

CCAPDEV Term 3, AY 2022 – 2023

Machine Project – Project Specifications Document

Groupings: At most 4 members in a group

SPECIFICATIONS

For the entirety of the MCO, your task is to create a web application based on the given set of web applications described below. The list describes the minimum features required to be implemented for the project, but the group may (and is encouraged) to implement additional features and capabilities for their selected project.

The MCO will be split into three different phases, each with their own expectations on the project's progress. You may refer to the phases' respective documents for the specifics on its deliverables.

The project is at most done in groups of 4. During the development phases, copying other group's work is not allowed and is punishable by a grade of 0.0 for the entire CCAPDEV course. A discipline case will be filed with the Discipline Office in case of academic dishonesty.

LIST OF PROJECT SPECIFICATIONS

Lab Reservation System

The following describes the features of a computer laboratory slot (seat) reservation web application. Groups must include at least three computer labs in the system. The minimum features required to be implemented for this project is as follows:

- *View Slot Availability*
 - A user may choose a computer lab and see the current available seats of the chosen lab. The user may opt to see the availability of the lab at other times for the next 7 days. A user must be a student user to reserve a seat.
 - Note: Users may see who reserved a seat, as long as the reservee did not make the reservation anonymously. Clicking on a user's name should link to that user's profile. Availability must also update periodically so that the user does not need to refresh every time.
- *Register*
 - A visitor must register if they want to reserve a lab slot. Here, a visitor must enter their DLSU email and a password. There should be 2 kinds of accounts, the student who can reserve lab slots, and the lab technician that can block a time slot for walk-in students.

- *Login*
 - After registering properly, a visitor may log-in. Upon logging in, the option to reserve a slot will be open. The user is given the option to be “remembered” by the website. When the user chooses this option, every log in and visit to the website will extend their “remember” period by 3 weeks.
- *Logout*
 - The user may log out from their account. This should cut short the “remember” period if it exists, and clears any session-related data.
- *Reserve*
 - Students can reserve slots that have not yet been taken. Lab slots are in intervals of 30 minutes. The student also has the option whether to reserve anonymously or not. The student can reserve more than one slot, and all the lab slots will be made under one reservation. A student may not reserve a previously reserved slot.
- *Reserve for a student*
 - Lab technicians can make a reservation for walk-in students.
- *Remove reservation*
 - Lab technicians have the ability to remove reservations of students who do not show up within 10 minutes of the reservation. This facility is only available 10 minutes of the actual reservation time. This will cancel the whole reservation.
- *Edit reservation*
 - A student can edit reservations they previously made.
 - Lab technicians are able to edit any reservations.
- *See reservations*
 - A user may check their reservations. They can see details such as the seat number, the laboratory, the date and time of request, and the date and time of reservation.
- *View / Edit User profile*
 - Registered users may edit their own profile, which includes a profile picture and description. Additionally, the profile should also list the student’s current reservations.
 - Users may also view another user’s public profile, but they are not able to edit it.
- *Delete User Account*
 - Students have the option to delete their own user, ultimately deleting their account and cancelling any pending reservations they may have.
- *Search for users / free slots*
 - Users can view all the available slots given a provided date and time, and the lab.
- *General*

- Good user experience. Visitors can easily navigate without help, all information is easy to access. Good visual design. Design suits the theme of the application, and is cohesive and consistent across the whole application
-

Forum Web Application

The following describes the features of a forum web application. Groups may choose to either design the forum web application as a general-interest forum or one that is more catered towards a specific interest group. The minimum features required to be implemented for this project is as follows:

- *View all posts*
 - Upon visiting the web page, an unregistered visitor may see the 15-20 most recently* uploaded post titles and a post description snippet. The user can see the next set of uploaded posts, it is up to the group whether to implement this in the same (auto-load) or another page. Clicking the post title will let the user view the post and the comments (see: view a post).
 - * The user may also see the most popular post based on a calculated ranking.
- *Register*
 - A visitor must register if they want to post or comment. Here, a visitor must enter their username and their password.
- *View a user profile*
 - Each user has their own page which shows their profile publicly. On the same page, a visitor may see the user's username, profile picture, and a short description. They may also see a portion of the user's latest posts and comments. The visitor may opt to see the rest of the posts and comments of the user.
- *Edit Profile*
 - A user that is logged in may edit their user profile, wherein they can add / modify a profile picture, and provide a short description (can be left empty).
- *Login*
 - After registering properly, a visitor may log-in. Upon logging in, the user can start posting and commenting. The user is given the option to be "remembered" by the website. When the user chooses this option, every log in and visit to the website will extend their "remember" period by 3 weeks.
- *Logout*

- The user may log out from their account. This should cut short the “remember” period if it exists, and clears any session-related data.
- *Post*
 - A user may make a text post. They must give a title for the post, and the body of the post. Additional points will be given for allowing markup (e.g., rich text editing) without the risk of cross site-scripting.
- *View a post*
 - A user may view any post they have a link to. This will load the title, the body of the post, and the comments of the post.
- *Comment*
 - A user can comment on any post, including their own. They may also reply to another user’s comments, and the comments can nest indefinitely.
- *Edit / Delete a post*
 - The owner of the post may edit their posts at any point. Edited posts should have an indication that it has been edited.
 - The owner of the post may delete their post as well.
- *Edit / Delete a comment*
 - The owner of the comment may edit their comments at any point. Edited comments should have an indication that it has been edited.
 - The owner of the comment may delete their comment.
- *Upvote / Downvote*
 - A user can upvote / downvote a post or a comment (including their own) once.
- *Search*
 - A visitor/user can search for posts by similarities in the title or the post body. By entering a search phrase/word, all posts containing it will appear as results.
- *General*
 - Good user experience. Visitors can easily navigate without help, all information is easy to access. Good visual design. Design suits the theme of the application, and is cohesive and consistent across the whole application

Restaurant / Store / Service Review Web Application

The following describes the features of a restaurant / store / service review web application. Groups may choose to either develop a review web application for either restaurants, stores, or services. For realism, groups may base the establishments featured in the application from actual establishments that are found around the campus. The minimum features required to be implemented for this project is as follows:

- *View establishments*

- Upon visiting the web page, an unregistered visitor may see a list of establishments that are featured in the web application.
 - The overall rating and a brief description of the establishment must be easily viewed/accessible from the list.
- *View an establishment's reviews*
 - Upon selecting an establishment, an unregistered visitor may see the top reviews. The user can also see the next set of top reviews--it is up to the group whether to implement this in the same (auto-load) or another page.
 - Reviews can be long, so some of the text in the preview could be truncated. Clicking the review will let the user view the review in full detail.
 - Reviews should also include how many registered users found the review helpful / unhelpful.
- *Register*
 - A visitor must register if they want to post a review. Here, a visitor must enter their username, their password, an avatar, and a short description (can be left empty).
- *Login*
 - After registering properly, a visitor may log-in. Upon logging in, the user can start posting reviews. The user is given the option to be "remembered" by the website. When the user chooses this option, every log in and visit to the website will extend their "remember" period by 3 weeks.
- *Logout*
 - The user may log out from their account. This should cut short the "remember" period if it exists, and clears any session-related data.
- *View a user profile*
 - Each user has their own page which shows their profile publicly. On the same page, a visitor may see the user's username, profile picture, and a short description. They may also see a portion of the user's latest reviews and comments. The visitor may opt to see the rest of the posts and comments of the user.
- *Edit Profile*
 - A user that is logged in may edit their user profile, wherein they can add / modify a profile picture, and provide a short description (can be left empty).
- *Create a review*
 - A logged-in user may create a review on a selected establishment. They must give a title and the body of the review.
 - A rating must also be given by the reviewer. It is up to the student what kind of rating system they choose to implement (e.g., 5-star rating, recommended/not recommended) as long as it can be aggregated to some degree.
 - Users must also be allowed to attach media (e.g., image, videos) to their review.

- Additional points will be given for allowing markup (e.g., rich text editing) without the risk of cross site-scripting.
- *Mark as Helpful / Unhelpful*
 - A logged-in user can mark a review as either helpful / unhelpful once.
- *Edit / Delete a review*
 - The owner of the review may edit their review at any point. Edited reviews should have an indication that it has been edited.
 - The owner of the post may delete their review as well.
- *Search Establishments / Reviews*
 - A visitor/user can search for establishments by similarities in the name or the description. By entering a search phrase/word, all establishment containing the keyword/phrase will appear. Visitors may also filter establishments based on its overall rating
 - A visitor/user can search for reviews on an establishment by similarities in the title or the post body. By entering a search phrase/word, all reviews containing it will appear as results.
- *Establishment owner response*
 - Groups must implement an additional user account role – establishment owner. This doesn't need to be integrated with the register system, and credentials can be created manually by the database admin.
 - Establishment owner accounts are tied to a single establishment, and can respond to reviews on their establishment. Their responses would then be published, making it visible to the public.
- *General*
 - Good user experience. Visitors can easily navigate without help, all information is easy to access. Good visual design. Design suits the theme of the application, and is cohesive and consistent across the whole application.

IMPORTANT: Note that for every instance of a create feature that adds data in the database of the web application, there should be corresponding features that will allow the corresponding user to read, update, and delete the data from the database. For example, if you have a feature that allows the user to create a post, the user should also be allowed to read/view the post, update/edit the post, and delete/remove the post.

WORKING WITH GROUPMATES

For this project, you are encouraged to work in groups of at most 4 members. Make sure that each member of the group has approximately the same amount of contribution for

the project. Problems with groupmates must be discussed internally within the group, and if needed, with the lecturer.