Computer Science Practical Project

Non-exam Assessment
AQA Computer Science Coursework 2020

Taisei Kikuta

Contents

Analysis

Background

Like most schools, The British School in Tokyo (BST) has access to multiple sports facilities and equipment that can be used by the staff and students. However, instead of being able to own and manage all of the facilities directly, many of them are owned by the neighbouring Showa Women's University (SWU). Because of this, the PE department is lacking a good monitoring and management system for the sports facilities and equipment used in the school. At the moment for SWU-owned facilities, the schools have to book their slots through a complex and difficult to understand calendar which is compulsory because The Showa University is the original owner of the campus, and they are in charge of organising everything. Not only is this calendar in Japanese, but it also only provides a limited amount of useful information that the school wants (such as equipment information). And in terms of BST-owned facilities, there are similarly not any good systems in place other than a calendar with limited information. As the school expands and develops in terms of available space (such as a new BST-owned sports ground about half an hour away), it is becoming significantly strenuous to manage the facilities. This cluttered system currently causes many issues for the PE department in trying to understand it and to implement it during the school week.

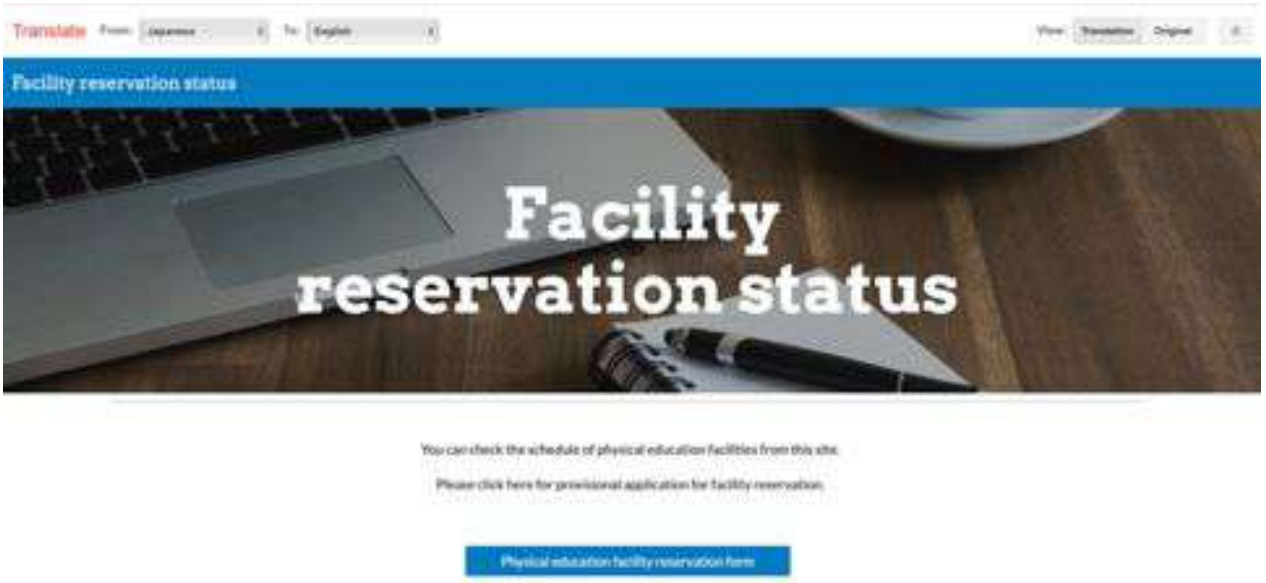
Their biggest concern with the current system is the inability to share information about facility and equipment usage, as well as availability of teachers and students (year groups) across the school community for both BST and non-BST owned facilities. This is particularly important for teachers that may be at risk of being double-booked with another lesson, or to identify specific year groups that may have a special session/lecture during a PE lesson. In addition, on special days such as sports days and inter-house competitions, there is currently no way to keep track of the equipment being used, or which teachers are free. Also, the PE department wants to be able to easily store and check what equipment they actually have, especially when there are several types. Currently, there is no way to do this, causing many of the teachers to forget what equipment they have purchased/added or removed from their storage spaces. Finally, they want a system that can inform them about the number of teachers and students in a particular house and year group to help them manage events by having at least a rough idea of numbers. All of these functions are missing from SWU's current system

The following are a few images of the current system:

体育施設予定表


体育施設１
グラウンド
今日
2020年 1月 5日 ～ 11日
週
月
予定リスト
1/5 (日)
1/6 (月)
1/7 (火)
1/8 (水)
1/9 (木)
1/10 (金)
1/11 (土)
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
21:00
22:00
23:00
Google Calendar


体育施設２
旧体育館（全面）
旧体育館（２階）
今日
2020年 1月 5日 ～ 11日
週
月
予定リスト
1/5 (日)
1/6 (月)
1/7 (火)
1/8 (水)
1/9 (木)
1/10 (金)
1/11 (土)
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
21:00
22:00
23:00
12:00 ～ 15:30
中高部クラブ
旧体育館（２階）
18:00 ～ 20:00
大学クラブ
旧体育館（全面）
12:05 ～ 13:45
BST授業
旧体育館（全面）
13:45 ～ 15:15
初等部授業
旧体育館（全面）
15:15 ～ 17:30
中高部クラブ
旧体育館（全面）
17:30 ～ 20:00
大学クラブ（AUBE）
旧体育館（２階）
12:05 - BST授業
12:45 ～ 14:30
中高部授業
旧体育館（全面）
14:30 ～ 15:15
BST授業
15:15 ～ 17:30
中高部クラブ
旧体育館（全面）
13:00 ～ 17:30
中高部クラブ
旧体育館（全面）
Google Calendar



It can be argued that one of the possible methods of fixing the language issue is just by using an online translator to convert the webpage into english. However, the results are not acceptable for use as it tends not to be accurate, and the calendars do not even show in the translated version.

To overcome the issues described above, the PE department has requested an additional, easy to use and understand online booking system which is managed and controlled by a back-end database. As stated in appendix B, the PE department wants a system where they can view all the BST bookings on one calendar/booking sheet, rather than having to scroll through different calendars like the current system. They also want the view page to display all the necessary information needed such as facilities, equipment, staff members in charge, specific year groups, and others. This way they can actually have the data that they want and share with other staff members. In terms of making the bookings, the PE department wants to be able to easily select and input the necessary information into a form which will then automatically add onto the bookings list/calendar. Additionally, they want the booking system to be adaptive, where it blocks access if there is a risk of double-booking of staff, facilities, and equipment. Also, they want to be able to delete bookings that have already been made just by simply selecting a group of checkboxes and pressing delete. For the booking process, they want to be able to select any (optional) specific house or year group that may be taking part in their event. Finally, with new equipment being added every year along with staff members changing, they want to be able to change and edit the database directly.

Although it would be most desirable if this new system could eliminate all other booking processes, this is simply not possible as the SWU booking calendar is required to be filled in, at least for SWU-owned facilities. Therefore, this is an acceptable limitation as the PE department have said that the main idea of this new booking management system is for it to go alongside the current calendar. This new system will be responsible for adding all the bookings for both BST and non-BST facilities, and providing a cleaner interface to allow all other staff members to view, and not just the PE teachers (which is the case for the SWU calendar).

Research

In terms of the back-end side of the program which is the main aspect, it is important to discuss the role of DBMS (Database Management System) software in the real world, and how it could be applied to this issue:


Above are one of the most commonly used DBMS. MySQL, PostgreSQL, SQLite, and Microsoft Access. Although Microsoft Access is a high level management system with many features, it comes with the limitation of being quite expensive to purchase, and it is also more compatible with Windows devices where my computer runs on MacOS. Therefore I will not be considering Microsoft Access for use. The main advantages and disadvantages of each DBMS is summarised below:

DBMS	Advantages	Disadvantages	Appropriate Use Cases
MySQL	• Has most of the data types • One of the most popular management systems so there are several third-party applications such as phpMyAdmin which support it • Compared to others, MySQL provides the fastest performance • Good security is provided with the ability to set up the password security for the root user, and it also supports user management which allows the owner to choose who has access • Easy to use due to its popularity and support for applications which simplify the processes	• MySQL provides both a free community version (which I will use) and a paid enterprise edition which provides users with all the features, whereas the free version lacks a few of them • Not the strongest for combatting concurrency and large data volumes, when multiple users try to edit the database simultaneously	• Good for web development by easily connecting the database to the HTML/PHP code • Good for if there is expected to be growth and expansion overtime
PostrgreSQL	• SQL compatibility, where it follows the	• Weak memory usage and	• Good when connecting the

	SQL standard the most, providing the most SQL commands and features compared to others ● Completely open-source and community-led software, meaning all users will be guaranteed all the features with all the support necessary	performance, where only a small amount of space is provided for every client which can be used up easily ● Not the most popular compared to the other systems like MySQL ● Due to its weak memory, speed can be reduced	database to other software or operating systems with its wide compatibility ● Good for complex situations because of its several features
SQLite	● Small footprint and is self-contained ● This small size allows for a user-friendly experience and ease of use ● Portable due to its size and simplicity	● However, with its small footprint, there are many lacking features and supported data types ● Weak security and no user management to allow and block user access ● Weak against concurrency	● Good for small scale projects that won't require much space or complexity, and for projects that won't be having much future development ● Good for embedded applications with its portability

To summarise this comparison, MySQL is the most common and popular out of the three, with great overall features, supported data types, speed, security, and its ease of use. This allows MySQL to be supported by many other third-party software that can further make the management and development process easier. As many websites use MySQL as their DBMS, it can also be suggested that MySQL is best suited for web development, and this is reflected from the advantages above.
In comparison, PostgreSQL provides the most features and due to the fact that it is fully open-source, it is guaranteed access to all those features. However with its weak memory and speed performance compared to MySQL, it can limit how far this can be used.

The main advantage to all of the listed software is that they provide a visual database designing tool software alongside their DBMS to help with the organisation. The main ones are MySQL Workbench, pgAdmin, DB Browser (for SQLite), and Microsoft Access has their own built-in visualiser to see the different relations and entities. Despite the fact that additional software needs to be installed, it definitely helps in the database development, design, maintenance, creation and administration processes, reducing the risk of errors when doing so just through SQL scripts, as it provides the visual representations of the tables for aid.
After testing out the different softwares like PostgreSQL, SQLite, and Microsoft Access, I found that MySQL along with MySQL Workbench was the easiest to use and understand, especially with its simplistic user interface.

PostgreSQL was also similarly good, but I had several issues trying to connect it to my machine and getting it to work smoothly, like the 'Port' errors. The same goes for SQLite, where the visual tool did not seem to function properly. Finally, Microsoft Access was also one of the best, but not only does it cost money to work with, but my machine is also a Mac, where it would be less effective to be running Microsoft software. Therefore, I could only test the software on the school computers, but not at home.
Due to the fact that installing all the necessary tools was quite time consuming, I found a separate method of installing everything at once, through something called MAMP (Mac, Apache, MySQL, PHP) which is a free, open source bundled software which allows the development of web-based database systems both on Mac AND Windows, which is perfect as the computers at school use Windows.
In terms of the language used to construct and manipulate the database, SQL (Structured Query Language) will be used as a standard. It comes with all the necessary terms and functions to run both basic and complex tasks/queries such as CREATE tables and UPDATE.

In terms of the front-end side of the solution, due to the fact that it will be web-based, PHP will be required which is also included in the MAMP bundle. PHP is a hypertext preprocessor and server-scripting language used for web development and building webpages. It is known as an old programming language, but still used frequently. The main advantages with PHP is that it allows the connection with MySQL databases through a few lines of extra code, and also the fact that it can include HTML (Hypertext Markup Language), CSS (Cascading Style Sheet), and some JavaScript. Therefore, the code can be written in different ways as long as it is mentioned, and will be flexible.
In this case, the PHP code would be responsible for making changes and manipulating the database via SQL, HTML for designing the website and how the retrieved database information will be displayed, and CSS for further details that cannot be made with HTML.

Initially I thought that this PHP/HTML code would be written on an IDE, but this was not the case. All the code could be written on a simple text document, and by changing the file extension to '.php' or even '.html', the computer would be able to recognise this as a web page.

Although this is not something that has been requested by the PE department, an additional layer of functionality that could be implemented to my system is being able to have access and connect to a staff Google Calendar and display the information that was added. In order to do this, the most appropriate method is by using an API (Application Programming Interface). This is often used in many online services which will request for account information, and this redirects the user to other sites, with access to the data retrieved from the user, like a global variable and inputting it into another function. In this case the Google Calendar API would be required, having the option to redirect the user to the BST PE/Sports staff calendar to check their booking.

Most systems in the professional world also use this type of format of sorting out, just on a much bigger scale, by using a back-end database management system software which automatically organises the overall system through some high level language plus SQL to send commands and queries to the database. And in the front-end using HTML and CSS to build up the visual website which displays the information on the database and also to allow user inputs to edit the database directly through connections between the front and back-end.

Client Discussions

For this project, my main client will be Mr Taaffe, a member of the PE staff, along with the rest of the PE department who will be using the product. I meet Mr Taaffe on a regular basis with the multiple A-Level PE lessons I have with him, so most of my conversations with him happen after lesson/school.

As seen in appendix B, the first few discussions that we had were just based on the current system that the school uses. We shared thoughts on both the advantages and disadvantages of the system, and the process which PE department have to go through each time they want to book a facility. Although the system can be easily accessible from any device and can be convenient, the drawbacks definitely outweighed the advantages. The main emphasis was on the language limitations, and not being able to understand the Japanese. In addition to that, the fact that the system was prioritising their own University student use, the inability to book equipment and other information such as supervising staff, and participating house/year group was a major problem for them. The first email (Appendix A) is the link to the current calendar system as they allowed me to take a look at it.

In the next few discussions we considered the possible solutions that I could produce as a replacement for the school. We listed down the information that would be most useful for the staff, plus some examples for each. I made sure to record this advice onto a mind map as seen in appendix C. We also talked about all the benefits it would provide for not only the PE staff but also the whole school. Not only would it be much easier to understand and use, but it would also provide more flexibility with the ability to add extra information that the current system doesn't allow for. This initial overview can be seen in appendix B.

After agreeing to this plan and having a basic idea of what the result would look like, I started my preparation for the project, firstly by listing all the data items that would be stored in the tables. The facilities, houses, year groups, and storage locations were fairly easy to list down individually because there are only a few of them on the campus, but the equipment was the most difficult to count, list, and follow through. Therefore, I asked the PE department for permission to take a look at the storage/sheds during free lessons to at least get a rough estimated number for each equipment, and its specific name. This can be seen in appendix D.

Another difficult aspect of the process was collecting all the information about the staff and the houses that they belong to, plus the number of students in each year group. As a result, I had to contact Mr Bickley, head of registrations in the school, to provide me with all the useful information about the

number of students and teachers there are in the secondary school, and this can be seen in appendix E.

Potential Solutions

I think the most suitable solution for this situation is to develop a similar system to those used in the professional world, where the whole system is tied together with a back-end database system which will deal with all the data handling and manipulating and maintaining of the database. As discussed above, the best DBMS for this situation is MySQL along with the visual tool MySQL Workbench which is also included in the bundle MAMP. Alongside this, in the middle section, the database back-end will be backed up by PHP and SQL commands and code to send in queries to be run, and to tell the database what it must do based on the information that was collected by the front-end. And finally in the front-end, HTML and CSS will be used to structure the webpages which will be used to show the database and any other useful information that may be wanted by the client.

The first thing that can be done is designing the back-end database and the relationships between tables/entities. This can be done by listing down all the categories of information that will need to be collected and stored, plus all its sub-categories which will be its attributes. For example in this situation, entities for the facility, equipment, and storage locations (and more) will be required. Then, to make the database as normalised as possible, each entity will need a unique identifier, plus any foreign keys to create the connections/relationships between other entities.

Once the design is complete, the tables and their attributes can then be implemented into the actual MySQL database through the Workbench tool by writing SQL queries. Then, the available collected data can then be imported directly into the tables as a csv file.

Once the back-end is complete, the necessary PHP code can be written to manipulate the database, and finally the HTML code to build-up the front-end. This explanation can be visualised using appendix H.

Criteria for Success

General/All Pages (Excluding homepage)

- Navigation bar at top of page
  - Back button to return to the previous page
  - Home button to return back to the home page
- Header at top of page to inform the user which section/page of the website they are on

Homepage

- Clear title at top of page and prompt to tell user to select an option
- Large buttons to redirect user to the different sections of the website
  - Button for 'View Booking' option
  - Button for 'Add Booking' option
  - Button for 'Delete Booking' option
  - Button for ' View/Edit Database' option

View Booking Section

View Booking Page

- Full-page table displaying all booking information
  - Table headers/columns
    - 'Booking ID'
    - 'Booking Made By'
    - 'Booking Made Date'
    - 'Event Name'
    - 'Start Time'
    - 'End Time'
    - 'Booked facility'
    - 'Booked equipment'
    - 'Booked user'
    - 'Booked house'
    - 'Booked year group'
  - Table rows should match the number of bookings in the database and display all bookings made
  - If there are no facility/equipment/user/house/year group booked for a single booking, display 'None'
  - If there are several facility/equipment/user/house/year group booked for a single booking display all of them one line at a time
  - If there are no bookings in the database, an empty table will be displayed informing the user that there are no bookings
- Filter form to allow user to choose which order they want to view the bookings
  - 'Oldest first' radio button (default so filled in first)
  - ' Newest first' radio button
  - Submit button which will re-order the bookings in the specified way
- 'Calendar View' button to redirect user to the Calendar section

Calendar View Section

Calendar View page

- Navigation bar
  - Displays the current month and year
  - 'Previous month' button to move back one month

        - ➢ ‘Next month’ button to move forwards one month
    - ❖ Calendar
        - ➢ Has all 7 columns/headers for each day of the week
        - ➢ The first and last days of the month start and finish on the correct date in the calendar
        - ➢ The date must match its day
        - ➢ If there is still space after the final day of the month, the remaining cells must be left blank
        - ➢ All date cells must have their date/number at the top of the cell
        - ➢ If there is a booking(s) on a given date, the cell must indicate this by displaying the number of events there are, below the date
        - ➢ If there is a booking(s) on a given date, there must be a ‘View’ button to redirect the user to a separate page where those specific bookings are displayed
        - ➢ If there are no booking(s) on a given date, it must display ‘No Events’ below the date

Specific Booking Page

- ❖ Redirected page after the user clicks ‘View’ on a particular date
- ❖ All bookings displayed for the particular date in table format (same as in the ‘View Booking’ section)

Add Booking Section

Add Booking Page/Form

- ❖ Form that allows user to input all the information for their new booking
    - ➢ Required parts indicated with red warning text
        - ■ Text box for the name (‘bookedBy’ field)
        - ■ Date/Time input for start time (‘startTime’ field)
        - ■ Date/Time input for end time (‘endTime’ field)
        - ■ If user doesn’t fill the 3 required fields, they should be blocked from proceeding to the next page
    - ➢ Optional parts
        - ■ Text box for the event name (‘eventName’ field)
        - ■ Checkbox for facility - Allow multiple selections (booking_facility)
        - ■ Multiple select options for equipment (booking_equipment)
        - ■ Multiple select options for user (booking_user)
        - ■ Checkbox for house - Allow multiple selections (booking_house)
        - ■ Checkbox for year group - Allow multiple selections (booking_yearGroup)
        - ■ ‘Reset’ button to allow user to restart the booking process
        - ■ ‘Next’ button to proceed to the next confirmation page
    - ➢ If the required fields are not filled in, they will be blocked

Booking Confirmation Page

- ❖ Each field of the form will have its own division on the page
    - ➢ Name section
        - ■ Program checks if the given name is valid (e.g. No numbers or punctuation, just letters)
        - ■ If the name is invalid, warning message is displayed to tell user to change the name
        - ■ If the name is valid, web page informs the user that it will be recorded as given
    - ➢ Start time section
        - ■ Program checks if the start time and end time given overlaps with any other existing bookings in the database
        - ■ If there are any overlaps, warning message is displayed that there is a risk of double booking of facilities/equipment/users

        - If there are any overlaps, the overlapping bookings will be displayed in table form (like in the ‘View Booking’ section)
        - If there is no overlap, web page informs the user that it will be recorded as given
    - End time section
        - Program checks if the start time is before the end time and the end time is after the start time (correct order)
        - If the order is invalid, warning message is displayed to tell user to change the times
        - If the order is valid, web page informs the user that it will be recorded as given
    - Facility section
        - Displays a list of all the facilities that the user chose in the form
        - If there is a booking overlap (covered in the start time section), web page will display a table of any overlapping facilities
        - If there are overlapping facilities, there will be a warning message telling the user that the booking can’t be made, and they must choose a different facility
    - Equipment section
        - Displays a list of all the equipment that the user chose in the form
        - If there is a booking overlap (covered in the start time section), web page will display a table of any overlapping equipment
        - If there are overlapping equipment, there will be a warning message telling the user that the booking can’t be made, and they must choose a different equipment
    - User section
        - Displays a list of all the user that the user chose in the form
        - If there is a booking overlap (covered in the start time section), web page will display a table of any overlapping users
        - If there are double booked users, there will be a warning message telling the user that the booking can’t be made, and they must choose a different user
    - House section
        - Displays a list of all the houses that the user chose in the form
    - Year group section
        - Displays a list of all the year groups that the user chose in the form
- ‘Book’ button at the bottom of the page
    - If there are any overlapping facilities/equipment/users, the button will be blocked so the user cannot press it
    - If there are no overlapping facilities/equipment/users and the user clicks on the ‘Book’ button, a warning message is displayed asking the user if they are sure to make the booking

Booking Added Page

- If the booking has been added to the database, display a message informing the user that it was successfully added
- Display the new booking that was just added to the database in table form
- Display the updated list of bookings

Delete Booking Section

Delete Booking Page

- Display all the bookings in table form (same as in the ‘View Booking’ section)
- Filter form to allow user to view the bookings in a specified order (same as in the ‘View Booking’ section)

- Additional column in the table to act as a form
    - Header for column must have a ‘Delete’ button to allow the user to press after selecting the bookings they want to remove, plus a ‘Reset’ button to allow the user to restart
    - Checkbox in each of the column to allow the user to check/fill in the bookings they want to delete
- ‘Clear’ button to allow the user to click and delete all the bookings
- When pressing ‘Delete’ or ‘Clear’, a warning message is displayed to ask the user if they are sure

Bookings Deleted Page

- If the user chose to delete specific bookings, display these bookings and inform that they were deleted
- If the user chose to delete all the bookings (clear), display all the bookings and inform that they were deleted
- Display the updated booking table
    - If all the bookings were deleted, display an empty table informing the user that there are no bookings

View/Edit Database Section

View/Edit Database page

- Large buttons that redirect the user to the specific pages/sections
    - ‘View/Edit owner’ button
    - ‘View/Edit facility’ button
    - ‘View/Edit storage’ button
    - ‘View/Edit equipment’ button
    - ‘View/Edit user type’ button
    - ‘View/Edit user’ button
    - ‘View/Edit house’ button
    - ‘View/Edit year group’ button

View/Edit Owner Section

View/Edit Owner page

- All owners displayed in a table format
    - Column for ‘Owner ID’
    - Column for ‘Owner Name’
    - Column for deleting owners (like ‘Delete Booking’ section)
        - Each row has a checkbox to allow the user to check/fill in the box
        - The column header has a ‘Delete’ button to allow user to delete select number of owners, and a reset button to allow user to restart
    - Clear button to allow user to delete all owners in the database
    - If the owner table is empty and there are no owners in the database, display an empty table informing the user that there are no owners
- ‘Add New Owner’ button to redirect user to Add Owner page
- ‘Make changes to Owner’ button to redirect user to edit owner page

Owner Deleted page

- If the owner was deleted from the database, display to inform the user that it was successfully deleted
    - If selected owners were deleted, display these owners
    - If all owners were deleted, display all the owners
- If the owner couldn’t be removed from the database, inform the user that it couldn’t be deleted
- Display the updated owner table

- If all owners were deleted, display an empty table informing the user that there are no owners

Add Owner page

- Form to allow the user to input the required information for a new owner
  - Text box to enter the owner's name (required)
    - If the text box is left blank and the user tries to add the owner, they will be blocked
  - 'Add' button to add the owner and proceed to the next page, and a rest button to restart if needed
  - When pressing the add button, a warning message should be displayed to ask the user if they are sure they want to add the owner

Owner Added page

- If the owner was added to the owner table, display a message informing the user that it was successfully added
- Display the updated owner table

Change Owner Details page

- Form to allow user to select an owner and change their details
  - Drop down menu with the owners as options (already selected with a default value)
  - Text box to input the owners new name (required)
    - If the user doesn't fill in the text box and tries to proceed, they will be blocked
  - 'Change' button to proceed and a 'Reset' button to restart if needed

Owner Details Changed page

- If the owner's details have been changed, display a message to inform the user that it has been successfully changed
- Display a table of the updated owners

View/Edit Facility Section

View/Edit Facility page

- All facilities displayed in a table format
  - Column for 'Facility ID'
  - Column for 'Facility Name'
  - Column for 'Owner Name'
  - Column for deleting facilities (like 'Delete Booking' section)
    - Each row has a checkbox to allow the user to check/fill in the box
    - The column header has a 'Delete' button to allow user to delete select number of facilities, and a reset button to allow user to restart if needed
  - Clear button to allow user to delete all facilities in the facility table
  - If the facility table is empty and there are no facilities in the database, display an empty table informing the user that there are no facilities
- 'Add New Facility' button to redirect user to Add Facility page
- 'Make changes to Facility' button to redirect user to edit facility page

Facility Deleted page

- If the facility was deleted from the database, display to inform the user that it was successfully deleted
  - If selected facilities were deleted, display these facilities
  - If all facilities were deleted, display all the facilities
- If the facilities couldn't be removed from the database, inform the user that it couldn't be deleted
- Display the updated facility table
  - If all facilities were deleted, display an empty table informing the user that there are no facilities

Add Facility page

- Form to allow the user to input the required information for a new facility
    - Text box to enter the facility's name (required)
        - If the text box is left blank and the user tries to add the facility, they will be blocked
    - Drop down menu to select an owner for the facility (already selected with a default value)
    - 'Add' button to add the facility and proceed to the next page, and a rest button to restart
    - When pressing the add button, a warning message should be displayed to ask the user if they are sure they want to add the facility

Facility Added page

- If the facility was added to the database, display a message informing the user that it was successfully added
- Display the updated facility table

Change Facility Details page

- Form to allow user to select an facility and change their details
    - Drop down menu with the facilities as options (already selected with a default value)
    - Drop down menu with the new owners as options (optional)
    - Text box to input the owners new name (optional)
    - 'Change' button to proceed and a 'Reset' button to restart if needed

Facility Details Changed page

- If the facility's details have been changed, display a message to inform the user that it has been successfully changed
    - Display each of the changes made (facility name and/or facility's owner)
- If the facility's details couldn't be changed, display a message to inform the user that the changes couldn't be made
- Display a table of the updated facilities

View/Edit Storage Section

View/Edit Storage page

- All storages displayed in a table format
    - Column for 'Storage ID'
    - Column for 'Storage Name'
    - Column for 'Facility Name'
    - Column for deleting storages (like 'Delete Booking' section)
        - Each row has a checkbox to allow the user to check/fill in the box
        - The column header has a 'Delete' button to allow user to delete select number of storages, and a reset button to allow user to restart
    - Clear button to allow user to delete all storages in the storage table
    - If the storage table is empty and there are no storages in the database, display an empty table informing the user that there are no storages
- 'Add New Storage' button to redirect user to Add storage page
- 'Make changes to storage' button to redirect user to edit storage page

Storage Deleted page

- If the storage was deleted from the database, display to inform the user that it was successfully deleted
    - If selected storages were deleted, display these storages
    - If all storages were deleted, display all the storages
- If the storages couldn't be removed from the table, inform the user that it couldn't be deleted
- Display the updated storage table
    - If all storages were deleted, display an empty table informing the user that there are no storages

Add Storage page

- Form to allow the user to input the required information for a new storage
  - Text box to enter the storage name (required)
    - If the text box is left blank and the user tries to add the storage, they will be blocked
  - Drop down menu to select a facility/location for the storage (already selected with a default value)
  - 'Add' button to add the storage and proceed to the next page, and a rest button to restart
  - When pressing the add button, a warning message should be displayed to ask the user if they are sure they want to add the storage

Storage Added page

- If the storage was added to the database, display a message informing the user that it was successfully added
- Display the updated storage table

Change Storage Details page

- Form to allow user to select an storage and change their details
  - Drop down menu with the storages as options (already selected with a default value)
  - Drop down menu with the new facilities as options (optional)
  - Text box to input the storages new name (optional)
  - 'Change' button to proceed and a 'Reset' button to restart if needed

Storage Details Changed page

- If the storage's details have been changed, display a message to inform the user that it has been successfully changed
  - Display each of the changes made to the storage (storage name and/or facility/location)
- If the storage's details couldn't be changed, display a message to inform the user that the changes couldn't be made
- Display a table of the updated storages

View/Edit Equipment Section

View/Edit Equipment page

- All equipment displayed in a table format
  - Column for 'Equipment ID'
  - Column for 'Equipment Name'
  - Column for 'Number Available'
  - Column for 'Storage Name'
  - Column for deleting equipment (like 'Delete Booking' section)
    - Each row has a checkbox to allow the user to check/fill in the box
    - The column header has a 'Delete' button to allow user to delete select number of equipment, and a reset button to allow user to restart
  - Clear button to allow user to delete all equipment in the storage table
  - If the equipment table is empty and there are no equipment in the database, display an empty table informing the user that there are no equipment
- 'Add New Equipment' button to redirect user to Add equipment page
- 'Make changes to equipment' button to redirect user to edit equipment page

Equipment Deleted page

- If the equipment was deleted from the database, display to inform the user that it was successfully deleted
  - If selected equipment were deleted, display these equipment
  - If all equipment were deleted, display all the equipment
- If the equipment couldn't be removed from the table, inform the user that it couldn't be deleted

- Display the updated equipment table
    - If all equipment were deleted, display an empty table informing the user that there are no equipment

Add Equipment page

- Form to allow the user to input the required information for a new equipment
    - Text box to enter the equipment name (required)
        - If the text box is left blank and the user tries to add the equipment, they will be blocked
    - Drop down menu to select a storage space for the equipment (already selected with a default value)
    - Number input to enter the number available
    - 'Add' button to add the equipment and proceed to the next page, and a rest button to restart
    - When pressing the add button, a warning message should be displayed to ask the user if they are sure they want to add the equipment

Equipment Added page

- If the equipment was added to the database, display a message informing the user that it was successfully added
- Display the updated equipment table

Change Equipment Details page

- Form to allow user to select an equipment and change their details
    - Drop down menu with the equipment as options (already selected with a default value)
    - Drop down menu with the new storages as options (optional)
    - Text box to input the new equipment name (optional)
    - Number input to enter the new number available (optional)
    - 'Change' button to proceed and a 'Reset' button to restart if needed

Equipment Details Changed page

- If the equipment details have been changed, display a message to inform the user that it has been successfully changed
    - Display each of the changes made to the equipment (equipment name and/or storage and/or number)
- If the equipment details couldn't be changed, display a message to inform the user that the changes couldn't be made
- Display a table of the updated equipment

View/Edit User Section

View/Edit User page

- All users displayed in a table format
    - Column for 'User ID'
    - Column for 'User Name'
    - Column for 'House'
    - Column for 'User Type'
    - Column for deleting users (like 'Delete Booking' section)
        - Each row has a checkbox to allow the user to check/fill in the box
        - The column header has a 'Delete' button to allow user to delete select users, and a reset button to allow them to restart
    - Clear button to allow user to delete all users in the user table
    - If the users table is empty and there are no users in the table, display an empty table informing the user that there are no users
- 'Add New User' button to redirect user to Add users page
- 'Make changes to user' button to redirect user to edit user page

User Deleted page

- If the user was deleted from the database, display to inform the user that it was successfully deleted
    - If selected users were deleted, display these users
    - If all users were deleted, display all the users
- If the users couldn't be removed from the table, inform the user that it couldn't be deleted
- Display the updated users table
    - If all users were deleted, display an empty table informing the user that there are none

Add User page

- Form to allow the user to input the required information for a new user/staff
    - Text box to enter the user name (required)
        - If the text box is left blank and the user tries to add the user, they will be blocked
    - Text box to enter the user ID (required)
        - If the text box is left blank and the user tries to add the user, they will be blocked
    - Drop down menu to select a house for the user (already selected with a default value)
    - Drop down menu to select a user type for the user (already selected with a default value)
    - 'Add' button to add the user and proceed to the next page, and a rest button to restart
    - When pressing the add button, a warning message should be displayed to ask the user if they are sure they want to add the user

User Added page

- If the user was added to the database, display a message informing the user that it was successfully added
    - Increase the number of users/staff in the house table by 1
- Display the updated users table

Change User Details page

- Form to allow user to select a user and change their details
    - Drop down menu with the users as options (already selected with a default value)
    - Drop down menu with the new house as options (optional)
    - Drop down menu with the new user types as options (optional)
    - Text box to input the new user name (optional)
    - Text box to input the new user email (optional)
    - 'Change' button to proceed and a 'Reset' button to restart if needed

User Details Changed page

- If the user details have been changed, display a message to inform the user that it has been successfully changed
    - Display each of the changes made to the user (user name and/or user ID and/or house and/or user type)
    - If the user's house was changed increase the number of staff/users by 1
- If the user details couldn't be changed, display a message to inform the user that the changes couldn't be made
- Display a table of the updated users

View/Edit User Type Section

View/Edit User Type page

- All user types displayed in a table format
    - Column for 'User Type ID'
    - Column for 'User Type Name'
    - Column for deleting user types (like 'Delete Booking' section)

        - Each row has a checkbox to allow the user to check/fill in the box
        - The column header has a ‘Delete’ button to allow user to delete select user types, and a reset button to allow them to restart
    - Clear button to allow user to delete all user types in the user type table
    - If the user types table is empty and there are no user types in the database, display an empty table informing the user that there are no user types
- ‘Add New User Type’ button to redirect user to Add user types page
- ‘Make changes to User Type’ button to redirect user to edit user type page

User Type Deleted page

- If the user type was deleted from the database, display to inform the user that it was successfully deleted
    - If selected user types were deleted, display these user types
    - If all user types were deleted, display all the user types
- If the user types couldn’t be removed from the table, inform the user that it couldn’t be deleted
- Display the updated user types table
    - If all user types were deleted, display an empty table informing the user that there are no user types

Add User Type page

- Form to allow the user to input the required information for a new user type
    - Text box to enter the user type name (required)
        - If the text box is left blank and the user tries to add the user type, they will be blocked
    - ‘Add’ button to add the user type and proceed to the next page, and a reset button to restart
    - When pressing the add button, a warning message should be displayed to ask the user if they are sure they want to add the user type

User Type Added page

- If the user type was added to the table, display a message informing the user that it was successfully added
- Display the updated user type table

Change User Type Details page

- Form to allow user to select a user type and change their details
    - Drop down menu with the user types as options (already selected with a default value)
    - Text box to input the new user type name (required)
        - If the user leaves this field blank, they will be blocked from adding the user type
    - ‘Change’ button to proceed and a ‘Reset’ button to restart if needed

User Type Details Changed page

- If the user types details have been changed, display a message to inform the user that it has been successfully changed
- If the user types details couldn’t be changed, display a message to inform the user that the changes couldn’t be made
- Display a table of the updated user types

View/Edit House Section

View/Edit House page

- All houses displayed in a table format
    - Column for ‘House ID’
    - Column for ‘House Name’
    - Column for ‘Number of Users’
    - Column for ‘Number of Students

        - Column for deleting houses (like ‘Delete Booking’ section)
            - Each row has a checkbox to allow the user to check/fill in the box
            - The column header has a ‘Delete’ button to allow user to delete select houses, and a reset button to allow them to restart
        - Clear button to allow user to delete all houses in the database
        - If the house table is empty display an empty table informing the user that there are no houses
    - ‘Add New House’ button to redirect user to Add House page
    - ‘Make changes to House’ button to redirect user to edit House page

House Deleted page

- If the houses were deleted from the database, display to inform the user that it was deleted
    - If selected houses were deleted, display these houses
    - If all houses were deleted, display all the houses
- If the houses couldn’t be removed from the table, inform the user that it couldn’t be deleted
- Display the updated houses table
    - If all houses were deleted, display an empty table informing the user that there are no houses

Add House page

- Form to allow the user to input the required information for a new house
    - Text box to enter the house name (required)
        - If the text box is left blank and the user tries to add the house, they will be blocked
    - ‘Add’ button to add the house and proceed to the next page, and a reset button to restart
    - When pressing the add button, a warning message should be displayed to ask the user if they are sure they want to add the house

House Added page

- If the house was added to the table, display a message informing the user that it was added
- Display the updated houses table

Change House Details page

- Form to allow user to select a house and change their details
    - Drop down menu with the houses as options (already selected with a default value)
    - Text box to input the new house name (optional)
    - Number input to enter the new number of students (optional)
    - ‘Change’ button to proceed and a ‘Reset’ button to restart if needed

House Details Changed page

- If the house’s details have been changed, display a message to inform the user that it has been successfully changed
    - Display each of the changes made (name and/or number of students
- If the house’s details couldn’t be changed, display a message to inform the user that the changes couldn’t be made
- Display a table of the updated houses

View/Edit Year Group Section

View/Edit Year Group page

- All year groups displayed in a table format
    - Column for ‘Group ID’
    - Column for ‘Group Name’
    - Column for ‘Number of Students
- ‘Make changes to Year Group button to redirect user to edit Year Group page

Change Year Group Details page

- Form to allow user to select a year group and change their details

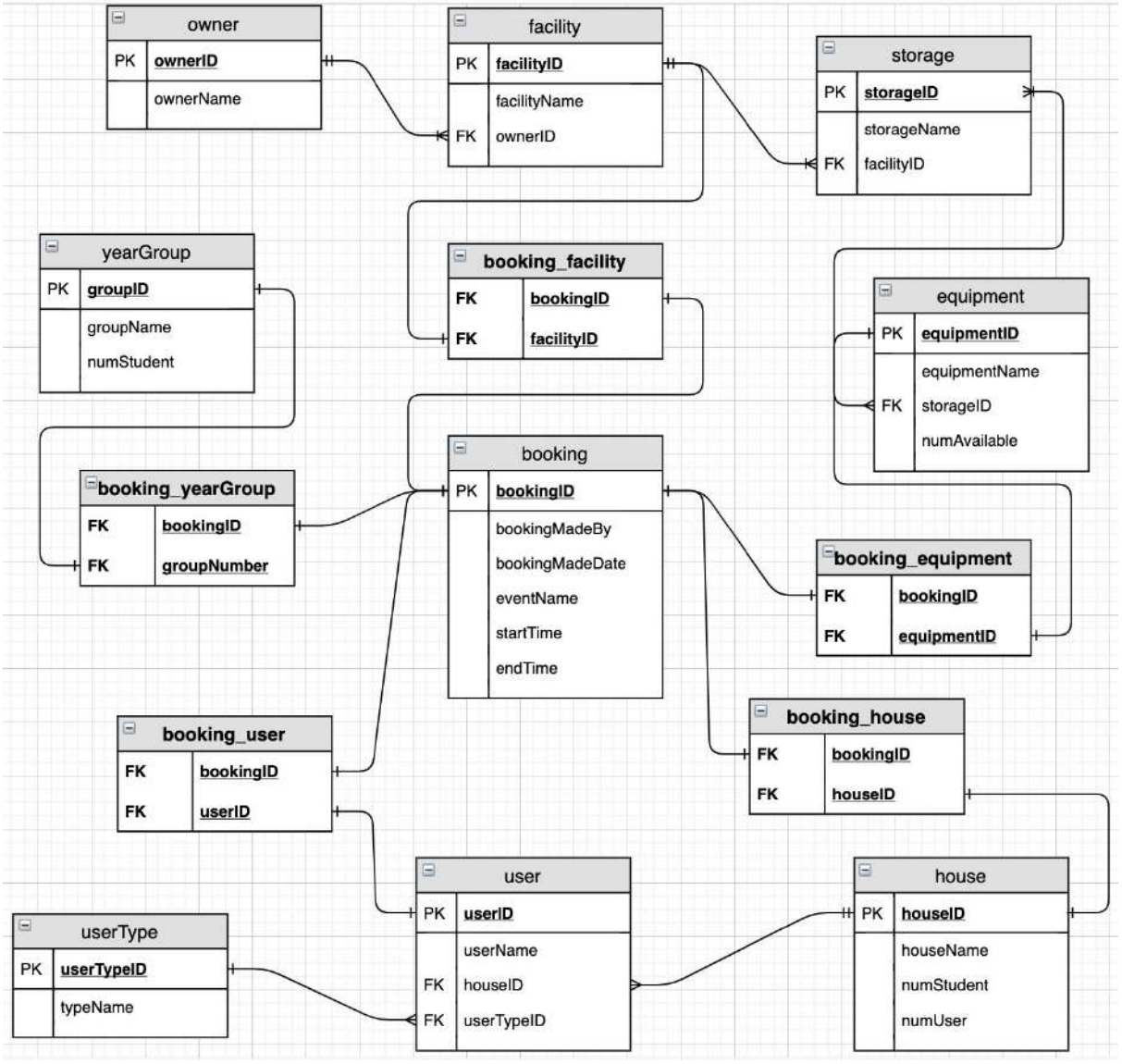
- Drop down menu with the year groups as options (already selected with a default value)
- Number input to enter the new number of students (required)
  - User will be blocked if not filled in
- ‘Change’ button to proceed and a ‘Reset’ button to restart if needed

Year Group Details Changed page

- If the year group’s details have been changed, display a message to inform the user that it has been successfully changed
- If the year group’s details couldn’t be changed, display a message to inform the user that the changes couldn’t be made
- Display a table of the updated year groups

Design

Database Relationship Design



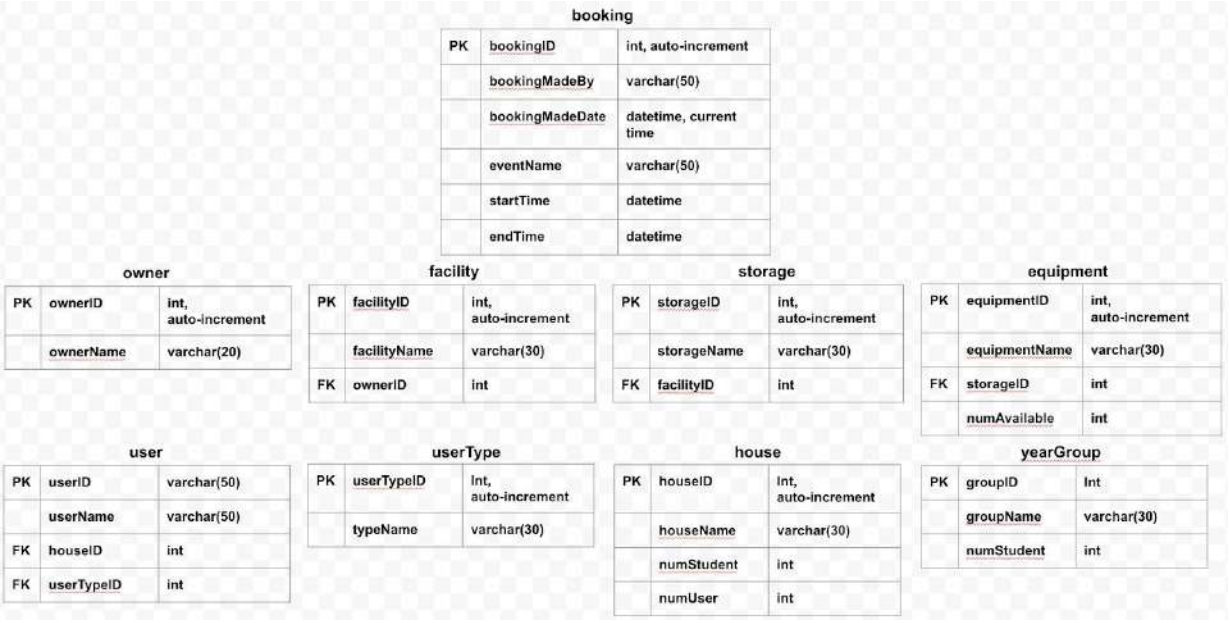
When developing the system, the database design is by far the most important compared to the other aspects as the database acts as the foundation for the whole system.
By designing the database thoroughly first, I have concluded that the 'booking' table will be the main, central entity from which all the other entities connect to and branch out from.

In terms of the booking entity itself, instead of having one large booking table to store all data such as facility/equipment/user/house/year group used in a particular booking, I have decided to separate the table into further connecting tables, and this can be seen from the ERD above. For example, the facilities used in a given booking will be stored in the 'booking_facility' table. This decision had been made due to the realisation that a single booking will most likely have more than one facility/equipment/user/house/year group. Therefore, it is not possible to store multiple values for a single booking. By having the connecting tables, multiple values can be assigned to one booking while keeping records and tables unique and normalised, hence the use of one-to-one and many-to-one relationships between tables and not many-to-many.

When adding or deleting a booking, the order in which records are added/removed first will be affected by this central table. In other words, when adding a new booking, a new record must first be added to the ‘booking’ table by inputting into the ‘bookingMadeBy’, ‘bookingMadeDate’, ‘eventName’, ‘startTime’ and ‘endTime’ records, where the ‘bookingMadeDate’ record is filled in automatically using the current date/time. Here, the ‘bookingID’ record is also automatically inputted using the ‘AUTO_INCREMENT’ function in MySQL. Once this is done and the ‘bookingID’ is determined, the other remaining information can be inputted into the connecting tables using the new ‘bookingID’. On the other hand, when deleting a booking, the connecting tables must have their records removed first before the central ‘booking’ table’s record is removed. In this case it is not possible to delete the record in the ‘booking’ table first because the connecting tables will have fields acting as foreign keys to the ‘booking’ table.

The ‘owner’ entity stores all the existing owners who manage the different facilities. Therefore the ‘ownerID’ acts as a unique PK which connects to the ‘facility’ entity.
The ‘facility’ entity stores the different facilities in which sports events take place in. Each facility will have a storage space where the equipment are stored, so the ‘facilityID’ is the PK that connects to the ‘storage’ entity.
The ‘storage’ entity stores the information about each storage space or shed where the equipment is stored.
The ‘equipment’ entity stores the information about all the equipment that can be used during an event, and the ‘storageID’ field references the ‘storage’ entity where it is stored inside.
The ‘user’ entity stores all the information about the staff and school representatives who will be supervising or leading the session. Each user will have their type and house, so these fields are connected and referenced in the ‘house’ and ‘userType’. Therefore the ‘house’ and ‘userType’ entities have further information about each of the users, and the ‘house’ entities can additionally be used in a booking through the connecting table.
Finally, the ‘yearGroup’ entity stores the number of students in each year group.

Data Types

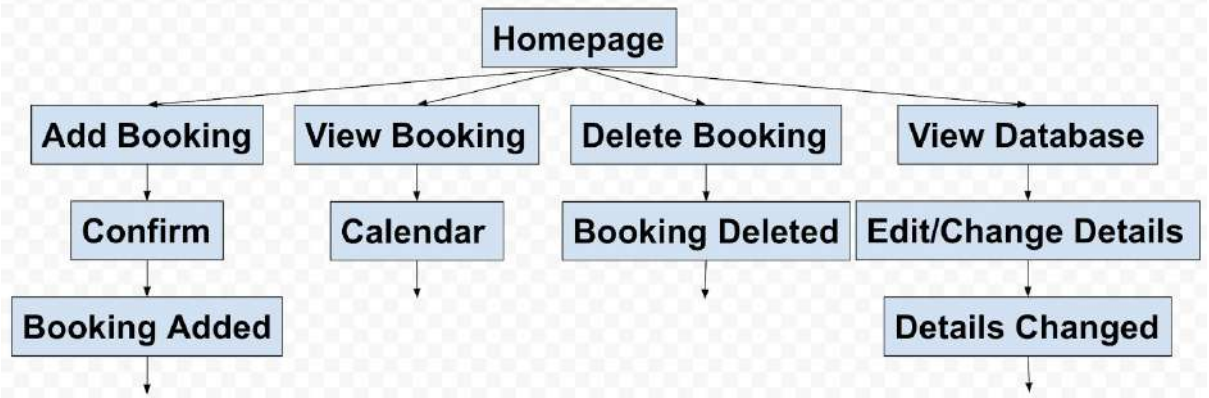


The use of appropriate data types is another important aspect of the product design. Although MySQL has support for several data types, my project only requires 3 main types: 'int', 'varchar' and 'datetime'. Firstly, the 'int' or integer data type is used to represent the number of a given object, for example the

number of available equipment or the number of students/teachers in a house. Additionally, the integer data type is used to provide each record/object with a unique ID (primary key). In most cases where the PK needs to be identified, the extra functionality of 'auto-increment' will be used. This enables the database to increase the number by one each time a new record is added, eliminating the process of having to manually input a unique value. The only time this will not be the case is in the 'Year Group' entity/table where the identifier for each year group is already predetermined and will always be different to each other with no new records expected to be added. The 'datetime' data type is only used in the 'Booking' entity to represent the date and time of when the booking itself was made, and when the assigned event starts/ends. The final data type is 'varchar' which can use a variety of different characters such as numbers and letters. This is used to represent names of different objects associated with its unique ID. The parameter (brackets) indicates the maximum number of characters that can be used per record/input value, and I have made this size inconsistent deliberately by considering what type of information will be stored for each entity. For example, the names of people/users and equipment will usually be longer and have an unknown range of possible values so has been set to 50 characters maximum. But in comparison, names for facility owners and houses will be predetermined where the name tends to be shorter so has been set to 20-30.

Website Structure

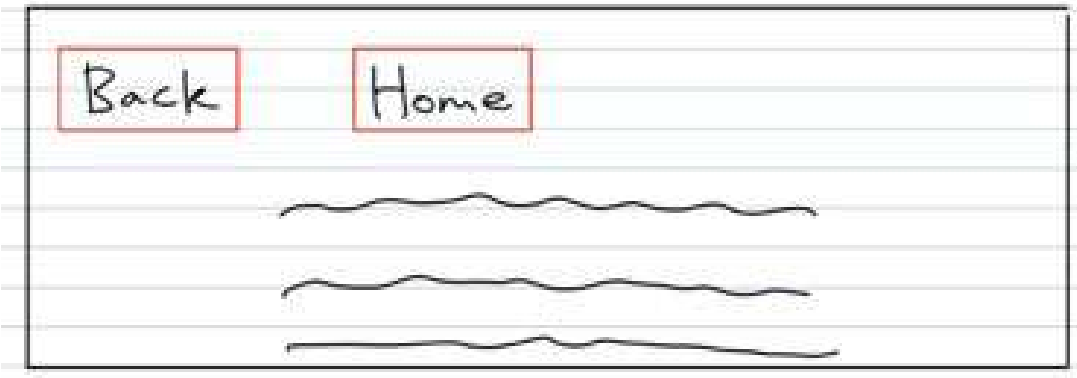
The website is the next main design aspect of the project. This is how the user will be able to interact with the back-end database and make changes to it. All the pages will be stored in the MAMP directory on the computer where the html files are accessed by the server. For my project, the web pages will be structured like a tree diagram where the root will be assigned to the home page where 4 different options (add booking, view booking, delete booking and view database) will be available to choose from. These 4 options will follow through to separate child nodes, and then these child nodes will act as parent nodes to further nodes.



Website UI Design

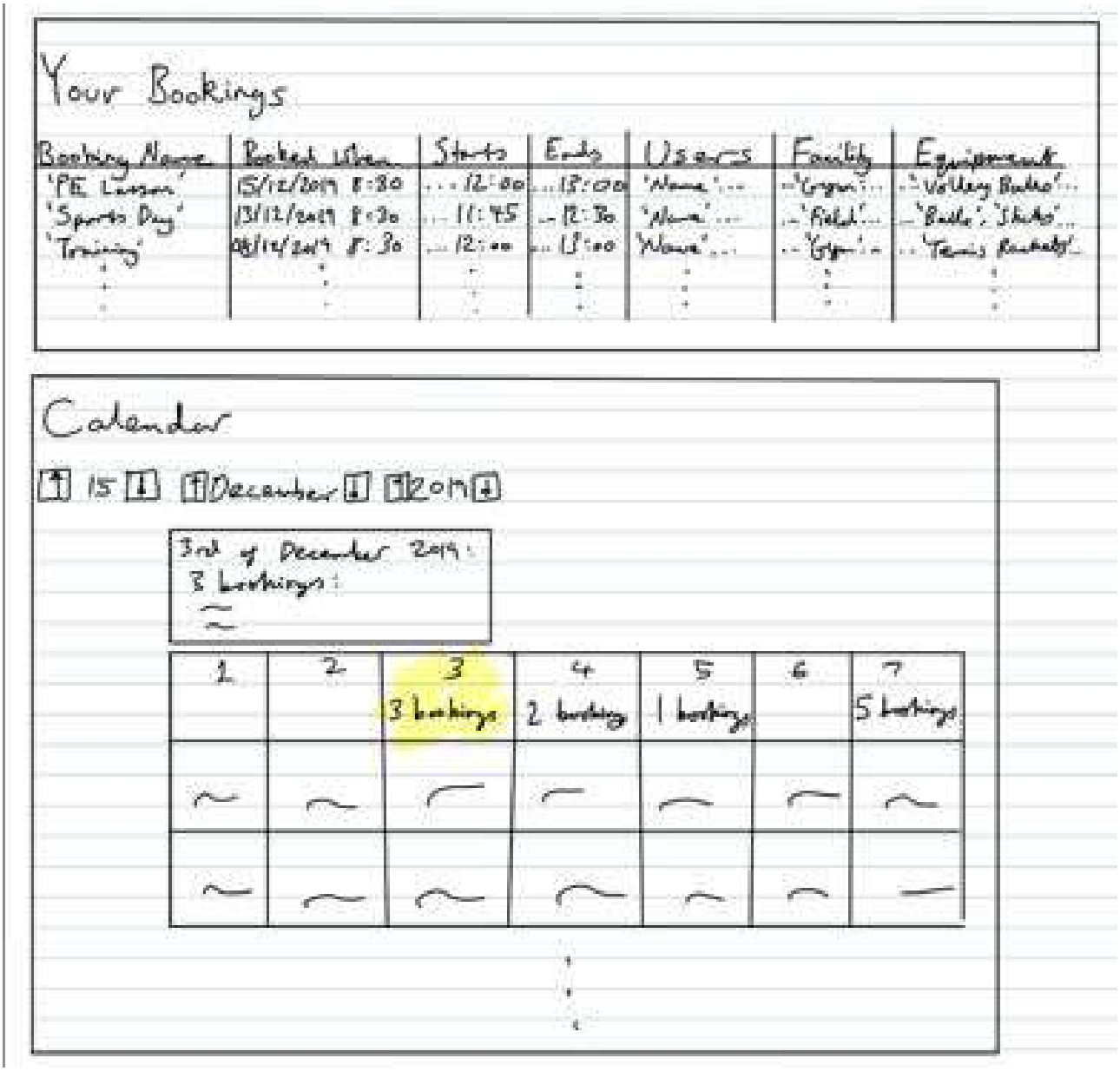
In terms of how the actual web pages and UI look, my idea is to make it as easy as possible to view, understand and navigate through. For example, in the home screen where the user first starts, the use of colourful and large buttons along with large text can allow the user to view the required information easily.

Above most web pages in my site (excluding the homepage) I will include two navigation buttons to go back to the previous page as well as a button to go back to the homepage. Although the browser already provides this, I believe that custom buttons would be beneficial not only for ease of use, but for times such as during the booking process where the user may need to go back a page and confirm their details. I have had previous problems where the browser navigation buttons took me back to the incorrect pages and caused frustration, so these custom buttons can be used to prevent that issue. However I will not include the back button for pages where the user should be blocked. For example this may be just after a booking was deleted as going back could risk errors occuring.



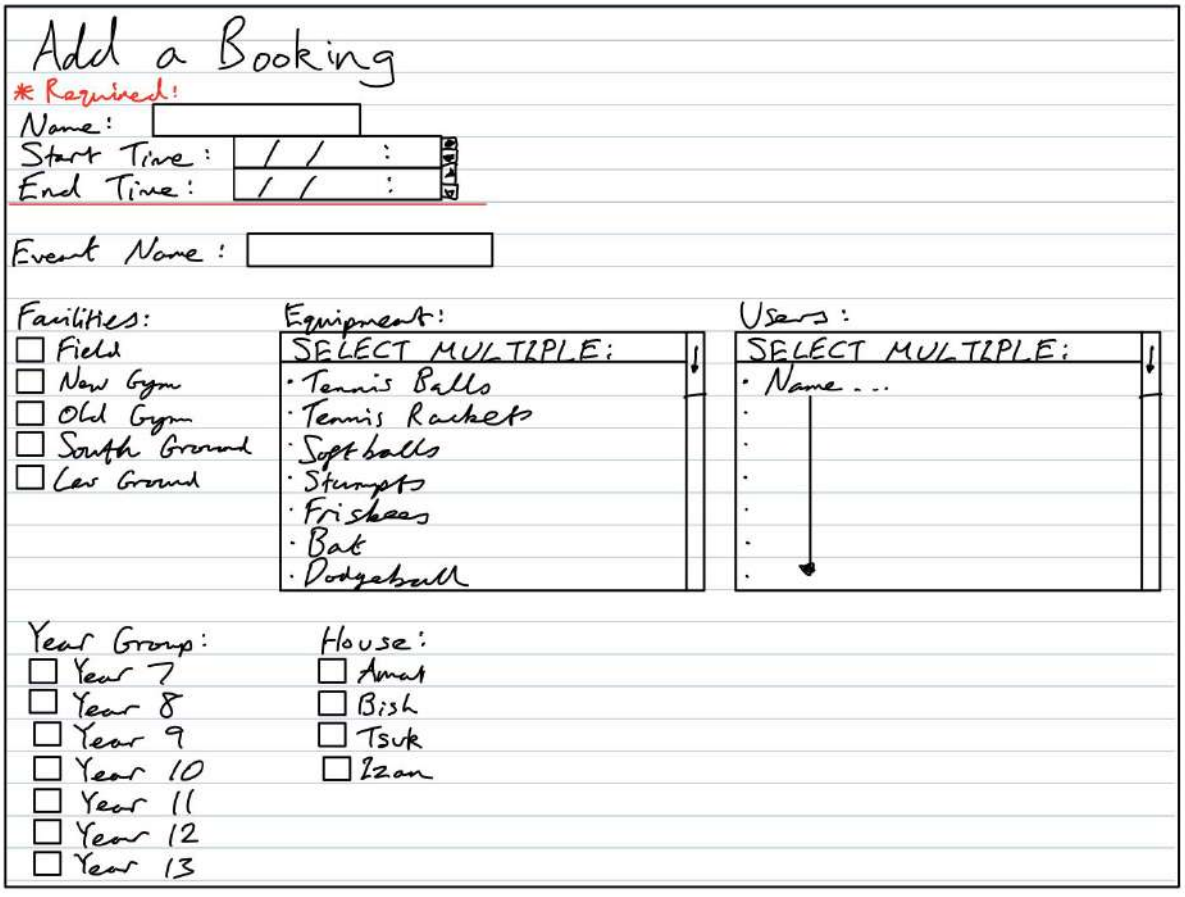
Displaying Bookings

In terms of displaying the bookings and useful information, I have designed and planned out 2 separate formats that can be used by choice. One of the formats will be a simple table displaying all the booking information. This table will have enough columns to include all the entity types. The webpage will retrieve the records from the ‘booking’ table and its connecting tables for each event, adding a new row to the table. If a booking has no facility/equipment/user/house/year group chosen, the cell will display ‘none’.
The second format is a calendar. This will look similar to a regular calendar, but on each date/cell, it will display the number of events there are, and if there are none, it will display ‘none’. Below this value, there will be a hyperlink or button that will allow the user to display the specific bookings that are starting and taking place on that date. Like all calendars, there will also be a header at the top that will display the current month, plus two buttons that will allow the user to move on to the next page or move back to the previous month.



Adding Bookings

For adding new bookings, my plan is to have a simplistic form based page which allows the user to enter all the information needed to make the booking. All sections of the booking will be based on each of the different database entities and will have a different type of form input. Firstly, the sections that will be part of the 'booking' table will all need to be filled out and required as this is the base of the booking. This includes fields such as 'name' (for the bookingMadeBy field), 'startTime' and 'endTime' for the booking. This is especially important as the times will later be used during the validation process to check for any overlap/double-booking with other, already existing bookings. Therefore these first 3 sections will be marked as 'Required' and will need to be filled out to proceed. The 'bookingMadeWhen' field can be done automatically by the database and SQL by entering the time in which the booking was added. Finally, the 'eventName' field can be made optional by using SQL queries to make the default event name 'PE Lesson' as most bookings are.
The remaining sections of the form will be added and be part of the other entities, or in this case the connecting tables. Therefore, each of the other sections will be made optional, while still allowing multiple items to be selected. Where there are only a few possible items to select such as year group, house, and facility, I will use checkboxes that can take multiple values. On the other hand, sections that can have a wide range of options such as equipment and supervising staff will be presented as a 'multiple select list' where the user can hold control/command and select multiple values. The webpage will have to connect to the database and retrieve all the values to display all the form options.



Deleting Bookings

The delete page will follow the same design and structure of the view page, where a table/list of the existing bookings will be displayed. But in this case, an extra column will be added with a checkbox in each row to allow the user to select multiple bookings to delete. Above the table, I will have a 'Delete' button that will take the selected values and redirect the user to a separate page which summarises the action made. Similarly, next to this button, I will have a 'Clear All' button that ignores any checkbox selections and redirects the user to the same or similar summary page where all bookings will be deleted from the database.

CSS (Cascading Style Sheets) Styling

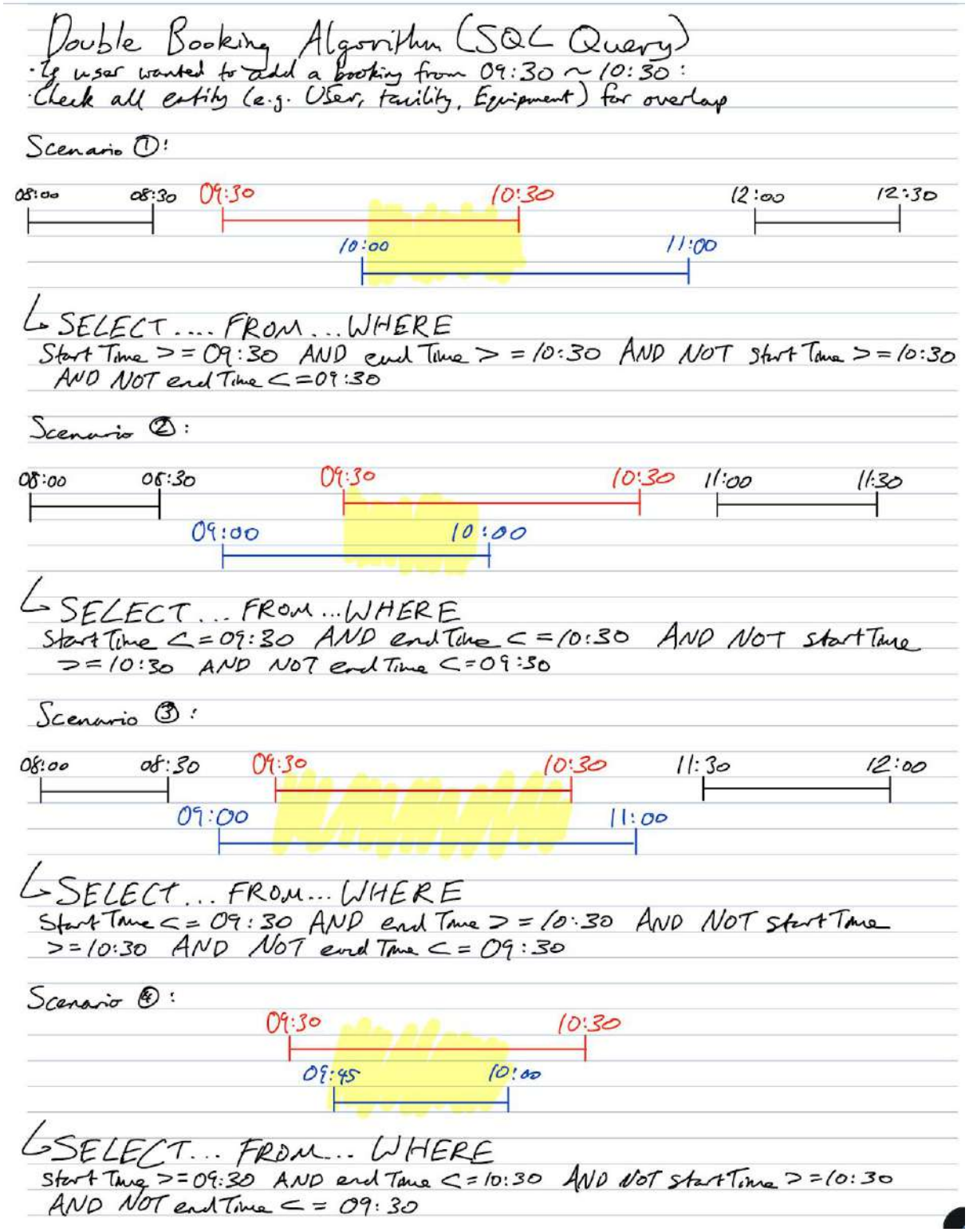
It is important that the website design remains consistent throughout the website. Having different types of buttons, navigation buttons and colours across web pages could potentially cause confusion and make the system less user-friendly. Therefore I plan to have one .css file within the MAMP directory which will have a standard set of design rules that all web pages will access and follow. In each web page's html/php file I will include the css file within the <head> and <style> tags. I will create custom classes which can be followed, for example, a button can follow a custom class that tells it to be a certain colour.

Security

This system is intended only to be accessed by the PE department and the school staff, so there are not many security concerns.

Double-Booking/Overlap Validation Algorithm (SQL)

One of the requests that the PE department had was the ability to see if any facilities, equipment and users were at risk of being double-booked with another lesson when adding a new booking. This was something missing from the current system and required the teachers to ask each other if there were any free slots. Therefore in my program, I have included an addition confirmation/validation page following the bookingform to make sure there are no scheduling issues.



When considering the overlap of bookings, I have identified the 4 different scenarios in which an overlap occurs. The first scenario is when another existing booking starts after another. The second is

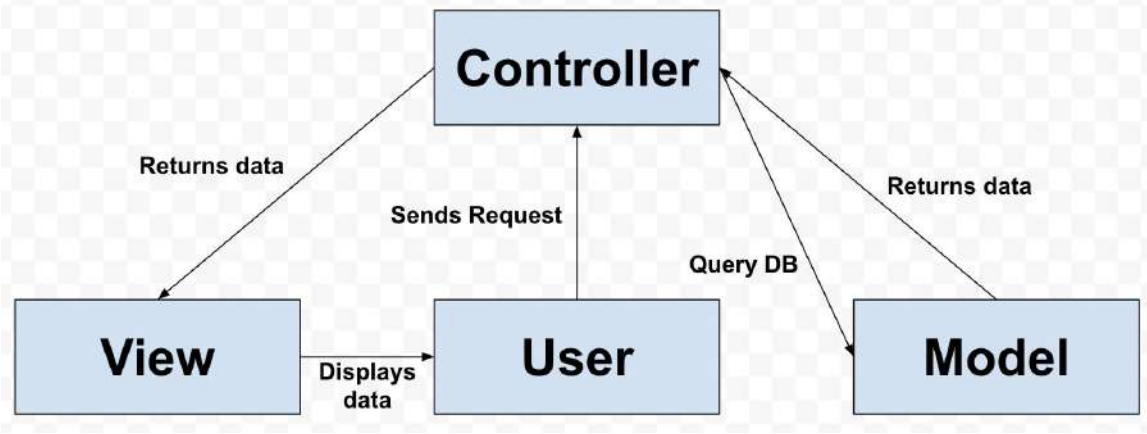
when the existing booking starts before a new one. The third is where an existing booking is within the time of another booking, and the final scenario is where the existing booking is running within the new one. For each of these scenarios, I have written different SQL statements that will all be run and return any records if any existing bookings fall within the range. When writing the SQL commands, there was one similarity in all of them. All queries had to include 2 NOT operators. This is so it excludes any bookings that could possibly match one of the first two comparisons but not actually be overlapping. Also, even if a booking is overlapping with another, this does not necessarily mean it is a double-book situation, as long as the items/staff needed are not over-run. Because of this, I have made sure to make the SELECT statements specific to the entity/item.

Programming Structure

The way in which the solution and program is written is also important for this project. I will be writing my solution in the OOP paradigm as I believe it will be more efficient and appropriate to this situation compared to POP. Although there are different ways in which a solution can be structured based on the final product and purpose, I will be following the MVC pattern/principle.

The MVC pattern stands for Model, View, and Controller and it is a popular way of organising code based on its main functions, where the main characteristic is that each section of the code has a distinct purpose. The 'Model' section tends to be based on real-world things. This code can hold raw data, or it will define the essential components of the final product. In other words, the model section tends to be based on the database where all the data will need to be retrieved from. The 'Controller' section acts as a liaison between the Model and the View, receiving user input and deciding what to do with it, and ties together the model and the view. Finally, the 'View' section is made up of all the functions that directly interact with the user. This is the code that builds up the main user interface, and otherwise defines how your user sees and interacts with it.

This method of coding makes it smoother to manage especially when the project is large scale, and it also allows revisiting the application to be more pleasant. Although some projects have all of the program on a single page, the advantage that my web-based solution has is that the PHP and HTML code can be split up into separate pages in the root folder/directory, where it can later be called (or 'included') in another section of code where necessary.

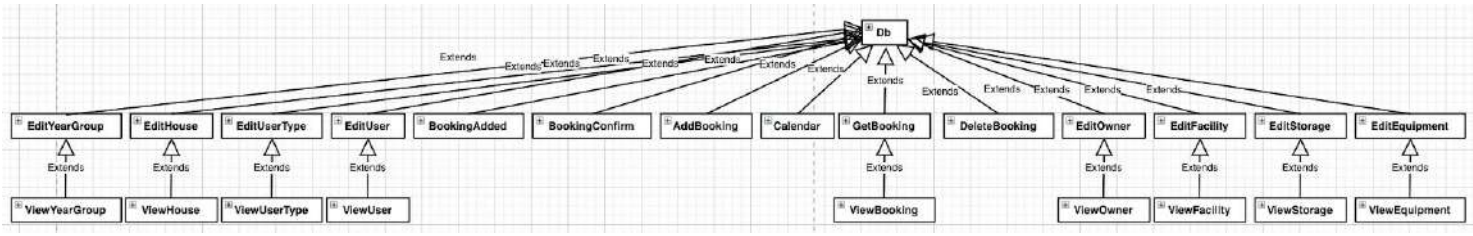


In my case, each section of my solution will be part of one of the 3 sections, and all sections will be a separate class that later be referenced by instantiating it. Firstly, the 'Model' section of the code will only consist of one class called 'Db' because for my project the database will be where all the necessary information will be retrieved from by the controller. Next, my 'Controller' section will consist of multiple classes such as Get Bookings' and are all based on sending SQL queries to the the database model section, retrieving the requested records, and then storing them in a list to be passed onto the next section. Finally, the 'View' section of code consists of classes such as 'View Bookings' where its main functions is to retrieve the list of records from the controller and displaying it in an appropriate way such as a table, or calendar. This is the basic structure of my solution but not all

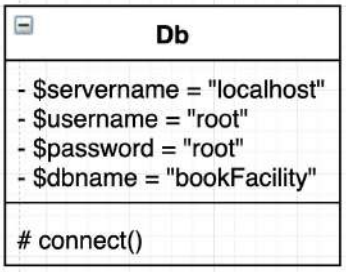
groups of classes such as the ‘Delete’ functions will have a dedicated view or controller page. For some instances, a form will be submitted and immediately redirected to the view section and running the code there without referencing a controller.

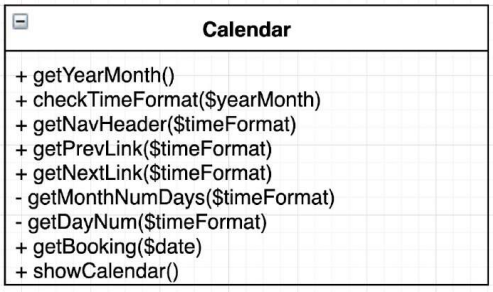
Class Diagram

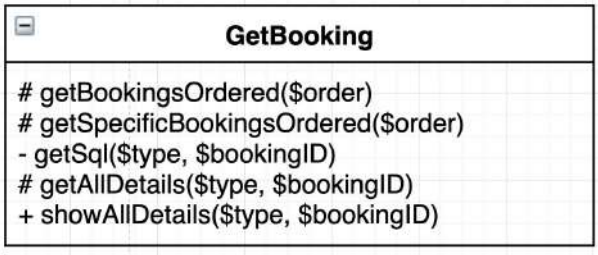
As described above, I will be writing my solution in OOP, and so a class diagram will be necessary to get a better picture of what the overall product will look like.

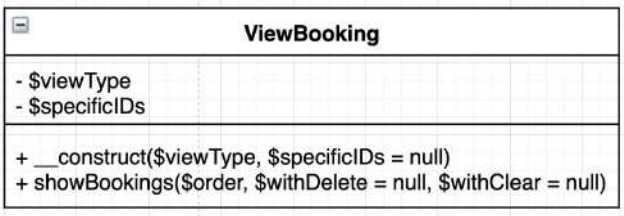


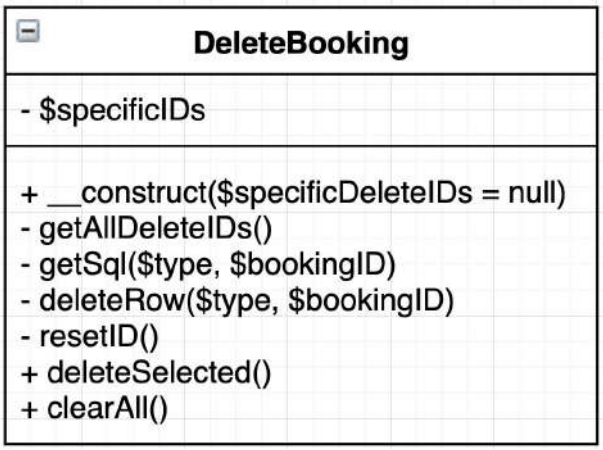
The class diagram above is structured in the MVC pattern. The first row, or the Db class will be the model section where it connects to the database. The next row of classes will be the control section where each class will extend from the Db class and retrieve data from the database by sending queries. The final row of classes will be the view sections where they extend from the control classes to display the data that was retrieved from the database.
Below are each of the classes in more depth showing each of the methods and properties it possesses:

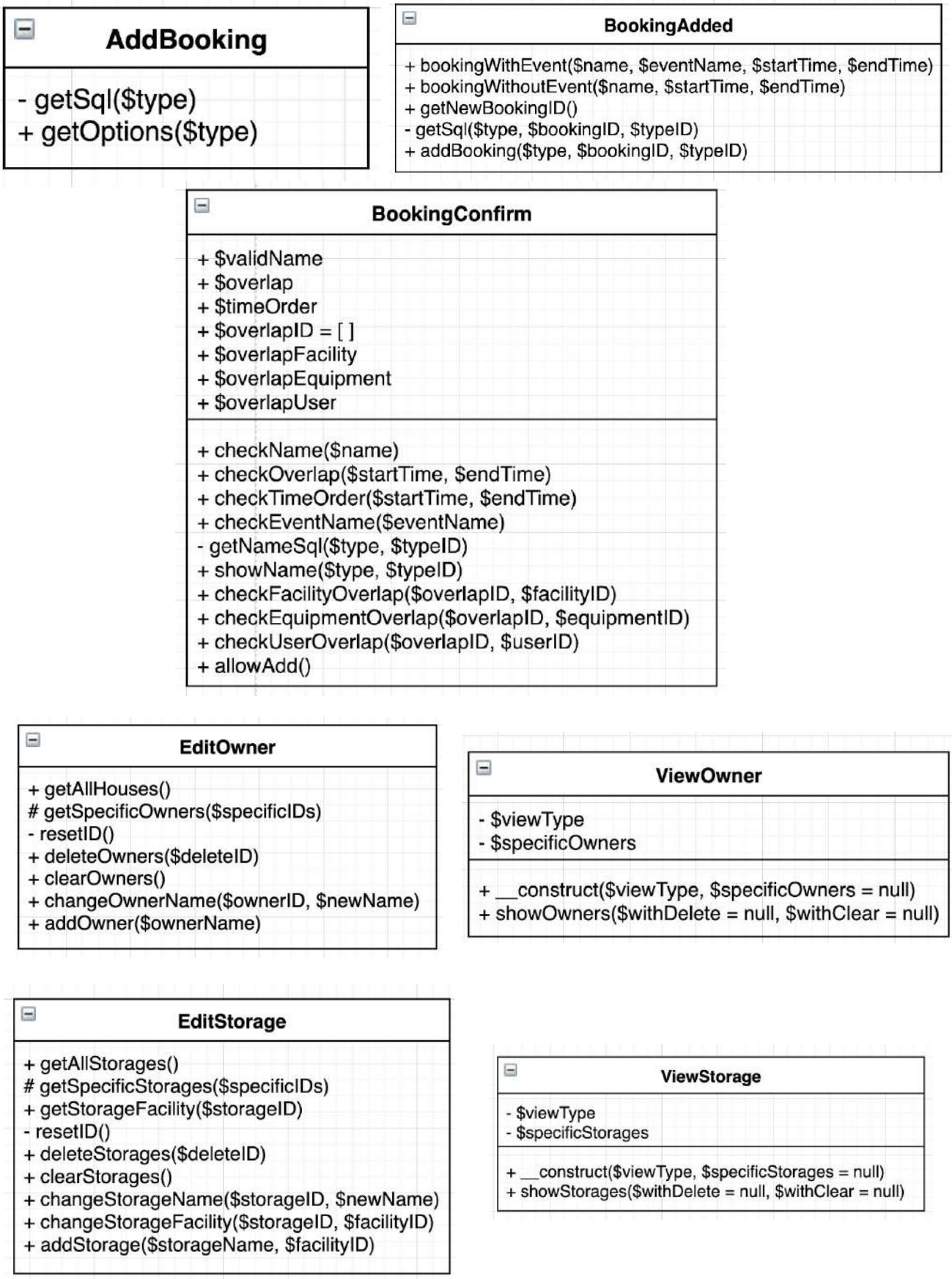


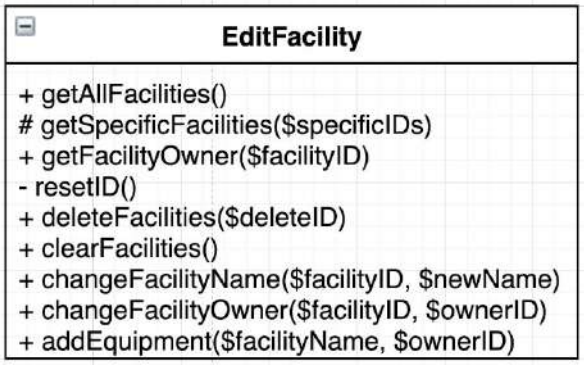


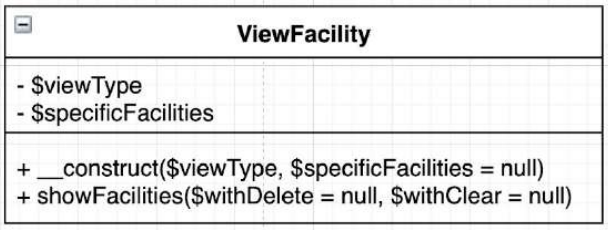


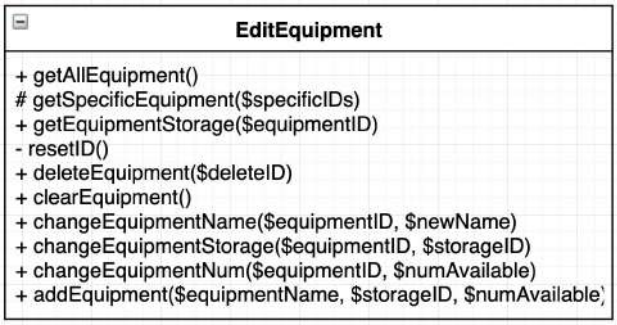


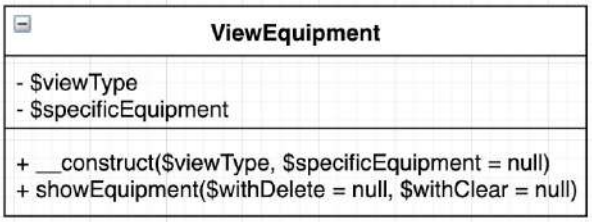


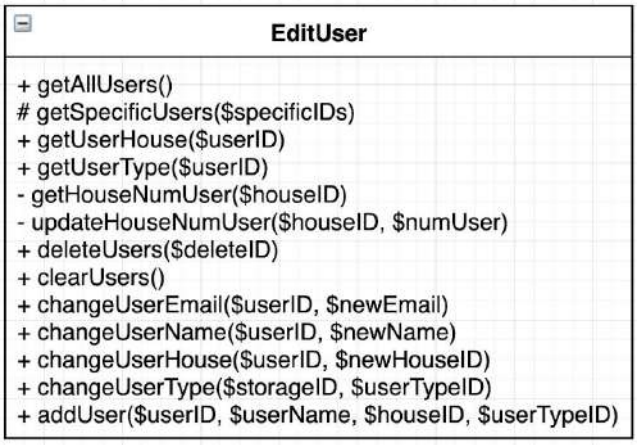

AddBooking
- getSql($type)
+ getOptions($type)
BookingAdded
+ bookingWithEvent($name, $eventName, $startTime, $endTime)
+ bookingWithoutEvent($name, $startTime, $endTime)
+ getNewBookingID()
- getSql($type, $bookingID, $typeID)
+ addBooking($type, $bookingID, $typeID)
BookingConfirm
+ $validName
+ $overlap
+ $timeOrder
+ $overlapID = [ ]
+ $overlapFacility
+ $overlapEquipment
+ $overlapUser
+ checkName($name)
+ checkOverlap($startTime, $endTime)
+ checkTimeOrder($startTime, $endTime)
+ checkEventName($eventName)
- getNameSql($type, $typeID)
+ showName($type, $typeID)
+ checkFacilityOverlap($overlapID, $facilityID)
+ checkEquipmentOverlap($overlapID, $equipmentID)
+ checkUserOverlap($overlapID, $userID)
+ allowAdd()
EditOwner
+ getAllHouses()
# getSpecificOwners($specificIDs)
- resetID()
+ deleteOwners($deleteID)
+ clearOwners()
+ changeOwnerName($ownerID, $newName)
+ addOwner($ownerName)
ViewOwner
- $viewType
- $specificOwners
+ __construct($viewType, $specificOwners = null)
+ showOwners($withDelete = null, $withClear = null)
EditStorage
+ getAllStorages()
# getSpecificStorages($specificIDs)
+ getStorageFacility($storageID)
- resetID()
+ deleteStorages($deleteID)
+ clearStorages()
+ changeStorageName($storageID, $newName)
+ changeStorageFacility($storageID, $facilityID)
+ addStorage($storageName, $facilityID)
ViewStorage
- $viewType
- $specificStorages
+ __construct($viewType, $specificStorages = null)
+ showStorages($withDelete = null, $withClear = null)

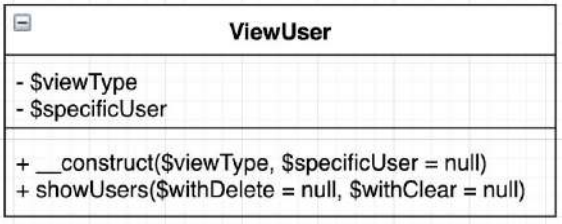

EditFacility
+ getAllFacilities()
# getSpecificFacilities($specificIDs)
+ getFacilityOwner($facilityID)
- resetID()
+ deleteFacilities($deleteID)
+ clearFacilities()
+ changeFacilityName($facilityID, $newName)
+ changeFacilityOwner($facilityID, $ownerID)
+ addEquipment($facilityName, $ownerID)

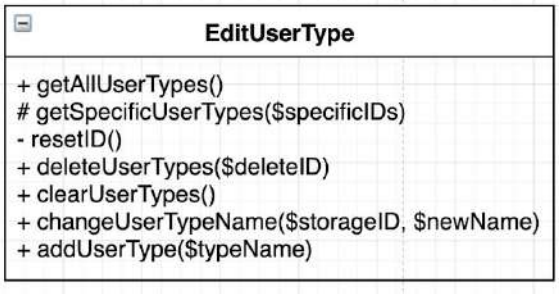

ViewFacility
- $viewType
- $specificFacilities
+ __construct($viewType, $specificFacilities = null)
+ showFacilities($withDelete = null, $withClear = null)

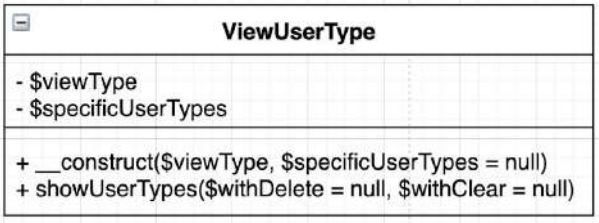

EditEquipment
+ getAllEquipment()
# getSpecificEquipment($specificIDs)
+ getEquipmentStorage($equipmentID)
- resetID()
+ deleteEquipment($deleteID)
+ clearEquipment()
+ changeEquipmentName($equipmentID, $newName)
+ changeEquipmentStorage($equipmentID, $storageID)
+ changeEquipmentNum($equipmentID, $numAvailable)
+ addEquipment($equipmentName, $storageID, $numAvailable)

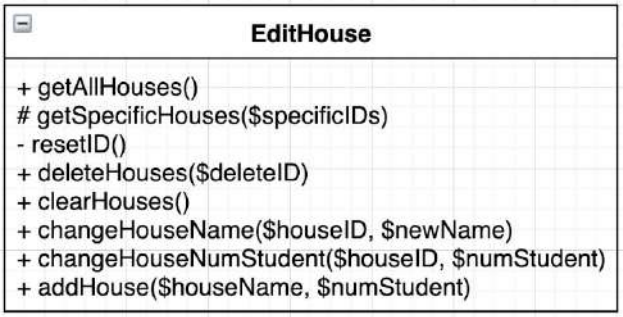

ViewEquipment
- $viewType
- $specificEquipment
+ __construct($viewType, $specificEquipment = null)
+ showEquipment($withDelete = null, $withClear = null)

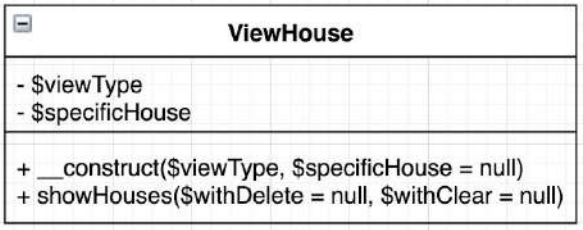

EditUser
+ getAllUsers()
# getSpecificUsers($specificIDs)
+ getUserHouse($userID)
+ getUserType($userID)
- getHouseNumUser($houseID)
- updateHouseNumUser($houseID, $numUser)
+ deleteUsers($deleteID)
+ clearUsers()
+ changeUserEmail($userID, $newEmail)
+ changeUserName($userID, $newName)
+ changeUserHouse($userID, $newHouseID)
+ changeUserType($storageID, $userTypeID)
+ addUser($userID, $userName, $houseID, $userTypeID)

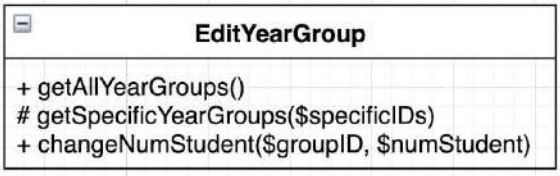

ViewUser
- $viewType
- $specificUser
+ __construct($viewType, $specificUser = null)
+ showUsers($withDelete = null, $withClear = null)

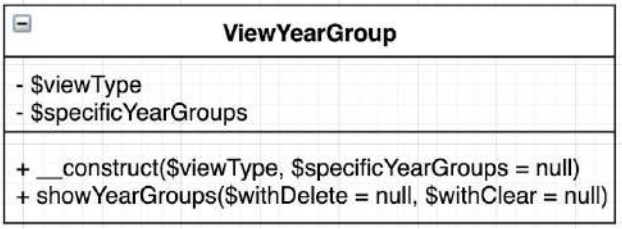

EditUserType
+ getAllUserTypes()
# getSpecificUserTypes($specificIDs)
- resetID()
+ deleteUserTypes($deleteID)
+ clearUserTypes()
+ changeUserTypeName($storageID, $newName)
+ addUserType($typeName)


ViewUserType
- $viewType
- $specificUserTypes
+ __construct($viewType, $specificUserTypes = null)
+ showUserTypes($withDelete = null, $withClear = null)


EditHouse
+ getAllHouses()
# getSpecificHouses($specificIDs)
- resetID()
+ deleteHouses($deleteID)
+ clearHouses()
+ changeHouseName($houseID, $newName)
+ changeHouseNumStudent($houseID, $numStudent)
+ addHouse($houseName, $numStudent)

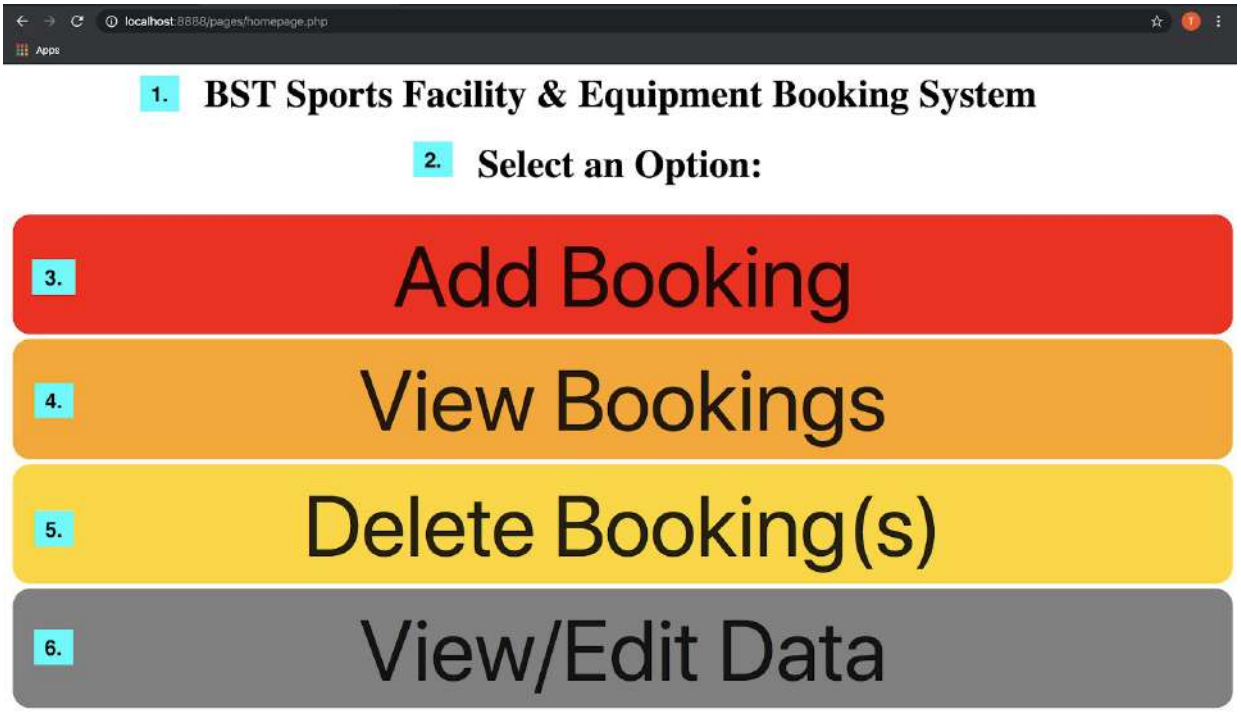

ViewHouse
- $viewType
- $specificHouse
+ __construct($viewType, $specificHouse = null)
+ showHouses($withDelete = null, $withClear = null)


EditYearGroup
+ getAllYearGroups()
# getSpecificYearGroups($specificIDs)
+ changeNumStudent($groupID, $numStudent)


ViewYearGroup
- $viewType
- $specificYearGroups
+ __construct($viewType, $specificYearGroups = null)
+ showYearGroups($withDelete = null, $withClear = null)

Testing

Homepage (homepage.php)



Test/Success Criteria

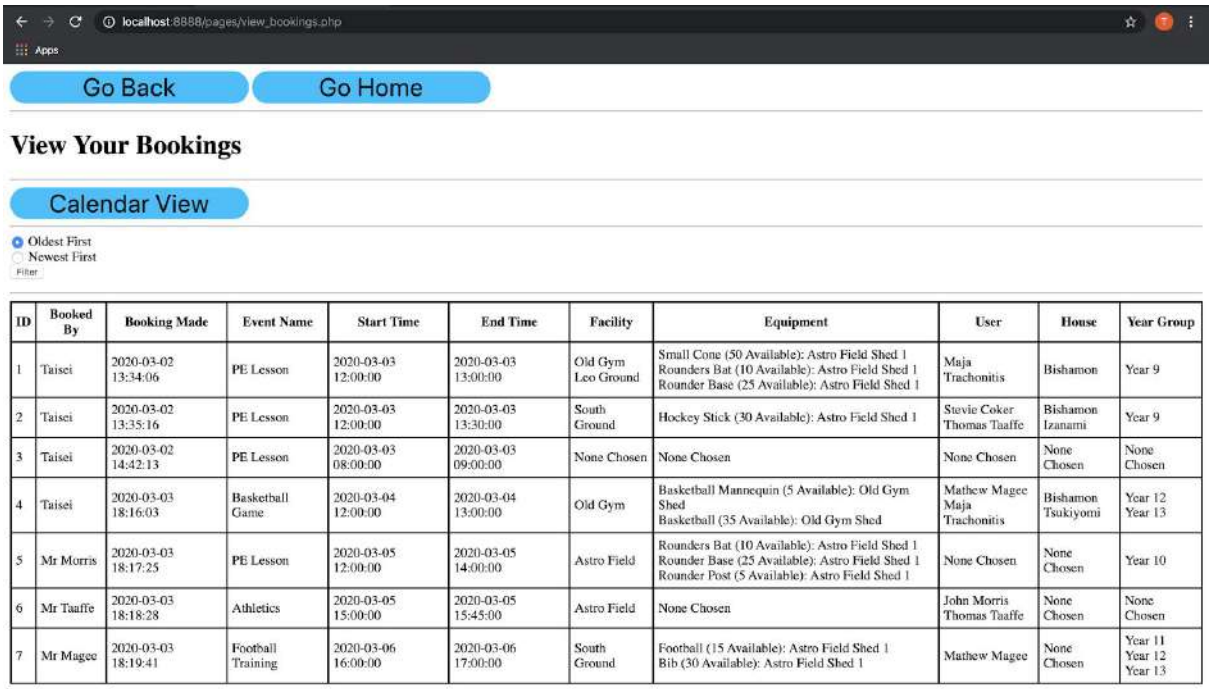
1. Clear title at top of page
2. Prompt to tell user to select an option

❖ Large buttons to redirect user to the different sections of the website:

3. Button for ‘Add Booking’ option
4. Button for ‘View Booking’ option
5. Button for ‘Delete Booking’ option
6. Button for ‘ View/Edit Database’ option

View Booking Section

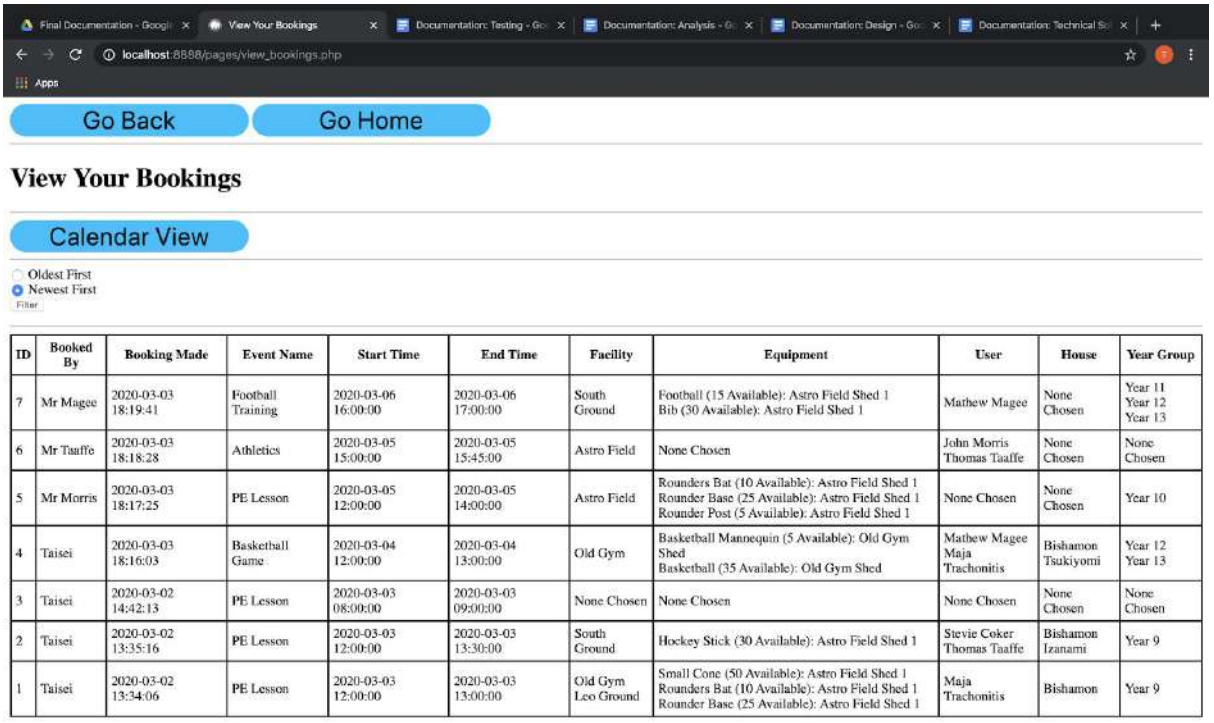
View Booking Page (view_bookings.php)



ID	Booked By	Booking Made	Event Name	Start Time	End Time	Facility	Equipment	User	House	Year Group
1	Taisei	2020-03-02 13:34:06	PE Lesson	2020-03-03 12:00:00	2020-03-03 13:00:00	Old Gym Leo Ground	Small Cone (50 Available): Astro Field Shed 1 Rounders Bat (10 Available): Astro Field Shed 1 Rounder Base (25 Available): Astro Field Shed 1	Maja Trachonitis	Bishamon	Year 9
2	Taisei	2020-03-02 13:35:16	PE Lesson	2020-03-03 12:00:00	2020-03-03 13:30:00	South Ground	Hockey Stick (30 Available): Astro Field Shed 1	Stevie Coker Thomas Taaffe	Bishamon Izanami	Year 9
3	Taisei	2020-03-02 14:42:13	PE Lesson	2020-03-03 08:00:00	2020-03-03 09:00:00	None Chosen	None Chosen	None Chosen	None Chosen	None Chosen
4	Taisei	2020-03-03 18:16:03	Basketball Game	2020-03-04 12:00:00	2020-03-04 13:00:00	Old Gym	Basketball Mannequin (5 Available): Old Gym Shed Basketball (35 Available): Old Gym Shed	Mathew Magee Maja Trachonitis	Bishamon Tsukiyomi	Year 12 Year 13
5	Mr Morris	2020-03-03 18:17:25	PE Lesson	2020-03-05 12:00:00	2020-03-05 14:00:00	Astro Field	Rounders Bat (10 Available): Astro Field Shed 1 Rounder Base (25 Available): Astro Field Shed 1 Rounder Post (5 Available): Astro Field Shed 1	None Chosen	None Chosen	Year 10
6	Mr Taaffe	2020-03-03 18:18:28	Athletics	2020-03-05 15:00:00	2020-03-05 15:45:00	Astro Field	None Chosen	John Morris Thomas Taaffe	None Chosen	None Chosen
7	Mr Magee	2020-03-03 18:19:41	Football Training	2020-03-06 16:00:00	2020-03-06 17:00:00	South Ground	Football (15 Available): Astro Field Shed 1 Bib (30 Available): Astro Field Shed 1	Mathew Magee	None Chosen	Year 11 Year 12 Year 13

Test/Success Criteria

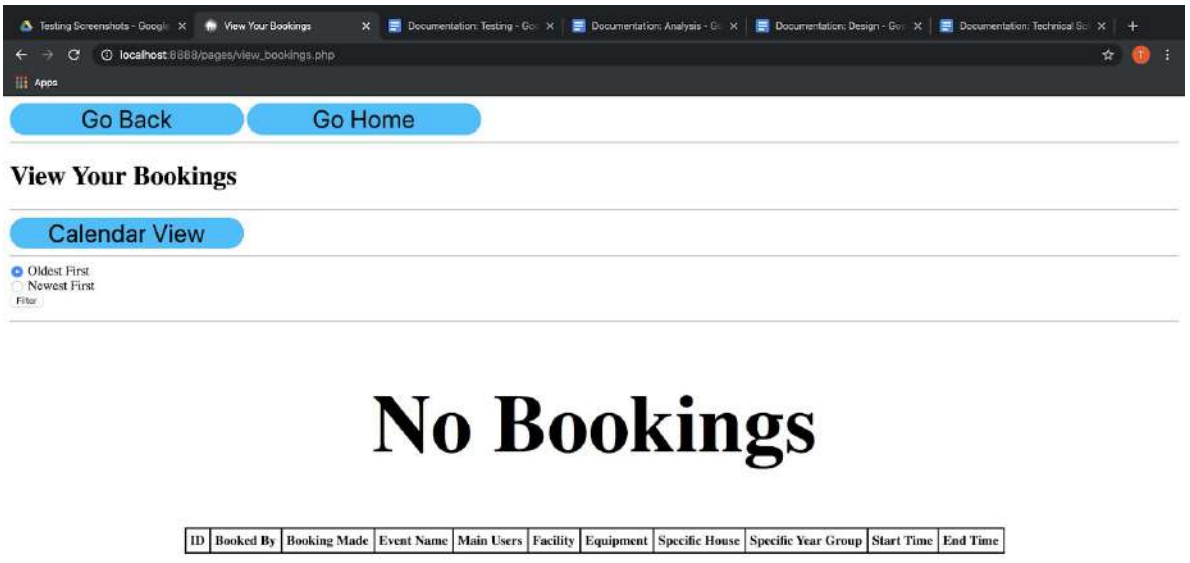
- Full-page table displaying all booking information
  - Table headers/columns
    - ‘Booking ID’
    - ‘Booking Made By’
    - ‘Booking Made Date’
    - ‘Event Name’
    - ‘Start Time’
    - ‘End Time’
    - ‘Booked facility’
    - ‘Booked equipment’
    - ‘Booked user’
    - ‘Booked house’
    - ‘Booked year group’
  - Table rows should match the number of bookings in the database and display all bookings made
  - If there are no facility/equipment/user/house/year group booked for a single booking, display ‘None’
  - If there are several facility/equipment/user/house/year group booked for a single booking display all of them one line at a time
- ‘Calendar View’ button to redirect user to the Calendar section



ID	Booked By	Booking Made	Event Name	Start Time	End Time	Facility	Equipment	User	House	Year Group
7	Mr Magee	2020-03-03 18:19:41	Football Training	2020-03-06 16:00:00	2020-03-06 17:00:00	South Ground	Football (15 Available): Astro Field Shed 1 Bib (30 Available): Astro Field Shed 1	Mathew Magee	None Chosen	Year 11 Year 12 Year 13
6	Mr Tauffe	2020-03-03 18:18:28	Athletics	2020-03-05 15:00:00	2020-03-05 15:45:00	Astro Field	None Chosen	John Morris Thomas Taaffe	None Chosen	None Chosen
5	Mr Morris	2020-03-03 18:17:25	PE Lesson	2020-03-05 12:00:00	2020-03-05 14:00:00	Astro Field	Rounders Bat (10 Available): Astro Field Shed 1 Rounder Base (25 Available): Astro Field Shed 1 Rounder Post (5 Available): Astro Field Shed 1	None Chosen	None Chosen	Year 10
4	Taisei	2020-03-03 18:16:03	Basketball Game	2020-03-04 12:00:00	2020-03-04 13:00:00	Old Gym	Basketball Mannequin (5 Available): Old Gym Shed Basketball (35 Available): Old Gym Shed	Mathew Magee Maja Trachonitis	Bishamon Tsukiyomi	Year 12 Year 13
3	Taisei	2020-03-02 14:42:13	PE Lesson	2020-03-03 08:00:00	2020-03-03 09:00:00	None Chosen	None Chosen	None Chosen	None Chosen	None Chosen
2	Taisei	2020-03-02 13:35:16	PE Lesson	2020-03-03 12:00:00	2020-03-03 13:30:00	South Ground	Hockey Stick (30 Available): Astro Field Shed 1	Stevie Coker Thomas Taaffe	Bishamon Izanami	Year 9
1	Taisei	2020-03-02 13:34:06	PE Lesson	2020-03-03 12:00:00	2020-03-03 13:00:00	Old Gym Leo Ground	Small Cone (50 Available): Astro Field Shed 1 Rounders Bat (10 Available): Astro Field Shed 1 Rounder Base (25 Available): Astro Field Shed 1	Maja Trachonitis	Bishamon	Year 9

Test/Success Criteria

- Filter form to allow user to choose which order they want to view the bookings
  - ‘Oldest first’ radio button (default so filled in first)
  - ‘ Newest first’ radio button
  - Submit button which will re-order the bookings in the specified way

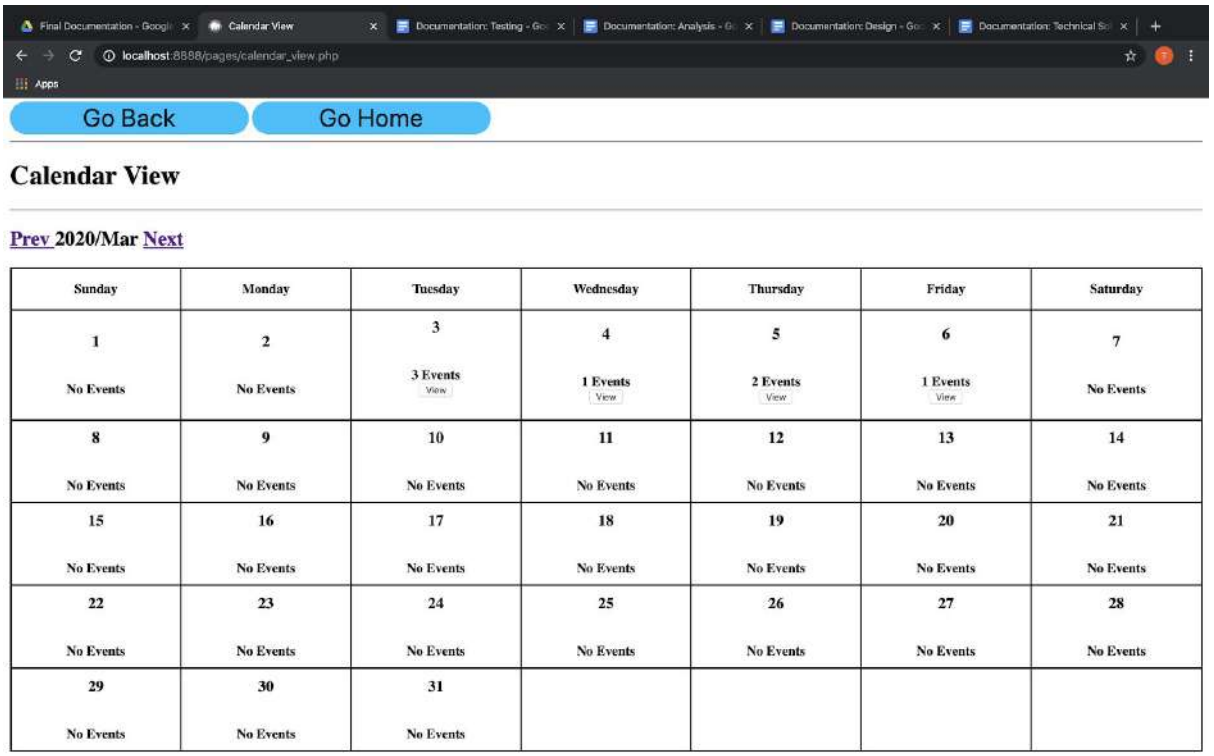


ID	Booked By	Booking Made	Event Name	Main Users	Facility	Equipment	Specific House	Specific Year Group	Start Time	End Time

- If there are no bookings in the database, an empty table will be displayed informing the user that there are no bookings

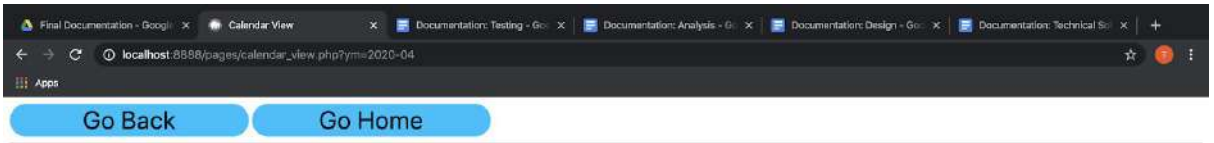
Calendar View Section

Calendar View page (calendar_view.php)



Test/Success Criteria

- Calendar
  - Has all 7 columns/headers for each day of the week
  - The first and last days of the month start and finish on the correct date in the calendar
  - The date must match its day
  - If there is still space after the final day of the month, the remaining cells must be left blank
  - All date cells must have their date/number at the top of the cell
  - If there is a booking(s) on a given date, the cell must indicate this by displaying the number of events there are, below the date
  - If there is a booking(s) on a given date, there must be a ‘View’ button to redirect the user to a separate page where those specific bookings are displayed
  - If there are no booking(s) on a given date, it must display ‘No Events’ below the date

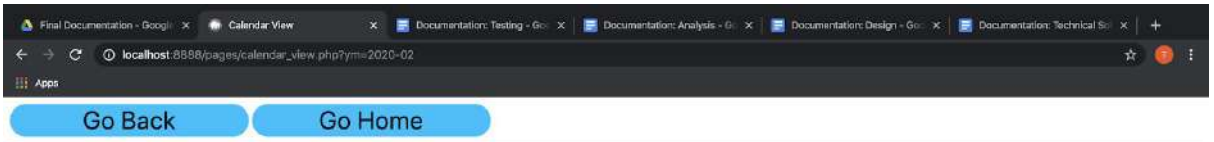


Go Back Go Home

Calendar View

Prev 2020/Apr Next

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
			1 No Events	2 No Events	3 No Events	4 No Events
5 No Events	6 No Events	7 No Events	8 No Events	9 No Events	10 No Events	11 No Events
12 No Events	13 No Events	14 No Events	15 No Events	16 No Events	17 No Events	18 No Events
19 No Events	20 No Events	21 No Events	22 No Events	23 No Events	24 No Events	25 No Events
26 No Events	27 No Events	28 No Events	29 No Events	30 No Events		



Go Back Go Home

Calendar View

Prev 2020/Feb Next

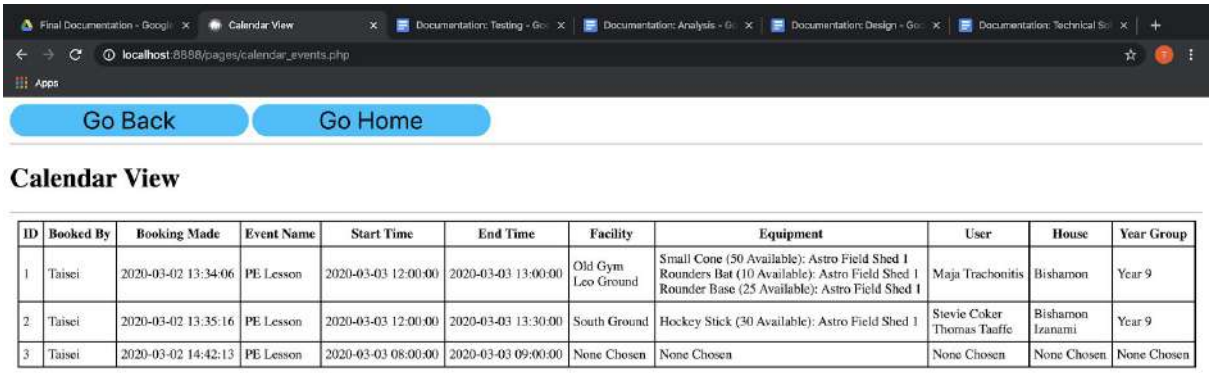
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
						1 No Events
2 No Events	3 No Events	4 No Events	5 No Events	6 No Events	7 No Events	8 No Events
9 No Events	10 No Events	11 No Events	12 No Events	13 No Events	14 No Events	15 No Events
16 No Events	17 No Events	18 No Events	19 No Events	20 No Events	21 No Events	22 No Events
23 No Events	24 No Events	25 No Events	26 No Events	27 No Events	28 No Events	29 No Events

localhost:8888/pages/calendar_view.php?ym=2020-01

Test/Success Criteria

- Navigation bar
  - Displays the current month and year
  - ‘Previous month’ button to move back one month
  - ‘Next month’ button to move forwards one month

Specific Booking page (calendar_events.php)



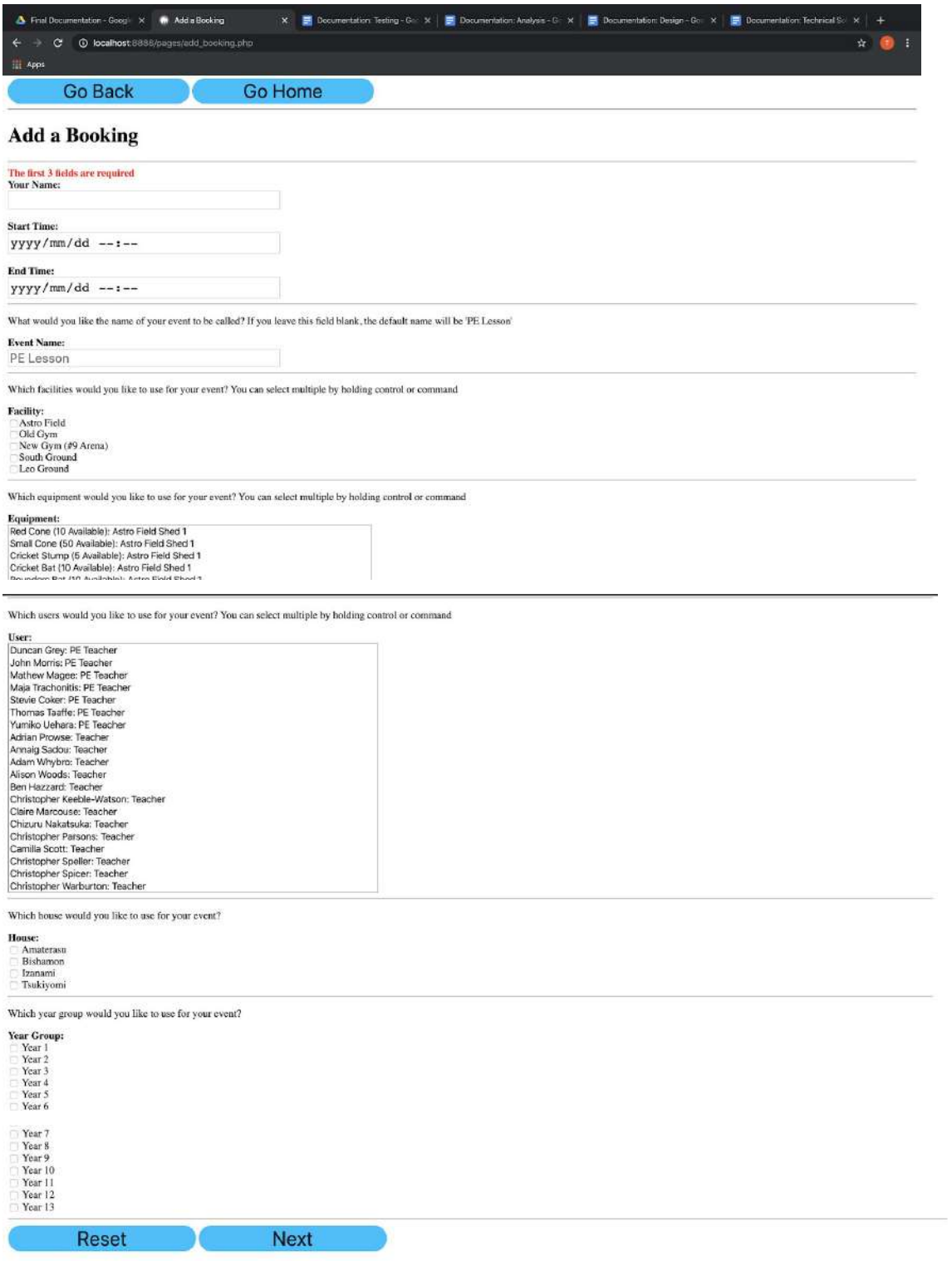
ID	Booked By	Booking Made	Event Name	Start Time	End Time	Facility	Equipment	User	House	Year Group
1	Taisei	2020-03-02 13:34:06	PE Lesson	2020-03-03 12:00:00	2020-03-03 13:00:00	Old Gym Leo Ground	Small Cone (50 Available): Astro Field Shed 1 Rounders Bat (10 Available): Astro Field Shed 1 Rounder Base (25 Available): Astro Field Shed 1	Maja Trachonitis	Bisharnon	Year 9
2	Taisei	2020-03-02 13:35:16	PE Lesson	2020-03-03 12:00:00	2020-03-03 13:30:00	South Ground	Hockey Stick (30 Available): Astro Field Shed 1	Stevie Coker Thomas Taaffe	Bisharnon Izanami	Year 9
3	Taisei	2020-03-02 14:42:13	PE Lesson	2020-03-03 08:00:00	2020-03-03 09:00:00	None Chosen	None Chosen	None Chosen	None Chosen	None Chosen

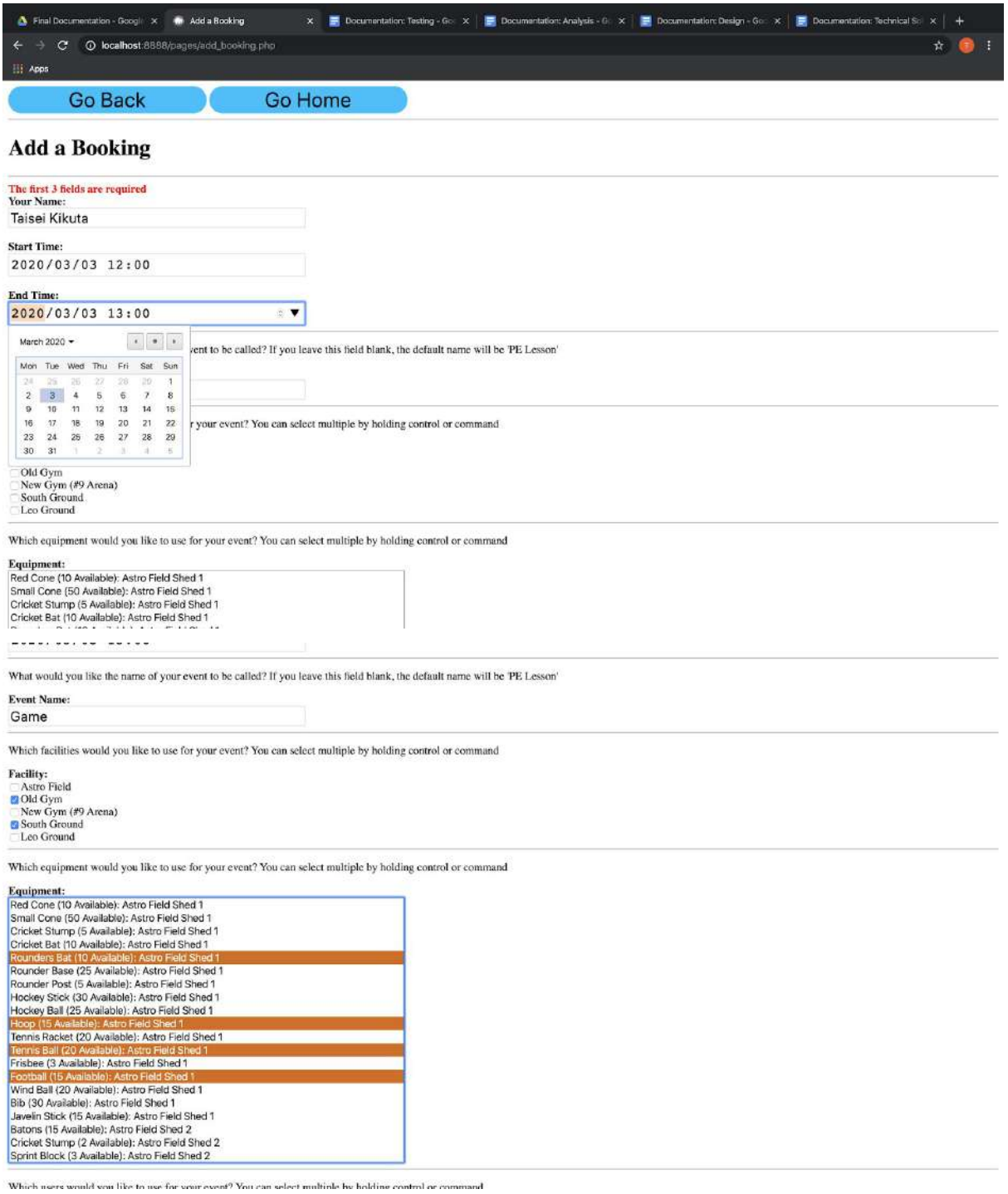
Test/Success Criteria

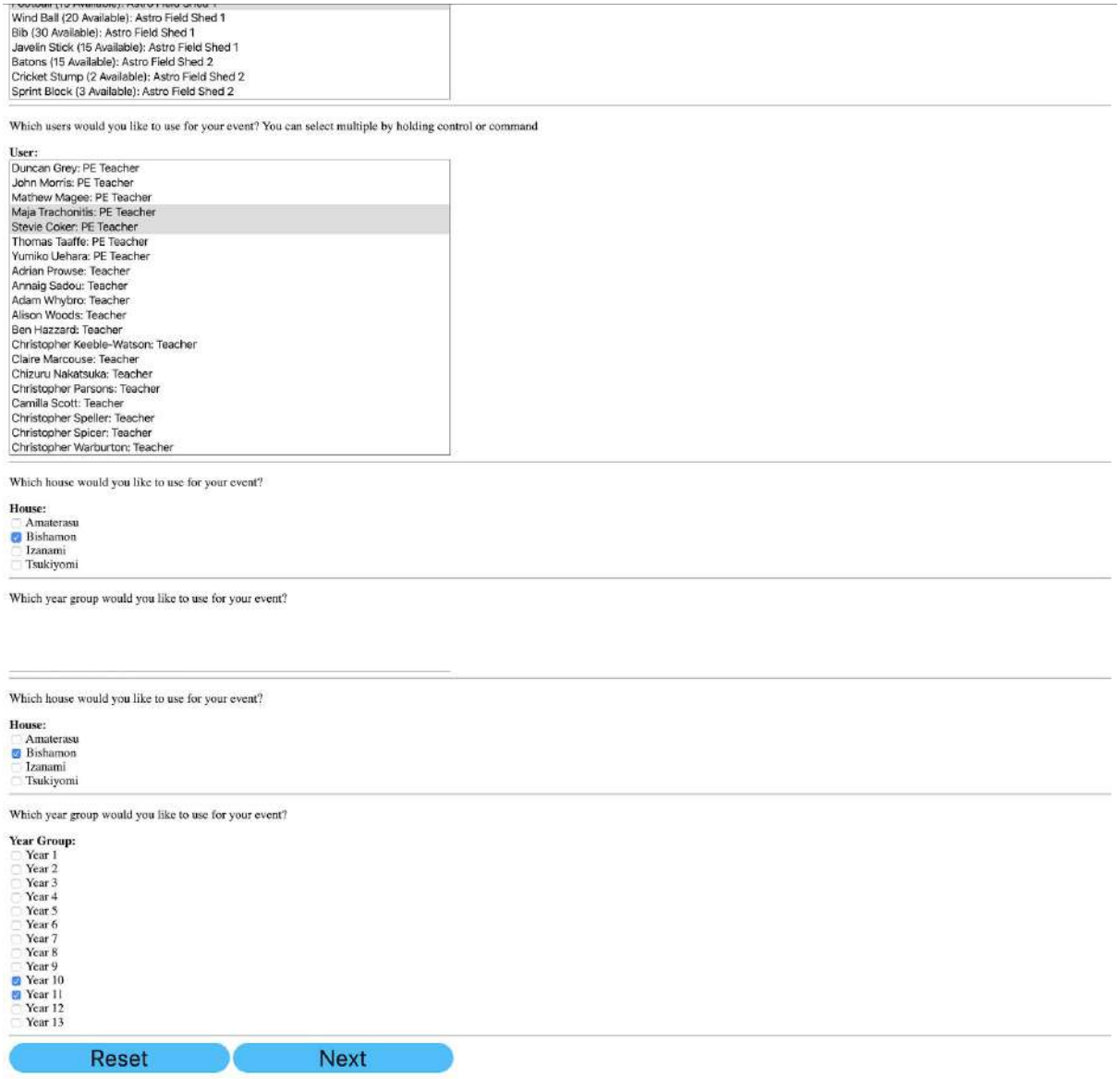
- Redirected page after the user clicks ‘View’ on a particular date
- All bookings displayed for the particular date in table format (same as in the ‘View Booking’ section)

Add Booking Section

Add Booking Page (add_booking.php)

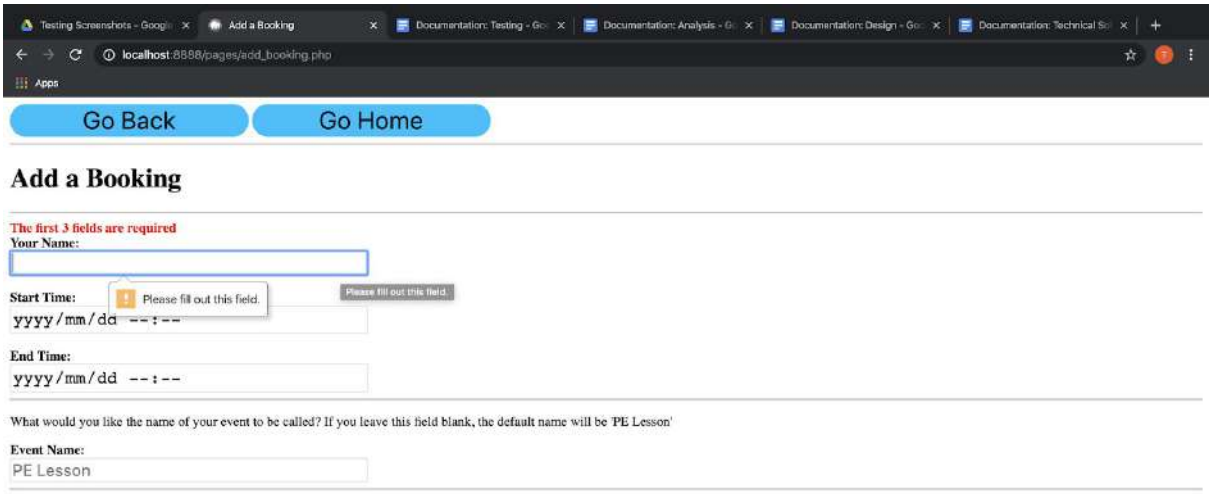



Final Documentation - Google
Add a Booking
Documentation: Testing - Go
Documentation: Analysis - Go
Documentation: Design - Go
Documentation: Technical Sp
localhost:8888/pages/add_booking.php
Apps
Go Back
Go Home
Add a Booking
The first 3 fields are required
Your Name:
Taisei Kikuta
Start Time:
2020/03/03 12:00
End Time:
2020/03/03 13:00
March 2020
Mon Tue Wed Thu Fri Sat Sun
24 25 26 27 28 29 1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31 1 2 3 4 5
vent to be called? If you leave this field blank, the default name will be 'PE Lesson'
r your event? You can select multiple by holding control or command
Old Gym
New Gym (#9 Arena)
South Ground
Leo Ground
Which equipment would you like to use for your event? You can select multiple by holding control or command
Equipment:
Red Cone (10 Available): Astro Field Shed 1
Small Cone (50 Available): Astro Field Shed 1
Cricket Stump (5 Available): Astro Field Shed 1
Cricket Bat (10 Available): Astro Field Shed 1
What would you like the name of your event to be called? If you leave this field blank, the default name will be 'PE Lesson'
Event Name:
Game
Which facilities would you like to use for your event? You can select multiple by holding control or command
Facility:
Astro Field
Old Gym
New Gym (#9 Arena)
South Ground
Leo Ground
Which equipment would you like to use for your event? You can select multiple by holding control or command
Equipment:
Red Cone (10 Available): Astro Field Shed 1
Small Cone (50 Available): Astro Field Shed 1
Cricket Stump (5 Available): Astro Field Shed 1
Cricket Bat (10 Available): Astro Field Shed 1
Rounders Bat (10 Available): Astro Field Shed 1
Rounder Base (25 Available): Astro Field Shed 1
Rounder Post (5 Available): Astro Field Shed 1
Hockey Stick (30 Available): Astro Field Shed 1
Hockey Ball (25 Available): Astro Field Shed 1
Hoop (15 Available): Astro Field Shed 1
Tennis Racket (20 Available): Astro Field Shed 1
Tennis Ball (20 Available): Astro Field Shed 1
Frisbee (3 Available): Astro Field Shed 1
Football (15 Available): Astro Field Shed 1
Wind Ball (20 Available): Astro Field Shed 1
Bib (30 Available): Astro Field Shed 1
Javelin Stick (15 Available): Astro Field Shed 1
Batons (15 Available): Astro Field Shed 2
Cricket Stump (2 Available): Astro Field Shed 2
Sprint Block (3 Available): Astro Field Shed 2



Test/Success Criteria

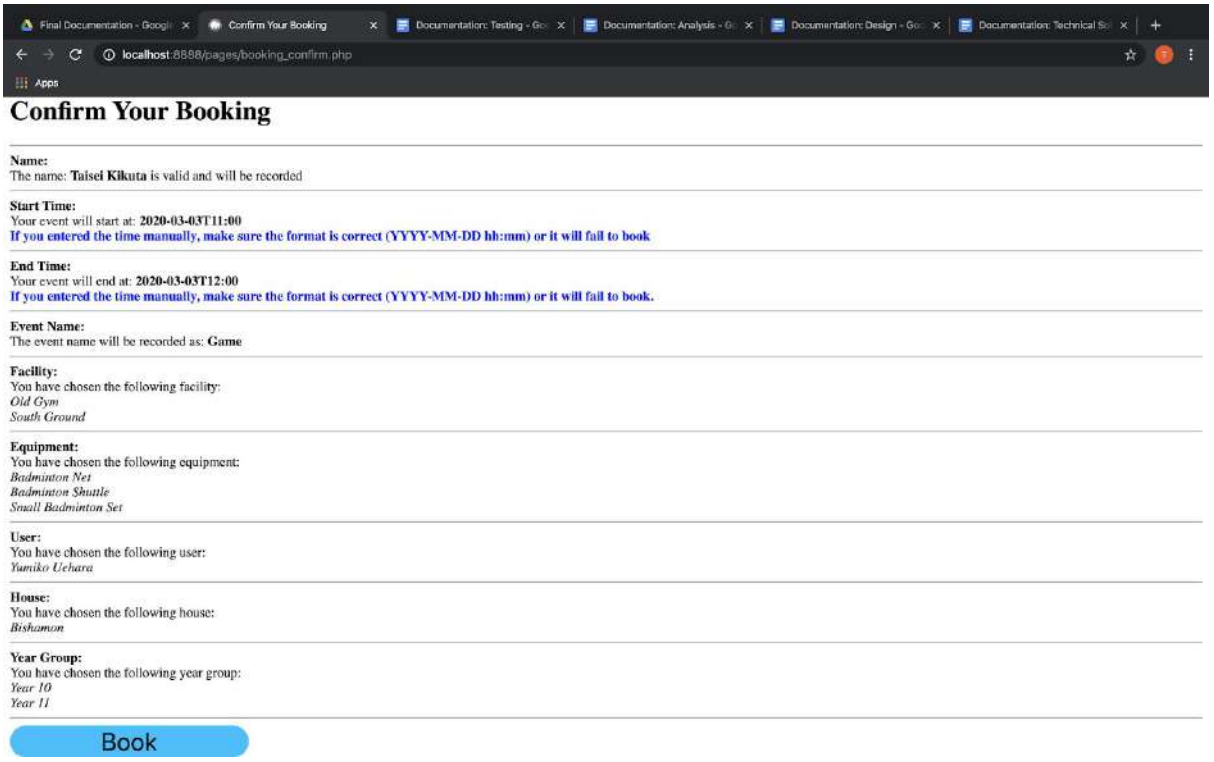
- ❖ Form that allows user to input all the information for their new booking
  - ➢ Required parts indicated with red warning text
    - ■ Text box for the name (‘bookedBy’ field)
    - ■ Date/Time input for start time (‘startTime’ field)
    - ■ Date/Time input for end time (‘endTime’ field)
    - ■ If user doesn’t fill the 3 required fields, they should be blocked from proceeding to the next page
  - ➢ Optional parts
    - ■ Text box for the event name (‘eventName’ field)
    - ■ Checkbox for facility - Allow multiple selections (booking_facility)
    - ■ Multiple select options for equipment (booking_equipment)
    - ■ Multiple select options for user (booking_user)
    - ■ Checkbox for house - Allow multiple selections (booking_house)
    - ■ Checkbox for year group - Allow multiple selections (booking_yearGroup)
    - ■ ‘Reset’ button to allow user to restart the booking process
    - ■ ‘Next’ button to proceed to the next confirmation page



Test/Success Criteria:

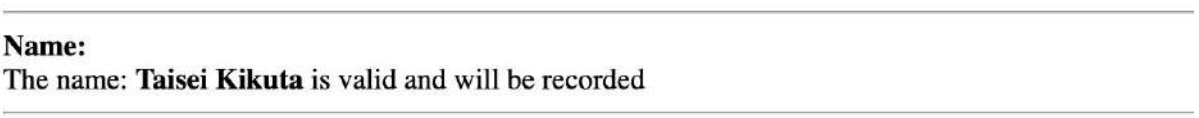
If the required fields are not filled in, they will be blocked

Confirm Booking page (booking_confirm.php)



Test/Success Criteria

- Each field of the form will have its own division on the page
  - Name section
  - Start time section
  - End time section
  - Facility section
  - Equipment section
  - User section
  - House section
  - Year group section
- ‘Book’ button at the bottom of the page
  - If there are no overlapping facilities/equipment/users and the user clicks on the ‘Book’ button, a warning message is displayed asking the user if they are sure to make the booking



Test/Success Criteria

- Name section
  - Program checks if the given name is valid (e.g. No numbers or punctuation, just letters)
  - If the name is invalid, warning message is displayed to tell user to change the name
  - If the name is valid, web page informs the user that it will be recorded as given

Start Time:
Your event will start at: **2020-03-03T12:00**
If you entered the time manually, make sure the format is correct (YYYY-MM-DD hh:mm) or it will fail to book
There is an event(s) that overlap (This is fine as long as no facilities/equipment/users are also overlapping):

ID	Booked By	Booking Made	Event Name	Start Time	End Time
1	Taisei	2020-03-03 21:12:04	PE Lesson	2020-03-03 12:00:00	2020-03-03 13:00:00

Start Time:
Your event will start at: **2020-03-03T13:00**
If you entered the time manually, make sure the format is correct (YYYY-MM-DD hh:mm) or it will fail to book

Test/Success Criteria

- Start time section
  - Program checks if the start time and end time given overlaps with any other existing bookings in the database
  - If there are any overlaps, warning message is displayed that there is a risk of double booking of facilities/equipment/users
  - If there are any overlaps, the overlapping bookings will be displayed in table form (like in the 'View Booking' section)
  - If there is no overlap, web page informs the user that it will be recorded as given

End Time:
Your event will end at: **2020-03-03T13:00**
If you entered the time manually, make sure the format is correct (YYYY-MM-DD hh:mm) or it will fail to book.
Your start time, 2020-03-03T14:00 is later than your end time, 2020-03-03T13:00 so please go back and re-write your times.

End Time:
Your event will end at: **2020-03-03T14:00**
If you entered the time manually, make sure the format is correct (YYYY-MM-DD hh:mm) or it will fail to book.

Test/Success Criteria

- End time section
  - Program checks if the start time is before the end time and the end time is after the start time (correct order)
  - If the order is invalid, warning message is displayed to tell user to change the times
  - If the order is valid, web page informs the user that it will be recorded as given

Facility:
You have chosen the following facility:
Astro Field
Old Gym
New Gym (#9 Arena)
Facility with overlapping schedules:

ID	Facility Name
1	Astro Field
1	Old Gym

Facility:
You have chosen the following facility:
Swimming Pool
Facility with overlapping schedules:

ID	Facility Name

Facility:
You have chosen the following facility:
Swimming Pool

Test/Success Criteria

- Facility section
  - Displays a list of all the facilities that the user chose in the form
  - If there is a booking overlap (covered in the start time section), web page will display a table of any overlapping facilities
  - If there are overlapping facilities, there will be a warning message telling the user that the booking can't be made, and they must choose a different facility

Equipment:
You have chosen the following equipment:
Small Cone
Cricket Stump
Cricket Bat
Equipment with overlapping schedules:

ID	Equipment Name
1	Small Cone
1	Cricket Stump
1	Cricket Bat

Equipment:
You have chosen the following equipment:
Frisbee
Football
Wind Ball
Bib
Equipment with overlapping schedules:

ID	Equipment Name

Equipment:
You have chosen the following equipment:
Frisbee
Football
Wind Ball
Bib

Test/Success Criteria

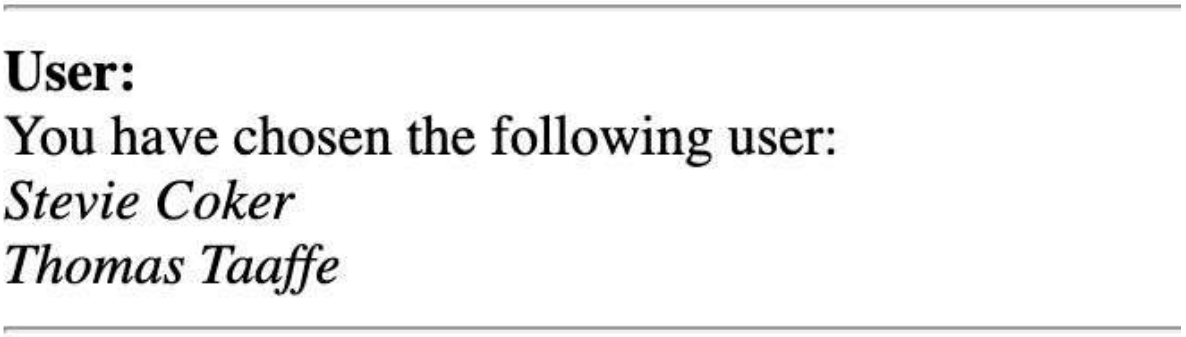
- Equipment section
  - Displays a list of all the equipment that the user chose in the form
  - If there is a booking overlap (covered in the start time section), web page will display a table of any overlapping equipment
  - If there are overlapping equipment, there will be a warning message telling the user that the booking can't be made, and they must choose a different equipment

User:
You have chosen the following user:
John Morris
Mathew Magee
User with overlapping schedules:

ID	User Name
1	John Morris
1	Mathew Magee

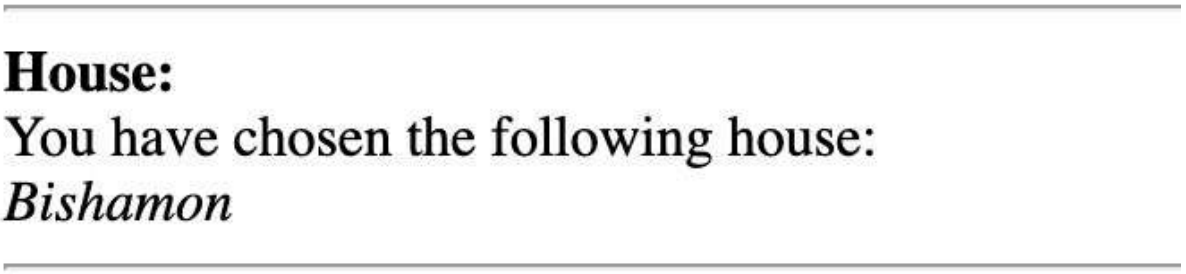
User:
You have chosen the following user:
Stevie Coker
Thomas Taaffe
User with overlapping schedules:

ID	User Name

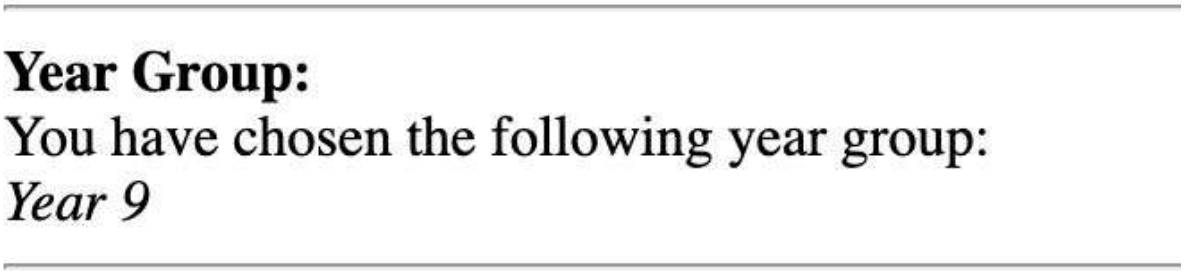


Test/Success Criteria

- User section
  - Displays a list of all the user that the user chose in the form
  - If there is a booking overlap (covered in the start time section), web page will display a table of any overlapping users
  - If there are double booked users, there will be a warning message telling the user that the booking can't be made, and they must choose a different user

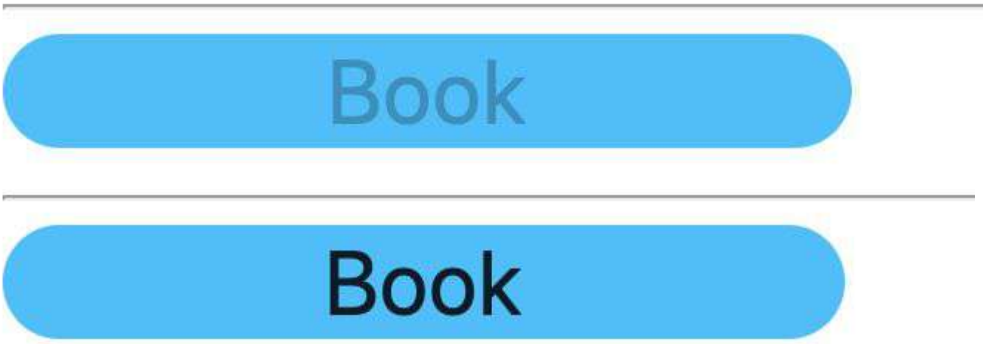


- House section
  - Displays a list of all the houses that the user chose in the form



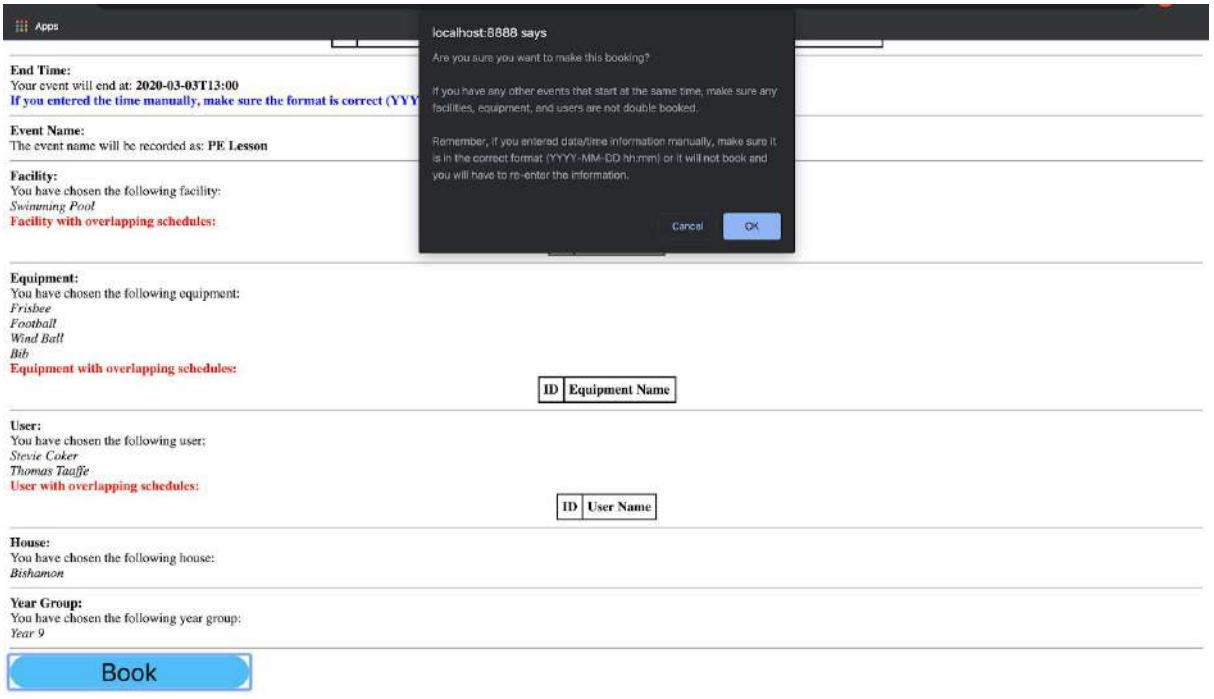
Test/Success Criteria

- Year group section
  - Displays a list of all the year groups that the user chose in the form



Test/Success Criteria

- ‘Book’ button at the bottom of the page
  - If there are any overlapping facilities/equipment/users, the button will be blocked so the user cannot press it



Test/Success Criteria

- If there are no overlapping facilities/equipment/users and the user clicks on the ‘Book’ button, a warning message is displayed asking the user if they are sure to make the booking

Booking Added page (booking_added.php)

Go Home

Booking Add

Booking has been added successfully

Your New Booking

ID	Booked By	Booking Made	Event Name	Start Time	End Time	Facility	Equipment	User	House	Year Group
3	Taisei Kikuta	2020-03-03 21:34:17	PE Lesson	2020-03-03 12:00:00	2020-03-03 13:00:00	Swimming Pool	Frisbee (3 Available): Astro Field Shed 1 Football (15 Available): Astro Field Shed 1 Wind Ball (20 Available): Astro Field Shed 1 Bib (30 Available): Astro Field Shed 1	Stevie Coker Thomas Taaffe	Bishamon	Year 9

Updated Booking List

ID	Booked By	Booking Made	Event Name	Start Time	End Time	Facility	Equipment	User	House	Year Group
1	Taisei	2020-03-03 21:12:04	PE Lesson	2020-03-03 12:00:00	2020-03-03 13:00:00	Astro Field Old Gym	Red Cone (10 Available): Astro Field Shed 1 Small Cone (50 Available): Astro Field Shed 1 Cricket Stump (5 Available): Astro Field Shed 1 Cricket Bat (10 Available): Astro Field Shed 1	John Morris Mathew Magee	Bishamon	Year 10 Year 11
2	Taisei	2020-03-03 21:12:30	PE Lesson	2020-03-03 11:00:00	2020-03-03 12:00:00	Old Gym New Gym (#9 Arena)	Cricket Bat (10 Available): Astro Field Shed 1 Rounders Bat (10 Available): Astro Field Shed 1 Rounder Base (25 Available): Astro Field Shed 1	Mathew Magee Maja Trachonitis Stevie Coker	Bishamon	Year 10
3	Taisei Kikuta	2020-03-03 21:34:17	PE Lesson	2020-03-03 12:00:00	2020-03-03 13:00:00	Swimming Pool	Frisbee (3 Available): Astro Field Shed 1 Football (15 Available): Astro Field Shed 1 Wind Ball (20 Available): Astro Field Shed 1 Bib (30 Available): Astro Field Shed 1	Stevie Coker Thomas Taaffe	Bishamon	Year 9

Test Success Criteria

- If the booking has been added to the database, display a message informing the user that it was successfully added
- Display the new booking that was just added to the database in table form
- Display the updated list of bookings

Delete Booking Section

Delete Booking Page (delete_booking.php)

Go Back | Go Home

Delete Your Bookings

- (•) Oldest First
- ( ) Newest First

Filter

Clear

Delete? Reset	ID	Booked By	Booking Made	Event Name	Start Time	End Time	Facility	Equipment	User	House	Year Group
☐	1	Taisei	2020-03-03 21:12:04	PE Lesson	2020-03-03 12:00:00	2020-03-03 13:00:00	Astro Field Old Gym	Red Cone (10 Available): Astro Field Shed 1 Small Cone (50 Available): Astro Field Shed 1 Cricket Stump (5 Available): Astro Field Shed 1 Cricket Bat (10 Available): Astro Field Shed 1	John Morris Mathew Magee	Bishamon	Year 10 Year 11
☐	2	Taisei	2020-03-03 21:12:30	PE Lesson	2020-03-03 11:00:00	2020-03-03 12:00:00	Old Gym New Gym (#9 Arena)	Cricket Bat (10 Available): Astro Field Shed 1 Rounders Bat (10 Available): Astro Field Shed 1 Rounder Base (25 Available): Astro Field Shed 1	Mathew Magee Maja Trachonitis Stevie Coker	Bishamon	Year 10
☐	3	Taisei Kikuta	2020-03-03 21:34:17	PE Lesson	2020-03-03 12:00:00	2020-03-03 13:00:00	Swimming Pool	Frisbee (3 Available): Astro Field Shed 1 Football (15 Available): Astro Field Shed 1 Wind Ball (20 Available): Astro Field Shed 1 Bib (30 Available): Astro Field Shed 1	Stevie Coker Thomas Taaffe	Bishamon	Year 9

Go Back | Go Home

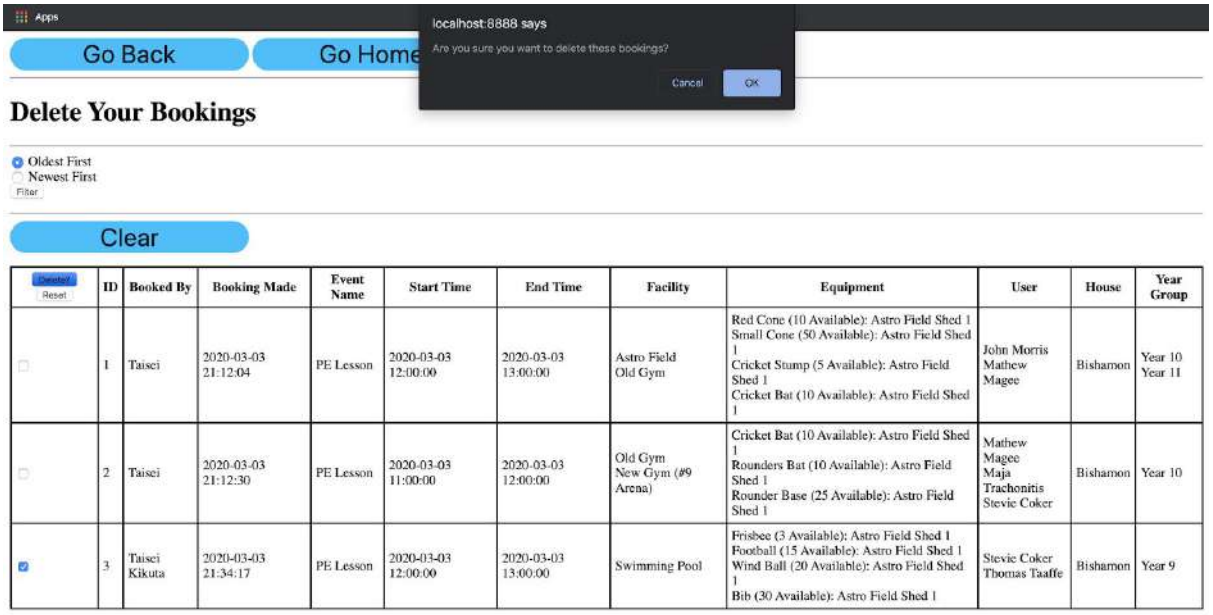
Delete Your Bookings

- ( ) Oldest First
- (•) Newest First

Filter

Clear

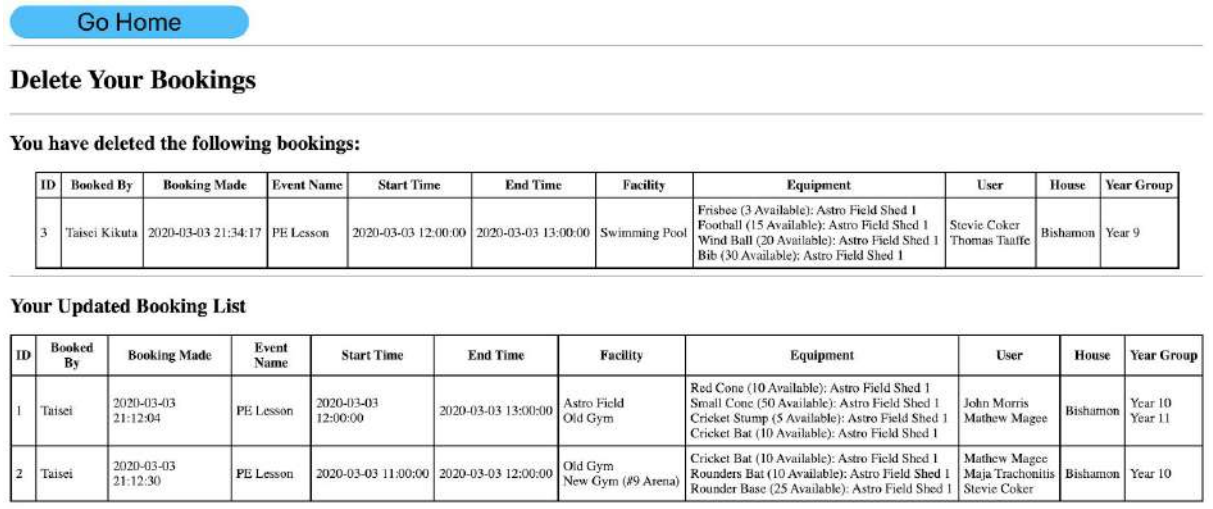
Delete? Reset	ID	Booked By	Booking Made	Event Name	Start Time	End Time	Facility	Equipment	User	House	Year Group
☐	3	Taisei Kikuta	2020-03-03 21:34:17	PE Lesson	2020-03-03 12:00:00	2020-03-03 13:00:00	Swimming Pool	Frisbee (3 Available): Astro Field Shed 1 Football (15 Available): Astro Field Shed 1 Wind Ball (20 Available): Astro Field Shed 1 Bib (30 Available): Astro Field Shed 1	Stevie Coker Thomas Taaffe	Bishamon	Year 9
☐	2	Taisei	2020-03-03 21:12:30	PE Lesson	2020-03-03 11:00:00	2020-03-03 12:00:00	Old Gym New Gym (#9 Arena)	Cricket Bat (10 Available): Astro Field Shed 1 Rounders Bat (10 Available): Astro Field Shed 1 Rounder Base (25 Available): Astro Field Shed 1	Mathew Magee Maja Trachonitis Stevie Coker	Bishamon	Year 10
☐	1	Taisei	2020-03-03 21:12:04	PE Lesson	2020-03-03 12:00:00	2020-03-03 13:00:00	Astro Field Old Gym	Red Cone (10 Available): Astro Field Shed 1 Small Cone (50 Available): Astro Field Shed 1 Cricket Stump (5 Available): Astro Field Shed 1 Cricket Bat (10 Available): Astro Field Shed 1	John Morris Mathew Magee	Bishamon	Year 10 Year 11



Test/Success Criteria

- Display all the bookings in table form (same as in the ‘View Booking’ section)
- Filter form to allow user to view the bookings in a specified order (same as in the ‘View Booking’ section)
- Additional column in the table to act as a form
  - Header for column must have a ‘Delete’ button to allow the user to press after selecting the bookings they want to remove, plus a ‘Reset’ button to allow the user to restart
  - Checkbox in each of the column to allow the user to check/fill in the bookings they want to delete
- ‘Clear’ button to allow the user to click and delete all the bookings
- When pressing ‘Delete’ or ‘Clear’, a warning message is displayed to ask the user if they are sure

Booking Deleted page (delete_bookings.view.php)



Go Home

Delete Your Bookings

You have deleted the following bookings:

ID	Booked By	Booking Made	Event Name	Start Time	End Time	Facility	Equipment	User	House	Year Group
1	Taisei	2020-03-03 21:12:04	PE Lesson	2020-03-03 12:00:00	2020-03-03 13:00:00	Astro Field Old Gym	Red Cone (10 Available): Astro Field Shed 1 Small Cone (50 Available): Astro Field Shed 1 Cricket Stump (5 Available): Astro Field Shed 1 Cricket Bat (10 Available): Astro Field Shed 1	John Morris Mathew Magee	Bishamon	Year 10 Year 11
2	Taisei	2020-03-03 21:12:30	PE Lesson	2020-03-03 11:00:00	2020-03-03 12:00:00	Old Gym New Gym (#9 Arena)	Cricket Bat (10 Available): Astro Field Shed 1 Rounders Bat (10 Available): Astro Field Shed 1 Rounder Base (25 Available): Astro Field Shed 1	Mathew Magee Maja Trachonitis Stevie Coker	Bishamon	Year 10

Your Updated Booking List

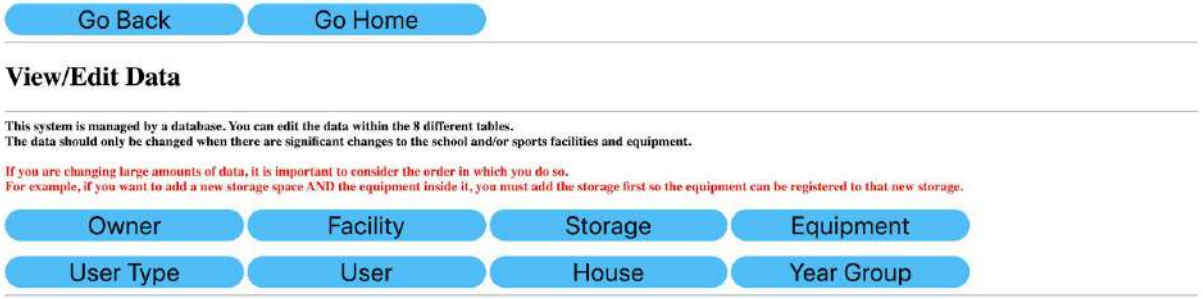
No Bookings

ID	Booked By	Booking Made	Event Name	Main Users	Facility	Equipment	Specific House	Specific Year Group	Start Time	End Time

- If the user chose to delete specific bookings, display these bookings and inform that they were deleted
- If the user chose to delete all the bookings (clear), display all the bookings and inform that they were deleted
- Display the updated booking table
  - If all the bookings were deleted, display an empty table informing the user that there are no bookings

View/Edit Database Section

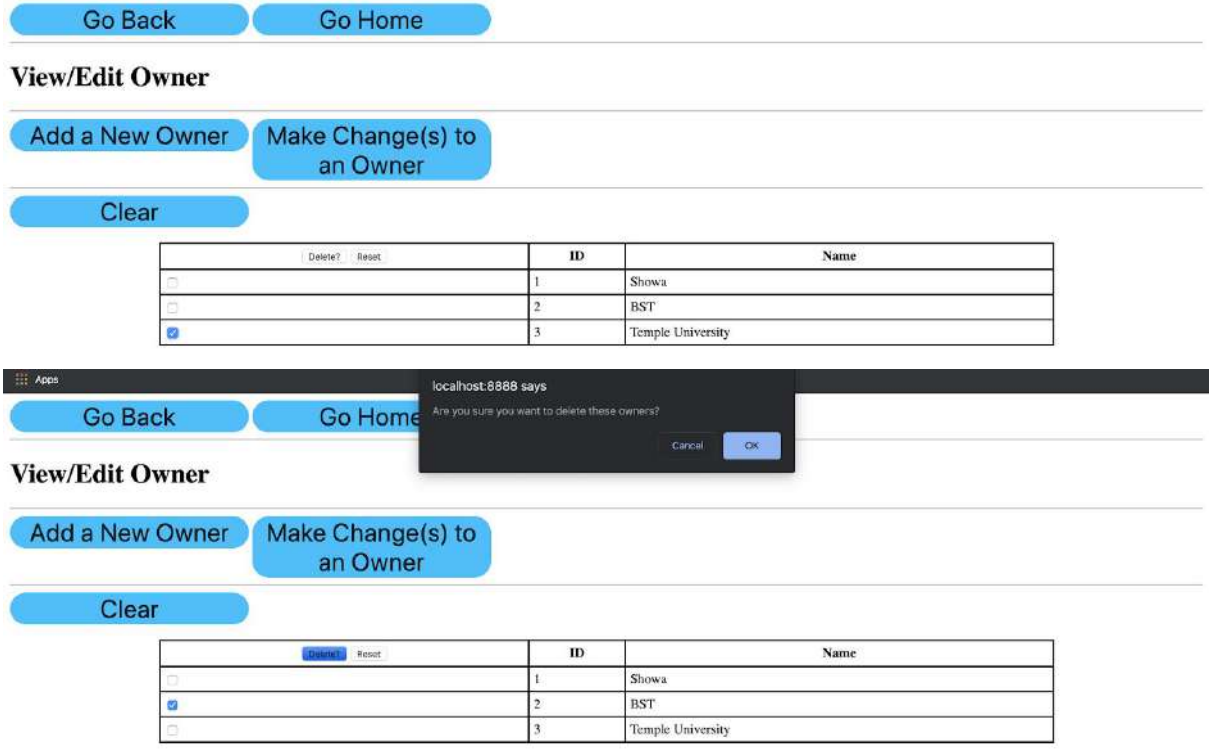
View/Edit Data page (edit_data.php)



Test/Success Criteria

- Large buttons that redirect the user to the specific pages/sections
  - ‘View/Edit owner’ button
  - ‘View/Edit facility’ button
  - ‘View/Edit storage’ button
  - ‘View/Edit equipment’ button
  - ‘View/Edit user type’ button
  - ‘View/Edit user’ button
  - ‘View/Edit house’ button
  - ‘View/Edit year group’ button

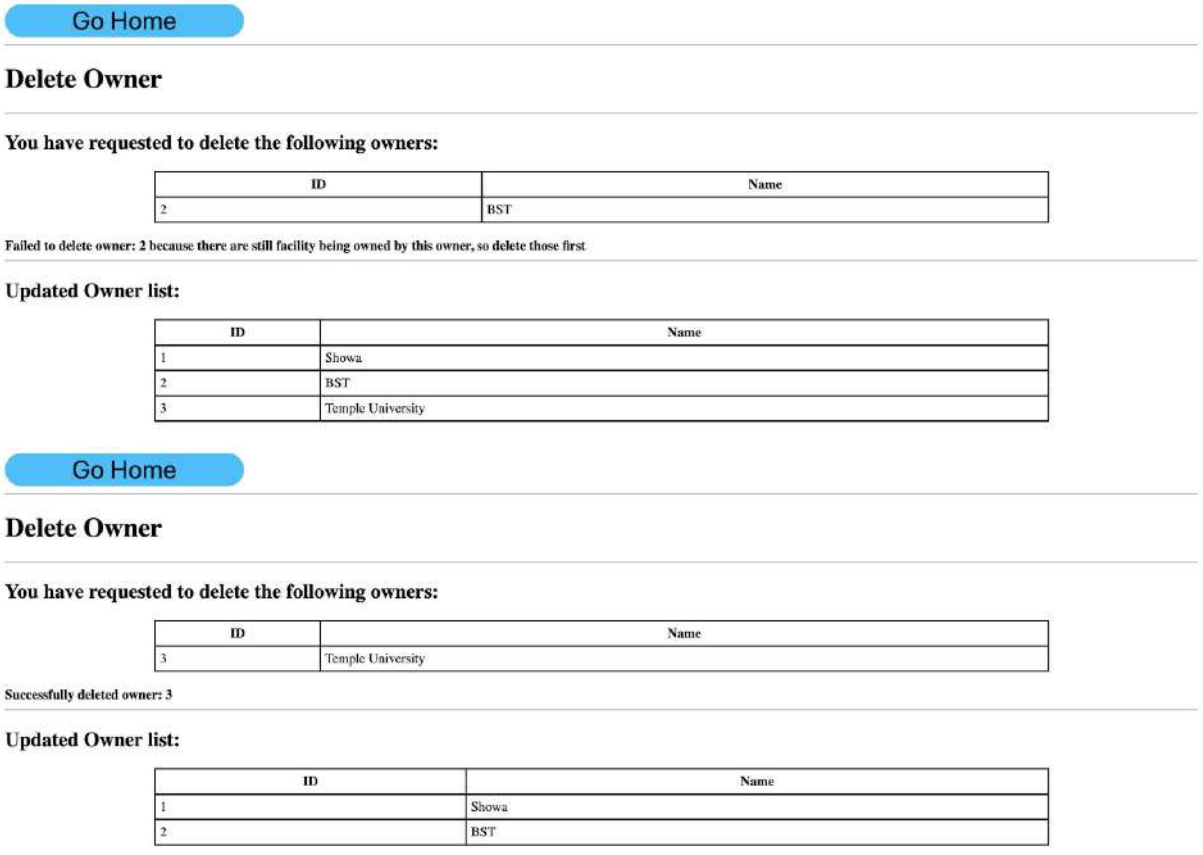
View/Edit Owner page (edit_owner.php)



Test/Success Criteria

- ❖ All owners displayed in a table format
  - ➢ Column for 'Owner ID'
  - ➢ Column for 'Owner Name'
  - ➢ Column for deleting owners (like 'Delete Booking' section)
    - ■ Each row has a checkbox to allow the user to check/fill in the box
    - ■ The column header has a 'Delete' button to allow user to delete select number of owners, and a reset button to allow user to restart
  - ➢ Clear button to allow user to delete all owners in the database
  - ➢ If the owner table is empty and there are no owners in the database, display an empty table informing the user that there are no owners
- ❖ 'Add New Owner' button to redirect user to Add Owner page
- ❖ 'Make changes to Owner' button to redirect user to edit owner page

Owner Deleted page (delete_owners.view.php)



Go Home

Delete Owner

You have requested to delete the following owners:

ID	Name
2	BST

Failed to delete owner: 2 because there are still facility being owned by this owner, so delete those first

Updated Owner list:

ID	Name
1	Showa
2	BST
3	Temple University

Go Home

Delete Owner

You have requested to delete the following owners:

ID	Name
3	Temple University

Successfully deleted owner: 3

Updated Owner list:

ID	Name
1	Showa
2	BST

Test/Success Criteria

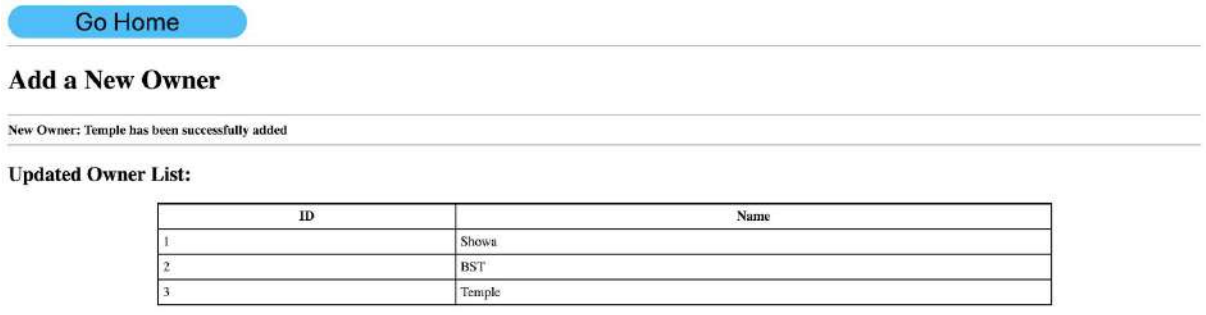
- ❖ If the owner was deleted from the database, display to inform the user that it was successfully deleted
  - ➢ If selected owners were deleted, display these owners
  - ➢ If all owners were deleted, display all the owners
- ❖ If the owner couldn't be removed from the database, inform the user that it couldn't be deleted
- ❖ Display the updated owner table
  - ➢ If all owners were deleted, display an empty table informing the user that there are no owners

Add Owner page (add_owner.php)

Test/Success Criteria

- Form to allow the user to input the required information for a new owner
  - Text box to enter the owner's name (required)
    - If the text box is left blank and the user tries to add the owner, they will be blocked
  - 'Add' button to add the owner and proceed to the next page, and a rest button to restart if needed
  - When pressing the add button, a warning message should be displayed to ask the user if they are sure they want to add the owner

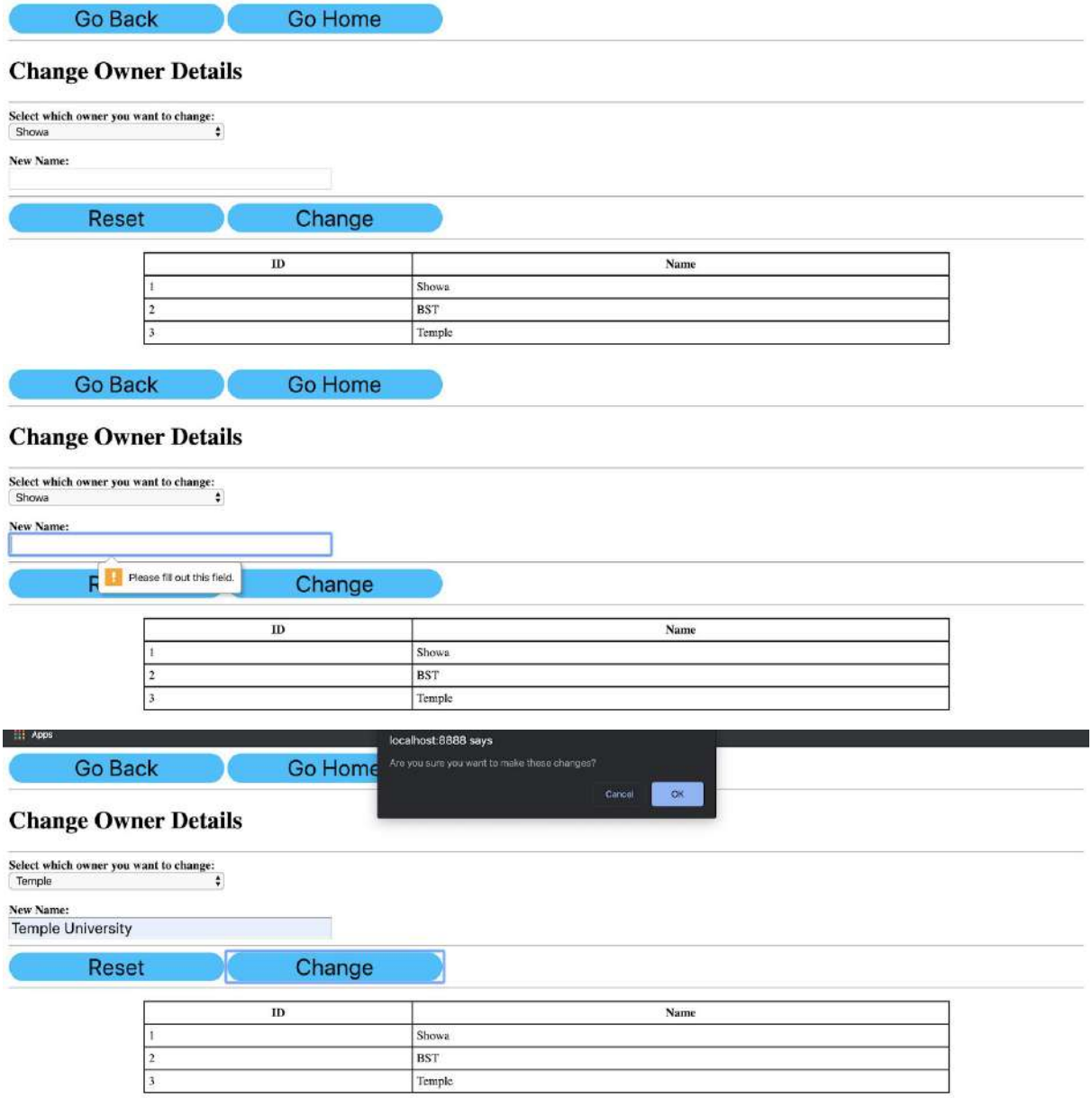
Owner Added page (add_owner.view.php)



Test/Success Criteria

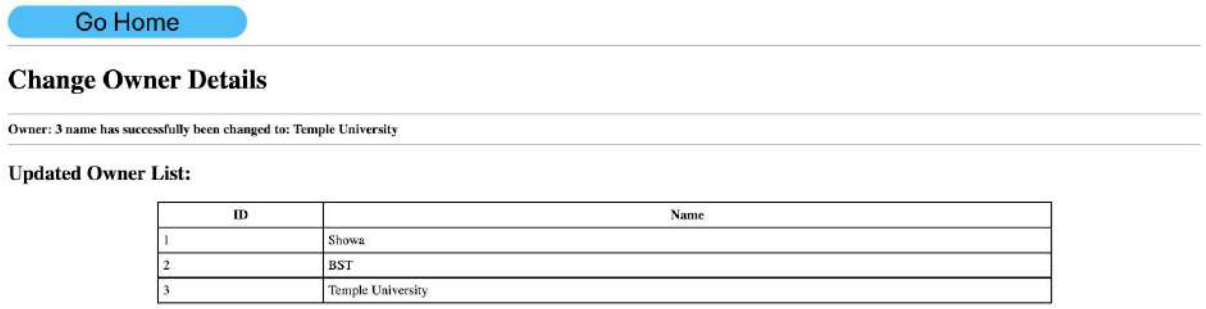
- If the owner was added to the owner table, display a message informing the user that it was successfully added
- Display the updated owner table

Change Owner Details page (change_owner.php)



- Form to allow user to select an owner and change their details
  - Drop down menu with the owners as options (already selected with a default value)
  - Text box to input the owners new name (required)
    - If the user doesn't fill in the text box and tries to proceed, they will be blocked
  - 'Change' button to proceed and a 'Reset' button to restart if needed

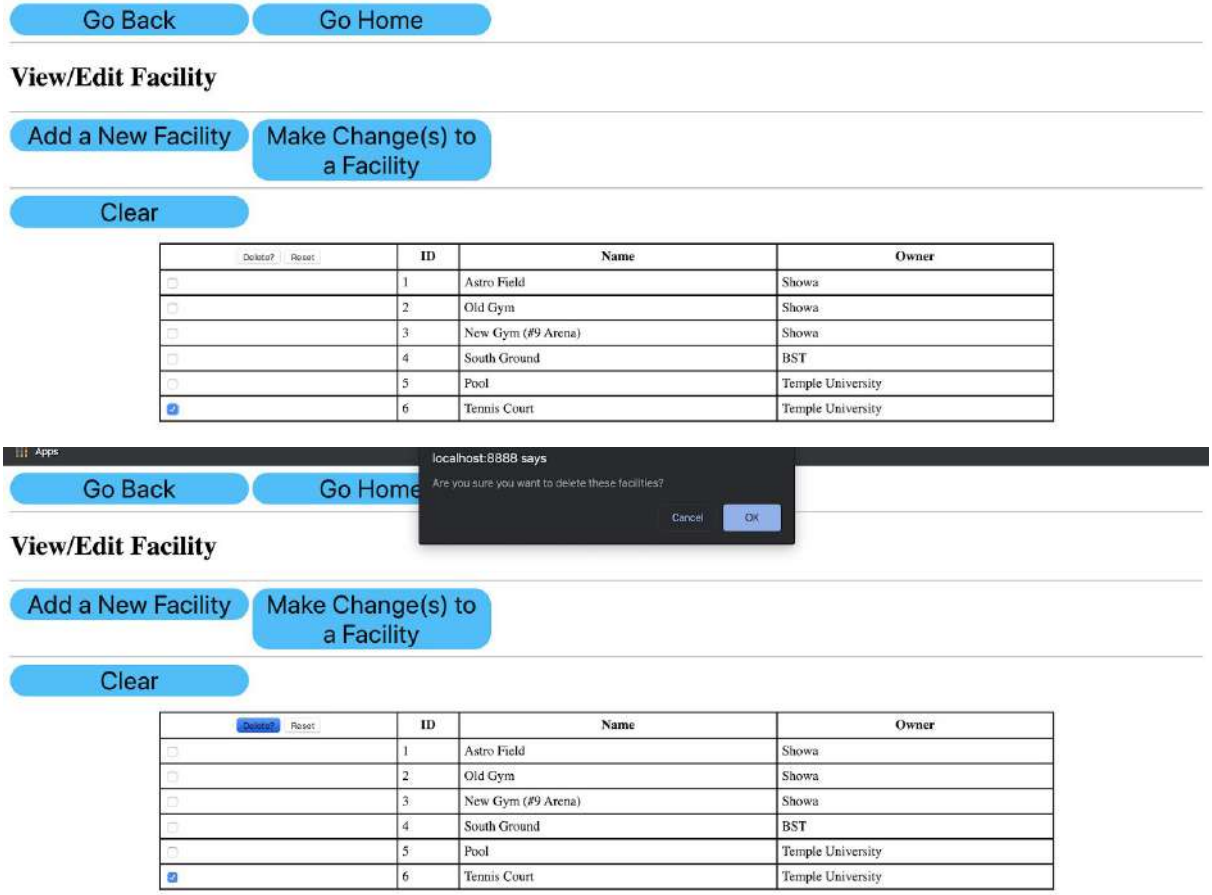
Owner Details Changed page (change_owner.view.php)



Test/Success Criteria

- ❖ If the owner's details have been changed, display a message to inform the user that it has been successfully changed
- ❖ Display a table of the updated owners

Edit Facility page (edit_facility.php)



Test/Success Criteria

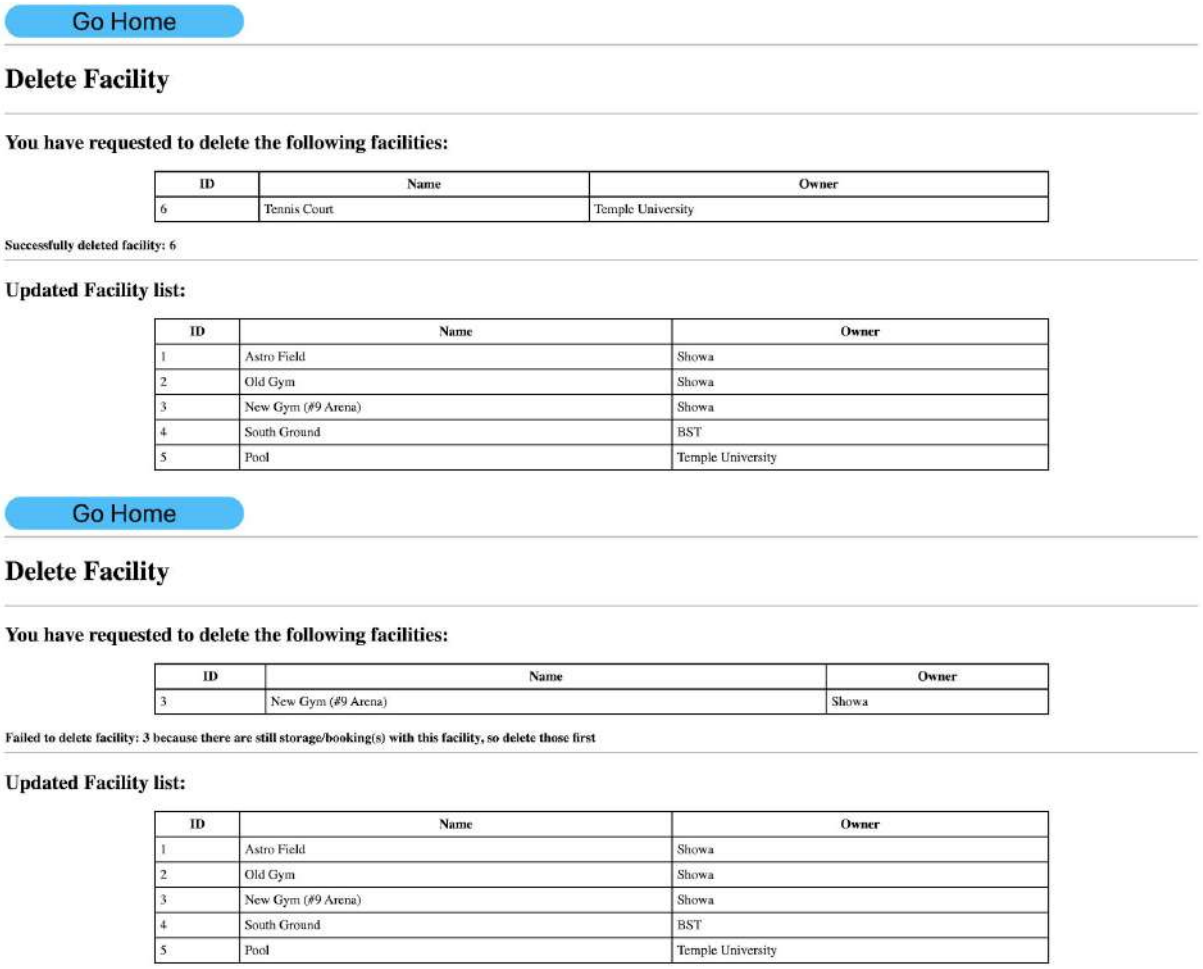
- ❖ All facilities displayed in a table format
  - ➢ Column for 'Facility ID'
  - ➢ Column for 'Facility Name'
  - ➢ Column for 'Owner Name'
  - ➢ Column for deleting facilities (like 'Delete Booking' section)
    - ■ Each row has a checkbox to allow the user to check/fill in the box
    - ■ The column header has a 'Delete' button to allow user to delete select number of facilities, and a reset button to allow user to restart if needed

- ➢ Clear button to allow user to delete all facilities in the facility table
- ➢ If the facility table is empty and there are no facilities in the database, display an empty table informing the user that there are no facilities

❖ ‘Add New Facility’ button to redirect user to Add Facility page

❖ ‘Make changes to Facility’ button to redirect user to edit facility page

Facilities Deleted page (delete_facilities.view.php)



Go Home

Delete Facility

You have requested to delete the following facilities:

ID	Name	Owner
6	Tennis Court	Temple University

Successfully deleted facility: 6

Updated Facility list:

ID	Name	Owner
1	Astro Field	Showa
2	Old Gym	Showa
3	New Gym (#9 Arena)	Showa
4	South Ground	BST
5	Pool	Temple University

Go Home

Delete Facility

You have requested to delete the following facilities:

ID	Name	Owner
3	New Gym (#9 Arena)	Showa

Failed to delete facility: 3 because there are still storage/booking(s) with this facility, so delete those first

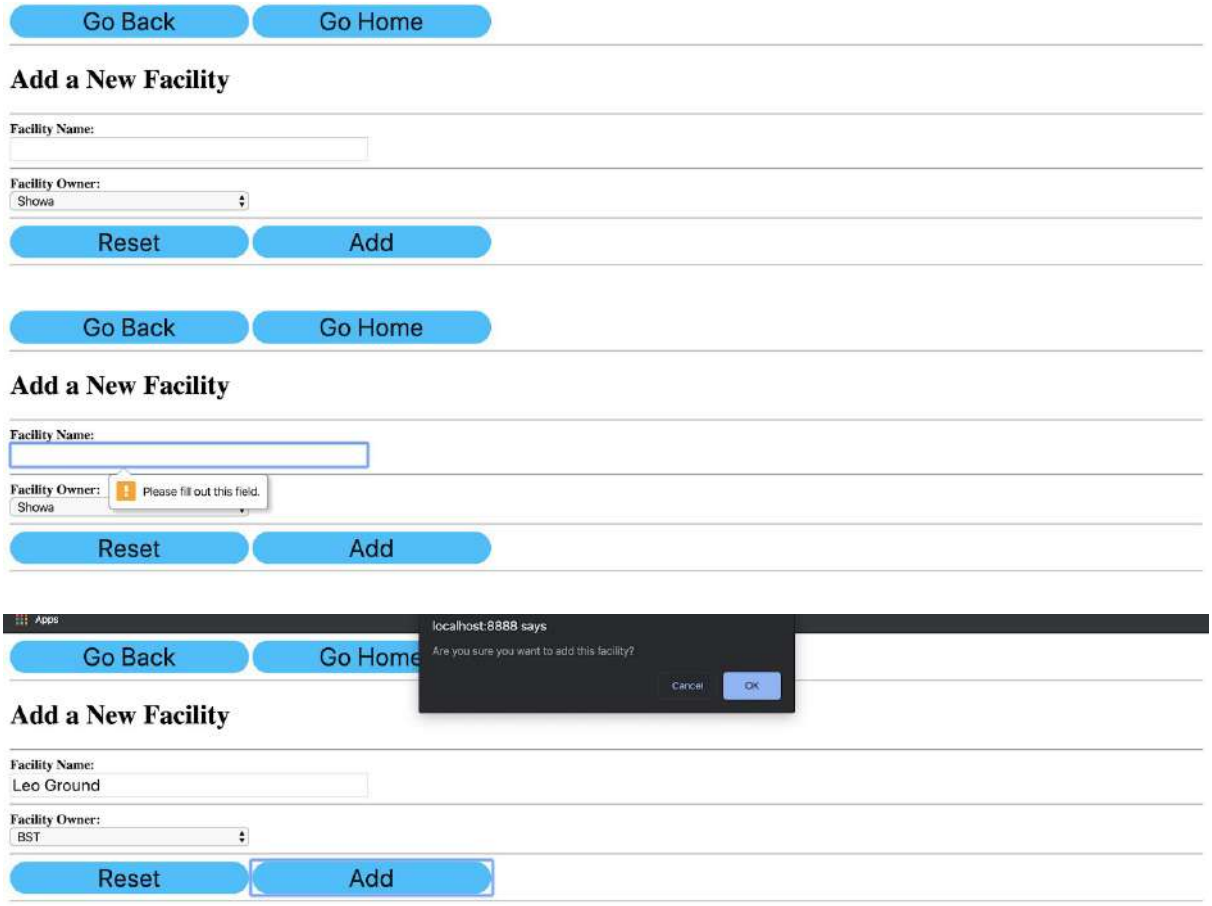
Updated Facility list:

ID	Name	Owner
1	Astro Field	Showa
2	Old Gym	Showa
3	New Gym (#9 Arena)	Showa
4	South Ground	BST
5	Pool	Temple University

Test/Success Criteria

- ❖ If the facility was deleted from the database, display to inform the user that it was successfully deleted
  - ➢ If selected facilities were deleted, display these facilities
  - ➢ If all facilities were deleted, display all the facilities
- ❖ If the facilities couldn’t be removed from the database, inform the user that it couldn’t be deleted
- ❖ Display the updated facility table
  - ➢ If all facilities were deleted, display an empty table informing the user that there are no facilities

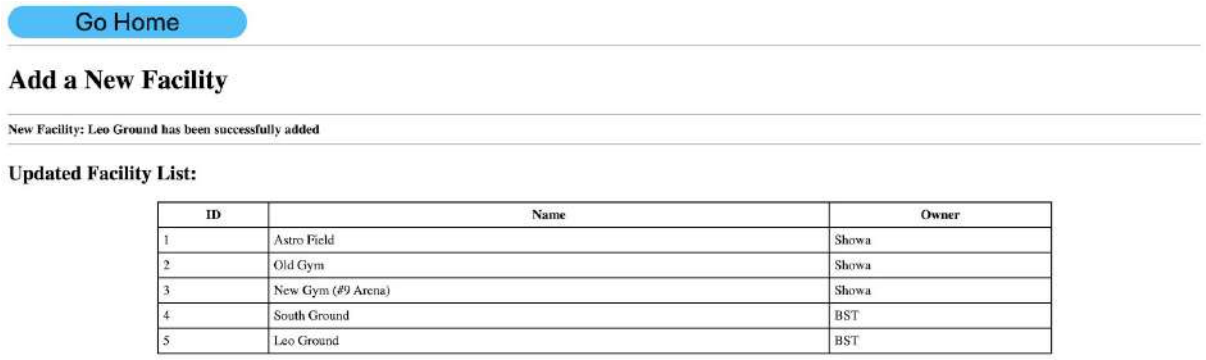
Add Facility page (add_facility.php)



Test/Success Criteria

- ❖ Form to allow the user to input the required information for a new facility
  - ➢ Text box to enter the facility's name (required)
    - ■ If the text box is left blank and the user tries to add the facility, they will be blocked
  - ➢ Drop down menu to select an owner for the facility (already selected with a default value)
  - ➢ 'Add' button to add the facility and proceed to the next page, and a rest button to restart
  - ➢ When pressing the add button, a warning message should be displayed to ask the user if they are sure they want to add the facility

Facility Added page (add_facility.view.php)



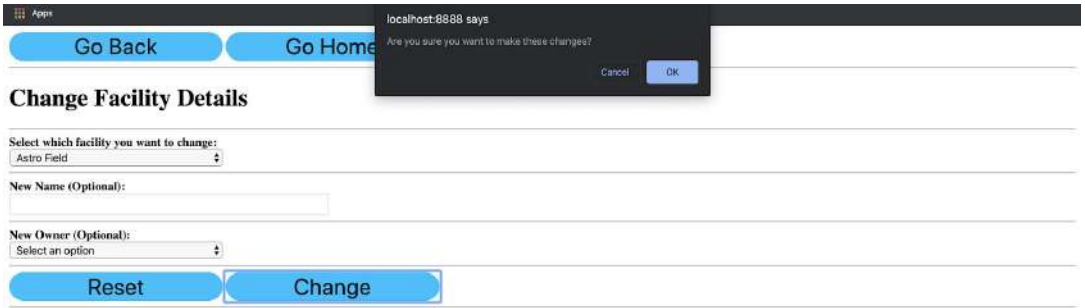
ID	Name	Owner
1	Astro Field	Showa
2	Old Gym	Showa
3	New Gym (#9 Arena)	Showa
4	South Ground	BST
5	Leo Ground	BST

Test/Success Criteria

- If the facility was added to the database, display a message informing the user that it was successfully added
- Display the updated facility table

Change Facility Details page (change_facility.php)

ID	Name	Owner
1	Astro Field	Showa
2	Old Gym	Showa
3	New Gym (#9 Arena)	Showa
4	South Ground	BST
5	Leo Ground	BST



ID	Name	Owner
1	Astro Field	Showa
2	Old Gym	Showa
3	New Gym (#9 Arena)	Showa
4	South Ground	BST
5	Leo Ground	BST

Test/Success Criteria

- Form to allow user to select an facility and change their details
  - Drop down menu with the facilities as options (already selected with a default value)
  - Drop down menu with the new owners as options (optional)
  - Text box to input the owners new name (optional)
  - ‘Change’ button to proceed and a ‘Reset’ button to restart if needed

Facility Details Changed page (change_facility.view.php)

Go Home

Change Facility Details

You made no changes to the facility

Updated Facility List:

ID	Name	Owner
1	Astro Field	Showa
2	Old Gym	Showa
3	New Gym (#9 Arena)	Showa
4	South Ground	BST
5	Leo Ground	BST

Go Home

Change Facility Details

Facility: 1 name has successfully been changed to: Astro Turf

Updated Facility List:

ID	Name	Owner
1	Astro Turf	Showa
2	Old Gym	Showa
3	New Gym (#9 Arena)	Showa
4	South Ground	BST
5	Leo Ground	BST

Go Home

Change Facility Details

Facility: 1 owner has successfully been changed to owner: 2

Updated Facility List:

ID	Name	Owner
1	Astro Turf	BST
2	Old Gym	Showa
3	New Gym (#9 Arena)	Showa
4	South Ground	BST
5	Leo Ground	BST

Go Home

Change Facility Details

Facility: 1 name has successfully been changed to: Astro Field
Facility: 1 owner has successfully been changed to owner: 1

Updated Facility List:

ID	Name	Owner
1	Astro Field	Showa
2	Old Gym	Showa
3	New Gym (#9 Arena)	Showa
4	South Ground	BST
5	Leo Ground	BST

Test/Success Criteria

New Name	New Owner	Expected Output
0	0	“No changes”
0	1	“Owner changed”
1	0	“Name changed”
1	1	“Name changed “Owner changed”

- ❖ If the facility’s details have been changed, display a message to inform the user that it has been successfully changed
  - ➢ Display each of the changes made (facility name and/or facility’s owner)
- ❖ If the facility’s details couldn’t be changed, display a message to inform the user that the changes couldn’t be made
- ❖ Display a table of the updated facilities

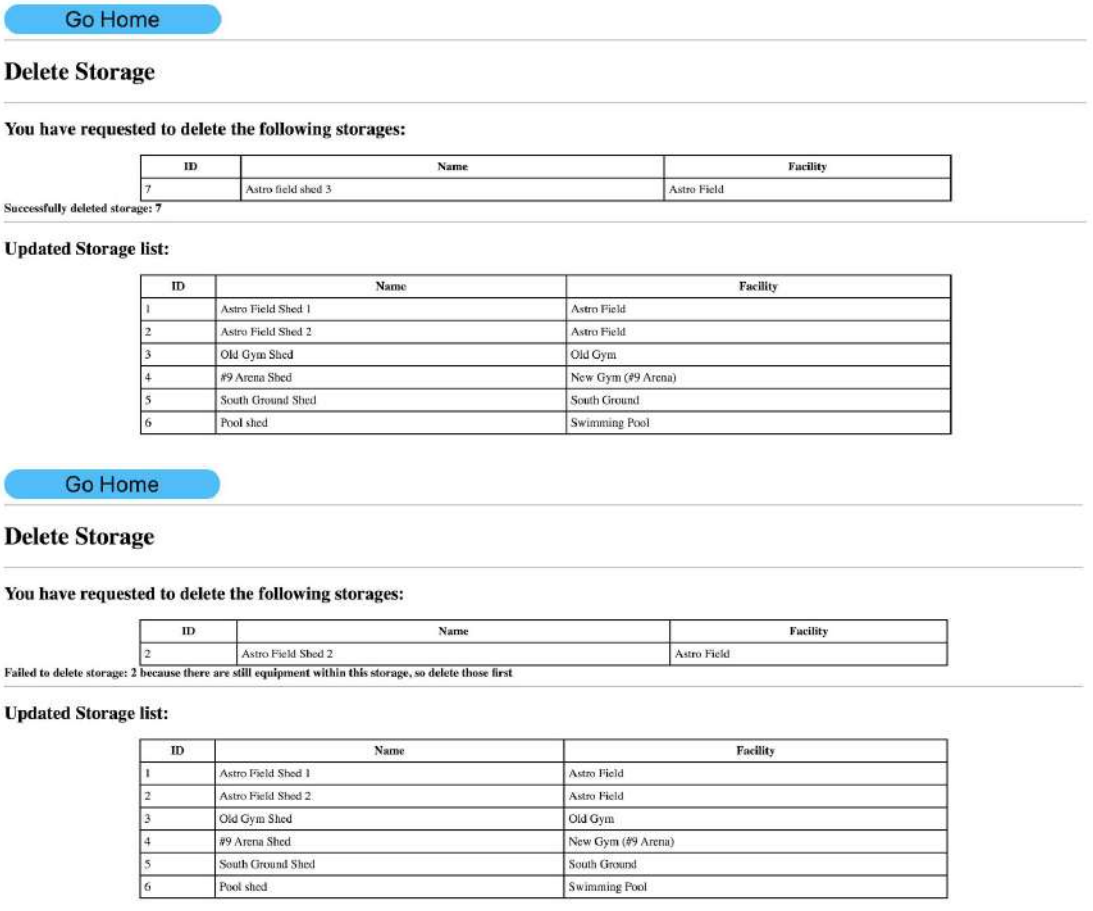
View/Edit Storage Section

View/Edit Storage page (edit_storage.php)

Test/Success Criteria

- ❖ All storages displayed in a table format
  - ➢ Column for ‘Storage ID’
  - ➢ Column for ‘Storage Name’
  - ➢ Column for ‘Facility Name’
  - ➢ Column for deleting storages (like ‘Delete Booking’ section)
    - ■ Each row has a checkbox to allow the user to check/fill in the box
    - ■ The column header has a ‘Delete’ button to allow user to delete select number of storages, and a reset button to allow user to restart
  - ➢ Clear button to allow user to delete all storages in the storage table
  - ➢ If the storage table is empty and there are no storages in the database, display an empty table informing the user that there are no storages
- ❖ ‘Add New Storage’ button to redirect user to Add storage page
- ❖ ‘Make changes to storage’ button to redirect user to edit storage page

Storages Deleted page (delete_storages.view.php)



Go Home

Delete Storage

You have requested to delete the following storages:

ID	Name	Facility
7	Astro field shed 3	Astro Field

Successfully deleted storage: 7

Updated Storage list:

ID	Name	Facility
1	Astro Field Shed 1	Astro Field
2	Astro Field Shed 2	Astro Field
3	Old Gym Shed	Old Gym
4	#9 Arena Shed	New Gym (#9 Arena)
5	South Ground Shed	South Ground
6	Pool shed	Swimming Pool

Go Home

Delete Storage

You have requested to delete the following storages:

ID	Name	Facility
2	Astro Field Shed 2	Astro Field

Failed to delete storage: 2 because there are still equipment within this storage, so delete those first

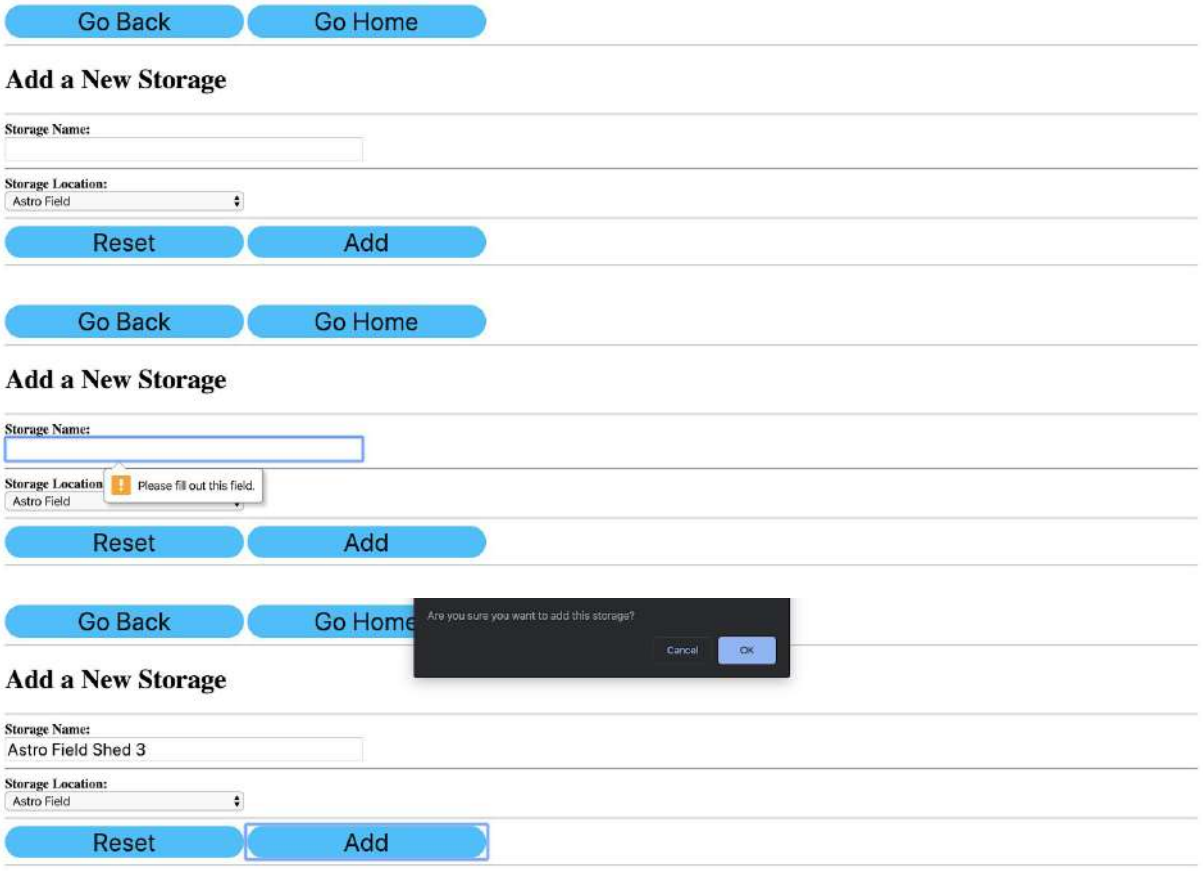
Updated Storage list:

ID	Name	Facility
1	Astro Field Shed 1	Astro Field
2	Astro Field Shed 2	Astro Field
3	Old Gym Shed	Old Gym
4	#9 Arena Shed	New Gym (#9 Arena)
5	South Ground Shed	South Ground
6	Pool shed	Swimming Pool

Test/Success Criteria

- ❖ If the storage was deleted from the database, display to inform the user that it was successfully deleted
  - ➢ If selected storages were deleted, display these storages
  - ➢ If all storages were deleted, display all the storages
- ❖ If the storages couldn’t be removed from the table, inform the user that it couldn’t be deleted
- ❖ Display the updated storage table
  - ➢ If all storages were deleted, display an empty table informing the user that there are no storages

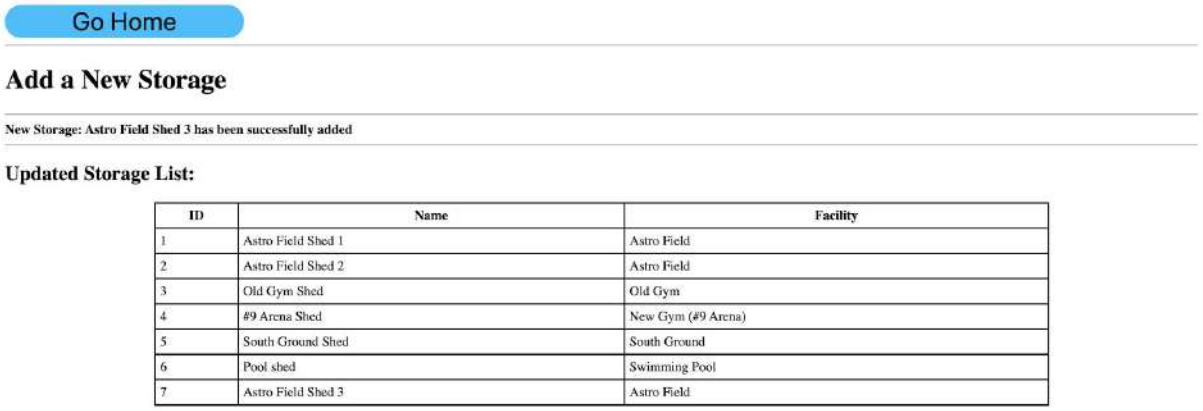
Add Storage page (add_storage.php)



Test/Success Criteria

- Form to allow the user to input the required information for a new storage
  - Text box to enter the storage name (required)
    - If the text box is left blank and the user tries to add the storage, they will be blocked
  - Drop down menu to select a facility/location for the storage (already selected with a default value)
  - ‘Add’ button to add the storage and proceed to the next page, and a rest button to restart
  - When pressing the add button, a warning message should be displayed to ask the user if they are sure they want to add the storage

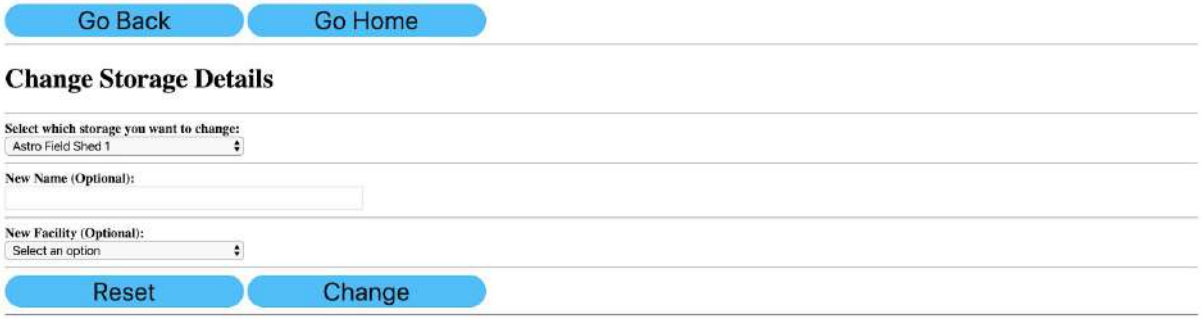
Storage Added page (add_storage.view.php)



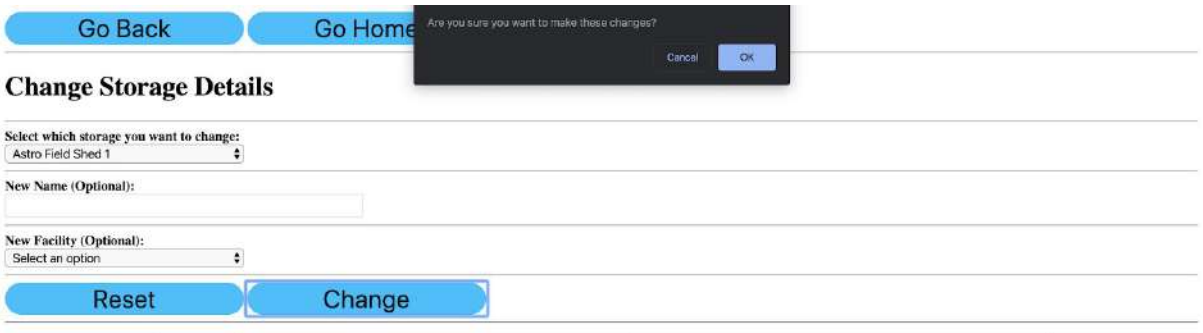
ID	Name	Facility
1	Astro Field Shed 1	Astro Field
2	Astro Field Shed 2	Astro Field
3	Old Gym Shed	Old Gym
4	#9 Arena Shed	New Gym (#9 Arena)
5	South Ground Shed	South Ground
6	Pool shed	Swimming Pool
7	Astro Field Shed 3	Astro Field

- If the storage was added to the database, display a message informing the user that it was successfully added
- Display the updated storage table

Change Storage Details page (change_storage.php)



ID	Name	Facility
1	Astro Field Shed 1	Astro Field
2	Astro Field Shed 2	Astro Field
3	Old Gym Shed	Old Gym
4	#9 Arena Shed	New Gym (#9 Arena)
5	South Ground Shed	South Ground
6	Pool shed	Swimming Pool
7	Astro Field Shed 3	Astro Field



ID	Name	Facility
1	Astro Field Shed 1	Astro Field
2	Astro Field Shed 2	Astro Field
3	Old Gym Shed	Old Gym
4	#9 Arena Shed	New Gym (#9 Arena)
5	South Ground Shed	South Ground
6	Pool shed	Swimming Pool
7	Astro Field Shed 3	Astro Field

Test/Success Criteria

- ❖ Form to allow user to select an storage and change their details
  - ➢ Drop down menu with the storages as options (already selected with a default value)
  - ➢ Drop down menu with the new facilities as options (optional)
  - ➢ Text box to input the storages new name (optional)
  - ➢ ‘Change’ button to proceed and a ‘Reset’ button to restart if needed

Storage Details Changed page (change_storage.view.php)

Go Home

Change Storage Details

You made no changes to the storage

Updated Storage List:

ID	Name	Facility
1	Astro Field Shed 1	Astro Field
2	Astro Field Shed 2	Astro Field
3	Old Gym Shed	Old Gym
4	#9 Arena Shed	New Gym (#9 Arena)
5	South Ground Shed	South Ground
6	Pool shed	Swimming Pool
7	Astro Field Shed 4	Astro Field

Go Home

Change Storage Details

Storage: 7 name has successfully been changed to: Astro Field Shed 3

Updated Storage List:

ID	Name	Facility
1	Astro Field Shed 1	Astro Field
2	Astro Field Shed 2	Astro Field
3	Old Gym Shed	Old Gym
4	#9 Arena Shed	New Gym (#9 Arena)
5	South Ground Shed	South Ground
6	Pool shed	Swimming Pool
7	Astro Field Shed 3	Astro Field

Go Home

Change Storage Details

Storage: 7 facility has successfully been changed to facility: 2

Updated Storage List:

ID	Name	Facility
1	Astro Field Shed 1	Astro Field
2	Astro Field Shed 2	Astro Field
3	Old Gym Shed	Old Gym
4	#9 Arena Shed	New Gym (#9 Arena)
5	South Ground Shed	South Ground
6	Pool shed	Swimming Pool
7	Astro Field Shed 3	Old Gym

Go Home

Change Storage Details

Storage: 7 name has successfully been changed to: Astro Field Shed 4
Storage: 7 facility has successfully been changed to facility: 1

Updated Storage List:

ID	Name	Facility
1	Astro Field Shed 1	Astro Field
2	Astro Field Shed 2	Astro Field
3	Old Gym Shed	Old Gym
4	#9 Arena Shed	New Gym (#9 Arena)
5	South Ground Shed	South Ground
6	Pool shed	Swimming Pool
7	Astro Field Shed 4	Astro Field

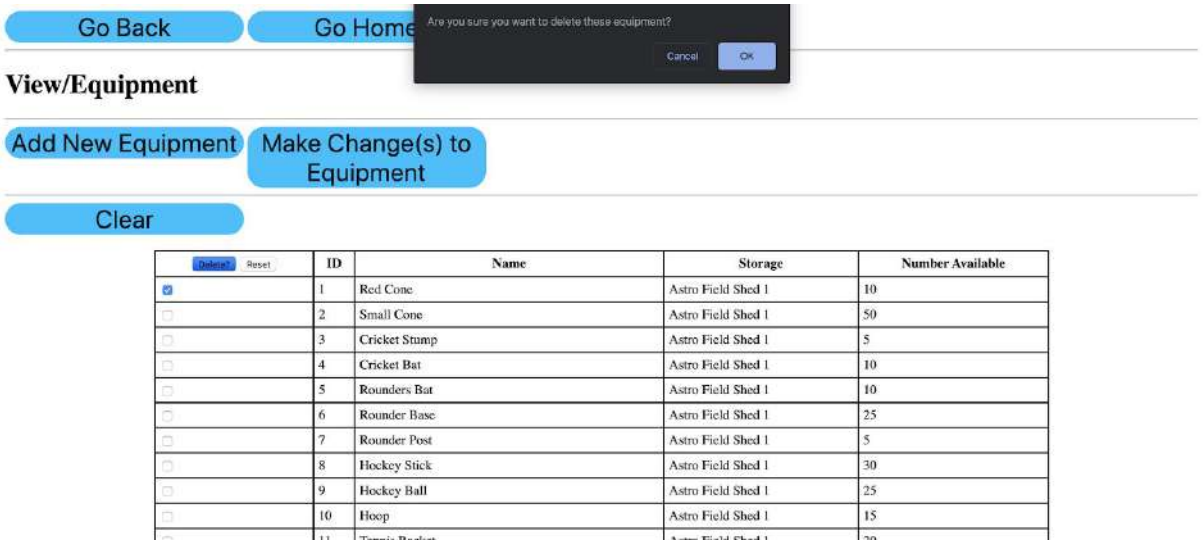
Test/Success Criteria

New Name	New Facility	Expected Output
0	0	“No changes”
0	1	“Facility changed”
1	0	“Name changed”
1	1	“Name changed “Facility changed”

- If the storage’s details have been changed, display a message to inform the user that it has been successfully changed
  - Display each of the changes made to the storage (storage name and/or facility/location)
- If the storage’s details couldn’t be changed, display a message to inform the user that the changes couldn’t be made
- Display a table of the updated storages

View/Edit Equipment Section

View/Edit Equipment page (edit_equipment.php)



Test/Success Criteria

- All equipment displayed in a table format
  - Column for ‘Equipment ID’
  - Column for ‘Equipment Name’
  - Column for ‘Number Available’
  - Column for ‘Storage Name’
  - Column for deleting equipment (like ‘Delete Booking’ section)
    - Each row has a checkbox to allow the user to check/fill in the box
    - The column header has a ‘Delete’ button to allow user to delete select number of equipment, and a reset button to allow user to restart
  - Clear button to allow user to delete all equipment in the storage table
  - If the equipment table is empty and there are no equipment in the database, display an empty table informing the user that there are no equipment
- ‘Add New Equipment’ button to redirect user to Add equipment page
- ‘Make changes to equipment’ button to redirect user to edit equipment page

Equipment Deleted page (delete_equipment.view.php)

Go Home

Delete Equipment

You have requested to delete the following equipment:

ID	Name	Storage	Number Available
1	Red Cone	Astro Field Shed 1	10
2	Small Cone	Astro Field Shed 1	50
3	Cricket Stump	Astro Field Shed 1	5

Failed to delete equipment: 1 because there are still bookings with this equipment, so delete those first
Failed to delete equipment: 2 because there are still bookings with this equipment, so delete those first
Failed to delete equipment: 3 because there are still bookings with this equipment, so delete those first

Updated Equipment list:

ID	Name	Storage	Number Available
1	Red Cone	Astro Field Shed 1	10
2	Small Cone	Astro Field Shed 1	50
3	Cricket Stump	Astro Field Shed 1	5
4	Cricket Bat	Astro Field Shed 1	10
5	Rounders Bat	Astro Field Shed 1	10
6	Rounder Base	Astro Field Shed 1	25
7	Rounder Post	Astro Field Shed 1	5
8	Hockey Stick	Astro Field Shed 1	30
9	Hockey Ball	Astro Field Shed 1	25

Go Home

Delete Equipment

You have requested to delete the following equipment:

ID	Name	Storage	Number Available
4	Cricket Bat	Astro Field Shed 1	10

Successfully deleted equipment: 4

Updated Equipment list:

ID	Name	Storage	Number Available
1	Red Cone	Astro Field Shed 1	10
2	Small Cone	Astro Field Shed 1	50
3	Cricket Stump	Astro Field Shed 1	5
5	Rounders Bat	Astro Field Shed 1	10
6	Rounder Base	Astro Field Shed 1	25

Success Criteria

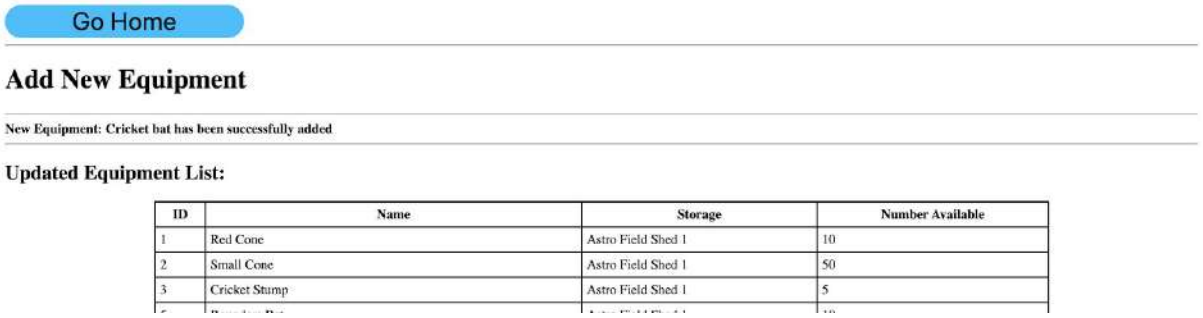
- If the equipment was deleted from the database, display to inform the user that it was successfully deleted
  - If selected equipment were deleted, display these equipment
  - If all equipment were deleted, display all the equipment
- If the equipment couldn't be removed from the table, inform the user that it couldn't be deleted
- Display the updated equipment table
  - If all equipment were deleted, display an empty table informing the user that there are no equipment

Add Equipment page (add_equipment.php)

Test/Success Criteria

- Form to allow the user to input the required information for a new equipment
  - Text box to enter the equipment name (required)
    - If the text box is left blank and the user tries to add the equipment, they will be blocked
  - Drop down menu to select a storage space for the equipment (already selected with a default value)
  - Number input to enter the number available
  - ‘Add’ button to add the equipment and proceed to the next page, and a rest button to restart
  - When pressing the add button, a warning message should be displayed to ask the user if they are sure they want to add the equipment

Equipment Added page (add_equipment.view.php)

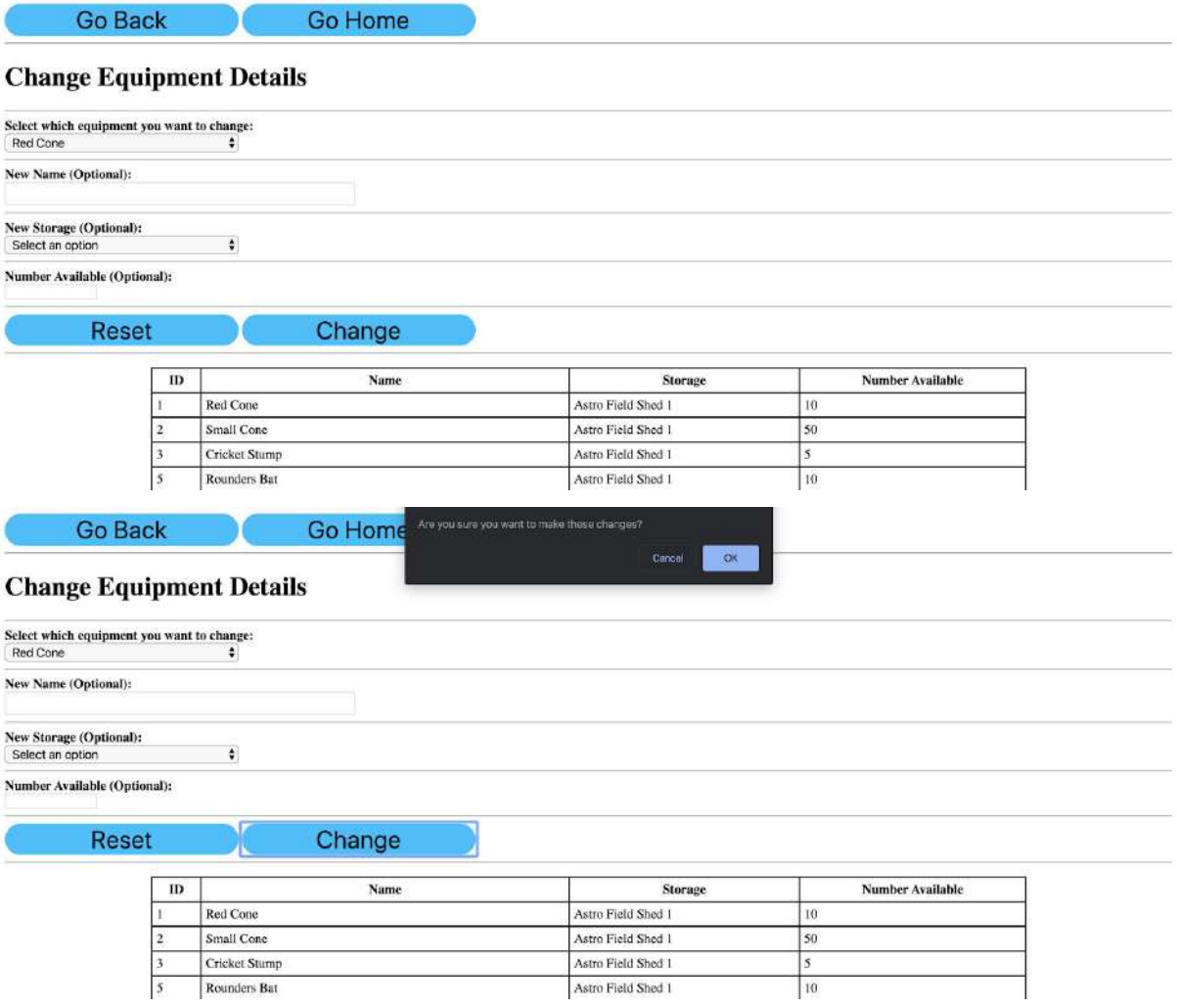


ID	Name	Storage	Number Available
1	Red Cone	Astro Field Shed 1	10
2	Small Cone	Astro Field Shed 1	50
3	Cricket Stump	Astro Field Shed 1	5

Test/Success Criteria

- If the equipment was added to the database, display a message informing the user that it was successfully added
- Display the updated equipment table

Change Equipment page (change_equipment.php)



ID	Name	Storage	Number Available
1	Red Cone	Astro Field Shed 1	10
2	Small Cone	Astro Field Shed 1	50
3	Cricket Stump	Astro Field Shed 1	5
5	Rounders Bat	Astro Field Shed 1	10

ID	Name	Storage	Number Available
1	Red Cone	Astro Field Shed 1	10
2	Small Cone	Astro Field Shed 1	50
3	Cricket Stump	Astro Field Shed 1	5
5	Rounders Bat	Astro Field Shed 1	10

Test/Success Criteria

- ❖ Form to allow user to select an equipment and change their details
  - ➢ Drop down menu with the equipment as options (already selected with a default value)
  - ➢ Drop down menu with the new storages as options (optional)
  - ➢ Text box to input the new equipment name (optional)
  - ➢ Number input to enter the new number available (optional)
  - ➢ 'Change' button to proceed and a 'Reset' button to restart if needed

Equipment Details Changed page (change_equipment.view.php)

Go Home

Change Equipment Details

You made no changes to the equipment

Updated Equipment List:

ID	Name	Storage	Number Available
1	Red Cone	Astro Field Shed 1	10
2	Small Cone	Astro Field Shed 1	50
3	Cricket Stump	Astro Field Shed 1	5
5	Rounders Bat	Astro Field Shed 1	10
6	Rounder Base	Astro Field Shed 1	25

Go Home

Change Equipment Details

Equipment: 1 number has successfully been changed to: 20

Updated Equipment List:

ID	Name	Storage	Number Available
1	Red Cone	Astro Field Shed 1	20
2	Small Cone	Astro Field Shed 1	50
3	Cricket Stump	Astro Field Shed 1	5
5	Rounders Bat	Astro Field Shed 1	10
6	Rounder Base	Astro Field Shed 1	25

Go Home

Change Equipment Details

Equipment: 1 storage has successfully been changed to storage: 2

Updated Equipment List:

ID	Name	Storage	Number Available
1	Red Cone	Astro Field Shed 2	20
2	Small Cone	Astro Field Shed 1	50
3	Cricket Stump	Astro Field Shed 1	5
5	Rounders Bat	Astro Field Shed 1	10
6	Rounder Base	Astro Field Shed 1	25

Go Home

Change Equipment Details

Equipment: 1 storage has successfully been changed to storage: 1
Equipment: 1 number has successfully been changed to: 10

Updated Equipment List:

ID	Name	Storage	Number Available
1	Red Cone	Astro Field Shed 1	10
2	Small Cone	Astro Field Shed 1	50
3	Cricket Stump	Astro Field Shed 1	5
5	Rounders Bat	Astro Field Shed 1	10

Go Home

Change Equipment Details

Equipment: 1 name has successfully been changed to: Red Cones

Updated Equipment List:

ID	Name	Storage	Number Available
1	Red Cones	Astro Field Shed 1	10
2	Small Cone	Astro Field Shed 1	50
3	Cricket Stump	Astro Field Shed 1	5
5	Rounders Bat	Astro Field Shed 1	10
6	Rounder Base	Astro Field Shed 1	25

Go Home

Change Equipment Details

Equipment: 1 name has successfully been changed to: Red Cone
Equipment: 1 number has successfully been changed to: 20

Updated Equipment List:

ID	Name	Storage	Number Available
1	Red Cone	Astro Field Shed 1	20
2	Small Cone	Astro Field Shed 1	50
3	Cricket Stump	Astro Field Shed 1	5
5	Rounders Bat	Astro Field Shed 1	10

Go Home

Change Equipment Details

Equipment: 1 name has successfully been changed to: Red Cones
Equipment: 1 storage has successfully been changed to storage: 2

Updated Equipment List:

ID	Name	Storage	Number Available
1	Red Cones	Astro Field Shed 2	20
2	Small Cone	Astro Field Shed 1	50
3	Cricket Stump	Astro Field Shed 1	5
5	Rounders Bat	Astro Field Shed 1	10

Go Home

Change Equipment Details

Equipment: 1 name has successfully been changed to: Red Cone
Equipment: 1 storage has successfully been changed to storage: 1
Equipment: 1 number has successfully been changed to: 10

Updated Equipment List:

ID	Name	Storage	Number Available
1	Red Cone	Astro Field Shed 1	10
2	Small Cone	Astro Field Shed 1	50
3	Cricket Stump	Astro Field Shed 1	5
5	Rounders Bat	Astro Field Shed 1	10

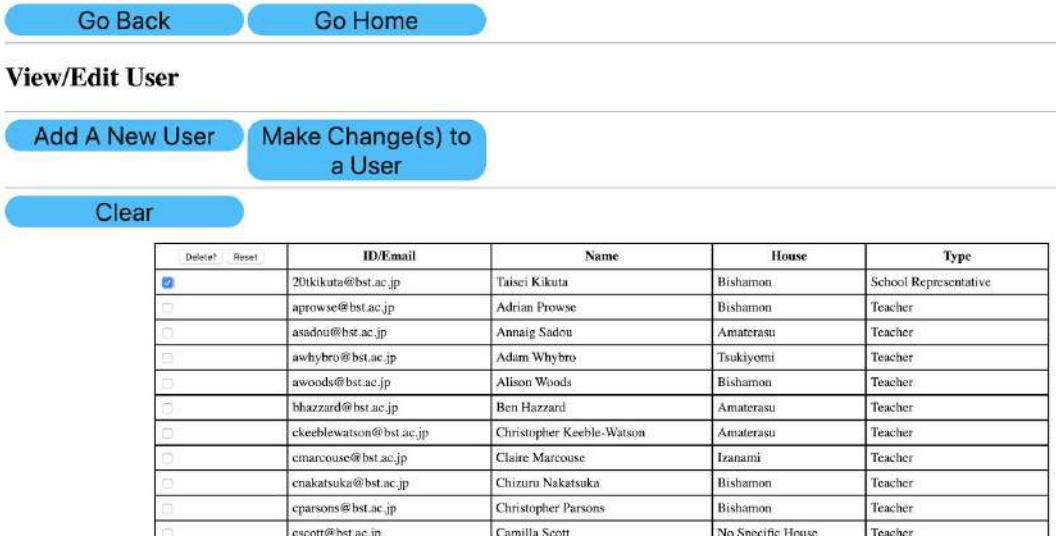
Test/Success Criteria

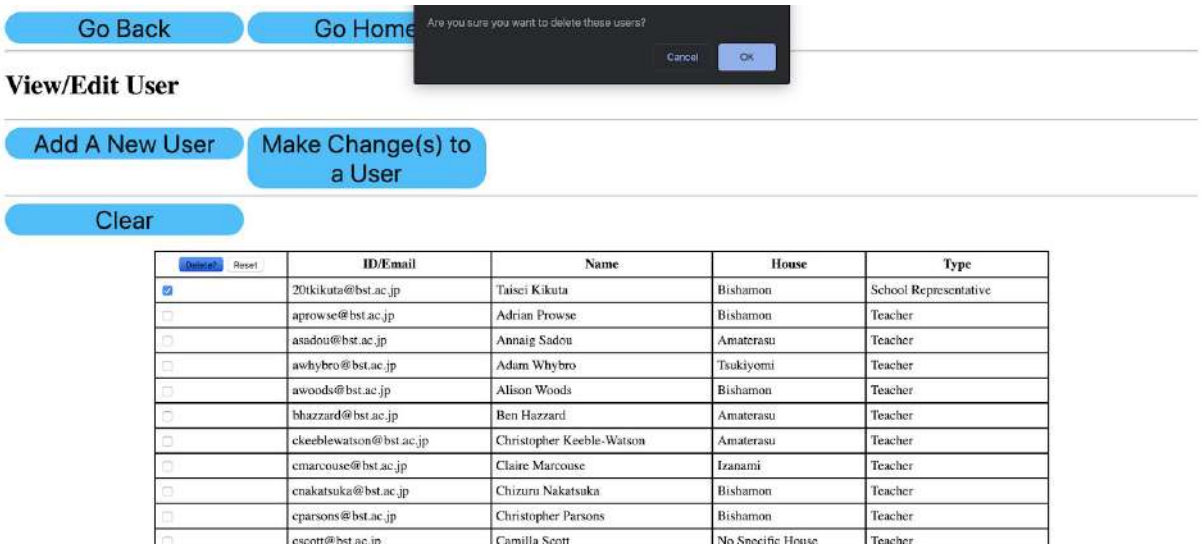
New Name	New Storage	New Number	Expected Output
0	0	0	“No changes”
0	0	1	“Number changed”
0	1	0	“Storage changed”
0	1	1	“Storage changed “Number changed”
1	0	0	“Name changed”
1	0	1	“Name changed” “Number changed”
1	1	0	“Name changed” “Storage changed”
1	1	1	“Name changed” “Storage changed” “Number changed”

- If the equipment details have been changed, display a message to inform the user that it has been successfully changed
  - Display each of the changes made to the equipment (equipment name and/or storage and/or number)
- If the equipment details couldn’t be changed, display a message to inform the user that the changes couldn’t be made
- Display a table of the updated equipment

View/Edit User Section

View/Edit User page (edit_user.php)

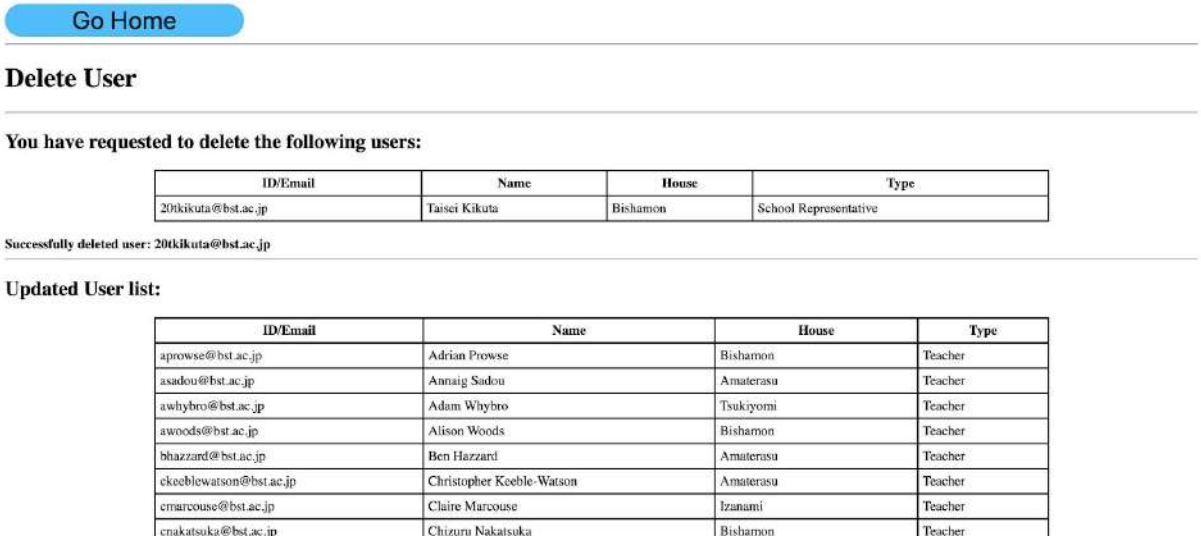


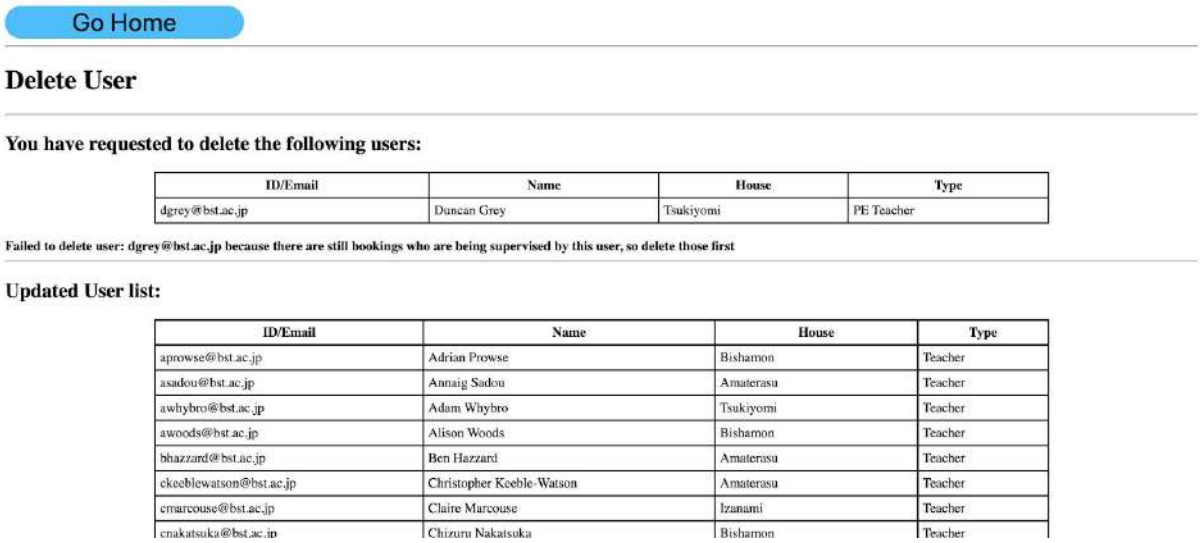


Test/Success Criteria

- All users displayed in a table format
  - Column for ‘User ID’
  - Column for ‘User Name’
  - Column for ‘House’
  - Column for ‘User Type’
  - Column for deleting users (like ‘Delete Booking’ section)
    - Each row has a checkbox to allow the user to check/fill in the box
    - The column header has a ‘Delete’ button to allow user to delete select users, and a reset button to allow them to restart
  - Clear button to allow user to delete all users in the user table
  - If the users table is empty and there are no users in the table, display an empty table informing the user that there are no users
- ‘Add New User’ button to redirect user to Add users page
- ‘Make changes to user’ button to redirect user to edit user page

User Deleted page (delete_users.view.php)

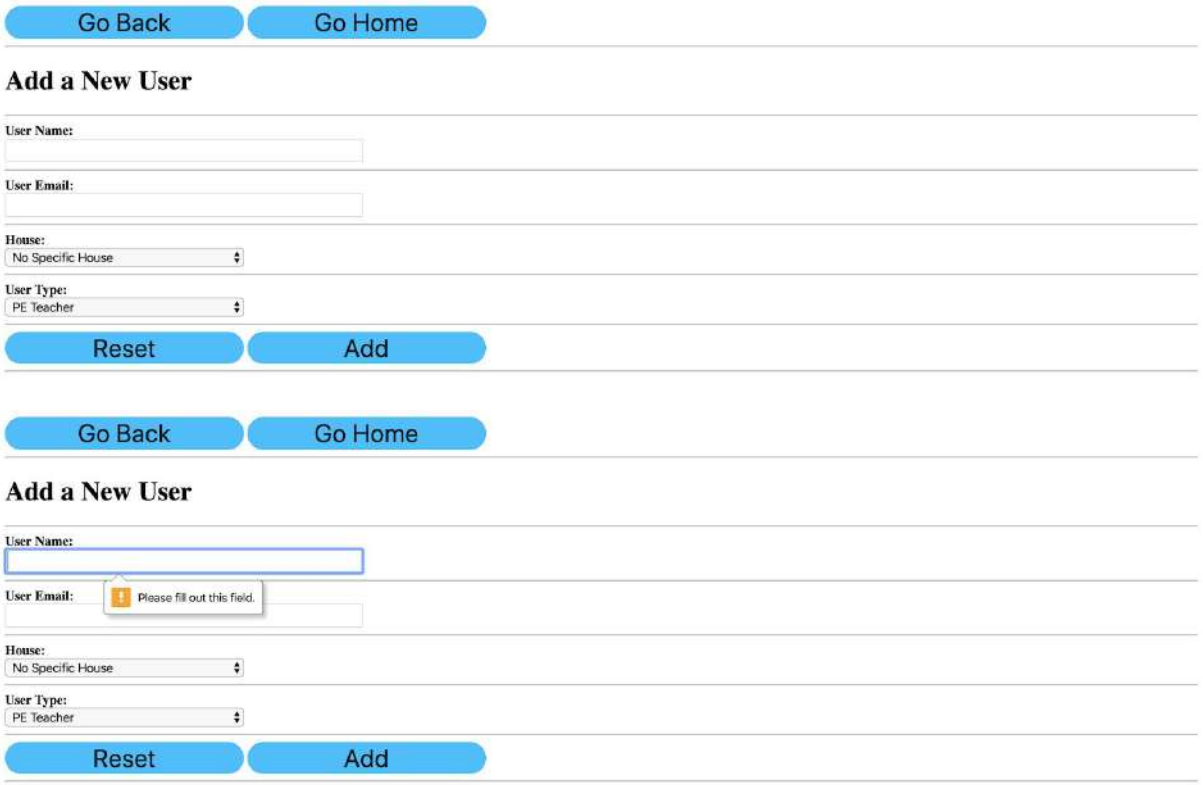


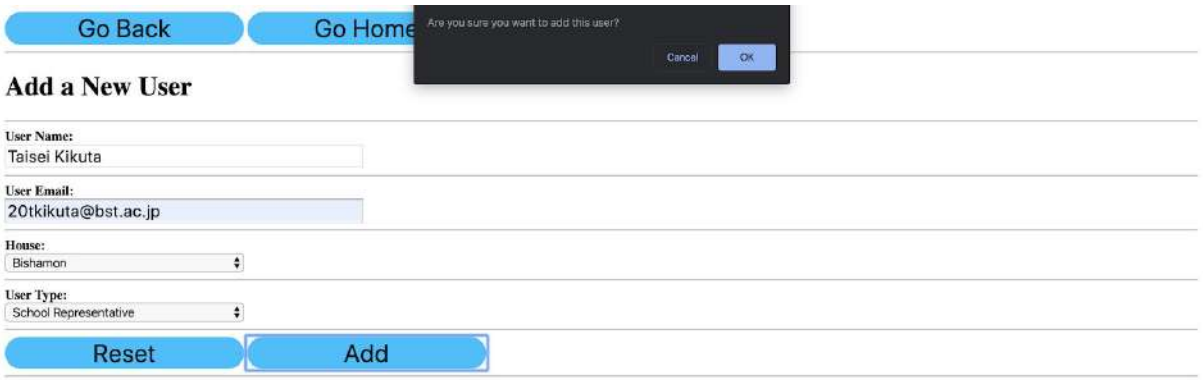


Test/Success Criteria

- If the user was deleted from the database, display to inform the user that it was successfully deleted
  - If selected users were deleted, display these users
  - If all users were deleted, display all the users
- If the users couldn't be removed from the table, inform the user that it couldn't be deleted
- Display the updated users table
  - If all users were deleted, display an empty table informing the user that there are none

Add User page (add_user.php)

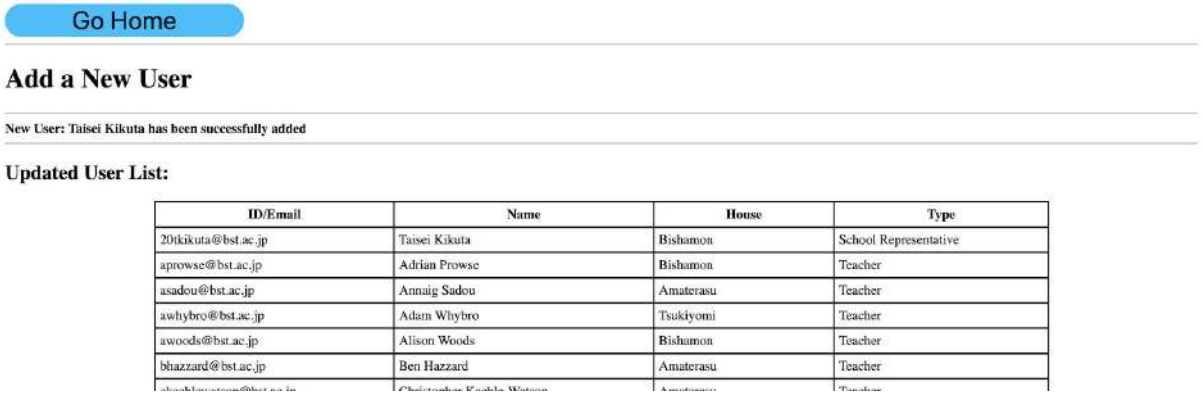




Test/Success Criteria

- Form to allow the user to input the required information for a new user/staff
  - Text box to enter the user name (required)
    - If the text box is left blank and the user tries to add the user, they will be blocked
  - Text box to enter the user ID (required)
    - If the text box is left blank and the user tries to add the user, they will be blocked
  - Drop down menu to select a house for the user (already selected with a default value)
  - Drop down menu to select a user type for the user (already selected with a default value)
  - 'Add' button to add the user and proceed to the next page, and a rest button to restart
  - When pressing the add button, a warning message should be displayed to ask the user if they are sure they want to add the user

User Added page (add_user.view.php)



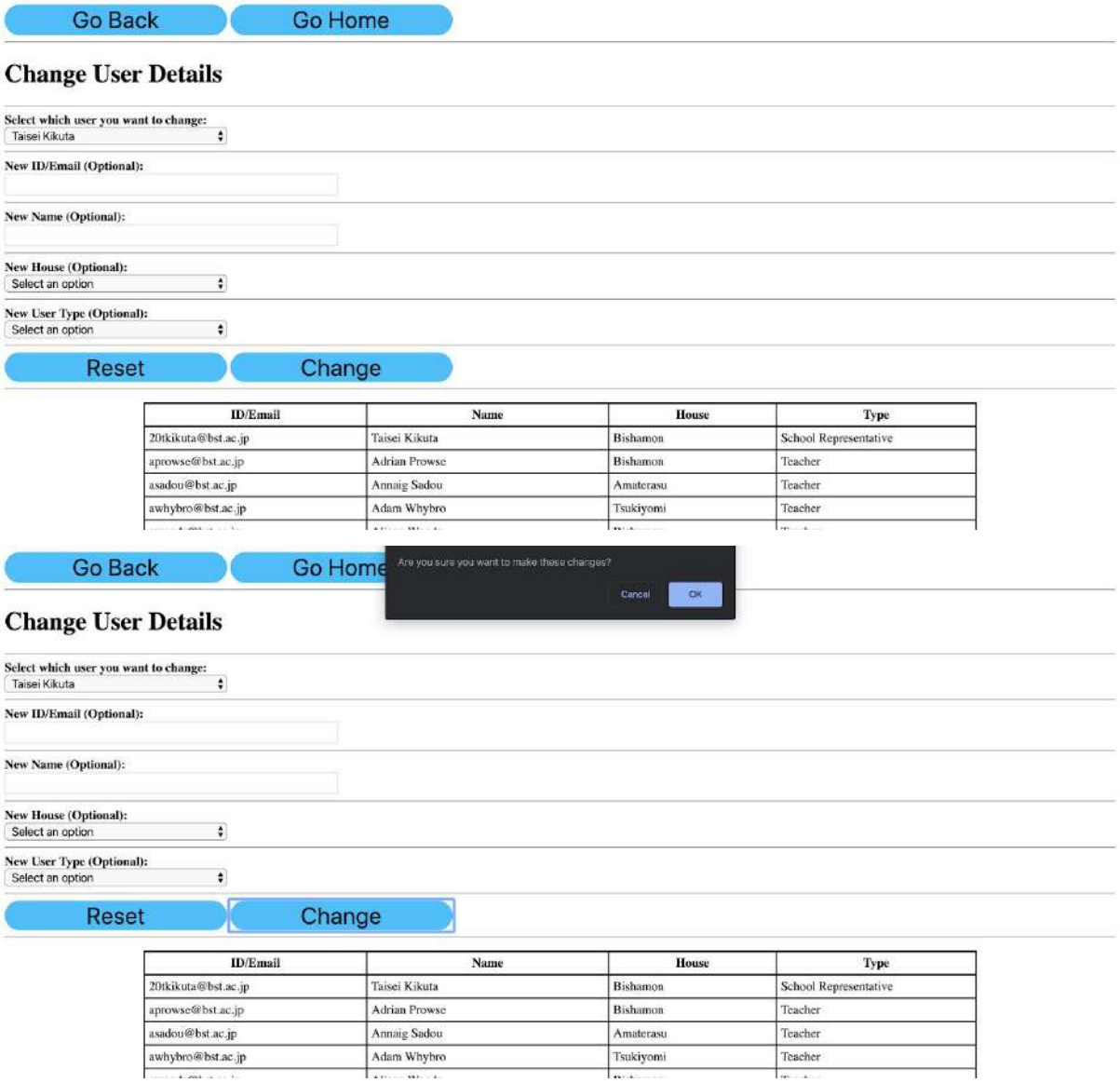
ID/Email	Name	House	Type
20tkikuta@bst.ac.jp	Taisei Kikuta	Bishamon	School Representative
aprowse@bst.ac.jp	Adrian Prowse	Bishamon	Teacher
asadou@bst.ac.jp	Annaig Sadou	Amaterasu	Teacher
awhybro@bst.ac.jp	Adam Whybro	Tsukiyomi	Teacher
awoods@bst.ac.jp	Alison Woods	Bishamon	Teacher
bhazzard@bst.ac.jp	Ben Hazzard	Amaterasu	Teacher

Number of users in 'Bishamon' house increased from 11 to 12

Test/Success Criteria

- If the user was added to the database, display a message informing the user that it was successfully added
  - Increase the number of users/staff in the house table by 1
- Display the updated users table

Change User Details page (change_user.php)



Test/Success Criteria

- Form to allow user to select a user and change their details
  - Drop down menu with the users as options (already selected with a default value)
  - Drop down menu with the new house as options (optional)
  - Drop down menu with the new user types as options (optional)
  - Text box to input the new user name (optional)
  - Text box to input the new user email (optional)
  - ‘Change’ button to proceed and a ‘Reset’ button to restart if needed

User Details Changed page (change_user.view.php)

Go Home

Change User Details

You made no changes to the user

Updated User List:

ID/Email	Name	House	Type
20tkikuta@bst.ac.jp	Taisei Kikuta	Bishamon	School Representative
aprowse@bst.ac.jp	Adrian Prowse	Bishamon	Teacher

Go Home

Change User Details

User: 20tkikuta@bst.ac.jp user type has successfully been changed to: 1

Updated User List:

ID/Email	Name	House	Type
20tkikuta@bst.ac.jp	Taisei Kikuta	Bishamon	PE Teacher
aprowse@bst.ac.jp	Adrian Prowse	Bishamon	Teacher

Go Home

Change User Details

User: 20tkikuta@bst.ac.jp house has successfully been changed to: 1

Updated User List:

ID/Email	Name	House	Type
20tkikuta@bst.ac.jp	Taisei Kikuta	No Specific House	PE Teacher
aprowse@bst.ac.jp	Adrian Prowse	Bishamon	Teacher

Go Home

Change User Details

User: 20tkikuta@bst.ac.jp house has successfully been changed to: 3
User: 20tkikuta@bst.ac.jp user type has successfully been changed to: 3

Updated User List:

ID/Email	Name	House	Type
20tkikuta@bst.ac.jp	Taisei Kikuta	Bishamon	School Representative
aprowse@bst.ac.jp	Adrian Prowse	Bishamon	Teacher

Go Home

Change User Details

User: 20tkikuta@bst.ac.jp name has successfully been changed to: Taisei

Updated User List:

ID/Email	Name	House	Type
20tkikuta@bst.ac.jp	Taisei	Bishamon	School Representative
aprowse@bst.ac.jp	Adrian Prowse	Bishamon	Teacher

Go Home

Change User Details

User: 20tkikuta@bst.ac.jp name has successfully been changed to: Taisei Kikuta
User: 20tkikuta@bst.ac.jp user type has successfully been changed to: 2

Updated User List:

ID/Email	Name	House	Type
20tkikuta@bst.ac.jp	Taisei Kikuta	Bishamon	Teacher
aprowse@bst.ac.jp	Adrian Prowse	Bishamon	Teacher

Go Home

Change User Details

User: 20tkikuta@bst.ac.jp name has successfully been changed to: Taisei
User: 20tkikuta@bst.ac.jp house has successfully been changed to: 1

Updated User List:

ID/Email	Name	House	Type
20tkikuta@bst.ac.jp	Taisei	No Specific House	Teacher
aprowse@bst.ac.jp	Adrian Prowse	Bishamon	Teacher

Go Home

Change User Details

User: 20tkikuta@bst.ac.jp name has successfully been changed to: Taisei Kikuta
User: 20tkikuta@bst.ac.jp house has successfully been changed to: 3
User: 20tkikuta@bst.ac.jp user type has successfully been changed to: 3

Updated User List:

ID/Email	Name	House	Type
20tkikuta@bst.ac.jp	Taisei Kikuta	Bishamon	School Representative

Go Home

Change User Details

User: 20tkikuta@bst.ac.jp email/ID has successfully been changed to: test@gmail.com

Updated User List:

ID/Email	Name	House	Type
aprowse@bst.ac.jp	Adrian Prowse	Bishamon	Teacher
asadou@bst.ac.jp	Annaig Sadou	Amaterasu	Teacher

Go Home

Change User Details

User: test@gmail.com user type has successfully been changed to: 1
User: test@gmail.com email/ID has successfully been changed to: 20tkikuta@bst.ac.jp

Updated User List:

ID/Email	Name	House	Type
20tkikuta@bst.ac.jp	Taisei Kikuta	Bishamon	PE Teacher
aprowse@bst.ac.jp	Adrian Prowse	Bishamon	Teacher

Go Home

Change User Details

User: 20tkikuta@bst.ac.jp house has successfully been changed to: 1
User: 20tkikuta@bst.ac.jp email/ID has successfully been changed to: test@gmail.com

Updated User List:

ID/Email	Name	House	Type
aprowse@bst.ac.jp	Adrian Prowse	Bishamon	Teacher
asadou@bst.ac.jp	Annaig Sadou	Amaterasu	Teacher

Go Home

Change User Details

User: test@gmail.com house has successfully been changed to: 3
User: test@gmail.com user type has successfully been changed to: 3
User: test@gmail.com email/ID has successfully been changed to: 20tkikuta@bst.ac.jp

Updated User List:

ID/Email	Name	House	Type
20tkikuta@bst.ac.jp	Taisei Kikuta	Bishamon	School Representative

Go Home

Change User Details

User: 20tkikuta@bst.ac.jp name has successfully been changed to: Taisei
User: 20tkikuta@bst.ac.jp email/ID has successfully been changed to: test@gmail.com

Updated User List:

ID/Email	Name	House	Type
aprowse@bst.ac.jp	Adrian Prowse	Bishamon	Teacher
asadou@bst.ac.jp	Annaig Sadou	Amaterasu	Teacher

Go Home

Change User Details

User: test@gmail.com name has successfully been changed to: Taisei Kikuta
User: test@gmail.com user type has successfully been changed to: 1
User: test@gmail.com email/ID has successfully been changed to: 20tkikuta@bst.ac.jp

Updated User List:

ID/Email	Name	House	Type
20tkikuta@bst.ac.jp	Taisei Kikuta	Bishamon	PE Teacher

Go Home

Change User Details

User: 20tkikuta@bst.ac.jp name has successfully been changed to: Taisei
User: 20tkikuta@bst.ac.jp house has successfully been changed to: 1
User: 20tkikuta@bst.ac.jp email/ID has successfully been changed to: test@gmail.com

Updated User List:

ID/Email	Name	House	Type
aprowse@bst.ac.jp	Adrian Prowse	Bishamon	Teacher

Go Home

Change User Details

User: test@gmail.com name has successfully been changed to: Taisei Kikuta
User: test@gmail.com user type has successfully been changed to: 3
User: test@gmail.com house has successfully been changed to: 3
User: test@gmail.com email/ID has successfully been changed to: 20tkikuta@bst.ac.jp

Updated User List:

ID/Email	Name	House	Type
20tkikuta@bst.ac.jp	Taisei Kikuta	Bishamon	School Representative
aprowse@bst.ac.jp	Adrian Prowse	Bishamon	Teacher

Test/Success Criteria

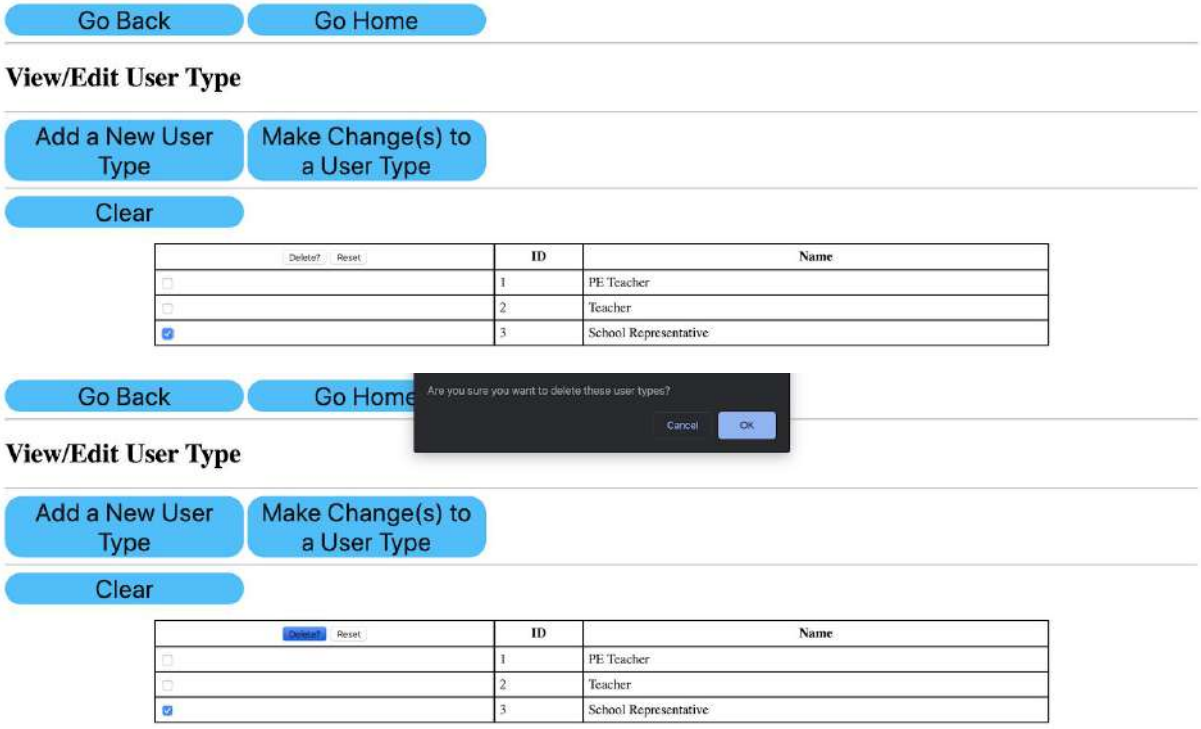
New ID/Email	New Name	New House	New User Type	Expected Output
0	0	0	0	“No changes”
0	0	0	1	“User type changed”
0	0	1	0	“House changed”
0	0	1	1	“User type changed” “House changed”
0	1	0	0	“Name changed”
0	1	0	1	“Name changed” “User type changed”
0	1	1	0	“Name changed” “House changed”
0	1	1	1	“Name changed” “House changed” “User type changed”
1	0	0	0	“ID changed”
1	0	0	1	“ID changed” “User type changed”
1	0	1	0	“ID changed” “House changed”
1	0	1	1	“ID changed” “House changed” “User type changed”
1	1	0	0	“ID changed” “Name changed”
1	1	0	1	“ID changed” “Name changed” “User type changed”
1	1	1	0	“ID changed” “Name changed” “House changed”
1	1	1	1	“ID changed” “Name changed” “House changed”

				“User type changed”

- ❖ If the user details have been changed, display a message to inform the user that it has been successfully changed
  - ➢ Display each of the changes made to the user (user name and/or user ID and/or house and/or user type)
  - ➢ If the user’s house was changed increase the number of staff/users by 1
- ❖ If the user details couldn’t be changed, display a message to inform the user that the changes couldn’t be made
- ❖ Display a table of the updated users

View/Edit User Type Section

View/Edit User page (edit_userType.php)



Test/Success Criteria

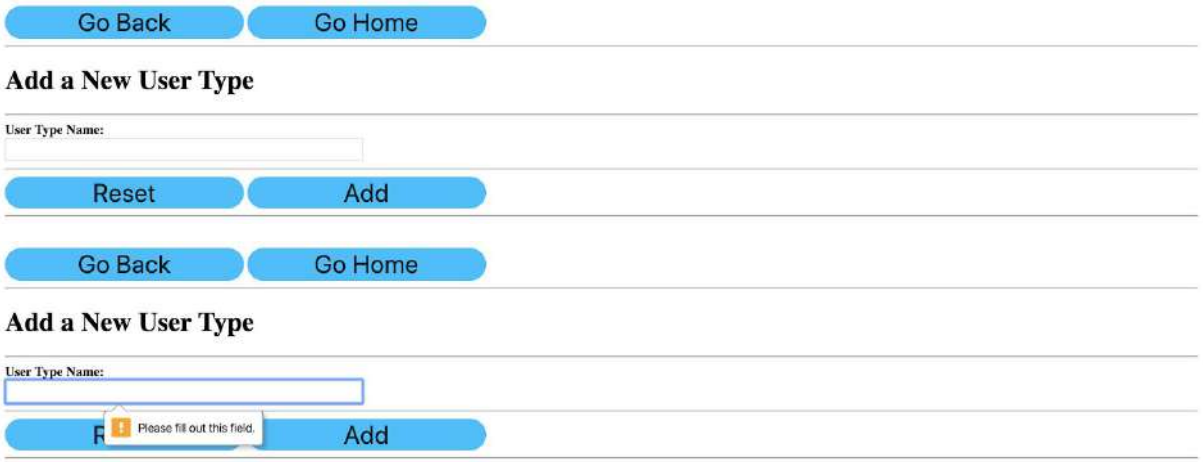
- ❖ All user types displayed in a table format
  - ➢ Column for ‘User Type ID’
  - ➢ Column for ‘User Type Name’
  - ➢ Column for deleting user types (like ‘Delete Booking’ section)
    - ■ Each row has a checkbox to allow the user to check/fill in the box
    - ■ The column header has a ‘Delete’ button to allow user to delete select user types, and a reset button to allow them to restart
  - ➢ Clear button to allow user to delete all user types in the user type table
  - ➢ If the user types table is empty and there are no user types in the database, display an empty table informing the user that there are no user types
- ❖ ‘Add New User Type’ button to redirect user to Add user types page
- ❖ ‘Make changes to User Type’ button to redirect user to edit user type page

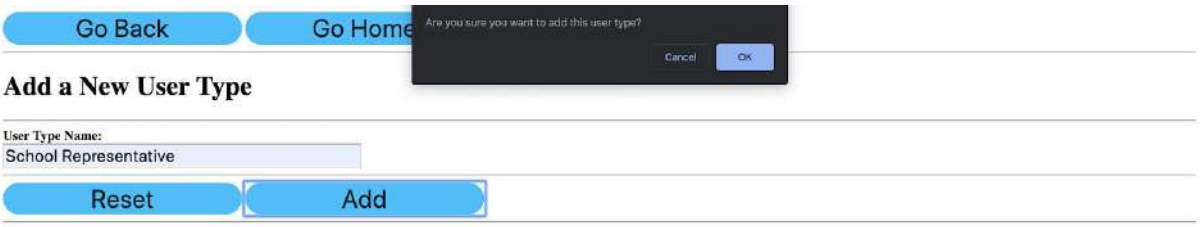
User Type Deleted page (delete_userTypes.view.php)

Test/Success Criteria

- If the user type was deleted from the database, display to inform the user that it was successfully deleted
  - If selected user types were deleted, display these user types
  - If all user types were deleted, display all the user types
- If the user types couldn't be removed from the table, inform the user that it couldn't be deleted
- Display the updated user types table
  - If all user types were deleted, display an empty table informing the user that there are no user types

Add User Type page (add_userType.php)

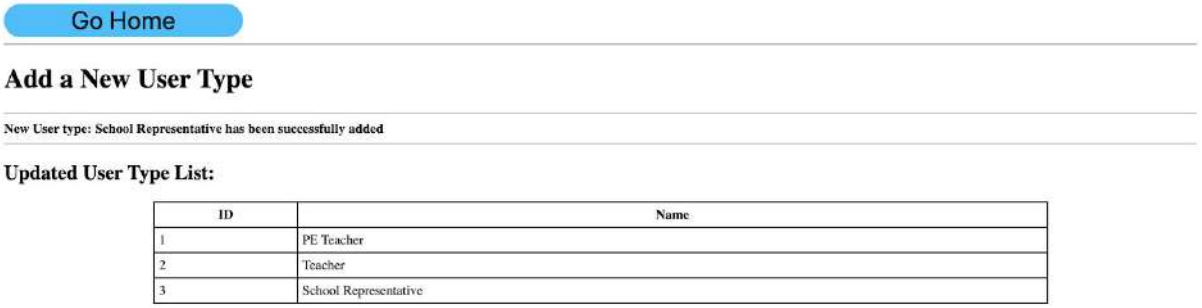




Test/Success Criteria

- Form to allow the user to input the required information for a new user type
  - Text box to enter the user type name (required)
    - If the text box is left blank and the user tries to add the user type, they will be blocked
  - 'Add' button to add the user type and proceed to the next page, and a reset button to restart
  - When pressing the add button, a warning message should be displayed to ask the user if they are sure they want to add the user type

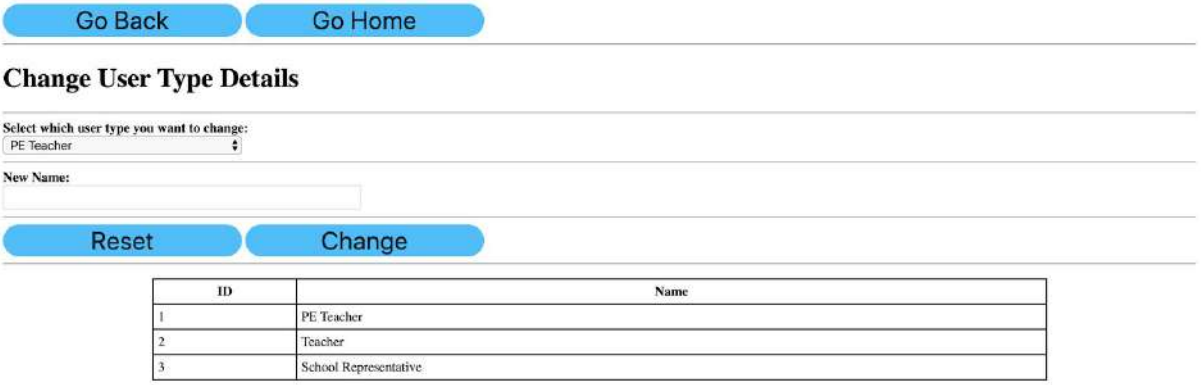
User Type Added page (add_userType.view.php)

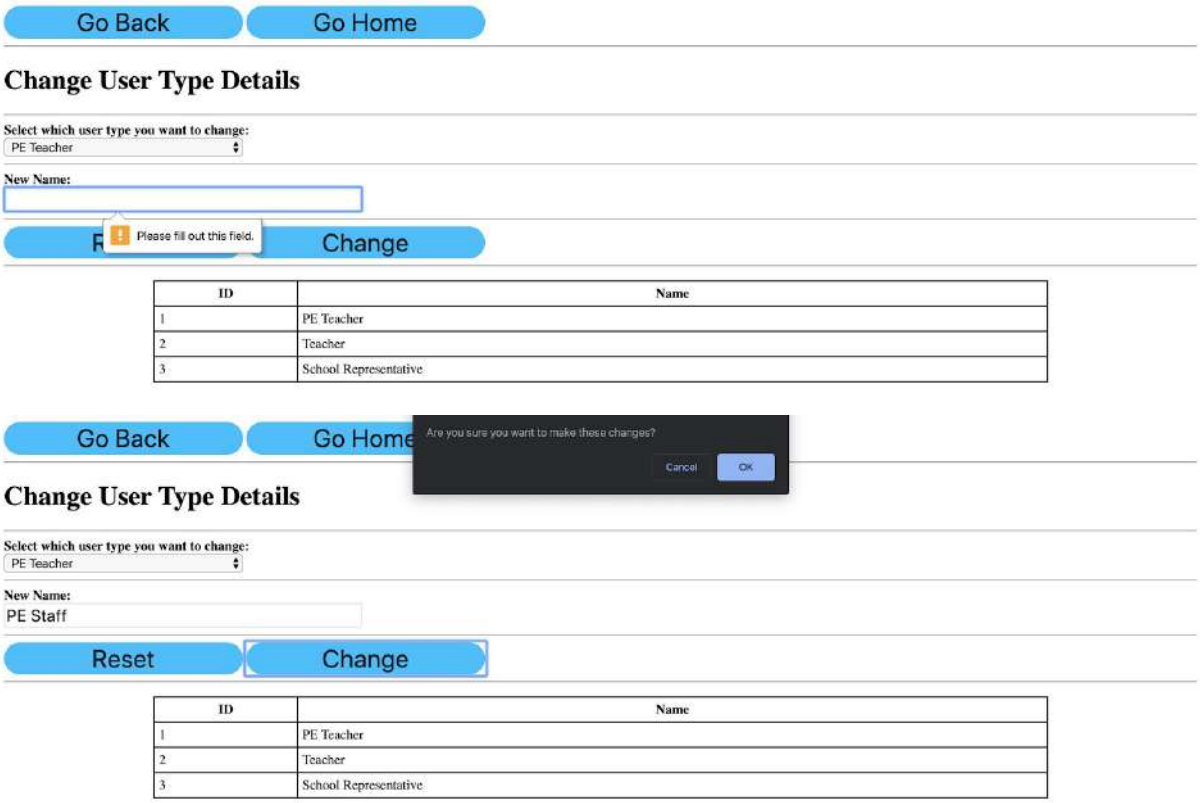


Test/Success Criteria

- If the user type was added to the table, display a message informing the user that it was successfully added
- Display the updated user type table

Change User Type Details page (change_userType.php)

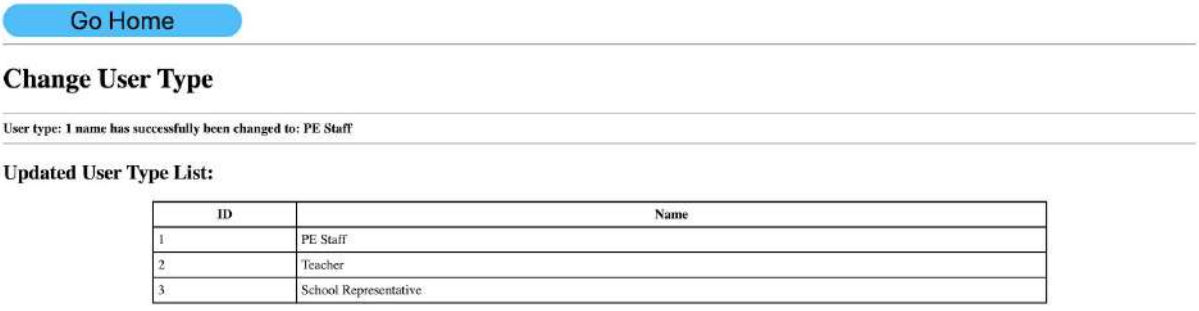




Test/Success Criteria

- Form to allow user to select a user type and change their details
  - Drop down menu with the user types as options (already selected with a default value)
  - Text box to input the new user type name (required)
    - If the user leaves this field blank, they will be blocked from adding the user type
  - 'Change' button to proceed and a 'Reset' button to restart if needed

User Type Details Changed (change_userType.vew.php)

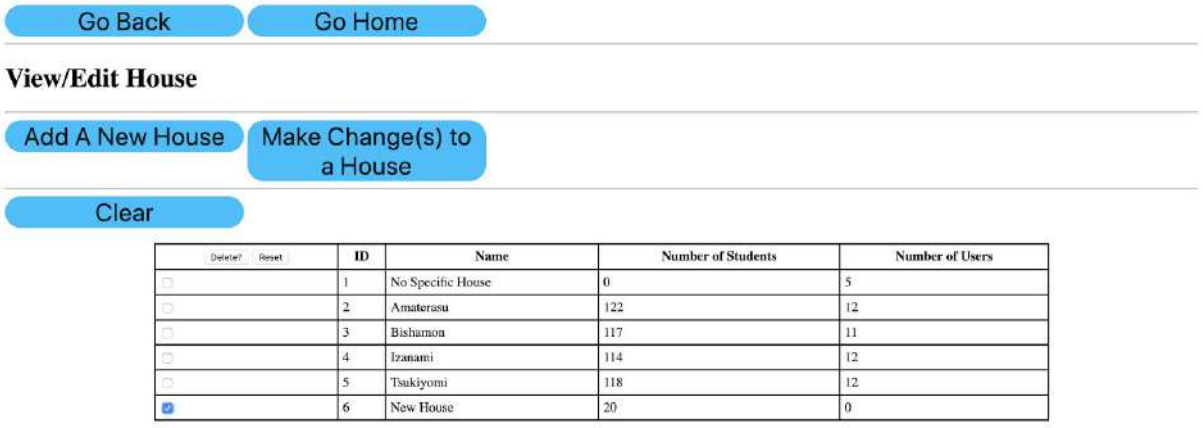


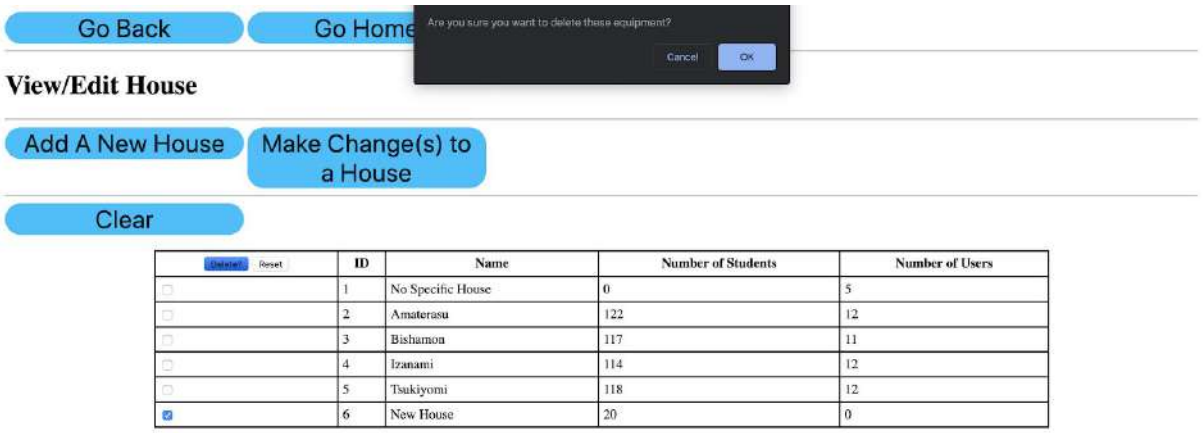
Test/Success Criteria

- If the user types details have been changed, display a message to inform the user that it has been successfully changed
- If the user types details couldn't be changed, display a message to inform the user that the changes couldn't be made
- Display a table of the updated user types

View/Edit House Section

View/Edit House page (edit_house.php)





Test/Success Criteria

- ❖ All houses displayed in a table format
  - ➢ Column for ‘House ID’
  - ➢ Column for ‘House Name’
  - ➢ Column for ‘Number of Users’
  - ➢ Column for ‘Number of Students
  - ➢ Column for deleting houses (like ‘Delete Booking’ section)
    - ■ Each row has a checkbox to allow the user to check/fill in the box
    - ■ The column header has a ‘Delete’ button to allow user to delete select houses, and a reset button to allow them to restart
  - ➢ Clear button to allow user to delete all houses in the database
  - ➢ If the house table is empty display an empty table informing the user that there are no houses

House Deleted page (delete_houses.view.php)

Go Home

Delete House

You have requested to delete the following houses:

ID	Name	Number of Students	Number of Users
6	New House	20	0

Successfully deleted house: 6

Updated House list:

ID	Name	Number of Students	Number of Users
1	No Specific House	0	5
2	Amaterasu	122	12
3	Bishamon	117	11
4	Izanami	114	12
5	Tsukiyomi	118	12

Go Home

Delete House

You have requested to delete the following houses:

ID	Name	Number of Students	Number of Users
3	Bishamon	117	11

Failed to delete house: 3 because there are still users within this house, so delete them first

Updated House list:

ID	Name	Number of Students	Number of Users
1	No Specific House	0	5
2	Amaterasu	122	12
3	Bishamon	117	11
4	Izanami	114	12
5	Tsukiyomi	118	12

Test/Success Criteria

- If the houses were deleted from the database, display to inform the user that it was deleted
  - If selected houses were deleted, display these houses
  - If all houses were deleted, display all the houses
- If the houses couldn't be removed from the table, inform the user that it couldn't be deleted
- Display the updated houses table
  - If all houses were deleted, display an empty table informing the user that there are no houses

Add House page (add_house.php)

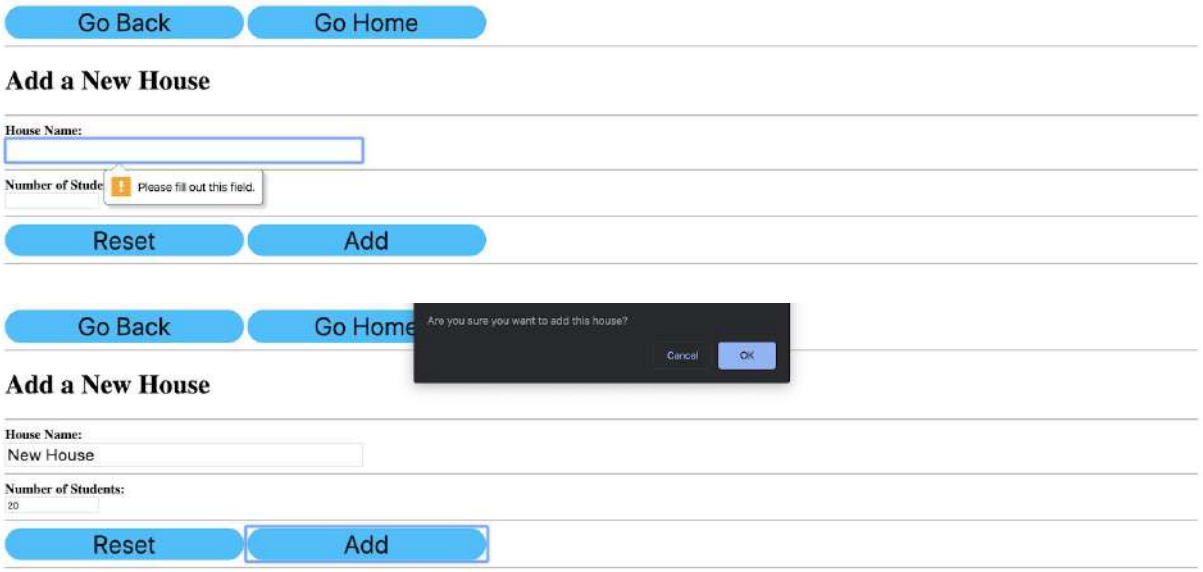
Go Back Go Home

Add a New House

House Name:

Number of Students:

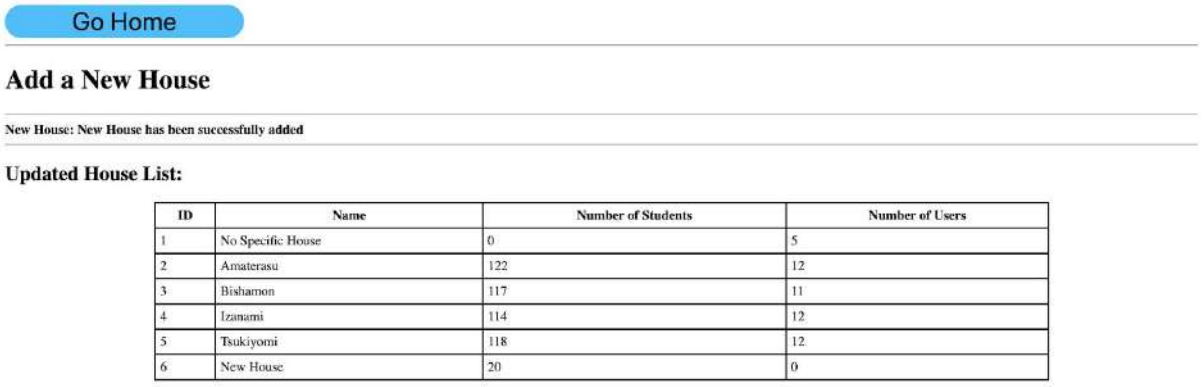
Reset Add



Test/Success Criteria

- Form to allow the user to input the required information for a new house
  - Text box to enter the house name (required)
    - If the text box is left blank and the user tries to add the house, they will be blocked
  - 'Add' button to add the house and proceed to the next page, and a reset button to restart
  - When pressing the add button, a warning message should be displayed to ask the user if they are sure they want to add the house

House Added page (add_house.view.php)

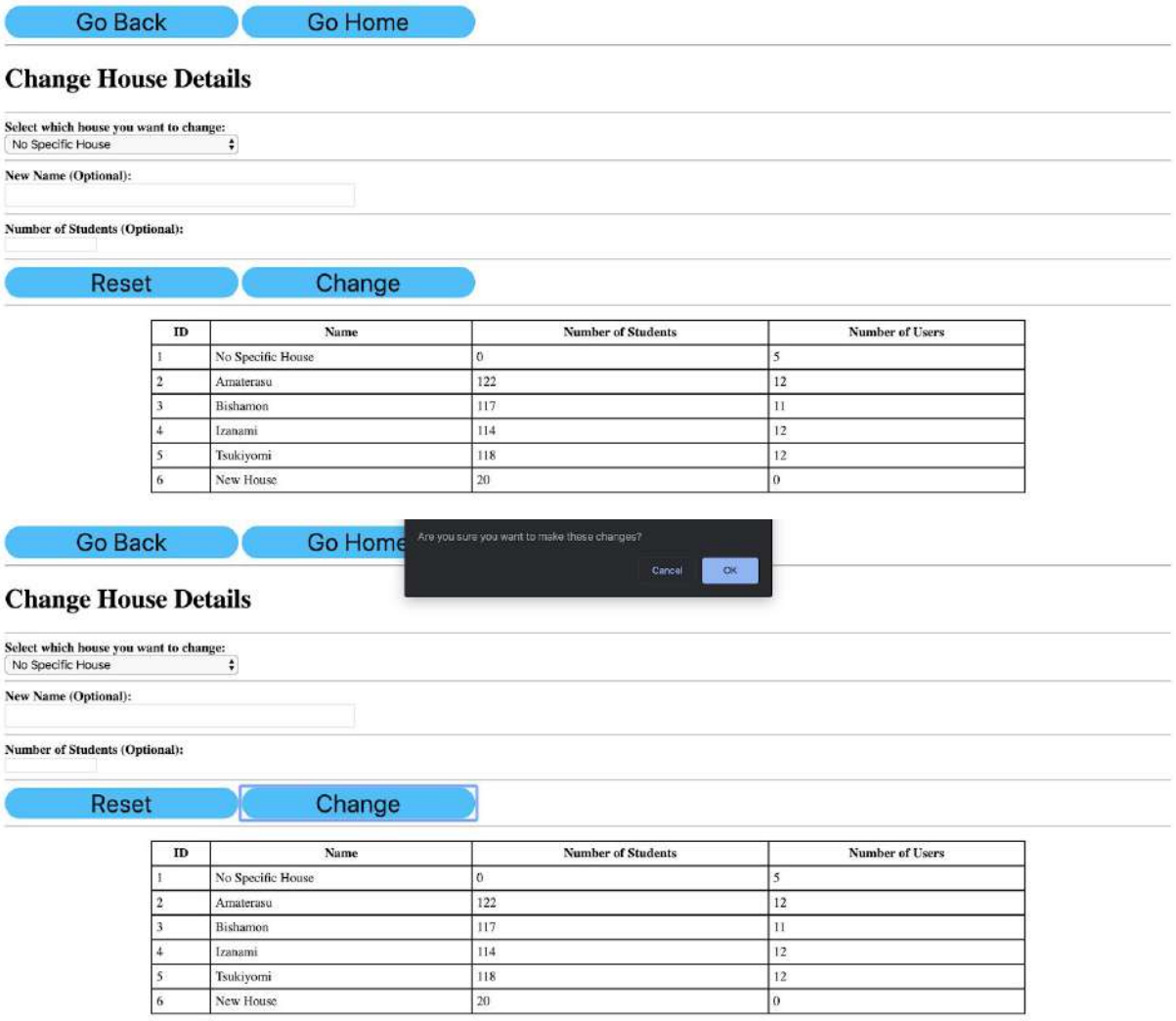


ID	Name	Number of Students	Number of Users
1	No Specific House	0	5
2	Amaterasu	122	12
3	Bishamon	117	11
4	Izanami	114	12
5	Tsukiyomi	118	12
6	New House	20	0

Test/Success Criteria

- If the house was added to the table, display a message informing the user that it was added
- Display the updated houses table

Change House Details page (change_house.php)



Test/Success Criteria

- Form to allow user to select a house and change their details
  - Drop down menu with the houses as options (already selected with a default value)
  - Text box to input the new house name (optional)
  - Number input to enter the new number of students (optional)
  - ‘Change’ button to proceed and a ‘Reset’ button to restart if needed

House Details Changed page (change_house.view.php)

Go Home

Change House Details

You made no changes to the house

Updated House List:

ID	Name	Number of Students	Number of Users
1	No Specific House	0	5
2	Amaterasu	122	12
3	Bishamon	117	11
4	Izanami	114	12
5	Tsukiyomi	118	12
6	New House	20	0

Go Home

Change House Details

House: 6 name has successfully been changed to: The Best House

Updated House List:

ID	Name	Number of Students	Number of Users
1	No Specific House	0	5
2	Amaterasu	122	12
3	Bishamon	117	11
4	Izanami	114	12
5	Tsukiyomi	118	12
6	The Best House	20	0

Go Home

Change House Details

House: 6 student number has successfully been changed to: 50

Updated House List:

ID	Name	Number of Students	Number of Users
1	No Specific House	0	5
2	Amaterasu	122	12
3	Bishamon	117	11
4	Izanami	114	12
5	Tsukiyomi	118	12
6	The Best House	50	0

Go Home

Change House Details

House: 6 name has successfully been changed to: The Worst House
House: 6 student number has successfully been changed to: 30

Updated House List:

ID	Name	Number of Students	Number of Users
1	No Specific House	0	5
2	Amaterasu	122	12
3	Bishamon	117	11
4	Izanami	114	12
5	Tsukiyomi	118	12
6	The Worst House	30	0

Test/Success Criteria

Test/Success Criteria

New Name	New Number of Students	Expected Output
0	0	“No changes”
0	1	“Number of students changed”
1	0	“New Name”
1	1	“Number of students changed” “New Name”

- ❖ If the house’s details have been changed, display a message to inform the user that it has been successfully changed
  - ➢ Display each of the changes made (name and/or number of students
- ❖ If the house’s details couldn’t be changed, display a message to inform the user that the changes couldn’t be made
- ❖ Display a table of the updated houses

View/Edit Year Group Section

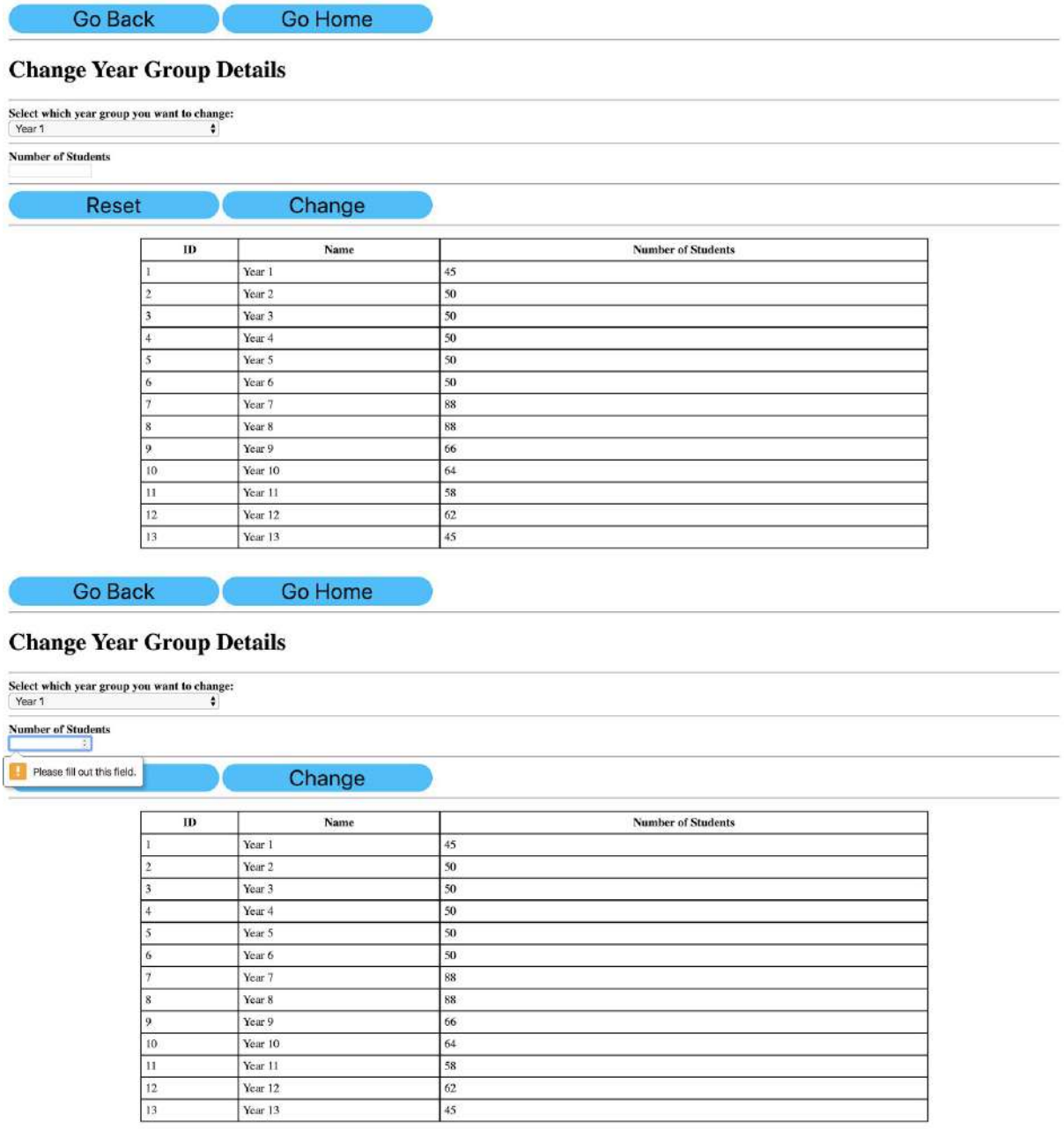
View/Edit Year Group page (edit_yearGroup.php)

ID	Name	Number of Students
1	Year 1	45
2	Year 2	50
3	Year 3	50
4	Year 4	50
5	Year 5	50
6	Year 6	50
7	Year 7	88
8	Year 8	88
9	Year 9	66
10	Year 10	64
11	Year 11	58
12	Year 12	62
13	Year 13	45

Test/Success Criteria

- ❖ All year groups displayed in a table format
  - ➢ Column for ‘Group ID’
  - ➢ Column for ‘Group Name’
  - ➢ Column for ‘Number of Students
- ❖ ‘Make changes to Year Group button to redirect user to edit Year Group page

Change Year Group Details page (change_yearGroup.php)



ID	Name	Number of Students
1	Year 1	45
2	Year 2	50
3	Year 3	50
4	Year 4	50
5	Year 5	50
6	Year 6	50
7	Year 7	88
8	Year 8	88
9	Year 9	66
10	Year 10	64
11	Year 11	58
12	Year 12	62
13	Year 13	45

ID	Name	Number of Students
1	Year 1	45
2	Year 2	50
3	Year 3	50
4	Year 4	50
5	Year 5	50
6	Year 6	50
7	Year 7	88
8	Year 8	88
9	Year 9	66
10	Year 10	64
11	Year 11	58
12	Year 12	62
13	Year 13	45

Test/Success Criteria

- Form to allow user to select a year group and change their details
  - Drop down menu with the year groups as options (already selected with a default value)
  - Number input to enter the new number of students (required)
    - User will be blocked if not filled in
  - ‘Change’ button to proceed and a ‘Reset’ button to restart if needed

Year Group Details Changed page (change_yearGroup.view.php)

Go Home

Change Year Group's Details

Year Group: 1 number of students has successfully been changed

Updated Year Group List:

ID	Name	Number of Students
1	Year 1	50
2	Year 2	50
3	Year 3	50
4	Year 4	50
5	Year 5	50
6	Year 6	50
7	Year 7	88
8	Year 8	88
9	Year 9	66
10	Year 10	64
11	Year 11	58
12	Year 12	62
13	Year 13	45

Test/Success Criteria

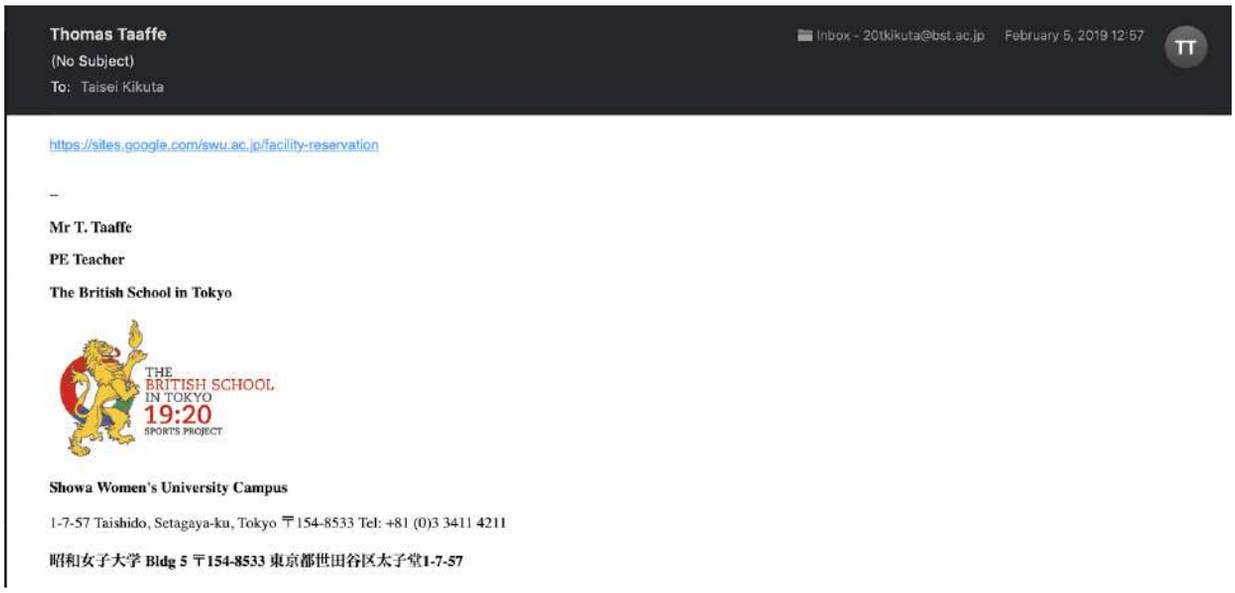
- If the year group's details have been changed, display a message to inform the user that it has been successfully changed
- If the year group's details couldn't be changed, display a message to inform the user that the changes couldn't be made
- Display a table of the updated year groups

Evaluation

After thorough testing and using the product in real world situations, I can conclude that this new system has improved the management and organisation of the PE department.
Appendix I shows the list of positive and negative feedback received from Mr Taaffe, a member of the PE department. As stated, all the positive points mentioned by the PE department were features and functions that were requested originally and part of the success criteria. For example, the simple to use and understand user interface and the ability to view all the useful information including facilities, equipment and users.
In terms of the negative points and factors to improve, much of the advice given was based on making the system more clear to the user. Firstly, he suggested I should include hyperlinks or popups for each piece of equipment to display an image of it, or their specific details so it would be easier to identify when adding or viewing a booking. Secondly, he would have liked to have seen an additional functionality in the add booking page to allow the user to select a specific number of each equipment rather than having to book the whole collection. This could definitely be a new feature I could add in the future, but it would require additional code and SQL queries that would change the 'numAvailable' column in the Equipment table each time. The third point was to have an option to be able to automatically send a confirmation email to the user after the booking was made, outlining the details mentioned in the 'booking_confirm.php' page. The final request was to just change the 'Clear' button to something else such as 'Delete All'. This button can be found on all pages where bookings or data can be deleted. Mr Taaffe found it quite confusing to have the 'Clear' button as he thought it would reset the form (which is what the 'Reset' button does) instead of deleting all records in the table. This is something that can be changed easily just by making some changes to the HTML code.
In addition to the advice given by the PE department, I personally had a few suggestions for myself that I could add next time if I were to do it again, or do something similar again. Firstly I would try to make the system more complex and advanced by using the Google Calendar API to tie in with the 'Calendar View' option. This option could automatically add the booking to the teacher's Google Calendar and also display this Google Calendar on the 'Calendar View' page.
In terms of making the programming and developing stage more simple, I could have made sure to write out and plan the OOP code earlier. This is because initially, I planned to write the whole solution in POP, and only half way through the process I decided to write it in OOP. Therefore the process of converting the code from POP to OOP was very time consuming and if I had started off with OOP, I could have saved lots of time.

Appendix

Appendix A



Thomas Taaffe
(No Subject)
To: Taisei Kikuta

Inbox - 20tkikuta@bst.ac.jp February 5, 2019 12:57

TT

https://sites.google.com/swu.ac.jp/facility-reservation

--

Mr T. Taaffe

PE Teacher

The British School in Tokyo

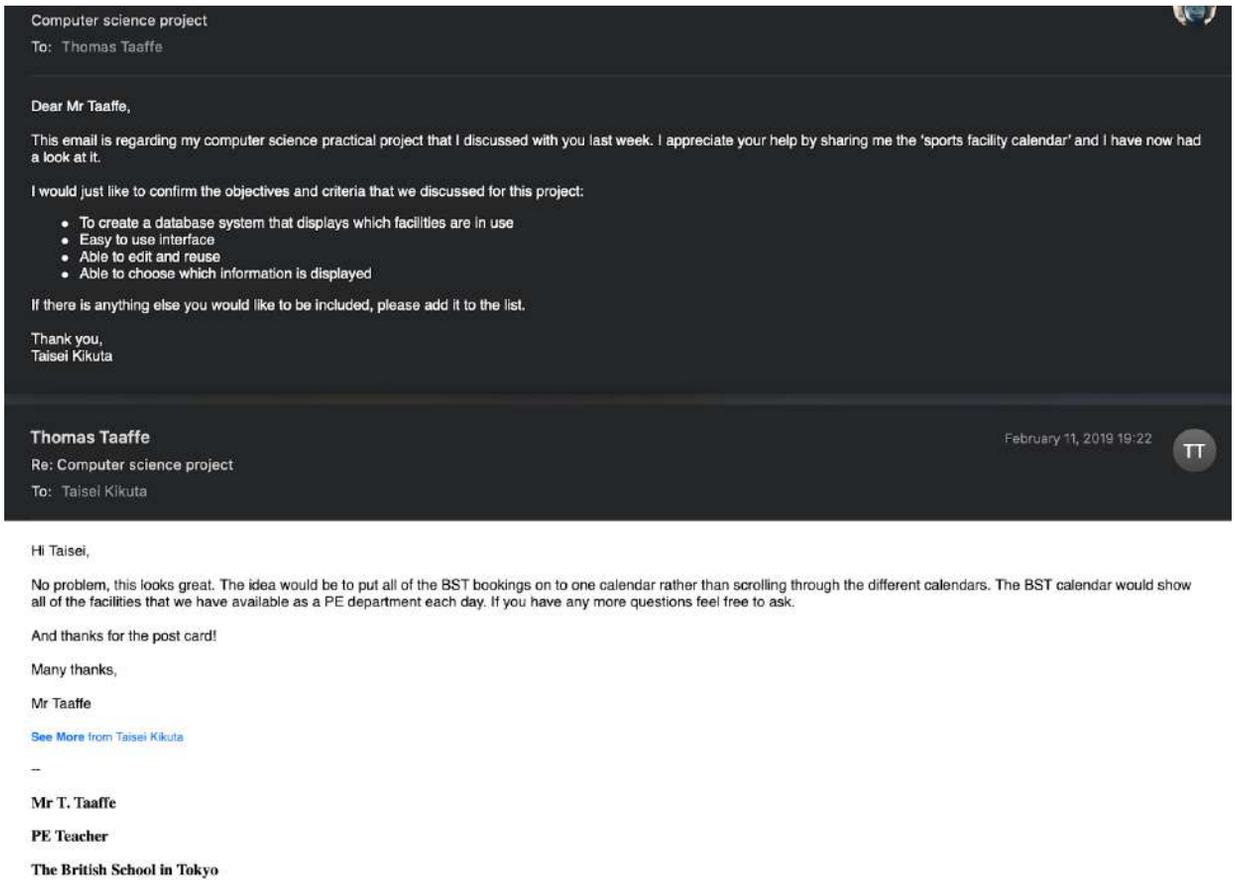
THE BRITISH SCHOOL IN TOKYO 19:20 SPORTS PROJECT

Showa Women's University Campus

1-7-57 Taishido, Setagaya-ku, Tokyo 〒154-8533 Tel: +81 (0)3 3411 4211

昭和女子大学 Bldg 5 〒154-8533 東京都世田谷区太子堂1-7-57

Appendix B



Computer science project
To: Thomas Taaffe

Dear Mr Taaffe,

This email is regarding my computer science practical project that I discussed with you last week. I appreciate your help by sharing me the 'sports facility calendar' and I have now had a look at it.

I would just like to confirm the objectives and criteria that we discussed for this project:

- To create a database system that displays which facilities are in use
- Easy to use interface
- Able to edit and reuse
- Able to choose which information is displayed

If there is anything else you would like to be included, please add it to the list.

Thank you,
Taisei Kikuta

Thomas Taaffe
Re: Computer science project
To: Taisei Kikuta

February 11, 2019 19:22

TT

Hi Taisei,

No problem, this looks great. The idea would be to put all of the BST bookings on to one calendar rather than scrolling through the different calendars. The BST calendar would show all of the facilities that we have available as a PE department each day. If you have any more questions feel free to ask.

And thanks for the post card!

Many thanks,

Mr Taaffe

See More from Taisei Kikuta

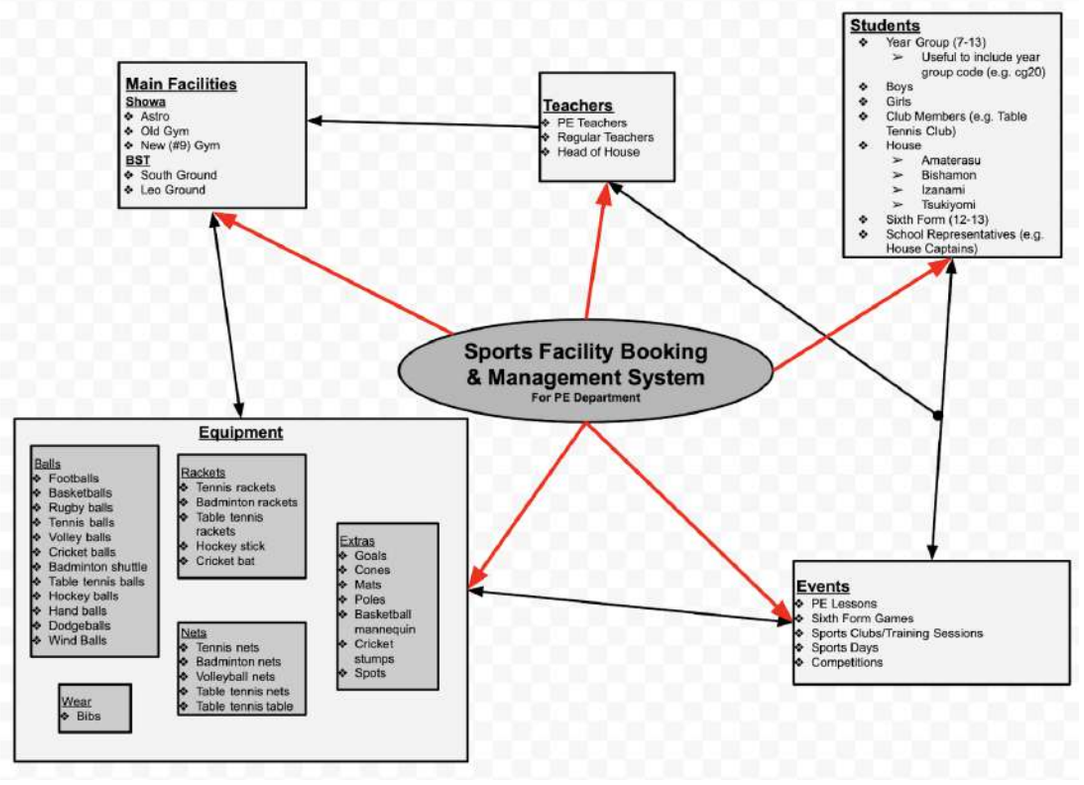
--

Mr T. Taaffe

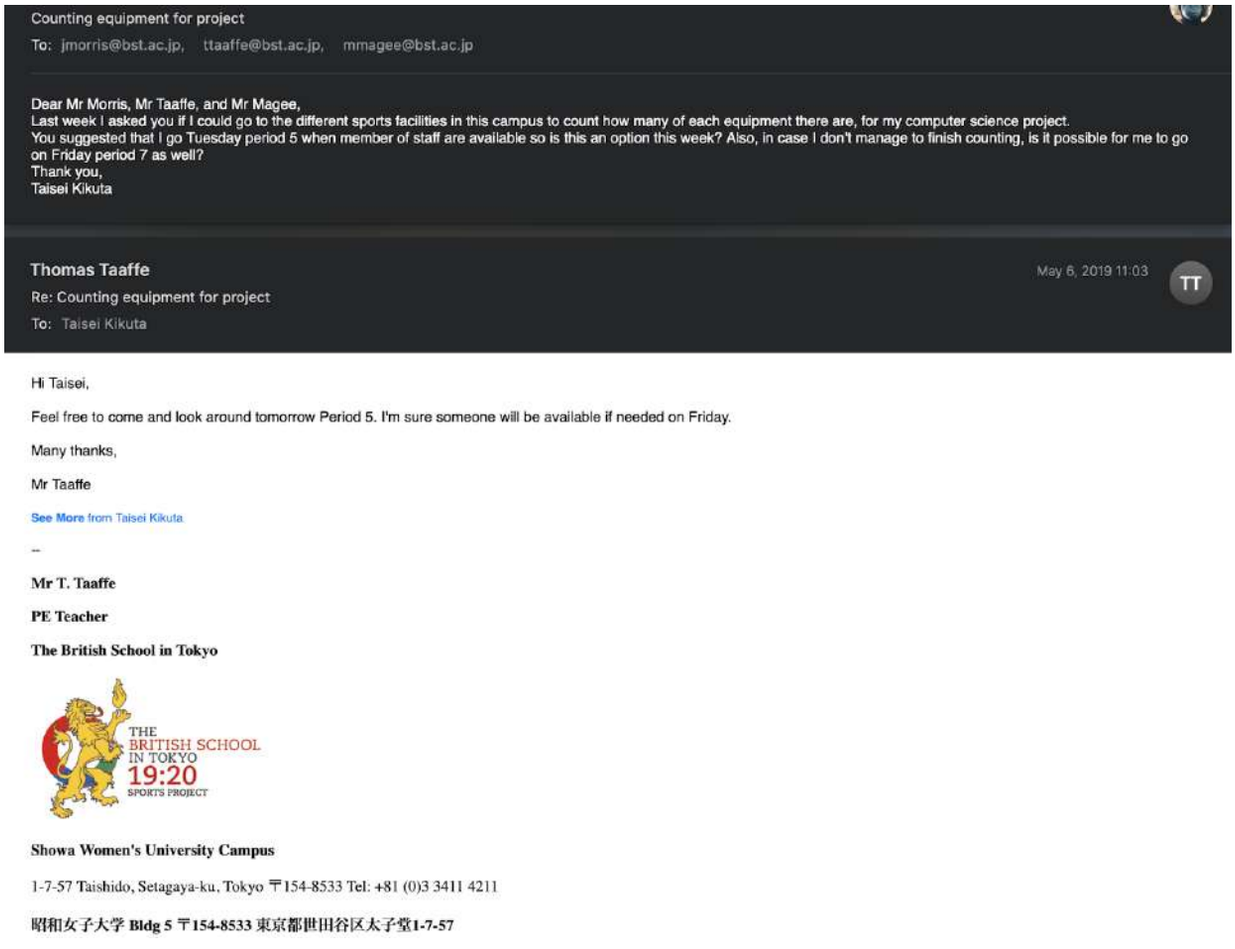
PE Teacher

The British School in Tokyo

Appendix C


Main Facilities
Showa
Astro
Old Gym
New (#9) Gym
BST
South Ground
Leo Ground
Teachers
PE Teachers
Regular Teachers
Head of House
Students
Year Group (7-13)
Useful to include year group code (e.g. cg20)
Boys
Girls
Club Members (e.g. Table Tennis Club)
House
Amaterasu
Bishamon
Izanami
Tsukiyomi
Sixth Form (12-13)
School Representatives (e.g. House Captains)
Sports Facility Booking & Management System
For PE Department
Equipment
Balls
Footballs
Basketballs
Rugby balls
Tennis balls
Volley balls
Cricket balls
Badminton shuttle
Table tennis balls
Hockey balls
Hand balls
Dodgeballs
Wind Balls
Rackets
Tennis rackets
Badminton rackets
Table tennis rackets
Hockey stick
Cricket bat
Nets
Tennis nets
Badminton nets
Volleyball nets
Table tennis nets
Table tennis table
Extras
Goals
Cones
Mats
Poles
Basketball mannequin
Cricket stumps
Spots
Wear
Bibs
Events
PE Lessons
Sixth Form Games
Sports Clubs/Training Sessions
Sports Days
Competitions

Appendix D



Counting equipment for project

To: jmorris@bst.ac.jp, ttaaffe@bst.ac.jp, mmagee@bst.ac.jp

Dear Mr Morris, Mr Taaffe, and Mr Magee,
Last week I asked you if I could go to the different sports facilities in this campus to count how many of each equipment there are, for my computer science project.
You suggested that I go Tuesday period 5 when member of staff are available so is this an option this week? Also, in case I don't manage to finish counting, is it possible for me to go on Friday period 7 as well?
Thank you,
Taisei Kikuta

Thomas Taaffe May 6, 2019 11:03

Re: Counting equipment for project

To: Taisei Kikuta

TT

Hi Taisei,

Feel free to come and look around tomorrow Period 5. I'm sure someone will be available if needed on Friday.

Many thanks,

Mr Taaffe

See More from Taisei Kikuta

--

Mr T. Taaffe

PE Teacher

The British School in Tokyo

THE BRITISH SCHOOL IN TOKYO 19:20 SPORTS PROJECT

Showa Women's University Campus

1-7-57 Taishido, Setagaya-ku, Tokyo 〒154-8533 Tel: +81 (0)3 3411 4211

昭和女子大学 Bldg 5 〒154-8533 東京都世田谷区太子堂1-7-57

Appendix E

To: abickley@bst.ac.jp

Dear Mr Bickley,

I am currently studying computer science as one of my A-Level courses.
For the course, we must complete a practical project where we find a problem and try to develop an advanced solution for it. For reference, my project is focusing on the school sports facilities and equipment, and developing a booking system and database for better organisation and management. Currently, I am working on creating tables for my database and entering important data such as equipment types.
For one of the tables, I am looking to include 'number of students in each year group', 'number of teachers in each house', and 'teacher names'. I believe you may have this information as a regular user of iSAMS and the registration system.

If possible, could you send me:

- Number of secondary students in each year group
- Number of secondary teachers in each house
- List of all secondary teachers' names

I understand that this may be a significant amount of information, so just a screenshot will be enough. And if some of this information is unavailable to you, please inform me about who else to contact.

Thank you,
Taisei Kikuta, Year 12

Andrew Bickley May 15, 2019 10:43 AB

Re: Information for computer science project

To: Taisei Kikuta

Dear Taisei,

No problem. The information you need is below:

Students per year group:

SCHOOL NO.	STATISTICS
Vertical ...	0
Nursery	35
Reception	76
Year 1	76
Year 2	86
Year 3	88
Year 4	84
Year 5	88
Year 6	84
Year 7	88
Year 8	88
Year 9	66
Year 10	64
Year 11	58
Year 12	62
Year 13	45
Totals	1088

Number of secondary teachers in each house:

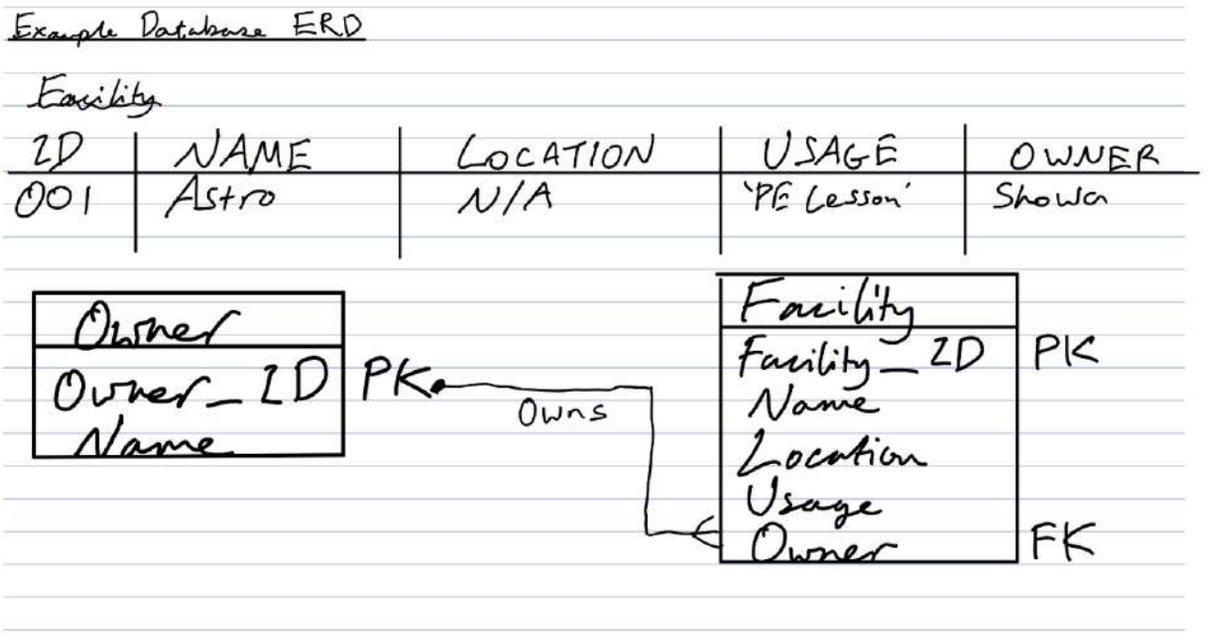
Amaterasu - 12 staff
Bishamon - 12 staff
Izanami - 12 staff
Tsukiyomi - 12 staff

Staff in Secondary:

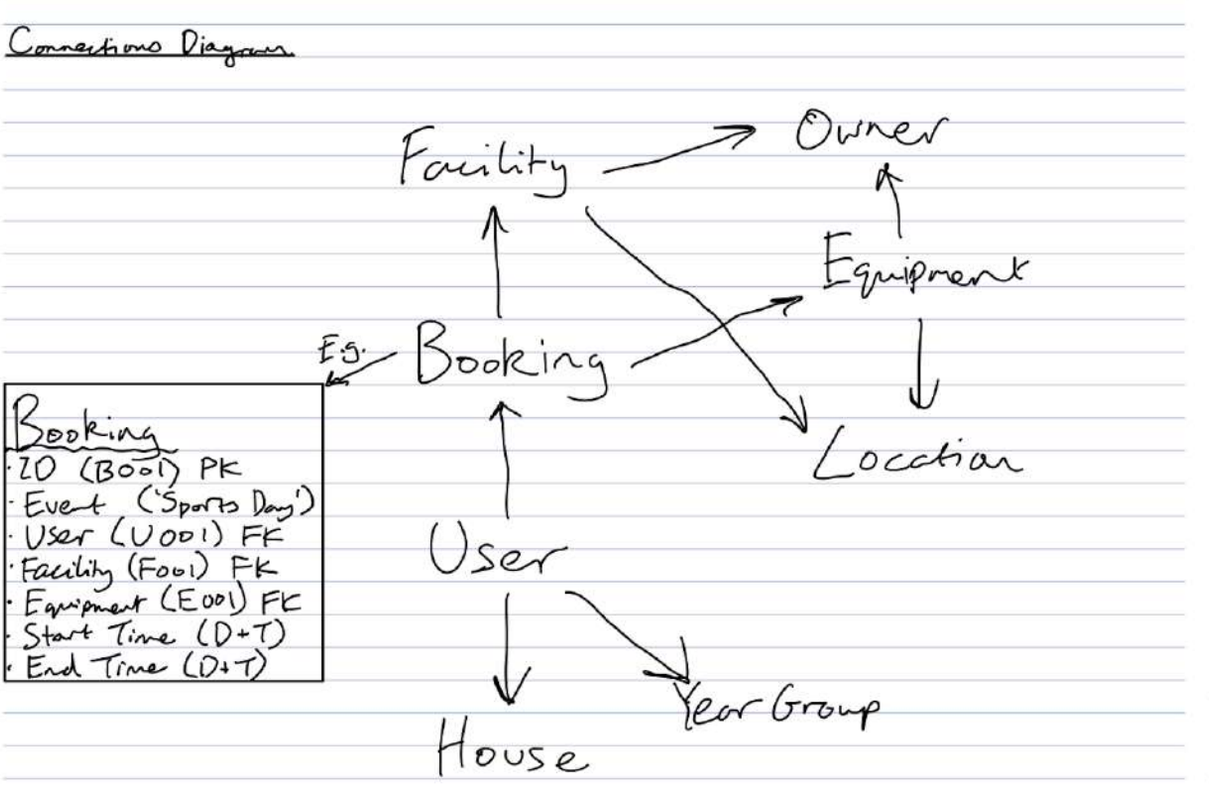
A1 – Mr Close; Ms Cerra
A2 – Mr Grimshaw; Mr Bell
A3 – Mrs Sadou; Mr Madine
A4 – Mr Hazzard; Ms Yamanouchi
A5 – Ms Kaneta; Mr Warburton
A6 – Mr Keeble-Watson; Ms Yoshizawa
B1 – Mr Pye; Mr Bramwell
B2 – Mr Spicer; Ms Obata
B3 – Mr Prowse; Miss Woods
B4 – Mr Balcombe; Mr Magee
B5 – Mr Paterson; Mr Bickley
B6 – Mr Parsons; Ms Nakasuka
I1 – Mr Beston; Ms Wery
I2 – Mrs Yamada; Mr Sewell
I3 – Ms Marcouse; Mr Taaffe
I4 – Mr Thomas; Mr Travis
I5 – Mr Wilkie; Ms Kikkawa; Mr Fellerman
I6 – Mr Fraser; Mr Knill
T1 – Mr Whybro; Mr Groarke
T2 – Mrs Bickley; Mr Meguro
T3 – Mr Speller; Mr Liebman
T4 – Mr Reid; Mr Grey
T5 – Dr McDarby; Mr Morris
T6 – Dr Matsumiya; Mr Price
Plus Ms Trachonitis, Ms Nectias, Ms Scott, Ms Coates-Jones

Hope that helps,
Mr Bickley

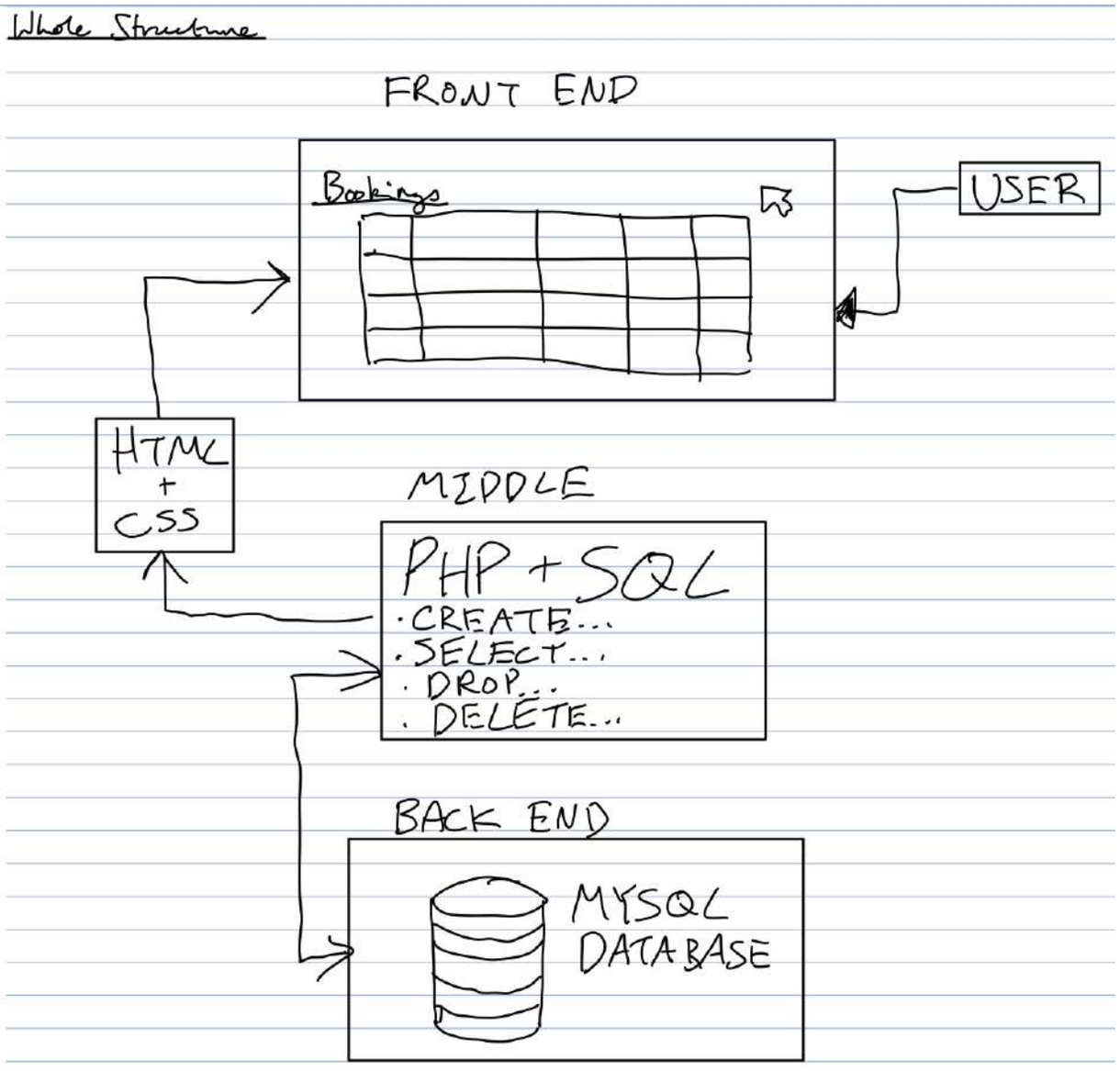
Appendix F



Appendix G



Appendix H


Whole Structure
FRONT END
Bookings
USER
HTML
+
CSS
MIDDLE
PHP + SQL
·CREATE...
·SELECT...
·DROP...
·DELETE...
BACK END
MYSQL
DATABASE

Appendix I

Positives

- Extremely easy to use and to navigate.
- Easy to see basic information, nothing missing.
- Easy to view and filter information
- Calendar view allows you to see all bookings over the month
- Easy to edit owners, facilities and other data

Points for improvement

- Pictures to allow users to see different facilities and available equipment
- Ability to select different quantities of equipment
- Email sent to user with confirmation of their booking
- 'Clear' button slightly confusing.

I hope this helps. Great work!

Thanks
--

Mr T. Taaffe

PE Teacher

The British School in Tokyo

THE BRITISH SCHOOL IN TOKYO 19:20 SPORTS PROJECT

Showa Women's University Campus

1-7-57 Taishido, Setagaya-ku, Tokyo 〒154-8533 Tel: +81 (0)3 3411 4211

昭和女子大学 Bldg 5 〒154-8533 東京都世田谷区太子堂1-7-57