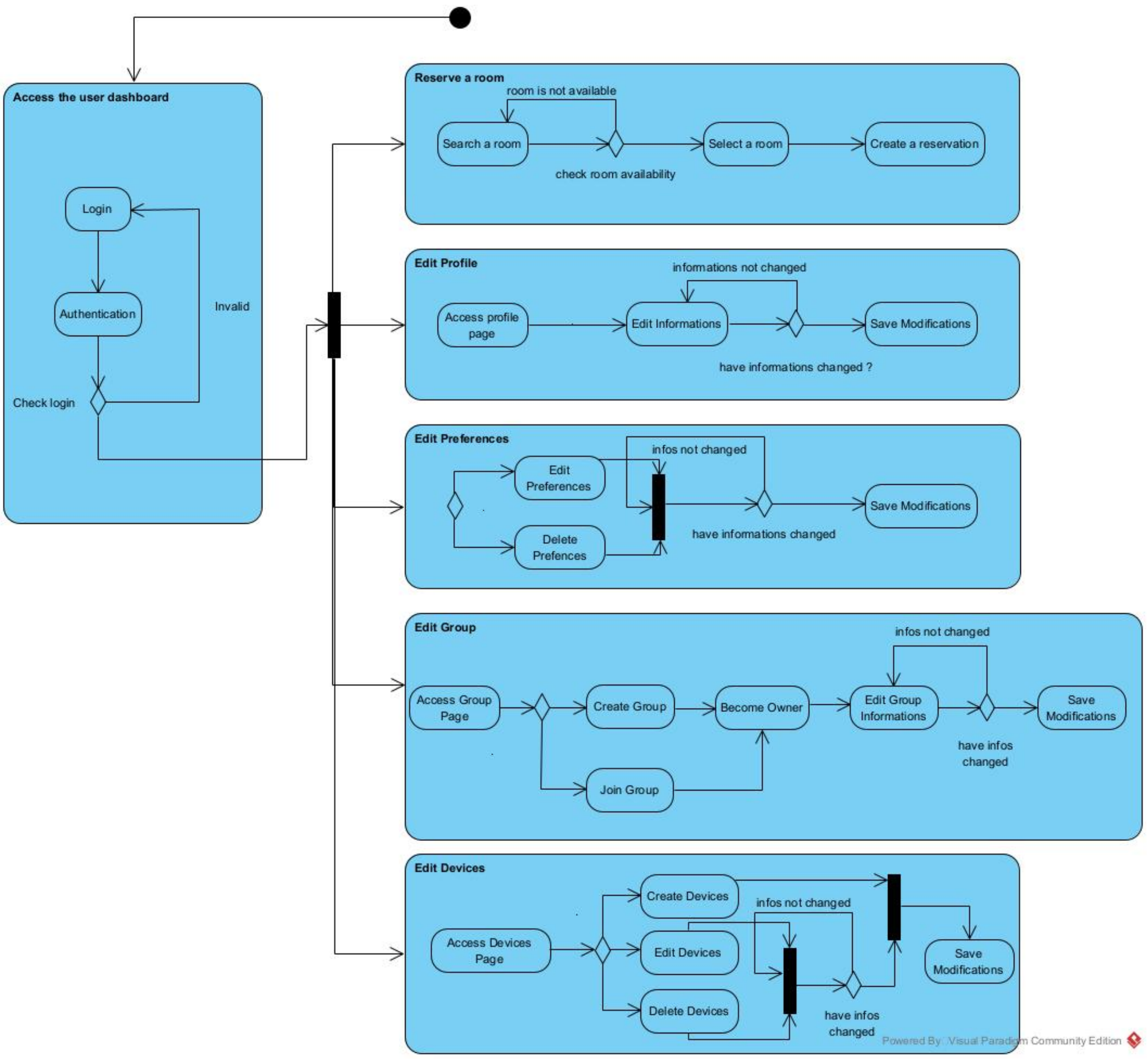
Reserve a Room:
User initiates the room reservation process.
The system allows the user to search for available rooms.
Room availability is checked.
If not available, the user is directed back to the room search.
If available, the user selects a room and creates a reservation.

Edit Profile:
User accesses the profile page.
Edits personal information.
The system checks if any information has changed.
If changed, modifications are saved.
If not changed, the user returns to the edit information step.

Edit Preferences:
Users have two choices: edit preferences or delete preferences.
If preferences are edited, the system saves modifications.
If preferences are not edited, the user goes back to the edit preferences step.

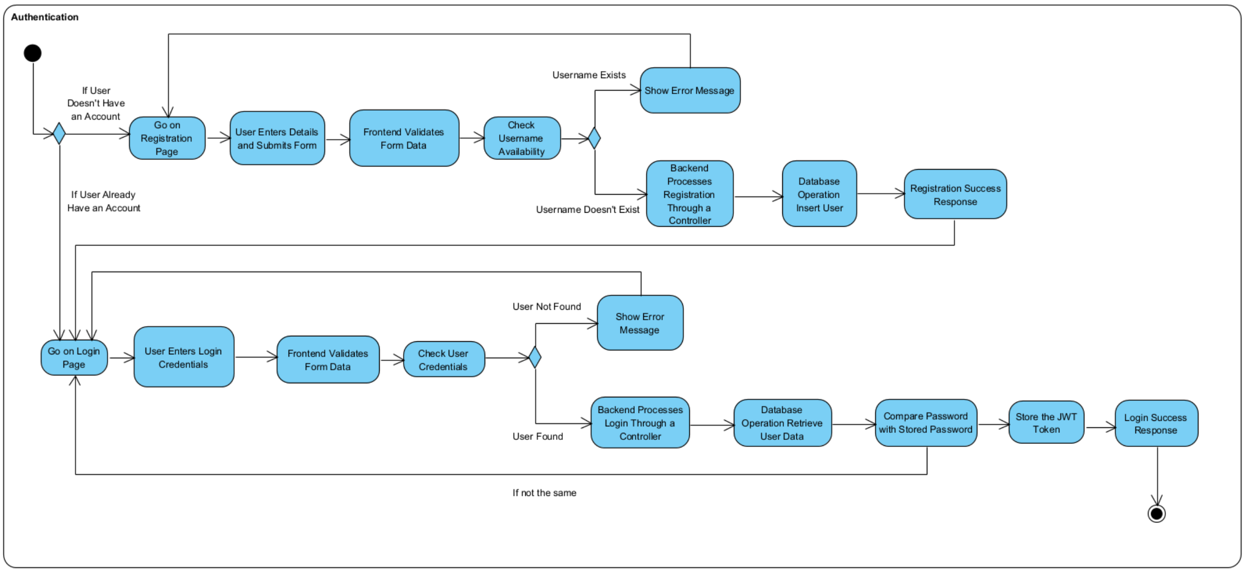
Edit Group:
User accesses the group page.
Two choices: create a group or join a group.
A member can become the owner and the owner can edit group information.
If group information is changed, modifications are saved.
If not changed, the user goes back to the edit group step.

Edit Devices:
User accesses the devices page.
Three choices: create, edit or delete devices.
If any modifications are made, the system saves the information.
If no modifications are made, the user goes back to the three choices.



Explanation for Authentication Activity Diagram (Vincent)

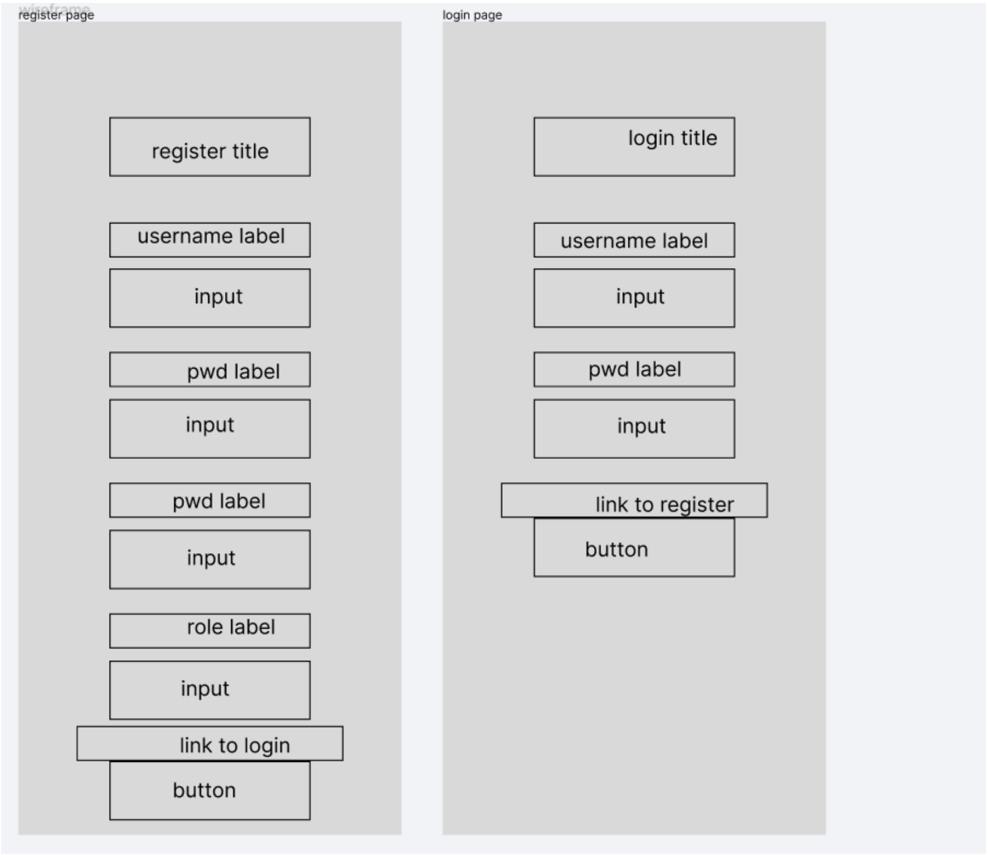
The user starts the authentication process by choosing between registration and login. Depending on the choice, the corresponding activity (register or login) is initiated. The process continues with the user entering data and submitting the form. Frontend validation and backend processes follow, ensuring username availability for registration or checking user credentials for login. For registration, the user's data is inserted into the database, and a success response is sent. For login, the user's data is retrieved from the database, password comparison is performed, the JWT is forged and stored in the cookies. The login success response concludes the process.


Authentication
If User Doesn't Have an Account
If User Already Have an Account
Go on Registration Page
User Enters Details and Submits Form
Frontend Validates Form Data
Check Username Availability
Username Exists
Show Error Message
Username Doesn't Exist
Backend Processes Registration Through a Controller
Database Operation Insert User
Registration Success Response
Go on Login Page
User Enters Login Credentials
Frontend Validates Form Data
Check User Credentials
User Not Found
Show Error Message
User Found
Backend Processes Login Through a Controller
Database Operation Retrieve User Data
Compare Password with Stored Password
Store the JWT Token
Login Success Response
If not the same

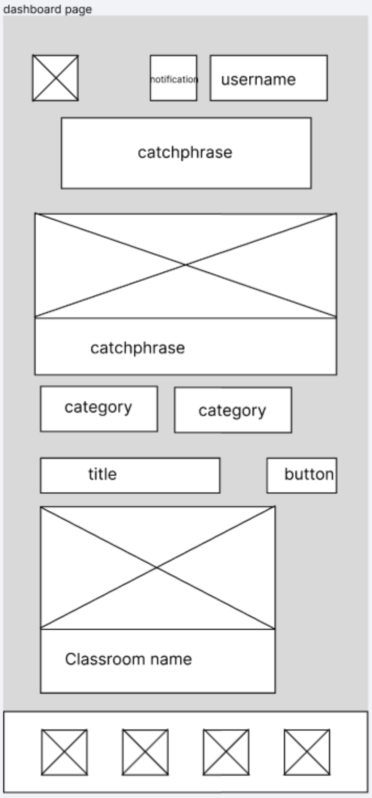
Wireframe and Mockup (we both did it)

There are supposed to be two wireframes but since we both worked on the wireframe through a shared figma document, we didn't split it into two files just to make two files.

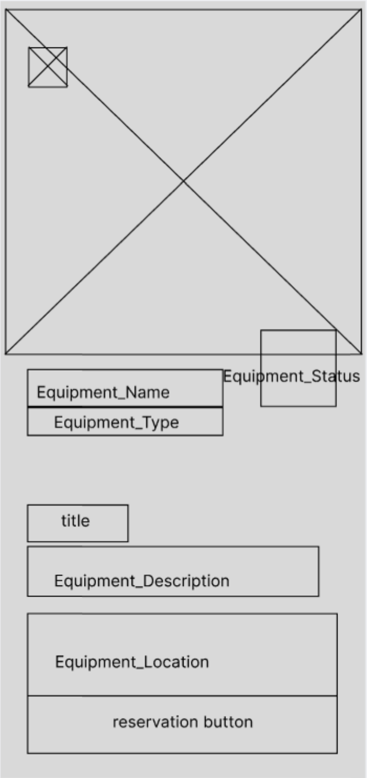
As for the mockup, it was not asked but since the idea of the mockup was explained to us in the course, we did one. It helped us to have a better idea of what we wanted on the pages and allowed us to make the design prettier.



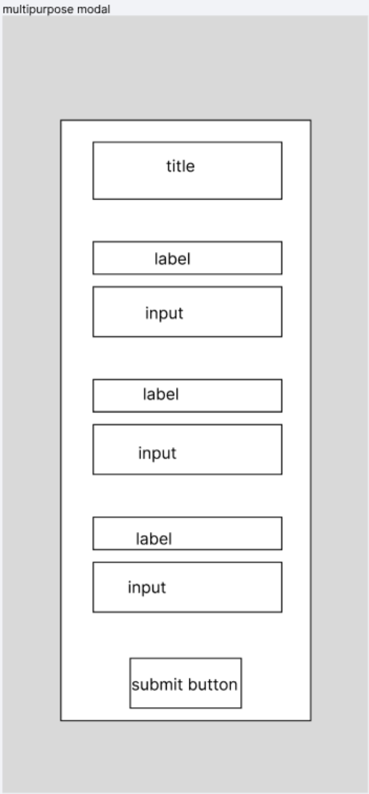
dashboard page



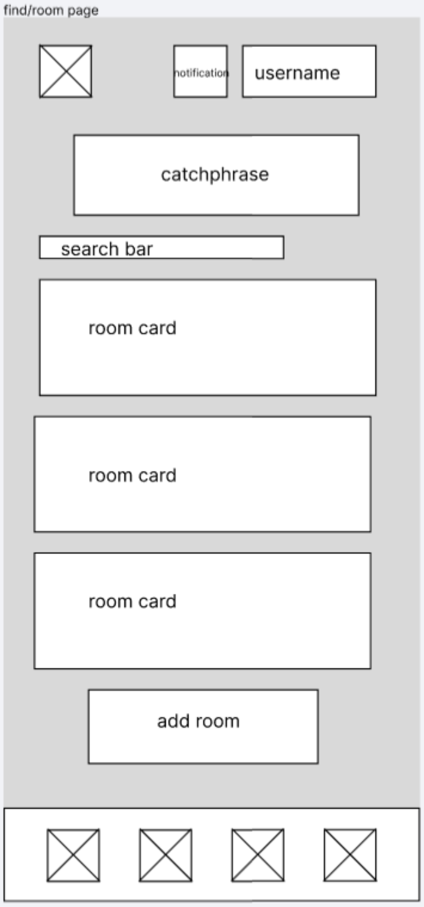
dashboard details



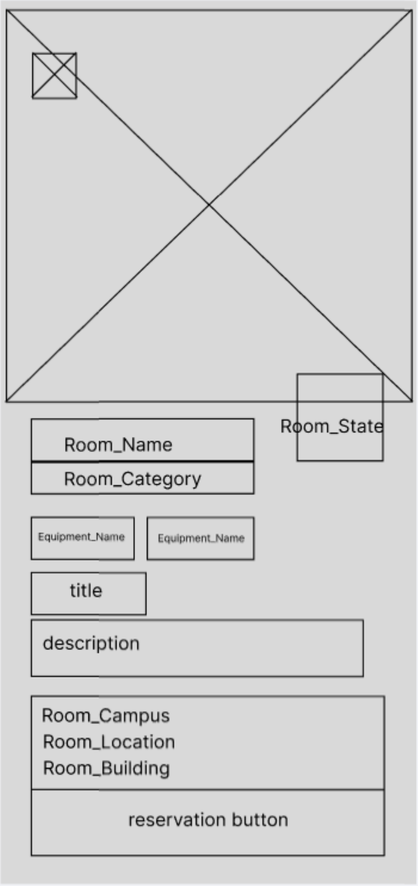
multipurpose modal



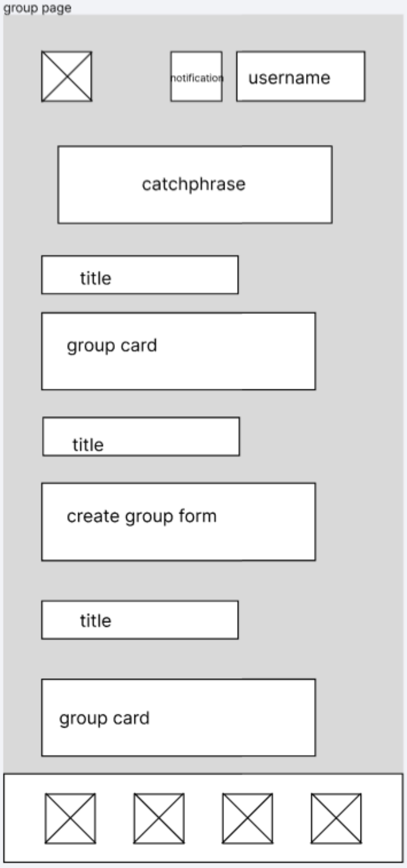
find/room page



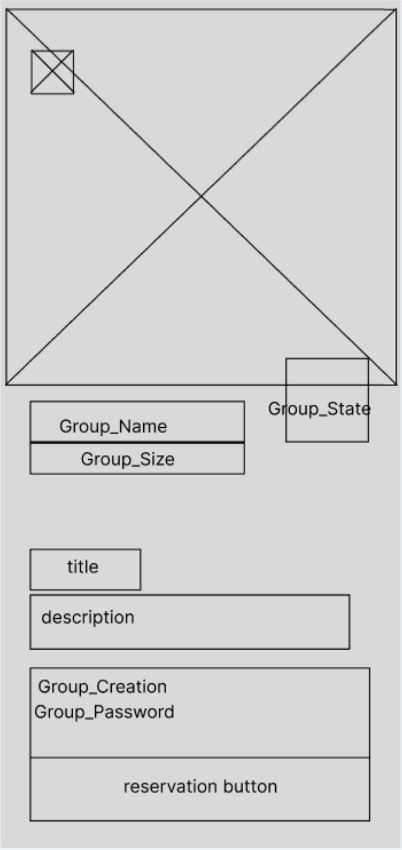
room details



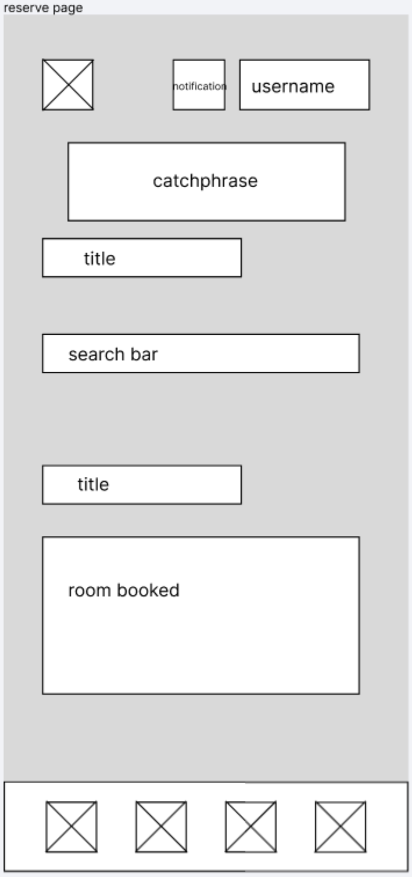
group page



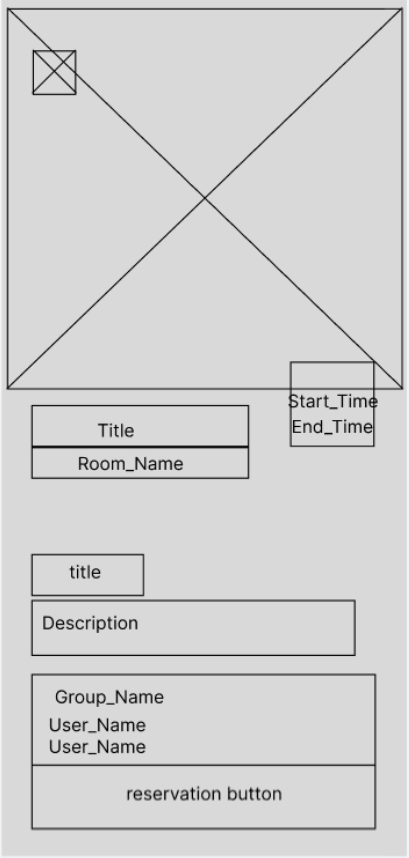
group details

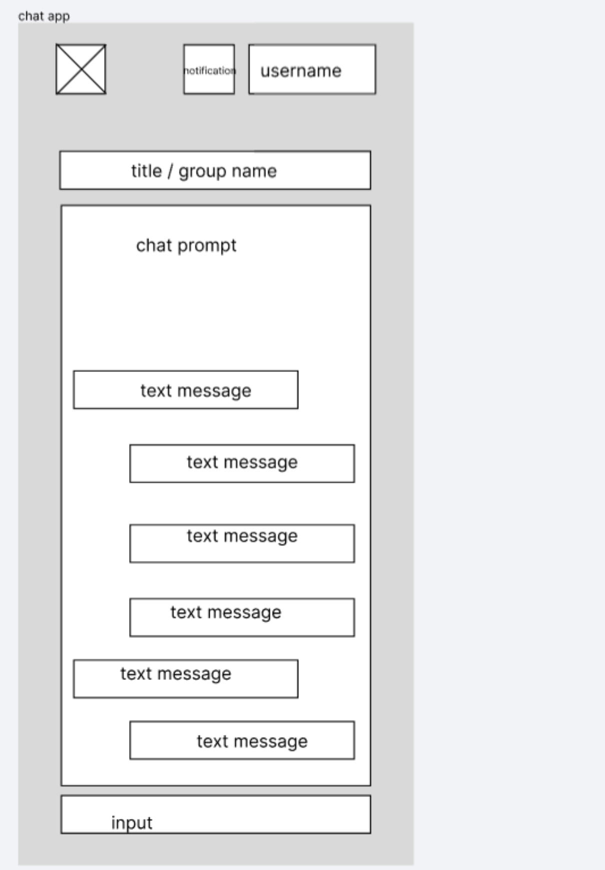


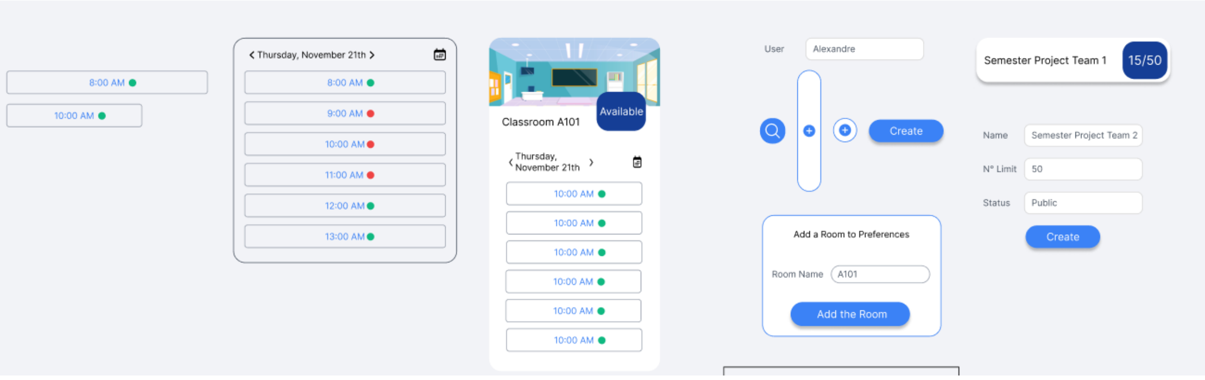
reserve page



reservation details







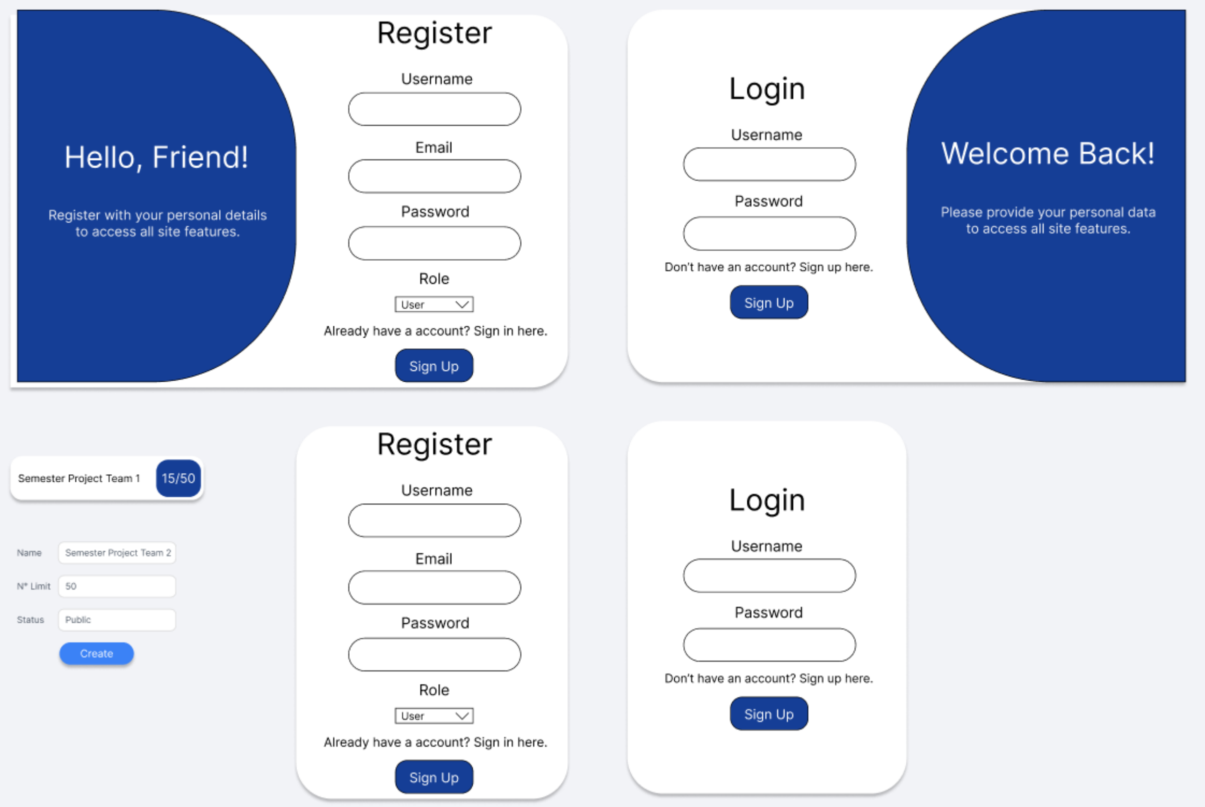
User Alexandre

Room Name A101

Name Semester Project Team 2

N° Limit 50

Status Public



Hello, Friend!

Register with your personal details to access all site features.

Register

Username

Email

Password

Role

User

Already have a account? Sign in here.

Sign Up

Login

Username

Password

Don't have an account? Sign up here.

Sign Up

Welcome Back!

Please provide your personal data to access all site features.

Semester Project Team 1 15/50

Name: Semester Project Team 2

N° Limit: 50

Status: Public

Create

Register

Username

Email

Password

Role

User

Already have a account? Sign in here.

Sign Up

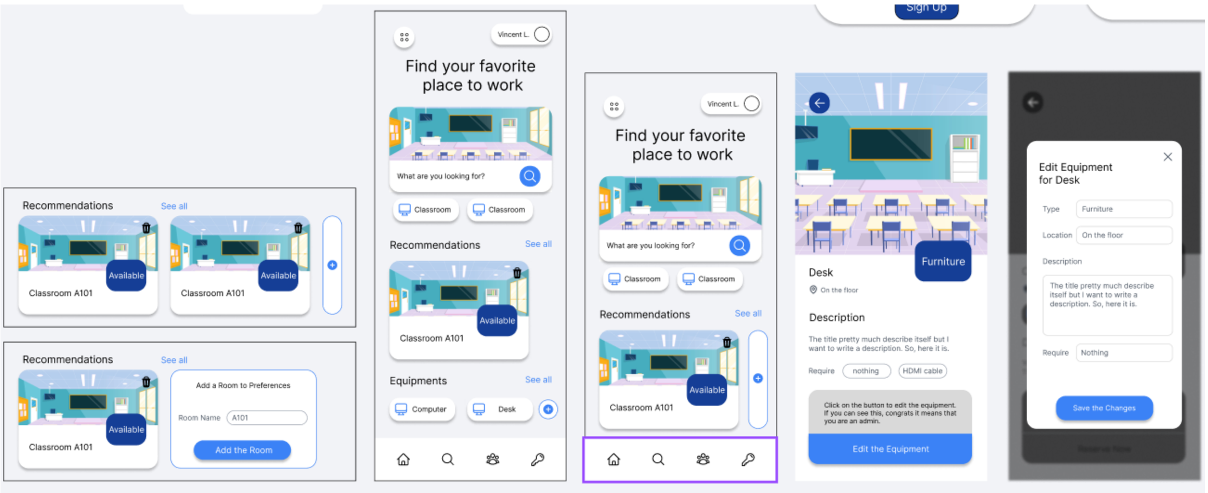
Login

Username

Password

Don't have an account? Sign up here.

Sign Up



Sign Up

Recommendations See all

Classroom A101 Available

Classroom A101 Available

Recommendations See all

Classroom A101 Available

Add a Room to Preferences

Room Name: A101

Add the Room

Vincent L.

Find your favorite place to work

What are you looking for?

Classroom Classroom

Recommendations See all

Classroom A101 Available

Equipments See all

Computer Desk

Vincent L.

Find your favorite place to work

What are you looking for?

Classroom Classroom

Recommendations See all

Classroom A101 Available

Furniture

Desk

On the floor

Description

The title pretty much describe itself but I want to write a description. So, here it is.

Require: nothing, HDMI cable

Click on the button to edit the equipment. If you can see this, congrats it means that you are an admin.

Edit the Equipment

Edit Equipment for Desk

Type: Furniture

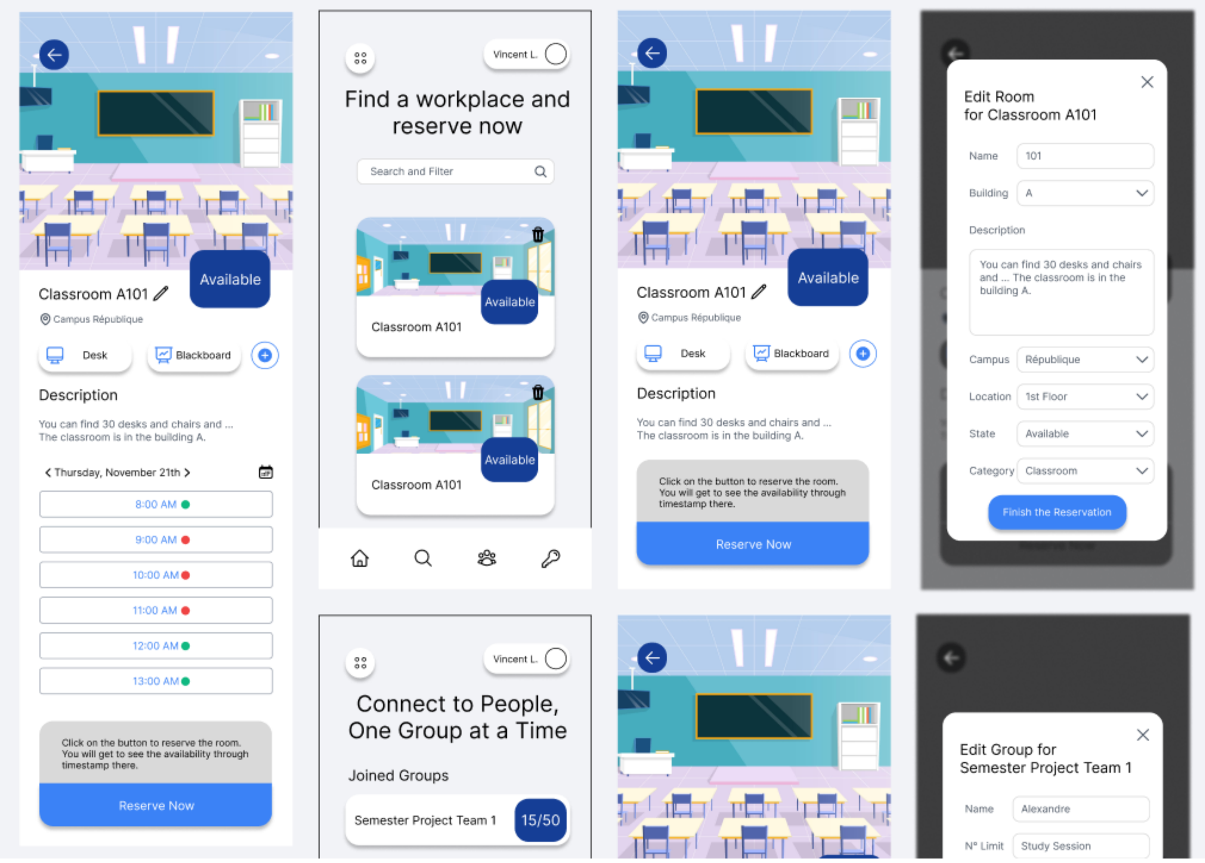
Location: On the floor

Description

The title pretty much describe itself but I want to write a description. So, here it is.

Require: Nothing

Save the Changes



Classroom A101
Available
Campus République
Desk
Blackboard

Description

You can find 30 desks and chairs and ... The classroom is in the building A.

Thursday, November 21th

8:00 AM ●
9:00 AM ●
10:00 AM ●
11:00 AM ●
12:00 AM ●
13:00 AM ●

Click on the button to reserve the room. You will get to see the availability through timestamp there.

Reserve Now

Vincent L.

Find a workplace and reserve now

Search and Filter

Classroom A101
Available

Classroom A101
Available

Classroom A101
Available
Campus République
Desk
Blackboard

Description

You can find 30 desks and chairs and ... The classroom is in the building A.

Click on the button to reserve the room. You will get to see the availability through timestamp there.

Reserve Now

Edit Room
for Classroom A101

Name: 101
Building: A
Description: You can find 30 desks and chairs and ... The classroom is in the building A.
Campus: République
Location: 1st Floor
State: Available
Category: Classroom

Finish the Reservation

Vincent L.

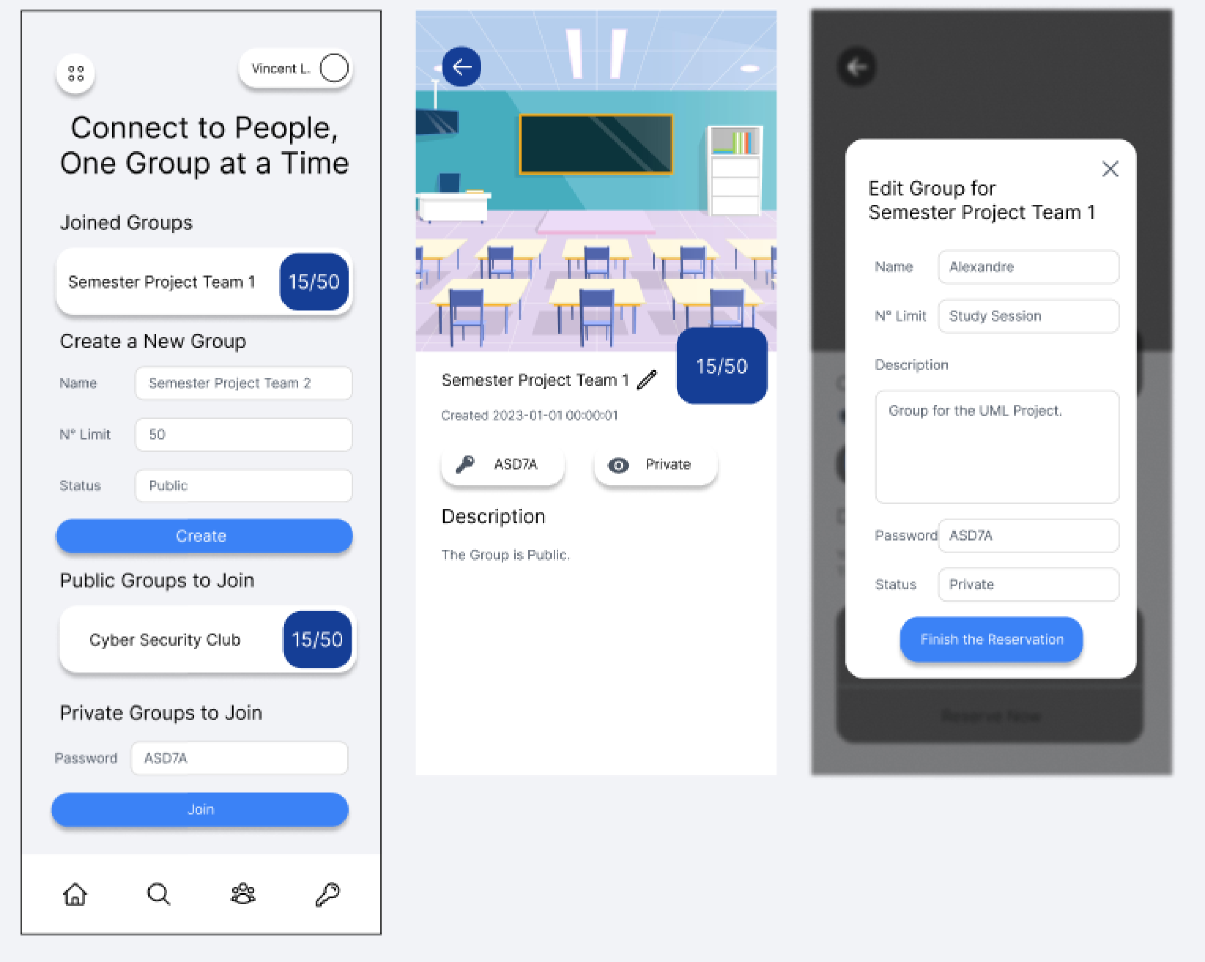
Connect to People, One Group at a Time

Joined Groups

Semester Project Team 1 15/50

Edit Group for Semester Project Team 1

Name: Alexandre
N° Limit: Study Session



Vincent L.

Connect to People, One Group at a Time

Joined Groups

Semester Project Team 1 15/50

Create a New Group

Name: Semester Project Team 2
N° Limit: 50
Status: Public

Create

Public Groups to Join

Cyber Security Club 15/50

Private Groups to Join

Password: ASD7A

Join

Semester Project Team 1
15/50
Created 2023-01-01 00:00:01

ASD7A
Private

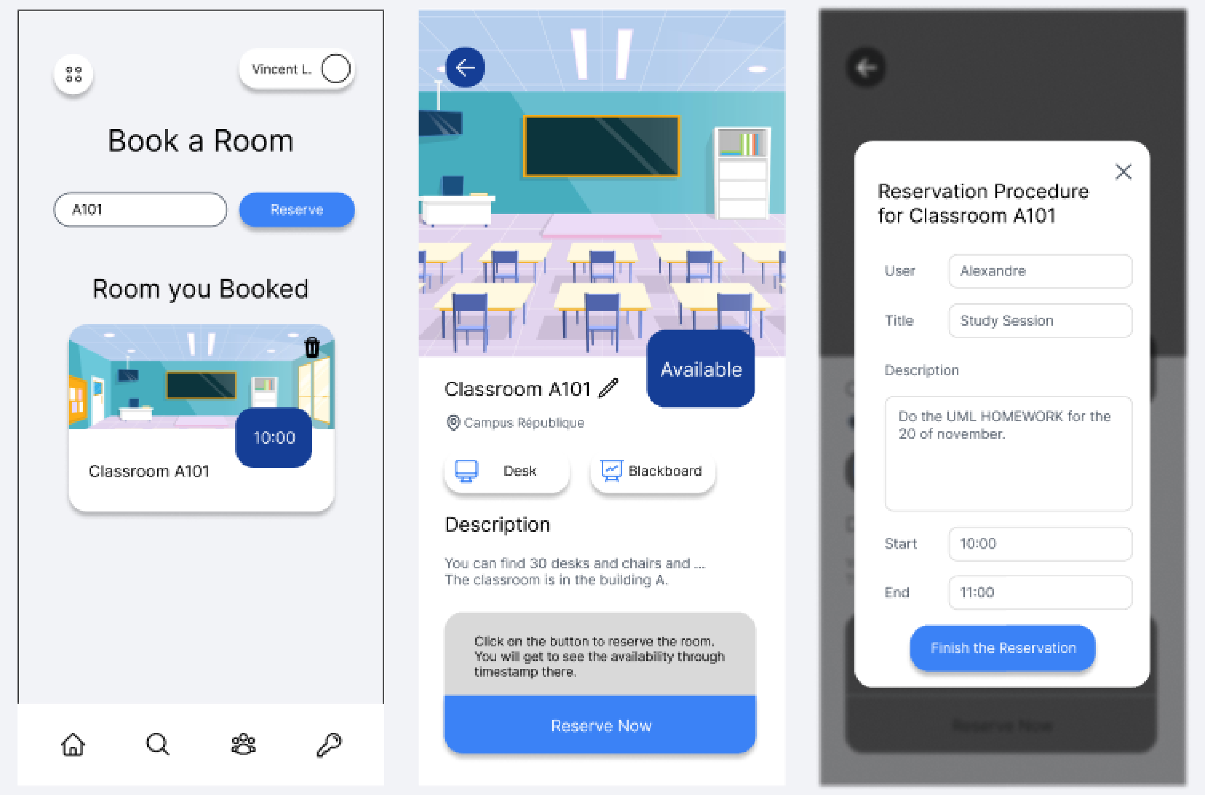
Description

The Group is Public.

Edit Group for Semester Project Team 1

Name: Alexandre
N° Limit: Study Session
Description: Group for the UML Project.
Password: ASD7A
Status: Private

Finish the Reservation



Vincent L.

Book a Room

A101

Reserve

Room you Booked

10:00

Classroom A101

Available

Classroom A101

Campus République

Desk

Blackboard

Description

You can find 30 desks and chairs and ...
The classroom is in the building A.

Click on the button to reserve the room.
You will get to see the availability through timestamp there.

Reserve Now

Reservation Procedure for Classroom A101

User Alexandre

Title Study Session

Description

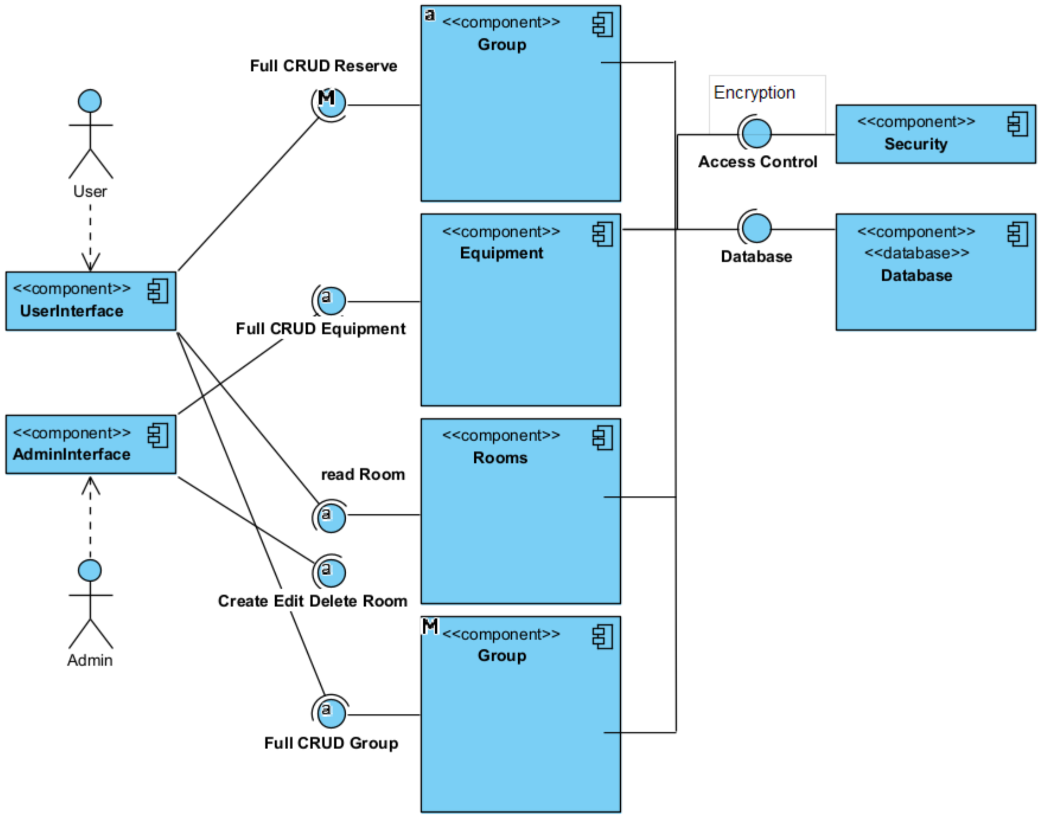
Do the UML HOMEWORK for the 20 of november.

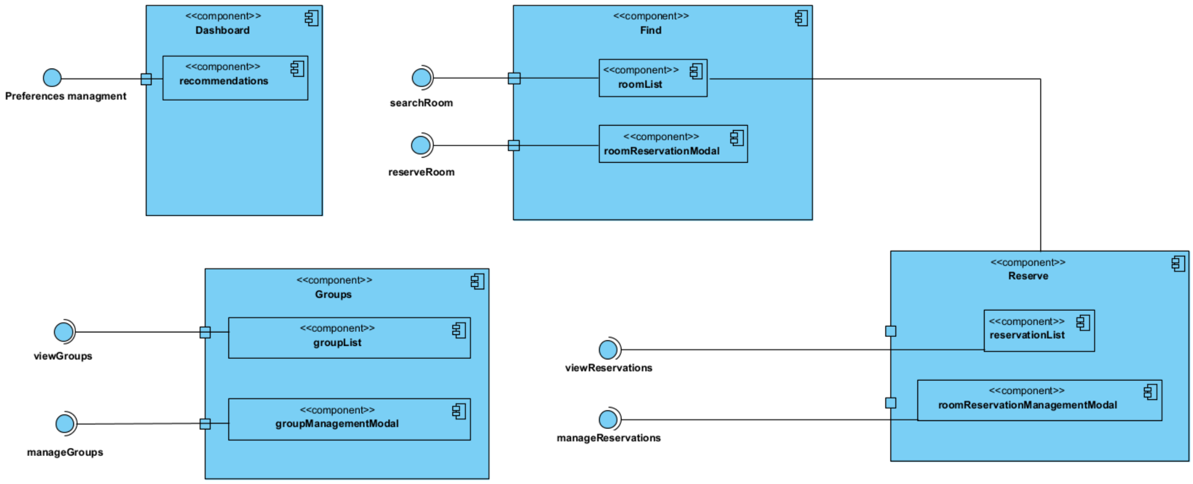
Start 10:00

End 11:00

Finish the Reservation

Component Diagrams (first Vincent & second Alexandre)





The sequence diagrams (We both did it)

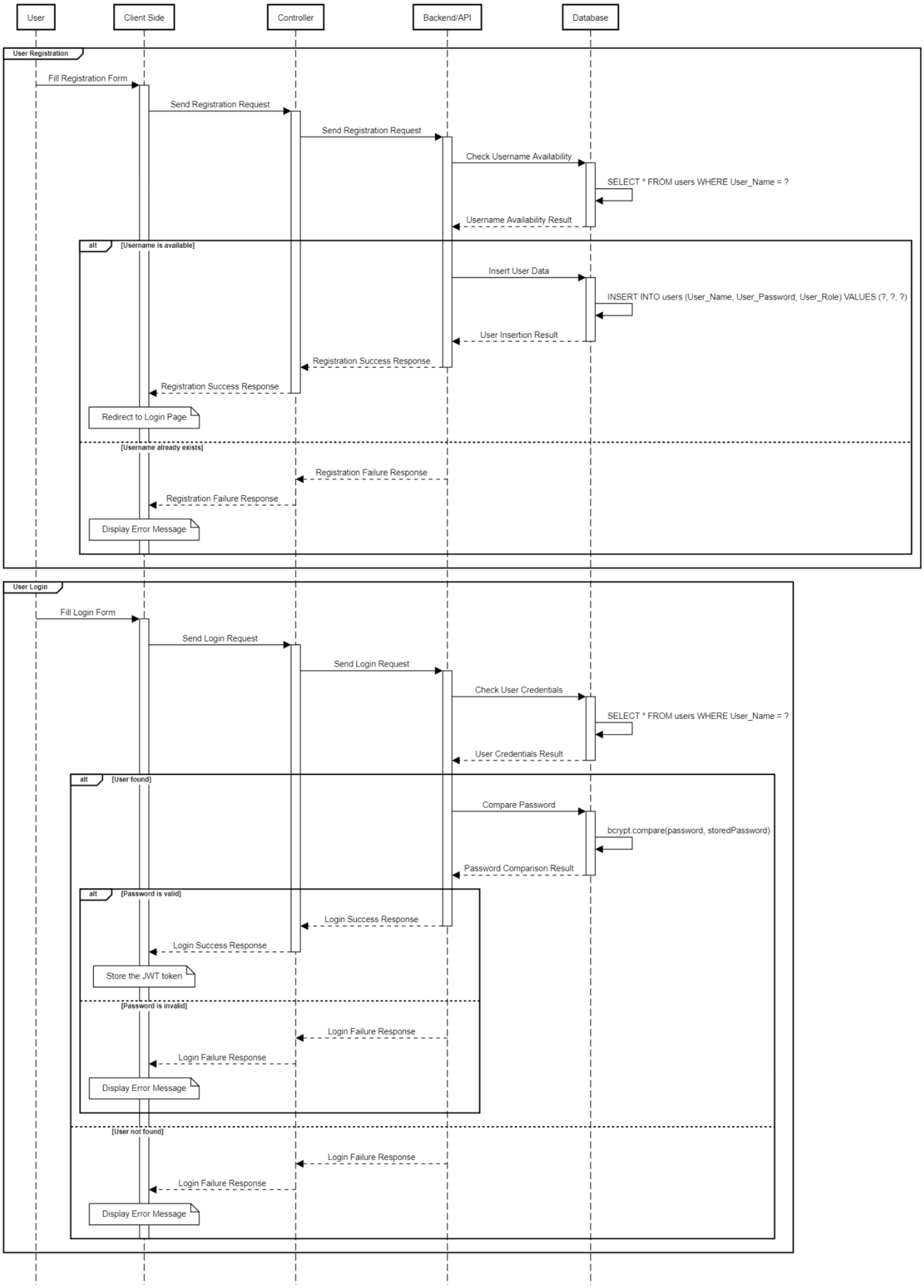
Explanation of Register

The sequence diagram starts with the user filling out the registration form on the client side.
The client side activates and sends a registration request to the frontend controller.
The controller activates and forwards the registration request to the backend/API.
The backend/API activates and performs a database query to check the availability of the chosen username.
If the username is available, the backend/API activates and inserts the user data into the database.
The registration result is then sent back to the controller which responds accordingly.
If the username already exists, a different response is sent back to the controller and an error message is displayed on the client side.

Explanation of Login

The sequence diagram starts with the user filling out the login form on the client side.
The client side activates and sends a login request to the frontend controller.
The controller activates and forwards the login request to the backend/API.
The backend/API activates and performs a database query to check the user's credentials.
If the user is found, the backend/API activates and compares the entered password with the stored password using bcrypt.
If the password is valid, a login success response is sent back to the frontend controller.
Otherwise, a login failure response is sent.
The controller responds accordingly, either storing the JWT token on success or displaying an error message on failure.

Authentication Sequence Diagram



Explanation of the General Sequence Diagram

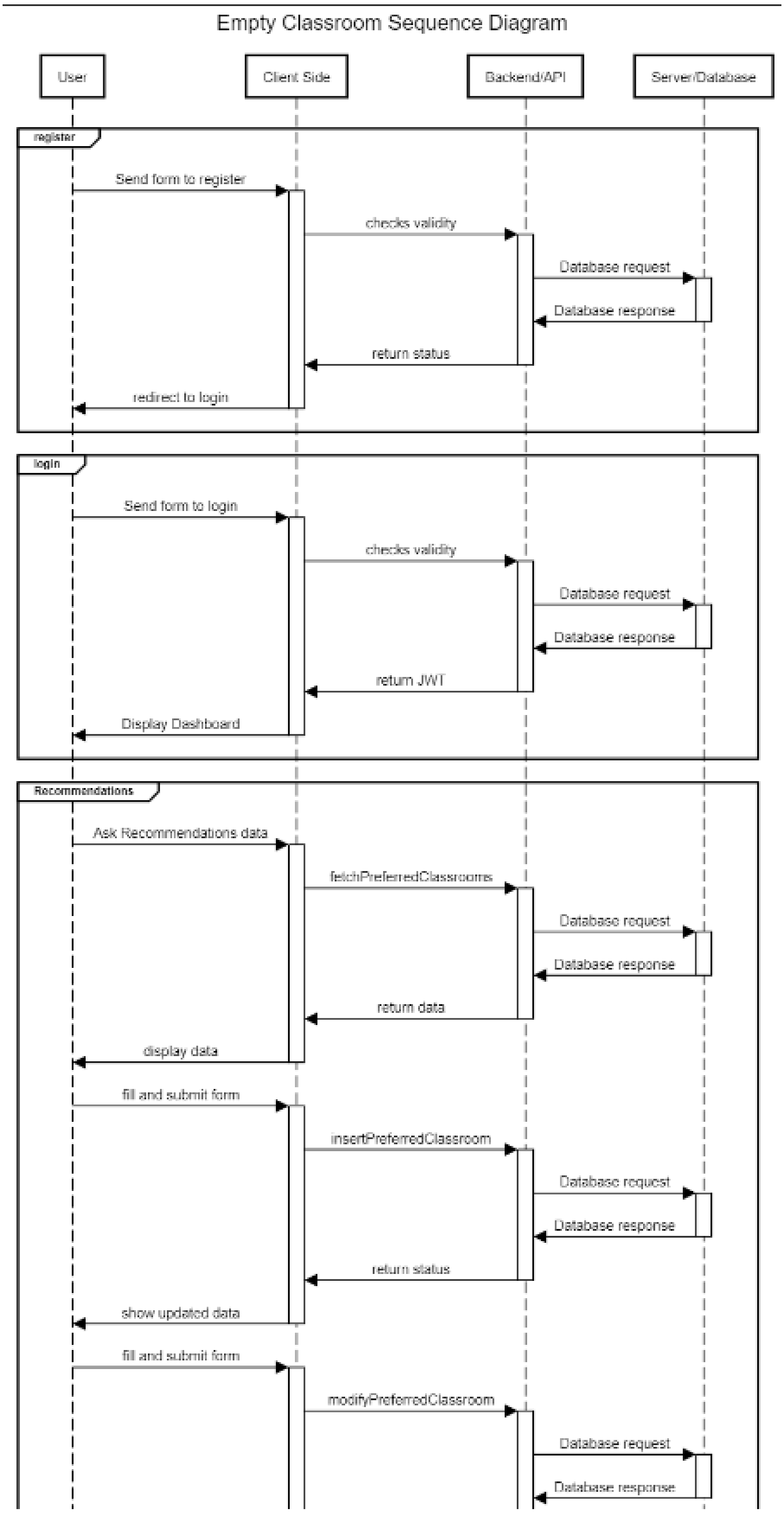
User Registration & Login:
The User sends a registration or login form to the Client Side.
The Client Side checks the validity of the form.
Backend/API handles the registration or login process.
For the registration, the Backend/API communicates with the Server/Database to check if the username is available.
For the login, the Backend/API communicates with the Server/Database to check user credentials.
The Server/Database responds and the Backend/API returns the status to the Client Side.
In addition to the status message, the backend/API sends a JWT (JSON Web Token) back to the client.
If successful, the Client Side redirects the User to the login page after the registration and displays the Dashboard to the User after the login.

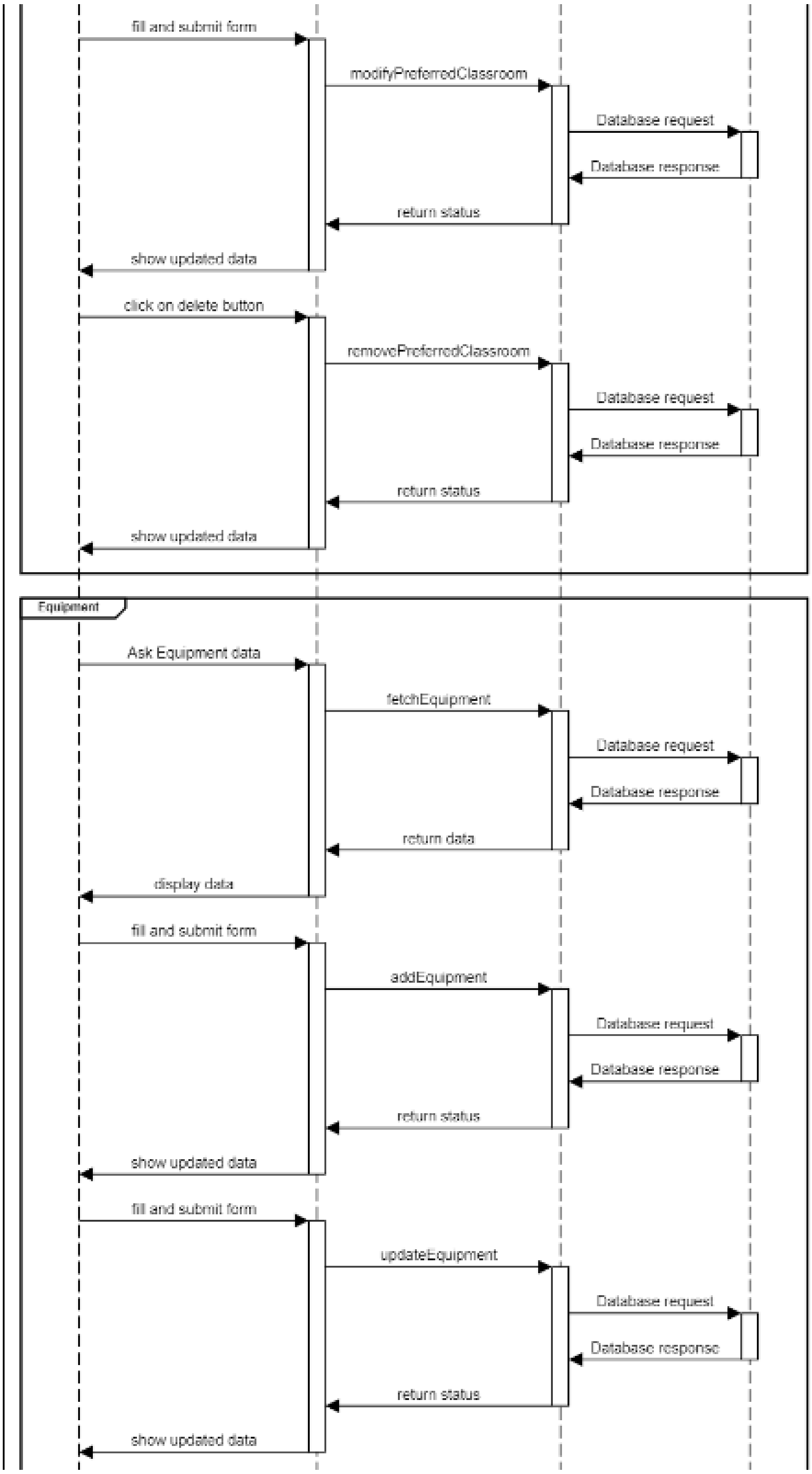
Recommendations Read Operation:
The User requests Recommendations data from the Client Side.
The Client Side activates and fetches Preferred Classrooms data from the Backend/API.
Backend/API communicates with the Server/Database to retrieve data.
The Server/Database responds and the Backend/API returns the data to the Client Side.
The Client Side displays the data to the User.

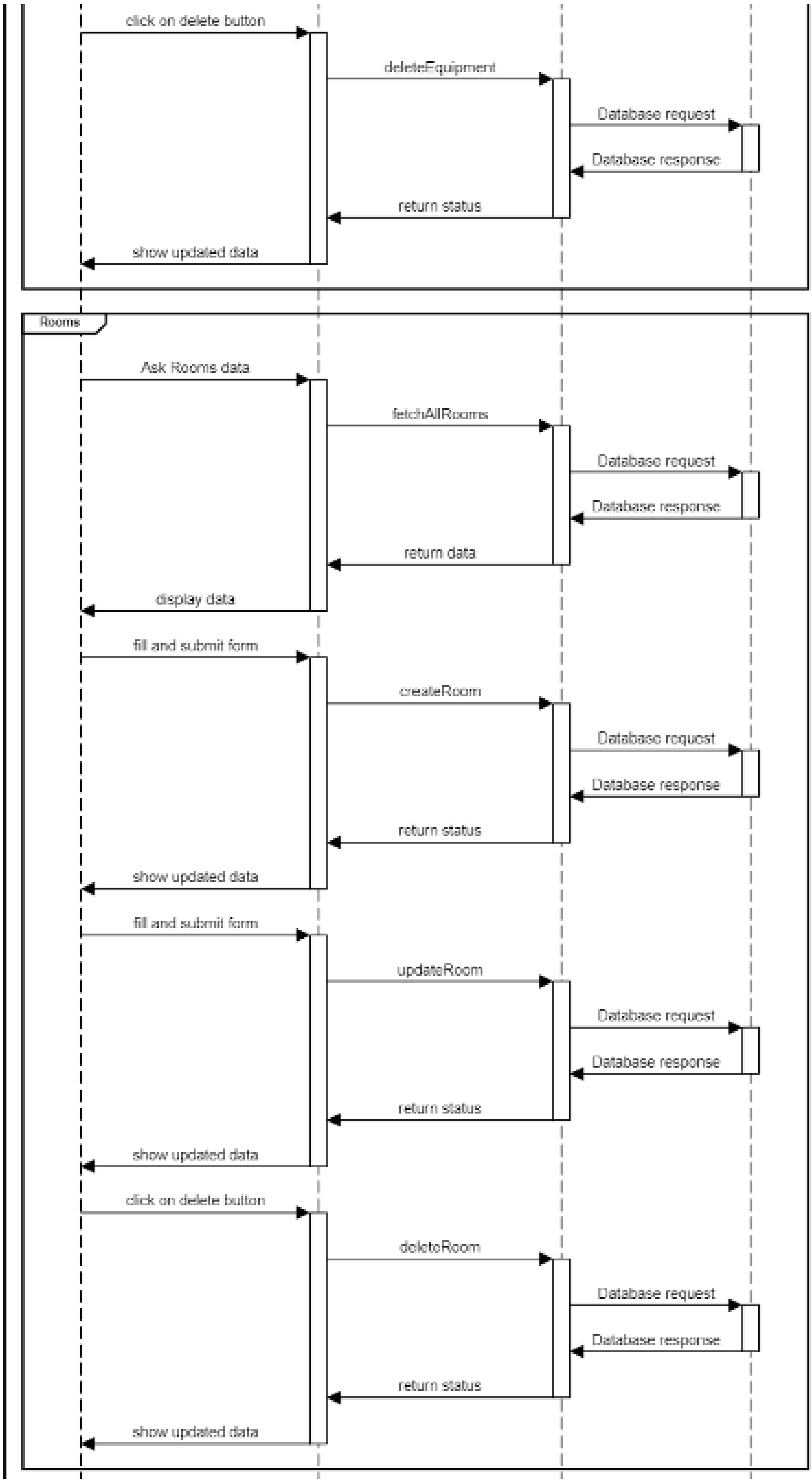
Equipment, Rooms, Groups, Reservations, Chat Conversation Read Operation:
Similar sequences follow for Equipment, Rooms, Groups, Reservations and Chat Conversation.
The User requests data, the Client Side fetches data from the Backend/API and the Backend/API communicates with the Server/Database.

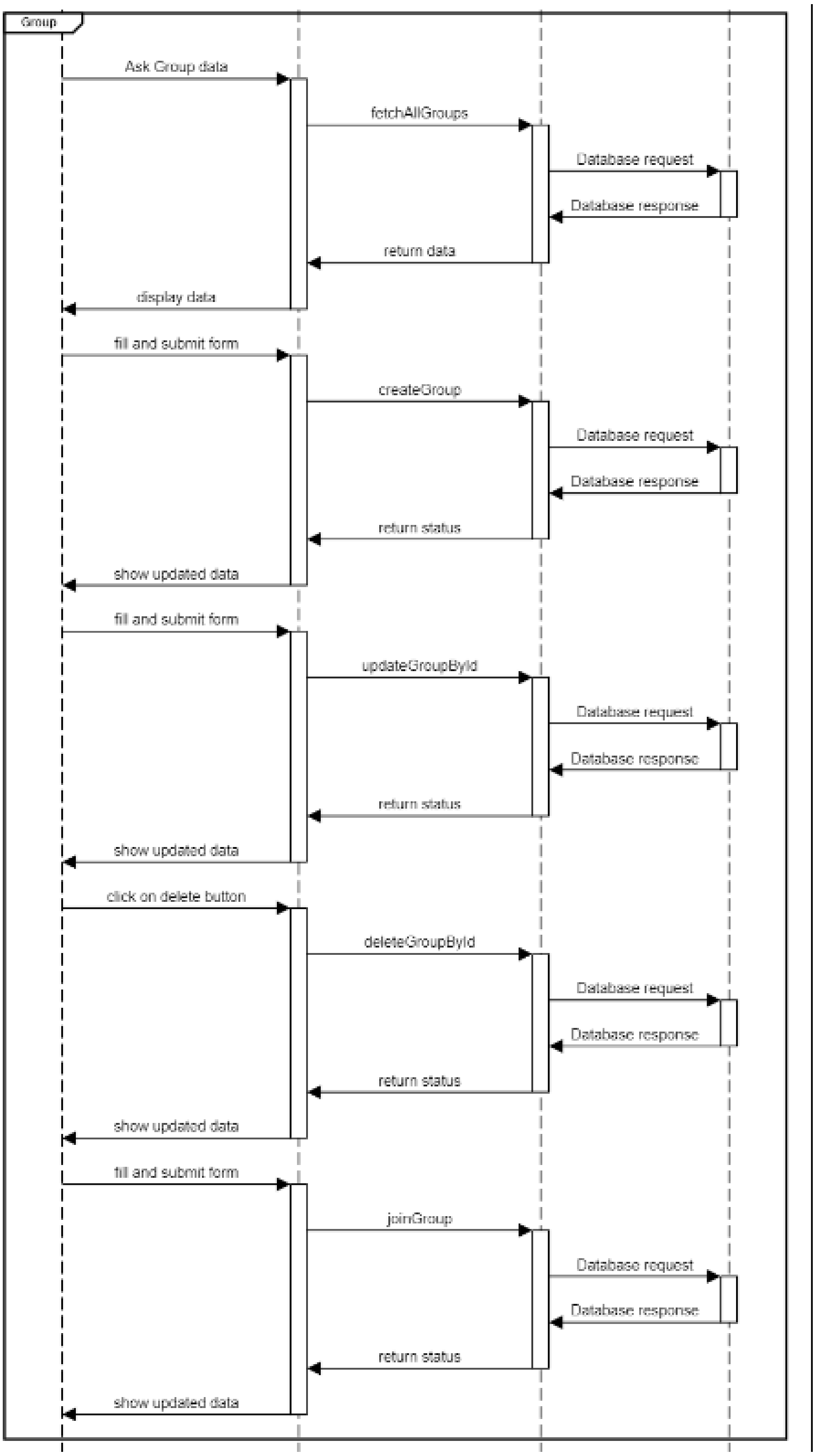
User Create, Edit & Delete:
For each section (Recommendations, Equipment, Rooms, Groups, Reservations, Chat Conversation), the User can interact by filling and submitting forms, leading to updates in the database (creation and editing) and showing the updated data.
The User can also click on a button to delete an object in the database.

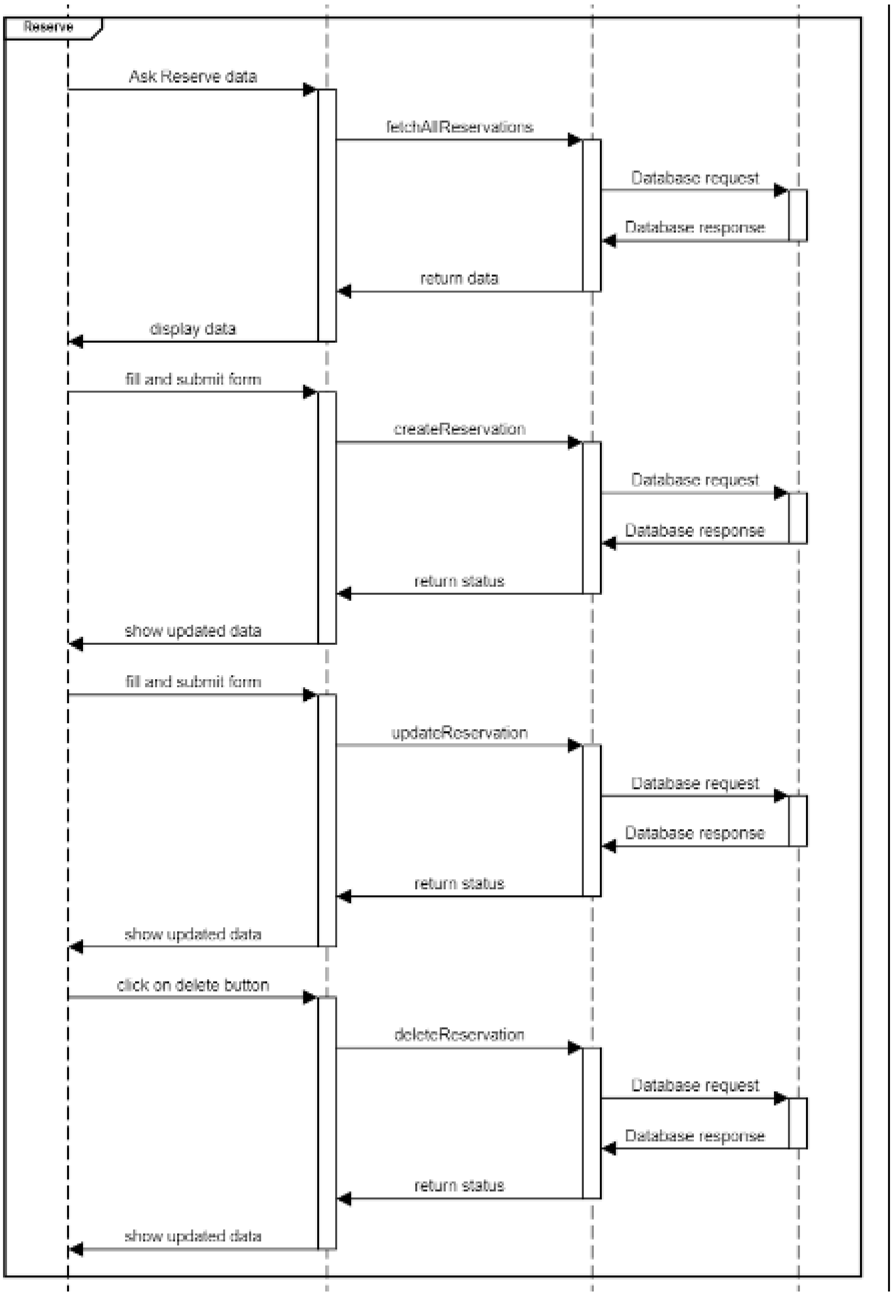
The Lifespan of each Participant:
Bars are used to represent the lifespan of each participant (User, Client Side, Backend/API, Server/Database) during specific actions.

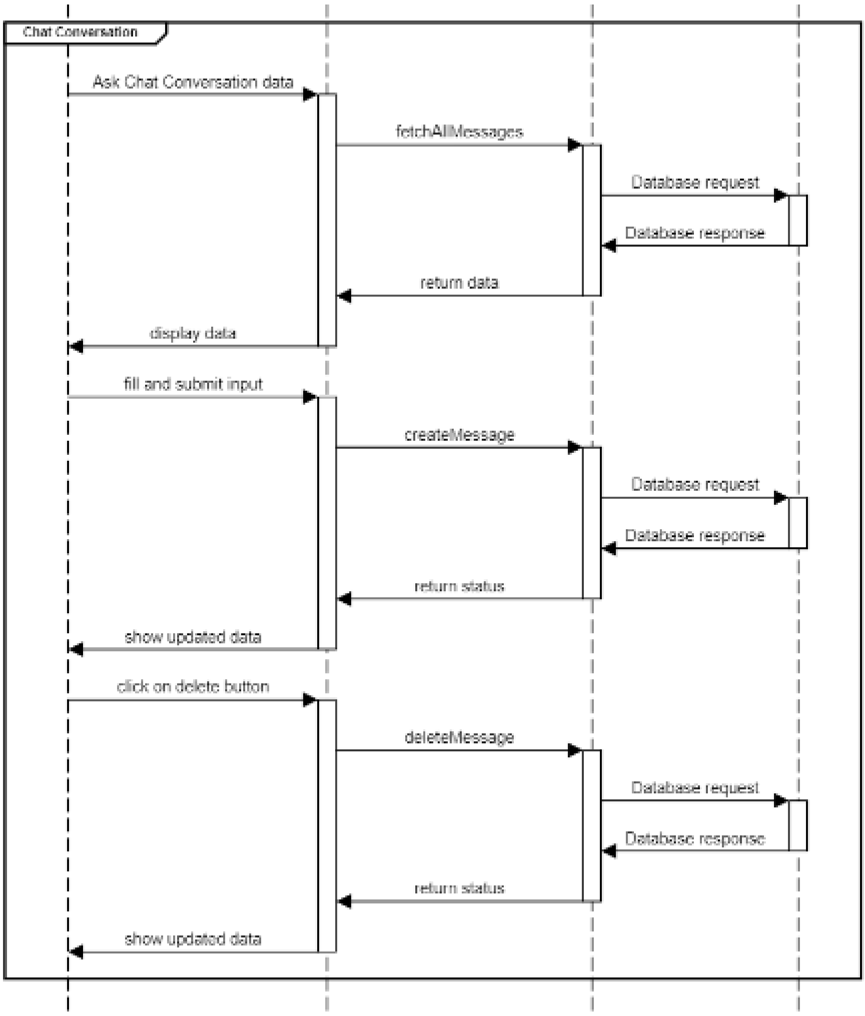

Empty Classroom Sequence Diagram
User
Client Side
Backend/API
Server/Database
register
Send form to register
checks validity
Database request
Database response
return status
redirect to login
login
Send form to login
checks validity
Database request
Database response
return JWT
Display Dashboard
Recommendations
Ask Recommendations data
fetchPreferredClassrooms
Database request
Database response
return data
display data
fill and submit form
insertPreferredClassroom
Database request
Database response
return status
show updated data
fill and submit form
modifyPreferredClassroom
Database request
Database response


fill and submit form
modifyPreferredClassroom
Database request
Database response
return status
show updated data
click on delete button
removePreferredClassroom
Database request
Database response
return status
show updated data
Equipment
Ask Equipment data
fetchEquipment
Database request
Database response
return data
display data
fill and submit form
addEquipment
Database request
Database response
return status
show updated data
fill and submit form
updateEquipment
Database request
Database response
return status
show updated data


click on delete button
deleteEquipment
Database request
Database response
return status
show updated data
Rooms
Ask Rooms data
fetchAllRooms
Database request
Database response
return data
display data
fill and submit form
createRoom
Database request
Database response
return status
show updated data
fill and submit form
updateRoom
Database request
Database response
return status
show updated data
click on delete button
deleteRoom
Database request
Database response
return status
show updated data

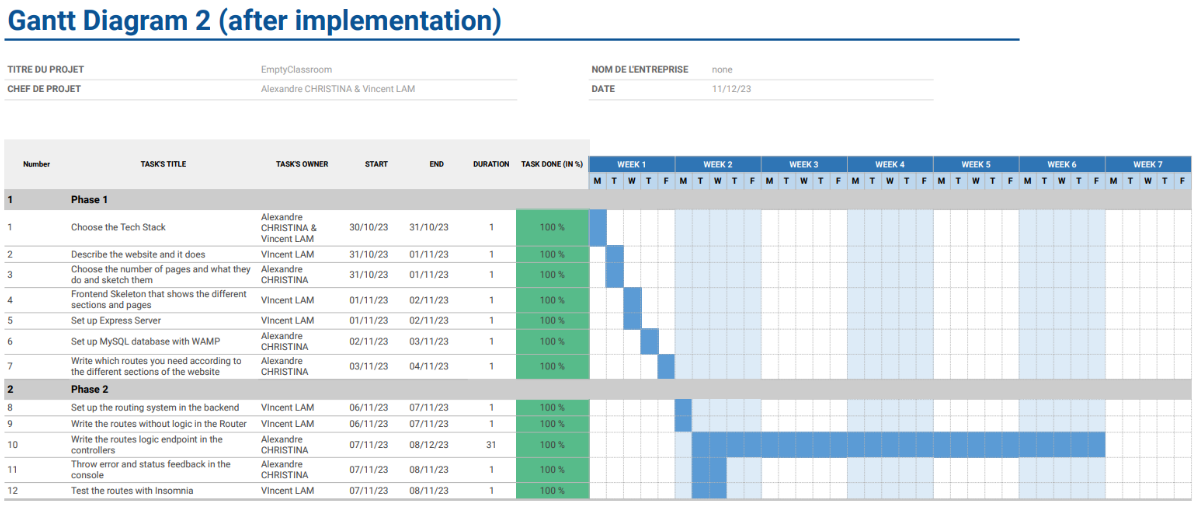

Group
Ask Group data
fetchAllGroups
Database request
Database response
return data
display data
fill and submit form
createGroup
Database request
Database response
return status
show updated data
fill and submit form
updateGroupById
Database request
Database response
return status
show updated data
click on delete button
deleteGroupById
Database request
Database response
return status
show updated data
fill and submit form
joinGroup
Database request
Database response
return status
show updated data


Reserve
Ask Reserve data
fetchAllReservations
Database request
Database response
return data
display data
fill and submit form
createReservation
Database request
Database response
return status
show updated data
fill and submit form
updateReservation
Database request
Database response
return status
show updated data
click on delete button
deleteReservation
Database request
Database response
return status
show updated data



Required Diagrams at the end

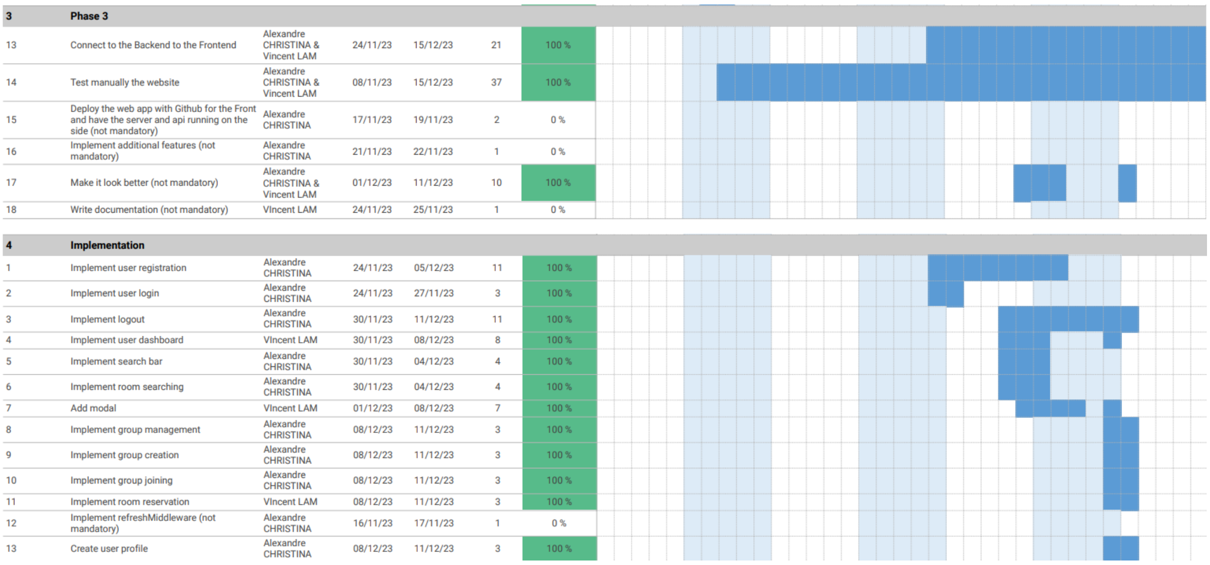
Gantt Diagram (Who worked where)



Gantt Diagram 2 (after implementation)

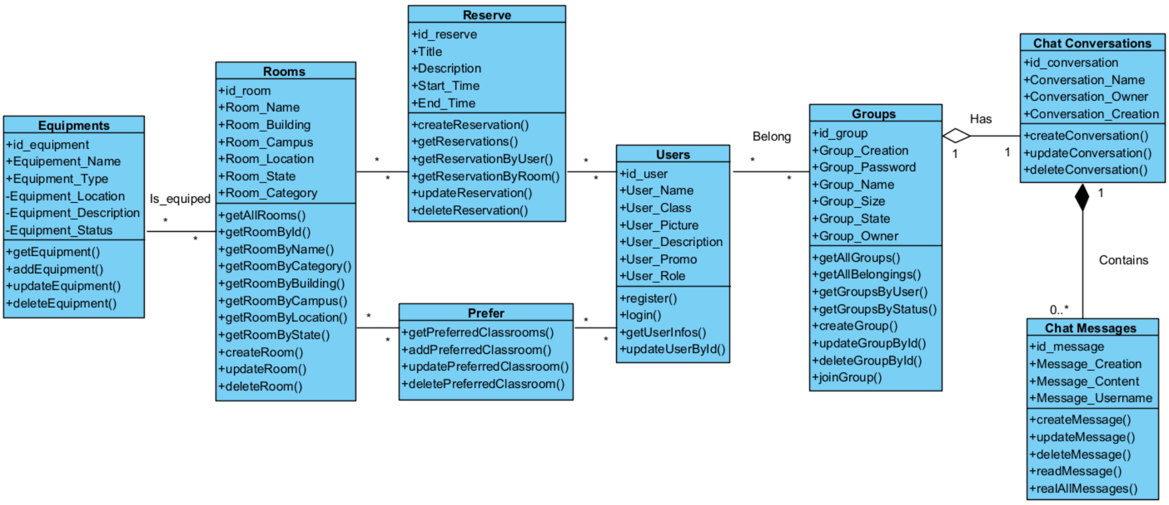
TITRE DU PROJET	EmptyClassroom	NOM DE L'ENTREPRISE	none
CHEF DE PROJET	Alexandre CHRISTINA & Vincent LAM	DATE	11/12/23

Number	TASK'S TITLE	TASK'S OWNER	START	END	DURATION	TASK DONE (IN %)
1	Phase 1					
1	Choose the Tech Stack	Alexandre CHRISTINA & Vincent LAM	30/10/23	31/10/23	1	100 %
2	Describe the website and it does	VIncent LAM	31/10/23	01/11/23	1	100 %
3	Choose the number of pages and what they do and sketch them	Alexandre CHRISTINA	31/10/23	01/11/23	1	100 %
4	Frontend Skeleton that shows the different sections and pages	VIncent LAM	01/11/23	02/11/23	1	100 %
5	Set up Express Server	VIncent LAM	01/11/23	02/11/23	1	100 %
6	Set up MySQL database with WAMP	Alexandre CHRISTINA	02/11/23	03/11/23	1	100 %
7	Write which routes you need according to the different sections of the website	Alexandre CHRISTINA	03/11/23	04/11/23	1	100 %
2	Phase 2					
8	Set up the routing system in the backend	VIncent LAM	06/11/23	07/11/23	1	100 %
9	Write the routes without logic in the Router	VIncent LAM	06/11/23	07/11/23	1	100 %
10	Write the routes logic endpoint in the controllers	Alexandre CHRISTINA	07/11/23	08/12/23	31	100 %
11	Throw error and status feedback in the console	Alexandre CHRISTINA	07/11/23	08/11/23	1	100 %
12	Test the routes with Insomnia	VIncent LAM	07/11/23	08/11/23	1	100 %



3	**Phase 3**					
13	Connect to the Backend to the Frontend	Alexandre CHRISTINA & Vincent LAM	24/11/23	15/12/23	21	100 %
14	Test manually the website	Alexandre CHRISTINA & Vincent LAM	08/11/23	15/12/23	37	100 %
15	Deploy the web app with Github for the Front and have the server and api running on the side (not mandatory)	Alexandre CHRISTINA	17/11/23	19/11/23	2	0 %
16	Implement additional features (not mandatory)	Alexandre CHRISTINA	21/11/23	22/11/23	1	0 %
17	Make it look better (not mandatory)	Alexandre CHRISTINA & Vincent LAM	01/12/23	11/12/23	10	100 %
18	Write documentation (not mandatory)	VIncent LAM	24/11/23	25/11/23	1	0 %
4	**Implementation**					
1	Implement user registration	Alexandre CHRISTINA	24/11/23	05/12/23	11	100 %
2	Implement user login	Alexandre CHRISTINA	24/11/23	27/11/23	3	100 %
3	Implement logout	Alexandre CHRISTINA	30/11/23	11/12/23	11	100 %
4	Implement user dashboard	VIncent LAM	30/11/23	08/12/23	8	100 %
5	Implement search bar	Alexandre CHRISTINA	30/11/23	04/12/23	4	100 %
6	Implement room searching	Alexandre CHRISTINA	30/11/23	04/12/23	4	100 %
7	Add modal	VIncent LAM	01/12/23	08/12/23	7	100 %
8	Implement group management	Alexandre CHRISTINA	08/12/23	11/12/23	3	100 %
9	Implement group creation	Alexandre CHRISTINA	08/12/23	11/12/23	3	100 %
10	Implement group joining	Alexandre CHRISTINA	08/12/23	11/12/23	3	100 %
11	Implement room reservation	VIncent LAM	08/12/23	11/12/23	3	100 %
12	Implement refreshMiddleware (not mandatory)	Alexandre CHRISTINA	16/11/23	17/11/23	1	0 %
13	Create user profile	Alexandre CHRISTINA	08/12/23	11/12/23	3	100 %

Class Diagram


Equipments
+id_equipment
+Equipement_Name
+Equipment_Type
-Equipment_Location
-Equipment_Description
-Equipment_Status
+getEquipment()
+addEquipment()
+updateEquipment()
+deleteEquipment()
Is_equiped
*
*
Rooms
+id_room
+Room_Name
+Room_Building
+Room_Campus
+Room_Location
+Room_State
+Room_Category
+getAllRooms()
+getRoomById()
+getRoomByName()
+getRoomByCategory()
+getRoomByBuilding()
+getRoomByCampus()
+getRoomByLocation()
+getRoomByState()
+createRoom()
+updateRoom()
+deleteRoom()
*
*
Reserve
+id_reserve
+Title
+Description
+Start_Time
+End_Time
+createReservation()
+getReservations()
+getReservationByUser()
+getReservationByRoom()
+updateReservation()
+deleteReservation()
*
*
Prefer
+getPreferredClassrooms()
+addPreferredClassroom()
+updatePreferredClassroom()
+deletePreferredClassroom()
*
*
*
*
Users
+id_user
+User_Name
+User_Class
+User_Picture
+User_Description
+User_Promo
+User_Role
+register()
+login()
+getUserInfos()
+updateUserById()
Belong
*
*
Groups
+id_group
+Group_Creation
+Group_Password
+Group_Name
+Group_Size
+Group_State
+Group_Owner
+getAllGroups()
+getAllBelongings()
+getGroupsByUser()
+getGroupsByStatus()
+createGroup()
+updateGroupById()
+deleteGroupById()
+joinGroup()
Has
1
1
Chat Conversations
+id_conversation
+Conversation_Name
+Conversation_Owner
+Conversation_Creation
+createConversation()
+updateConversation()
+deleteConversation()
1
Contains
0..*
Chat Messages
+id_message
+Message_Creation
+Message_Content
+Message_Username
+createMessage()
+updateMessage()
+deleteMessage()
+readMessage()
+realAllMessages()