

Pandemic Pal

External Requirements

Team Explorer (ICBC 10)

Version 0.1

February 4, 2021

Stakeholders

Team Members

Sylvester Chenne
David Huang
Felix Law
TaeGyun Lim
Sohan Mathew
Nicholas Pun
Michael Zhang
Ray Zhang

Sponsors

Jaslyn Nguyen
Matt Dockerty

TA

Asem Ghaleb

Revision History

Version Number	Date	Comments
Version 0.1	February 4, 2021	Initial version

Sign-Off

ICBC Sponsor	Date	Signature
Jaslyn Nguyen	_____	_____
Matt Dockerty	_____	_____

Table of Contents

General Problem Statement	3
Business Requirement Summary	3
Other Systems Impacted By This Project	4
Groups/Individuals Impacted By This Project	4
Business Constraints/Assumptions/Dependencies	4
Requirement Traceability Matrix	4
Use Case Diagram	7
Functional Requirements / Use Cases	10
Non-Functional Requirements	21

General Problem Statement

The goal of this project is to design and implement a web application to be used by ICBC employees and administrators. The functionality of the web application can be classified into three modules: a desk reservation system, an online social forum and a mechanism to manage pickup of physical mail. These three services are available to all users. In addition, the application places additional privileges/responsibilities upon administrators, which mainly relate to being able to change the content that is visible to regular users.

The required steps for successful completion of this project can be roughly split into the following.

- 1) Constructing a browser-based interface that facilitates usage of the 3 modules described above. Supported browsers should include the Microsoft Edge desktop browser and Safari mobile browser.
- 2) Constructing a robust backend system that implements the functionality of the 3 modules
- 3) Creating a database system to be integrated with our backend. The database system enables persistent storage of important information generated by our application, such as reservation times and social forum posts.

Business Requirement Summary

Below, we outline the business requirements in terms of the three modules described in the above section. The complete details are provided in the use cases under the Functional Requirements section of this document

(Module 1) Reservation:

- All users must be able to secure an available reservation for an office desk space within the next 6 months
- All users must be able to redact their own reservations
- All users must be able to view their own reservations
- Admins must be able to upload new floor plan layouts
- Admins must be able to update floor plan layouts
- Admins must be able change the list of buildings/associated floors/associated desks presented to all users

(Module 2) Discussion Forum / News Updates:

- All users can see and post comments in discussion posts
- All users can see news updates such as company announcements and wellness tips

- Admins must be able to add / delete new discussion posts / categories and news updates

(Module 3) Mail Concierge:

- All users can create and manage a physical mail request
- Admins must be able to view and modify mail requests made from users

Other Systems Impacted By This Project

If Pandemic Pal is deployed by ICBC, it will be connected to their database. Their database could be disrupted by the information being uploaded from Pandemic Pal. For example, if the discussion boards have many posts and comments, this could take up a fair amount of memory in the database. We do not anticipate this project bearing significant load on ICBC's internal network capacities.

Groups/Individuals Impacted By This Project

Pandemic Pal will be used by ICBC employees and administrators. For employees, it will make managing workspace hours more efficient and flexible. For administrators, this provides more organized and transparent details of working hours.

Business Constraints/Assumptions/Dependencies

We are required to integrate our application with Microsoft Active Directory to implement a single sign-in step. An assumption given to us by the sponsors is that all potential users of our application will already have an account through this service. Consequently, our application will not have a “Create an account” feature.

The functionality categorized by “Module 2”, as described in the Business Requirement Summary of this document, includes allowing users to receive notifications relevant to their office location, COVID-19 or general wellness. As instructed by the sponsor, it is assumed that the administrator would be the publisher of these notifications.

Requirement Traceability Matrix

Each row below encodes a specific requirement of the project. For further information regarding a particular row/requirement, consult its “Type” attribute. If the attribute value

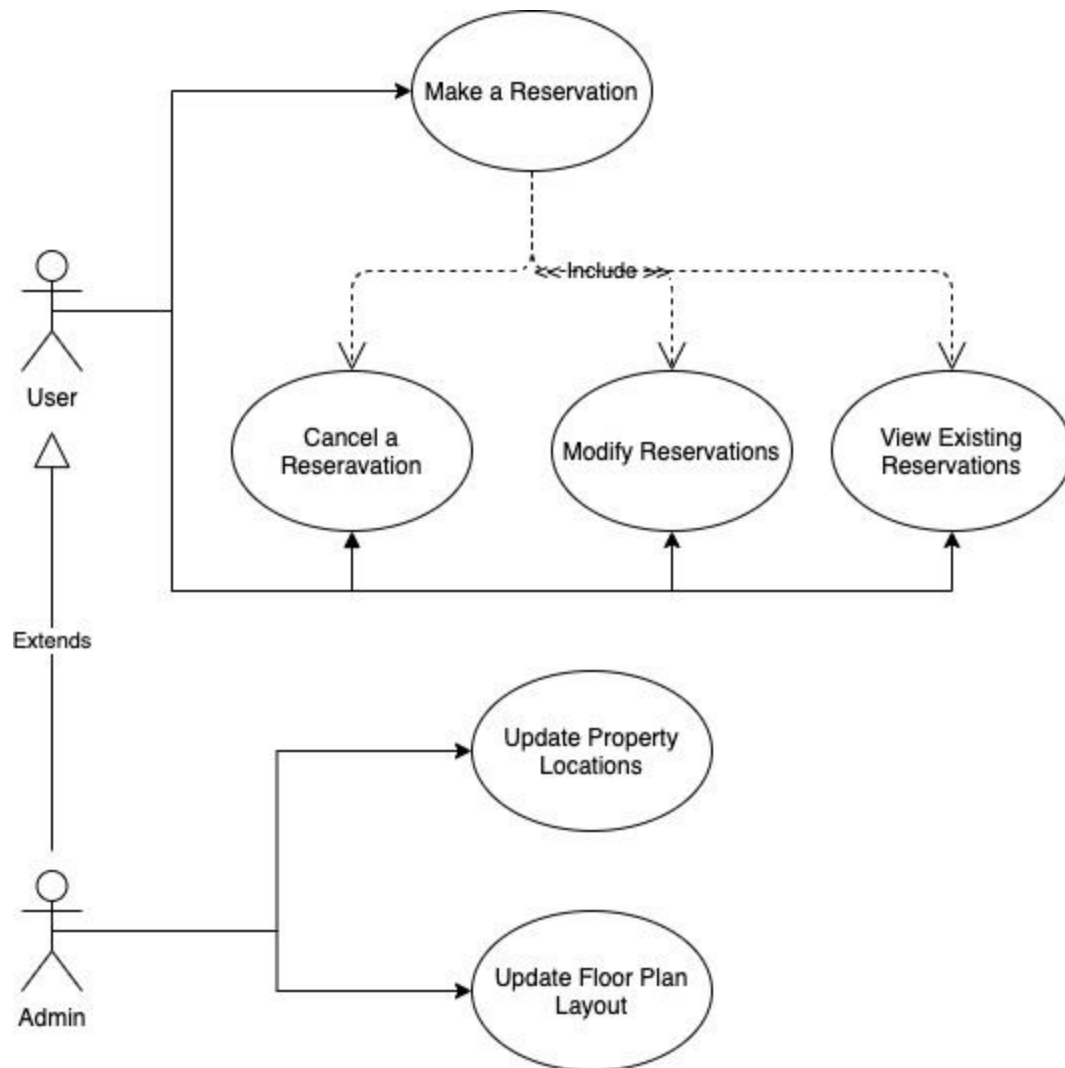
is “functional”, see the “Functional Requirements / Use Cases” section of this document.
Otherwise, see the “Non-Functional Requirements” section.

REQ No.	Reference	Requirement Description	Type	Priority	MVP (Y/N)	Related Test Case	Test Status	Comments
1.0.0	Module 1 - use case 1	Make a Reservation(s)	Functional	1	Y	TBD	TBD	
1.1.0	Module 1 - use case 2	Cancel a Reservation(s)	Functional	1	Y	TBD	TBD	
1.2.0	Module 1 - use case 3	View Existing Reservation(s)	Functional	2	Y	TBD	TBD	
1.3.0	Module 1 - use case 4	View All Existing Reservations	Functional	2	Y	TBD	TBD	
1.4.0	Module 1 - use case 5	Update Property Location(s)	Functional	1	Y	TBD	TBD	
1.5.0	Module 1 - use case 6	Update Floor Plan Layouts	Functional	1	Y	TBD	TBD	
1.6.0	Module 1 - use case 7	Optional : Modify Reservation	Functional	3	N	TBD	TBD	
1.7.0	Module 1 - use case 8	Optional: Generate Outlook entries for reservations	Functional	3	N	TBD	TBD	
2.0.0	Module 2 - use case 1	Comment on Post	Functional	1	Y	TBD	TBD	
2.1.0	Module 2 - use case 2	Create Discussion Categories	Functional	1	Y	TBD	TBD	
2.2.0	Module 2 - use case 3	Create Discussion Post	Functional	1	Y	TBD	TBD	
2.3.0	Module 2 - use case 4	Delete Discussion Comments	Functional	2	Y	TBD	TBD	
2.4.0	Module 2 - use case 5	Delete Discussion Category	Functional	2	Y	TBD	TBD	
2.5.0	Module 2 - use case 6	Delete Discussion Post	Functional	2	Y	TBD	TBD	

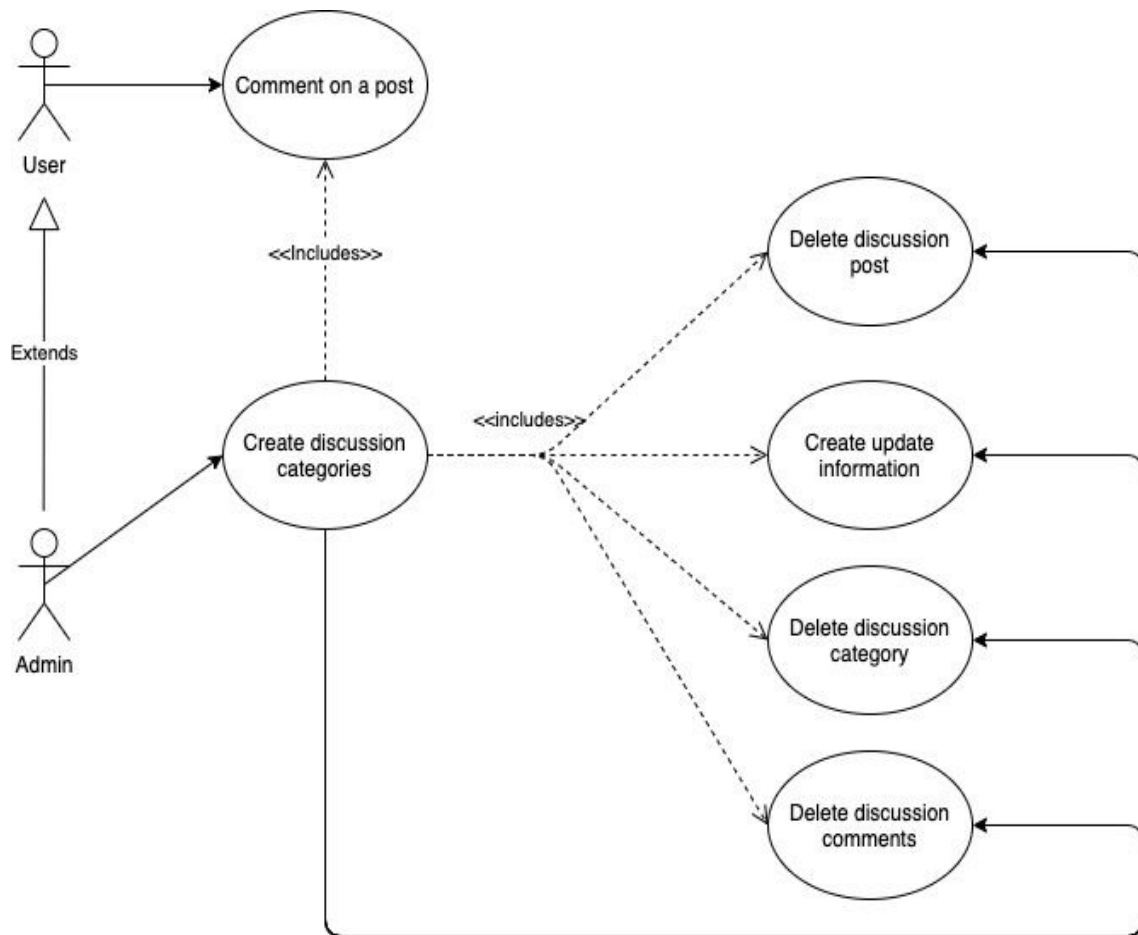
2.6.0	Module 2 - use case 7	Create Update Information	Functional	1	Y	TBD	TBD	
3.0.0	Module 3 - use case 1	Submit Mail Management Request	Functional	1	Y	TBD	TBD	
3.1.0	Module 3 - use case 2	Modify / Close Own Mail Management Request(s)	Functional	2	Y	TBD	TBD	
3.2.0	Module 3 - use case 3	Enter Mail Into System	Functional	1	Y	TBD	TBD	
3.3.0	Module 3 - use case 4	View, Action and / or Provide Feedback on Mail Request	Functional	2	Y	TBD	TBD	
4.0.0	NFR - Performance	Perform At Consistent Speed	Non- Functional	3	Y	TBD	TBD	
4.1.0	NFR - Performance	Handle 50 Concurrent Users	Non- Functional	3	Y	TBD	TBD	
4.2.0	NFR - Performance	Handle 1000 Live Registrations	Non- Functional	3	Y	TBD	TBD	
4.3.0	NFR - Performance	DBA Script to Archive Old Registration Data	Non- Functional	3	Y	TBD	TBD	
4.4.0	NFR - UI	Support for Microsoft Edge	Non- Functional	1	Y	TBD	TBD	
4.5.0	NFR - UI	Support Desktop and Mobile	Non- Functional	1	Y	TBD	TBD	
4.6.0	NFR - UI	Optional: Link map to Clickable Hot-Spot	Non- Functional	3	N	TBD	TBD	

Use Case Diagrams

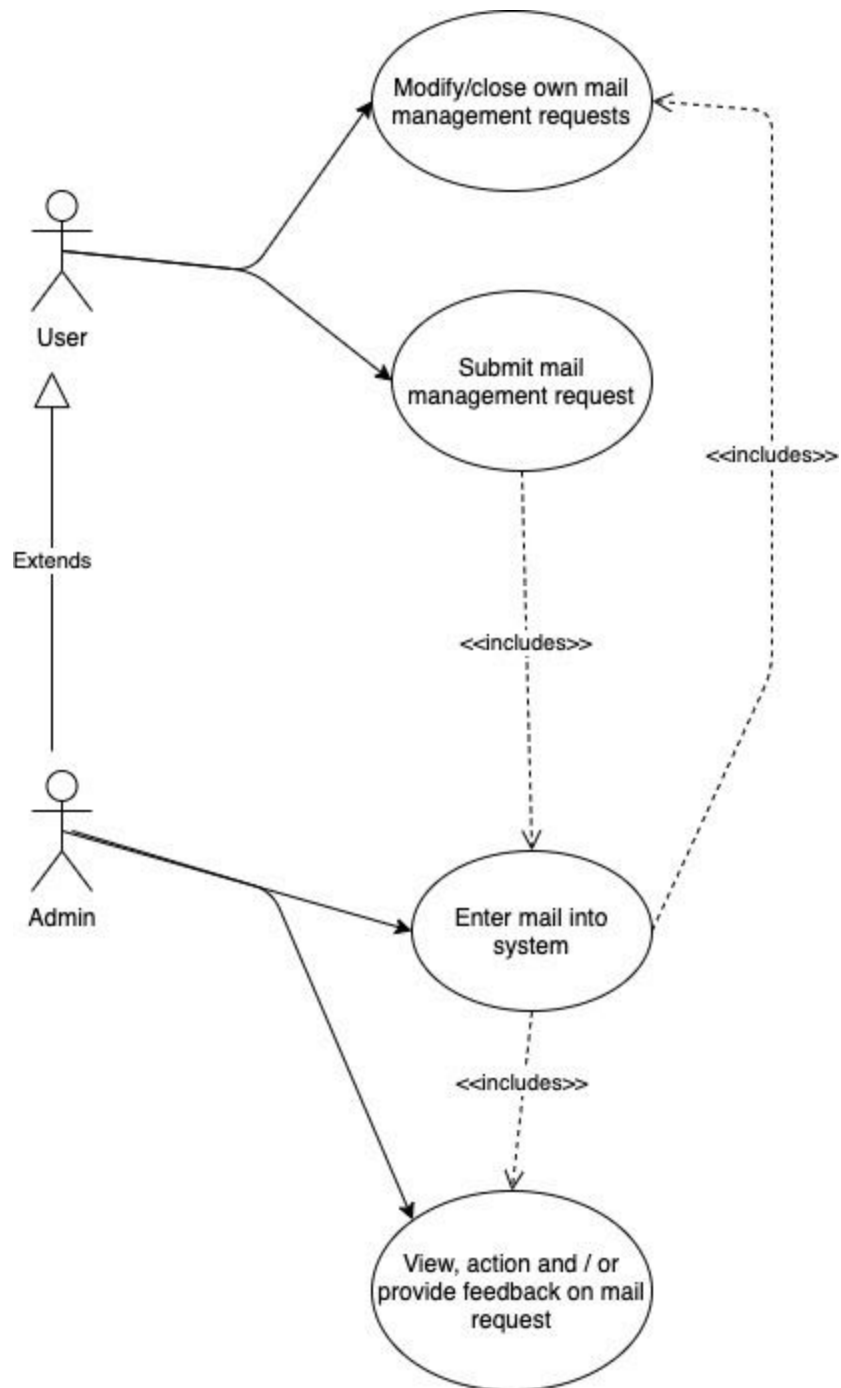
1) Module #1: “Where I’m Working”



2) Module #2: “Staying Connected”



3) Module #3: “Mail Management Concierge”



Functional Requirements / Use Cases

Preliminary Definitions:

A **user** refers to a regular user of the application, ie. an ICBC employee.

An **admin** refers to an administrator of the application. The decision of who should be appointed as an administrator within ICBC is left to the sponsor. In addition to being able to use the application as a regular user, admins are also able to control what other users see on the application (see use cases below for more details)

Module 1:

Make a Reservation(s)	
Description	A user should be able to make a reservation for their preferred work location (building, floor, workstation) for any number of days
Primary Actor	User
Preconditions	<ul style="list-style-type: none">- A user is logged into the system- All dates the user is reserving for are within the next 6 months
Postconditions	<ul style="list-style-type: none">- If all selected dates are available, a new reservation record is created within the system with the supplied work location and time.
Main Scenario	<ol style="list-style-type: none">1. User selects "Create a Reservation"2. App. displays "New Reservation" module<ol style="list-style-type: none">a. User selects locationb. User selects available reservation date(s)3. User submits reservation4. App. creates reservation5. App. displays successful feedback
Alternate Scenarios	<ul style="list-style-type: none">- (2a) User selects reservation date(s) first<ul style="list-style-type: none">- Then (2b) user would select from available locations second- (5) User submitted incomplete reservation<ul style="list-style-type: none">- App. displays unsuccessful feedback- (5) Desk could not be reserved due to conflicting reservations<ul style="list-style-type: none">- App. displays unsuccessful feedback

Cancel a Reservation	
Description	A user should be able to cancel a reservation they previously made
Primary Actor	User
Preconditions	<ul style="list-style-type: none"> - A user is logged into the system - User has a pre-existing reservation
Postconditions	<ul style="list-style-type: none"> - The selected reservation is cancelled, allowing others to reserve it
Main Scenario	<ol style="list-style-type: none"> 1. User selects "View Reservation" 2. App. displays "View Reservation" module 3. User selects "Cancel Reservation" <ol style="list-style-type: none"> a. User selects reserved dates 4. User submits reservation cancellation 5. App. cancels reservation(s) 6. App. displays successful feedback
Alternate Scenarios	<ul style="list-style-type: none"> - (4) User selected no dates <ul style="list-style-type: none"> - Submission button is not clickable - (6) Cancellation was unsuccessful <ul style="list-style-type: none"> - App. displays unsuccessful feedback

View Existing Reservation(s)	
Description	A user should be able to view all upcoming reservations they made
Primary Actor	User
Preconditions	<ul style="list-style-type: none"> - A user is logged into the system
Postconditions	<ul style="list-style-type: none"> - The user must be able to see a list of all upcoming reservations, including reservation details such as date, time and location of each reservation
Main Scenario	<ol style="list-style-type: none"> 1. User selects "View Reservations" 2. App. displays "View Reservation" module 3. App. displays a list of reservations along with reservation details
Alternate	<ul style="list-style-type: none"> - (3a) User has no reservations

Scenarios	- App. displays “No reservations” message
------------------	---

View All Existing Reservation(s)	
Description	Allows users to view all existing reservations to understand how busy the office space may get on particular dates / locations.
Primary Actor	User
Preconditions	- A user is logged into the system
Postconditions	- For a given number of dates selected by the user, they can see the number of other people who have reserved for the same date and building and floor
Main Scenario	<ol style="list-style-type: none"> 1. From a calendar interface, the user selects a number of dates that they are interested in. 2. The user specifies a particular building and floor. 3. The app displays how many other reservations exist for the selected building and floor, for each selected date.
Alternate Scenarios	N/A

Update Property Location(s)	
Description	The admin may change the list of buildings/associated floors/associated desks that a regular user is able to see when making a reservation
Primary Actor	Admin
Preconditions	- The admin is logged into the system
Postconditions	<ul style="list-style-type: none"> - Building names may be added or removed from the list of reservable buildings - If a building name is added, the user will be able to see the list of its associated floors and desk locations when making a reservation for that building - If a building name is removed, all users will be unable to make reservations for that building. Users with already existing reservations for that building are not affected.

Main Scenario	<i>(Assumes admin is adding a building)</i> <ol style="list-style-type: none"> 1. Admin selects 'Update Property Locations' 2. Admin sees list of current reservable buildings. . 3. Admin specifies the name, address and number of floors of the building to be added. 4. Admin confirms add, making them visible to users.
Alternate Scenarios	<i>(Assumes admin is removing a building)</i> <ol style="list-style-type: none"> 1. Admin selects 'Update Property Locations' 2. Admin sees list of current reservable buildings. . 3. Admin chooses a building to remove from list. 4. Admin confirms removal, making them visible to users.

Update Floor Plan Layouts	
Description	An admin should be able to upload new floor plan layouts for specific buildings and floors as JPG files.
Primary Actor	Admin
Preconditions	<ul style="list-style-type: none"> - The file format must be JPG or JPEG - The file size must be less than 10Mb - The user signed in must have admin access - The uploaded image must have labelled sections (non-technical)
Postconditions	<ul style="list-style-type: none"> - The floor plan is now saved within the database and the user will now be able to view it when making a reservation
Main Scenario	<ol style="list-style-type: none"> 1. Admin selects "Update Floor Plans" 2. Admin selects the building and floor number in which they want to update the plan for 3. Current floor plan is visible to Admin 4. Admin uploads image of floor plan 5. Image is displayed on screen and admin confirms that this is the image they want to use 6. Admin is notified that the floor plan for the floor number that they selected has been successfully updated.
Alternate Scenarios	<ul style="list-style-type: none"> - (4) Updated wrong file type <ul style="list-style-type: none"> - Admin is notified with an error message

Optional : Modify Reservation	
Description	A user should be able to modify a reservation they previously made
Primary Actor	User
Preconditions	<ul style="list-style-type: none"> - A user is logged into the system - User has a pre-existing reservation
Postconditions	<ul style="list-style-type: none"> - The selected reservation is modified
Main Scenario	<ol style="list-style-type: none"> 1. User selects "View Reservation" 2. App. displays "View Reservation" module 3. User selects "Edit Reservation" <ol style="list-style-type: none"> a. User selects reserved dates b. User selects new location 4. User submits reservation edit 5. App. edit reservation(s) 6. App. displays successful feedback
Alternate Scenarios	<ul style="list-style-type: none"> - (4) User created no changes <ul style="list-style-type: none"> - Submission button is not clickable - (6) Cancellation was unsuccessful <ul style="list-style-type: none"> - App. displays unsuccessful feedback

Optional : Generate Outlook entries for Reservations	
Description	A user who has reserved a desk will have a Microsoft Outlook calendar entry/entries generated for the corresponding date(s)
Primary Actor	User
Preconditions	<ul style="list-style-type: none"> - User is logged into the system - User has made a reservation successfully
Postconditions	<ul style="list-style-type: none"> - User has a new Outlook calendar entry
Main Scenario	<ol style="list-style-type: none"> 1. User successfully completes "Make a Reservation(s)" use case. 2. For each reservation made by the user, an entry for the corresponding date appears in the user's Outlook calendar.
Alternate	N/A

Scenarios	
------------------	--

Module 2:

Comment on Post	
Description	A user should be able to comment on an existing post
Primary	User
Preconditions	<ul style="list-style-type: none"> - The user is logged into the system - A discussion post exists that the user wishes to comment on
Postconditions	<ul style="list-style-type: none"> - The new comment is added, and visible under the discussion post
Main Scenario	<ol style="list-style-type: none"> 1. User selects a post 2. User selects "make reply" 3. App. displays a "reply" text entry box 4. User fills in reply 5. User selects "post reply" 6. App. adds the reply with user's name under the discussion post
Alternate Scenarios	<ul style="list-style-type: none"> - (4) User leaves box empty <ul style="list-style-type: none"> - App. prompts to fill in reply field

Create Discussion Categories	
Description	An admin should be able to create a discussion category for all users to be able to create posts under.
Primary Actor	User
Preconditions	<ul style="list-style-type: none"> - Administrator is logged into system - Name of category created does not exist as of the time of creation
Postconditions	<ul style="list-style-type: none"> - Discussion category is created
Main Scenario	<ol style="list-style-type: none"> 1. Admin clicks on "Create Discussion Category"

	<ol style="list-style-type: none"> 2. Admin fills out required fields for creating a discussion <ol style="list-style-type: none"> a. Name 3. Discussion category is created
Alternate Scenarios	<ul style="list-style-type: none"> - (3) Required fields are not fully filled out <ul style="list-style-type: none"> - Admin is prompted to fill out said required fields - (3) Discussion category name already exists <ul style="list-style-type: none"> - Admin is prompted to change the discussion category name to another name that is not in use

Create Discussion Post	
Description	A user should be able to create a discussion post for all users to be able to comment under.
Primary Actor	User
Preconditions	<ul style="list-style-type: none"> - User is logged into system
Postconditions	<ul style="list-style-type: none"> - Discussion post is created
Main Scenario	<ol style="list-style-type: none"> 1. User clicks on "Create Discussion Post" 2. User fills out required fields for creating a discussion post <ol style="list-style-type: none"> a. Name 3. Discussion post is created
Alternate Scenarios	<ul style="list-style-type: none"> - (3) Required fields are not fully filled out <ul style="list-style-type: none"> - User is prompted to fill out said required fields

Delete Discussion Comments	
Description	An admin can delete a discussion comment if the admin deems it as inappropriate, etc.
Primary Actor	Admin
Preconditions	<ul style="list-style-type: none"> - Admin is logged into the system - A discussion comment must exist
Postconditions	<ul style="list-style-type: none"> - The targeted discussion comment
Main Scenario	<ol style="list-style-type: none"> 1. Admin clicks on "Delete Discussion Comment"

	<ol style="list-style-type: none"> 2. System asks for confirmation 3. Targeted discussion comment is deleted
Alternate Scenarios	<ul style="list-style-type: none"> - (3) Discussion comment is deleted by another admin right before action is confirmed <ul style="list-style-type: none"> - Error Message is displayed notifying the admin that the discussion comment has been deleted already

Delete Discussion Category	
Description	An admin can delete a discussion category if the admin deems it as inappropriate, etc.
Primary Actor	Admin
Preconditions	<ul style="list-style-type: none"> - Admin is logged into the system - A discussion category must exist
Postconditions	<ul style="list-style-type: none"> - The targeted discussion category, all of its discussion posts and all of its respective comments are deleted
Main Scenario	<ol style="list-style-type: none"> 1. Admin clicks on "Delete Discussion Category" 2. System asks for confirmation 3. Targeted discussion category is deleted
Alternate Scenarios	<ul style="list-style-type: none"> - (3) Discussion category is deleted by another admin right before action is confirmed <ul style="list-style-type: none"> - Error message is displayed notifying the admin that the discussion post has been deleted already

Delete Discussion Post	
Description	An admin can delete a discussion posting if the admin deems it as inappropriate, etc.
Primary Actor	Admin
Preconditions	<ul style="list-style-type: none"> - Administrator is logged into the system - A discussion post must exist within a discussion category
Postconditions	<ul style="list-style-type: none"> - The targeted discussion post and all of its comments are

	deleted and cannot be accessed any longer
Main Scenario	<ol style="list-style-type: none"> 1. Admin clicks on “Delete Discussion Post” 2. System asks for confirmation of the action 3. Targeted discussion post is deleted
Alternate Scenarios	<ul style="list-style-type: none"> - (3) Discussion post is deleted by another admin right before action is confirmed <ul style="list-style-type: none"> - Error message is displayed notifying the admin that the discussion post has been deleted already

Create Update Information	
Description	An administrator updates the information and tips to be shown to all users (location relevant, COVID news, wellness tips)
Primary Actor	Admin
Preconditions	<ul style="list-style-type: none"> - Administrator is logged into system
Postconditions	<ul style="list-style-type: none"> - New updates are pushed to the top of alerts and tips as visible on homepage
Main Scenario	<ol style="list-style-type: none"> 1. Administrator selects “Create New Alert/Tip” 2. Administrator fills in relevant fields <ol style="list-style-type: none"> a. Category of information b. Body text of information 3. Administrator selects “Submit Alert/Tip” 4. App. updates new alert/tip to be on top of Alerts/Tips visible on homepage
Alternate Scenarios	<ul style="list-style-type: none"> - (2a, 2b) Administrator does not fill in all necessary fields <ul style="list-style-type: none"> - Error message is displayed if Administrator attempts to submit without filling them in

Module 3:

Submit Mail Management Request	
Description	A user wants their physical mail to be managed by a colleague, and submits a request detailing what they want.

Primary Actor	User
Preconditions	<ul style="list-style-type: none"> - User is logged into system - Mail management request is not already open
Postconditions	<ul style="list-style-type: none"> - A mail management request is created, and visible to administrators
Main Scenario	<ol style="list-style-type: none"> 1. User selects "Handle Mail" in order to be in the system for their mails to be managed 2. User fills in relevant fields <ol style="list-style-type: none"> a. instruction type b. forwarding address if choosing "forward" 3. User selects "Submit Request"
Alternate Scenarios	<ul style="list-style-type: none"> - (2) Required fields are not filled out <ul style="list-style-type: none"> - The app prompts the user or admin to submit all the required information

Modify / Close Own Mail Management Requests	
Description	A user can modify / close their request for mail management given that the request is still open.
Primary Actor	User
Preconditions	<ul style="list-style-type: none"> - The user is logged in - The request belongs to the user that is logged in - The request needs to be still open
Postconditions	<ul style="list-style-type: none"> - Request is modified or closed
Main Scenario	<ol style="list-style-type: none"> 1. The requests that the user or admin made for mail handling is still open 2. The user can then choose to modify or close <ol style="list-style-type: none"> a. If the user wishes to modify, they can make changes to their request b. If the user wishes to close their request, the request will be closed 3. The requests are modified or closed
Alternate Scenarios	<ul style="list-style-type: none"> - (1) The request is already closed <ul style="list-style-type: none"> - The request can't be modified or closed

Enter mail into system	
Description	New mail (or package) arrives at ICBC, and an administrator picking it up wishes to alert its targeted recipient
Primary Actor	Admin
Preconditions	<ul style="list-style-type: none"> - Administrator is logged into system - Physical mail not already logged into system
Postconditions	<ul style="list-style-type: none"> - New mail is logged into database - The targeted recipient is alerted
Main Scenario	<ol style="list-style-type: none"> 1. Mail arrives at ICBC 2. Administrator selects "Enter New Mail" 3. Administrator fills in necessary fields <ol style="list-style-type: none"> a. Mail sender (optional - defaults to blank) b. Mail recipient c. Mail type (optional - defaults to envelope) 4. Administrator selects "Log Mail" 5. App. records new mail into database 6. App. generates new alert notifying mail recipient
Alternate Scenarios	<ul style="list-style-type: none"> - (3b) Recipient field is not filled in <ul style="list-style-type: none"> - App. prompts to fill in necessary field

View, Action and / or Provide Feedback on mail request	
Description	An admin can open up a mail request to view the details of the request, modify the status of the request and add additional comments or feedback for the requesting user to see.
Primary Actor	Admin
Preconditions	<ul style="list-style-type: none"> - Admin is logged into the system - A mail request must exist
Postconditions	<ul style="list-style-type: none"> - View: nothing is changed - Action: request is modified based on actions and notification is sent to requesting user - Feedback: comment is added to the request and notification is sent to requesting user
Main Scenario	<ol style="list-style-type: none"> 1. Admin opens up a Mail Request 2. Mail request is displayed for viewing

	<ul style="list-style-type: none"> a. Status of request can be modified (“Completed”, “In Progress”, etc.) b. Comment can be added to the request 3. Notification for change in request is sent to requesting user (if 2a or 2b is done)
Alternate Scenarios	<ul style="list-style-type: none"> - (2a) If another admin changed the status of the request before the status is modified, an error message will notify the current admin that the status already has been modified

Non-Functional Requirements

- Performance Capabilities
 - Must be able to perform system functions at a reasonably consistent speed:
 - Application must be able to perform tasks without getting in the way of work.
 - Must be able to handle up to 50 concurrent users
 - Must be able to handle up to 1000 live workplace registrations in the system
 - A DBA script to offload lapsed dates into an ‘archive table’ would be needed to maintain high system performance
- UI capabilities
 - Must support UI on Edge
 - Must work on desktop and on mobile
 - Optional: Provide the user with a visual method to reserve a desk, by clicking on spatial regions corresponding to desk locations
- The following is dependent on ICBC’s current system environment:
 - Backup needs:
 - Any system failures will have to be recovered from backups made on ICBC’s servers
 - Security needs
 - The system will have to be fallproof.
 - Costs
 - There is no cost for developing the system as we will be either hosting servers locally or using free server services
 - Costs of database servers and web servers are dependant on what ICBC decides to use when deploying