

Project Summary

The main customer is the manager and people in Southwest innovation Research Lab who want to develop a website for managing a shared workspace. The users form as teams and managed by admin/staff. Each team will have a team leader who can make reservation of the shared space area for the team to work. The admin/staff can create trainings that the space users can take. The customer also wants to track the reservation and training, knowing who is/will be using the space and who takes the trainings. Stakeholders are the Southwest innovation Research Lab, Prof. Ritchey, and TAs (Akshit Kharbanda and Abhishek Sinha).

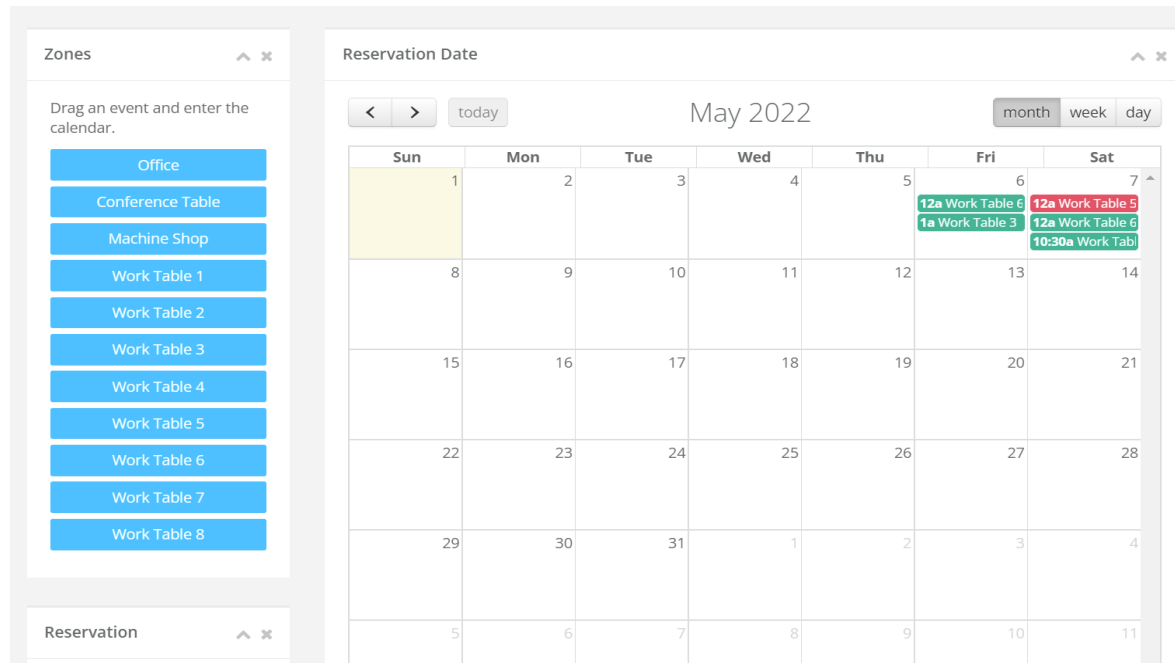
First of all, the app would track which users finished the training. The administrator could create training and change the training results. The shared space is divided into several zones so that different teams would create the reservations for the spaces that would show the time slots on some calendar format, which people can choose to reserve the spaces if they will be noisy or they need quiet environment. The app or web interface would make long-standing reservations for weekly meetings and short-term meetings. The users in the app or web interface would be divided into three roles: standard users that can be a member of one or more teams, or no teams and can see the calendar and sign up for training, team leaders (aside from the role of a standard user) that should be able to add/remove people from their team (both as a user or as a lead) and to create reservations, and staff including administrator should be able to modify/create teams, promote leaders, create trainings, update records, and so on.

User Stories

Iteration 1

View Reservations: As a member, I want to view the reservations of my teams so that I know when to use the space. As a team leader, I want to view all reservations so that I can make new reservations based on available time slots. As an administrator, I want to view all reservations so that I can decide when to hold the training.

This is a 2-point medium level feature, we implemented this feature as the general idea that people who use this website can view different reservations, and they can click on different places in the website to do their own operations.



Cancel a reservation: As a team leader, I want to cancel my reservations, so that I can change the time our team use the space. As an administrator, I want to cancel any reservation, so that I can prioritize the training I want to hold.

This is a 1-point straightforward feature, we implemented this feature at the beginning of the project and worked properly.

Reservation History

Home / Reservation / **Reservation History**

Title ↕	Zone ↕	Team ↕	Type ↕	Start Time ↕	End Time ↕	Action ↕
> re2	Work Table 3	team_1	Quiet	May 6, 2022, 1 a.m.	May 6, 2022, 9:30 a.m.	Delete
> re3	Work Table 6	team_1	Quiet	May 6, 2022, midnight	May 7, 2022, midnight	Delete
> re2	Work Table 5	team_1	Noisy	May 7, 2022, midnight	May 8, 2022, midnight	Delete
> 1	Work Table 4	team_1	Quiet	May 7, 2022, 10:30 a.m.	May 7, 2022, 4:30 p.m.	Delete
> re3	Work Table 6	team_1	Quiet	May 7, 2022, midnight	May 8, 2022, midnight	Delete

Create a reservation: As a team leader, I can create a reservation, so that I can book the shared space area for team use

This is a 3-point complex level feature, since this feature is very crucial and will be the core functionality of this project. We implemented and finished this feature by correctly

avoiding the conflicts of open, closed and quiet reservations by dragging the reservation zones with the type into the calendar.

Work Table 1

Work Table 2

Work Table 3

Work Table 4

Work Table 5

Work Table 6

Work Table 7

Work Table 8

Reservation

Team

team_3

☒ Sync to Google Calendar
 ☒ Send notification

Title

reservation

Will you make noise when using the space?

No

Do you need a reservation with a quiet environment?

No

Description

Submit

Reset

Work Table 7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

1

2

3

4

5

6

7

8

9

10

11

Iteration 2

View team information: As a member, I want to view my team information including all teammates and the team leader, so that I know who I am working with.

This is a 2-point medium level feature, we implemented and finished this feature that people can click on the team name and see all the information of the team.

jinhao4

member

Home

Reservation

Teams

My Teams

Training

Welcome to Southwest Innovation Research Lab Shared Space. Log out

team_3 Members

Home / My Teams / Team Detail

Username	Email	Role	Join Time	Action
jinhao	jeffreypan@tamu.edu	Leader	May 1, 2022, 3:21 p.m.	
jinhao4	jinhaopan0319@gmail.com	Member	May 1, 2022, 3:24 p.m.	

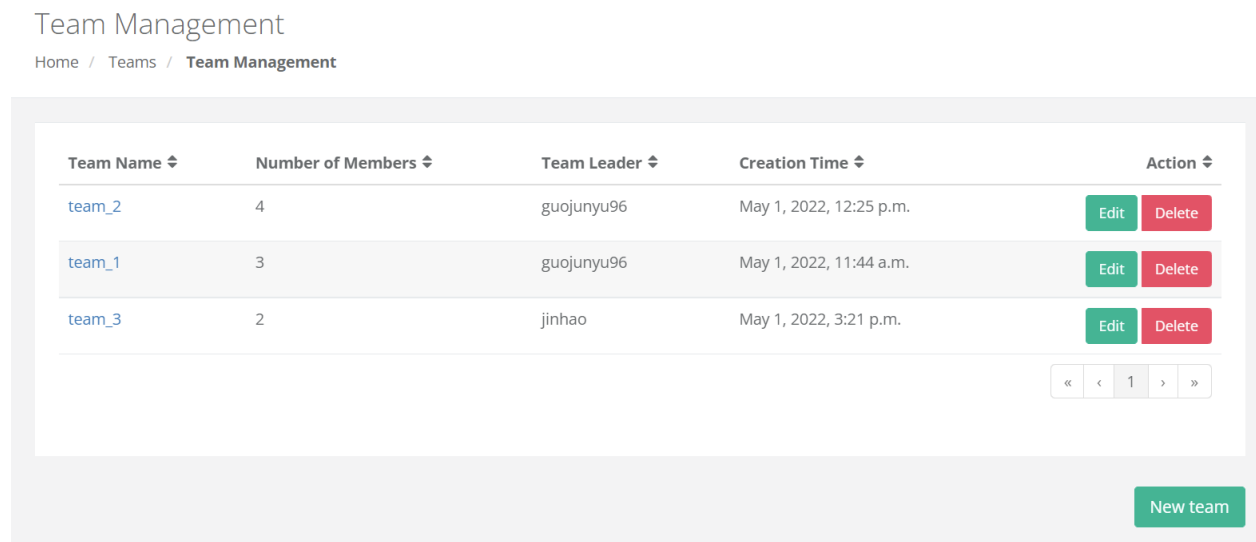
« < 1 > »

Copyright ©Southwest Innovation Research Lab 2020-2023

Texas A&M University TAMU @College Station

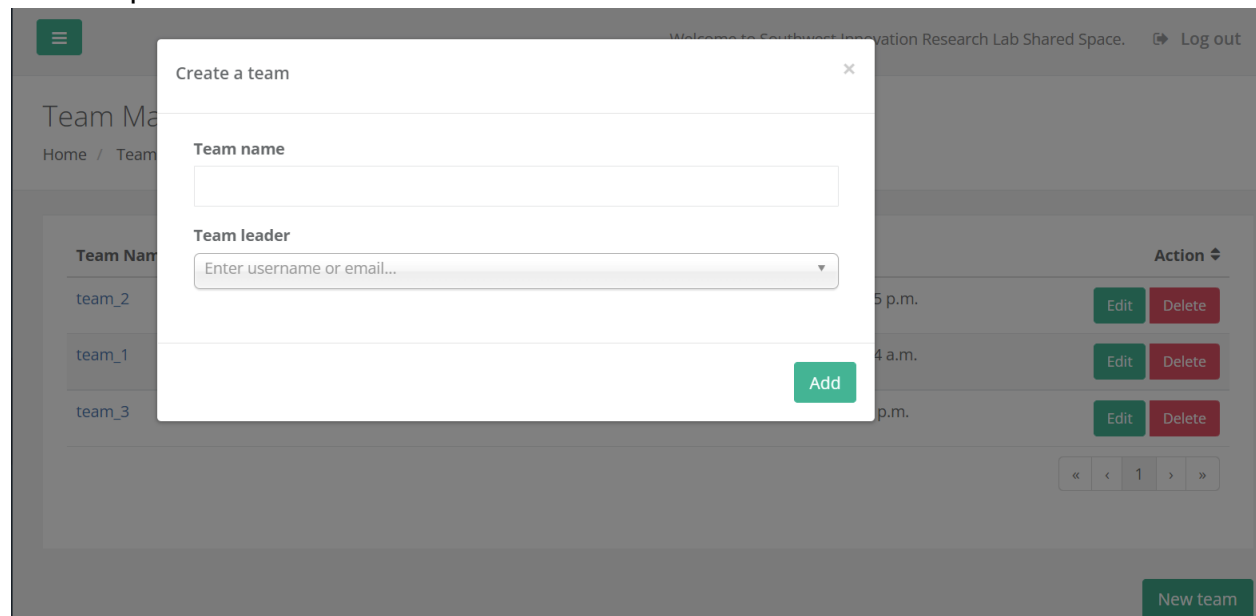
Remove teams: As an administrator, I want to remove teams that no longer use the space, so that I do not need to manage them any more.

This is a 1-point straightforward feature, we implemented and finished this feature that offer the option for people who create teams and would like to delete it due to some issues.



Create teams: As an administrator, I want to create teams, so that people can use the space together

This is a 2-point medium level feature, we implemented and finished this feature that administrator would be able to create teams so that team leaders and team members can do operations later.



User log in: As a user, I want to log in using my Google account, so that I can use the functionalities of the application. As a user, I can log in with a username and password, so that I can use the functionalities of the application.

This is a 3-point complex feature, we implemented and finished this feature that user can use their google account to log in as different user roles.

Welcome to **Southwest Innovation Research Lab**

Log in

Username:

Password:

Remember Me: ☐

Login

Do not have an account?

Create an account

OR

Sign in with Google

Southwest Innovation Research Lab © 2020-2023

Texas A&M University **TAMU** @College Station

Copyright ©Southwest Innovation Research Lab 2020-2023

Sign in with Google

Choose an account

to continue to shared-space.herokuapp.com

Jinhao Pan
jeffreypa@tamu.edu

Jinhao Pan
jinhaopan0319@gmail.com

Use another account

To continue, Google will share your name, email address, language preference, and profile picture with shared-space.herokuapp.com.

Iteration 3

Promote or demote team leaders: As an administrator, I want to promote a member of a team to its team leader or demote a team leader to a normal member, so that I can determine who are the team leaders

This is a 1-point straightforward level feature, we implemented and finished this feature that offer the option to administrator to modify the roles of team leaders.

Edit a team

Team name

team_3

Team leader

jinhao (jeffreypa@tamu.edu)

jinhao (jeffreypa@tamu.edu)

jinhao4 (jinhaopan0319@gmail.com)

Team Management

Home / Team Management

Team Name	Team Leader	Action
team_2	5 p.m.	Edit Delete
team_1	4 a.m.	Edit Delete
team_3	p.m.	Edit Delete

« < 1 > »

Manage team members: As a team leader, I want to add a member to my team or remove a member from my team, so that I can form the desirable team.

This is a 2-point medium level feature, we implemented and finished this feature to offer the options to team leaders to change their teams.

team_3 Members

Home / My Teams / Team Detail


Username ↕	Email ↕	Role ↕	Join Time ↕	Action ↕
jinhao	jeffreypa@tam.u.edu	Leader	May 1, 2022, 3:21 p.m.	
jinhao4	jinhaopan0319@gmail.com	Member	May 1, 2022, 3:24 p.m.	<button>Delete</button>

« < 1 > »

Invite Member

View Training: As an administrator/staff, I want to view all the training records, who registered for the training, and who passed the training, so that I can track the training information of everyone who is using the shared space. As a member, I can view all training on the calendar so that I know when and where the training will be held.

This is a 2-point medium level feature, we implemented and finished this feature that administrator can view the training information and records.


admin
admin ▾

Home

Users

Reservation

Teams

Training ▾

Training Management

Registered Training

Training Registration

Welcome to Southwest Innovation Research Lab Shared Space. [Log out](#)

Training Management

Home / Training / Training Management

Training ↕	Zone ↕	Start Time ↕	End Time ↕	Status ↕	Action ↕
> training1	Work Table 5	May 1, 2022, 12:27 a.m.	May 5, 2022, 10:50 p.m.	Not Start	<button>Finish</button> <button>Delete</button>
> training2	Work Table 2	May 3, 2022, 11:50 a.m.	May 4, 2022, 3:55 p.m.	Not Start	<button>Finish</button> <button>Delete</button>
> training3	Work Table 6	May 1, 2022, 12:23 p.m.	May 2, 2022, 2:45 p.m.	Not Start	<button>Finish</button> <button>Delete</button>

« < 1 > »

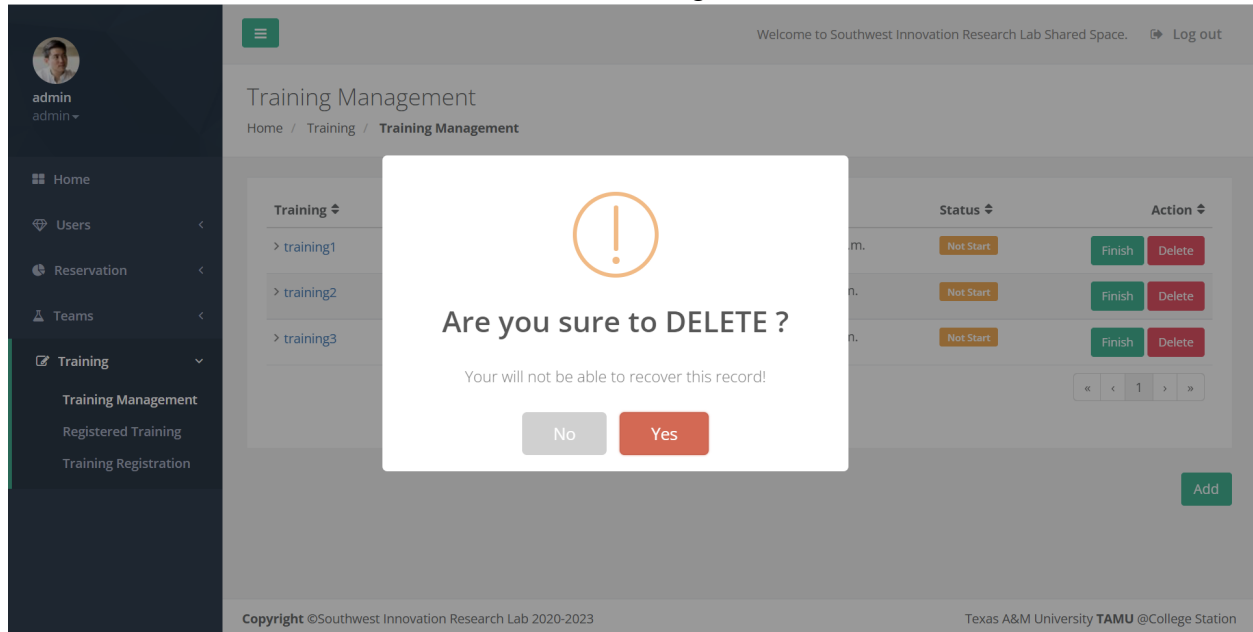
Add

Copyright ©Southwest Innovation Research Lab 2020-2023

Texas A&M University TAMU @College Station

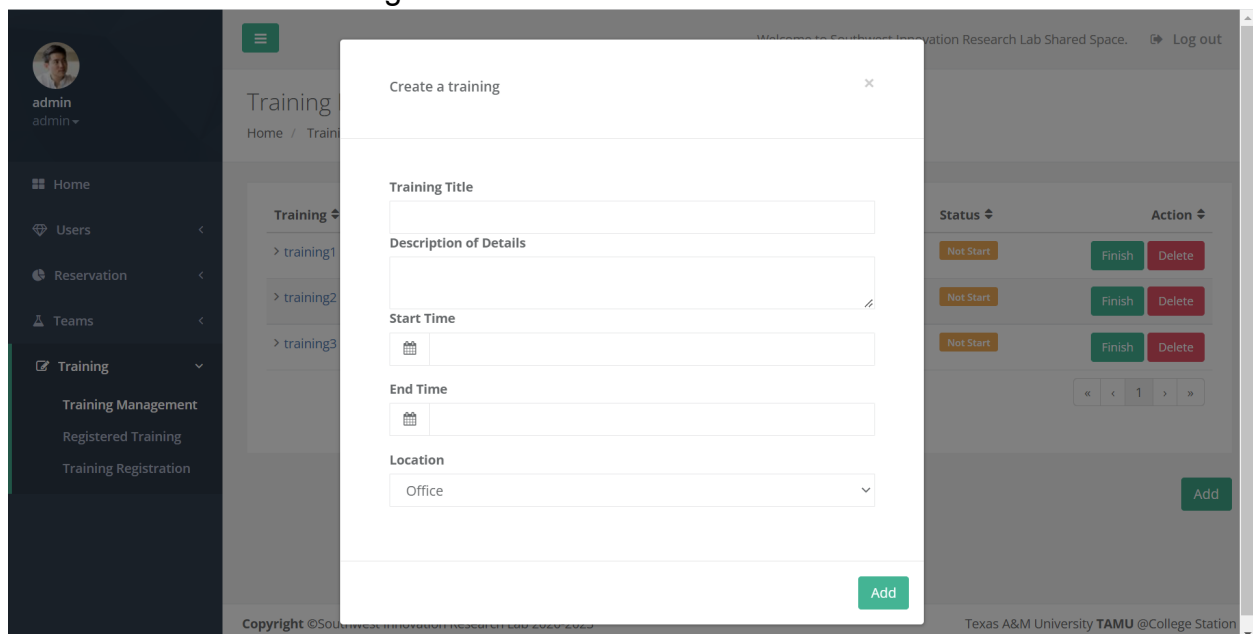
Cancel Training: As an administrator, I want to cancel the existing training, so that I can reschedule the training at any time.

This is a 1-point straightforward level feature, we implemented and finished this feature that administrator would be able to cancel trainings



Create Training: As an administrator/staff, I want to create training, so that the members who want to use the shared space can take part in learning basic security rules about space usage. As a member, I can register a training so that I can join it when it starts.

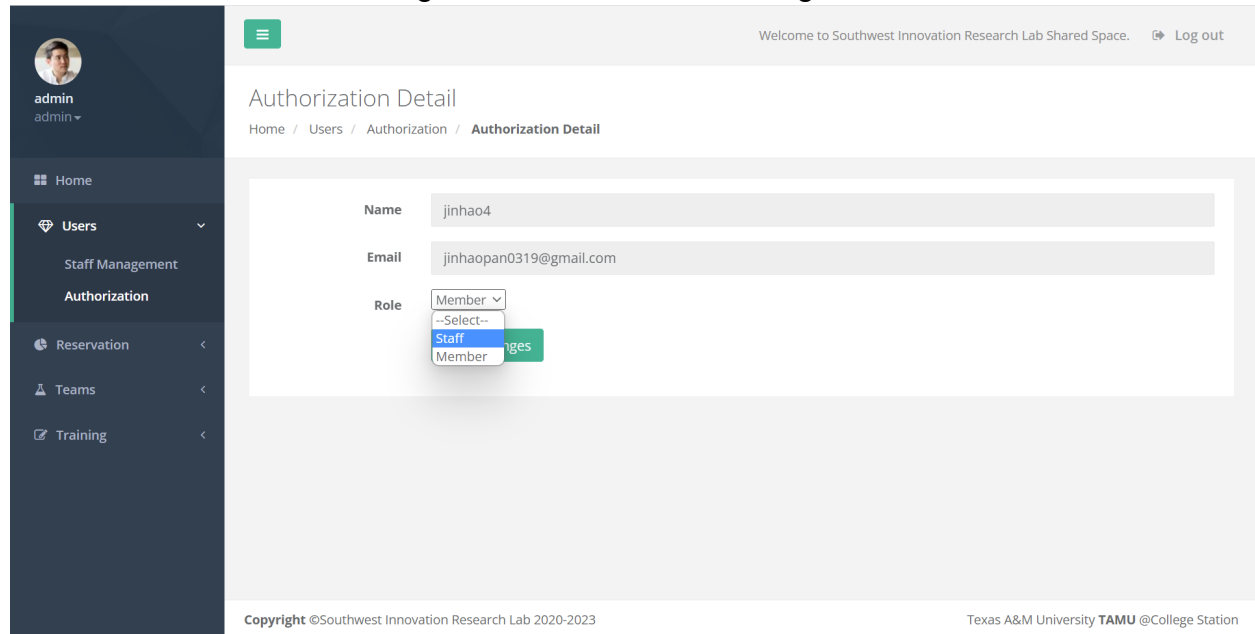
This is a 2-point medium level feature, we implemented and finished this feature via the same mechanics of creating reservation idea.



Iteration 4

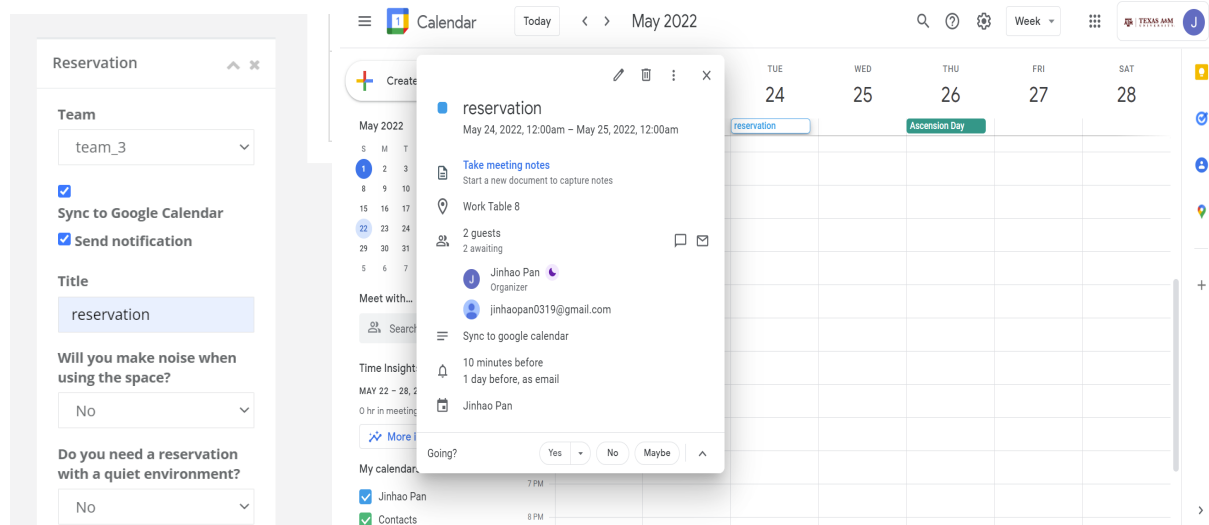
Change user roles: As an administrator, I want to change any user's role, so that I can assign some users to be the "staff" to help manage the application

This is a 1-point straightforward level feature, we implemented and finished this feature that administrator could change roles of users like adding more staffs.



Synchronizing on Google Calendar: As a team leader, I can sync the reservation I made to the Google Calendar and send invitation to all team members, so that the team can better know about the schedules.

This is a 2-point medium level feature, we implemented and finished this after we finished all major functionalities of this project, and this would enable user to receive the event information and add into the google calendar automatically.



Hand-drawn wireframes for a reservation system. The top section shows a 'Layout' with a 'Long-Term' reservation table and a 'Short-Term' reservation table. The 'Long-Term' table has columns for Area 1, Area 2, and Area 3, with dates like Feb 2 - Feb 20, Feb 25 - Mar 15, and Mar 1 - Mar 20. The 'Short-Term' table has columns for Area 1, Area 2, and Area 3, with dates like 0, 1, 12, 21, 24. Below the layout is a 'Long-Term Reserve' form with fields for Team, Desc, Start, End, and a Submit button. To the right is a 'Short-Term Reserve' form with a table for Area 1, Area 2, and Area 3, and a Submit button. At the bottom, there are two more forms: one for 'Click' with a table for Area 1, Area 2, and Area 3, and another for 'Submit' with a table for Area 1, Area 2, and Area 3.

Long-Term Reserve

Team :

Desc :

Feb 2022, Friday

1	2	3	4
11/11/11	11/11/11	11/11/11	11/11/11

→ reserved

→ normal

27. 28

Short-Term Reserve

Area 1

Area 2

Area 3

10

10

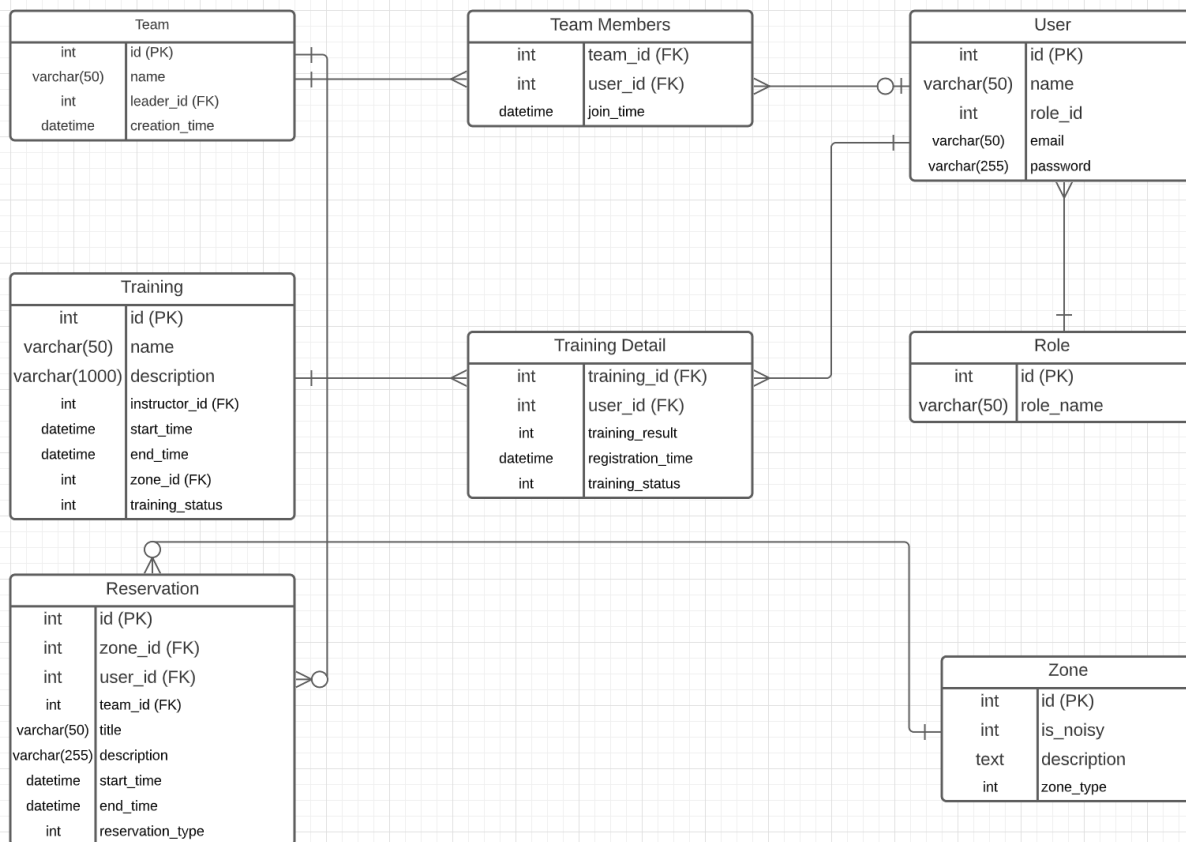
10

20

click

Area 1, to 2nd Area
From: 11:00
To: 15:00
DESC <input type="text"/>
Submit <input type="text"/>

Database ER Diagram



Team roles

Junjie Wang: Scrum Master
Yupeng Hou: Product Owner
Junyu Guo
Tianyi Liu
Jinhao Pan

In iteration 0, the scrum master was Yupeng Hou and the product owner was Junjie Wang. After that, the member roles were as above showed.

Iteration Progress

Iteration0: we created different user stories and gave points to different stores.

Iteration1: we finished the basic functionalities of the reservation module along with creating, canceling to avoid conflicts by showing them in the calendar. We finished 6 points in this iteration.

Iteration2: we added login in property this time and finished basic functionalities of the team module, which we could create teams, delete teams, and view team information. We finished 8 points in this iteration.

Iteration3: we modified the user log in property in order to classify different user roles, we finished all functionalities of the team module that we could manage team members and change team leaders, and we finished basic functionality of training that we could create training, view training and delete training. We finished 8 points in this iteration.

Iteration4: we made the functionality that administrators could change user roles and that the admin could add new staff members. And we connect google calendar API into the app so that if people successfully created a reservation, they would receive the information and put the event into their google calendar. We finished 3 points in total.

Meetings with The Customer

2/23 10:00 am, we held the first meeting with the customer, we talked about the basic requirements of this project (user stories) and the timeline of finishing this project.

3/2 10:00 am, we held a meeting with the customer to ask questions about some confusing concepts of this project and how to start our iteration 1.

3/14 10:00 am, we demoed how to create reservations and cancel reservations of this project under the Django platform.

3/28 10:15 am, we demoed creating/canceling reservations via calendar, implementing most functionalities of the team module, and adding layout of the reserving spaces.

4/11 10:15 am, we demoed the full functionality of the team module that we could show all the team details and manipulate all of them via creating, deleting and modifying. And we showed part of the training module of creating and deleting.

4/25 10:15 am, we demoed the full functionality of this project with reservation management, team management and training management with different user roles (admin, staff, team leader, and team member) and connect to the google calendar.

BDD/TDD

We use the Django testing framework to implement BDD and TDD processes. More specifically, we exploit the client object to simulate different users with different web requests and check whether the frontend behaviors and the backend updates are within expectation.

For the BDD process, we simulate four types of users to login, generate several sets of requests from the user stories, and test the behaviors on the displayed pages and the database. For example, according to the user stories, we should display a button to create teams on the Team List page for super users (admin and staff), but shouldn't when it comes to the others. As a super user, if we click the button, it should pop up a form allowing us to fill the team name and choose a user as the team leader. To test whether it behaves correctly, we simulate the login as an administrator, send a GET

request to the corresponding url and then check whether it responds with a list of normal users. However, as a normal user (neither an admin or a staff member), we won't get the list. To enhance the robustness of our system, we also test it with invalid requests. For example, a super user or team leader are enabled to invite or delete members. However, they can't add a member who is an admin or a staff member. Besides, any super user can delete any members in a team, while the team leader can't delete himself. In addition, a normal member in a team can neither invite or delete any members. To simulate those scenarios, it requires us to login as each role (admin, staff, team leader and normal member), try to invite or delete any kind of users (a super user, the leader of the team, a member other than the leader, or someone not a member in this team) and check the response error code and the behavior of the database. After generating the test cases, we modify our implementation until it passes them all. Thus, finally we have our system totally follow the user stories defined at the beginning. For the TDD process, we create unit test cases for implemented functions. We use this development mechanism mainly to check the correctness of routing process and database operations. For example, we test whether it returns the correct templates when we send a GET request to some url. Besides, whether the TeamMember model truly returns all members of a team when we call the function `TeamMember.get_team_members` with `team_id` as the input. TDD process helps us to avoid covert mistakes in commonly used functions.

The benefits of BDD/TDD: These two development models can help the developers find bugs more efficiently. First, we think of different cases more carefully when creating test cases, thus many bugs can be fixed much earlier than seeing it on the website. Besides, those tools help us check bugs automatically and persistently. It not only saves much time of manually testing, but also avoids missing some history bugs after we change the code.

The problems we met: To follow the customer's requirement, we use Django to develop the web application, which means we need to find BDD/TDD test tools ourselves. There are several professional tools like `behave_django`, however, we met problems to embed the installation and testing process in the Github Action. This is the reason why we decided only to use the `django testing framework` finally.

Configurations

Most of the configuration was included in the `reservation/settings.py`. It contains the database configurations, third-party library configurations, and application configurations.

Also, for some initial data of the database, we put it in a manually generated migration file `"0002_auto_20220314_1558.py"`. It contains the initial zone list, the role list, default admin account creation, and the social account creation for Google sign-in.

For security reason, all the secret keys/API keys/passwords are read from the environment variables. Their values are put in the encrypted

In this project, we didn't do any spikes.

We used mainly 7 branches in our project: A main branch (for release), a development branch, and 5 feature branches (for feature development) for each team member. In each iteration, we pull down the updated code from the development branch and merged to our own branch. Then, each member develop the features on its own branch. After development and testing, we will merge the feature branch to the development branch and do regression testing with the Github CI/CD. If there is any bug, we would checkout a temporary bug-fix branch, fix the bug, and merge it back to the development branch. After each iteration, we will release the current version to the main branch.

Issues with Heroku

1. When deploying on Heroku, since we are using python with Django, the database migration process needs to be embedded in the deployment script. According to the document of Heroku, the migration process can be put in the deployment file Procfile.
2. Since Heroku free tier account has 20 connections limit to PostgreSQL database and Django would use one-time short connection for each database query. It will occasionally cause connection limit exceed error. To solve this problem, in the settings.py, set the `CONN_MAX_AGE` (default is 0 which means not to use persistent connections for database queries) in DATABASES configuration. It can turn to use persistent connections for queries so that the number of connections can be reduced.
3. Since all secret keys are read from the environment variables, on Heroku, all the secrets are needed to be configured as environment variables in the application settings.

Issues with Cloud9/AWS

1. About extending the volume in AWS. While working on the project, we found that the volume size is not enough to hold it. While trying to extend the volume, one of the members met a problem that AWS only allows to enlarge the volume but not to reduce it. She asked to reduce it because she extended it to a size out of the free package. At this time, she removed the volume and tried to attach a new one, which was not a good choice because after that she found she couldn't do the attachment. The reason might be that considering the previous volume is the only one with the server, it should hold the operating system, so if it's removed, the whole server would no longer work. This experience just taught us a lesson that we should never remove the only volume on a server.

Issues in the Project

1. About unnecessary page refresh after closing a modal. During the team creation, it needs a floating dialog containing a form. We implemented this by the Modal component in jQuery. Traditionally, after clicking the submit button, the page is refreshed to fetch the updated list from the database, while after clicking the close button, it shouldn't. However, we couldn't remove the refreshment after clicking the close button. Besides, although it looked like a page refreshment, we found that it didn't actually reload the page for not fetching new data from the backend. We are still looking for solutions to this problem.

Tools

Below are the tools we used in the application:

Django-allauth (<https://django-allauth.readthedocs.io/en/latest/installation.html>)

- Provide multiple authentication approaches (including username+password and Google sign-in)

Google-api-python-client (<https://developers.google.com/calendar/api/guides/overview>)

- Used for Google sign-in and Google Calendar synchronization

Coverage (<https://coverage.readthedocs.io/en/6.3.2/>)

- Used to generate the testing covery report

Pytz, python-dateutil

- Used for process datetime data

Django_heroku

- Used for Heroku Deployment

Getting Started

****IMPORTANT****

Please note that all development guide/deployment guide/configuration files/dependencies/static assets are included in the README.md or the Github repository. Please check them in the repository.

Links

Github: <https://github.com/Einsgate/SwIRL-Shared-Space>

Pivotal tracker: <https://www.pivotaltracker.com/n/projects/2555021>

Heroku: <https://shared-space.herokuapp.com/>

Slides Video and Demo Video

<https://youtu.be/D-wiYSeXV5E>