Charter Scope and Requirements Document

Project 7: Hot-Desking App - Admin Side

University of British Columbia Okanagan

COSC 499, Summer 2022

Damyn Filipuzzi: damynfilipuzzi@gmail.com

Ravi Bullock: ravioliasb@gmail.com

Serena Chen: serenachiyin@gmail.com

Suyash Shingare: suyashshingare@gmail.com

1. Project Objective

The objective of this project is to develop an application that will allow students, staff, and visitors of the University of British Columbia to seek and reserve available desks on their campus. The development of the application is in an effort to resolve the problem of individuals not being able to find available spaces to work and study on their campus. We will be developing the administrative component of the application that allows administrators to define workspaces (buildings, floors, rooms, and desks) around their campus. Administrators will also be able to define the resources within these spaces, assign usage rules to users, and provide the users with the times when these spaces are available. Furthermore, administrators will be able to manage users by assigning user roles and relationships, deleting users, and changing user permissions. Lastly, administrators will be able to access usage statistics of workspaces to gain feedback on how space is being used and improve on space management in the future.

2. Criteria / Goals

- The website is able to run on a desktop browser
- Administrators are able to define workspaces on the website based on buildings, rooms, and desks
- Administrators are able to change user permissions
- Administrators are able to define user roles and relationships
- Administrators are able to delete users from the application
- Administrators are able to access usage statistics

3. Major Milestones

Week	Milestones
June 5th - 11th	● Create navigation mocks ● Create empty page mockups ● Create basic website
June 12th - 18th	● Create administrator login ● Design database for resources
June 19 - June 25	● Implement means for administrators to define usage policies ● Implement means for administrators to manage users (banning, user roles, etc.)
June 26 - July 2nd	● Dockerize project ● Begin set up for testing
July 03	● Prepare project status report ● Create MVP design review presentation
July 10	● Upload building, floor, and room plans for demo ● Define resources visually
July 17	● Implement means for administrators to access user statistics on the website
July 24	● Implement means for administrators to close down desks temporarily ● Allow system to email notifications to users if desk closes unexpectedly ● Allow system to email notification to user if their permissions are changed
Aug 15th - 19th	● Create final report ● Delivery final code

4. Stakeholders

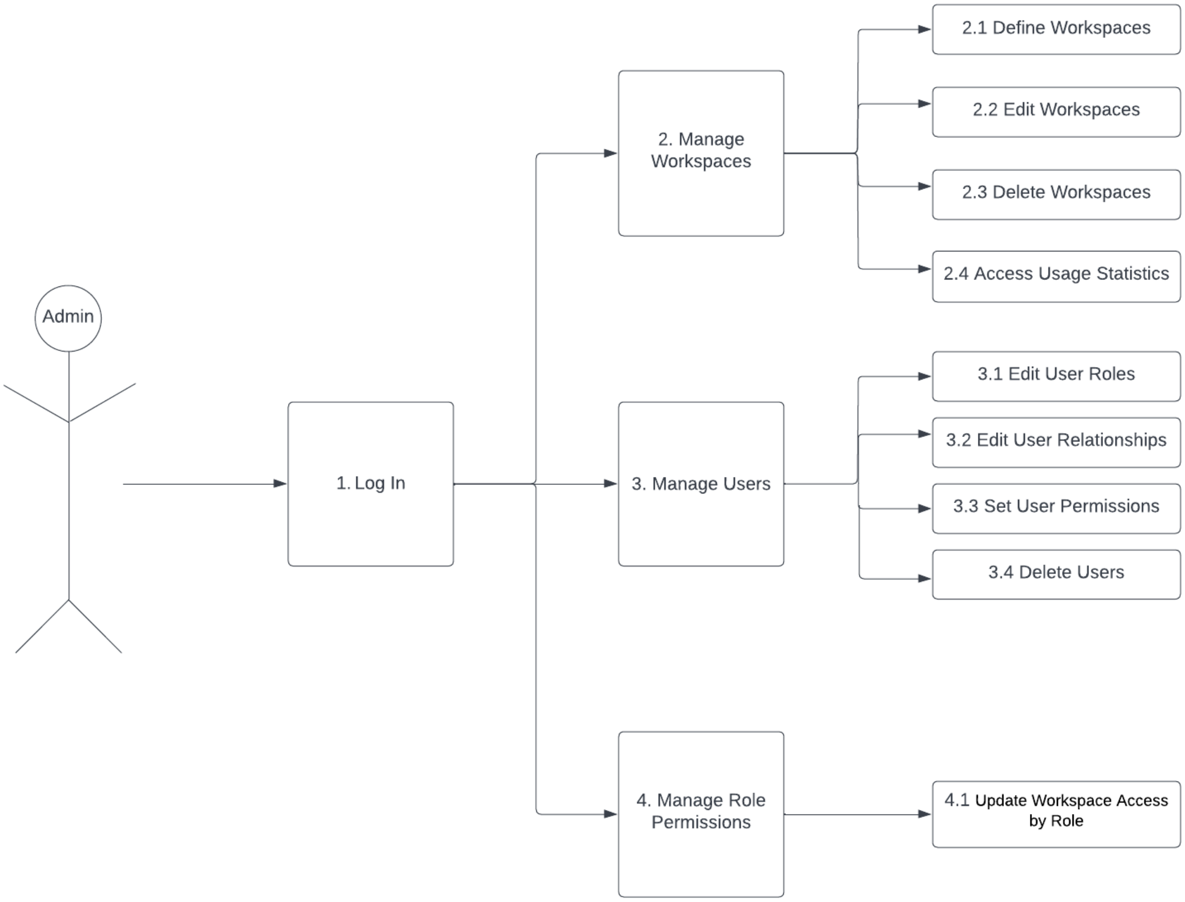
- Dean’s Office Irving K. Barber Faculty of Science
- Dr. Bowen Hui
- UBC
- Project 7 Team - Admin side
- Project 6 Team - User side
- Administrators using the Application
- Users of the Application

5. User Groups and UML Diagram

5.1 User Groups

- Administrators - Assumed to have low-technical literacy

5.2 UML Diagram



5.3 Use Case Breakdown

Use Case ID	Use Case Name	Actor(s)
1	Log in	Administrator
2	Manage Workspaces	Administrator
2.1	Define Workspaces	Administrator
2.2	Edit Workspaces	Administrator
2.3	Delete Workspaces	Administrator
2.4	Access Usage Statistics	Administrator
3	Manage Users	Administrator
3.1	Edit User Roles	Administrator
3.2	Edit User Relationships	Administrator
3.3	Set User Permissions	Administrator
3.4	Delete User	Administrator
4	Manage Role Permissions	Administrator
4.1	Update Workspace Access by Role	Administrator

Use Case ID:	1
Use Case:	Log in
Actor(s):	Administrator
Description:	Administrators will be able to log into the system with their authorized account using username and password
Pre-Condition:	● Administrator is human ● Administrator has an account ● Administrator is logged out

Post-Condition:	● Administrator is logged in ● Administrator will be able to access admin panel
Flow:	1. Administrator will click on website 2. Administrator will input their username 3. Administrator will input their password 4. Administrator will gain access to Admin Panel

Use Case ID	2
Use Case	Manage Workspaces
Actors	Administrator
Description	Administrators will be able to manage all building, rooms, desks, and resources
Pre-Condition:	● Administrator is logged in ● Administrator accesses Admin Panel
Post-Condition:	● Administrator manages buildings, rooms, desks, and resources
Flow	1. Administrator will log in 2. Administrator will access Admin Panel 3. Administrator will click on Manage Workspaces button

Use Case ID	2.1
Use Case	Define Workspaces
Actors	Administrator
Description	Administrators will be able to upload blueprints for the campus map, floor plans of buildings, and the arrangement of desks.
Pre-Condition:	● Administrator is logged in ● Administrator accesses Admin Panel
Post-Condition:	● Administrator defines workspaces based on their resources and location
Flow:	1. Administrator will log in 2. Administrator will access Admin Panel 3. Administrator will click on Manage Workspaces button

	4. Administrator will click on Define Workspace button

Use Case ID	2.2
Use Case	Edit Workspace
Actors	Administrator
Description	Administrators will be able to edit the building plans, floor plans, and room plans
Pre-Condition:	● Administrator is logged in ● Administrator accesses Admin Panel
Post-Condition:	● Administrator changes previously defined workspaces
Flow:	1. Administrator will log in 2. Administrator will access Admin Panel 3. Administrator will click on Manage Workspaces button 4. Administrator will click on Edit Workspace button

Use Case ID	2.3
Use Case	Delete Workspace
Actors	Administrator
Description	Administrator will be able to delete plans of campuses, buildings, floors, and rooms
Pre-Condition:	● User is logged in ● User accesses Admin Panel
Post-Condition:	● Workspace removed ● Empty space created
Flow:	1. Administrator will log in 2. Administrator will access Admin Panel 3. Administrator will click on Manage Workspaces button 4. Administrator will click on Delete Workspace button

Use Case ID	2.4
Use Case	Access Usage Statistics
Actors	Administrator
Description	Administrators will be able to see the usage statistics of workspaces
Pre-Condition:	● Administrator is logged in ● Administrator accesses Admin Panel
Post-Condition:	● Administrator views tables of usage statistics of each workspace
Flow:	● Administrator will login ● Administrator will access Admin Panel ● Administrator will click on Manage Campus button ● Administrator will click on Access Usage Statistics button

Use Case ID	3
Use Case	Manage Users
Actors	Administrator
Description	Administrators will be able to change user roles, relationships, and permissions. They will also be able to ban and delete users.
Pre-Condition:	● Administrator is logged in ● Administrator accesses Admin Panel
Post-Condition:	● Administrator edits information on users
Flow:	1. Administrator will log in 2. Administrator will access Admin Panel 3. Administrator will click on Manage Users button

Use Case ID	3.1
Use Case	Edit User Role
Actors	Administrator
Description	Administrator can edit user role as being administrator, undergraduate student, graduate student, or staff member
Pre-Condition:	● Administrator is logged in ● Administrator accesses Admin Panel
Post-Condition:	● Administrator edits user roles
Flow:	1. Administrator will log in 2. Administrator will access Admin Panel 3. Administrator will click on Manage Users button 4. Administrator will search for a certain user 5. Administrator will click on Edit User Role button

Use Case ID	3.2
Use Case	Edit User Relationships
Actors	Administrator
Description	Administrator can edit user relationships, such as if they have a supervisor
Pre-Condition:	● Administrator is logged in ● Administrator accesses Admin Panel
Post-Condition:	● Administrator edits user relationships
Flow:	1. Administrator will log in 2. Administrator will access Admin Panel 3. Administrator will click on Manage Users button 4. Administrator will search for certain user 5. Administrator will click on Edit User Relationship button

Use Case ID	3.3
Use Case	Set User Permissions
Actors	Administrator
Description	Administrator can set permissions for users to restrict or open access for users
Pre-Condition:	● Administrator is logged in ● Administrator accesses Admin Panel
Post-Condition:	● Administrator restricts users from using certain workspaces ● Administrator allows users to use certain workspaces
Flow:	1. Administrator will log in 2. Administrator will access Admin Panel 3. Administrator will click on Manage Users button 4. Administrator will search for certain user 5. Administrator will click on Set User Permissions button

Use Case ID	3.4
Use Case	Delete Users
Actors	Administrator
Description	Administrator can delete users from the database
Pre-Condition:	● Administrator is logged in ● Administrator accesses Admin Panel
Post-Condition:	● Administrator deletes user from database
Flow:	1. Administrator will log in 2. Administrator will access Admin Panel 3. Administrator will click on Manage Users button 4. Administrator will search for certain user 5. Administrator will click on Delete User button

Use Case ID	4
Use Case	Manage Role Permissions
Actors	Administrator
Description	Administrator can change the permissions that each role of admin, undergraduate student, graduate student, and staff member has
Pre-Condition:	● Administrator is logged in ● Administrator accesses Admin Panel
Post-Condition:	● Admin grants permissions based on user roles
Flow:	1. Administrator will log in 2. Administrator will access Admin Panel 3. Administrator will click on Manage Role Permissions button

Use Case ID	4.1
Use Case	Update Workspace Access by Role
Actors	Administrator
Description	Administrator updates permissions to restrict or open up access to workspace based on user role
Pre-Condition:	● Administrator is logged in ● Administrator accesses Admin Panel
Post-Condition:	● Administrator will grant permissions to certain roles to access certain workspace ● Administrator will restrict permissions to certain roles to access certain workspaces
Flow:	1. Administrator will log in 2. Administrator will access Admin Panel 3. Administrator will click on Manage Role Permissions button 4. Administrator will click on specific role 5. Administrator will click on Restrict/Grant Access button

6. Requirements

6.1 Functional requirements

- The system stores data on resources, desks, rooms, and buildings
- The system will handle requests made by administrators
- The system will be able to generate workspace usage statistics
- The system will show desks on a room map
- The system must allow for the addition of a building map
- The system allows for the creation of desk resources
- The system will communicate with the database to store and receive information
- The system will send email alerts to users if a booked desk becomes unavailable
- The system will send email alerts to users if they are banned or unbanned

6.2 Non-functional requirements

- First Deliverables are done by the end of June
- Final Milestone done by the end of August
- Website must be secure and easy to navigate
- The project will be dockerized

6.3 Technical Requirements

- MySQL for database query language

- Website will be optimized for desktop use (Admin side)

6.4 User requirements

- Administrators will be able to operate the website with low technical literacy
- Administrators will define desks, rooms, and buildings
- Administrators will define the resources in the room
- Administrators will manage workspaces by adding desks, resources, furniture, etc. to a room
- Administrators will associate policies around bookings
- Administrators will set how far in advance the user can make a booking.
- Administrators will set how many bookings a user can make within a certain time frame.
- Administrators will set the duration of a booking.
- Administrators will generate reports- i.e. tables, graphs, tallies- based on space utilization by rooms and desks.
- Administrators will define the relationship between users.
- Administrators will define user groups.
- Administrators will manage users - i.e. faculties/departments- info booking
- Administrators will attach users to departments.
- Administrators will override user set info.
- Administrators will set permissions for certain users.
- Administrators will be able to ban users by stopping them from booking

6.5 Stretch Goals

- The application will be optimized to work on mobile devices

7. Assumptions and Constraints

7.1 Assumptions

- Team members will each have at least one functional device to work on for the project
- Team members will have knowledge on project management
- Team members will complete project within the declared timeframe
- Team members will not encounter computer issues
- Team member will have proper support

7.2 Constraints

- There is very limited time during the summer session to complete project
- Some team members have limited knowledge on web development
- Project sponsor will have limited time to meet with team members

8. High-level risks

- Poor synergy between two teams working on the project
- Merge conflicts
- Requirements changing
- A team member drops out of the course
- A team member gets sick

- A team member goes on vacation
- Website goes offline

9. Development Process

A Kanban development process will be used and managed with Github Projects:

1. Weekly in-person progress meetings on Tuesdays and Fridays with the client and Group 6
2. Weekly code reviews on Wednesdays
3. Review the completed tasks for the application
4. Plan the next tasks using the backlog
5. Release demos

10. Work Breakdown Structure

	Estimated Hours and Assignment			
First stage (Gathering requirements)	**Damyn**	**Suyash**	**Serena**	**Ravi**
Gather Project Requirements	6	6	6	6
Charter and Scope Draft	3	3	3	3
UML, ER Diagram	4	4	4	4
Total Number of Hours Assigned Per Team Member	**13**	**13**	**13**	**13**

Weekly Average of Hours Assigned Per Team Member	**1.8**	**1.8**	**1.8**	**1.8**

	Estimated Hours and Assignment			
Database Implementation	**Damyn**	**Suyash**	**Serena**	**Ravi**
Migration creation	6	1	1	1
Updating tables	4	4	4	4
Creating seeders for database	6	1	3	1
Total Number of Hours Assigned Per Team Member	**16**	**6**	**8**	**6**
Weekly Average of Hours Assigned Per Team Member	**2.3**	**0.9**	**1.1**	**0.9**

	Estimated Hours and Assignment			
Website	**Damyn**	**Suyash**	**Serena**	**Ravi**
Setup framework and docker	10	0	0	0
Create user login/registration	1	1	1	1

Admin panel	6	6	6	6
User Management	0	4	0	4
Data Management	8	4	8	4
Usage Statistics	2	3	3	3
Input Form for building/rooms/desks	3	3	3	3
Data Visualization Tool (Plotting on a map)	5	5	5	5
Total Number of Hours Assigned Per Team Member	**35**	**26**	**26**	**26**
Weekly Average of Hours Assigned Per Team Member	**5**	**3.7**	**3.7**	**3.7**

	Estimated Hours and Assignment			
Testing	**Damyn**	**Suyash**	**Serena**	**Ravi**
Database Testing	4	4	4	4
Unit Testing	4	4	4	4

Integration Testing	4	4	4	4
Total Number of Hours Assigned Per Team Member	**12**	**12**	**12**	**12**
Weekly Average of Hours Assigned Per Team Member	**1.7**	**1.7**	**1.7**	**1.7**

	Estimated Hours and Assignment			
Prototyping	**Damyn**	**Suyash**	**Serena**	**Ravi**
Create Prototypes	5	5	5	5
Release Prototypes with Documentation	1	1	1	1
Total Number of Hours Assigned Per Team Member	**6**	**6**	**6**	**6**
Weekly Average of Hours Assigned Per Team Member	**0.9**	**0.9**	**0.9**	**0.9**

	Estimated Hours and Assignment			
Documentation and Presentation	**Damyn**	**Suyash**	**Serena**	**Ravi**
Review README files on Github	1	1	1	1
Review comments on code	2	2	2	2
Create PowerPoint Presentation on Project	4	4	4	4
Total Number of Hours Assigned Per Team Member	**6**	**6**	**6**	**6**
Weekly Average of Hours Assigned Per Team Member	**0.9**	**0.9**	**0.9**	**0.9**

	Summary			
	Damyn	**Suyash**	**Serena**	**Ravi**
Total Weekly Average of Hours Assigned Per Team Member	**12.6**	**9.9**	**10.1**	**9.9**

11. Approvals

Project Sponsor **Signature** **Date**

Project Manager **Signature** **Date**