Declaration Cover Sheet for Project Submission

SECTION 1 *Student to complete*

Name: Liam Halpin
Student ID: x13565097
Supervisor: Eamon Nolan

SECTION 2 Confirmation of Authorship

The acceptance of your work is subject to your signature on the following declaration:

I confirm that I have read the College statement on plagiarism (summarized overleaf and printed in full in the Student Handbook) and that the work I have submitted for assessment is entirely my own work.

Signature: Liam Halpin Date: 13/05/2018

NB. If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the College's Disciplinary Committee. Should the Committee be satisfied that plagiarism has occurred this is likely to lead to your failing the module and possibly to your being suspended or expelled from college.

Complete the sections above and attach it to the front of one of the copies of your assignment,

What constitu135tes plagiarism or cheating?
The following is extracted from the college's formal statement on plagiarism as quoted in the Student Handbooks. References to "assignments" should be taken to include any piece of work submitted for assessment.

Paraphrasing refers to taking the ideas, words or work of another, putting it into your own words and crediting the source. This is acceptable academic practice provided you ensure that credit is given to the author. Plagiarism refers to copying the ideas and work of another and misrepresenting it as your own. This is completely unacceptable and is prohibited in all academic institutions. It is a serious offence and may result in a fail grade and/or disciplinary action. All sources that you use in your writing must be acknowledged and included in the reference or bibliography section. If a piece of writing proves difficult to paraphrase, or you want to include it in its original form, it must be enclosed in quotation marks and credit given to the author.

When referring to the work of another author within the text of your project you must give the author's surname and the date the work was published. Full details for each source must then be given in the bibliography at the end of the project

Penalties for Plagiarism
If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the college's Disciplinary Committee. Where the Disciplinary Committee makes a finding that there has been plagiarism, the Disciplinary Committee may recommend

- that a student's marks shall be reduced
- that the student be deemed not to have passed the assignment
- that other forms of assessment undertaken in that academic year by the same student be declared void
- that other examinations sat by the same student at the same sitting be declared void

Further penalties are also possible including

- suspending a student college for a specified time,
- expelling a student from college,
- prohibiting a student from sitting any examination or assessment.,
- the imposition of a fine and
- the requirement that a student to attend additional or other lectures or courses or undertake additional academic work.

Technical Report

Liam Halpin

X13565097

Table of Contents

Executive Summary

The purpose of this report is to document, describe follow and track the development on an online presentation schedule which will be developed for a 4th year project for examination.

This report contains the contents of why the application is being developed, it will describe all the early and later stage research and investigation taken to establish how the application will be developed and what technologies will be used to achieve the development of the application.

Introduction

Background

This application will address the issue around creating schedules that lecturers of colleges and universities face each year when it comes to organizing presentations for examiners or final year course material that needs to be presented in front of a board of examiners. The application will release the lecturer of having to manually go and check each of the faculties timetables and then check each of the student's timetables.

Trying to accommodate for 100 plus students and facilitate for 30-40 plus lecturers and ensuring that the schedule is equal, fair and covers every aspect needed isn't an easy manual process and needs to be re-evaluated by each of the staff and students involved and re-written out again takes time energy and is a complex process and time consuming on the administrator organizing the events.

This application will allow the administrator to log into the web application and simply enter in all the data requirements needed to generate a schedule for students and lecturers.

The application will then generate a schedule that is suitable for both the lecturers and examiners and supervisors. The application will address all the issues encountered by the administrator and accomplish what they achieve quicker and more efficient and it's all done by automation.

Aims

The aims of the project application are:

- Research, design and develop a web application in which a lecturer or module administrator can create a schedule for students required to present coursework material to the facility and lecturers of the university in a few more clicks of a mouse.
- Develop the application in such a way that the administrator has the flexibility to generate a schedule in the format that the university is accustomed to.
- Develop a web application that will contain functional constraints to take the human aspect of the schedule into perspective. Implement the application to take break time and work loads into consideration.
- Create an application that offers the administrator a variety of options in actionable items such as exporting the schedule or saving it to the local device.
- Document the process of building the application and demonstrate implementation approaches.
- Showcase all the research and investigation into the technical aspects and conclude the findings and useful topic areas.

Technologies

The technologies used in the application will be a range of languages used for creating web applications. The application will be deployed onto a server and hosted from there.
The application will be connected to a database that will be hosted on a server and will save the input data submitted by the end user. It will use several languages such as JavaScript, PHP, HTML, CSS, SQL, Bootstrap and other online programming languages.

These languages will be used to accomplish the development of the applications user interface and writing and retrieving the data to and from the database and communicating to the servers. These languages will also be used for the implementation of security and writing test ware and testing infrastructure.

External applications and API's will be used to achieve functionality such as importing a spreadsheet into the database and storing/writing the data to the database. They will also be used for writing the data retrieved from the database into different formats to extend the user's ability to download a copy of the schedule in a spreadsheet format that can then be distributed.

Structure

The structure of this report is:

- Overview - An introduction about the overview of the project, what it plans to achieve and why it is being developed. Also, the aims of the project and what technologies and approach will be followed and used to achieve the development of the application.

- System – in the system chapter of the technical report a description off all the systems functional requirements and non-functional requirements have been listed out. Each of the requirements will have a small description of the requirement and a small mockup of how the use case will flow. This chapter also contains the systems architecture with a graphical representation of the system and a description. Graphical user interfaces are provided of how the application will look on completion. Testing techniques are described, and approach is explained.

- Conclusions – The conclusions chapter will describe all the conclusions that occurred after the research and investigation had been completed. It lists out and describes the following.

    - Advantages
    - Disadvantages
    - Opportunities
    - Limits

- Further research and development – In this chapter the report describes how the system could evolve and what other areas have potential to evolve. Also, what could be accomplished if there was a longer period and resources available and committed to the development of the application. It also describes the potential industries that the application could adapt to.

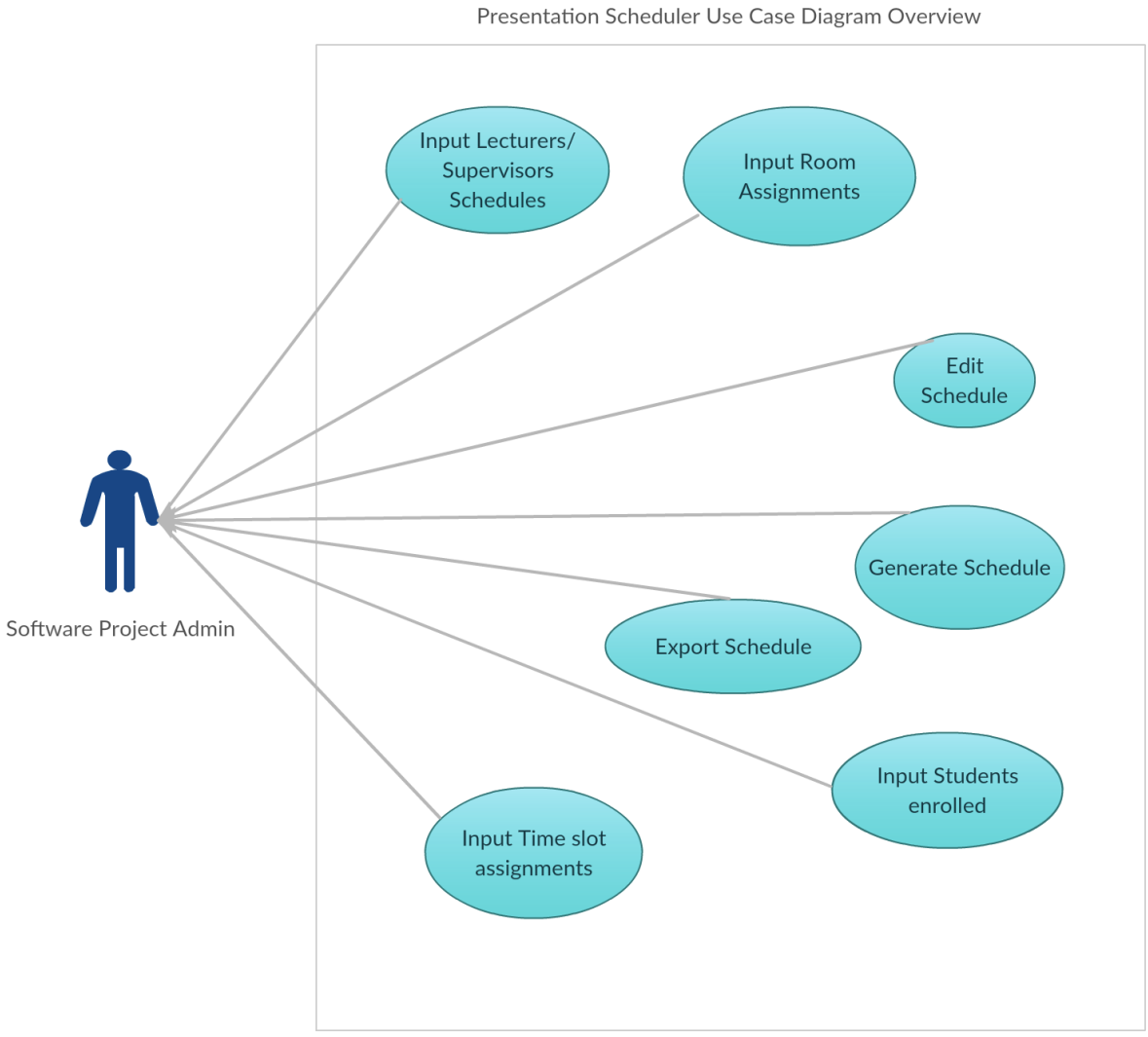
System

Requirements

Functional Requirements

The following are the list of functional requirements need when implementing the system. The following are ordered in rank of highest priority. The first requirement will be the highest priority and the second being the next highest so on and so forth.

Use Case Diagram for system overview



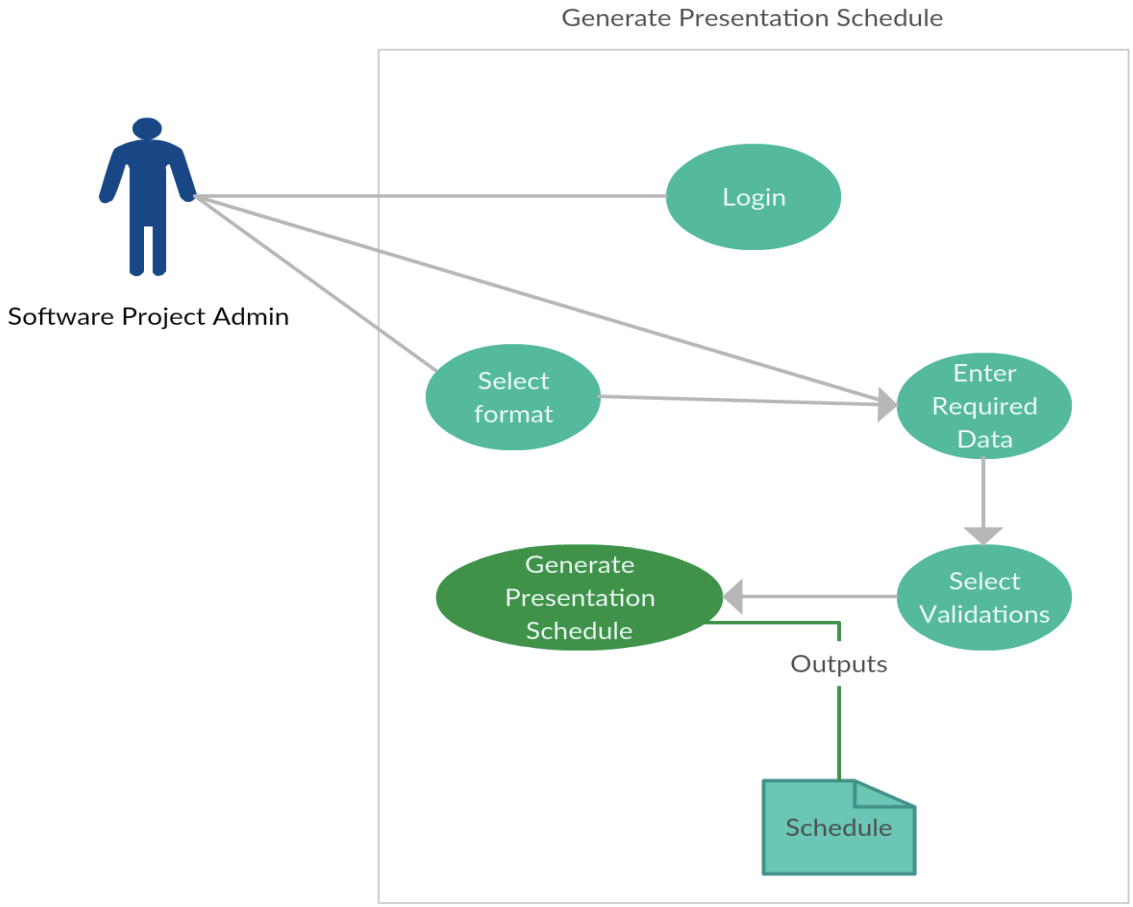
Generate Schedule Requirement

Scope:

Generating the Schedule after the user has entered all the data they required and the validations they desire is the highest use case. This is the core function of the Software and it contains the most complex aspect of the application.

Description.
This use case describes the main functional area of the application. This is the area in which the application will take all the data entered in the system and compute all permutations and validations to output the administrators schedule for the final year students, supervisors and lecturers.

Use Case Diagram for Generate Schedule



Flow Description.
This use case is one of the final slow steps of the application. To execute it, the user will need to be signed into the application and enter all the data required to execute the use case. Although this is the final step of the flow it is the highest value use case as the application will not serve its purpose if it does execute this use case. The Actor will need to log into the application, enter all the necessary data required such as room allocation and time slots etc. Then the Actor will be able to Generate Presentation Schedule and the system will compute the Schedule and output it to the Actor.

Precondition.

The application will be in a waiting state with all the required data stored in the Database before the execution of this use case and it will be ready to compute the schedule once the Actor initiates the use case.

Activation

This use case starts when the <Software Project Admin> Clicks on Generate Presentation Schedule Button on the Applications User Interface.

Main Flow.

1. The system identifies all input data has been entered and is ready to execute the computation of a schedule.
2. The <Software Project Admin> will select 'Generate Presentation Schedule' on screen.
3. The system will then begin designing the schedule.
4. The system will then store the presentation Schedule and wait for the <Software Project Admin> to select view Schedule.
5. The <Software Project Admin> will then select 'View Schedule'.
6. The system will retrieve the presentation schedule and display it on screen to the user.
7. The <Software Project Admin> will then export/save the Schedule.
8. The system will export/save the schedule to the <Software Project Admin> device.

Termination.

The system will present a UI to the <Software Project Admin> asking if they would like to create another schedule or go back to the home page of the application.

Post Condition

The system will enter back into a wait state at the home page of the application waiting to either log out or begin to create another schedule depending on what the <Software Project Admin> wants to do.

Export Schedule Requirement

Use Case.

Once the <Software Project Admin> has completed generating and creating a schedule, the system should offer an option to allow the <Software Project Admin> to Export or save the schedule either allowing them to print it off or export it externally to the device to allow them to make copies for distribution.

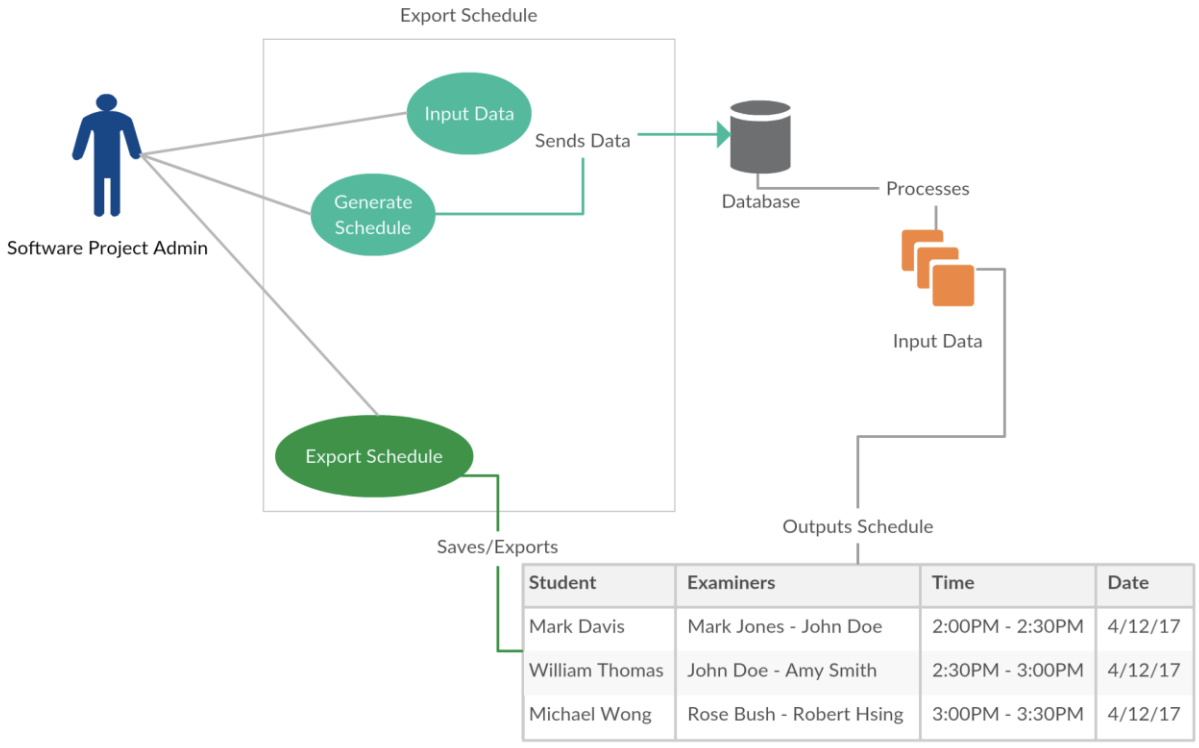
Scope.

The scope of this requirement it to allow the user to save the generated Schedule to allow them to distribute it to the lecturers and supervisors and students involved in the process.

Description.

Once the <Software Project Admin> is satisfied with the presentation schedule that has been created by the application the System will offer the option for the <Software Project Admin> to save and export the schedule to allow them to distribute or publish the schedule to the students and examiners.

Use Case Diagram for Export Schedule



Student	Examiners	Time	Date
Mark Davis	Mark Jones - John Doe	2:00PM - 2:30PM	4/12/17
William Thomas	John Doe - Amy Smith	2:30PM - 3:00PM	4/12/17
Michael Wong	Rose Bush - Robert Hsing	3:00PM - 3:30PM	4/12/17

Flow description.

The system will have generated a presentation schedule for the <Software Project Admin> and will display options for the user to execute, such as preview or export. Once the <Software Project Admin> decides on the action they want to execute, the flow will begin to save and export the schedule and save it to the device. This is the last step in the overall flow of the application and is not a mandatory use case each time the <Software Project Admin> uses the application. This means if the <Software Project Admin> wants to preview the schedule and may want to make some additional changes to schedule, the <Software Project Admin> doesn't have to save the schedule once it has been created.

Precondition.
The system will be in a completion state as the schedule has been generated and now it's up to the <Software Project Admin> to decide what they want the system to do.

Activation.
This use case starts when the <Software Project Admin> clicks export on the applications UI.

Main flow

1. The system has identified that the schedule has been created and waits for the <Software Project Admin> to decide the next action.
2. The <Software Project Admin> will then click export on the application's UI.
3. The system will then save a copy of the schedule to the device.
4. The <Software Project Admin> will then have an electronic version of the schedule.

Alternative flow.

Alternative flow 1: User does not want to download a copy of the schedule.

1. The system has identified that the schedule has been created and waits for the <Software Project Admin> to decide the next action.
2. The <Software Project Admin> decides to edit the data for the schedule to add in additional data or time slots or book different rooms.
3. The system will go back to the data input screen
4. The <Software Project Admin> will then input the data
5. The <Software Project Admin> will then generate the new schedule
6. The use case will continue position 1 of the main flow.

Termination
The system presents the home screen to the <Software Project Admin>.

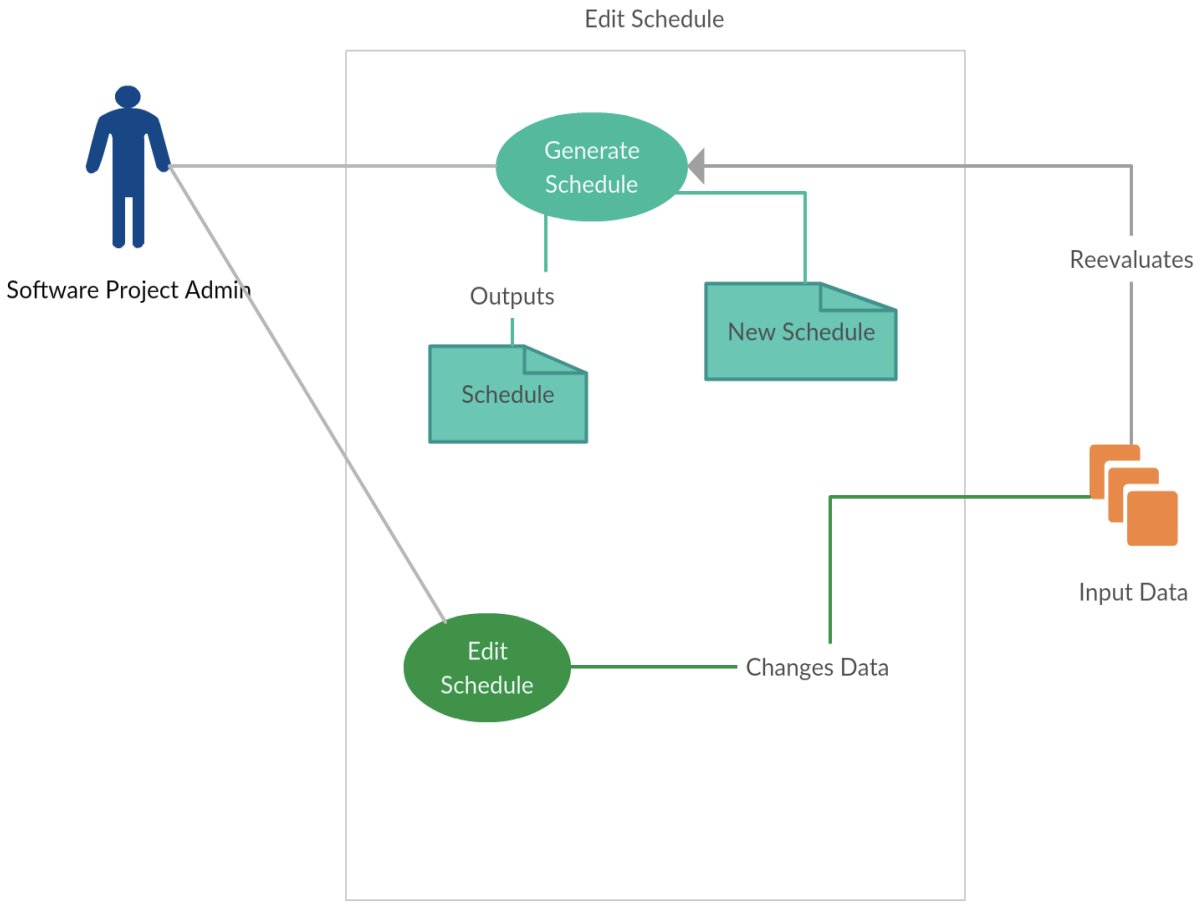
Post condition
The system will go into a wait state again once the flows have been completed.

Edit Schedule Requirement

Scope.
The scope of this requirement is to allow the <Software Project Admin> to make changes to a schedule that has been created by the application if they see an area which won't be applicable for certain lecturers or students. This will allow them to enhance the schedule for optimal results.

Description
If the <Software Project Admin> needs to make amendments to the schedule that the application has created for them due to lecturers not being available for a period or a student can't attend for personal circumstances, then the application will allow them to make amendments to it to make it as ideal as possible for everyone. It will re-enter the data and create a new schedule with the new additional changes.

Use Case Diagram for Edit Schedule.



Flow Description
The system will have generated a schedule after the specified data has been entered by the <Software Project Admin>. There may be exceptions where the <Software Project Admin> may need to accommodate for different circumstances for either a lecturer or a student. This flow will allow them to enter the restrictions to satisfy as many needs as possible and ensure everyone is accounted for and can attend the presentations at the given time. This flow will use the generated schedule

Precondition
The system will be in a wait state. Once the schedule has been generated it goes back to asking the <Software Project Admin> what is the next action they want to take with the Schedule.

Activation
This use case is activated when the <Software Project Admin> selects 'Edit Schedule' on the application's 'UI', which will activate the edit requirement.

Main Flow

1. The system will have generated a schedule.
2. The <Software Project Admin> will inspect the schedule the system has outputted.
3. The <Software Project Admin> will select 'Edit Schedule'.
4. The <Software Project Admin> will select and enter the amendments they wish to include.
5. The system will ask to user do they want to regenerate the schedule.
6. The <Software Project Admin> will then select 'Reschedule'.
7. The system will then generate a new schedule.
8. The system will then output the new schedule.
9. The <Software Project Admin> will review the new schedule.

Alternative Flow

1. From position 9 of the Main flow – if the <Software Project Admin>is still not happy or needs to make some more adjustments, the system will allow the user to 'Edit Schedule'.
2. The <Software Project Admin> will select 'Edit Schedule'.
3. The use case will continue at position 4 of the main flow.

Termination
The system will return to the home screen once the <Software Project Admin> is happy and selects continue.

Post Condition.
The system will go into a wait state until the <Software Project Admin> needs to use it again.

Input Supervisors/Lecturers Schedule Requirement.

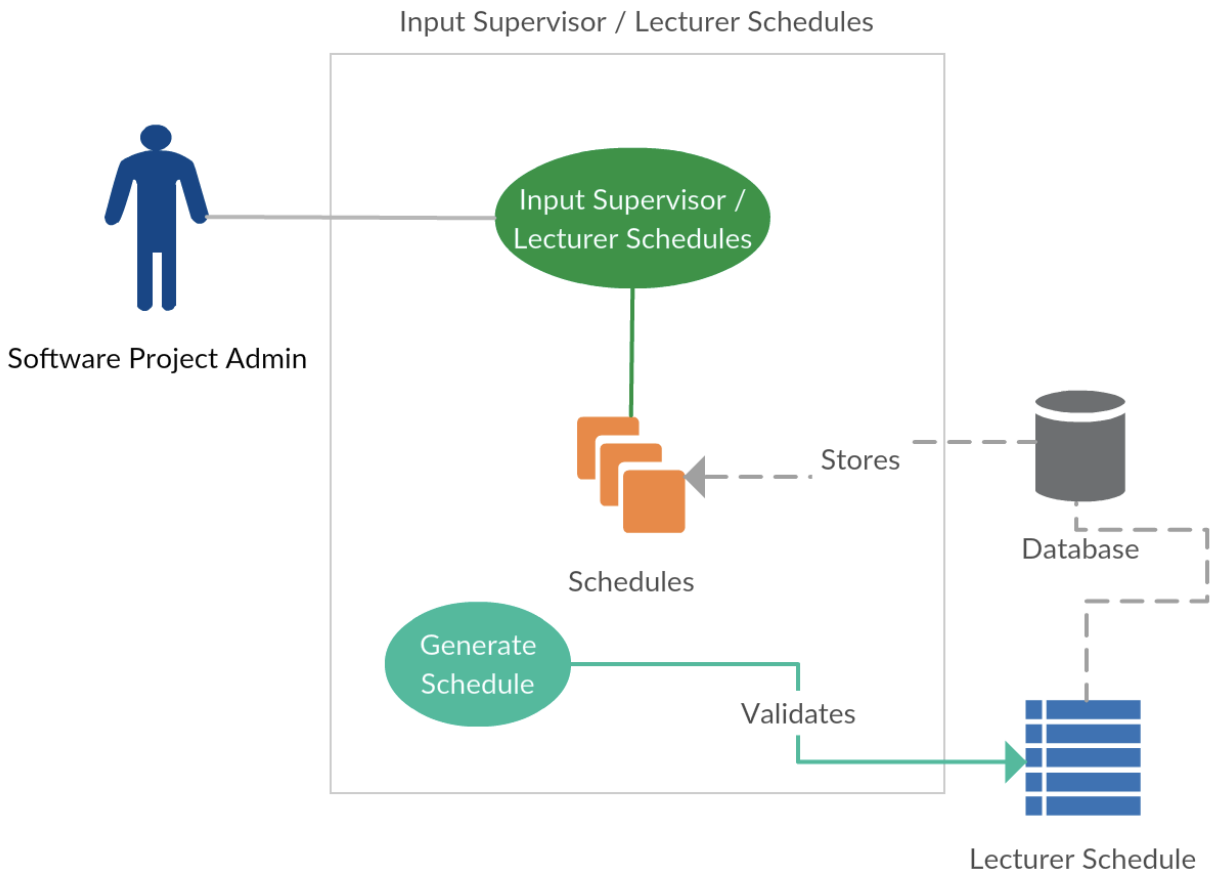
Scope

The scope of this is to get the <Software Project Admin> to enter in the supervisors and lecturers' schedules for their teaching times and classes. This will allow the system to identify where is acceptable to place them. This system will automatically know not to assign a time slot for a lecturer if they are scheduled to teach a class at a certain time. This will ensure the schedule is efficient and correct.

Description

This use case will be used to store that data of the lecturer and supervisor's schedules into the database, these will be then used to identify what times that the lecturers are available to sit for presentation. This will be critical in making the system dynamic and automated and not create a schedule that is unusable to the <Software Project Admin>.

Use Case Diagram for input Supervisors/Lecturers Schedule Requirements



Flow Description.
This flow will be in the beginning of the data set up. Without this data input from the user the application will generate a schedule, but it will place lecturers during times that they are not available which would be disruptive to the entire college. Lecturers will need to miss giving their lectures and they may get an unfair amount of presentations to attend.

Precondition.
The system will be in initialization mode with no data entered into the database. Once the user inputs the data it will be written to the database and stored to be used later in the flow.

Activation
This is activated when the user is requested to enter in the lecturer's/supervisor's data.

Main Flow

1. The system will prompt the user to "Please enter in the lecturers and or supervisors Schedule"
2. The <Software Project Admin> will then enter in the schedules for the lecturers.
3. The system will create a table in the database and store the table into the database.
4. The system will notify the <Software Project Admin> that the data has been stored.
5. The system will bring the <Software Project Admin> to the next process of creating the schedule.

Exceptional Flow.

1. The <Software Project Admin> tries to enter in the data.
2. The system cannot connect to the database.
3. The system will notify the <Software Project Admin> of the error and execute the precautions to take in this event by returning the <Software Project Admin>to the main home page and insisting they "try again later"

Termination
The use case is terminated once the user has submitted the data and the system has stored successfully.

Post Condition.
The system is then in a wait state after the system has directed the <Software Project Admin> to the next stage of the process.

Input Room Assignments Requirement

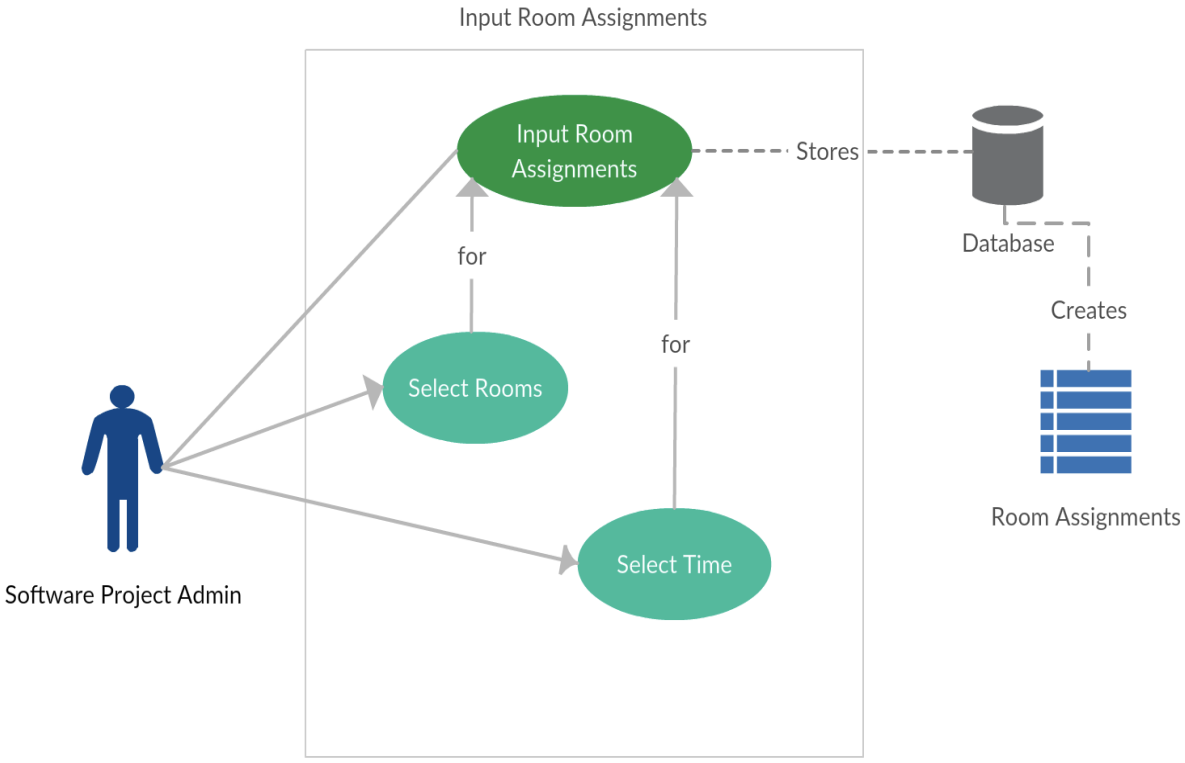
Scope

The scope of this requirement is to get the <Software Project Admin> to enter in the rooms and the time scale they want the presentations to take place in, for example Room 101 from 9:00AM – 5:00PM. This data will be used to write the schedules data for informational purposes.

Description

This requirement is used to obtain the data needed for displaying what room the students will be presenting in and where the lecturers need to be for the presentation. This data is also used by the system for formatting the table that the system will write to.

Use Case Diagram for Input Room Assignments



Flow Description
This flow is part of the data set up that the system needs for writing a schedule alongside the previous requirement “Enter Supervisors/Lecturers Schedules”.

Precondition.
The system will have created the schedules table and will then be left in a wait state for the room assignments input

Activation
This use case is activated after the <Software Project Admin> has submitted the schedules of the supervisors and lecturers and gets directed to this use case.

Main Flow.

1. The system will prompt the <Software Project Admin> to enter in the room that the <Software Project Admin> is booking for the presentation.
2. The <Software Project Admin> will then enter in the rooms that are going to be used for the presentations.
3. The system will then ask the <Software Project Admin> for which time the schedule should accommodate for.
4. The <Software Project Admin> will enter in the time frames needed for the rooms and presentations.
5. The system will create a new database table based on the <Software Project Admin> input.
6. The system will store the data to the database.
7. The system will direct the <Software Project Admin> to the next stage of process in creating a schedule.

Exceptional Flow.

1. The <Software Project Admin> tries to enter in the data.
2. The system cannot connect to the database.
3. The system will notify the <Software Project Admin> of the error and execute the precautions to take in this event by returning the <Software Project Admin>to the main home page and insisting they “try again later”

Termination.
The use case is terminated once the user has submitted the data and the system has stored successfully.

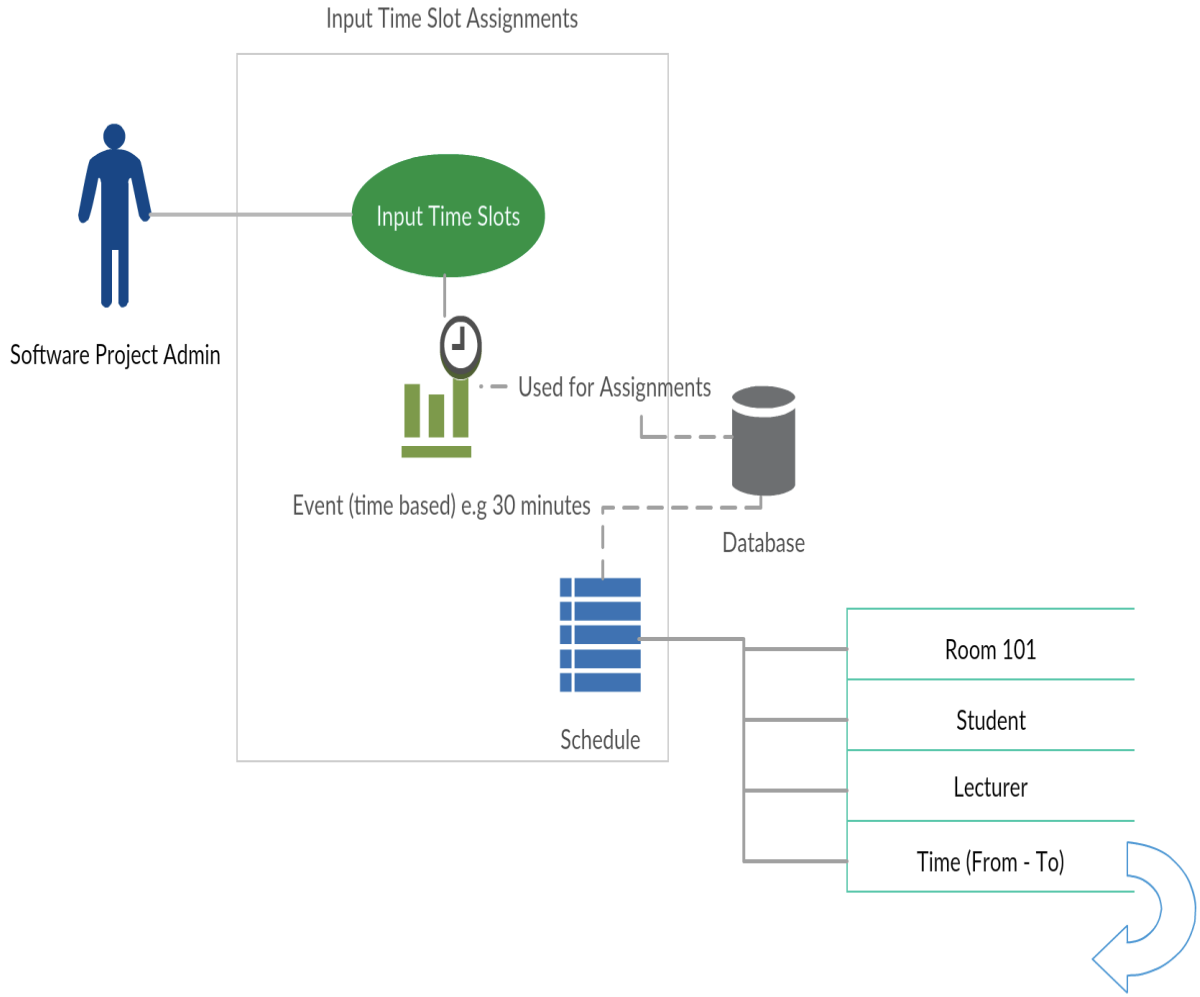
Post Conditions
The system is then in a wait state after the system has directed the <Software Project Admin> to the next stage of the process.

Input Time Slot Assignments

Scope
The scope of this requirement is to instruct the <Software Project Admin> to specify how long each student will have to present to the supervisor and lecturer.

Description
The use of this use case is to get the <Software Project Admin> to specify how long they are allowing the students to present for. This piece of data will be used regarding the time slots allocated for each student. The database will generate each record knowing only to assign each student a specific amount of time to present. The database will know when to stop once a certain time has been reached and presentations have ended for the day.

Use Case Diagram for Input Time Slot Assignments



Flow Description
This use case comes as part of the data set up from <Software Project Admin> and is stored in the database like the other data inputs. Once the data is stored the next use case of the flow can be activated.

Precondition
The system is in waiting state waiting for the input to store in the database.

Activation
This use case is activated after the system prompts the user to enter in the time slots needed.

Main Flow

1. The system will prompt the <Software Project Admin> to enter in a time slot allocation.
2. The <Software Project Admin> will enter the time slot that they require.
3. The system will write the data to the database and store it.
4. The system will redirect the <Software Project Admin> to the next stage of the process.

Exceptional Flow.

1. The <Software Project Admin> tries to enter in the data.
2. The system cannot connect to the database.
3. The system will notify the <Software Project Admin> of the error and execute the precautions to take in this event by returning the <Software Project Admin>to the main home page and insisting they “try again later”

Termination
The use case will terminate once the data has been successfully saved to the database.

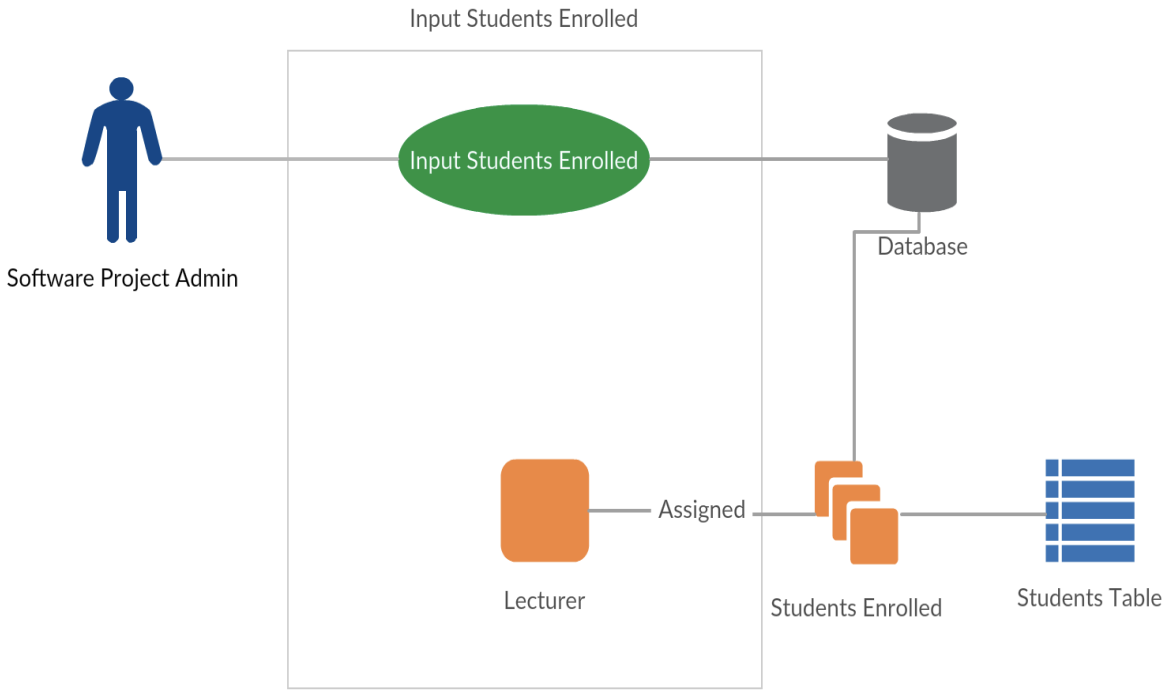
Post Condition
The system will enter a wait state after the use case has terminated.

Input Students Enrolled Requirement

Scope
The scope of this use case is to prompt the <Software Project Admin> to enter in all the students enrolled for the module that need to present their project material.

Description
This use case is used for obtaining more data regarding what students need to be assigned a time slot for presenting. This use case will prompt the user to enter in the data and the system will store the information into the database after creating the table for them.

Use Case Diagram for Input Students Enrolled Requirement.



Flow Description
This is part of the data input flow alongside the other ones illustrated above. This is one of the final pieces of data needed to construct the schedule.

Precondition
The system will be in a waiting state for the <Software Project Admin> to enter in the required data.

Activation
This use case is activated when the system prompts the user for the data.

Main Flow

1. The system will prompt <Software Project Admin> to input the students still enrolled in the module.
2. The <Software Project Admin> will enter in the data.
3. The system will create a table in the database.
4. The system stores the data to the database.
5. The system will redirect the <Software Project Admin> to the next stage of the process.

Exceptional Flow.

1. The <Software Project Admin> tries to enter in the data.
2. The system cannot connect to the database.
3. The system will notify the <Software Project Admin> of the error and execute the precautions to take in this event by returning the <Software Project Admin>to the main home page and insisting they “try again later”

Termination
The use case is terminated once the data has been stored into the database and the tables have been created.

Post Condition
The system goes into a wait state until the <Software Project Admin> is ready to generate the schedule.

Non – Functional Requirements

Performance and Response Time Requirement

Performance will need to be considered and implemented into the system. The database driving the system will need to write data to the database and dynamically create the presentation in the background without affecting the users experience. The database will need to start creating the table after each piece of data gets committed to it so as not to delay the creation of the schedule and keep the user waiting a period for a schedule. The creation of the schedule should take no longer than 5 minutes to create, finalize and post the schedule to the user for review. To achieve this in the system, the database will begin to design the schema of the database through each stage of the input. All that will be left to do is run a validation check on the data to ensure the data is validated, has the correct functional constraints implemented and the data has integrity.

Ensuring that the system performs and has a fast paste response time will allow the user to use the application without obstructing or delaying their daily transactions and workload.

Availability Requirement

The system will need to be available on request 24 hours a day 7 day a week. Giving the user the ability to login to the system and create a schedule for any year of a given course for any number of students for any type of coursework. This relieves the faculty using the system for other work and gives them an on-demand system to work with.

Security Requirement

The application will not store any sensitive information regarding any student or lecturer or the Software Project administrator, this means the data being used in the application doesn't need to be encrypted or locked away from hackers.

The framework of the database will need to be secure to ensure the schema and the systems business logic for the creation of a schedule is not tampered or modified in anyway. If a potential hacker decided to attack the web application and change the logic or schema of the applications database, then a Software Project admin uses the application to create a schedule for 250 students to demonstrate their final year project in front of lecturers, supervisors and potentially external examiners and have put trust into the application only to find out on the day that all 250 students are presenting at the same time to the same lecture, it compromises the integrity of the application and also will affect all of the personnel involved and can compromise vast areas of the year.

The application will need security implemented into the system and an authentication feature for the end user to ensure each user of the application is permitted and does not have ill intentions to harm the system. Precautions will be implemented into the system to ensure all possible threats have preventions included into them.

Reliability Requirement.

The system needs to output correct robust data to the end user. Ensuring the user can place trust into the system with peace of mind that the schedule created is fair, correct and efficient and reliable.

Maintainability Requirement

The application must be maintainable. Adding new security features or new API or libraries to the application will be essential. This will allow the application to enhance its features and not make it difficult to maintain the databases and servers running the application.

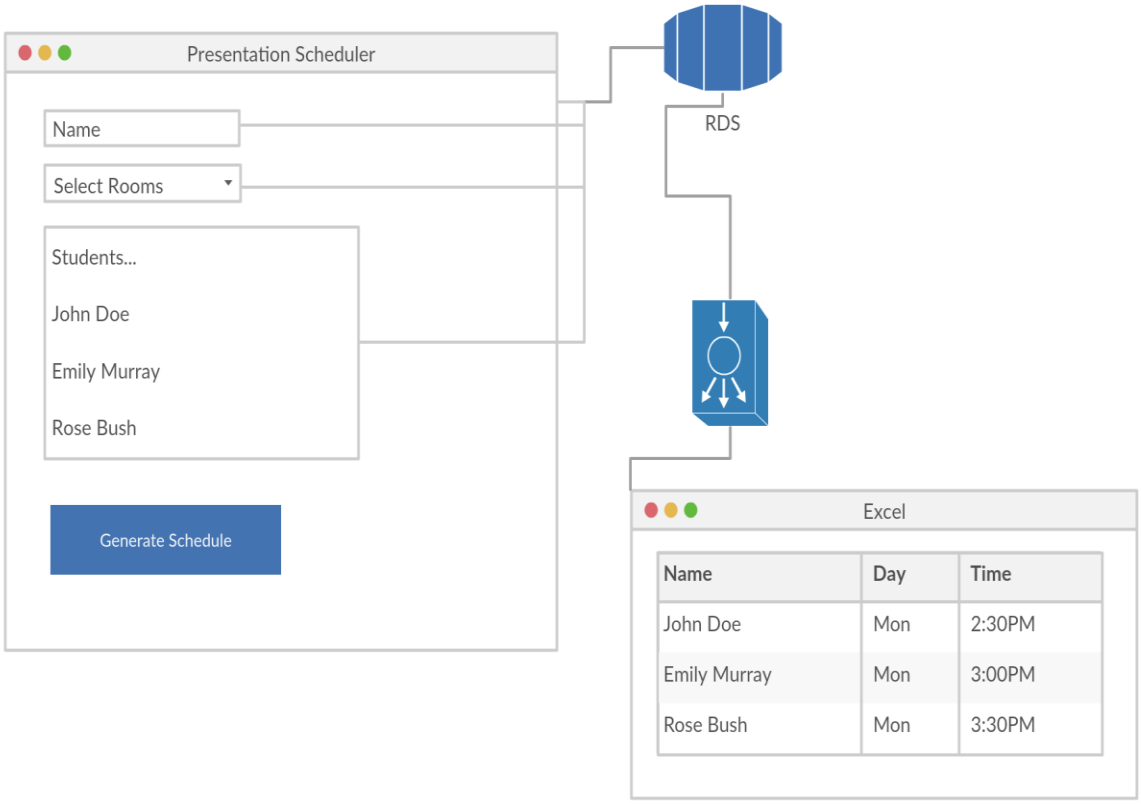
Extendibility Requirement

Developing the application to allow its ability to extend is a requirement for reasons such as if the application was to be sourced or taken up by an education enterprise or organisation, allowing it to extend its functionality to the different branches will be beneficial and can give the application more power to grow.

Resource Utilization Requirement

One main concern for this application will be the response time of the database, writing new data to the database while concurrently running validations and maintaining its dynamic aspect will be an issue that will have to be monitored closely. Once this is done the application needs to format the data with a suitable format that will be user friendly.

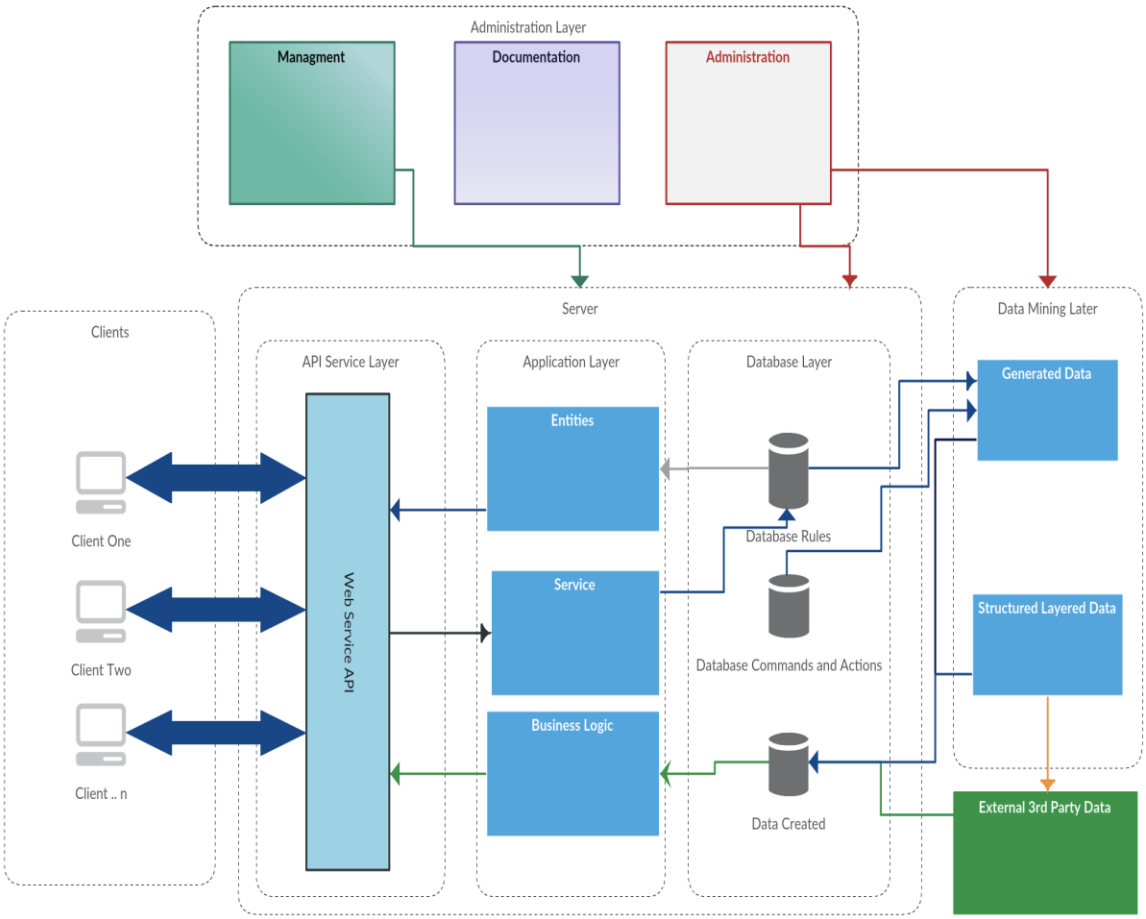
Interface Requirements



The interface will need to use an external tool to extract the data generated from the database and then store it into a format that can be used and opened with 3rd party software such as open source spreadsheets or excel spreadsheets. This application will make it possible for users to extract data and save an electronic copy of the data. In doing so, the application will support more formats over time and a new feature can be added to export into different formats or even export the data as JSON or XML in the future and be included in a web service API for another application.

Design and Architecture

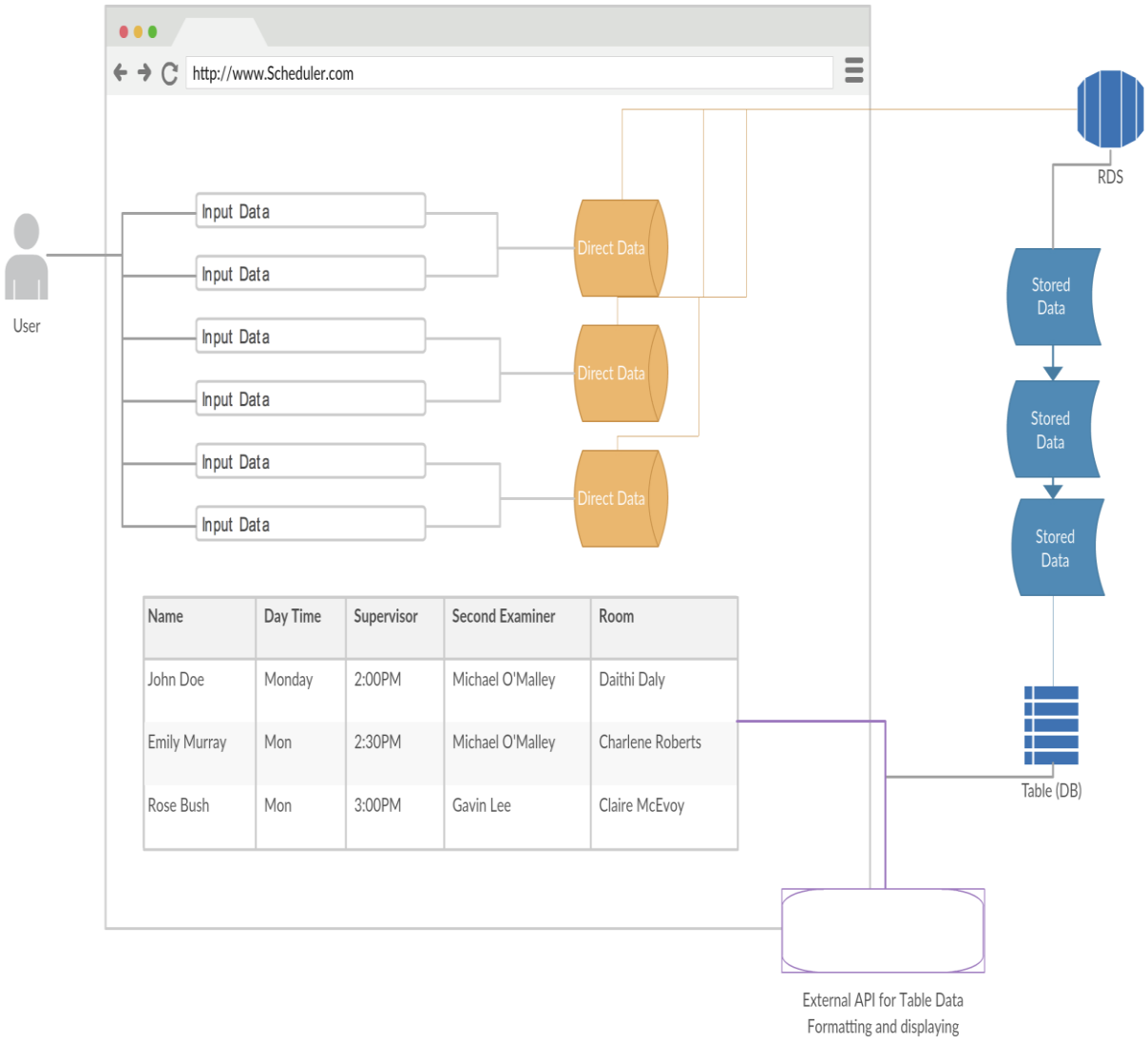
System Architecture



The system architecture will be a standard to how most Web applications are deployed to date. The client side will be the users of the systems sending the requests. The requests will run through any API Services implemented during the deployment – Depending on the selected domain and server that the application gets deployed to this may vary.

The application layer will contain all the applications interfaces and will be the controllers that the user will interact with to process the transactions. Databases will send and retrieve the data back and forth through several layers. Some of the databases will dynamically create data as the user is entering in the input data while the others will have predetermined data for functional purposes.

External Application

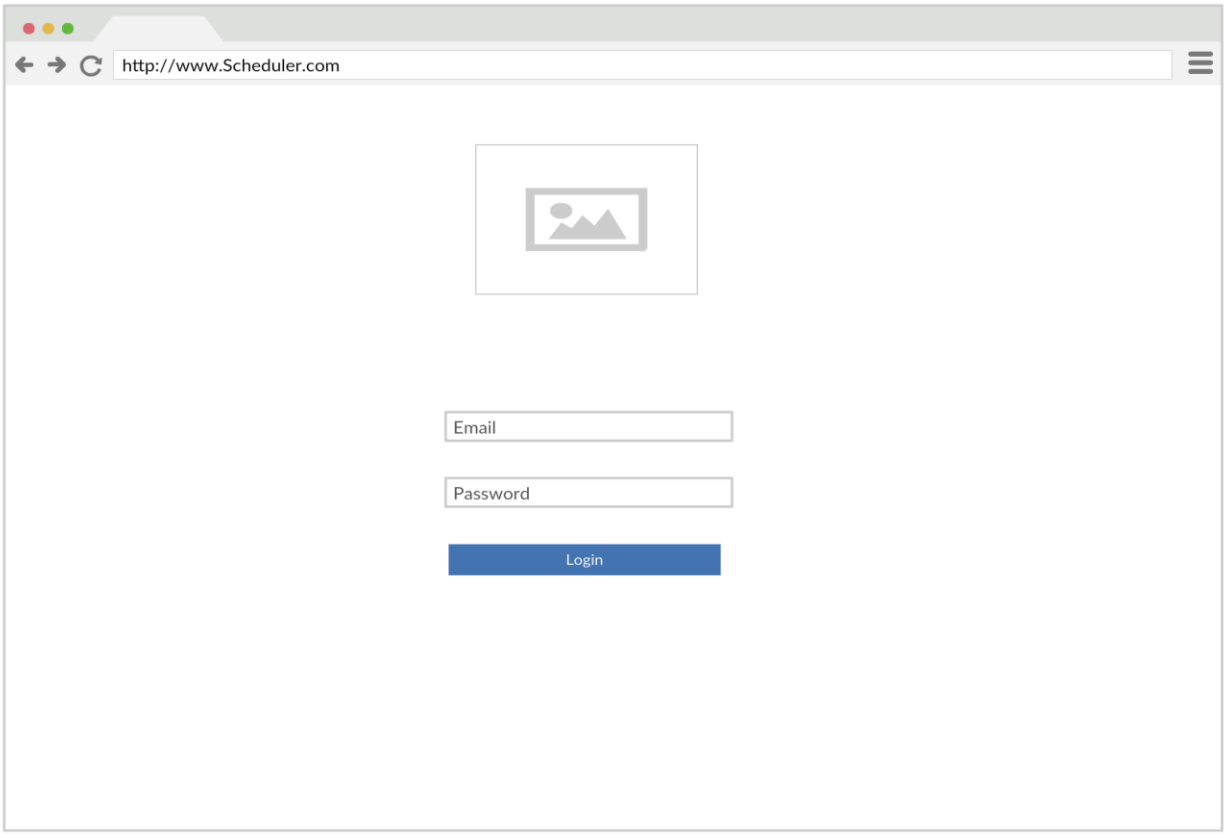


A third-party tool will need to be used for formatting the data being displayed to the user. The data will be retrieved from the database and stored into a table once a copy of a schedule has been created by the application. Once the data has been retrieved it will be stored in a table format and will need to be formatted in a nice display.

Also, with dynamic display changes for the user the third-party application will support this feature and code will need to be written to tolerate these changes. Once the changes have been made they need to be written to the database so the new data for a row on the table must be overwritten.

Graphical User Interface

Login GUI



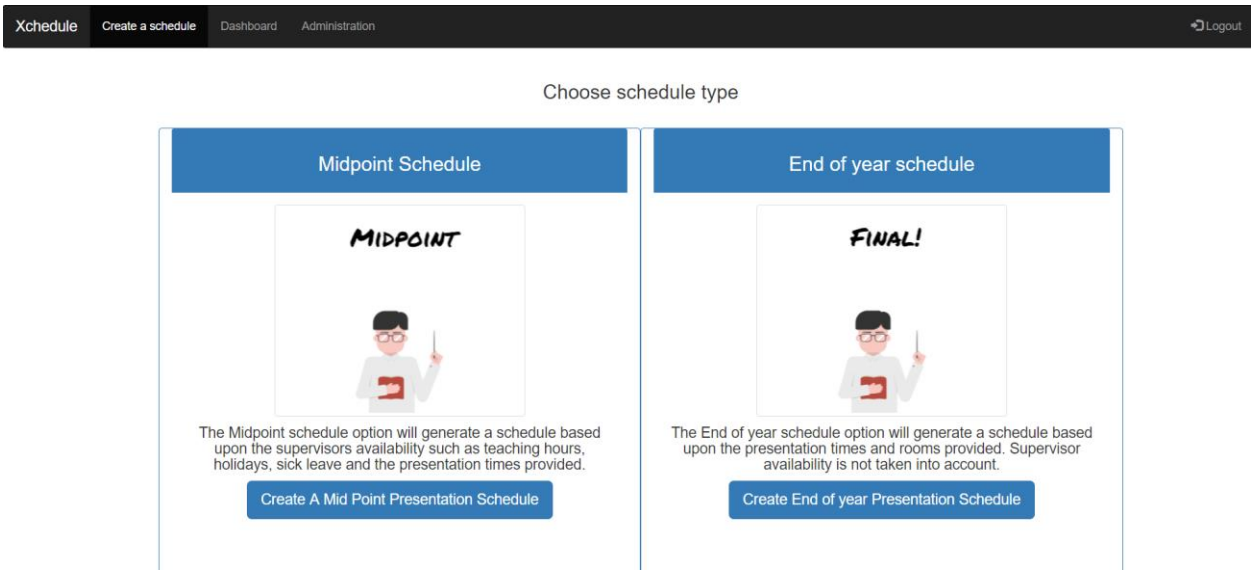
Description of Login GUI

The login page illustrated above will be used for authentication for the Software Project administrator. Creating a login feature to only allow permitted personnel to access the application will ensure no student or standard lecturer will have the ability to access it and amend schedules, dates etc.

In doing this, this could be developed to incorporate external educational systems in which a software project admin could use functionality from Moodle and external data from Moodle to make inputting the data redundant and the application could access enrollments and even extend to allow the college or university to submit its data to the application which would relieve more of the administration process of the end user.

The login page will allow the end user to log in and will also have helpful links to online documentation and other helpful and user tips.

Options Menu GUI

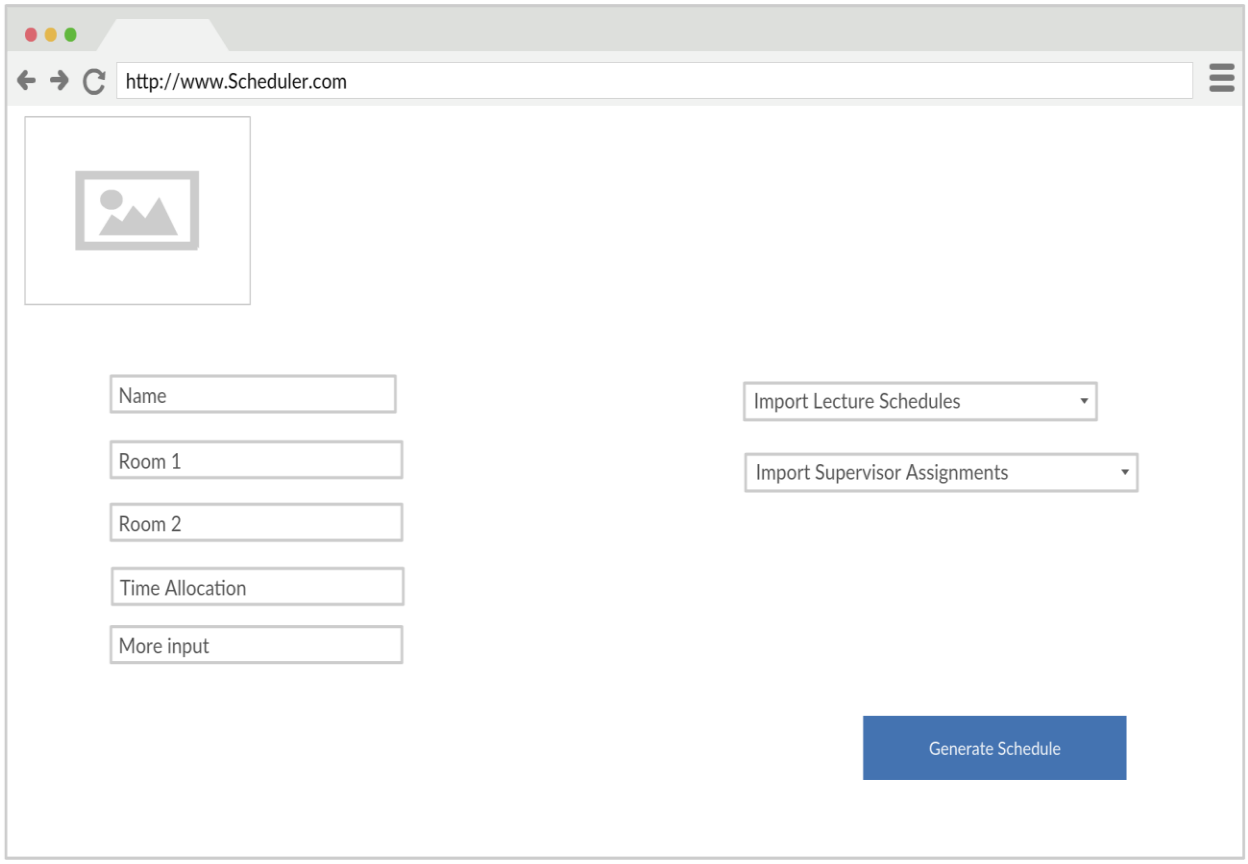


Description of Options Menu GUI

The options page displays the option the application allows the user to act upon. There are several actions that the user may want to take such as “Create Midpoint Presentation Schedule” which will then redirect the user to the processes of the creation of a schedule.

The second option is “Create End of year Presentation Schedule” which will then redirect the user to the processes of the creation of a schedule.

Create Schedule GUI



Description of Create Schedule GUI

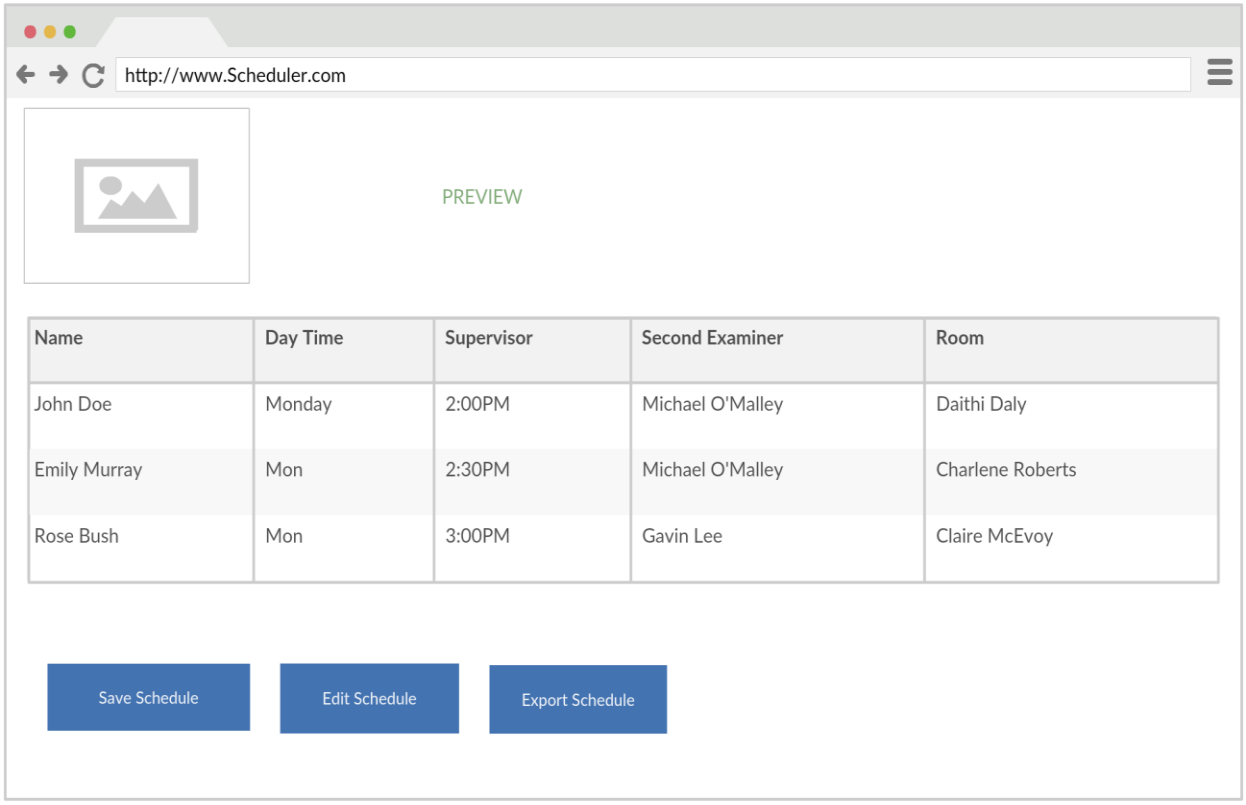
The create schedule page illustrated above will be the page(s) that take the core input data needed by the user for the application to make the schedule. This data will be driven along with the actions database that will contain the constraints or business logic of the of the application.

The create page can be designed as above or it can be designed as individual stages using "Next" functionality which will just stream the process of creating the schedule. This might not be a user-friendly design as it introduces more clicks into the system which is not customer or user friendly.

This screen will need to contain containers and upload links to allow the user to upload files containing time-tables of the lecturers which the database will go and retrieve to establish boundaries.

Once the data has been selected and stored to the database the application will generate a loading screen which has not been illustrated in this documentation as it is not redeemed as a main feature page. Once the schedule has been created, the preview window illustrated below will load.

Preview Schedule GUI



Description of Preview Schedule GUI

The preview window will be used once the creation of the schedule has been completed. The information will be retrieved from the database and displayed using a style like a HTML table except with more designs and formatting.

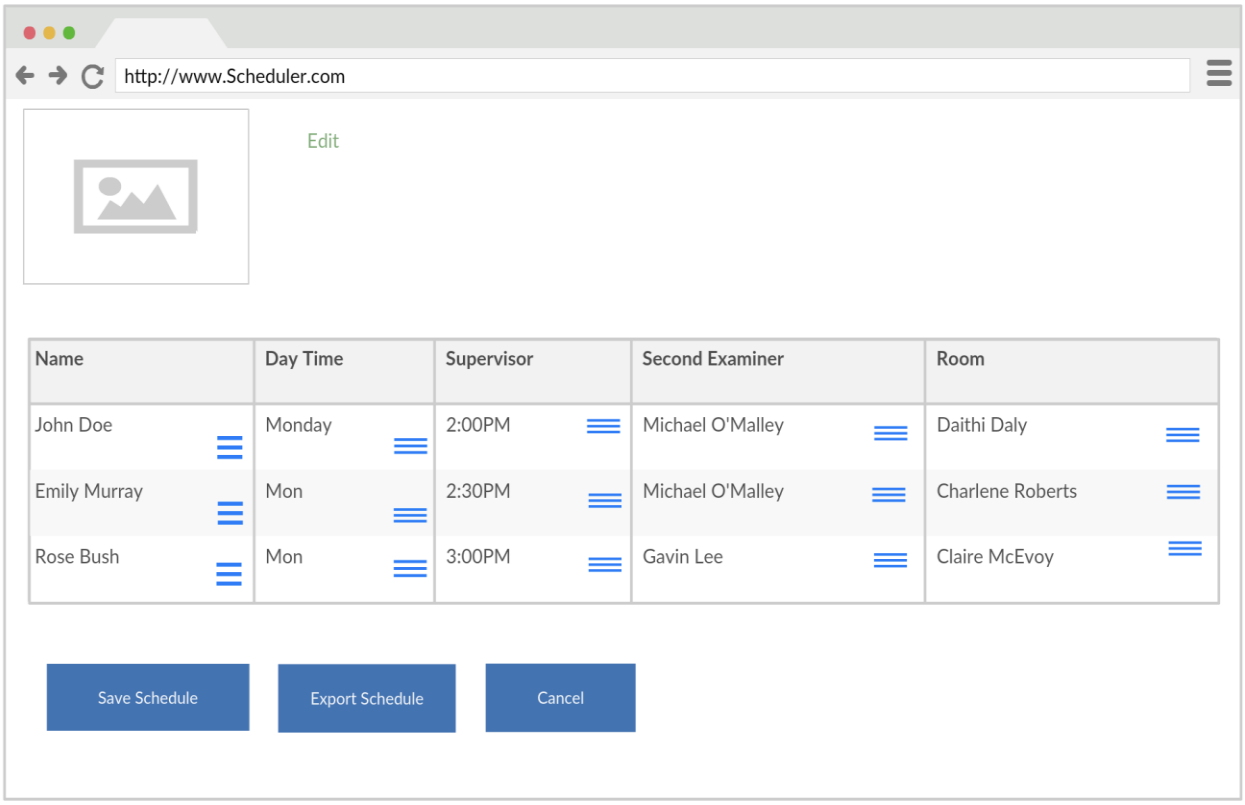
The preview window is used to show the newly generated schedule and allow the user to view the schedule that has been created by the application. From the preview window the user is prompted with several options.

Save Schedule – This is used to commit the schedule to the database and include it in the list of ones showed in the view schedule option from the options page illustrated in the second GUI.

Edit Schedule – Selecting this option will open the schedule in edit mode and allow the user to make some small minor changes that will be reflected into the database upon completion.

Export Schedule – This option will save the schedule to the database and then will export the schedule into a spreadsheet format and save a copy of it onto the local device. This will enable the user to upload onto an enterprise education application or send copies of this schedule via email.

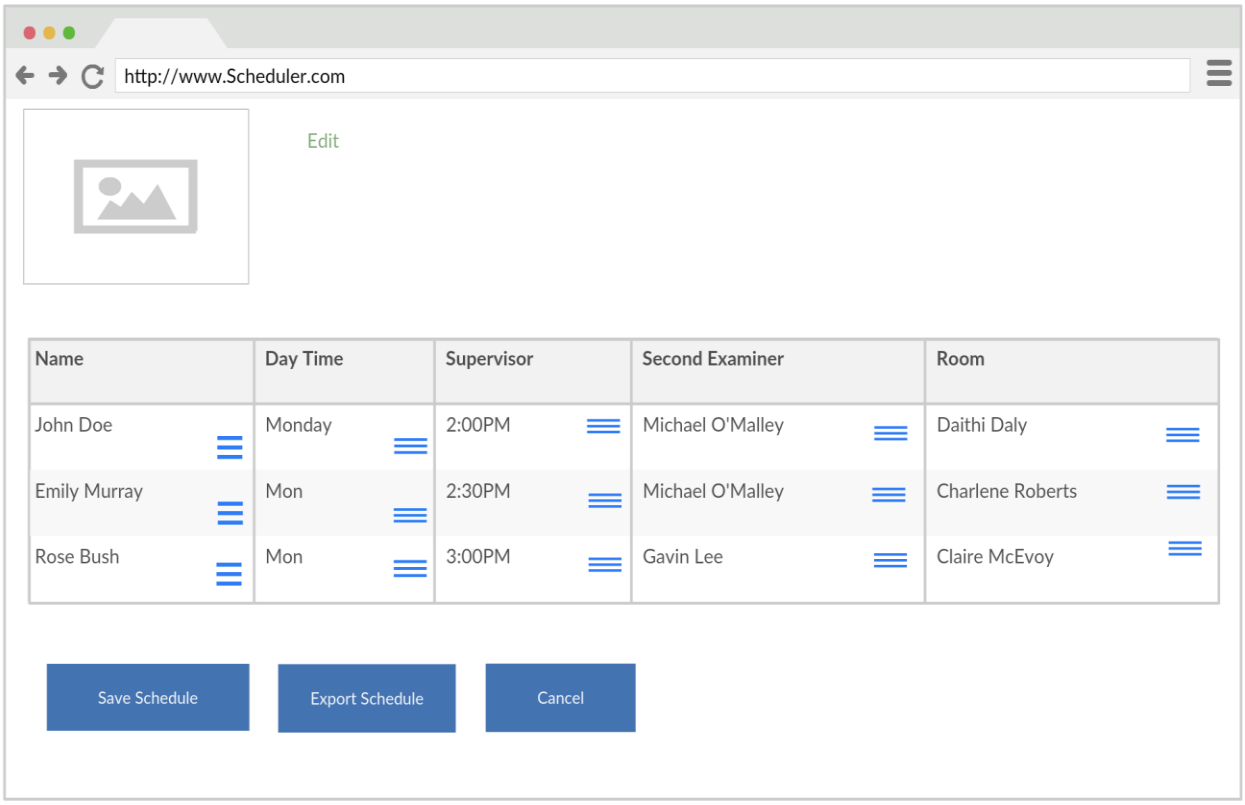
Edit Schedule GUI



Description of Edit Schedule GUI

The edit schedule will be used in two flows of the application. The edit schedule page will be the directed page of the application once the user has created a new schedule and the application is showing the preview page. If the user needs to make some minor changes to the schedule the application will open the schedule in an edit mode and let the user either dynamically move slots around or remove slots or change the entry of a lecturer or supervisor etc.

View Schedule GUI



Description of View Schedule GUI

View Schedule will be selected from the Options menu, depending on the schedule that wants to be accessed “Create Midpoint Presentation schedule” or “Create End of year Presentation Schedule” each of these options will be show the current schedule that is in the application.

Testing

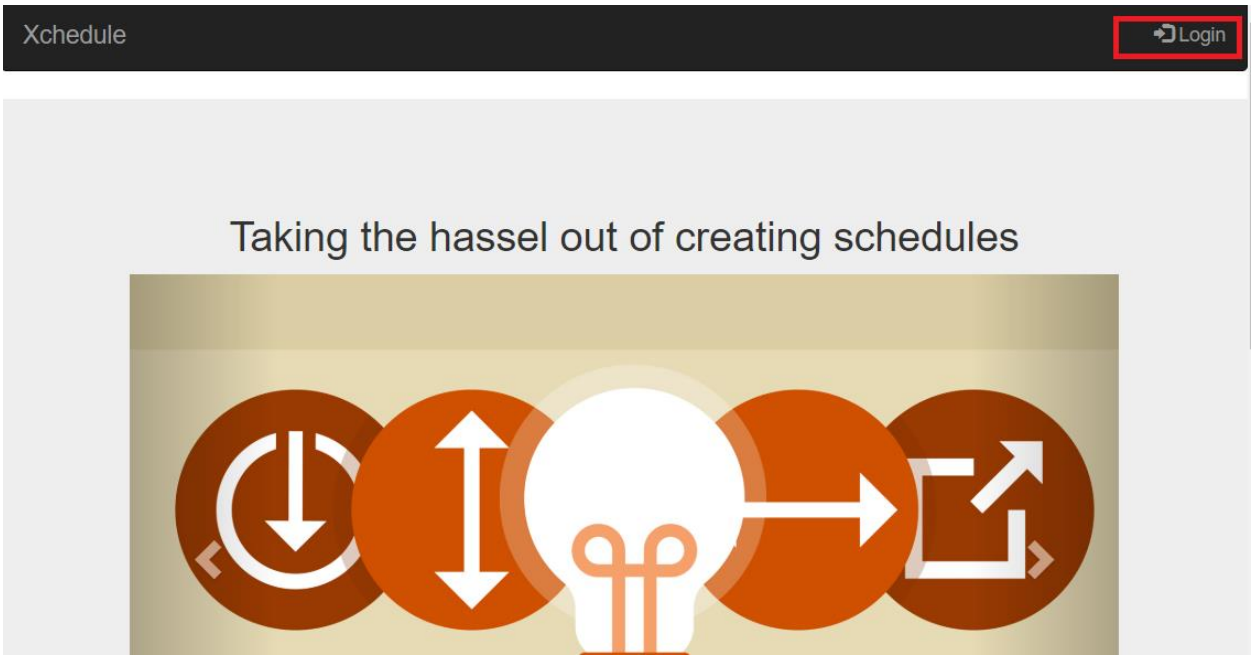
Functional testing

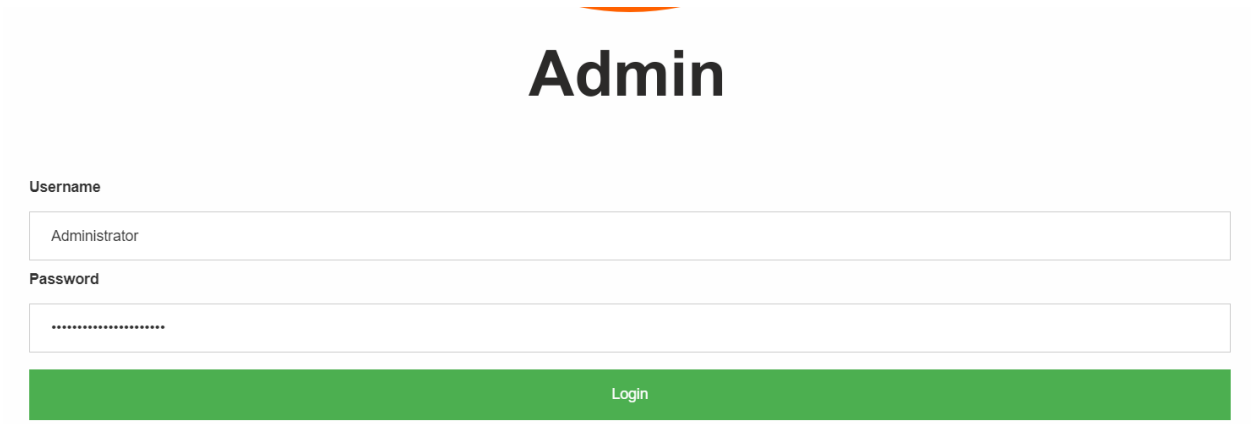
The applications functional requirements were tested by doing a full slow through the system.

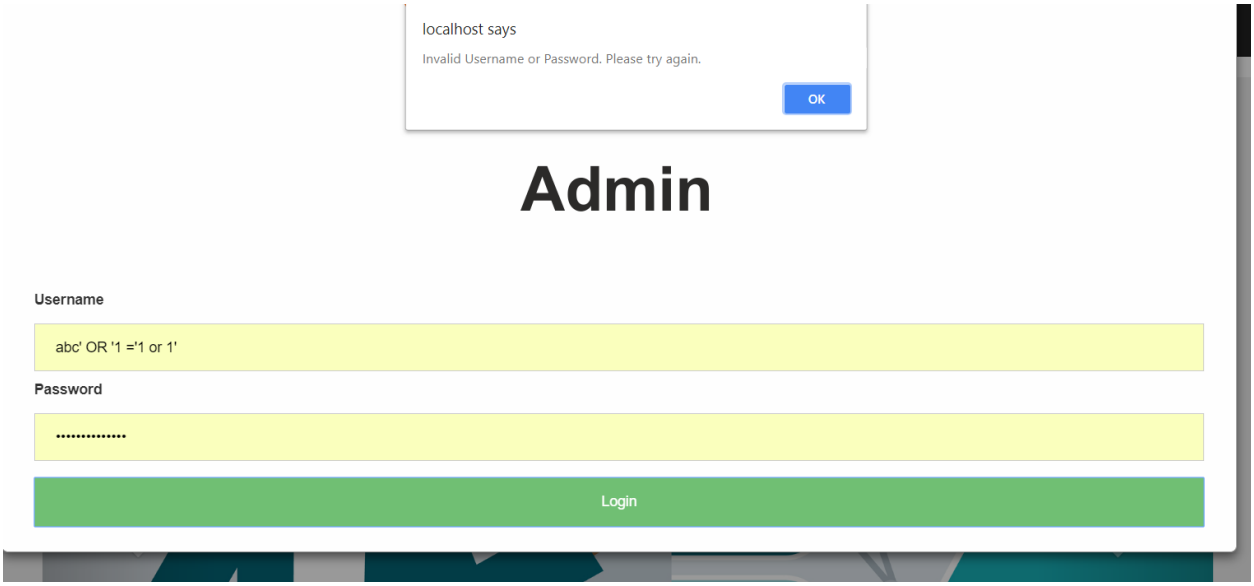
The following functional requirements were tested during this process.

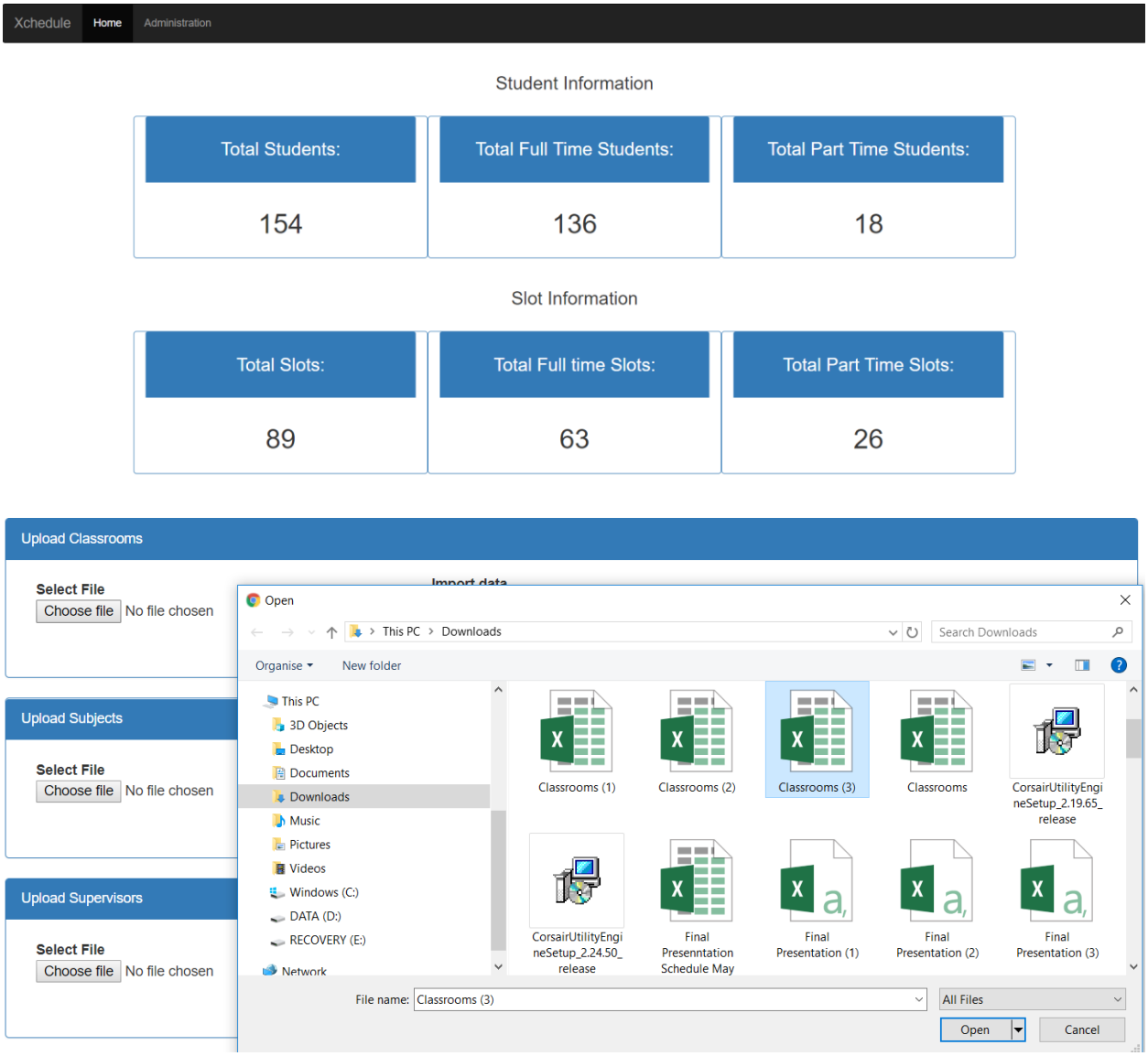
- Login
- Invalid login validations and SQL injection attempts
- Input validations
- Data successfully being store into the application after user submits the data.
- Algorithm validation – Ensuring supervisors were not double booked.
- Exportation of data. – Ensuring the spreadsheets were being exported correctly with the correct formatting and data.
- Dashboard contained the correct data regarding supervisors, students etc.

The images below illustrate the functional testing carried out.




Admin
Username
Administrator
Password
Login


localhost says
Invalid Username or Password. Please try again.
OK
Admin
Username
abc' OR '1 ='1 or 1'
Password
Login


Xchedule
Home
Administration
Student Information
Total Students:
154
Total Full Time Students:
136
Total Part Time Students:
18
Slot Information
Total Slots:
89
Total Full time Slots:
63
Total Part Time Slots:
26
Upload Classrooms
Select File
Choose file
No file chosen
Upload Subjects
Select File
Choose file
No file chosen
Upload Supervisors
Select File
Choose file
No file chosen
Open
This PC
Downloads
Search Downloads
Organise
New folder
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Windows (C:)
DATA (D:)
RECOVERY (E:)
Network
Classrooms (1)
Classrooms (2)
Classrooms (3)
Classrooms
CorsairUtilityEngineSetup_2.19.65_release
CorsairUtilityEngineSetup_2.24.50_release
Final Presenntation Schedule May
Final Presentation (1)
Final Presentation (2)
Final Presentation (3)
File name:
Classrooms (3)
All Files
Open
Cancel

Conclusions

After evaluating the research and investigation into the structure of the systems architecture, the requirements, the background, the aims and purpose of the project some conclusions occurred. Advantages of the application and disadvantage of the application were identified. As explained in the system evaluation section some opportunities are listed and described and the limits of the application. Below is a summary of the conclusions that occurred.

Advantages

- The application reduces the administration process of the administrator when trying to create schedules for academic lecturers and students.
- The application is web based so no need for software installation onto the users' local device.
- The application follows standard web application structures and architecture which should reduce the number of issues while development is underway due to resources available online.

Disadvantages

- The application is required to be fully developed and completed in a 6-month time frame with only one developer working on development in all areas. This has limited the number of functional features inspired as the time frame restricts the amount that can be developed.
- The application is structured towards a small number of formats regarding universities layout for conducting student presentations. However, the application will have the capability to allow more structurers that universities nationwide have adapted to individually.

Opportunities

- The application has many areas of growth identified. Opening it up with other formats so it can allow several types of institutions such as constructing company seminars or organizing job interviews for recruiters.
- The application has potential to be integrated with universities systems, which would drive the data better in terms of all the input data needed in the first release that needs to be uploaded by the administrator. Whereas, if the application was connected and has access to the data stored for the university, this process would be eliminated, and the resulting schedule would be instant.

Limits

- The applications time frame for development.
- Developer restriction to one.

Further Development or Research

The systems evolution should allow the application to expand and take on new API's and it can be incorporated into educational systems used by universities worldwide.

If the system was enabled and implemented in a universities system and had access to the databases for all student and lecturers' timetables and all the universities rooms and assignments of when the staff and facility is free to use, the application would only need the input of the students needed to attend. The application would then instantly be able to generate a schedule. This would open the application to be used by all the staff of the university to allow all lecturers create schedules for their classes in the matter of seconds.

The application would only need one administrator to ensure all the logic was implemented into the database schema correctly and maintain sufficiently.

The application could then have additional features implemented once it has access to the universities data, such as creating terminal exam schedules, allocate student supervisors going by student's degree and the lecturers top 3 specializations for assignment.

The application can be expanded to reduce all the administration process needed daily for creating and organizing these events and can reduce the work load of a lot of faculty staff involved in the process.

The system has a lot of potential to grow and if developed and considered with the right functions in mind it could potentially be a dynamic application that can facilitate thousands of different formats used in different universities.

The application should have potential to grow outside of the educational region and be up taken by companies that may conduct hundreds of interviews a year or organize external events that need scheduling. This application has a lot of potential and has the capacity to grow bigger and take on more new and useful features for the educational industries and potentially become a huge aspect of running universities.

Some disadvantages of the project are the time scope of how long there is to develop the application and there is only one developer working on it at any given time for all the research, design, testing and developing.

Appendix

Project Proposal Including Project Plan

Goals and Objectives

Goal One

The first goal is to research, design and development a web-based application wherein a 4th Year Project administrator can create a schedule where students present on a certain day and time to two examiners.

Objectives

- Research the technologies needed to implement a Web based application that will allow a login and allow an end user to input data needed to create a presentation schedule.
- Research into scheduling techniques and approaches to build up the correct logic and use the correct logic implementation and constraints on it.

Goal two

Develop the application to be dynamic and allow flexibility on different aspects of the presentation such as the number of students, the number of examiners and the number of days for the presentations including the times of the presentation.

Objectives

- Develop extensive algorithms and logic to enable the administrator to enter in different data for the requirements of the universities which may differ from different universities.
- Examples would be the number of days' presentations are held might be 2 for one module or 5 for a different module with a mix of different numbers of parameters such as number of examiners and the amount of time allocated for the presentation.
- Research what data is needed to cover all the aspects of a presentation schedule.

Goal three

Develop a web application scheduler that has validations for conflict detection. This will ensure that there are no double bookings for an examiner or that an examiner isn't booked in for a time slot that they are not available for and that each examiner is scheduled fairly and evenly and no one examiner has an extensive amount of more presentations to examine than other lecturers. (Ratio per examiner should be even and fair)

Objectives

- Develop validations around the applications logic to ensure there are no double bookings for the examiners.
- Develop validations around the logic to ensure no examiner has a vast amount of presentations more than the other lecturers.
- Ensure that the user can select the conditions that suite the module the best.
- Ensure the user can enter in the data required for the application to run the logic for the validations.

Background

The initiative for this project is due to the struggle academic lecturers face each year trying to organize and hold presentations for the final year students. Part of the final year curriculum is to develop a software project in the case of the computer science students.

These software projects need to be demonstrated twice throughout the year. Once half way through the first semester where the students will demonstrate their research and a prototype to two different lecturers. Their supervisor and an additional lecturer of the college who will examine and grade the presentations.

The second time that the software project needs to be demonstrated is at the end of the second semester, when the project has been completed and delivered. This is demonstrated by the students to two lecturers (not including a supervisor for the student).

Several rooms are booked and used for the presentations for the entire day over a 4-day period.

The head of the software project module needs to write up a schedule that suits the lecture and the supervisor of the student presenting. Trying to ensure that no lecturer or supervisor is booked twice for a different time slot. Ensure that the lecturers are getting a break between teaching their students and marking the presentation and that each lecturer is booked fairly, and no lecturer has a higher number of presentations in comparison to other lecturers.

This can be time consuming and difficult to accommodate all the lecturers and keep everyone satisfied.

By implementing a service that allows the head of the module to input the data needed for the lecturers such as timetables for them to identify when they're free. Also, the student details such as name, stream position etc.

Once the input data has been entered the service will be able to run business logic on the data and output a schedule that covers all the students that need to present and ensures that the lecturers are available for the time slot for the student for both the midpoint and the end of year presentations. By making this service dynamic it can also be used for different types of presentation throughout the year for any situation. This will make it easier and more efficient and reduce the amount of work needed to organize the presentations for the head of the module.

Technical Approach

This project will be hosted on the web, to accomplish that it will need to be hosted on a server and it will need to be able to contain a large amount of data when the user inputs it, so it will need to be performant when it comes to memory, to accomplish the best result I will consider which is the best service that can achieve this.

Research into how web services can handle spreadsheet data and output a well-designed HTML table to allow the user to either export the processed data into an excel spreadsheet or a printable version. Consideration of open source API's for this requirement to analyze what external tools and frameworks are available to use that can be taken advantage of to reach a high standard.

There will be a need to discuss some business logic with the head of the software project module to design the business logic rules that will need to be implemented and work close with them as they will have the best input as they must experience the difficulty with this on a yearly basis and will have real life experience.

For research material, there is a large amount of available resources online and in academic text books.

The first stage will be allowing an end user to input data. Once that is implemented, the logic that is driving the schedule can be developed and for the final stage develop the GUI and have lecturers test the service.

Once the above has been completed, then some minor enhancements could be considered that could make the service better and improve technical aspects such as speed and performance.

Existing similar products on the market

There are a variety of similar products available on the market today but from some the research carried out there are few that will contain the same functionality as the application that has been proposed.

Some of the existing applications are:

- Assisstant.to
- Boomerang Calendar
- Celendy
- FreeBusy
- Doodle
- Rally
- Time Bridge

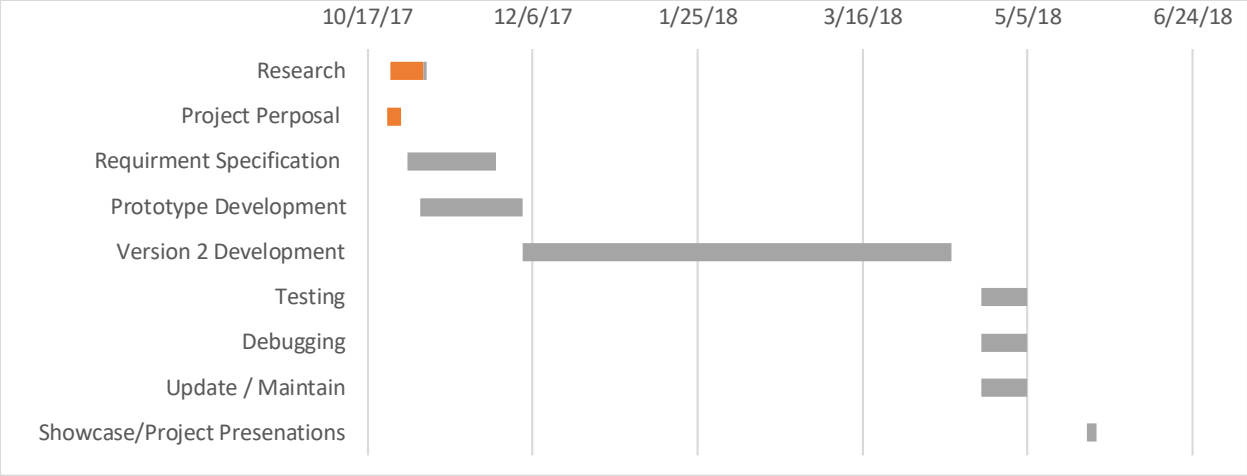
These are great applications, but they do not remove the need for an administrator to manually go and identify current timetables for staff and still need to go and create a schedule, the only service this applications offer is storing it onto the application in a nice format and integrate it with Gmail.

The application that is proposed for this will eliminate the need for that and offer more than most of the applications available today. The application is different and stands out and has far more capabilities.

Special resources required

For this project, there is no special resources required other than the use of actual academic resources such as previous schedules from previous years and a sample of a colleges timetable.

Project Plan



Technical Details

To accomplish the goals set out for this project a variation of libraries to incorporate with JavaScript and PHP and JSON. Various coding languages that will be used include CSS, PHP, HTML, JSON, ASP and Java.

The libraries that will be taken advantage of will be those that can extract data from spreadsheets and output a spreadsheet with the completed presentation schedule. From primary overview of research there are a few that could be useful;

- Tableizer- Used for forming spreadsheet input into a HMTL table.
- Microsoft API's for embedding and export data into Office
- Java libraries for transforming data into XML or JSON back into useable computed data.

Evaluation

For technical testing, input data such as a spreadsheet with some mocked data containing mocked supervisors and student details along with some information around timetables for lecturers.

Testing at a high level will be to check the data and ensure that the data is correct what is being searched upon. For Example, to verify the application knows that a lecturer is free at a given time, write a unit test to verify that it is what it returns. Write Unit tests to verify that the data is correct and get the count of students, the count of supervisors, the limit of time slots allocated to each of the supervisors. Run another test to verify all students have a time slot allocated to them etc.

For a system test, input pre-determined data that will be done using the same logic as the application and run the application and verify that when raw data is input, it runs the logic correctly and outputs what is expecting to be output.

In the beginning, use a small amount of input data and then continuously add more to it. The key to knowing that the system is working is the ability to add any amount of constraints such as changing the number of lectures expected to attend the presentation or the length of time per presentation and it should still output a fair schedule for the lecturers and still validate that all the students have a time slot allocated to them.

Access is available to real life end users for this application. Requesting lecturers to test the system and use the system and obtain feedback regarding improvements that could be made to the application or what they like about it or if it worked correctly and as expected.

Ideally, 2 weeks prior to the end date for testing and completeness to ensure the final deliverable is complete and fully functional with bug free code base. If testing and bug review is complete enhancements to it and leave time for further testing.

Appendix

Project Analysis and Design Specification

Table of Contents

Document Version Control

Version	Implemented by	Revision Date	Approved By	Approval Date	Reason
1.0	Liam Halpin	30/1/2018	Eamon Nolan	5/2/2018	Initial Design Draft
2.0	Liam Halpin	7/2/2018	Eamon Nolan	21/2/2018	Added ERD to the document which was missing in version 1
3.0	Liam Halpin	21/2/2018	Eamon Nolan	21/2/2018	Changed ERD Diagram to use crows foot notation instead

Introduction

Purpose of the Project Analysis and Design Document

The purpose of the Project Analysis and Design Document is to document, describe and illustrate the design of the project and all the analysis gone into the research and development stages of the application. The design of the application will include architecture design which includes all the different aspects of the architectures such as the software architecture including the frameworks and libraries called within the application. The hardware architecture of the application including where the application is going to be stored and deployed such as the cloud architecture or the servers. Performance and security architecture needs to be described, this will give a good overview as to how the application is going to be secured and that thought has been put into the designing stage for the project which in recent years has been proven to be a vital process in application development. Performance architecture to improve the speed and output of the application and how that is going to be achieved. This document will also describe the logical views and the some of the pseudo code related to some of the constraints of the application.

Documenting all the above is used for illustration and sharing knowledge of what the project entails. It demonstrates to the development team the hardware and software that will be used to accomplish building the application in both a high level over view and in-depth overview of the design of the application. It can also identify dependencies for other dev teams and stakeholders that will need to collaborate on the project. It can demonstrate the actual layout and UI of the application which can give PM's a broad overview of what to expect.

General overview and design guidelines and approach

The design of the application should contain a user friendly and easy to navigate UI for the user to use. The technologies to achieve this will be a variation of web technologies such as HTML, CSS and bootstrap with possible entries of JavaScript. For the Communication between the server and the database back and forth from the client side, PHP will be used to send queries to the Database and Server and for updating the records of the database.

Recent release of the OWASP-TOP 10 2017 will be used as a guidance when considering security threats and prevention. This may need the use of some libraries or frameworks that may have built in solutions already considered that have been developed already.
The focus of the application is to reduce the workload of an administrator for a college trying to organize schedules and to ensure that the application is outputting accurate information. This means validations and testing will be crucial in the product design and should potentially include a suite of automation tests to ensure the data is correct.

Assumptions, Constraints and Standards

Some standards that need to be met during the development of the applications can be referenced from the W3C standards and guidelines. This is the current standards for a broad range of areas such as how code should be written, including the style and separation of classes and files and to how the web application should function. Some of the standards that will need to be followed when developing the application can include but are not limited to: -

- HMTL&CSS
- Graphics
- Mobile Web
- Accessibility
- Privacy

During the development of the application it is vital to follow the standards to develop a functional and compliant application.

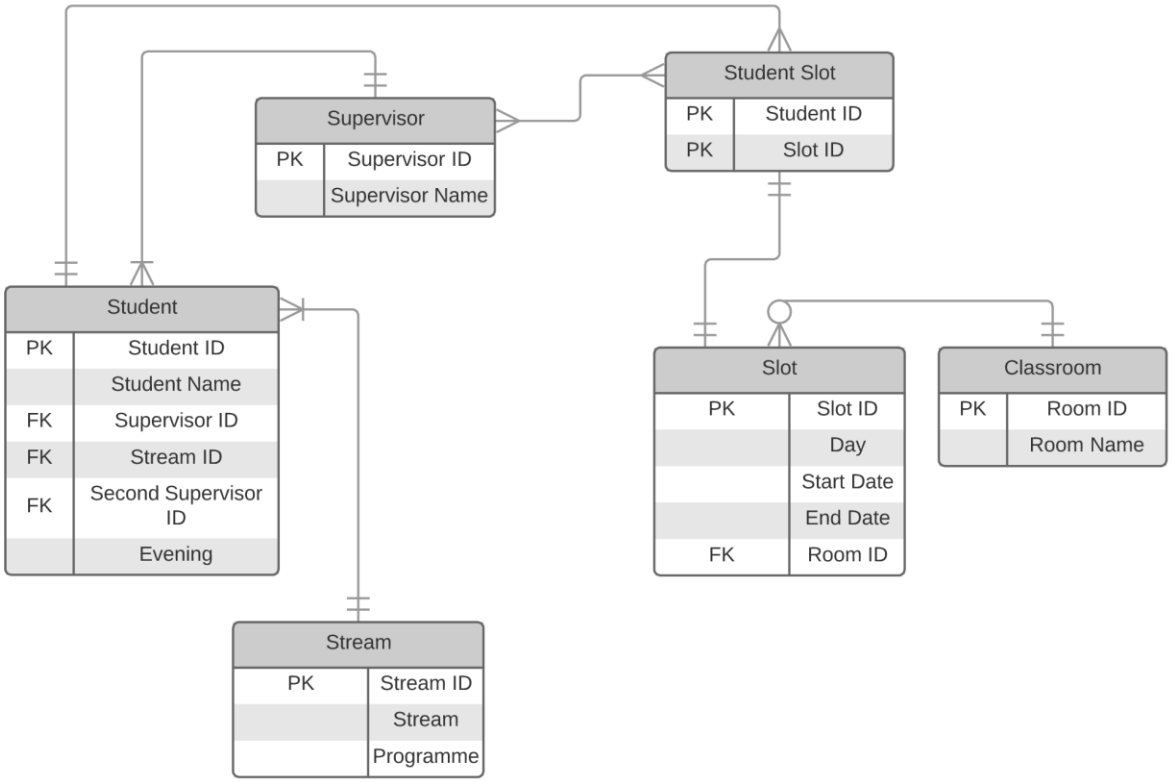
Some constraints that may occur is needing to upskill on some new technical areas or at least increase the knowledge or the learning curve for the technologies to implement the application.

Architecture Design

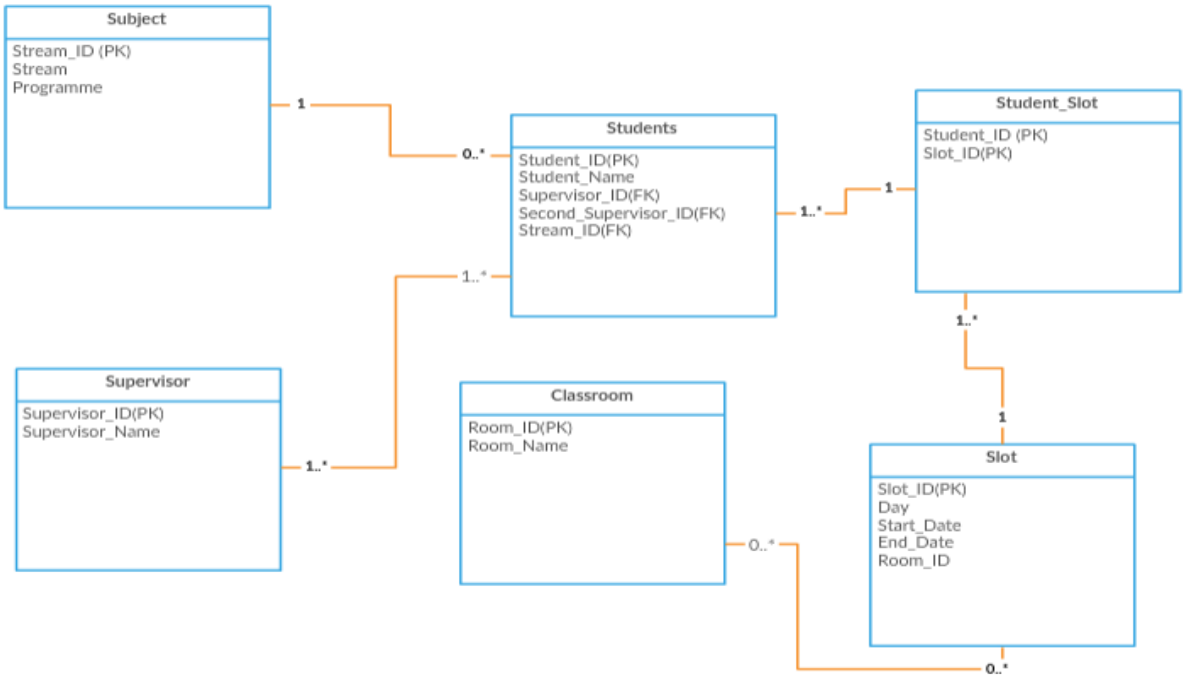
Database design

From the database design illustrated below, the database will be a relational database design and each table will be related to another in one way or another via primary key to foreign key relationship. The reasoning for this design is so it ensures that the validation algorithm can be implemented. The next design will have a timetable for the college linked to the rest of the scheme for timing and scheduling around the lectures timetables which will be used to create mid-point presentation. Once the database schema contains the data for the college timetable the database clauses will be able to restrict certain conditions to ensure supervisors and lecturers are scheduled evenly, haven't been double booked for the same time for a different student and that they are not acting as the main supervisor and the second or third examiner. The database above will be implemented using SQL and PHP will handle communication from the server side. The ability to insert variables to suit the software project admin will be taken in from the user interface and used as part of the logic for creating the schedule. Once all the data has been entered, views and temporary database tables can be created to manipulate the data and output a valid schedule. The values that the database should be able to take as an input should be certain constraints like the number of rooms, the length of the presentation and the days that the software project admin wants to have the presentations held on.

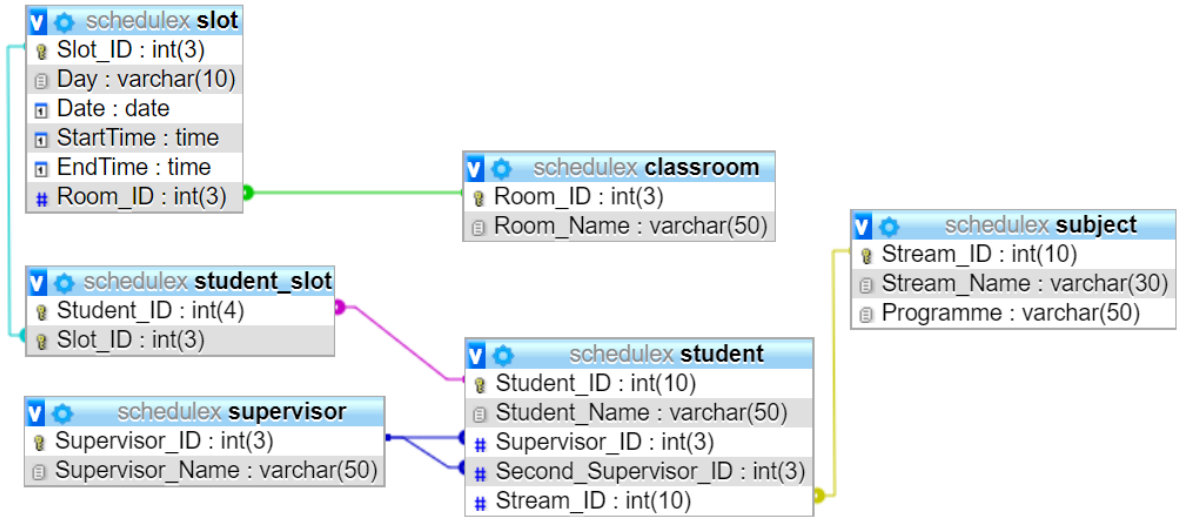
Entity Relationship Diagram Design



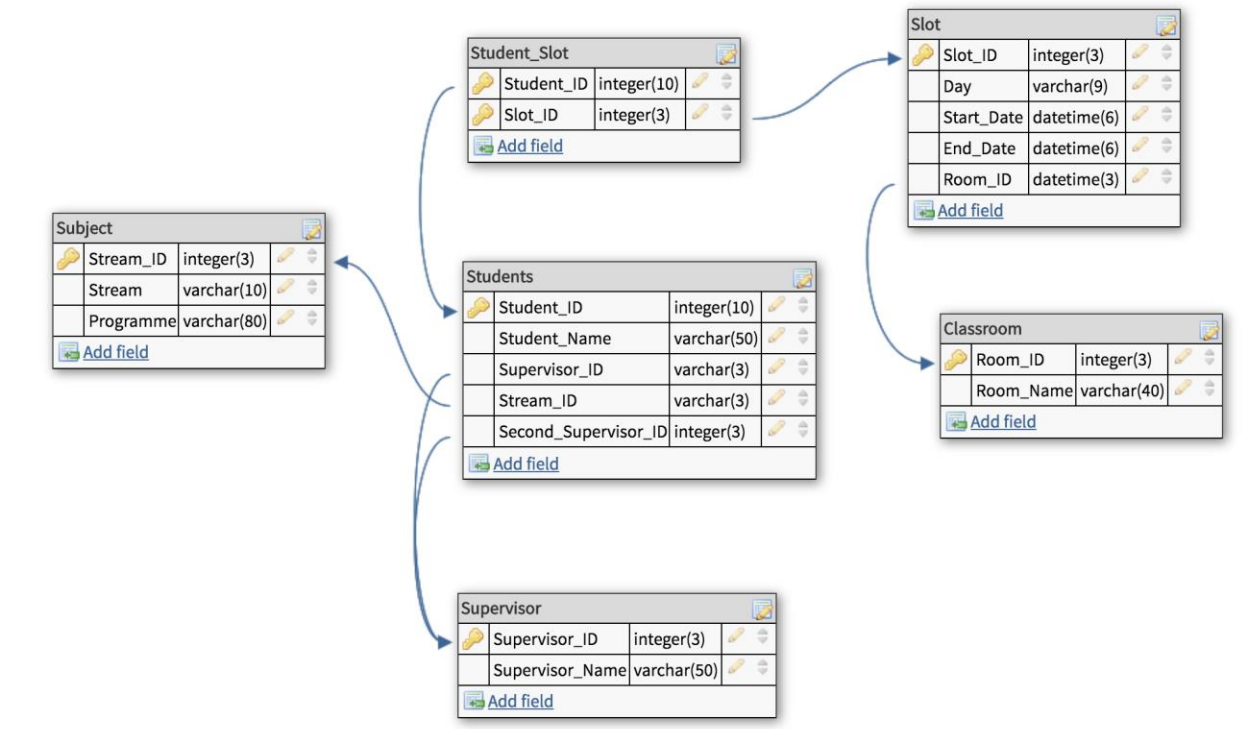
Entity Relationship Diagram



Logical Level ERD

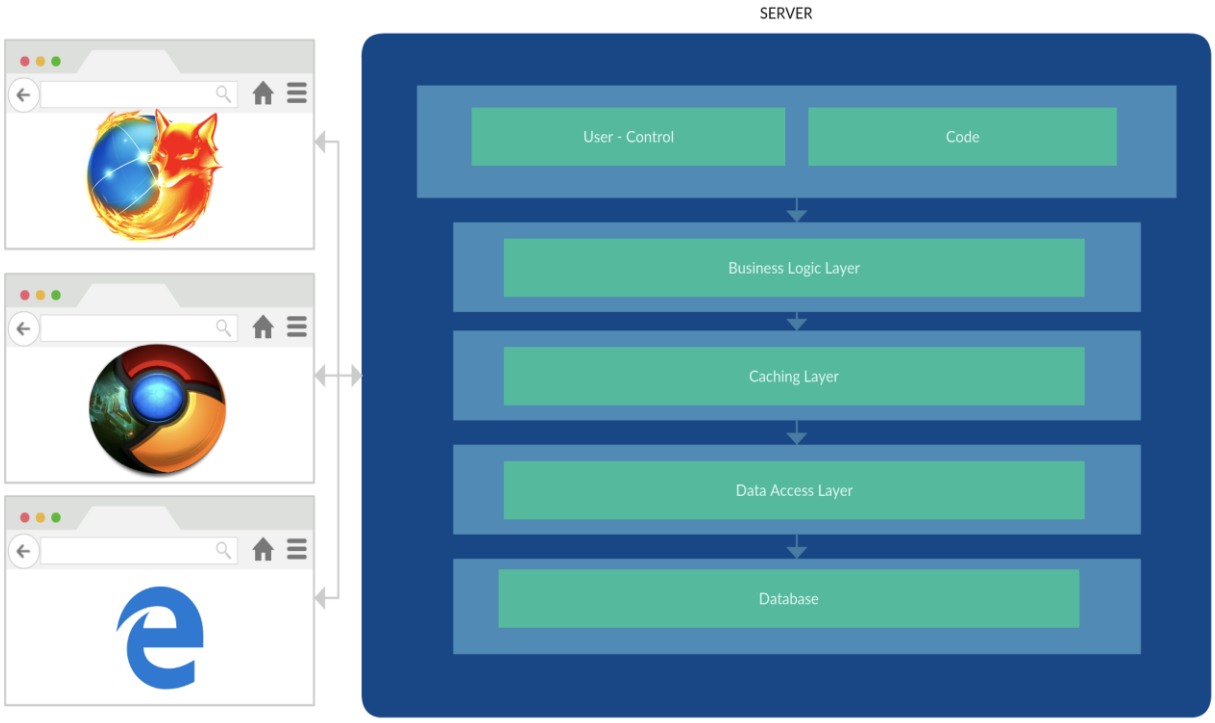


Server View ERD



First Database Scheme Phase Diagram

Logical View



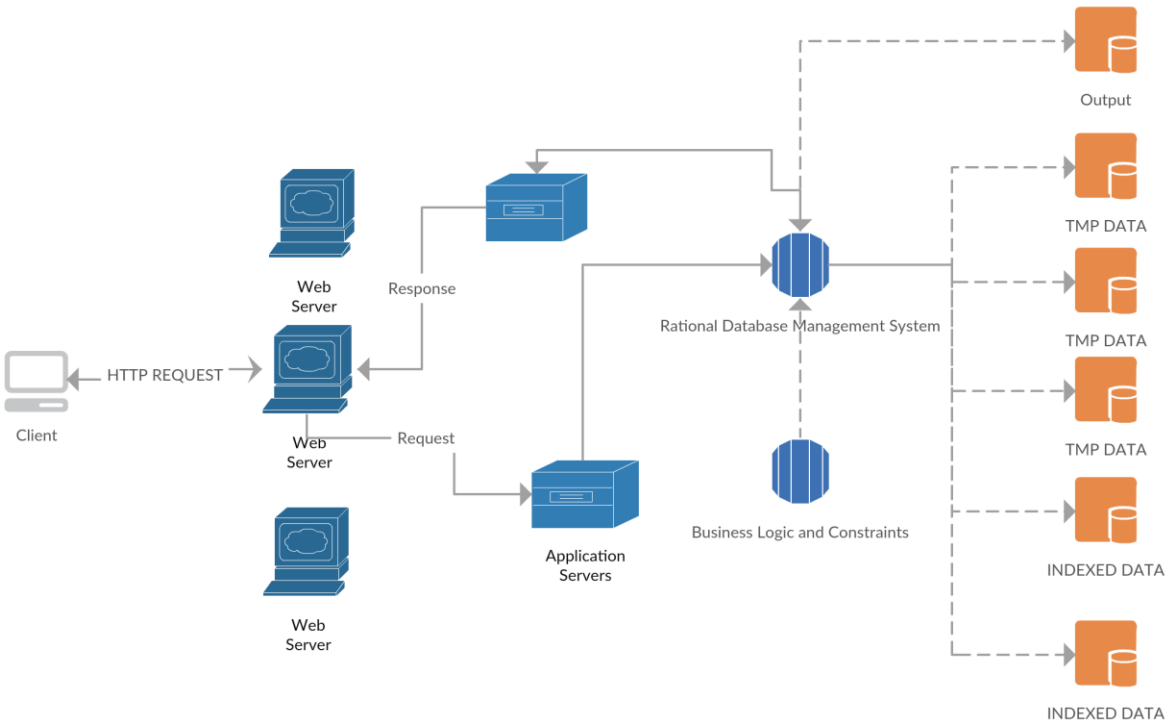
Logical View Diagram

The logical view should complement the hardware architecture of the application. The logical view from the above illustration, demonstrates the clients interacting with the servers and going through the layers in a flowing motion.

It begins with the client accessing the presentation layer. The presentation layer will contain the user controls, which is the actions that the user will perform on the application and the code that is running and creating the UI. The presentation layer will be interacting with the business logic layer that will contain all the different flows of the application. Through the caching layer, data will be stored in cached memory, such as certain objects like login tokens and previously worked on items.

The data access layer and the database will receive the information from the previous layer. It will accept the data once the user has access to it and then it will perform all the constraints and validations and store the result into the database which can then be passed back up the layer chain and shown to the user. Simple but effective, this layering approach should support the requirements of the application and should enable the user to execute and flow through the use cases.

Hardware Architecture



Hardware Architecture Diagram

The hardware architecture is like most web applications. The design of this one will have a client in which the HTTP request is sent to the web servers that will be hosting the application. Once communication and access has been made between the client and the server the architecture will then begin communicating with the application servers in which the data structure will be implemented. The application servers will be the servers in which the code and the database management system will be located. The architecture is simple and effective and should cover all the requirements of the project needs and use cases.

Software Architecture

All the software will be running on a server. All SQL scripts, PHP files, JavaScript, CSS, HTML file are all stored on a web server. This will enable a user to access them through a web browser once the device has access to WIFI or is secured through the internet via a LAN. The database will also be deployed onto a web server and will be running from the back end using PHP for communicating back and forth with data inputs carrying the SQL data.

Security Architecture

The security architecture will be based and designed in accordance with the OWASP top 10 2017 release candidate that was officially certified in 2017. The OWASP top ten illustrate the top ten attacks that businesses and organizations suffer from the most and need to put up defenses against. The list is as follows;

- SQL Injection.
- Broken Authentication.
- Sensitive Data Exposure.
- XML External Entities.
- Broken Access Control.
- Security Misconfiguration.
- Cross-Site Scripting(XSS).
- Insecure Deserialization.
- Using Components with Known Vulnerabilities.
- Insufficient Logging & Monitoring.

The architecture should implement and defend against these threats at a minimum or at least defend against the ones that will be the most damaging for the application in question, which will be using SQL, so the list above will apply to the application.

Software Communication Architecture (SCA)

The web application will be hosted on a server, which means there are a few communication methods needed to communicate with the application itself and send transactions between the application and the database.

To communicate from the client side to the server side, the application will be hosted on a server which means it will hold its own domain in which users will send a request to the host from the client side via the internet. The application will be reached and then the user can then interact with the application and access the features once logged in successfully.

Transparent to the user there will be continuous communication occurring to the database. Communication to and from the database will be achieved by using PHP which will work tightly with SQL for CRUD functionality and retrieving the data for displaying.

Performance

The performance metrics of the web application will be based on several aspects including the response time when sending the http request to the server from the client side. How quick can the application process the data that has been entered and send to the database and store it in

preparation for display. The application will need to validate data and ensure the data has been processed correctly and stored correctly.

With the application storing data after schedules have been created and saving them to be accessed in the future, it may begin to slow down depending on how often the application is used to create a schedule. A possible solution for this will be indexing and ensuring the structure of the database has been designed sufficiently. There are several other solutions such as a backup including how much CPU power will be allocated to the database server and load balancing if it gets too much.

Given that one of the main goals of the application is to reduce the work load of a software project admin and reduce the amount of time that is needed to create schedules for students, to present to a board of examiners, the application needs to output the schedule quickly and not leave the admin waiting days for it to build a schedule. This is accomplished by ensuring the validations and logic of the application are designed well and implemented efficiently. This is a main goal of the overall project so it's vital to get it correct as best as possible.

System Design

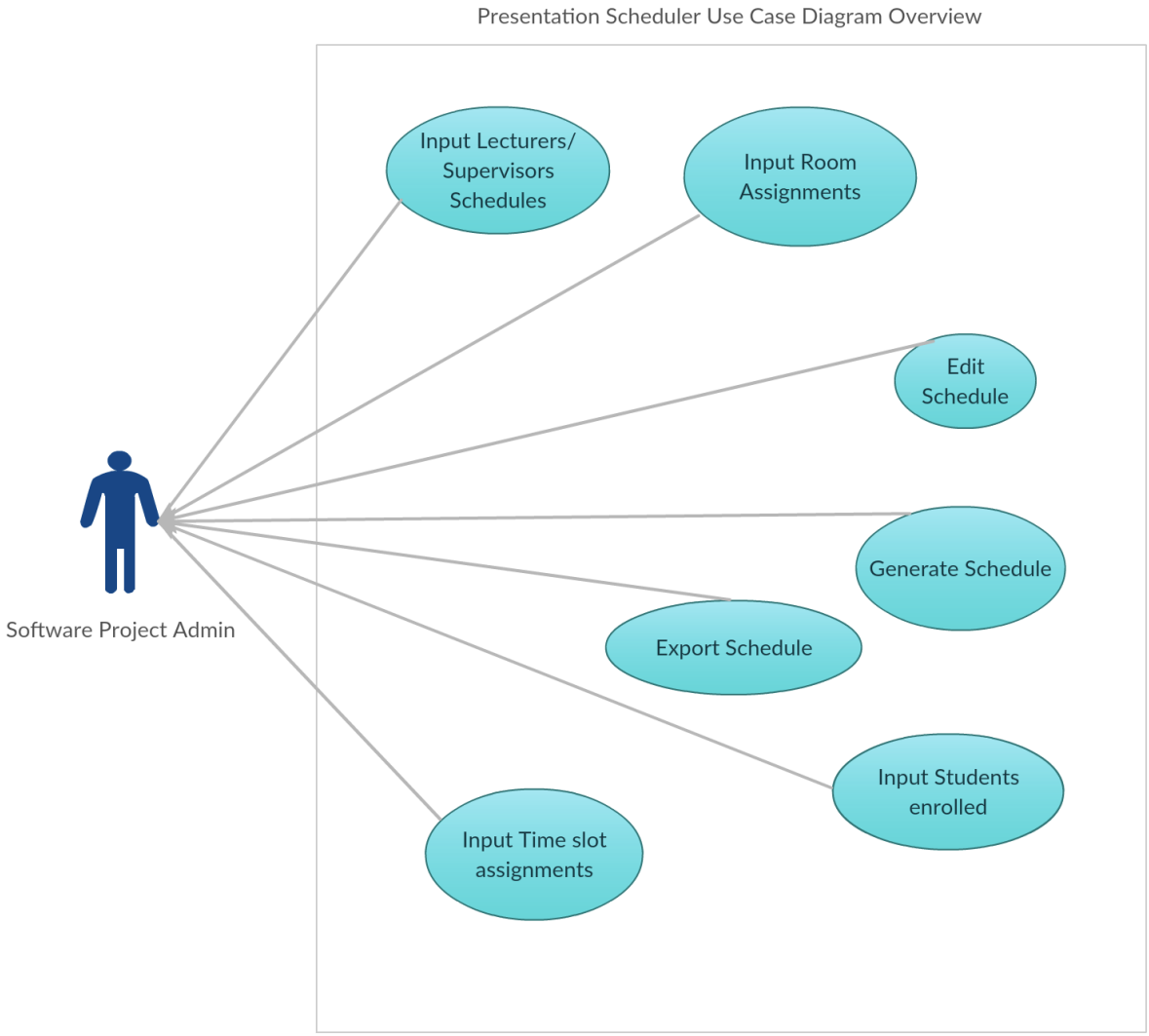
Use-Cases

Requirements

Functional Requirements

The following are the list of functional requirements need when implementing the system. The following are ordered in rank of highest priority. The first requirement will be the highest priority and the second being the next highest so on and so forth.

Use Case Diagram for system overview



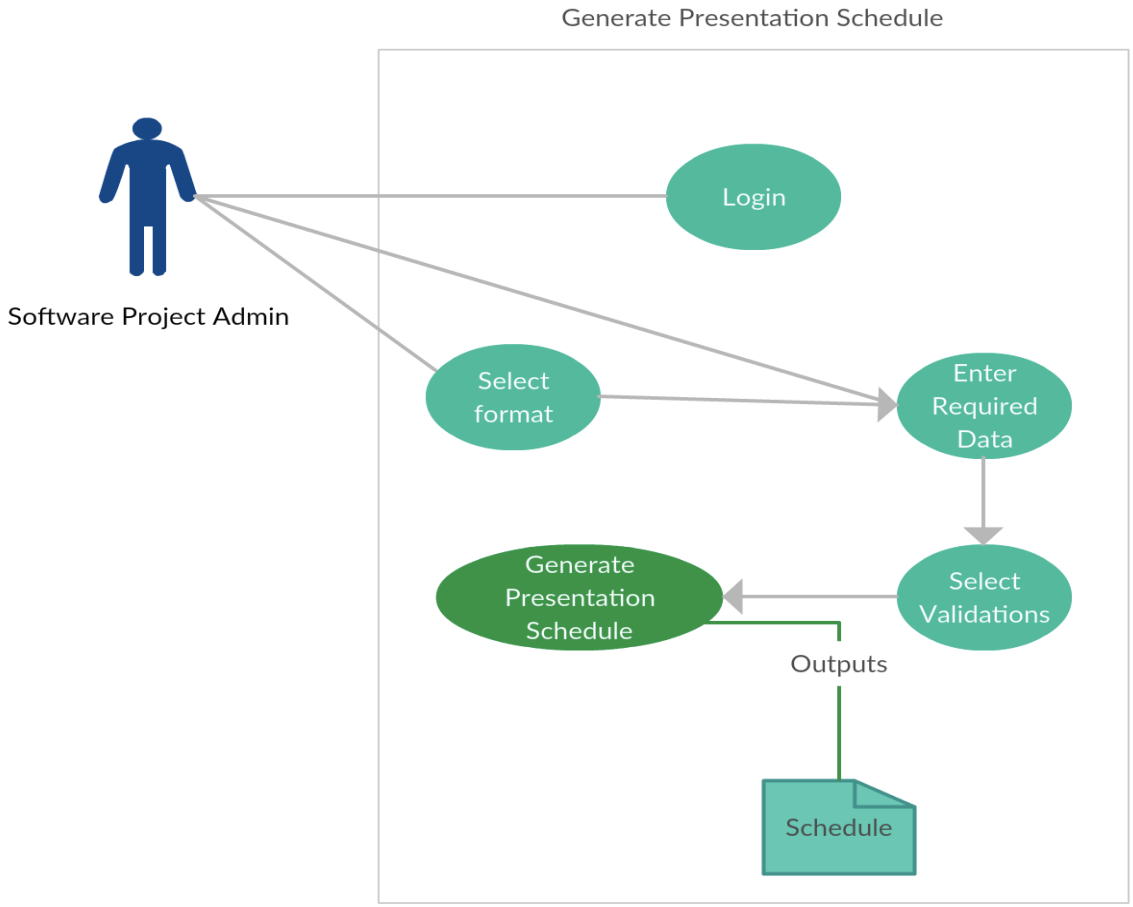
Generate Schedule Requirement

Scope:

Generating the Schedule after the user has entered all the data they required and the validations they desire is the highest use case. This is the core function of the Software and it contains the most complex aspect of the application.

Description.
This use case describes the main functional area of the application. This is the area in which the application will take all the data entered in the system and compute all permutations and validations to output the administrators schedule for the final year students, supervisors and lecturers.

Use Case Diagram for Generate Schedule



Flow Description.
This use case is one of the final slow steps of the application. To execute it, the user will need to be signed into the application and enter all the data required to execute the use case. Although this is the final step of the flow it is the highest value use case as the application will not serve its purpose if it does execute this use case. The Actor will need to log into the application, enter all the necessary data required such as room allocation and time slots etc. Then the Actor will be able to Generate Presentation Schedule and the system will compute the Schedule and output it to the Actor.

Precondition.

The application will be in a waiting state with all the required data stored in the Database before the execution of this use case and it will be ready to compute the schedule once the Actor initiates the use case.

Activation

This use case starts when the <Software Project Admin> Clicks on Generate Presentation Schedule Button on the Applications User Interface.

Main Flow.

9. The system identifies all input data has been entered and is ready to execute the computation of a schedule.
10. The <Software Project Admin> will select 'Generate Presentation Schedule' on screen.
11. The system will then begin designing the schedule.
12. The system will then store the presentation Schedule and wait for the <Software Project Admin> to select view Schedule.
13. The <Software Project Admin> will then select 'View Schedule'.
14. The system will retrieve the presentation schedule and display it on screen to the user.
15. The <Software Project Admin> will then export/save the Schedule.
16. The system will export/save the schedule to the <Software Project Admin> device.

Termination.

The system will present a UI to the <Software Project Admin> asking if they would like to create another schedule or go back to the home page of the application.

Post Condition

The system will enter back into a wait state at the home page of the application waiting to either log out or begin to create another schedule depending on what the <Software Project Admin> wants to do.

Export Schedule Requirement

Use Case.

Once the <Software Project Admin> has completed generating and creating a schedule, the system should offer an option to allow the <Software Project Admin> to Export or save the schedule either allowing them to print it off or export it externally to the device to allow them to make copies for distribution.

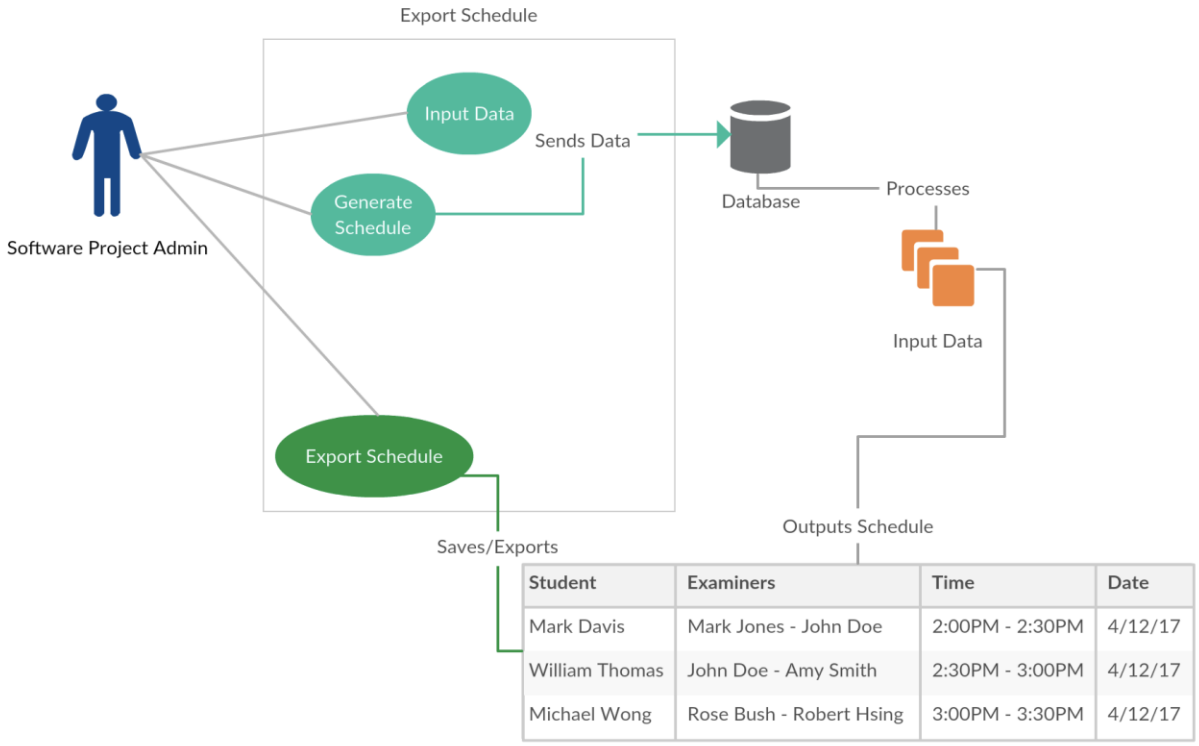
Scope.

The scope of this requirement it to allow the user to save the generated Schedule to allow them to distribute it to the lecturers and supervisors and students involved in the process.

Description.

Once the <Software Project Admin> is satisfied with the presentation schedule that has been created by the application the System will offer the option for the <Software Project Admin> to save and export the schedule to allow them to distribute or publish the schedule to the students and examiners.

Use Case Diagram for Export Schedule



Student	Examiners	Time	Date
Mark Davis	Mark Jones - John Doe	2:00PM - 2:30PM	4/12/17
William Thomas	John Doe - Amy Smith	2:30PM - 3:00PM	4/12/17
Michael Wong	Rose Bush - Robert Hsing	3:00PM - 3:30PM	4/12/17

Flow description.

The system will have generated a presentation schedule for the <Software Project Admin> and will display options for the user to execute, such as preview or export. Once the <Software Project Admin> decides on the action they want to execute, the flow will begin to save and export the schedule and save it to the device. This is the last step in the overall flow of the application and is not a mandatory use case each time the <Software Project Admin> uses the application. This means if the <Software Project Admin> wants to preview the schedule and may want to make some additional changes to schedule, the <Software Project Admin> doesn't have to save the schedule once it has been created.

Precondition.
The system will be in a completion state as the schedule has been generated and now it's up to the <Software Project Admin> to decide what they want the system to do.

Activation.
This use case starts when the <Software Project Admin> clicks export on the applications UI.

Main flow

5. The system has identified that the schedule has been created and waits for the <Software Project Admin> to decide the next action.
6. The <Software Project Admin> will then click export on the application's UI.
7. The system will then save a copy of the schedule to the device.
8. The <Software Project Admin> will then have an electronic version of the schedule.

Alternative flow.

Alternative flow 1: User does not want to download a copy of the schedule.

7. The system has identified that the schedule has been created and waits for the <Software Project Admin> to decide the next action.
8. The <Software Project Admin> decides to edit the data for the schedule to add in additional data or time slots or book different rooms.
9. The system will go back to the data input screen
10. The <Software Project Admin> will then input the data
11. The <Software Project Admin> will then generate the new schedule
12. The use case will continue position 1 of the main flow.

Termination
The system presents the home screen to the <Software Project Admin>.

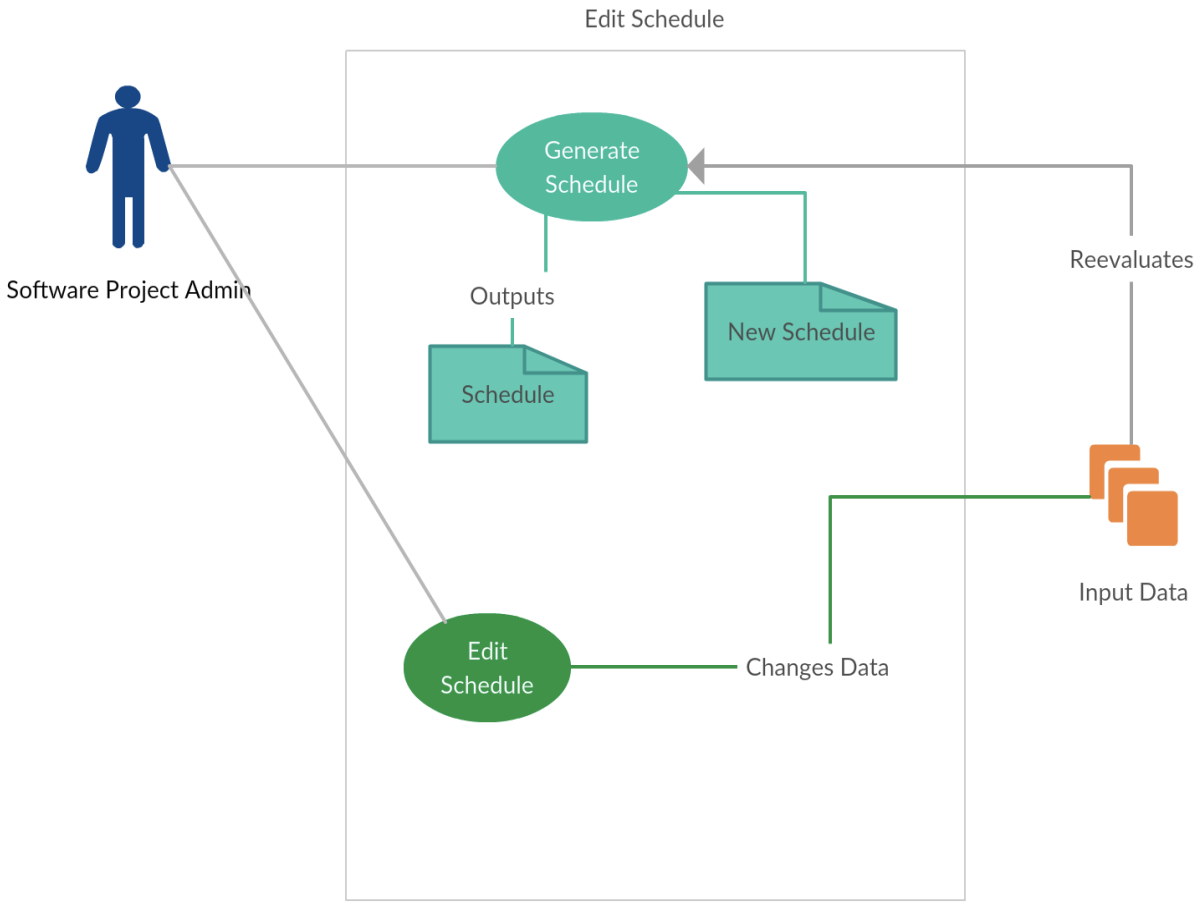
Post condition
The system will go into a wait state again once the flows have been completed.

Edit Schedule Requirement

Scope.
The scope of this requirement is to allow the <Software Project Admin> to make changes to a schedule that has been created by the application if they see an area which won't be applicable for certain lecturers or students. This will allow them to enhance the schedule for optimal results.

Description
If the <Software Project Admin> needs to make amendments to the schedule that the application has created for them due to lecturers not being available for a period or a student can't attend for personal circumstances, then the application will allow them to make amendments to it to make it as ideal as possible for everyone. It will re-enter the data and create a new schedule with the new additional changes.

Use Case Diagram for Edit Schedule.



Flow Description
The system will have generated a schedule after the specified data has been entered by the <Software Project Admin>. There may be exceptions where the <Software Project Admin> may need to accommodate for different circumstances for either a lecturer or a student. This flow will allow them to enter the restrictions to satisfy as many needs as possible and ensure everyone is accounted for and can attend the presentations at the given time. This flow will use the generated schedule

Precondition
The system will be in a wait state. Once the schedule has been generated it goes back to asking the <Software Project Admin> what is the next action they want to take with the Schedule.

Activation
This use case is activated when the <Software Project Admin> selects 'Edit Schedule' on the application's 'UI', which will activate the edit requirement.

Main Flow

10. The system will have generated a schedule.
11. The <Software Project Admin> will inspect the schedule the system has outputted.
12. The <Software Project Admin> will select 'Edit Schedule'.
13. The <Software Project Admin> will select and enter the amendments they wish to include.
14. The system will ask to user do they want to regenerate the schedule.
15. The <Software Project Admin> will then select 'Reschedule'.
16. The system will then generate a new schedule.
17. The system will then output the new schedule.
18. The <Software Project Admin> will review the new schedule.

Alternative Flow

4. From position 9 of the Main flow – if the <Software Project Admin>is still not happy or needs to make some more adjustments, the system will allow the user to 'Edit Schedule'.
5. The <Software Project Admin> will select 'Edit Schedule'.
6. The use case will continue at position 4 of the main flow.

Termination
The system will return to the home screen once the <Software Project Admin> is happy and selects continue.

Post Condition.
The system will go into a wait state until the <Software Project Admin> needs to use it again.

Input Supervisors/Lecturers Schedule Requirement.

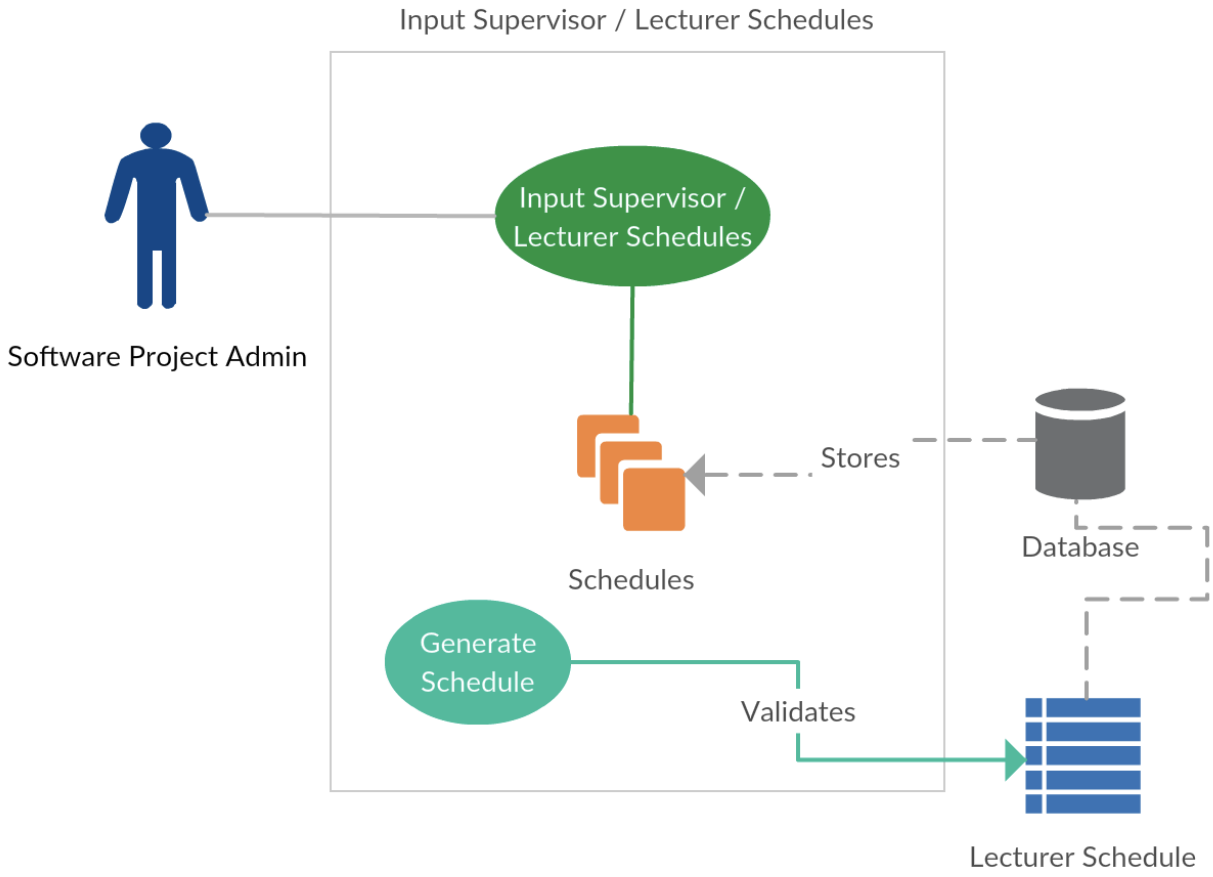
Scope

The scope of this is to get the <Software Project Admin> to enter in the supervisors and lecturers' schedules for their teaching times and classes. This will allow the system to identify where is acceptable to place them. This system will automatically know not to assign a time slot for a lecturer if they are scheduled to teach a class at a certain time. This will ensure the schedule is efficient and correct.

Description

This use case will be used to store that data of the lecturer and supervisor's schedules into the database, these will be then used to identify what times that the lecturers are available to sit for presentation. This will be critical in making the system dynamic and automated and not create a schedule that is unusable to the <Software Project Admin>.

Use Case Diagram for input Supervisors/Lecturers Schedule Requirements



Flow Description.
This flow will be in the beginning of the data set up. Without this data input from the user the application will generate a schedule, but it will place lecturers during times that they are not available which would be disruptive to the entire college. Lecturers will need to miss giving their lectures and they may get an unfair amount of presentations to attend.

Precondition.
The system will be in initialization mode with no data entered into the database. Once the user inputs the data it will be written to the database and stored to be used later in the flow.

Activation
This is activated when the user is requested to enter in the lecturer's/supervisor's data.

Main Flow

6. The system will prompt the user to "Please enter in the lecturers and or supervisors Schedule"
7. The <Software Project Admin> will then enter in the schedules for the lecturers.
8. The system will create a table in the database and store the table into the database.
9. The system will notify the <Software Project Admin> that the data has been stored.
10. The system will bring the <Software Project Admin> to the next process of creating the schedule.

Exceptional Flow.

4. The <Software Project Admin> tries to enter in the data.
5. The system cannot connect to the database.
6. The system will notify the <Software Project Admin> of the error and execute the precautions to take in this event by returning the <Software Project Admin>to the main home page and insisting they "try again later"

Termination
The use case is terminated once the user has submitted the data and the system has stored successfully.

Post Condition.
The system is then in a wait state after the system has directed the <Software Project Admin> to the next stage of the process.

Input Room Assignments Requirement

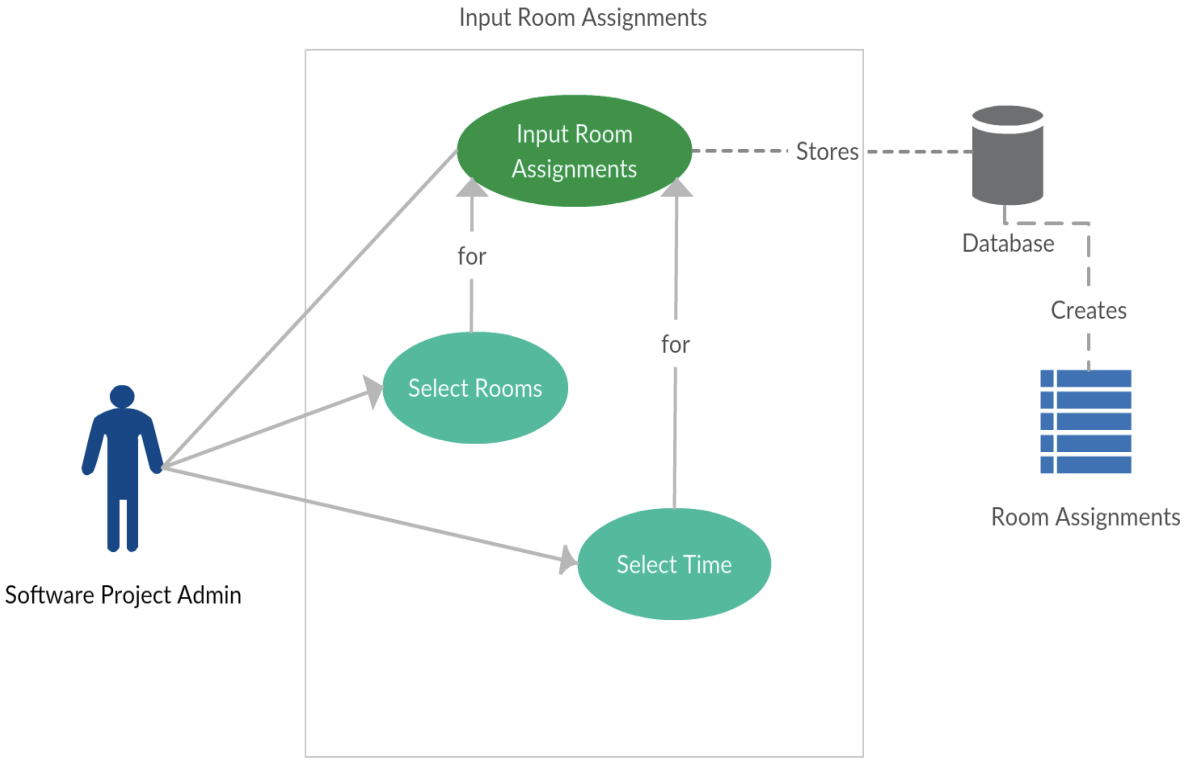
Scope

The scope of this requirement is to get the <Software Project Admin> to enter in the rooms and the time scale they want the presentations to take place in, for example Room 101 from 9:00AM – 5:00PM. This data will be used to write the schedules data for informational purposes.

Description

This requirement is used to obtain the data needed for displaying what room the students will be presenting in and where the lecturers need to be for the presentation. This data is also used by the system for formatting the table that the system will write to.

Use Case Diagram for Input Room Assignments



Flow Description
This flow is part of the data set up that the system needs for writing a schedule alongside the previous requirement “Enter Supervisors/Lecturers Schedules”.

Precondition.
The system will have created the schedules table and will then be left in a wait state for the room assignments input

Activation
This use case is activated after the <Software Project Admin> has submitted the schedules of the supervisors and lecturers and gets directed to this use case.

Main Flow.

8. The system will prompt the <Software Project Admin> to enter in the room that the <Software Project Admin> is booking for the presentation.
9. The <Software Project Admin> will then enter in the rooms that are going to be used for the presentations.
10. The system will then ask the <Software Project Admin> for which time the schedule should accommodate for.
11. The <Software Project Admin> will enter in the time frames needed for the rooms and presentations.
12. The system will create a new database table based on the <Software Project Admin> input.
13. The system will store the data to the database.
14. The system will direct the <Software Project Admin> to the next stage of process in creating a schedule.

Exceptional Flow.

4. The <Software Project Admin> tries to enter in the data.
5. The system cannot connect to the database.
6. The system will notify the <Software Project Admin> of the error and execute the precautions to take in this event by returning the <Software Project Admin>to the main home page and insisting they “try again later”

Termination.
The use case is terminated once the user has submitted the data and the system has stored successfully.

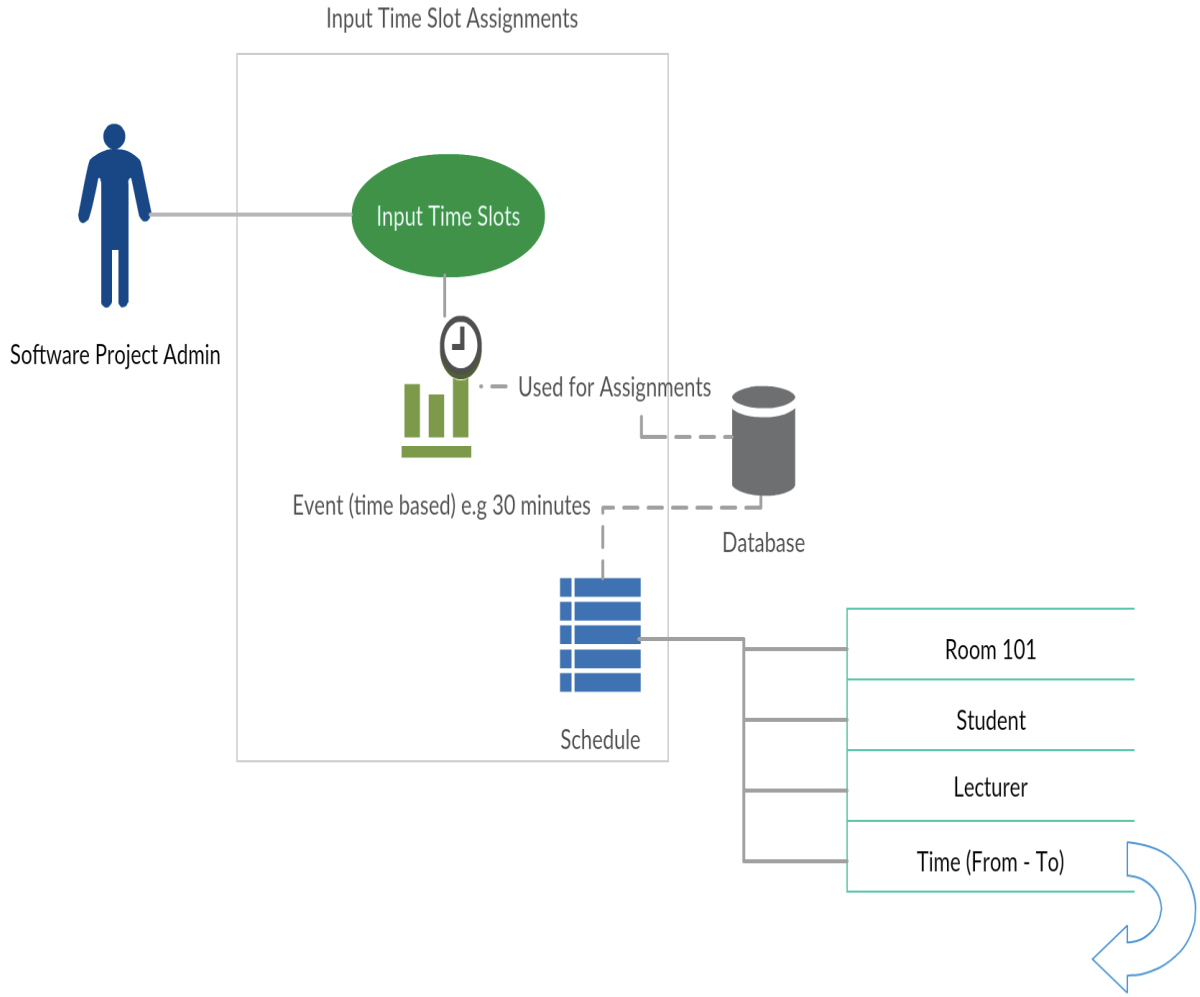
Post Conditions
The system is then in a wait state after the system has directed the <Software Project Admin> to the next stage of the process.

Input Time Slot Assignments

Scope
The scope of this requirement is to instruct the <Software Project Admin> to specify how long each student will have to present to the supervisor and lecturer.

Description
The use of this use case is to get the <Software Project Admin> to specify how long they are allowing the students to present for. This piece of data will be used regarding the time slots allocated for each student. The database will generate each record knowing only to assign each student a specific amount of time to present. The database will know when to stop once a certain time has been reached and presentations have ended for the day.

Use Case Diagram for Input Time Slot Assignments



Flow Description
This use case comes as part of the data set up from <Software Project Admin> and is stored in the database like the other data inputs. Once the data is stored the next use case of the flow can be activated.

Precondition
The system is in waiting state waiting for the input to store in the database.

Activation
This use case is activated after the system prompts the user to enter in the time slots needed.

Main Flow

5. The system will prompt the <Software Project Admin> to enter in a time slot allocation.
6. The <Software Project Admin> will enter the time slot that they require.
7. The system will write the data to the database and store it.
8. The system will redirect the <Software Project Admin> to the next stage of the process.

Exceptional Flow.

4. The <Software Project Admin> tries to enter in the data.
5. The system cannot connect to the database.
6. The system will notify the <Software Project Admin> of the error and execute the precautions to take in this event by returning the <Software Project Admin>to the main home page and insisting they “try again later”

Termination
The use case will terminate once the data has been successfully saved to the database.

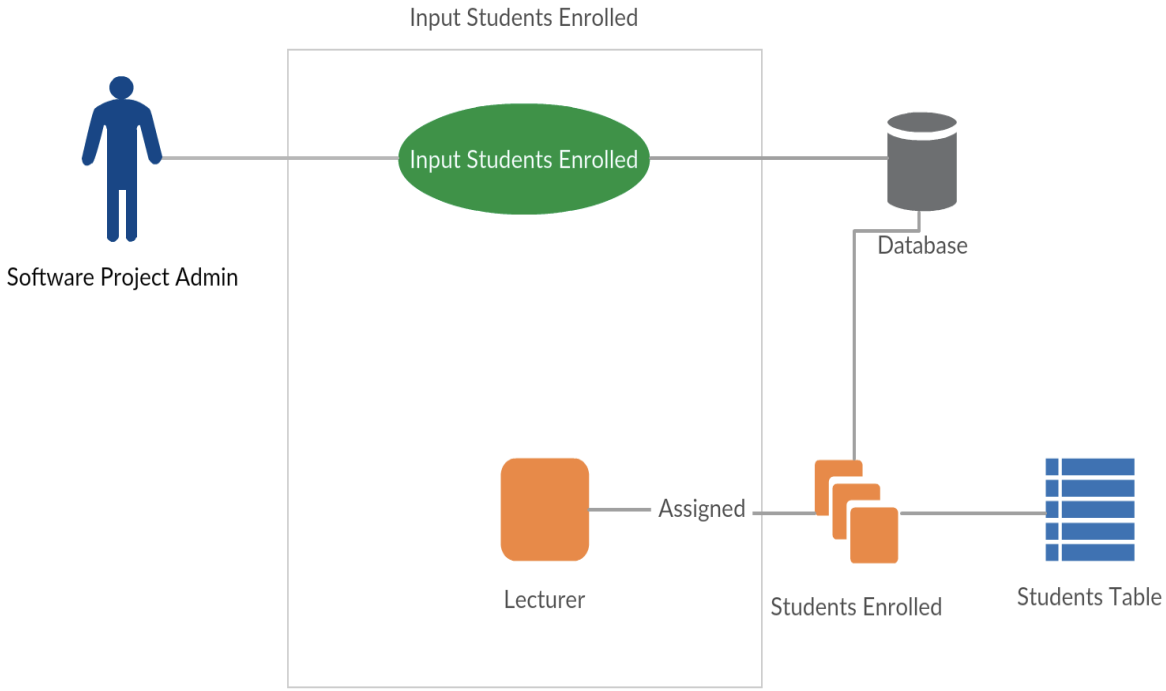
Post Condition
The system will enter a wait state after the use case has terminated.

Input Students Enrolled Requirement

Scope
The scope of this use case is to prompt the <Software Project Admin> to enter in all the students enrolled for the module that need to present their project material.

Description
This use case is used for obtaining more data regarding what students need to be assigned a time slot for presenting. This use case will prompt the user to enter in the data and the system will store the information into the database after creating the table for them.

Use Case Diagram for Input Students Enrolled Requirement.



Flow Description
This is part of the data input flow alongside the other ones illustrated above. This is one of the final pieces of data needed to construct the schedule.

Precondition
The system will be in a waiting state for the <Software Project Admin> to enter in the required data.

Activation
This use case is activated when the system prompts the user for the data.

Main Flow

6. The system will prompt <Software Project Admin> to input the students still enrolled in the module.
7. The <Software Project Admin> will enter in the data.
8. The system will create a table in the database.
9. The system stores the data to the database.
10. The system will redirect the <Software Project Admin> to the next stage of the process.

Exceptional Flow.

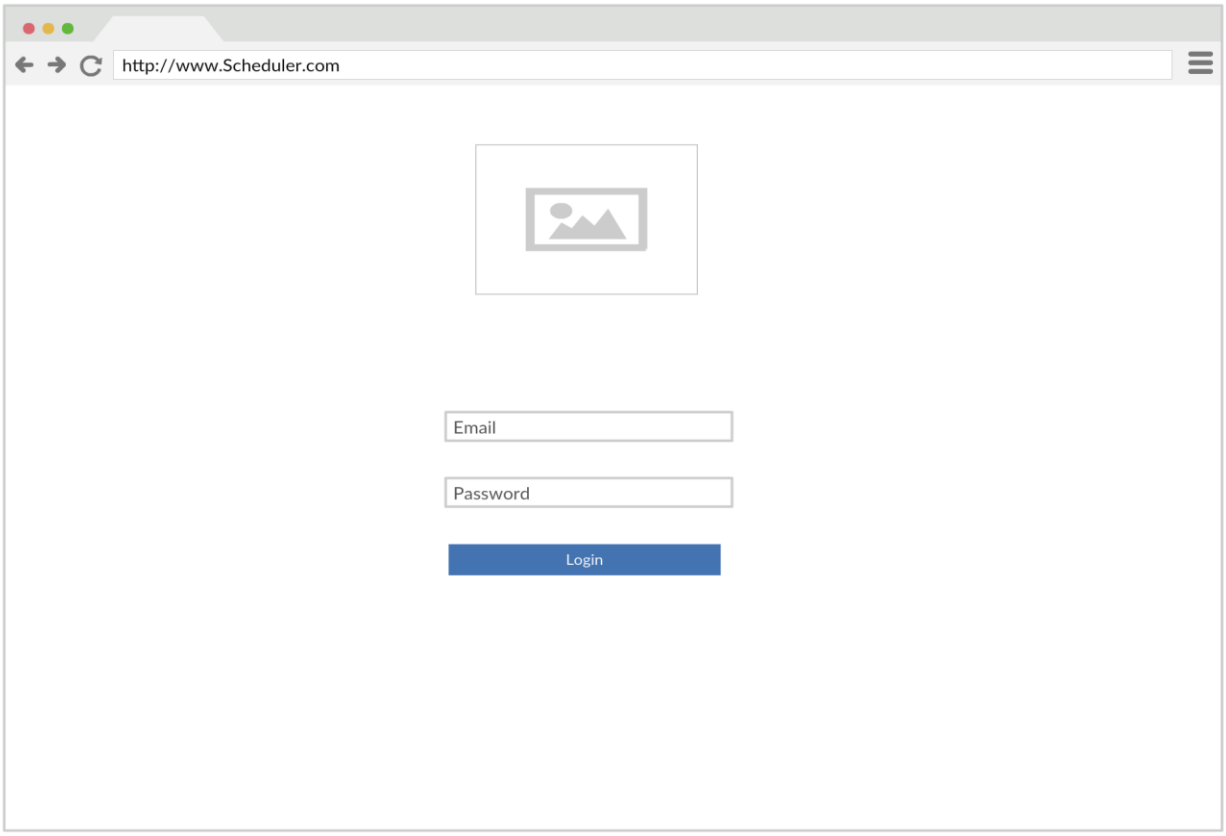
4. The <Software Project Admin> tries to enter in the data.
5. The system cannot connect to the database.
6. The system will notify the <Software Project Admin> of the error and execute the precautions to take in this event by returning the <Software Project Admin>to the main home page and insisting they “try again later”

Termination
The use case is terminated once the data has been stored into the database and the tables have been created.

Post Condition
The system goes into a wait state until the <Software Project Admin> is ready to generate the schedule.

Graphical User Interface

Login GUI



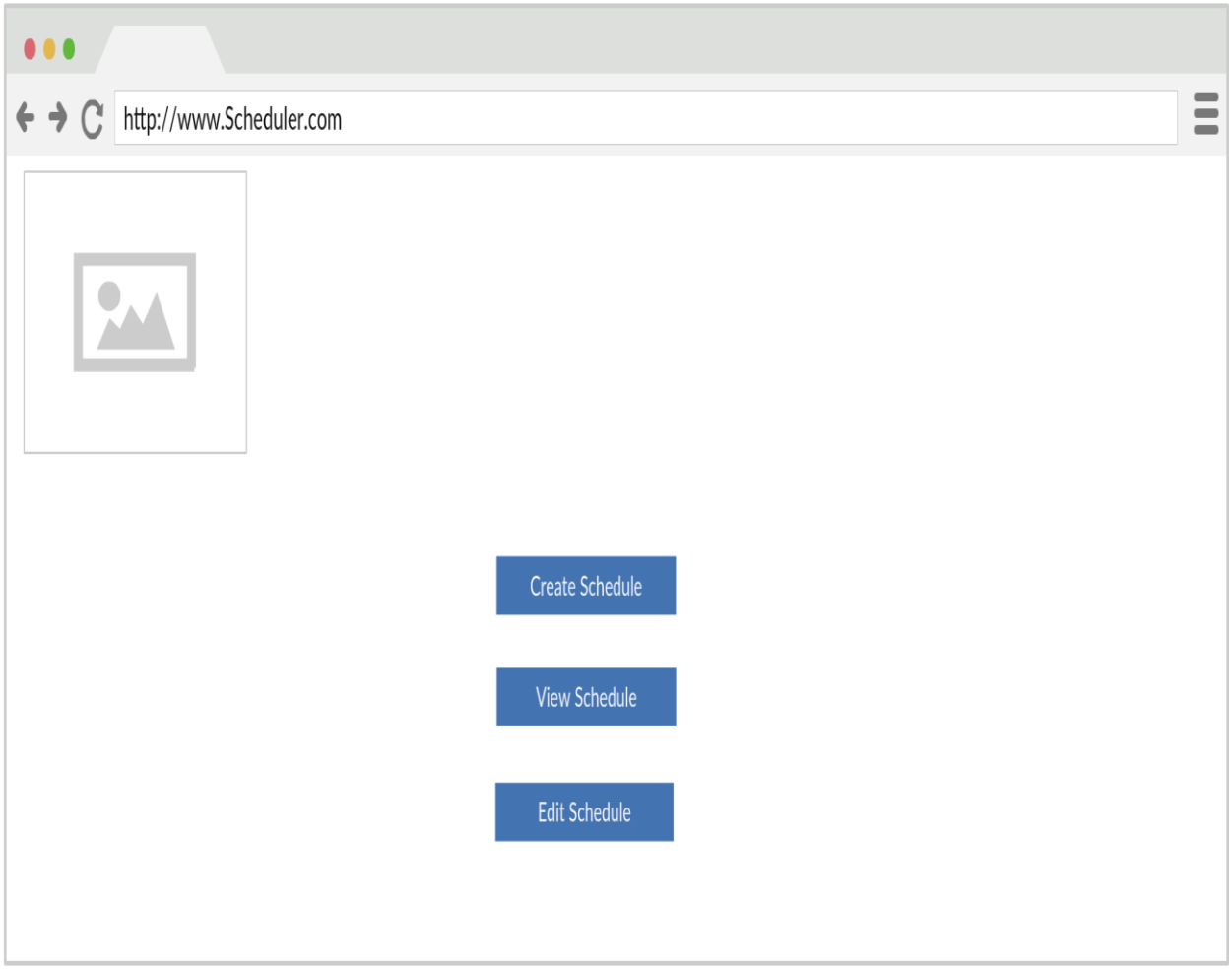
Description of Login GUI

The login page illustrated above will be used for authentication for the Software Project administrator. Creating a login feature to only allow permitted personnel to access the application will ensure no student or standard lecturer will have the ability to access it and amend schedules, dates etc.

In doing this, this could be developed to incorporate external educational systems in which a software project admin could use functionality from Moodle and external data from Moodle to make inputting the data redundant and the application could access enrollments and even extend to allow the college or university to submit its data to the application which would relieve more of the administration process of the end user.

The login page will allow the end user to log in and will also have helpful links to online documentation and other helpful and user tips.

Options Menu GUI



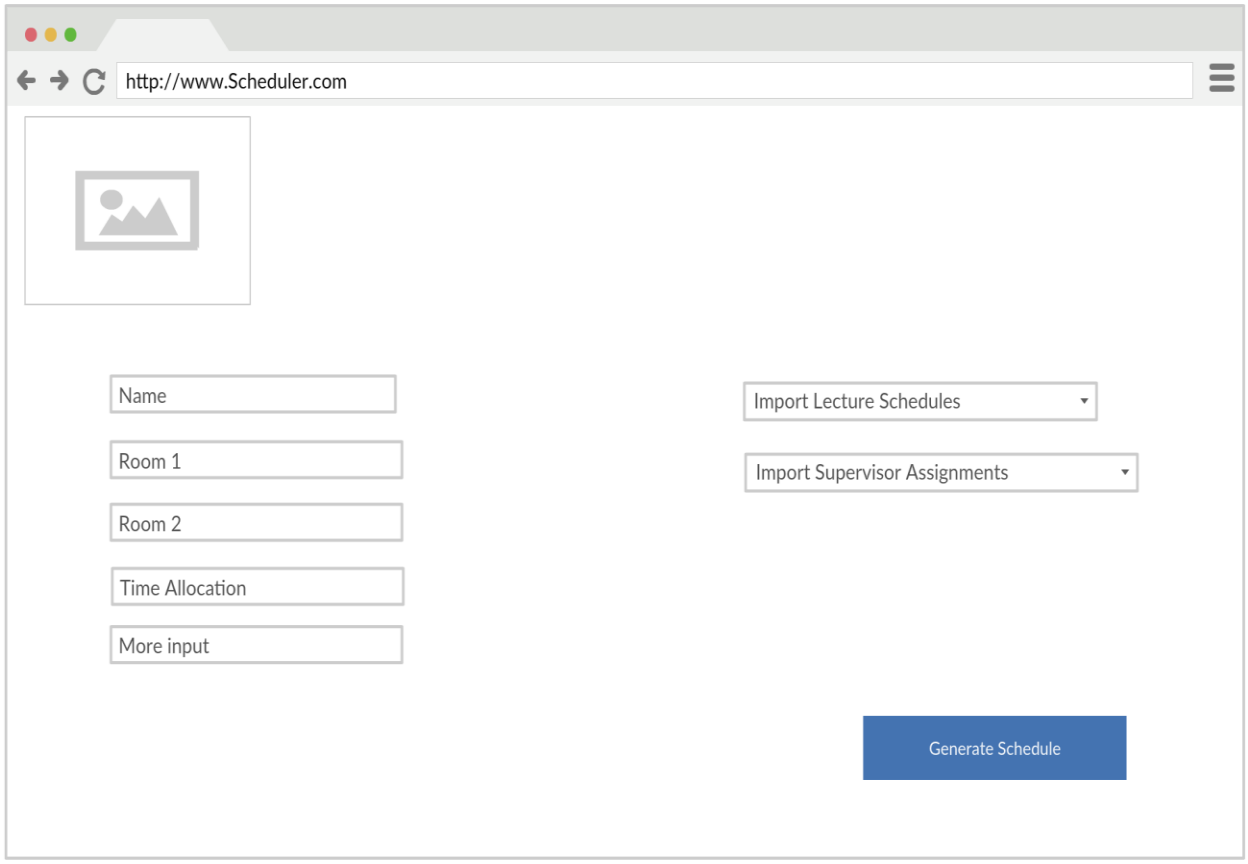
Description of Options Menu GUI

The options page displays the option the application allows the user to act upon. There are several actions that the user may want to take such as “Create Schedule” which will then redirect the user to the processes of the creation of a schedule.

Then located underneath the create option will be the view schedule. The view schedule will offer the user the option to view previous schedules created and from there on they will have the option to amend the schedule or leave it as be or simply download another copy of the schedule etc.

Below the View Schedule option will then be an Edit Schedule option in which the user will select Edit Schedule and then modifications to a certain Schedule structure or make minor adjustments to address some unseen events such as a change of supervisor or a second examiner falls sick or a student receives an extension. This will give the user the ability to change the flexibility of the schedule and more area and power to change and reevaluate the schedule.

Create Schedule GUI



Description of Create Schedule GUI

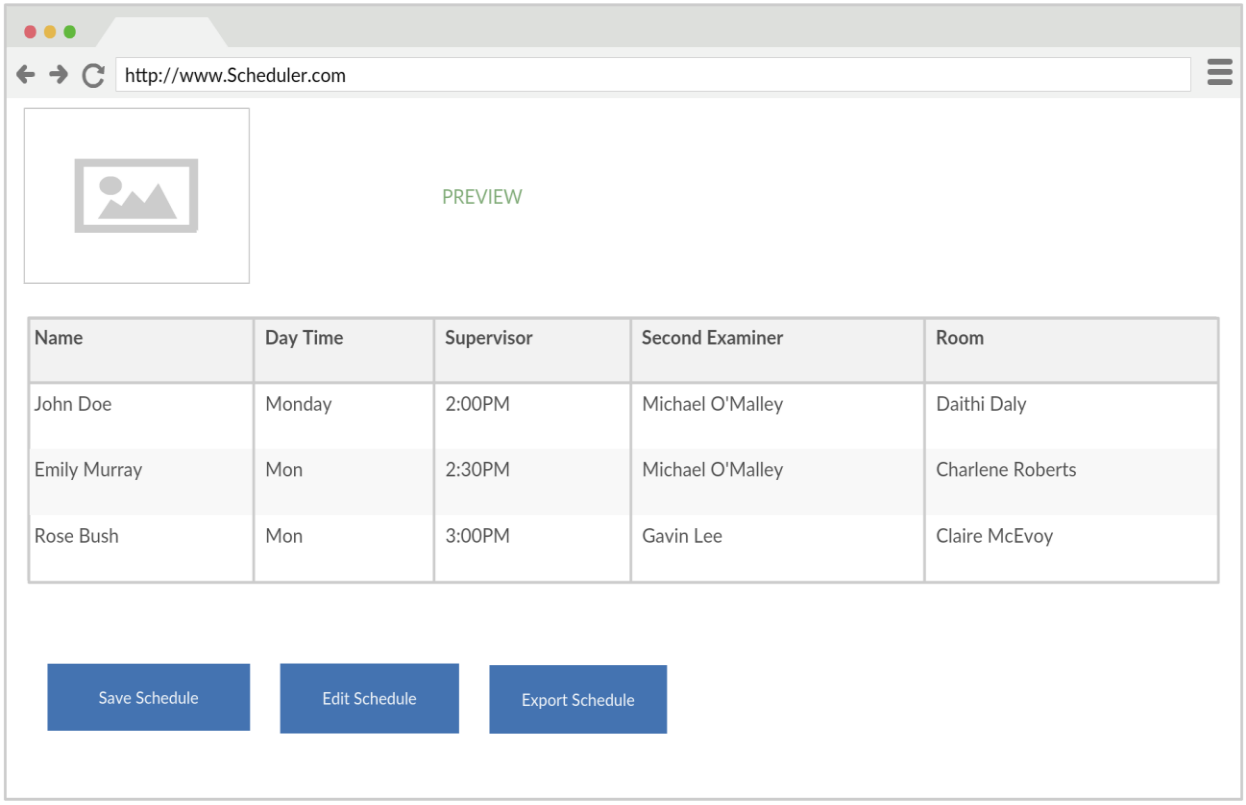
The create schedule page illustrated above will be the page(s) that take the core input data needed by the user for the application to make the schedule. This data will be driven along with the actions database that will contain the constraints or business logic of the of the application.

The create page can be designed as above or it can be designed as individual stages using "Next" functionality which will just stream the process of creating the schedule. This might not be a user-friendly design as it introduces more clicks into the system which is not customer or user friendly.

This screen will need to contain containers and upload links to allow the user to upload files containing time-tables of the lecturers which the database will go and retrieve to establish boundaries.

Once the data has been selected and stored to the database the application will generate a loading screen which has not been illustrated in this documentation as it is not redeemed as a main feature page. Once the schedule has been created, the preview window illustrated below will load.

Preview Schedule GUI



Description of Preview Schedule GUI

The preview window will be used once the creation of the schedule has been completed. The information will be retrieved from the database and displayed using a style like a HTML table except with more designs and formatting.

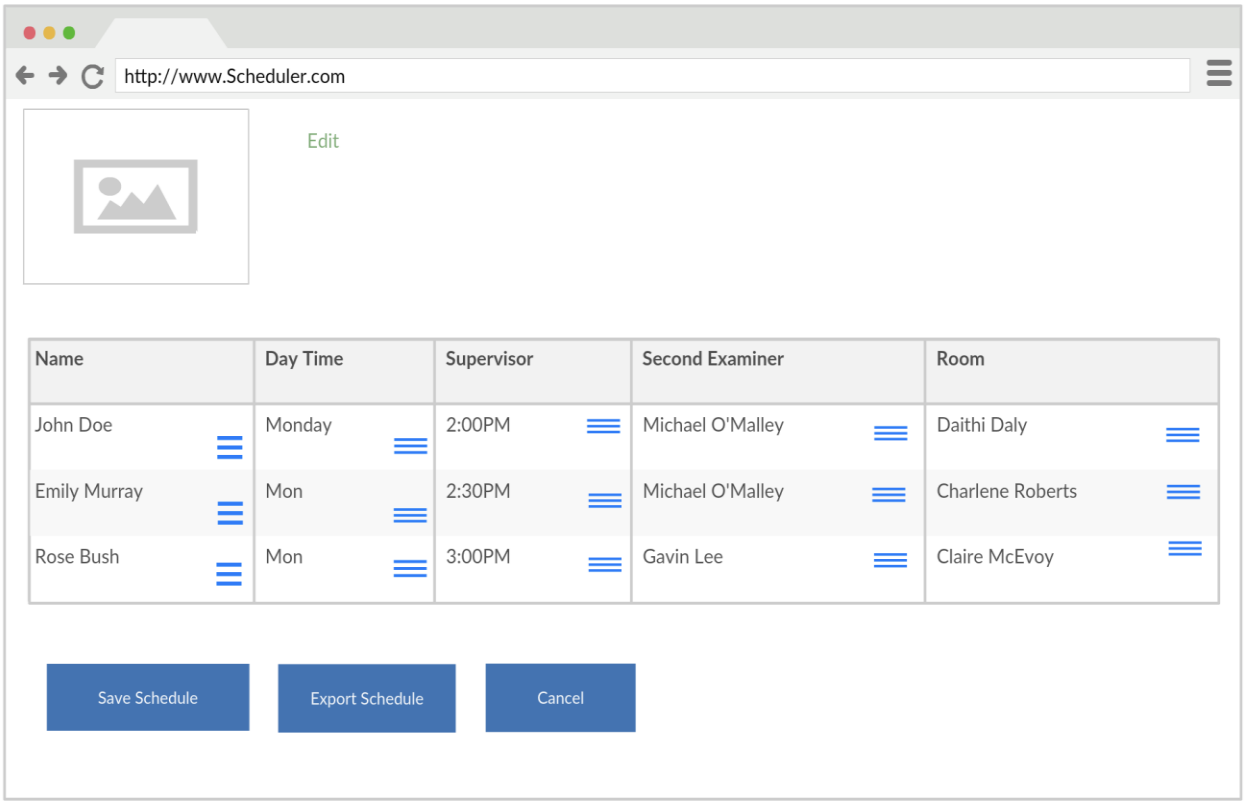
The preview window is used to show the newly generated schedule and allow the user to view the schedule that has been created by the application. From the preview window the user is prompted with several options.

Save Schedule – This is used to commit the schedule to the database and include it in the list of ones showed in the view schedule option from the options page illustrated in the second GUI.

Edit Schedule – Selecting this option will open the schedule in edit mode and allow the user to make some small minor changes that will be reflected into the database upon completion.

Export Schedule – This option will save the schedule to the database and then will export the schedule into a spreadsheet format and save a copy of it onto the local device. This will enable the user to upload onto an enterprise education application or send copies of this schedule via email.

Edit Schedule GUI



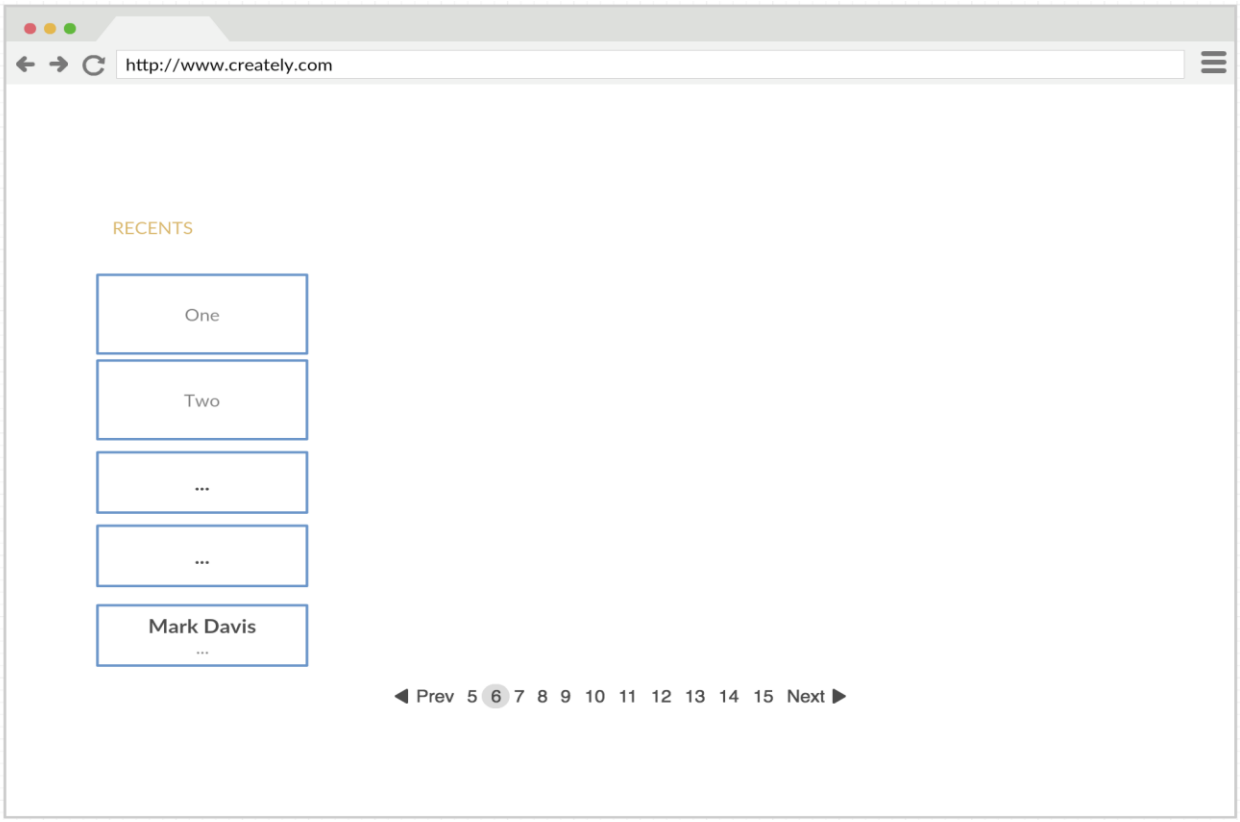
Description of Edit Schedule GUI

The edit schedule will be used in two flows of the application. The edit schedule page will be the directed page of the application once the user has created a new schedule and the application is showing the preview page. If the user needs to make some minor changes to the schedule the application will open the schedule in an edit mode and let the user either dynamically move slots around or remove slots or change the entry of a lecturer or supervisor etc.

The edit page can also be accessed through the options menu which will then prompt the user to select a schedule they would like to modify; this selection is made through a list of schedules saved into the database that will be displayed on the screen like the View Schedule option.

For editing the schedule, a 3rd party application will be implemented into the system that handles HMTL table data and will run library in the background to add the dynamic functionality.

View Schedule GUI



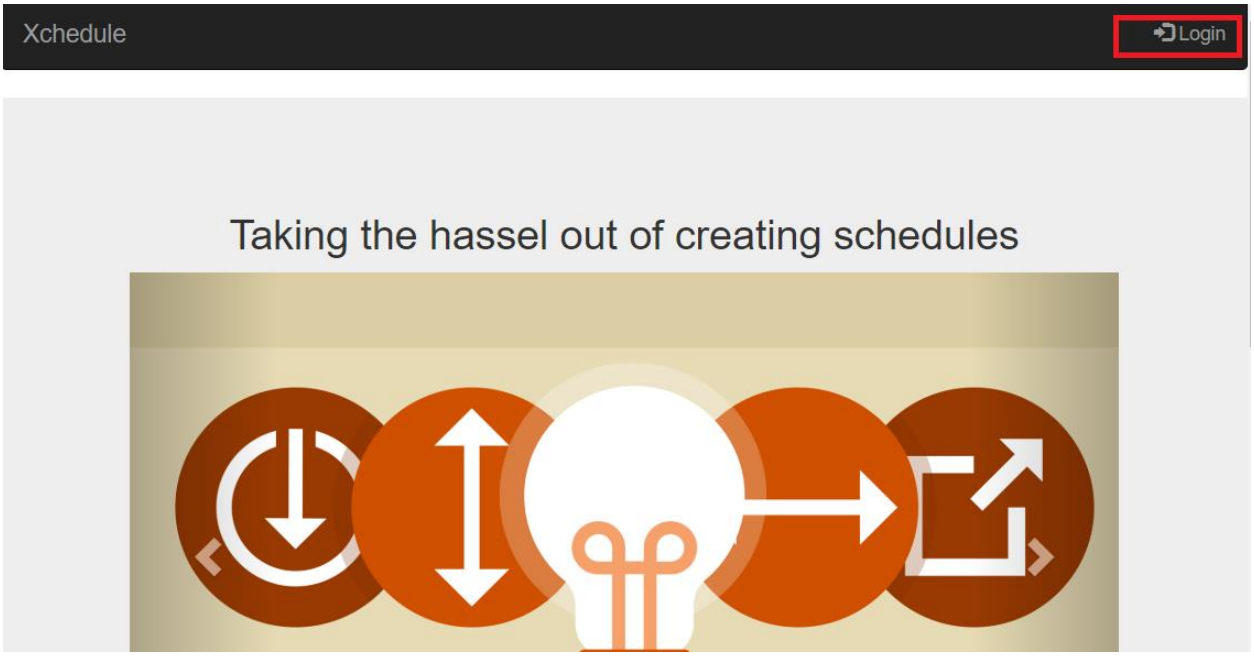
Description of View Schedule GUI

View Schedule will display a list of stored schedules created by the application from previous execution allowing them to view previous schedules that have been generated. The application will open the schedule and it will give further actions that can be taken by the user such as exporting the selected schedule or editing and saving new changes to it.

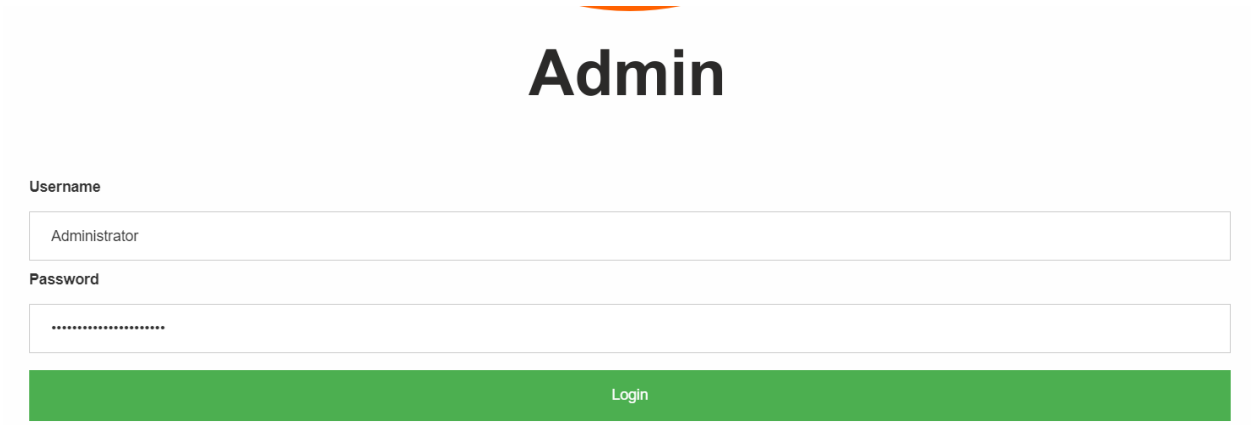
User Manual

Login

To Log into the application, go the home page and click the “Login” icon in the top right as illustrated below.

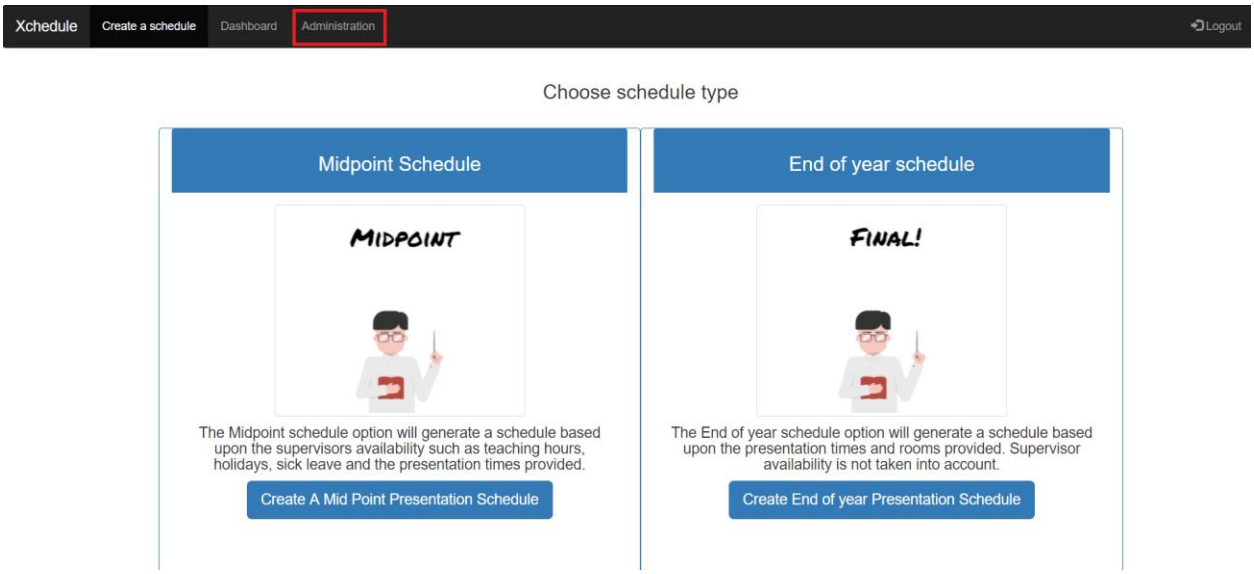


Enter in the credentials Username: Administrator, Password: “Masteradmin” into the username and password fields as illustrated below and click login.



Input the data required

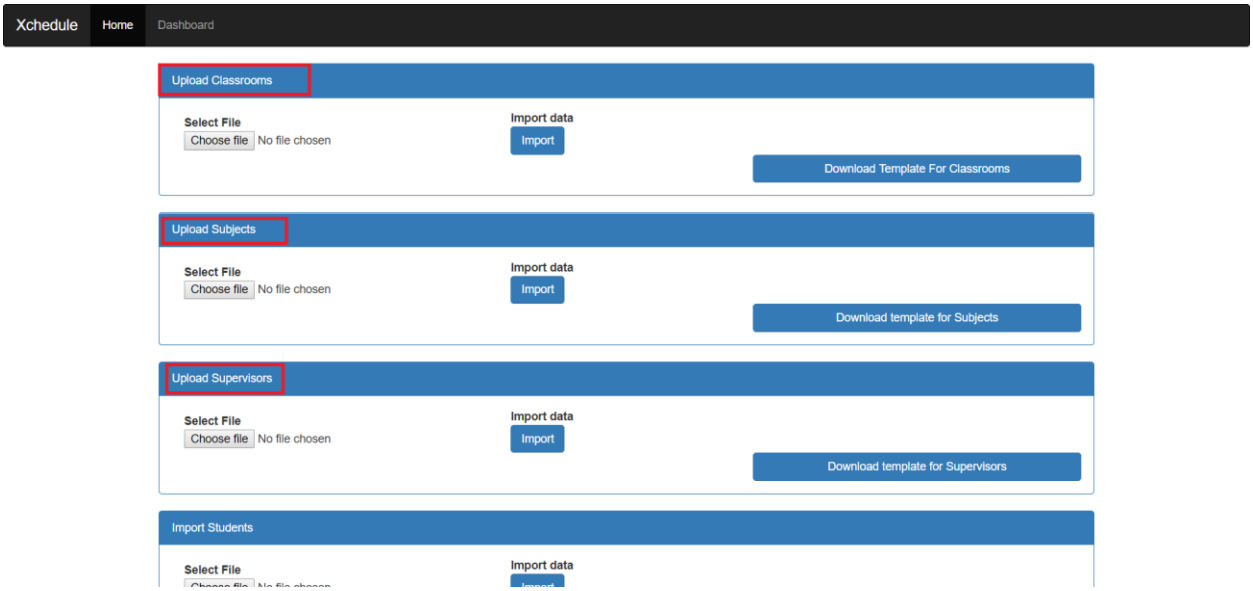
The application works off the data the project administrator needs to organize to schedule presentations. Now go to the Administration tab located at the top of the navigation bar to navigate to the administration page as illustrated below.



The Administration screen will then load. On this section you will see the following lists of required data and must be completed in order as shown on the screen. The required data list is as follows;

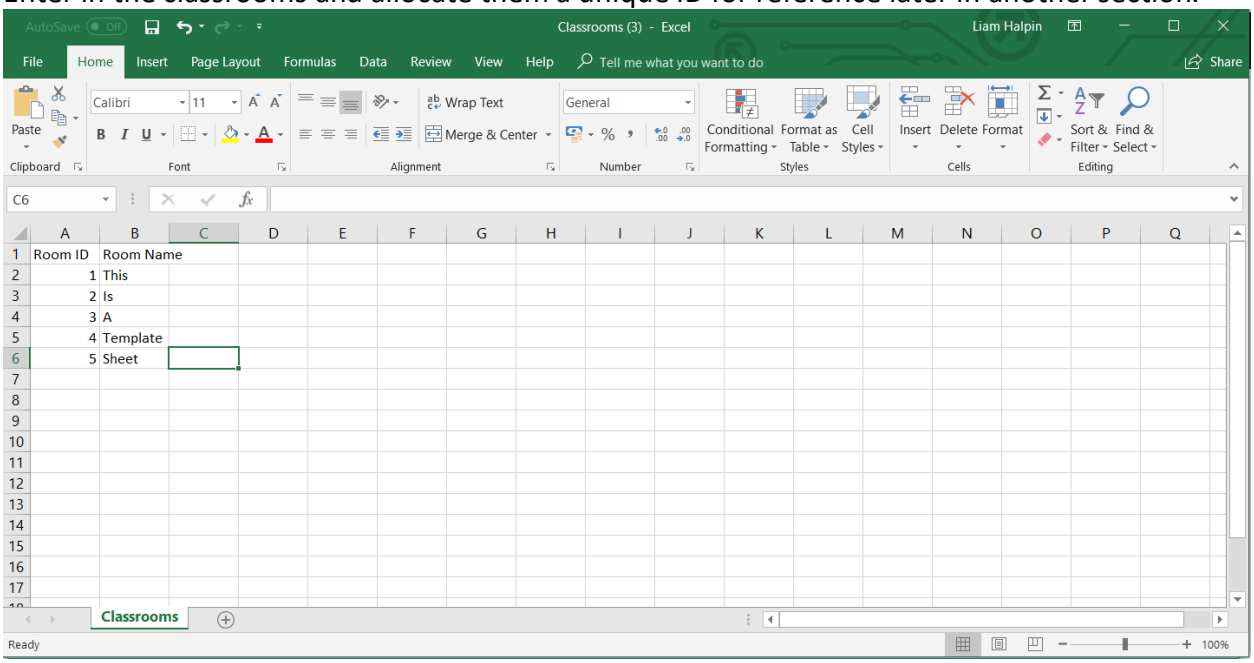
1. Classrooms – The rooms that the presentation will take place.
2. Subject – The list of subjects that the students are enrolled in.
3. Supervisors – The supervisors that will be taking part in the supervision of students and participating in the examination process.
4. Students – The list of students that are enrolled for the final year.
5. Slots – The slots to be allocated to students to present.
6. Supervisor availability – List of supervisor times that the supervisors cannot be scheduled for.

These will be imported from the administration screen as illustrated below.



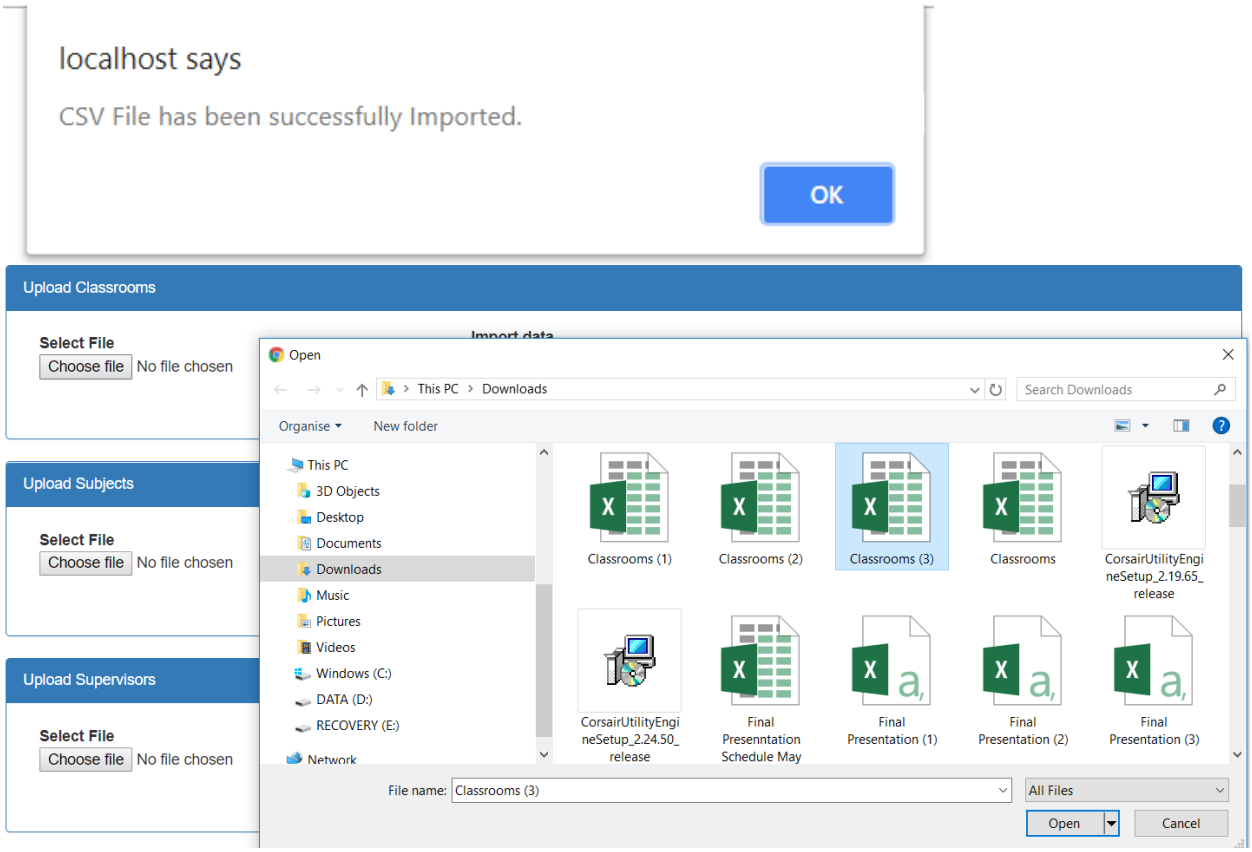
To import the data, first download an excel template and enter in the data needed and shown by the template. For this example, the classroom section will be shown. To download the template, click the "Download Template for Classrooms" button as shown below and open the excel sheet.

Enter in the classrooms and allocate them a unique ID for reference later in another section.

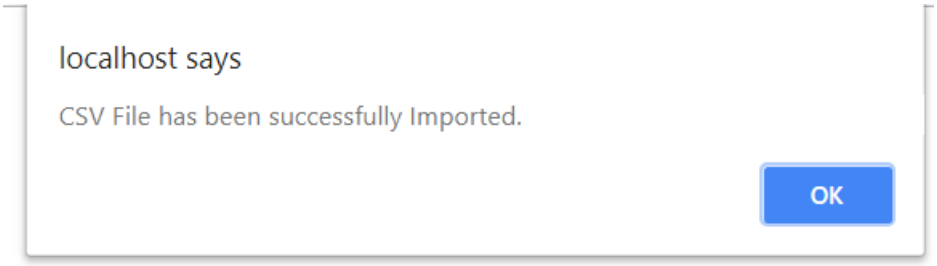


Save the sheet as a CSV (Comma delaminated) and then return to the application.

Select “Choose file” and select the template that was saved previously with the classroom information stored.



Once the sheet has been selected then select the “Import” button and the data will be stored into the application.

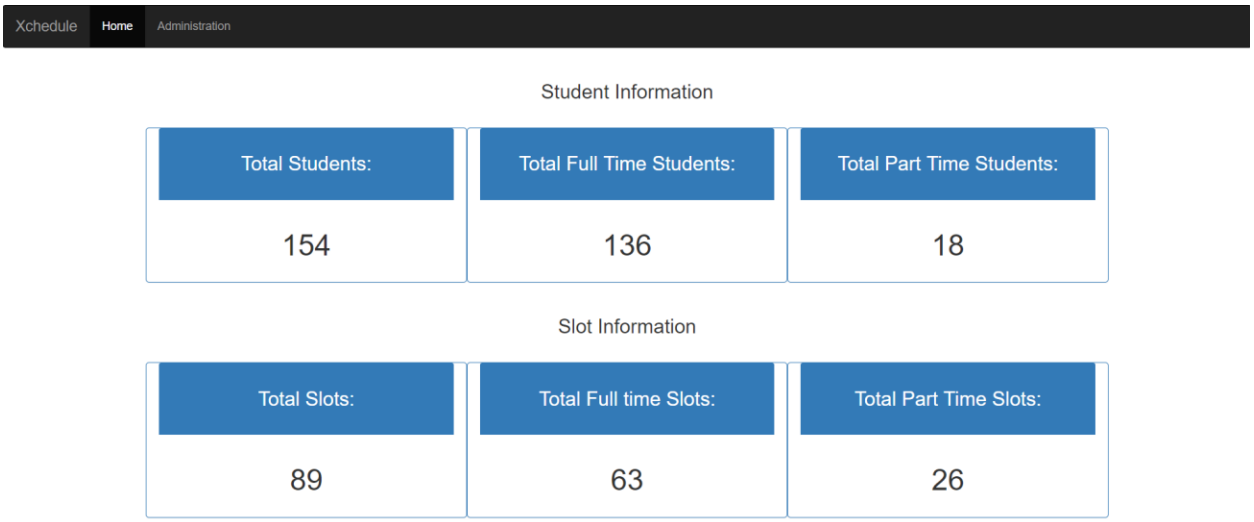


Continue this process until all the information has been store to the application in the administration section.

Dashboard

The dashboard is a utility that shows the administrator information about the data they have submitted. The information is useful to show the admin how many students are in the application the number of slots and supervisors are participating and the number of rooms that are being used throughout the presentations.

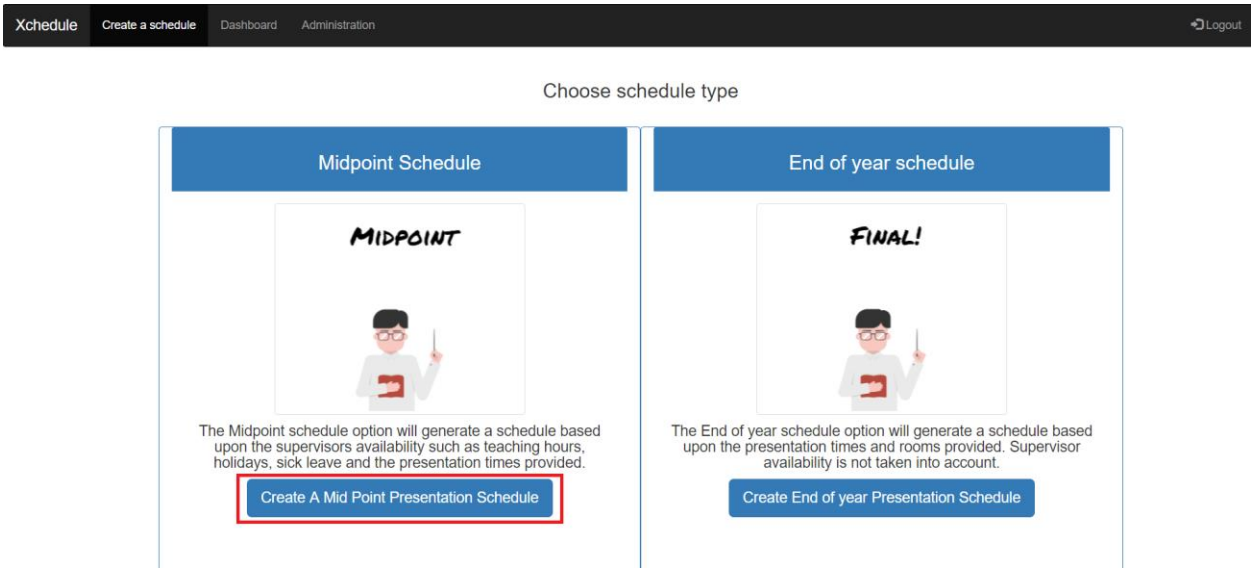
After importing the information, go to the dashboard and verify your data as illustrated below.



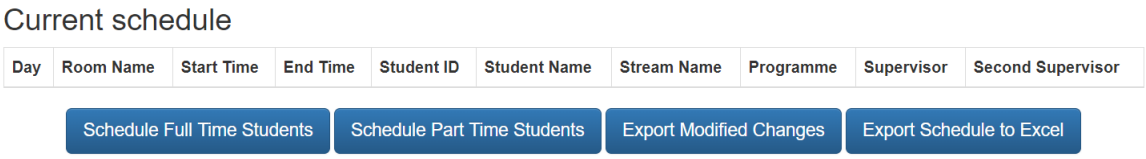
Create a schedule

Now, let's move onto the creation of the schedule. Navigate to the "Create a schedule". For this example, we will create a Mid-point schedule.

Select "Create a mid-point presentation schedule".



We are then taken to the midpoint view of the schedule. This section shows the current schedule for the midpoint if one exists. Otherwise it will just shoe an empty schedule with headers on it like illustrated below.



There are four options to choose from in this section.

- Schedule Full Time Students.
- Schedule Part Time Students.
- Export Modified Changes.
- Export Schedule to Excel.

The first option will schedule all the full-time students enrolled in the current year. The second option will schedule the part time students. Once any student is scheduled they will appear in the timetable as illustrated below.

Current schedule

Day	Room Name	Start Time	End Time	Student ID	Student Name	Stream Name	Programme	Supervisor	Second Supervisor
Wednesday	A	2018-05-23 09:00:00	10:00:00	13115898	LUNDENGO JOELLA MVUMBI	BUSINESS ANALYST	BSHTM	RON ELLIOTT	ARGHIR MOLDOVAN
Wednesday	A	2018-05-23 10:00:00	11:00:00	13120778	GOLUBEVA ANNELI	CYBER SECURITY	BSHC	RON ELLIOTT	ARGHIR MOLDOVAN
Wednesday	A	2018-05-23 11:00:00	12:00:00	13127390	KELLY DAVID	INTERNET OF THINGS	BSHC	RON ELLIOTT	CATHERINE MULWA
Wednesday	A	2018-05-23 12:00:00	13:00:00	13300191	LAWLOR DAVID	INTERNET OF THINGS	BSHC	PAUL STYNES	CATHERINE MULWA
Wednesday	A	2018-05-23 13:00:00	14:00:00	13326556	MURPHY CHRISTOPHER	BUSINESS ANALYST	BSHTM	PAUL STYNES	CRISTIAN RUSU
Wednesday	A	2018-05-23 14:00:00	15:00:00	13335131	O CONNOR LOUISE	DATA ANALYTICS	BSHC	PAUL STYNES	CRISTIAN RUSU
Wednesday	A	2018-05-23 15:00:00	16:00:00	13361536	FERREIRA MARLIN	CLOUD	BSHTM	PAUL STYNES	CRISTIAN RUSU
Wednesday	A	2018-05-23 16:00:00	17:00:00	13399691	PUSPURE DZULIJA	CYBER SECURITY	BSHC	PAUL STYNES	CRISTIAN RUSU
Wednesday	A	2018-05-23 17:00:00	18:00:00	13408172	JORDAN CONOR	BUSINESS ANALYST	BSHTM	PAUL HAYES	CRISTIAN RUSU

If in the event a student cannot be scheduled, the application will store the students that can't be scheduled and show them to the administrator. This will reduce the amount of admin that's needed and then the admin can add new slots that both supervisors can be assigned to. The failed students will be displayed in the following table.

Students who couldn't be assigned to slots

Supervisor	Second Supervisor	Student ID	Student Name
IRINA TAL	JOSEPHINE ANDREWS	14353171	HEFFERNAN MARK
GLEN WARD	JOSEPHINE ANDREWS	14357556	QUINN DARREN
GLEN WARD	IRINA TAL	14357576	MURPHY CURTIS
GLEN WARD	IRINA TAL	14364196	WONG ALEX
GLEN WARD	IRINA TAL	14365561	CORCORAN ALEXANDER
FRANCES SHERIDAN	IRINA TAL	14376696	CUNNINGHAM JOHN

The application allows the administrator to make changes on demand in the schedule from the view shown above. This is a useful function for changing lecturers around or swapping students out for different slots if needed. The "Export Modified Changes" button allows the user to make changes to the proposed schedule and export the changes and will download the students who couldn't be assigned. To make changes to the schedule select the box that you want to modify and enter in the new desired student/time/supervisor.

The following is an example of changing the schedule.

Day	Room Name	Start Time	End Time	Student ID	Student Name	Stream Name	Programme	Supervisor	Second Supervisor
Wednesday	A	2018-05-23 09:00:00	10:00:00	13115898	Hello!	BUSINESS ANALYST	BSHTM	RON ELLIOTT	ARGHIR MOLDOVAN
Wednesday	A	2018-05-23 10:00:00	11:00:00	13120778	GOLUBEVA ANNELI	CYBER SECURITY	BSHC	RON ELLIOTT	ARGHIR MOLDOVAN

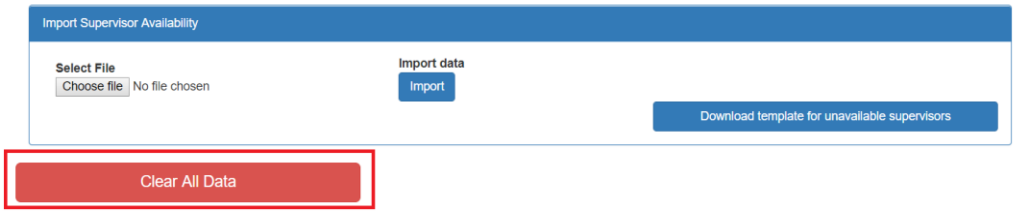
Checking the spreadsheet for changes. – Notice the Student Name “Hello!”.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Day	Room Name	Start Time	End Time	Student ID	Student Name	Stream Na	Programm	Supervisor	Second Supervisor			
2	Wednesda	A	23/05/2018 09:00	10:00:00	13115898	Hello!	BUSINESS	BSHTM	RON ELLIC	ARGHIR MOLDOVAN			
3	Wednesda	A	23/05/2018 10:00	11:00:00	13120778	GOLUBEVA ANNELI	CYBER SEC	BSHC	RON ELLIC	ARGHIR MOLDOVAN			
4	Wednesda	A	23/05/2018 11:00	12:00:00	13127390	KELLY DAVID	INTERNET	BSHC	RON ELLIC	CATHERINE MULWA			

If the schedule that the application has created is suitable for the supervisor, the administrator can download the schedule and it will be downloaded in the form of a CSV excel sheet.

The same principles apply to the End of year presentation the same format and flow.

Now that the schedule has been created, a new has come and gone and it’s time to start the final year over again. It’s time to start all over again. To do this, navigate back to the administration page and at the bottom of the administration page there is a “Clear All Data!” button as shown below.



This button clears all the data and allows the administrator to reuse the application again to create new midpoint and end of year presentations.