Gym

A local gym has asked you to build a piece of software to help them to manage memberships, and register members for classes.

MVP

* The app should allow the gym to create and edit Members
* The app should allow the gym to create and edit Classes
* The app should allow the gym to book members on specific classes
* The app should show a list of all upcoming classes
* The app should show all members that are booked in for a particular class

Inspired By

* Glofox, Pike13

Possible Extensions

* Classes could have a maximum capacity, and users can only be added while there is space remaining.
* The gym could be able to give its members Premium or Standard membership. Standard members can only be signed up for classes during off-peak hours.
* The Gym could mark members and classes as active/deactivated. Deactivated members/classes will not appear when creating bookings.

**STRUCTURE**

* There is a gym
* Customers:
  + Can access many classes
  + id
  + name
  + (membership type)
  + (membership status)
* Gym Classes:
  + Can be accessed by many customers
  + id
  + name (XL STRENGTH, catchy names)
  + type (strength, extreme, Olympic..)
  + date
  + Start time
  + Duration or End time
  + (repeats)
  + (instructor)
* The Customers and Classes DB will be Joined as we have a Many to Many relationship
  + name = bookings
  + customer\_id
  + class\_id
  + (date of class) -> incase we want to limit some users to one class a day, some other parameters for so many sessions per month

**Still To Do:**

1. All CSS
2. Booking DB
   1. (update)
3. Update and complete diagrams
4. Evidence on testing

**Extensions:**

1. If a customer has already been booked into the class, they can no longer attend that same class
2. Customer Class
   1. Membership level
   2. Membership status
   3. Number of sessions had this month
3. Session Class
   1. Maximum capacity
4. Rooms
   1. Can add another DB of rooms
   2. Each session is assigned to a room
   3. Each room and each session has a capacity
5. Change all date/times to actual date/times
6. Customer duplication – Add more unique details to allow check
7. Time overlap check in sessions

**Basic functionality done:**

* Customers:
  + See all in one page
  + See individual
  + Edit, delete, add new
* As above, for sessions and bookings

**Extra functionality:**

* Bookings can check for duplicates (did not do for customers, as we don’t want customers with the same name to suffer)

**Logbook of my process:**

1. Went through each brief, what’s most interesting, what would I do for each one, how could I grow each one, and chose my brief
2. Made an Excel file for all my draft drawings, made my DB drawings, classes. Thought what do I want the user to be able to do
3. Set my rough plan:
   1. Set up a git repo, make all my files/folders, get my standard code in, make my classes and some instances, test
   2. Set up my repositories and controllers, with plain HTML for testing
   3. Test all my basic functionality of initial desires
   4. Work through the pages with CSS and images, make it look good
   5. Start on further improvements
4. Made all my files/folders
5. Got my standard code in (e.g. in app.py)
6. Made my classes
7. Made some class instances/objects in console.py
   1. Then found the first thing I need, is to save them
8. Started in customer repository, made save/delete
9. Then same for session, then booking, which uses customer and session
10. Think about what I want to be able to do first on the website – I want to see my customers, see my sessions, and see my bookings in lists, and individually (not necessarily bookings individually at this point)
11. Then made the select all and select by id functions in customers, session, and select all for bookings
12. Test my DB connections work in the terminal by viewing the tables
13. Stopped, recheck over plans, what else do I want to do? I want to edit, delete, and add new
14. Started with delete – Had a problem, session was tied to a customer through a booking. Simple fix, then had to redo terminal psql etc, and tested. Then tested in website, works
15. Edit function likewise had a problem, it would redirect but it wouldn’t change anything. Problem was related to the ID
16. Then made the edit function for customer. And repeated both for session
17. Then added new function for customer, session, and booking
18. We can now:
    1. View (all and individual), Add, Edit and Remove Customers
    2. View (all and individual), Add, Edit and Remove Sessions
    3. View (all), Add, and Delete bookings
19. Added a feature to check for duplicates to bookings
20. Repeated for customers
21. Repeated for sessions
22. Next step – Editing bookings. This will involve editing Customers and Sessions
23. Fully test every form, and record how this was done
24. Update all diagrams
25. Format the website with CSS, including Images
26. Then onto extensions. What else would be useful, that we’ve noted en route?
27. Changing the dates and times to actual dates/times
28. Added a membership status to Customers, which will allow/disallow them from booking sessions if expired
29. Added a membership\_level to customers, currently just for a view
30. Then a tally of how many sessions a customer has had in total or in that month
31. Adding rooms, with maximum capacities. Or maximum capacities to Sessions