

3rd Year Physics Lab Booking System

Instruction Manual

User Documentation

Prepared Exclusively for: Dr. Xue (Sue) Yang

by

Christopher Ballouz,

Julio Deak,

Mohammad Heydari,

Albert Hyde,

Kartik Gupta,

Puravin Sivaganam

COMP3615, School Of Information Technologies, University of Sydney

CONTENTS

| | |
|-----------------------------------|----------|
| INSTRUCTION MANUAL..... | 1 |
| DOWNLOAD THE APPLICATION | 1 |
| SYSTEM SETUP INSTRUCTIONS..... | 1 |
| ADMINISTRATOR SETUP | 2 |
| STUDENT BOOKINGS | 3 |
| WORK COMPLETED | 4 |
| ADMINISTRATOR FUNCTIONALITY | 4 |
| STUDENT FUNCTIONALITY | 12 |
| ADMINISTRATOR AND STUDENT | 17 |

Instruction Manual

Download The Application

1. Go to Downloads on the Physics Lab Booking Project Page on BitBucket
2. Click on “Download Repository”

System Setup Instructions

1. Open up a terminal window and go to the directory that holds the application from the downloaded repository
2. Setup up the gems and bundler (dependency management system)

```
$ sudo gem install bundler
```

```
$ sudo bundle install
```

3. Load up the schema and the seeds file using the following command:

```
$ bundle exec rake db:setup
```

4. Start the server using the following command:

```
$ rails server
```

5. Direct to opened port on web browser

```
http://localhost:3000
```

ADMINISTRATOR SETUP

1. Login into admin
2. Define the calendar specifications for the current year and semester if not altered on the seeds file
 - Select the Calendar tab on the navigation bar
 - Start date (for semester and semester break)
 - End date (for semester and semester break)
 - Click Save
3. Add students using a .csv file or by manually adding one student each time.
 - Select the Students tab on navigation bar. On the Students page, below the table, there are options to add a “new student” or “import Student List”
4. Add the experiments manually by defining each experiment’s details
 - Select the Experiments tab on the navigation bar. On the Experiments page, below the list of experiments on the table, there is an option, “New Experiment”, to select below the table.
5. Define the available dates that an experiment(s) is available during the semester
 - Select the Calendar tab on the navigation bar
6. The experiments can have different dates indicating their availability
 - Toggle an experiment from the list of experiments on the right side of the calendar. Click on the set button and then choose the dates on the calendar that the experiment will be available.

STUDENT BOOKINGS

1. Login into Student with one of the accounts created
2. Make a booking on the Students page
 - Select Book Experiment on the navigation bar. Information will be provided regarding the number of sessions the student can make and the experiments that can be booked.
3. Choose an experiment to book
 - Click book experiment next to the specific experiment on the list and then the page with the calendar to select the dates for the bookings will be shown
4. Choose to book an experiment with two sessions over two weeks or over a week
 - Toggle “1 session per week” or toggle “2 sessions per week”
5. Choose to book an experiment with four sessions over four weeks, one session per week, or over two weeks, two sessions each week.
 - Toggle “1 session per week” or toggle “2 sessions per week”
6. Choose the dates on the calendar that are marked in green
 - Toggle the check box, if visible, for the particular dates and then select “Book Experiment”
7. View Bookings made from the Students page
 - Select “Bookings Summary” from the navigations bar and there will be a table displayed showing the experiments booked by the student

Work Completed

Administrator functionality

Task:

Fix the bug in adding new student

User Story:

As an admin, I want to be able to add students with their personal details.

Description:

Administrator of the system needs to create students based on their personal details i.e. SID, name, email address, and password. SID and email need to be unique and they need to follow a pattern to match up.

Changes made:

- app/mailer/system_mailer.rb
 - Added new email address to be used for email notification.
- config/environments/development.rb
 - Made changes regarding the new email address and the configuration about gmail mailing service
- app/models/student.rb
 - Added a simple regular expression for email validation.
- spec/models/student_spec.rb
 - Unit tests for student model has been done

How to implement it:

1. Login as an Administrator
2. Select "Students" from the navigation bar
3. click on 'New Student'
4. A form will be shown with empty field for student's details.
5. Enter the required fields, and hit 'Create Student' button.
6. Another page will be shown with newly created student's detail.

Task:

Fix the bug in import student list

User Story:

As an admin, I want to import/upload a list of students

Description:

Administrator can upload a csv formatted file, based on a predefined format, to upload a list of students. The CSV file must follow the format that is being described in the relevant view; otherwise an error will be shown. Each student in the list will then be created and saved into the database. A Confirmation email will be sent to each student to indicate that they have been added to the system.

Changes made:

- app/controllers/students_controller.rb
 - I added the changes to write the selected year and semester to the system cache, so it can be used in other system business logic such as the csv_import_controller file.
- app/controllers/csv_import_controller.rb
 - I read the save data in the system cache to allow the entries in the list to be added based on those values set by the admin.
- /app/mailer/system_mailer.rb
 - Made changes to include more information for the new user
 - Because of the previous changes, I had to change the 'system_mailer' methods call to conform with new methods signature.
- app/views/csv_import/student_list.html.erb
 - I changed the view to give the user more information about the format and requirements of the csv file. Changes were made to the html, by adding the information necessary and highlighting the text in the view to indicate it's importance

How to implement it:

1. Login in as an administrator
2. Click on import file
3. Click on 'Import Student List'.
4. If user wants to import the list for specific year and semester. (S)He needs to select the year and semester on top of the page and click on 'List' then click 'Import

Student List'

5. `http://{ local host:port number}/csv_import/student_list` page will be showed up, which gives user some information about the student list.
6. User click on 'Browse...' to find the file (s)he wants. Another window opens up to let user to find the file.
7. After finding the file user can close the window and (s)he goes back to previous page.
8. Now user needs to click on 'Upload CSV' button.
9. System evaluates the entries in the list and creates an account for each entry and send email notification for each student's email address.
10. If successful, system shows successful message on top of csv import page.

Task:

view -- list of experiments and a calendar. Select experiments and days to set up calendar for specific experiments

User Story:

As an admin, I want to view the list of experiments side by side with the calendar

Description:

Administrator is able to view the list of all experiments next to the Calendar.

Changes made:

- `views/calendars/index.html.erb`
 - Changed the style of calendar to also have “float:left”. This will align the calendar to the left of the page.
 - Added new experiment table. The table will list all experiments name with a radio button next to it.
 - Within experiment table, there is a submit button named ‘set’ to send a post variable, storing the checked experiment’s id.
- `controllers/calendar_controller.rb`
 - At method `index`, added new instance variable called `experiments`. This variable stores all the experiments and passes the information to the calendar view.
 - Added a new method called `update_experiment`. The method is invoked when the submit button ‘set’ in calendar page is pressed. It will catch the post request and store the checked experiment’s id in the global class variable `@@exp_selected`. When post request was unsuccessful, error notice will come up stating “Failed Request”.

How to implement it:

1. Login as administrator
2. Navigate to Calendar page.
3. If there are any experiments, you should see the names of experiments within the table next to the calendar.

Task:

Setup dates for available experiments in current semester. Different experiments may have different calendars (different days). In current system, all experiments use the same calendar

User Story:

As an admin, I want to select dates for each experiment in the current semester

Description:

The administrator is able to view the experiment list side by side with the calendar and select the availability for each experiment individually or combined

Changes made:

- controllers/calendars_controller.rb
 - Added global class variable: @@exp_selected to store the selected experiment by the administrator when setting availability through the function setAvailable
 - In the Index method: Added instance variable @display_experiment which will store the selected experiment.
 - In the toggle_calendar method: Changed to toggle only the selected experiment.
 - In the update_experiment method: Created the method to handle the post variable sent whenever the Submit button in the Experiment table is pressed. The method reads post variable and stores the experiment id to @@exp_selected
 - When no experiment is selected, raises a notice 'Please select experiment!'
- views/calendars/index.html.erb
 - Added a new table (experiment table). The table lists the names of all created experiments. A radio button has been added next to each name and can be selected. The table contains a submit button at the bottom that when clicked will send post variables stored with the particular experiment's id that refer to the experiment checked
 - The style of the calendar table was modified so it floated left of the page

- models/calendar.rb
 - The methods now takes in one more argument “experiment_id”
- config/routes.rb
 - Added the line “/calendars/update_experiment” to tell rails where the post request for update_experiment is being handled
- db/schema.rb
 - A new attribute, “experiment_id”, was added to the calendar table in the database

How to implement it:

1. Login as an admin
2. Go to the calendar’s page
3. The page will show a list of experiments on the right side of the calendar. Toggle (check) the experiment or experiments that want to be selected to modify their availability.
4. Click on the “set” button below
5. Now select the dates on the calendar that will be marked as available for these particular experiments

Task:

Admin is able to make and delete any bookings at any time

User Story:

As an admin, I want to be able to cancel bookings at any time for enrolled students.

Description:

As issues can arise during a semester an admin should be able to cancel any booking made without having to worry about changing any settings. The admin level then over rules any other conditions in place and allows them to cancel the booking.

Changes made:

- app/views/make_booking/view_experiment.html.erb
 - Added in a check when cancelling a booking that checks wither the current role is the admins and, if true does not check any other logic and removes the booking.

How to implement it:

1. Login as admin

2. Change the settings so if a student were to cancel he wouldn't be allowed
3. Make a booking for a student
4. View the student booking, you have the option to cancel it even when the student does not have the ability to do the same

Task:

Send emails [when a booking is made or cancelled] to affected students for notification of the changes

The emails sent out will have a backup in the system and be viewed as required. It displays what experiments have been booked and done so far for each student.

User Story:

As an admin, I want email notifications sent when making/cancelling a booking

Description:

Email notification needs to be sent whenever the administrator makes a booking or cancels a booking for a student. Email notification will be sent to the student's email address that has been registered in the system.

Changes made:

- app/mailers/system_mailer.rb
 - Added the following two methods: 'cancelBookingEmail' and 'makeBookingEmail'. Gets the inputs for the booking and sends it to the mailer.
 - Added the relevant view for the two newly created methods. These are the following views: 'cancelBookingEmail.html.erb' and 'makeBookingEmail.html.erb'. The views contain the email content.
- app/controllers/make_booking_controller.rb
 - Added the methods calls to newly created methods in 'system_mailer.rb', whenever admin make/cancel a booking. So now email notification will be sent whenever a booking is being made or cancelled.

How to implement it:

1. Login in as an administrator
2. Go to the Students tab

3. Click on 'Book experiment' for a particular student
4. Then make a booking or cancel a booking
5. Check your email once the transaction is confirmed and an email will contain the booking details

Task:

Admin is able to define parameters X [days to cancel a booking].

User Story:

As an admin I can set days prior to when a student cancels booking.

Description:

To prevent students from constantly cancelling bookings and the like, the admin can impose a deadline on how many days before a booking a student can no longer cancel. This can be changed at any time by an admin and will prevent a student from cancelling a booking within that number of days from then on.

Changes made:

- app/views/settings/index.html.erb
 - Changed the wording so to better reflect what the function on the system page did
 - Adding in some code so that it is no longer a blank form but one that only allows certain inputs, eg between 100 days and 1 day

How to implement it:

1. Go to setting page
2. Change the number in the day's allowed box
3. Hit save
4. A Success message should then be displayed
5. Now if a student tries to cancel a booking to late, the delete bookings option should be greyed out

Task:

View bookings Summary [for students]

User Story:

As an admin, I want to view the experiments booked and completed for each student

Description:

Administrator wants to know each Students state for each experiment. The state can be “Nothing’ if the student has not booked for that experiment, “Booked” if the student has booked for that experiment or “Complete” if the booking date has been completed.

Not integrated into user story: information about total free sessions, total required sessions, and total booked sessions

Changes made:

- controllers/make_booking_controller.rb
 - In index method, added the code to retrieve all bookings made by the user and set each experiments relative status to instance array variable @experiment_status.
- views/make_booking/index.html.erb
 - Added if statement to distinguish administrator and students. If logged in as admin, new row in the table with header “Status” appears. The row will list each experiments status.

How to implement it:

1. Login as administrator
2. Navigate to Students
3. Select ‘book experiment’ for particular Student
4. You will see table of Students experiment detail. The table should include row with header named “Status”, listing the status of all available experiments.

Student Functionality

Task:

Student booking should be made evenly across the semester.

Constraints not implemented by previous team:

1. Students must complete one experiment before starting another experiment.
2. Students are not allowed to do two different experiments in one week.

User Story:

As a student, I can make bookings evenly across the semester

Description:

The student should not be allowed to make a booking for another experiment in one week and experiments should only be booked after another. All these changes will not affect the administrators ability to make bookings.

Changes made:

- `app/controllers/make_booking_controller.rb`
 - Added a condition to look at each of the students bookings and compare to the other dates that have been booked. If there is a conflict then raises a warning and does not make the booking.
 - Check to ensure that this only applies to bookings made by students and not administrators

How to implement:

1. Login as a student
2. Navigate to Book experiment
3. Select the experiment
4. Make a booking on the calendar and if the booking is made on the same week as another booking then there will be a warning provided and a booking will not be made

Task:

Make online bookings for the experiments, based on their required CPs

Students are not allowed to over-book sessions (eg. With 4cps, students cannot book over 10 sessions).

User Story:

As a student, I want to book experiments, based on my credit points

Description:

Information is provided to the student about the max number of session bookings possible and the number of sessions already booked. Using the information, the student understands the constraints of their credit point requirements and will be provided a message depending on whether the booking was successful or over the max number of sessions possible.

Changes made:

- stylesheets/lab3b.css.scss
 - Added three new classes (info, data, and data-num) to improve the presentation of the information displayed on the “book experiment” page
- controllers/make_booking_controller.rb
 - Switched the maximum number of experiments booked to the maximum number of sessions by multiplying the amount of credit points for the student by two
 - Modified the “book experiment” trigger from checking the number of bookings in the week to check the student’s current experiment booking will not go over their maximum number of sessions.
 - A warning is raised if the number of sessions is exceeded and information is provided to the student in the following manner:
 - “Unable to make booking as it exceeds max number of sessions available. Please contact the administrator”.
 - The booking is not saved when a warning is raised.
- views/make_booking/index.html.erb
 - Added the CSS classes to modify the presentation of the information about the student’s bookings
 - Removed the information that showed the maximum number of experiments booked in this week and the number of experiment required.

- Added information on the number of sessions booked and the maximum number of sessions for a particular student

How to implement it:

1. Login as a student
2. Navigate to the “book experiment” page
3. View the information about your current bookings
4. Choose an experiment and select it
5. Make a booking for the experiment
6. If the number of sessions are within the limit then a success message will be shown otherwise a warning message will be displayed in red showing that the number of sessions have been exceeded

Task:

Each student can only do a single experiment session at a time (in booking calendar for each experiment, the days should be white, which are already booked by others for the same experiment and which are already booked by this student for other experiments).

User Story:

As a student, I can only make one experiment booking session per day

Description:

The student is able to identify the dates that they have made a booking for another session.

Changes made:

- stylesheets/lab3b.css.scss
 - Modified the colour scheme of the navigation bar and header
 - Enabled the scroll bar to be visible for the navigation section
 - Configured the colours for the calendar and switched it to red, green and white
 - Modified the font for the calendar
- controllers/make_booking_controller.rb
 - Switched the @semester_booking variable to retrieve the data for the number of bookings made by the student for each experiment available
 - Changed queries to reflect the correct number of sessions booked and number of bookings made

- Added the variable `@bookings_exp` to get all the bookings made by the student for the current experiment selected from the “book experiment” page. It ensures the student is only able to delete the session booked for the current experiment selected.
 - Configured a loop to cycle through all the bookings and flag the bookings made by the student for the currently selected experiment. There is a condition added that checks in the database if the booking date matches up with the selected experiments bookings. All bookings made by a student are shown in white for all sessions booked.
 - Added a loop to cycle through all the bookings made by other students for a particular experiment chosen by a student. These booked dates are then shown in red and will not be available for the student making a booking
 - Added a control statement that loops through the dates selected and checks if there is a booking by the student on that particular date. Raises a warning when a booking is already made on the selected date(s).
- `views/calendars/index.html.erb`
 - Added the CSS class to modify the presentation of the calendar so the colours are different and the hyperlink modifications are removed to each cell on the calendar
 - `views/make_booking/view_experiment.html.erb`
 - Used the `@bookings_exp` variable to allow the student to delete only the currently selected experiment’s sessions booked

How to implement it:

1. Login as a student
2. Go to the “Book experiment” page
3. Select an experiment
4. Observe the colour codes and the dates that are available, unavailable and booked and distinguish between the white cells, booked by the student, and red cells, booked by another student.
5. Now a student is able to identify when they have made a session booking for particular dates.

Task:

Students should be allowed to cancel the booking they made, but only eg. more than X (2/3/4?) days before the booked dates.

User Story:

As a student, I want to be able to cancel a booking a certain number(X) days before the booked date

Description:

Administrator selects certain number (say this number is X). Students can make a cancellation if the day is X days before the booked date.

Changes made:

The feature can be found in “controllers/make_booking_controller.rb” at view_experiment method and in “views/make_booking/view_experiment.html.erb” at line 80~100.

How to implement it:

1. Login as student
2. Go to book experiments and select any experiment
3. If the experiment is booked and if today’s date is more than X days before the booked date then the “Delete Bookings” appears below the calendar. When pressed on the button, the experiment becomes unbooked.
4. If the experiment is booked and if today’s date is equal or less than X days before the booked date then the “Delete Bookings” button does not appear and instead, notice appears stating the Student cannot cancel the booking to this experiment.

Task:

[Students are] not allowed to cancel the booking (cancellation must be done via admin).

User Story:

As an admin, I want to disable the option for students to cancel a booking

Description:

Given I am an admin, when I tick the checkbox to either say YES or NO to 'Allow Students to cancel bookings', it will assign the "Days prior to bookings" a 5 or 0 respectively.

Changes made:

- app/views/settings/index.html.erb
 - A check box has been added that either allows or disallows students to cancel their booking. Yes and No both have their individual values.
- app/views/make_booking/view_experiment.html.erb
 - A new if statement has been integrated that takes the new value that was assigned from the check box to cancel or not to cancel the booking. The other

way to not allow bookings to be cancelled was already implemented which was "Days prior". Thus a new message would be shown when the booking is not allowed to be cancelled.

How to implement it:

1. Create a Student
2. Create an experiment
3. Assign him/her to the experiment
4. Allocate the calendar dates
5. Go to the settings page tick NO to disallow students to cancel bookings.
6. When a student goes back to the experiment and tries to cancel the booking, it is faded out and a message appears "You are not allowed to cancel a booking."

Administrator and Student

Task:

There is a bug in make_booking of current system

Description:

The bug caused issues for all the tasks related to making and deleting a booking.

Changes made:

- app/models/booking_stat.rb
 - There was an issue with storing statistics related to the semester. The problem was resolved by removing the section for stat.semesterStart.

How to implement it:

1. Login to students
2. Navigate to bookings
3. Make a booking and there should not be an error

Task:

There should be a confirmation window to make sure they want to book on these days before further action.

User Story:

As a student or admin I want to get a confirmation when making a booking.

Description:

When I am either a student or an admin and I make a booking in the system, once the booking is successfully inputted to the database the current user will receive a success message that is clear and obvious about the booking being successfully made.

Changes made:

- controllers/make_booking_controller.rb
 - Removed a line of code that was causing an error in the booking process, this allowed bookings to be correctly made.
- views/make_booking/view_experiment.html.erb
 - Added an alert using javascript that will confirm with the student the dates of the booking before committing the booking transaction
- stylesheets/lab3b.css.scss
 - Updated the CSS so that the success message is now displayed in a big green box at the top of the page after a booking is successful

How to implement it:

1. Create a Student
2. Create an experiment
3. Select the experiment and select it on the calendar to book it
4. A confirmation box will appear before booking is finalised and by clicking 'yes' the booking will be confirmed and by clicking 'no', the booking will not be made
5. If you booked correctly and it was accepted a green box should be on the top of the page