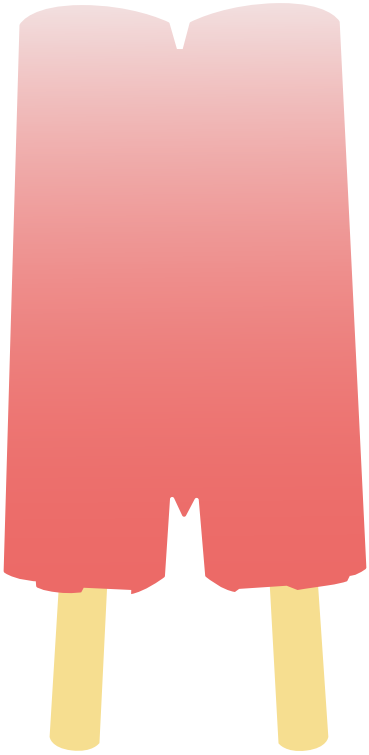


mATE

# Initial Design Report

## CSCI3100 - Group: E6 - mATE


Ao Kin Pong 1155096429
Hui Lok Hin 1155133188
Lam Pui Hong 1155093597
Lao Kam Fung 1155096606
Wong Chun Wai Tom 1155109240

# 1.1 – Project Overview

mATE is a social networking dining application specifically designed for optimal restaurant sharing and communication experience. Combining restaurant searching functions and social networking functions into one, mATE allows users to create and build their very own personal network with reliable restaurants and friends, in which they can discover, participate in, and share their dining experience in various restaurants. By a focus on photos, tags, and ratings, mATE also ensures it is not the companies or restaurants that define the presentation of themselves, but the users, thus creating a truly user-based environment that users can rely on to build their own circle of desirable restaurants. mATE undoubtedly provides one of the best restaurant-searching and sharing experience among all the restaurants-rating app currently in the market.

## 1.2 – Objective

mATE is all focused in user experience. We aim to provide a normal restaurant-searching/rating application with a twist of social networking experience. Hence, there are two main objectives: 1) between users and restaurant, to allow users search, discover, and rate restaurants, and 2) inter-users, to allow users to communicate and share said restaurants – or even users, depends on what the hidden gems are – to other users. For the former, mATE aims to provides a page where hot and popular restaurants and users are recommended to users. Users can search for entities by specifying fields like name, types, tags, and location; upon found, posts can be created with respect to the restaurant and are visible to all parties. For the latter, mATE aims to provides a feed where recent activities, i.e. posts or shares by other users, are displayed to users. Relevant notifications are sent to users, who can then receive information from their trusted partners to slowly build up their own network of restaurants that suits their taste. Aside from the above, restaurant creation and claiming ownership of restaurants are provided in cases where restaurants are not in the database yet.

# 1.3 – Expected Customers and Market

Our target customers would be 15-40 years old Internet generation who love spending time finding delicious food and their favourite restaurants. They are not content with the current searching system of existing restaurant finding and reviewing applications, deeming them full of anonymous reviews and unable to correctly reflect the restaurants pros and cons. Meanwhile, they find the current social network, as a substitute for searching in the dining apps, provides too little information on restaurants and too vague to use. As pleasing as the photos and information they find in the network, when they stumble onto a potential restaurant and wish to know more reviews and photos, they have to search again, transfer to another application, or use online searching engine. mATE aim to target these enthusiasts who wish to get the benefits of both platforms, while share and gain these knowledge, places, and dishes to and from a community they built around (free promotion!).

# 1.4 – System features and Interfaces

mATE has a general interface of a social network application with a twist on what the users can access. There are two types of entities: users and restaurants. Users 1) have their own profile, 2) can like and comment posts, and 3) can follow and share other entities, while restaurants only have their own page that users can follow and save.

mATE has in total five main features:

1) **Activity feed**: All recent activities – namely restaurants reviews, restaurant/user sharing, and restaurant check-ins – of followed entities are displayed. Users can interact with these posts as in normal social networks.

2) **Search and Discover**: It displays popular restaurants and users in the area. If users decide to search, the page will display the searching result.

3) **Profile**: It displays the user's own information and posts which users can edit. When there are new activites or interactions from following entities, it notifies the user (if they desire to be notified).

4) **Post or Review Creation**: Users can choose to write a review for a restaurant or check into a restaurant. If the restaurant they desire to review are not included in the server, they can create it, waiting for owners to claim (in order to further edit restaurant information).

5) **Reservation**: Users can make reservation on restaurants' page.

# 2 – Background

Currently there are only two main restaurant-searching and rating applications in Hong Kong. These applications allow people to search for specific cuisines in certain area to find their desired restaurants. Often, users are required to go through a lot of comments to determine whether the restaurant is up to their expectation, let alone what food or dish is the best the restaurant can provide. Worst still, these comments are made by anonymous people where a portion of which might be paid to write opinions that are in favor for the restaurants.

Meanwhile, an increasing number of people, ranging from teenagers to middle-aged adults, start to switch their attention to social media and use their tagging function. An array of photos is presented to the users, providing them much more insight than comments. Embedded with description and posted by accounts that users can engage and verify, these posts created another platform achieving the same purpose (albeit not being the intended use of the application). However, highly customizable tags simply mean there are too much to choose from; a lot of information such as location and dish type are condensed into a single hashtag – the only one user can search, with no additional information or filters. To acquire more information, comments, and reviews of that restaurant, another queries, again with vague searching methods and troublesome to user, are required.

All in all, the current restaurant-searching applications fail to let users engage in a network, while social networks fail to provide a suitable follow-up on the enquiry of restaurants.

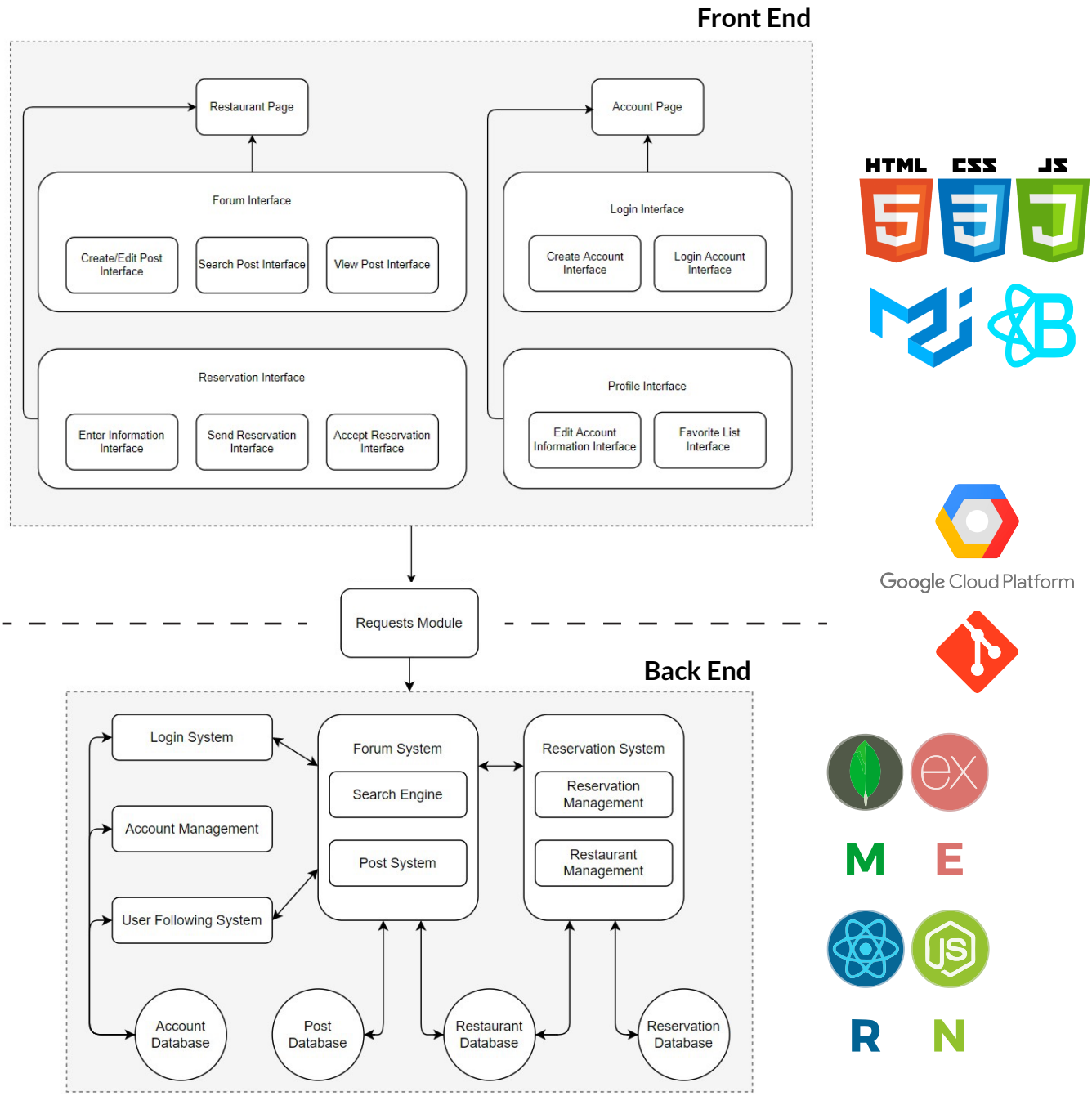
# 2 – Background

Thus, mATE is created.

It serves the function of both applications above and offers more. Search has never been easier: for one, user can still search normally by location, cuisine, and name; on top of that, user can search by multiple tags (including but not limited to description, food types, and price) and filter to find the food they so wanted. Restaurants only consists of basic information and the rest are reviews of the restaurants. Within the reviews centered in photos and tags, users are provided with visual representation of the food to decide their view towards the restaurant without having to read much description – an option nevertheless that are still available to the user.

Implementation of a social network encourages user to find entities of their interest. No matter with friends or families, users can follow their accounts and check if any interesting restaurants they have recently visited; then, one can head to the restaurant info page in one click, check the reviews, and add them to their list if one desires. Gradually, users create their own circle of restaurants and users, discovering new places to dine in while encouraging others to explore more. No more anonymous comments, empty accounts, or paid reviews when all the users' reviews are available for one to check, all while maintaining a trustworthy network to share personal experiences or simply a check-in.

# 3.1 - Archietecture Diagram







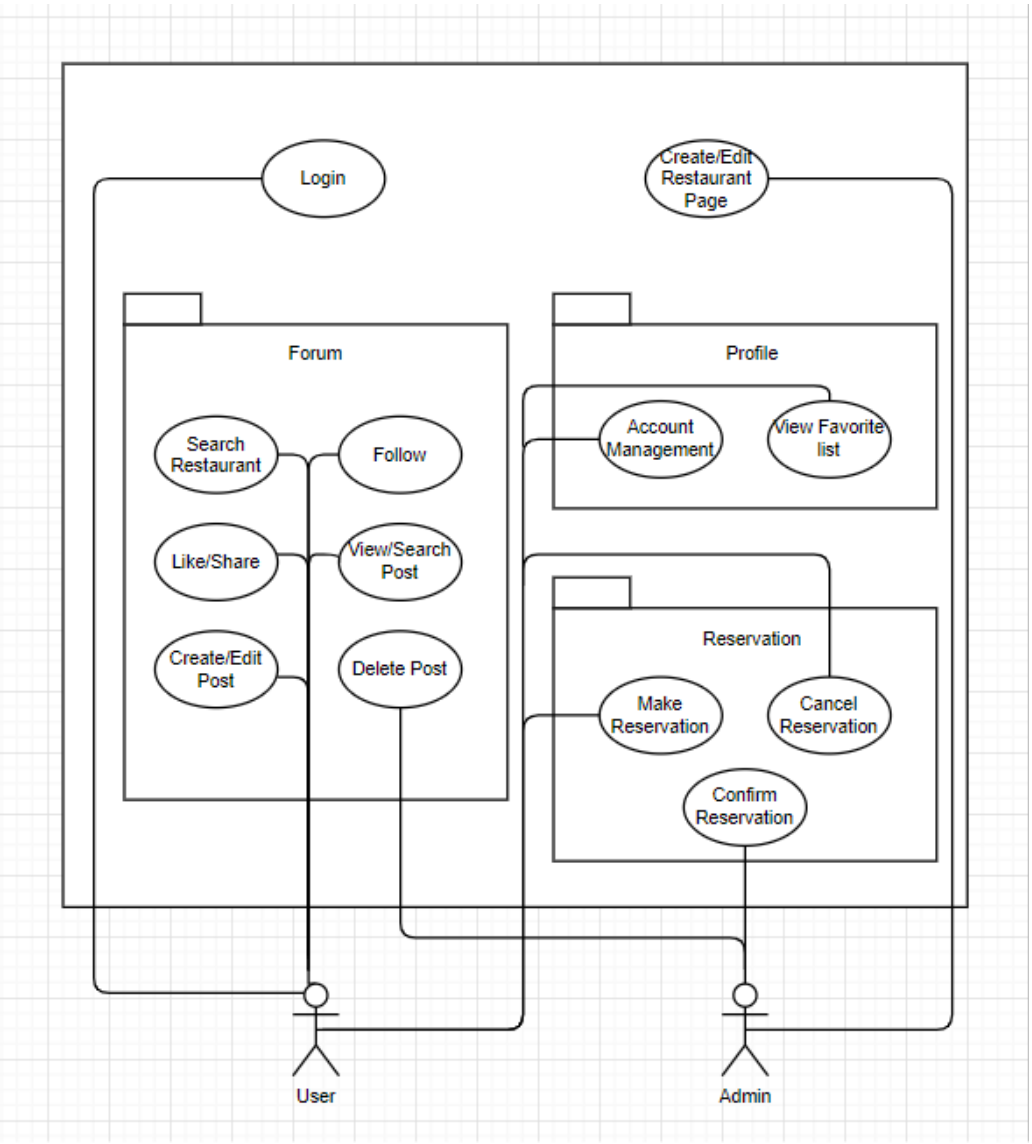




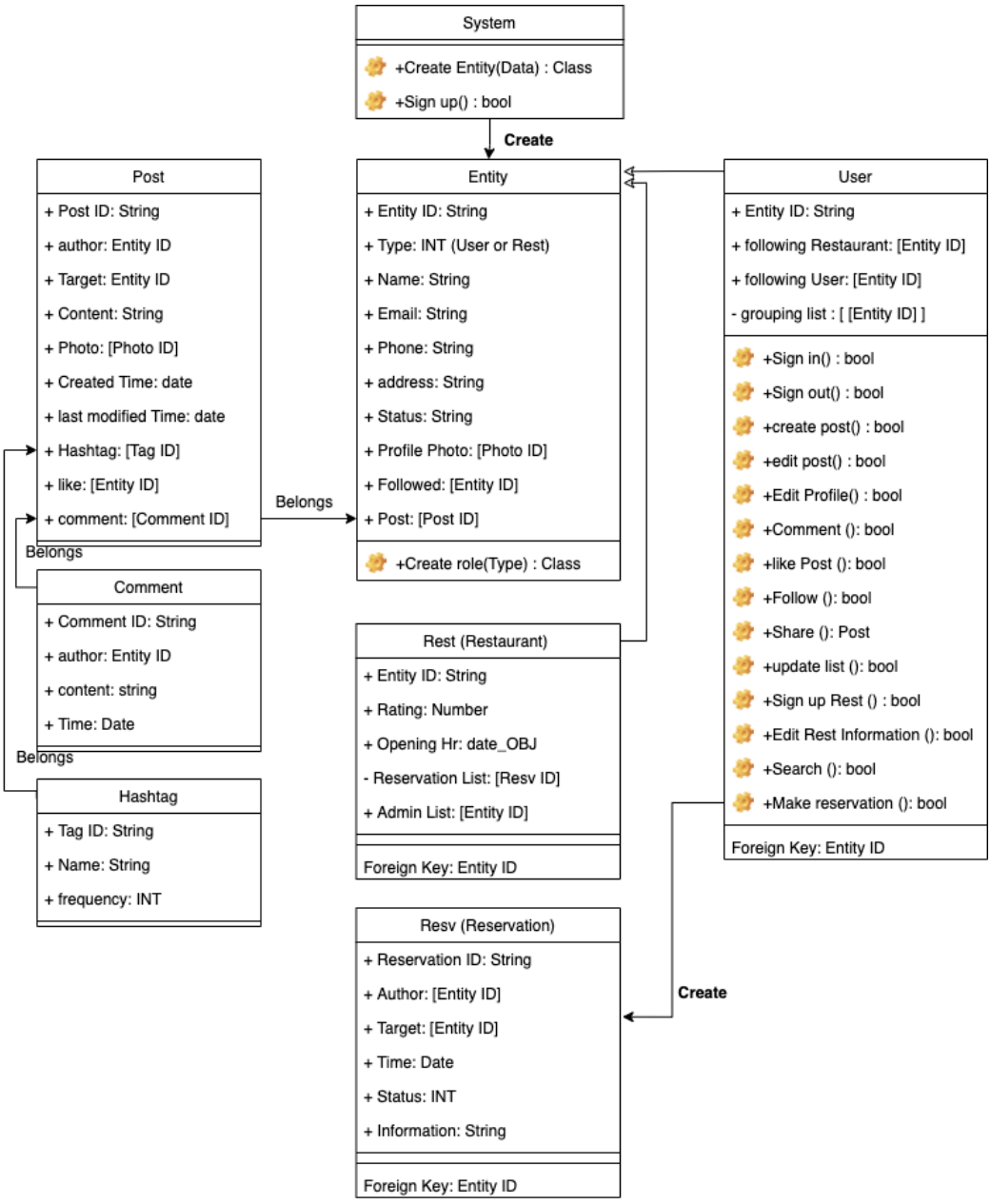


# 3.2 – Use Case Diagram

The following figure shows the system's functionality for users. Admin is a special type of user that can enjoy extra features that are excluded to normal users.



# 3.3.1 - Class Diagram



# 3.3.2 - Class Description

## Entity

| | | |
|---|---|---|
| **entityID** | string | Unique entity ID. |
| **type** | integer | 0: User, 1: Restaurant |
| **name** | string | Name of the entity. |
| **email** | string | Email of the entity. |
| **phone** | string | Phone number of the entity. |
| **address** | string | Address of the entity. |
| **status** | string | Gender (type=0) or opening status (type=1) of the entity. |
| **profilePhoto** | string | A list of photo references of the entity. |
| **followed** | User | A list of users who followed the entity. |
| **post** | Post | A list of posts (type=0) or reviews (type=1) that the entity created. |
| **createRole()** | Create User or Restaurant based on type. | |

## Rest - Restaurant

| | | |
|---|---|---|
| **entityID** | Entity | Unique entity ID (ref: Entity). |
| **rating** | Number (float) | The rating of the restaurant. |
| **openingHr** | [string, string] | A list (length of 7, one per day) of the opening and closing hour of the restaurant. |
| **admin** | User | A list of users that can edit the restaurant. |
| **reservation** | Resv | A list of users that can edit the restaurant. |

# 3.3.2 - Class Description

## User

| | | |
|---|---|---|
| **entityID** | Entity | Unique entity ID (ref: Entity). |
| **followingRest** | Entity (Rest) | A list of restaurants that the user followed. |
| **followingUser** | Entity (User) | A list of users that the user followed. |
| **groupList** | Entity (Rest) | A list of entity lists that the user saved. |
| **signIn()** | Sign into the system. | |
| **signOut()** | Sign out of the system. | |
| **search()** | Search an entity by tags. | |
| **createPost()** | Create a post (i.e., check-in, share, or review). | |
| **editPost()** | Edit a created post. | |
| **like()** | Like a post. | |
| **follow()** | Follow an entity. | |
| **share()** | Share the link of the page. | |
| **updateList()** | Save/update/remove a restaurant into a list (included favorite). | |
| **createRest()** | Create a restaurant entry. | |
| **claimRest()** | Submit a claim for a restaurant's ownership. | |
| **editRest()** | Edit a restaurant information (if applicable). | |
| **search()** | Search an entity by tags. | |
| **makeResv()** | Submit a reservation request for restaurant. | |

## Resv - Reservation

| | | |
|---|---|---|
| **resvID** | string | Unique reservation ID. |
| **author** | Entity (User) | The user that submits the reservation request. |
| **target** | Entity (Rest) | The restaurant that reviews the request. |
| **time** | Date | The time that the reservation request is made. |
| **status** | Number (int) | Not approved (0) or approved (1). |
| **information** | string | All information regarding the reservation. (Time, number of people, what kind of table, etc.) |

# 3.3.2 – Class Description

## Post

| | | |
|---|---|---|
| **postID** | string | Unique post ID. |
| **author** | Entity (User) | The author of the post. |
| **target** | Entity | The target entity of the post. |
| **content** | string | The content (description) of the post. |
| **photo** | string | A list of photo references of the entity. |
| **createdTime** | Date | The time that the post is created. |
| **modifiedTime** | Date | The time that the post is last modified. |
| **hashtag** | Hashtag | A list of tags that the post possesses. |
| **like** | Entity (User) | A list of users that liked the post. |
| **comment** | Comment | A list of comments that the post possesses. |

## Comment

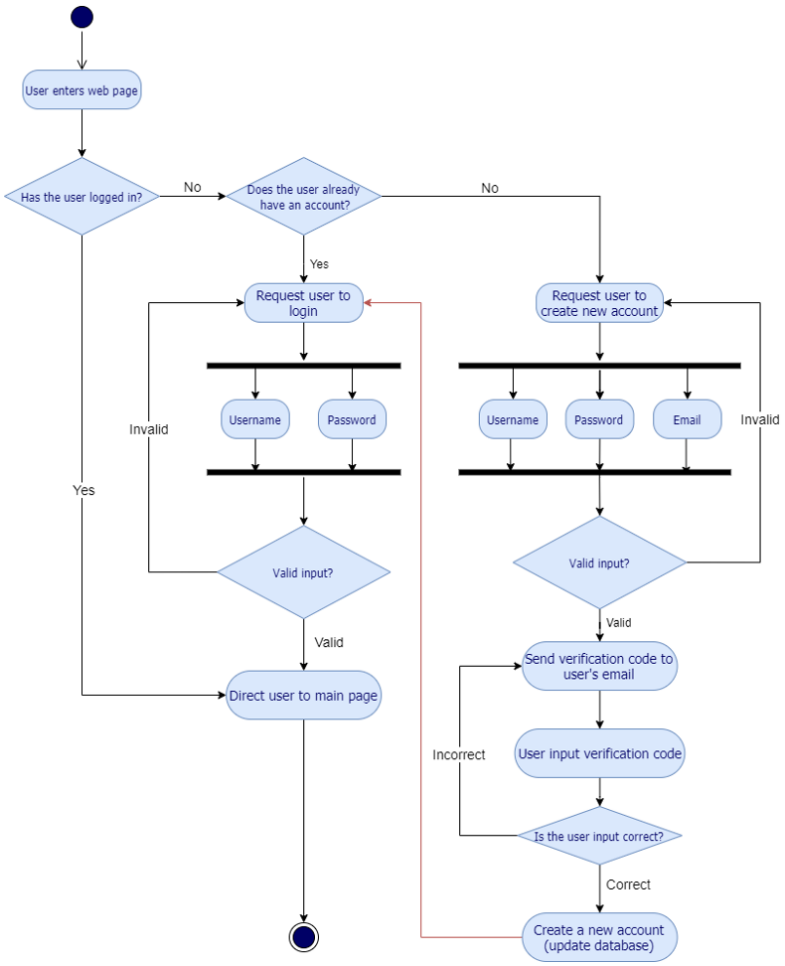
| | | |
|---|---|---|
| **commentID** | string | Unique comment ID. |
| **author** | Entity (User) | The author of the comment. |
| **content** | string | The content of the comment. |
| **time** | Date | The time that the comment is created. |

## Hashtag

| | | |
|---|---|---|
| **tagID** | string | Unique tag ID. |
| **name** | string | Name of the tag. |
| **frequency** | Number (int) | The number of usages of the tag. |

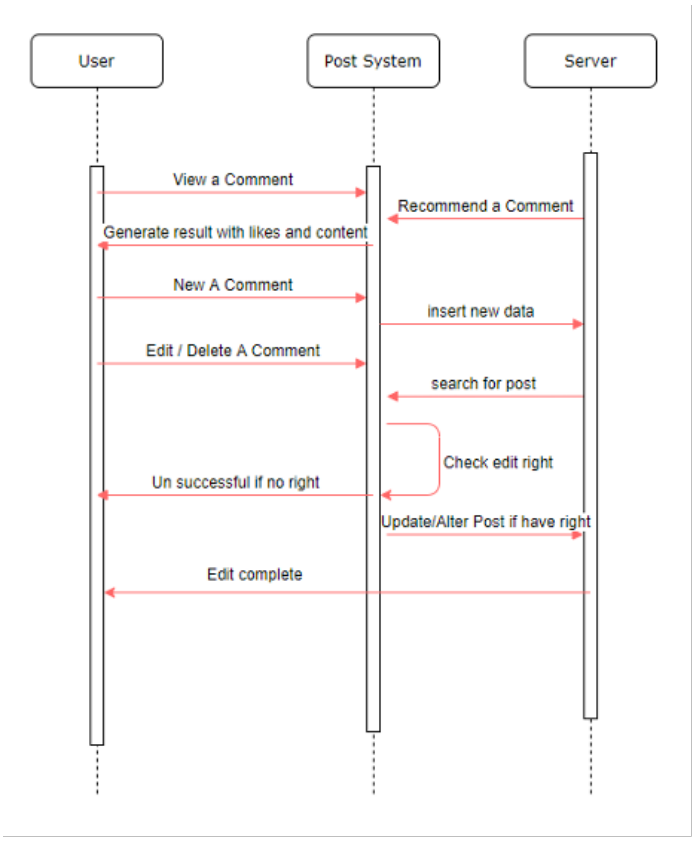
# 3.4.1 - Login and Sign-in System

The following figure showsthe registration and login procedures. All users have to login their own account  to access mATE platform. Users that do not have an account will be directed to create a new one. To create a new account, users are required to provide username, password and email. Verification code will be sent to the user's email for registration. User with an existing account only needs to provide username and password to login. Only logged in users will be directed to the main page.



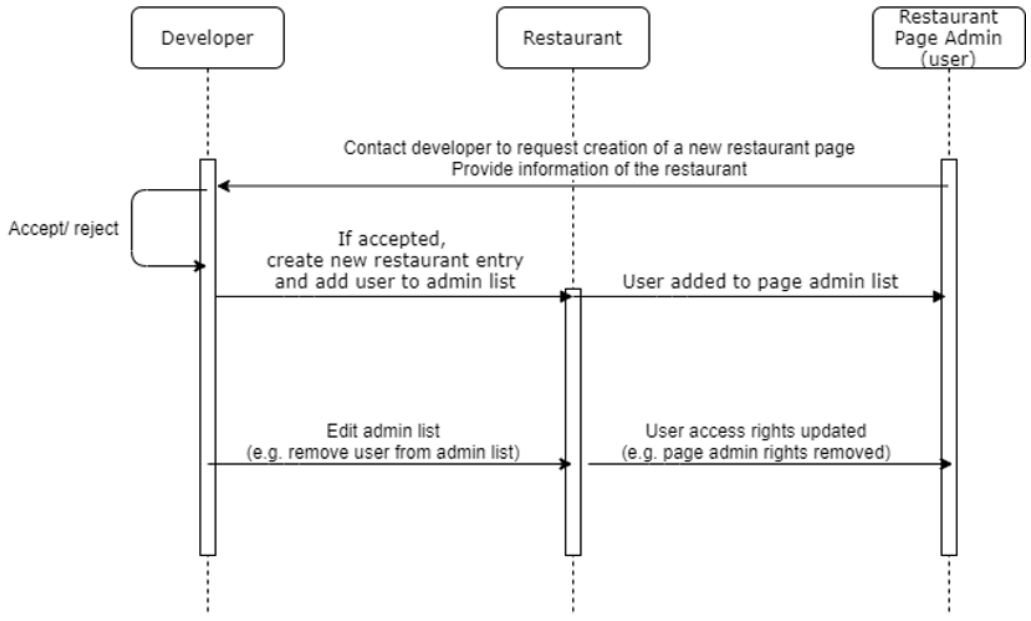
# 3.4.5 - Creating Content

When a user views a post, we will generate recommended comment belongs to this post reference to the comment likes and views number. User may comment regarding the post or a specific comment. Whenever user type in #, hashtag mode would be activated. System will grab data from DB with user's typing. When user wrote a comment and clicked submit button. System will check if the user is the owner of that restaurant. If yes, we will add owner in front of the username of the comment and those hashtags will not be counted as tag increase under that comment.



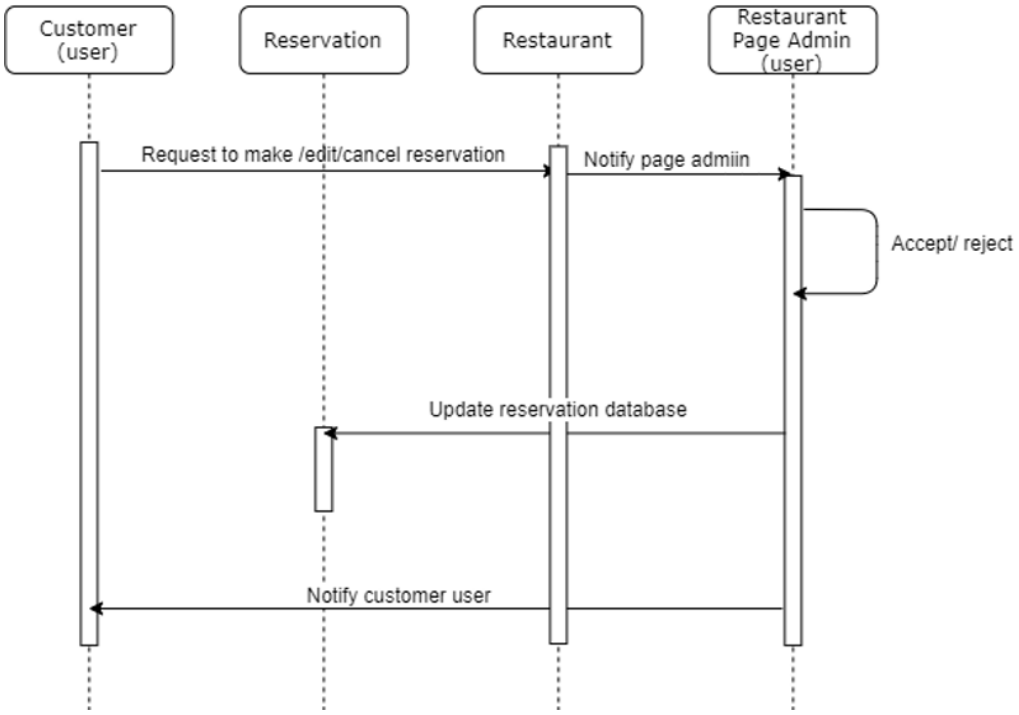
# 3.4.2 – Creating New Restaurant

The following figure shows the procedures of how a restaurant owner, which originally is a normal user at the web page, can create their own restaurant page in mATE. First the owner has to send the request to the platform developer and provide relevant information of the restaurant (location, phone number) to the developer. The request will be processed by the developer. If accepted, a new restaurant page will be created and the restaurant database will be updated accordingly. The account of the restaurant owner will be added to the admin list of that restaurant page, so that they can manage their own restaurant page. *The platform developer reserves the right to edit/remove the restaurant page at any time.



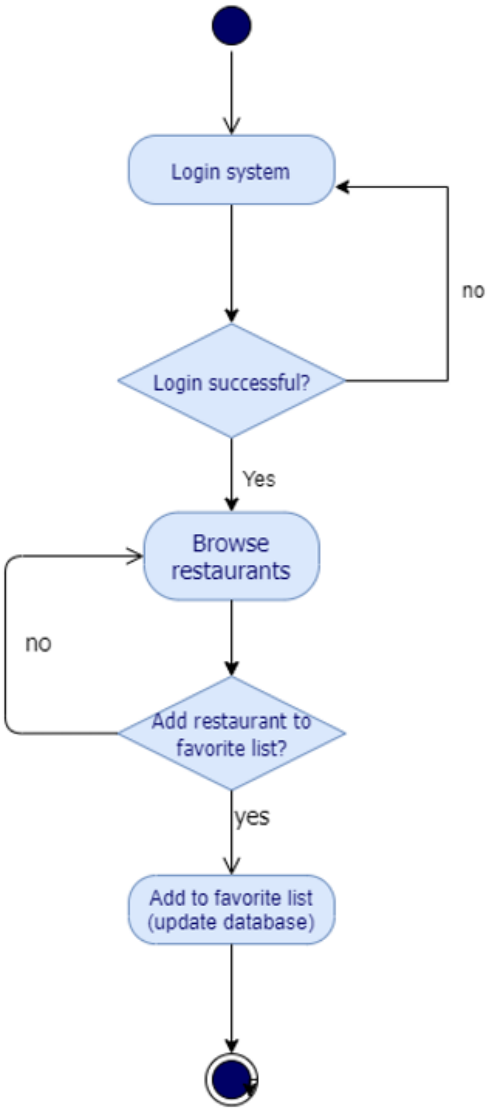
# 3.4.3 – Reservation

The following figure shows the procedures of making an online reservation at a restaurant. First the user opens the page of the restaurant, where they can find the section for making reservation. To make a reservation, users provide relevant information (arrival time, number of people) at the reservation page, which will be sent to the restaurant page admin. The reservation request will be processed by the restaurant page admin (accept/reject). The reservation database will be updated accordingly and the customer user will be notified. After making a successful reservation, customers can request to edit or cancel the reservation.

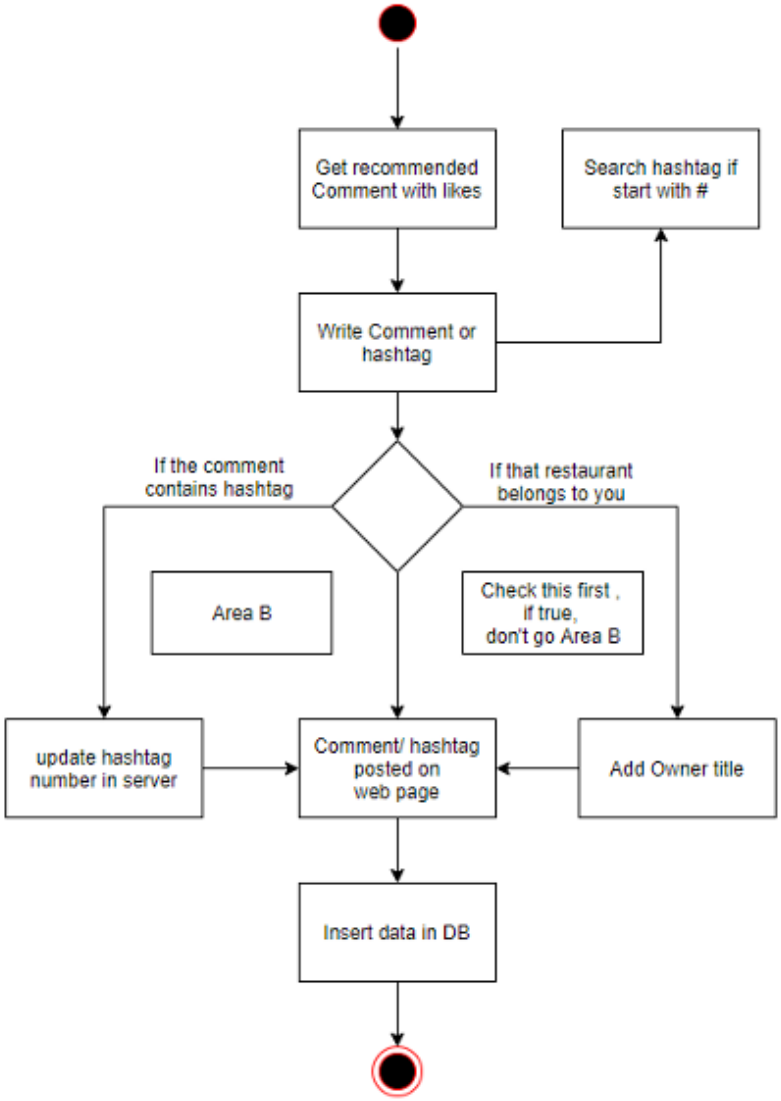


# 3.4.4 – Saving Restaurant

The figure on the right shows how a user can add a restaurant to their favorite list. First, the logged in user can browse the restaurant pages to look for the restaurant they like. If the user would like to add the restaurant to their favorite list, they can click the button “Add to favorite list”. Restaurants that are in the favorite list will be displayed to the user at the section “Favorite Restaurant”.

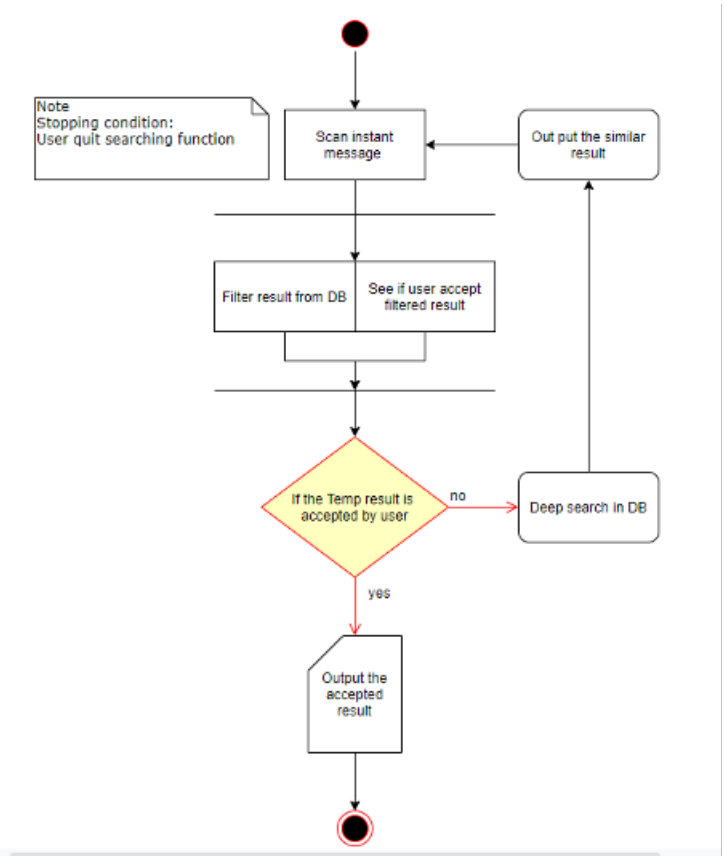



# 3.4.5 - Creating Content



# 3.4.6 – Searching

When user searches, system will scan the current input and grab a list of similar result from the database. Those data would be stored locally and keep matching and recommending searching result to user. If user accepted search result, result will output the required data and function end. In some cases, if users delete a word and that ruins the searching condition from the database. System will grab data from the databasesagain and repeat the step above. The search is ended when search complete or user end the searching function. The matching between user and local will commit once user change searching input.



# 3.4.6 – Searching