**REQUIREMENTS ANALYSIS DOCUMENT**

NPD Coding

**Members**

PATRICK MOYLES – K00242839

NATHAN O’CONNOR – K00243135

DAVID O’MAHONY – K00236744

Contents

[1 Introduction 2](#_Toc56169750)

[1.1 Purpose of System 2](#_Toc56169751)

[1.2 Scope of System 2](#_Toc56169752)

[2 Current Services Description 3](#_Toc56169753)

[2.1 Context Diagram 3](#_Toc56169754)

[2.2 External Entity List with Description 4](#_Toc56169755)

[2.3 Outputs (with Recipients) 5](#_Toc56169756)

[2.4 Inputs (with Sources) 6](#_Toc56169757)

[2.5 List of Documents 7](#_Toc56169758)

[2.6 Initial List of Requirements 8](#_Toc56169759)

[3 Proposed System 9](#_Toc56169760)

[3.1 Overview 9](#_Toc56169761)

[3.2 Functional Requirements 9](#_Toc56169762)

[3.2.1 List of Requirements 10](#_Toc56169763)

[3.3 Non-Functional Requirements 11](#_Toc56169764)

[3.3.1 Usability 11](#_Toc56169765)

[3.3.2 Reliability 11](#_Toc56169766)

[3.3.3 Performance 12](#_Toc56169767)

[3.3.4 Supportability 12](#_Toc56169768)

[3.3.5 Scalability 12](#_Toc56169769)

[3.3.6 Security 13](#_Toc56169770)

[3.4 Use Case Model 13](#_Toc56169771)

[3.4.1 Use Case Definition 13](#_Toc56169772)

# 1 Introduction

PATRICK MOYLES – K00242839

## 1.1 Purpose of System

The purpose of the new system is to replace the Sullimar Academy of Music current system, which is primarily a paper-based system. This system was described by the client as unreliable, slow and too prone to human error. Further discussions with the client revealed that they wanted to move from paper-based to a computer-based system. We then asked the client what information they would like to be stored, should certain privileges be put in place for certain users, exams, exam results etc.

DAVID O’MAHONY – K00236744

## 1.2 Scope of System

Sullimar Academy of Music current system is mainly paper based. The User would come into the academy/ or call and would have to fill out a paper-based form, all the academy’s data on students, teachers, course etc are all in paper form.

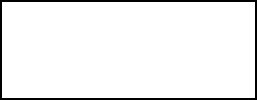
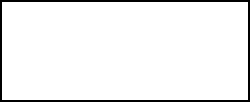
With an increase in business over the last few years the current management cannot provide an effective service to its students and Users. In the current system there has been problems with cancellations and losing cheques, this is due to the lack of sufficient automation of the system and human error.

The academy wishes to transition from a paper-based system to an online system.

# 2 Current Services Description

DAVID O’MAHONY – K00236744

## 2.1 Context Diagram



Show List

Bookings

Box Office

Student Details

Timetable

Payment

Student Results

Exam details

*Examiner*

Course details

Admin details

Payroll

*Teacher*

Enroll in course

Fees

*Student*

*Management System*

NATHAN O’CONNOR – K00243135

## 2.2 External Entity List with Description

|  |  |
| --- | --- |
| **External Entity** | **Description** |
| General Manager | Handles external operations like exams and 1 to 1 tuition |
| External Teacher | Teaches 1 to 1 tuition with students at centres across the country |
| External Examiner | Marks the examinations |
| Exam Customers | Pay to take an exam |
| Parents | Register and pay for their child / children to take classes. |
| Students of 1 to 1 Tuition | Pay to learn instrument(s) in a 1 no 1 setting with an external teacher |
| Students of Leaving Cert Program | Pay to do Leaving Cert level music class content and take the Leaving Cert exam in music |
| Students of Diploma/Degree level Programs | Students apply to a range of diploma and degree level music courses that run in the academy |
| Box Office Customers | Customers that book to view box office performances |
| Box Office Staff | Employees hired to work at the box office |
| Box Office Performers | Performers perform at the box office in front of a live audience |

PATRICK MOYLES – K00242839

## 2.3 Outputs (with Recipients)

|  |  |  |  |
| --- | --- | --- | --- |
| **Recipient** | **Title** | **Description** | **Type** |
| Administrator | Admin wants to add a student | Student is added | Internal |
| Administrator | Admin wants to update a student | Student is updated | Internal |
| Customer | Customer wants to take an exam (User does not have to be a student) | Student is registered for the exam | External |
| Administrator | Admin wants to create a timetable | Timetable is created | Internal |
| Customer | Customer wants to register for a course | Student is registered for that course and provided a student number | External |
| Office Staff | Delete Course | A lack of students has caused staff to delete it | Internal |
| Office Staff | Update Course | The grade of a course needs to be changed from grade 1 to grade 2 | Internal |

DAVID O’MAHONY – K00236744

## 2.4 Inputs (with Sources)

|  |  |  |  |
| --- | --- | --- | --- |
| **Recipient** | **Title** | **Description** | **Type** |
| User | Sign Up | User wants to sign up for a course | External |
| User | Wants to login | User wants to login to their course | External |
| User | Cancel a registration for a course | User wants to cancel a registration for a course and switch to another course | External |
| User | Exam | Users can book an exam online | External |
| User | Box office | User can buy ticket for a performance | External |
| User | Class | Users book a class | External |
| User | Payments | User can pay for fees | External |
| User | Orders | User can view their orders | Internal |
| User | Logout | User logs out after session ends | Internal |
| User | Results | User can view results | Internal |
| User | Timetable | User can view timetable | Internal |
| User | Cancelled Orders | User can cancel an order from orders menu | External |

NATHAN O’CONNOR – K00243135

## 2.5 List of Documents

|  |  |
| --- | --- |
| **Document** | **Description** |
|  |  |
| Exam Application Form | Customers will fill out the form to apply for an exam, detailing their name, subject/instrument, grade, county and nearest town. Customers can also fill out the preferred centre name if known. |
| Candidate Result Sheet | Filled out by an examiner. Details centre the exam took place, exam date, exam number, exam time, examiner, applicant, subject/instrument and grade level. Also displays the marks based on the subject/instrument. |
| Exam Receipt | Receipt for proof of payment to take an exam. |
| Grade Certification | Certificate to show a candidate passed an exam in a particular subject/instrument. Details the grade level the candidate passed in. |
| 1 to 1 Tuition Form | A form to sign a person up for 1 to 1 tuition, the person must fill out the candidate’s name, contact details, address, subject/instrument(s), grade level. |
| Leaving Cert Program Form | Form to sign up for the SAOM Leaving Cert program. |
| Leaving Cert Program Certification | Certificate to show the student passed the leaving cert exam in music at the SAOM. |
| Diploma/Degree Program Form | Form to sign up for a Diploma/Degree program, must fill out candidate details and which program. |
| Diploma/Degree Certification | SAOM Certificate showing the diploma/degree program the student has completed. |
| Invoice | A request for payment sent to the student/parent. |
| Box Office Booking Form | Filled out to make a booking to view a box office performance. |
| Box Office Ticket | A ticket to attend a box office performance, detailing what performance and the date & time of the performance. |

NATHAN O’CONNOR – K00243135

## 2.6 Initial List of Requirements

* Add, update and delete exams.
* Add, update and delete candidates taking exam.
* Add, update and delete examiners.
* Allocate examiner.
* Add, update and delete exam books.
* Add, update and delete candidate results.
* Add, update and delete Internal Teachers for Leaving Cert / Diploma / Degree Programs.
* Add, update and delete External Teachers for 1 to 1 Tuition.
* Add, update and delete general managers.
* Add, update, withdraw and delete Students.
* Add, update and delete Leaving Cert Program(s).
* Add, update and delete Diploma Programs.
* Add, update and delete Degree Programs.
* Add, update and delete 1 to 1 Tuition Classes.
* Add, update and delete Student Timetables.
* Add, update and delete Teacher Timetables.
* Users can register/create accounts.
* Admins/Teachers/Users can login.
* Admins/Teachers/Users can logout.
* Users can browse classes / programs.
* Users can browse Box Office Performances.
* Users can apply to attend exams.
* Registered users can apply for 1 to 1 Tuition Classes.
* Registered users can apply to Leaving Cert / Diploma / Degree Programs.
* Add, update and delete Box Office performances.
* Public can Book to view performances.
* Add, update and delete Instruments.

# 3 Proposed System

PATRICK MOYLES – K00242839

## 3.1 Overview

The new system is designed to replace the old the system which was paper-based, to a new computer-based one. This system needs to able to keep track of student information, like name, address, phone number, email, emergency contact details, student id and their course name. We also need to store information about courses. This could include details like, course name, course id, course coordinator, course length and course level. We also need to set up administrators so they can make changes to the system, like update, delete, add course/student details. We also need to store information on teachers/lecturers like that of a student.

# 3.2 Functional Requirements

NATHAN O’CONNOR – K00243135

## 3.2.1 List of Requirements

1. View course details
2. Modify course details
3. Add new course details
4. Delete course details
5. Update Student result (External)
6. View exam details by location
7. Process external exam (allocate examiner to centre)
8. Book an exam
9. Generate Result Sheet
10. Update a book
11. View book orders
12. Add a book
13. Delete a book
14. Purchase a book
15. Maintenance request form
16. View timetable (teacher)
17. Process timetable (teacher)
18. Book a course
19. Book a lesson
20. Process timetable (student fulltime)
21. View timetable (student)
22. Allocate teacher and room to student (private tuition)
23. Delete Student
24. View Student details
25. Change registered details (user)
26. Register Student
27. Update Student
28. Update staff/teacher
29. Delete staff/teacher
30. View staff details
31. Add staff/teacher
32. Remove items from teacher’s area
33. Update items in teacher’s area
34. Add items to teacher’s area
35. Update event details
36. View event details
37. Add event details
38. Book event
39. Delete event details
40. Generate annual sales report from all facets of business including book sales and box office sales
41. Generate monthly sales report from book sales
42. Display Dashboard
43. Generate monthly sales report for external exams
44. Generate monthly sales report from fulltime course
45. Generate monthly sales report from box office sales
46. Generate monthly sales report from private tuition
47. Login/Logout
48. Generate maintenance request list for mgt

# 3.3 Non-Functional Requirements

DAVID O’MAHONY – K00236744

## 3.3.1 Usability

Usability is very important when it comes to good functioning website. The website needs to be able to achieve the desired goal of the website, it must be quick and efficient enough for the user to achieve their goal on the website. The colours and styles we use on the website is very important to the user’s experience, we must use colours that best represent the site and provide a good experience. The website should be a good experience for the user and be able to navigate through therefore we should be conscious of the colours we use and the size of the objects in the site.

DAVID O’MAHONY – K00236744

## 3.3.2 Reliability

Reliability is very important when it comes to good functioning website. Up time requirements refer to the total amount of time that the system is available for end-use applications. We should try to meet a goal of 24/7/365 uptime as this will give the best experience to students and teachers.

NATHAN O’CONNOR – K00243135

## 3.3.3 Performance

Performance is very important when creating a web application because it directly affects user experience when users are using the web application. If the site performs slowly it is more likely users won’t return and they won’t recommend the service to other people. It is important to create a web application that operates as fast as possible for users to have a good experience, but it is also important for when staff and admins want to run functions on the web application.

NATHAN O’CONNOR – K00243135

## 3.3.4 Supportability

Supportability is necessary to ensure the web application is up and running as much as possible. Support will be provided in-house by the admins of the web application. Supportability is needed for updates to the web application and changes like improving technologies being made available and scaling up of the web application. The main reason for supportability is for maintenance and bug fixing.

DAVID O’MAHONY – K00236744

## 3.3.5 Scalability

As the school grows and has become more reliant on online systems it is very important to take into account can the system handle the growth. The best way to insure that the system can handle the growth is to do regular checks and slowly update the system so that is does not crash and is able to handle the exponential growth of the school and therefore the system.

PATRICK MOYLES – K00242839

## 3.3.6 Security

Security is very important in making sure technology does not fall into the wrong hands. This can be done by making sure there is a proper firewall installed on the system, ensuring all computers have proper passwords for all sections. Make sure all computers are never left unattended while logged in.

# 3.4 Use Case Model

## 3.4.1 Use Case Definition

|  |
| --- |
| **Use Case By:** PATRICK MOYLES – K00242839 |
| **Use Case Name:** Maintenance Request Form |
| **Use Case Description:** This option will allow a user to fill out Maintenance Request Form. |
| **Participating Actor(s):** Online User |
| **Entry Conditions:**   1. Needs to be a registered user 2. User invokes the Make Maintenance Request Form feature in the settings menu |
| **Flow of Events:**   1. The system responds by displaying a form, requesting the user to fill out the form, headings include problem description, room number, computer number. 2. The user enters in these details and clicks send request. 3. The system responds by confirming the room and computer number exist in the system. If this entry is correct, then a confirmation message is displayed, letting the user know the request was sent. |
| **Exit Condition (s):** Maintenance Request was sent |
| **Exceptions: None** |

|  |
| --- |
| **Use Case By:** PATRICK MOYLES – K00242839 |
| **Use Case Name:** View Book Orders |
| **Use Case Description:** This will allow an Admin to view the book orders |
| **Participating Actor(s):** Administrator |
| **Entry Conditions:**   1. Needs to be an Administrator 2. The admin invokes the view book orders feature in the admin menu |
| **Flow of Events:**   1. The system responds by displaying all the current book orders 2. The admin browses the list of book orders |
| **Exit Condition (s):** The admin exits the view the book orders list |
| **Exceptions: None** |

|  |
| --- |
| **Use Case By:** PATRICK MOYLES – K00242839 |
| **Use Case Name:** Generate Monthly Sales Report from Fulltime course |
| **Use Case Description:** This option will allow an admin to Generate Monthly Sales Report for Fulltime courses |
| **Participating Actor(s):** Administrator |
| **Entry Conditions:**   1. Needs to be an Administrator 2. The admin invokes the view Generate Monthly Sales Report from Fulltime course in the admin menu |
| **Flow of Events:**   1. The system responds by displaying a form with text boxes, including course name, course id, total cost, profit etc. This information will be automatically generated by hidden formulas. 2. The admin will be able to view this information, export this document as a pdf or print it. |
| **Exit Condition (s):** The admin exits the Monthly Sales Report from Fulltime course form |
| **Exceptions: None** |

|  |
| --- |
| **Use Case By:** PATRICK MOYLES – K00242839 |
| **Use Case Name:** View Student Details |
| **Use Case Description:** This option will allow the Admin to view student details |
| **Participating Actor(s):** Administrator |
| **Entry Conditions:**   1. Needs students to be registered before their visible 2. Needs to be a Registered Admin |
| **Flow of Events:**   1. The system responds by displaying a screen with a list of registered users 2. The Admin can respond by clicking on a user profile and read more details about that specific user. 3. The Admin can exit the profile and view another |
| **Exit Condition (s):** a student profile has been viewed |
| **Exceptions: None** |

|  |
| --- |
| **Use Case By:** PATRICK MOYLES – K00242839 |
| **Use Case Name:** View Timetable (teacher) |
| **Use Case Description:** This option will allow a teacher to view their timetable |
| **Participating Actor(s):** Teacher |
| **Entry Conditions:**   1. Teacher needs to be registered 2. Needs to have a timetable assigned to them 3. Needs to have invoked the view teacher timetable function |
| **Flow of Events:**   1. The system responds by displaying a list of timetables 2. The teacher will select his/her timetable 3. The system responds by displaying the selected timetable 4. The teacher responds by viewing their timetable exiting when finished |
| **Exit Condition (s):** Timetable has been viewed |
| **Exceptions: None** |

|  |
| --- |
| **Use Case By:** PATRICK MOYLES – K00242839 |
| **Use Case Name:** Update a Book |
| **Use Case Description:** this allows an administrator to update a books detail |
| **Participating Actor(s):** Administrator |
| **Entry Conditions:**   1. Must be a registered user 2. The invoke update book details function has been invoked |
| **Flow of Events:**   1. The system responds by displaying a dialog box with the selected book fields being filled with their original data. This data includes things like book name, book price, book description etc. 2. The Admin can select any one of these fields and change the details to whatever they like. The user can then hit the update button to update the chosen book 3. The system responds by validating the data in each one of the fields. If this data is correct the system, then updates the chosen book |
| **Exit Condition (s):** Book has been successfully updated |
| **Exceptions: None** |

|  |
| --- |
| **Use Case By:** PATRICK MOYLES – K00242839 |
| **Use Case Name:** View Course Details |
| **Use Case Description:** This will allow a user to view a courses detail |
| **Participating Actor(s):** Base User (Does not need to be registered) |
| **Entry Conditions:**   1. The user invokes the view a course details feature from the menu |
| **Flow of Events:**   1. The system responds by displaying a list of courses with their name and course id. 2. The user responds by clicking into the selected course. The course name, course id, and course description will also be displayed. |
| **Exit Condition (s):** The user has found the course and bin able to view it |
| **Exceptions: None** |

|  |
| --- |
| **Use Case By:** PATRICK MOYLES – K00242839 |
| **Use Case Name:** View Exam Details by Location |
| **Use Case Description:** This will allow a Registered User to view Exam Details by Location |
| **Participating Actor(s):** Registered User |
| **Entry Conditions:**   * 1. Must be registered user   2. Must have invoked the view Exam Details by Location feature from the menu |
| **Flow of Events:**   1. The system responds by displaying a list of exam results by location. 2. The user responds by clicking into the selected exam location which will display details like the exam location, exam description, building number, room number and if a reader or helper of any kind is assigned to that room during an exam. 3. The user can then view these details and select others if they choose |
| **Exit Condition (s):**   1. An exam location detail has been selected and viewed |
| **Exceptions: None** |

|  |
| --- |
| **Use Case By:** PATRICK MOYLES – K00242839 |
| **Use Case Name:** View Staff Details |
| **Use Case Description:** This will allow an administrator to view staff details |
| **Participating Actor(s):** Administrator |
| **Entry Conditions:**   * 1. Must be a registered user   2. Must have invoked the view staff details feature from the menu |
| **Flow of Events:**   1. The system responds by displaying a list of currently active teachers that work at the school, a picture of the teacher is displayed as well as basic information like name, email, phone etc. 2. The administrator can then click into their profile to find out more information about this teacher like what courses they work, what subjects they teach and any other noticeable skills. |
| **Exit Condition (s):** A teachers profile has been viewed |
| **Exceptions: None** |

|  |
| --- |
| **Use Case By:** PATRICK MOYLES – K00242839 |
| **Use Case Name:** Update Student Results |
| **Use Case Description:** this option will allow a student’s exam results to be updated |
| **Participating Actor(s)**   1. Administrator |
| **Entry Conditions:**   1. A student’s exam results need to be updated 2. The administrator invokes the update student results feature from the menu |
| **Flow of Events:**   1. The system responds by displaying a list of all current active students, at the top of the page is a search bar where a student can be searched by the student id. 2. The administrator can respond by using this search function or can scroll down till they find the student exam results they wish to update. Once the profile is found they can click into it to begin updating process. 3. The system responds by loading in the student’s profile with all the default information being displayed in various text boxes 4. The administrator can respond by finding the field they wish to update and clicking the update button |
| **Exit Condition (s):** A students exam results have been updated |
| **Exceptions: None** |

|  |
| --- |
| **Use Case By:** PATRICK MOYLES – K00242839 |
| **Use Case Name:** Add a Book |
| **Use Case Description:** a administrator wants to add a new book to the online library |
| **Participating Actor(s):** Administrator |
| **Entry Conditions:**   1. A new book needs to be added to the online store 2. The add new book feature has been invoked from the menu |
| **Flow of Events:**   1. The system responds by displaying a text fields that are required to be filled out so details of this new book can be added to the system, book name, decription book id etc. 2. The administrator responds by filling out all these text fields and clicking add book once this is done. 3. The system responds by validating the information and adding the book to the store if the validation requirements have been met |
| **Exit Condition (s):** A book has been added |
| **Exceptions: None** |

|  |
| --- |
| **Use Case By:** PATRICK MOYLES – K00242839 |
| **Use Case Name:** Display Dashboard |
| **Use Case Description:** The administrator would like to view the dashboard so they can do a function |
| **Participating Actor(s):** Administrator |
| **Entry Conditions:**   1. An administrator would like to view the dashboard 2. The view dashboard feature after login has been invoked |
| **Flow of Events:**   1. The system responds by displaying the dashboard 2. The administrator can pick from a few options like, adding a student/teacher or deleting a book or updating it. |
| **Exit Condition (s):** the dashboard has been viewed |
| **Exceptions: None** |

|  |
| --- |
| **Use Case By:** PATRICK MOYLES – K00242839 |
| **Use Case Name:** Generate Monthly Sales Report for External Exams |
| **Use Case Description:** this will allow an administrator to generate a Monthly Sales Report for External Exams |
| **Participating Actor(s):** Administrator |
| **Entry Conditions:**   1. Needs to be an administrator 2. A new Monthly Sales Report for External Exams needs to be generated 3. The Monthly Sales Report for External Exams feature has been invoked |
| **Flow of Events:**   1. The system responds by invoking the display dashboard feature 2. The admin clicks the Generate Monthly Sales Report for External Exams button on the dashboard 3. The system responds by displaying a list of all the monthly reports 4. The admin clicks into the monthly report 5. The system will respond a list of what months to generate the report from 6. The admin selects the desired month and year 7. The system responds by displaying the Monthly Sales Report for External Exams report for that specific month and year. |
| **Exit Condition (s):** a Monthly Sales Report for External Exams has been generated |
| **Exceptions: None** |

|  |
| --- |
| **Use Case By:** NATHAN O’CONNOR – K00243135 |
| **Use Case Name:** Add Staff/Teacher |
| **Use Case Description:**  Use case to add new staff or new teachers to the system. |
| **Participating Actor(s):** Admin |
| **Entry Conditions:**  The admin is logged in.  The add staff/teacher function has been invoked. |
| **Flow of Events:**   1. The system responds by displaying the add staff/teacher screen and requests the admin to enter the staff/teachers details and press continue. 2. The admin enters the details and clicks on continue. 3. The system responds by displaying the entered details of the new staff/teacher and requests the admin to click confirm or cancel. 4. If the admin clicks confirm.    1. The system responds by displaying a message saying the staff/teacher has been added while showing the staff/teacher’s details. 5. If the admin clicks cancel.    1. The system responds by displaying a message saying the staff/teacher has not been added. |
| **Exit Condition (s):**  The staff or teacher has been added to the system.  The admin cancels the process of adding the staff or teacher. |
| **Exceptions:**  Invalid details added. |

|  |
| --- |
| **Use Case By:** NATHAN O’CONNOR – K00243135 |
| **Use Case Name:** Book a course |
| **Use Case Description:**  Use case to allow a user to book a full-time course in the Sullimar Academy of Music. |
| **Participating Actor(s):** User |
| **Entry Conditions:**  The user has registered.  The user is logged in.  The book a course function has been invoked. |
| **Flow of Events:**   1. The system responds by displaying the book a course screen and requests the applicant to choose what course they would like to attend then press continue. 2. The user chooses a course and clicks on continue. 3. The system responds by displaying a screen requesting the applicant to enter their contact details then press continue. 4. The user enters all of the required details and clicks on continue. 5. The system responds by displaying all the details on a screen back to the user and requests the user to click on finalise booking or cancel. 6. If the user clicks finalise booking.    1. The system responds by displaying a screen with fees and requests the user to enter payment details then click continue.    2. The user enters their payment details and clicks on continue.    3. The system responds by displaying the payment details back to the user and requests the user to click pay.    4. The user clicks on pay. 7. If the user clicks on cancel.    1. The system responds by displaying a message saying that the booking has been cancelled. |
| **Exit Condition (s):**  The user has booked a course.  The user has cancelled their booking. |
| **Exceptions:**  Invalid details added.  Invalid Payment details.  Payment declined. |

|  |
| --- |
| **Use Case By:** NATHAN O’CONNOR – K00243135 |
| **Use Case Name:** Book a lesson |
| **Use Case Description:**  Use case to allow a user to book a lesson or to allow a parent to book a lesson for their child. |
| **Participating Actor(s):** User |
| **Entry Conditions:**  The user has registered.  The user is logged in.  The book a lesson function has been invoked. |
| **Flow of Events:**   1. The system responds by displaying the book a lesson screen and requests the user to choose what lesson they would like to book then press continue. 2. The user chooses a lesson and clicks on continue. 3. The system responds by displaying a screen requesting the user to enter the applicants details then press continue. 4. The user enters all of the required details and clicks on continue. 5. The system responds by displaying a screen requesting the user to enter contact details then press continue. 6. The user enters the contact details and clicks on continue. 7. The system responds by displaying all the details on a screen back to the user and requests the user to click on finalise booking or cancel. 8. If the user clicks finalise booking.    1. The system responds by displaying a screen with lesson costs and requests the user to enter payment details then click continue.    2. The user enters their payment details and clicks on continue.    3. The system responds by displaying the payment details back to the user and requests the user to click pay.    4. The user clicks on pay. 9. If the user clicks on cancel.    1. The system responds by displaying a message saying that the booking has been cancelled. |
| **Exit Condition (s):**  The user has booked a lesson.  The user has cancelled their booking. |
| **Exceptions:**  Invalid details added.  Invalid Payment details.  Payment declined. |

|  |
| --- |
| **Use Case By:** NATHAN O’CONNOR – K00243135 |
| **Use Case Name:** Book an exam |
| **Use Case Description:**  Use case for when a person wants to book an exam. |
| **Participating Actor(s):** User |
| **Entry Conditions:**  The book an exam function has been invoked. |
| **Flow of Events:**   1. The system responds by displaying the book an exam screen and requests the user to enter all of their details then click continue. 2. The user enters all their details and clicks on continue. 3. The system responds by displaying a screen requesting the user to enter the details of the exam(s) they’d like to take, and the details of the preferred examination centre then click continue. 4. The user enters all of the exam details and clicks on continue. 5. The system responds by displaying all the details on a screen back to the user and requests the user to click on finalise booking or cancel. 6. If the user clicks finalise booking.    1. The system responds by displaying a screen with the price and requests the user to enter payment details then click continue.    2. The user enters their payment details and clicks on continue.    3. The system responds by displaying the payment details back to the user and requests the user to click pay.    4. The user clicks on pay. 7. If the user clicks on cancel.    1. The system responds by displaying a message saying that the booking has been cancelled. |
| **Exit Condition (s):**  The user has booked an exam.  The user has cancelled their booking. |
| **Exceptions:**  Invalid details added.  Invalid Payment details.  Payment declined.  Exam attendance full. |

|  |
| --- |
| **Use Case By:** NATHAN O’CONNOR – K00243135 |
| **Use Case Name:** Change Registered Details (User) |
| **Use Case Description:**  Use case to allow a user to update their registered details. |
| **Participating Actor(s):** User |
| **Entry Conditions:**  The user has registered.  The user is logged in.  The change registered details function has been invoked. |
| **Flow of Events:**   1. The system responds by displaying the user’s account details and requests the user to make changes then click confirm or cancel. 2. If the user makes changes to their details, then clicks confirm.    1. The system responds by updating their account details then displays a screen back to the user with the updated account details. 3. If the user clicks on cancel.    1. The system responds by displaying a message saying the user details have not been changed. |
| **Exit Condition (s):**  The user has changed their registered details.  The user has cancelled changing their details. |
| **Exceptions:**  Invalid details added. |

|  |
| --- |
| **Use Case By:** NATHAN O’CONNOR – K00243135 |
| **Use Case Name:** Delete a Book |
| **Use Case Description:**  Use case for the admin to select a book and delete the book from the system. |
| **Participating Actor(s):** Admin |
| **Entry Conditions:**  The admin is logged in.  The delete a book function has been invoked. |
| **Flow of Events:**   1. The system responds by listing all the books on the system and requests the admin to select a book to delete. 2. The admin selects a book to delete. 3. The system responds by displaying the books details and prompts the admin to delete or cancel. 4. If the admin clicks delete.    1. The book is removed from the system and a message is displayed saying the book has been deleted. 5. If the admin clicks cancel.    1. The system responds by returning to the admin page. |
| **Exit Condition (s):**  The admin has deleted a book.  The admin has cancelled deleting a book. |
| **Exceptions:**  The book could not be removed from the system. |

|  |
| --- |
| **Use Case By:** NATHAN O’CONNOR – K00243135 |
| **Use Case Name:** Generate monthly sales report from box office sales |
| **Use Case Description:**  Use case to generate a monthly sales report from the box office sales. Displays the details of the box office sales for that month on the sales report. |
| **Participating Actor(s):** Admin |
| **Entry Conditions:**  The admin has logged in.  The generate monthly sales report from box office sales function has been invoked. |
| **Flow of Events:**   1. The system responds by generating a monthly sales report from box office sales and displays the report to the admin (Detailing daily sales, daily booking amounts, percentage of sales for that month, total sales and time period) then requests the admin to click continue. 2. The admin clicks continue. 3. The system responds by saving the report and displaying a message stating the report has been saved. |
| **Exit Condition (s):**  The admin has generated a monthly sales report from box office sales. |
| **Exceptions:**  The system fails to generate a report. |

|  |
| --- |
| **Use Case By:** NATHAN O’CONNOR – K00243135 |
| **Use Case Name:** Generate monthly sales report from private tuition |
| **Use Case Description:**  Use case to generate a monthly sales report from private tuition. Displays the details of the private tuition sales for that month on the sales report. |
| **Participating Actor(s):** Admin |
| **Entry Conditions:**  The admin has logged in.  The generate monthly sales report from private tuition function has been invoked. |
| **Flow of Events:**   1. The system responds by generating a monthly sales report from private tuition and displays the report to the admin (Detailing daily sales, amount of private tuition lessons per day, percentage of sales for that month, total sales and time period) then requests the admin to click continue. 2. The admin clicks continue. 3. The system responds by saving the report and displaying a message stating the report has been saved. |
| **Exit Condition (s):**  The admin has generated a monthly sales report from private tuition. |
| **Exceptions:**  The system fails to generate a report. |

|  |
| --- |
| **Use Case By:** NATHAN O’CONNOR – K00243135 |
| **Use Case Name:** Process External Exam |
| **Use Case Description:**  Use case to process an external exam and allocate an examiner to a centre. |
| **Participating Actor(s):** Admin |
| **Entry Conditions:**  The admin has logged in.  The process external exam function has been invoked. |
| **Flow of Events:**   1. The system responds by displaying a screen requesting the admin to enter the exam details (subject/instrument, grade, start time, end time, date, price) then press continue. 2. The admin enters the exam details and presses continue. 3. The system responds by displaying a list of available exam centres and prompts the admin to select a centre. 4. The admin selects a centre. 5. The system responds by displaying a list of available examiners to allocate to the exam and prompts the admin to select an examiner. 6. The admin selects an examiner to allocate to the exam. 7. The system responds by displaying a screen detailing the exam details, exam centre and examiner details then prompts the admin to click process or cancel. 8. If the admin selects process.    1. The system responds by processing the exam. 9. If the admin selects cancel.    1. The system responds by returning the admin to the admin page. |
| **Exit Condition (s):**  The admin has processed an external exam.  The admin has cancelled the processing of an external exam. |
| **Exceptions:**  Invalid exam details.  Failed to add exam centre.  Failed to allocate examiner. |

|  |
| --- |
| **Use Case By:** NATHAN O’CONNOR – K00243135 |
| **Use Case Name:** Process Timetable (Teacher) |
| **Use Case Description:**  Use case to process the timetable for a teacher. |
| **Participating Actor(s):** Admin |
| **Entry Conditions:**  The admin has logged in.  The process timetable function has been invoked. |
| **Flow of Events:**   1. The system responds by displaying a screen listing all the teachers and requests the admin to select a teacher. 2. The admin selects a teacher. 3. The system responds by displaying the teachers details and    1. FOR each subject/instrument the teacher teaches the system requests the admin to select a room, time and day then click continue.    2. The admin enters the details and clicks continue.    3. The system responds by checking availability and requests the admin to re-enter details if the availability check fails. 4. The admin entered the last subject/instrument timetable details and clicked continue. 5. The system responds by displaying the teachers details and the timetable for the teacher then prompts the admin to click process or cancel. 6. If the admin clicks process.    1. The system responds by processing the timetable. 7. If the admin clicks cancel.    1. The system responds by returning the admin to the admin page. |
| **Exit Condition (s):**  The admin has processed the timetable.  The admin cancelled processing the timetable. |
| **Exceptions:**  No rooms available. |

|  |
| --- |
| **Use Case By:** DAVID O’MAHONY – K00236744 |
| **Use Case Name:** Add items to teachers’ area |
| **Use Case Description:** This will allow admin to add items to teacher’s area |
| **Participating Actor(s):** Admin |
| **Entry Conditions:**   1. An admin has logged in 2. The add items to teacher’s area function is invoked |
| Flow of Events:   1. They system responds by showing the admin home page and request the admin to select an option 2. The admin selects add items to teacher’s area button. 3. The system responds by requesting the admin to select the teacher’s area they wish to add items to. 4. The admin enters the teacher’s details. 5. The system responds requestion the admin to enter what items they wish to add. 6. The admin enters the items they wish to add. 7. The system responds by saving the details |
| **Exit Condition (s):**   1. The admin has saved the changes and logged out. |
| **Exceptions:** |

|  |
| --- |
| **Use Case By:** DAVID O’MAHONY – K00236744 |
| **Use Case Name:** Add new course details |
| **Use Case Description:** This will allow admin to add new course details |
| **Participating Actor(s):** Admin |
| **Entry Conditions:**  1. An admin has logged in  2. The add new course details function is invoked |
| **Flow of Events:**   1. They system responds by showing the admin home page and request the admin to select an option 2. The admin selects the show course details button. 3. The system responds by showing the list of courses available and the options available to the admin. 4. The admin selects the add new course detail button 5. System responds by displaying course and request the admin to select a course to add new details. 6. Admin selects a course and adds new course details and clicks the save button. 7. The system responds by saving changes to database. |
| **Exit Condition (s):**  1. The admin has saved the changes and logged out. |
| **Exceptions:** |

|  |
| --- |
| **Use Case By:** DAVID O’MAHONY – K00236744 |
| **Use Case Name:** Delete course details |
| **Use Case Description:** This will allow admin to delete course details |
| **Participating Actor(s):** Admin |
| **Entry Conditions:**   1. An admin has logged in 2. The delete course details function is invoked |
| **Flow of Events:**   1. They system responds by showing the admin home page and request the admin to select an option 2. The admin selects the show course details button. 3. The system responds by showing the list of courses available and the options available to the admin. 4. The admin selects the delete course detail button 5. System responds by displaying course and request the admin to select a course to delete details. 6. Admin selects a course and delete course details and clicks the save button. 7. The system responds by saving changes to database. |
| **Exit Condition (s):**   1. The admin has saved the changes and logged out. |
| **Exceptions:** |

|  |
| --- |
| **Use Case By:** DAVID O’MAHONY – K00236744 |
| **Use Case Name:** Generation a maintenance request list for management |
| **Use Case Description:** This will gen generation a maintenance request list for management |
| **Participating Actor(s):** Admin |
| **Entry Conditions:**   1. An admin has logged in 2. The generation a maintenance request list for management is invoked |
| **Flow of Events:**   1. The system responds by generation a maintenance request list for management and displays it to the admin (Details include list of maintenance request form) The system responds by displaying save button 2. The admin clicks the save button 3. The system responds by saving the report. |
| **Exit Condition (s):**   1. The admin has generation a maintenance request list for management |
| **Exceptions:** |

|  |
| --- |
| **Use Case By:** DAVID O’MAHONY – K00236744 |
| **Use Case Name:** Generate Result Sheet |
| **Use Case Description:** This will allow admin to generate result sheet |
| **Participating Actor(s):** Admin |
| **Entry Conditions:**   1. An admin has logged in 2. The generate result sheet function is invoked |
| **Flow of Events:**   1. They system responds by showing the admin home page and request the admin to select an option 2. The admin selects generate result sheet button. 3. The system responds by requesting the admin to select the student they wish to generate a result sheet for. 4. The admin enters the student’s details. 5. The system responds requestion the admin to choose which exam results they wish to use to generate a results sheet 6. The admin enters the exam details. 7. The system responds generation a result sheet |
| **Exit Condition (s):**   1. The admin has saved the changes and logged out. |
| **Exceptions:** |

|  |
| --- |
| **Use Case By:** DAVID O’MAHONY – K00236744 |
| **Use Case Name:** View Timetable (student) |
| **Use Case Description:** This will allow a student to view their timetable. |
| **Participating Actor(s):** Teacher |
| **Entry Conditions:**   1. A student wants to log in 2. The view student timetable function is invoked |
| **Flow of Events:**   1. They system responds by showing the student the login page 2. The student enters in login details 3. The system responds by verifying details and showing the student the home page with the option available. 4. The student clicks the view timetable option 5. The system responds by showing the student their timetable |
| **Exit Condition (s):**   1. The student logs out |
| **Exceptions:**  The student enters wrong log in detail.  The system responds by requestion them to enter the correct details |

|  |
| --- |
| **Use Case By:** DAVID O’MAHONY – K00236744 |
| **Use Case Name:** Register Student |
| **Use Case Description:** This will allow admin to register student |
| **Participating Actor(s):** Admin |
| **Entry Conditions:**   1. An admin has logged in 2. The register student function is invoked |
| **Flow of Events:**   1. They system responds by showing the admin home page and request the admin to select an option 2. The admin selects the register student button. 3. The system responds by displaying a form for the admin to enter. This form will have various details that must be entered e.g. name, age, address, course, student type etc. 4. The admin enters the student’s details. 5. The system responds by showing the students details. 6. The admin clicks the save button. 7. The system responds by saving changes to database. |
| **Exit Condition (s):**   1. The admin has saved the changes and logged out. |
| **Exceptions:** |

|  |
| --- |
| **Use Case By:** DAVID O’MAHONY – K00236744 |
| **Use Case Name:** Purchase a book |
| **Use Case Description:** This option will allow a user to purchase book. |
| **Participating Actor(s):** On-line User |
| **Entry Conditions:**  1. A user wants to buy a book.  2. The Purchase a book function is invoked |
| **Flow of Events:**  1. User searches website for a book(s).  2. System responds by searching for all matches relating to users search criteria.  3. User selects a book to purchase for search results.  4. System responds by creating shopping cart. adds selected item(s) to the shopping cart  5. Customer process by selecting the checkout button  6. System responds by invoking login function.  7. User confirms purchase.  8. System responds by verifying payment authentication, records order, send email confirmation to user. |
| **Exit Condition (s):** |
| **Exceptions:**  if payment is rejected, system notifies user and cancels order |

|  |
| --- |
| **Use Case By:** DAVID O’MAHONY – K00236744 |
| **Use Case Name:** Process timetable (student fulltime) |
| **Use Case Description:** This will Process timetable for a student fulltime |
| **Participating Actor(s):** Admin |
| **Entry Conditions:**   1. An admin has logged in 2. Process timetable (student fulltime) has been invoked |
| **Flow of Events:**   1. The system responds by displaying a list of all fulltime student and request the admin to select a student. 2. The admin selects a student. 3. The system responds by displaying the students details and for each lesion/instrument the student has chosen to learn the system requests the admin to select a room, time and day then click continue. 4. The admin enters the details and clicks continue 5. The system responds by checking availability and requests the admin to enter re-enter details if the availability check fails. 6. The system responds by displaying the student details and the timetable for the student then prompts the admin to click process or cancel. 7. The admin clicks procced. 8. The system responds by processing the timetable. |
| **Exit Condition (s):**   * The admin has processed the timetable. * The admin cancelled processing the timetable. |
| **Alternatives:**  7. the admin clicks cancel  7.1 The system responds by returning the admin to the admin page. |

|  |
| --- |
| **Use Case By:** DAVID O’MAHONY – K00236744 |
| **Use Case Name:** Modify Course details |
| **Use Case Description:** This will allow an admin to modify the details of a course |
| **Participating Actor(s):** Admin |
| **Entry Conditions:**  1. The admin selects the modify details option on the system.  2. The modify Course details function is invoked |
| **Flow of Events:**   1. Admin logs into the system 2. System responds by invoking login function 3. Admin selects the course to modify 4. System responds by displaying course 5. Admin modifies the course details and saves the changes 6. The system responds by saving changes to database |
| **Exit Condition (s):**  Admin logs out of system. |
| **Exceptions:** |

|  |
| --- |
| **Use Case By:** DAVID O’MAHONY – K00236744 |
| **Use Case Name:** Login/Logout |
| **Use Case Description:** This option will allow a member to login or logout |
| **Participating Actor(s):** On-line User |
| **Entry Conditions:**  1. A new member needs login or logout  2. The user invokes the login/logout feature on the website |
| **Flow of Events:**  1. The System responds by displaying a dialog box requesting the user to enter username and password, and to click Ok.  2. The user enters username and password and clicks Ok.  3. The system responds by checking if username and password are valid. If valid system will respond by displaying confirmation message and logs user in.  4. user choses to log out.  5. System responds by logging out. System displays home page. |
| **Exit Condition (s):** 1. User has logged in or logged out. |
| **Exceptions:**  System display message “username/Password is invalid” |

`

|  |
| --- |
| **Use Case By:** DAVID O’MAHONY – K00236744 |
| **Use Case Name:** Allocate teacher and room to student (private tuition) |
| **Use Case Description:** This will allow admin to allocate teacher and room to student (private tuition) |
| **Participating Actor(s):** Admin |
| **Entry Conditions:**   1. An admin has logged in 2. The allocate teacher and room to student (private tuition) function is invoked |
| **Flow of Events:**   1. They system responds by showing the admin home page and request the admin to select an option 2. The admin selects Allocate teacher and room to student (private tuition) button. 3. The system responds by requesting the admin to enter the student they wish for a teacher and a room to be allocated to. 4. The admin enters the student details. 5. The system responds requesting the admin to enter the teacher they wish to allocate. 6. The admin enters the teacher they want to allocate. 7. The system responds by showing the available rooms and times available for that teacher and student. 8. The admin selects the time and room for the teacher and student. 9. The system responds by saving the details to the database |
| **Exit Condition (s):**   1. The admin has saved the changes and logged out. |
| **Exceptions:** |

|  |
| --- |
| **Use Case By:** DAVID O’MAHONY – K00236744 |
| **Use Case Name:** Update Student Details |
| **Use Case Description:** This will allow admin to Update Student Details |
| **Participating Actor(s):** Admin |
| **Entry Conditions:**   1. An admin has logged in 2. The update student details function is invoked |
| **Flow of Events:**   1. They system responds by showing the admin home page and request the admin to select an option 2. The admin selects the update student details button. 3. The system responds by displaying a list of students and request the admin to select a student 4. The admin selects a student. 5. The system responds by showing the students details that can now be updated. 6. The admin updates the students details and clicks the save button. 7. The system responds by saving changes to database. |
| **Exit Condition (s):**   1. The admin has saved the changes and logged out. |
| **Exceptions:** |