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 colors and styles we use on the website is very important to the user’s experience, we must use colors 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 colors we use and the size of the objects in the site.

K00236744

3.3.2

Reliability

Reliability is very important when it comes to good functioning website. Up time requirements refers 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.

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.

|  |
| --- |
| 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. |
| Alternatives:  None. |
| Exceptions: |

|  |
| --- |
| 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. |
| Alternatives:  None. |
| Exceptions: |

|  |
| --- |
| 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. |
| Alternatives:  None. |
| Exceptions: |

|  |
| --- |
| 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 |
| Alternatives:  None. |
| Exceptions: |

|  |
| --- |
| 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. |
| Alternatives:  None. |
| Exceptions: |

|  |
| --- |
| 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 |
| Alternatives:  None. |
| Exceptions:  The student enters wrong log in detail.  The system responds by requestion them to enter the correct details |

|  |
| --- |
| 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. |
| Alternatives:  None. |
| Exceptions: |

|  |
| --- |
| 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): |
| Alternatives:  None. |
| Exceptions:  if payment is rejected, system notifies user and cancels order |

|  |
| --- |
| 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 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. |
| Alternatives:  None. |
| Exceptions: |

|  |
| --- |
| 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. |
| Alternatives:  None. |
| Exceptions:  System display message “username/Password is invalid” |

`

|  |
| --- |
| 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. |
| Alternatives:  None. |
| Exceptions: |

|  |
| --- |
| 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. |
| Alternatives:  None. |
| Exceptions: |