****

**Student Progress Tracking System**

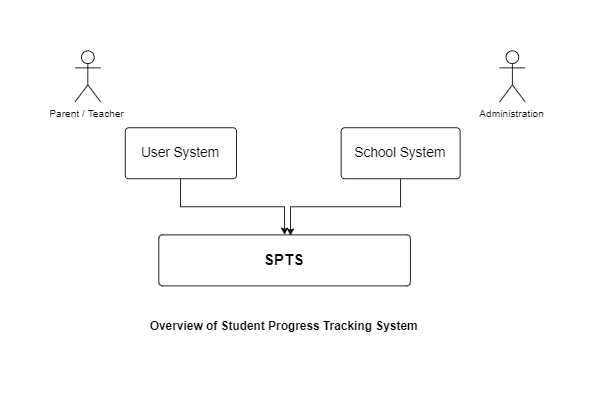
|  |  |
| --- | --- |
| Project Title | Under Study Tracking System |
| Objective: | The system is designed for “School System” to help parents and teachers to stay in communication to keep check on student also for tracking the student progress details. |
| Undertaken By: | Abdullah Hameed Khan  Muhammad Mustafa  Talal Bin Khizer Hayat  Muhammad Atif Javed |
| Supervised By: | Shakir Rashid Khan Khattak |
| Date Started: | November ,2022 |
| Date Completed: | November, 2022 |
| Tools & Technology | Technologies   1. React & Bootstrap & MUI 2. NodeJs & ExpressJs 3. JavaScript 4. MongoDB   Tools   1. MongoDB Compass & Community Server & Node js & Visual Studio Code |
| System Used: | 1. Simple 2.0 GHz 2. 512 MB free Memory or more 3. Internet Connection 4. Chrome Browser or any other 5. Localhost (127.0.0.1) |

**METAPHYSICAL**

This arrangement of **“Under study Tracking System”** for **“School System”** is planned in the light of prerequisites of school the executives. This framework worried about the computerization of all manual work identifying with parent, understudy, and educator individual record for the simplicity of organization/head. Keeping up with understudy participation record, assessment record, charge record, every day action, and schoolwork record physically by instructor influences showing abilities and execution of an educator.

During stockpiling of data certain approval checks have been given which works on the exactness of the framework and make it significant. Benefits of mechanized framework over manual framework are higher on the grounds that it gives the office of online recovery of data and records needed by the association.

## Product Overview



### Functional Requirements

Functional requirements describe the specific behavior of the system that what a system should do. It explains functions of a software system or its component. Functional requirements are supported by [non-functional requirements,](http://en.wikipedia.org/wiki/Non-Functional_Requirements) which impose constraints on the design or implementation (such as reliability, security or performance requirements).

Functional Requirement of the project are as follows:

### Sign in:

According to the scenario the unique id and Role is assigned to the user which is saved in the database So that parents can sign in with the assigned id (Email & Password) and can see the progress of their children, teachers can update the records and principal can keep a check on the records of students.

### Student report:

In this module the parent can see the progress of the children which is updated on the basis of their monthly test, class participation etc.

### Upcoming Events:

In upcoming events administration will add the events and the upcoming events of the week will be visible to the user on dashboard and these events can be updated anytime by the administration.

#### Fee Challan:

Challan form will be available every time parents can easily take a print and submit it rather than standing in queue.

### Feedback:

Parent can send feedback to the teacher which will be visible to the administration also and teacher and administration can send response on the received feedback related to student using this module.

### Appointment:

If an existing parent have missed the PTM organized and they want detailed discussion on the matter of their children, they can ask for an appointment which will be accepted or rejected accordingly by the school (teacher/principal).

### Attendance maintenance:

Parents can have a check on attendance to know whether his/her child is regular or not.

Teachers can mark attendance according to their class. Authentication is required to mark

And view attendance.

#### Leave applications:

Submitting leave application in person may be hectic sometimes, so parents will have the facility to submit the application regarding his child through this system just by a click away.

#### Class and Subject Management:

Principle and authority can add and remove class and relevant subjects. They can assign subjects to a

Particular class.

#### Holidays Management:

Authority can assign new holidays they will be available to the dashboard of end-user.

#### Exams Management:

Authority and Teachers can add new exam for a specific class with specific subject. They can add, update and delete exams.

### Non-Functional Requirements:

Nonfunctional requirement are requirements which shows the quality attribute of the system. Nonfunctional requirement of our application are as follows:

#### Availability:

The application is accessible anytime. It requires availability of internet to work.

#### Portability:

The application is reliable it is tested on multiple devices. It is working fine on all systems.

### Usability:

The interface is easy to understand and user can easily navigate within application.

#### Reliability:

System should be reliable. Abnormal behavior should not be shown by the application. Moreover, system must have less errors and maximum errors should be handled by the system without interrupting services provided to its user.

#### Performance Requirements:

The system should work according to the design and requirements of the user. The system should give fast response to the user.

#### Security:

If a user does not provide correct username or password, he/she cannot use the system.

### Product Perspective

The system will have a Graphical User Interface to enhance look and feel of the application and to provide usability. All pages of the system will follow consistent layout and clear structure Our Application is consists of following interfaces:

* GUI for Admin
* GUI for Parent
* GUI for Teacher

#### Graphical User Interface (GUI) for Parent

* Interface on a website.
* Parent has to enter sign in details in order to log in the authenticated account.
* Once logged in, parent can then check student’s report.
* Can see and download fee challan.
* Can view upcoming events & holidays.
* Can view and send feedback.
* Can send leave application.
* Can view and update profile.
* Can ask for appointment with teacher or principal.

### Graphical User Interface (GUI) For Teacher

* There is also interface for registered members.
* Registered users require login to access it.
* Once logged in teacher can perform various functions.
* Can add exam details.
* Can view and mark attendance.
* Can view and respond to a feedback.
* Can accept or reject appointment request.
* Can view and update profile.
* Can view upcoming events & holidays.

**View and Mark Attendance**

* Teacher can view student attendance, absent or leave.
* System will ask to select class and date of attendance.
* There is radio button to mark attendance in front of every student.
* Admin can mark attendance and save it by clicking the button.

**View and Manage Exams**

* Teacher can add new exam, update and delete.
* System will ask to select class date subject and of course total marks of exam.
* There is a field to add title and description of exam.

**View and Manage Leave Application**

* Teacher can view Leave Applications and details of each Application.
* System will ask to either Approve or Reject Application.

**Manage Appointment**

* Teacher can accept and reject the appointment.
* System will display all the appointment details.

**Daily Activity**

* Teacher can add new activity to a particular class and date with Title of an activity.
* System will display all the activities.
* Teacher can update and delete activity.

### Graphical User Interface (GUI) for Admin

* Interface for admins will be on website.
* Admins require login to access it.
* There areas in which admin can perform their operation are: Manage Students, Manage Teachers, Manage Class, Manage Subjects, Manage Events, Manage Holiday, Manage Exams, manage attendance, Manage Fee, manage appointments and Manage Feedback.
* Can view and update profile.

**Manage Students and Teachers and Parents**

* Admin can add, delete and update Students and Teachers.
* System will ask to enter all the details to add any Student. If their Parent already registered

You can select and then register new student. If parent not register yet you can register

new Parent.

* There is a delete button in front of all Teachers and Students to delete it.
* Admin can edit Students and teachers with the help of update option.

**Manage Events**

* Admin can add, delete and update the Events.
* System will ask to enter all the details to add any Events.
* There is a delete button in front of all Events to delete it.
* Admin can edit Events with the help of update option.

**Manage Class**

* Admin can add, delete and update the class.
* System will ask to enter all the details to add any class.
* There is a delete button in front of all classes to delete it.
* Admin can edit class with the help of update option.

**Manage Subjects**

* Admin can add, delete and update the subjects.
* System will ask to enter all the details to add any subject.
* There is a delete button in front on all subjects to delete it.
* Admin can edit subject with the help of update option.

**View Attendance**

* Admin can view student attendance, absent or leave.
* System will ask to select class and date of attendance.
* There is radio button to mark attendance in front of every student.
* Admin can mark attendance and save it by clicking the button.

**Manage Holiday**

* Admin can add, delete and update the Holiday.
* System will ask to enter all the details to add any Holiday.
* There is a delete button in front on all Holidays to delete it.

**View and Manage Exams**

* Admin can add new exam, update and delete.
* System will ask to select class date subject and of course total marks of exam.
* There is a field to add title and description of exam.

**View and Manage Leave Application**

* Admin can view Leave Applications and details of each Application.
* System will ask to either Approve or Reject Application.

**Daily Activity**

* Admin can add new activity to a particular class and date with Title of an activity.
* System will display all the activities.
* Admin can update and delete activity.

**Manage Fee**

* Admin can add, delete and update the Fee.
* System will ask to enter all the details to add Fee for any class.
* There is a delete button in front on all Fee Rows to delete it.
* Admin can edit fee with the help of update option.

**Manage Appointment**

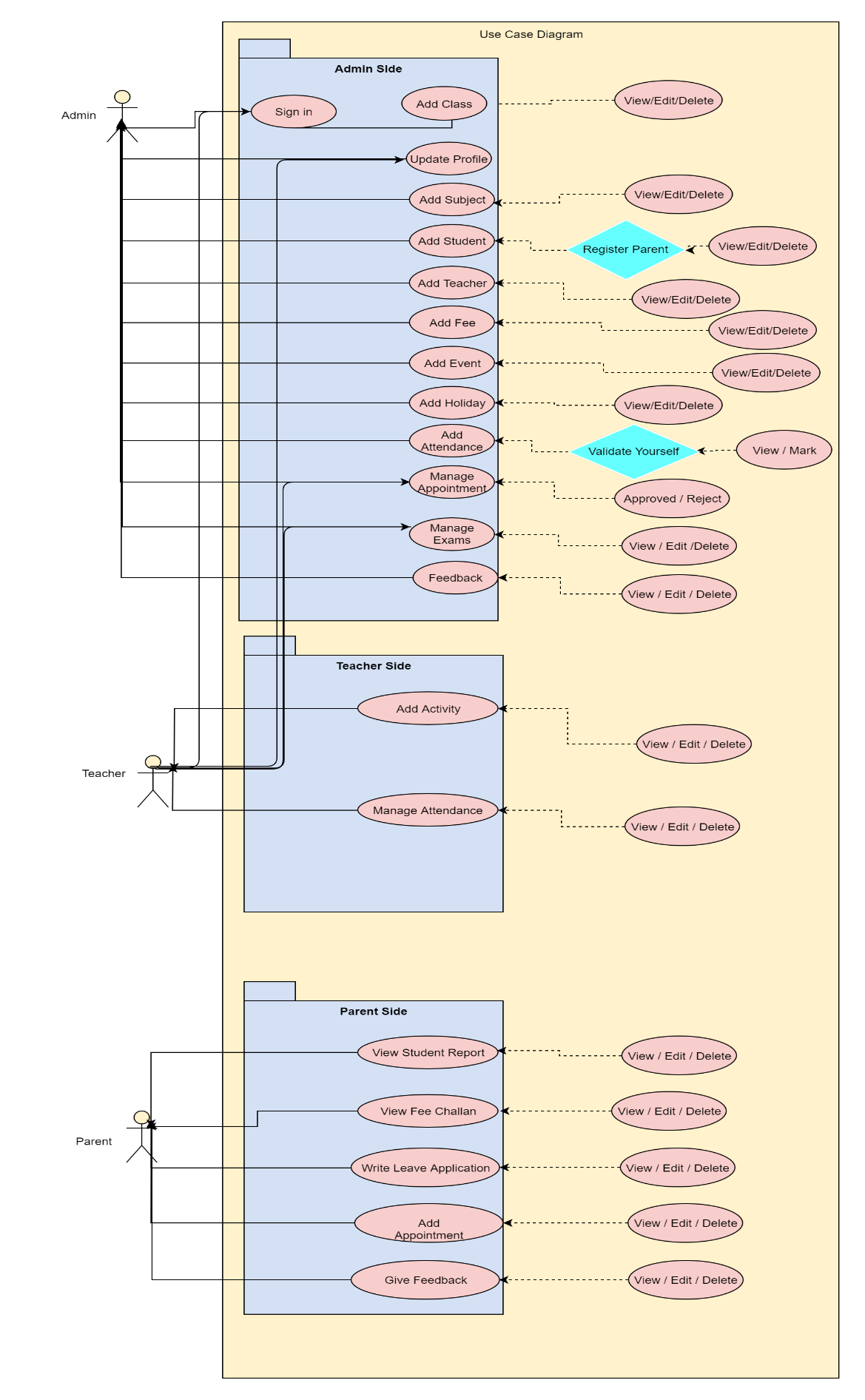
* Admin can accept and reject the appointment.
* System will display all the appointment details.
* Admin can approve or reject appointment.

**Manage Feedback**

* Admin can view feedback.
* System will show a list of received feedback.

### Use Case Diagram of Student Progress Tracking System:

Following is the use case diagram of the project:



#### Sign in:

|  |  |
| --- | --- |
| UC- 001 |  |
| Title | Sign In |
| Primary Actor | Parent, Teacher, Administration |
| Preconditions | Application must be installed in user’s device and user must visit the application. |
| Post Condition | User successfully Signed into application. |
| Main Success scenario | |
| Actor’s Action and Intension | System Responsibility |
| User click to the sign in button. | System displays the sign in fields |
| Alternative scenario | |
| User clicks the sign in button | System is not able to display the sign in fields |

#### Change Password:

|  |  |
| --- | --- |
| UC- 002 |  |
| Title | Change Password |
| Primary Actor | Parent, Teacher, administration. |
| Preconditions | user must sign in |
| Post Condition | Password updated successfully |

### Add Teacher:

|  |  |
| --- | --- |
| UC- 003 |  |
| Title | Add Teacher |
| Primary Actor | Administration |
| Preconditions | Admin must have a login id and should be logged in successfully. |
| Post Condition | Admin successfully adds the teacher |
| Main Success scenario | |
| Actor’s Action and Intension | System Responsibility |
| Admin clicks the add teacher | System displays the form where admin can enter teacher’s information |
| Alternative scenario | |
| Admin clicks the add teacher and it will not work | System is not able to add teacher |

#### Update Teacher:

|  |  |
| --- | --- |
| UC- 004 |  |
| Title | Update Teacher |
| Primary Actor | Administration |
| Preconditions | Admin must have a login id and should be logged in successfully. |
| Post Condition | Admin successfully updates the teacher details |
| Main Success scenario | |
| Actor’s Action and Intension | System Responsibility |
| Admin clicks the Update Teacher option | Systems opens the page where admin can update teacher details |
| Alternative scenario | |
| Admin clicks the Update Teacher option and it will not work | System is not able to open the page from where admin can update teacher details |

### 

### Delete Teacher:

|  |  |  |
| --- | --- | --- |
| UC- 005 |  | |
| Title | Delete Teacher | |
| Primary Actor | Administration | |
| Preconditions | Admin must have login id and must be logged in. | |
| Post Condition | Admin successfully deleted teacher. | |
| Main Success scenario | | |
| Actor’s Action and Intension | System Responsibility |
| Admin clicks the Delete Teacher Button | System displays success message |
| Alternative scenario | |
| Admin clicks the Delete Teacher and it will not work. | System is not able to delete teacher |

### Add Student:

|  |  |
| --- | --- |
| UC- 006 |  |
| Title | Add Student |
| Primary Actor | Administration |
| Preconditions | Admin must have a login id and should be logged in successfully. |
| Post Condition | Admin successfully adds the student |
| Main Success scenario | |
| Actor’s Action and Intension | System Responsibility |
| Admin clicks the add student | System displays the form where admin can enter student’s information |
| Admins selects new/existing parent | System displays success message |
| Alternative scenario | |
| Admin clicks the add student and it will not work | System is not able to add student | |

|  |  |
| --- | --- |
| UC- 007 |  |
| Title | Update Student |
| Primary Actor | Administration |
| Preconditions | Admin must have a login id and should be logged in successfully. |
| Post Condition | Admin successfully updates the student details |
| Main Success scenario | |
| Actor’s Action and Intension | System Responsibility |
| Admin clicks the Update Student option | Systems opens the page where admin can update student’s details |
| Alternative scenario | |
| Admin clicks the Update student option and it will not work | System is not able to open the page from where admin can update student details |

#### Delete Student:

|  |  |
| --- | --- |
| UC-008 | |
| Title | Delete Student |
| Primary Actor | Administration |
| Preconditions | Admin must have login id and must be logged in. |
| Post Condition | Admin successfully deleted student. |
| Main Success scenario | |
| Actor’s Action and Intension | System Responsibility |
| Admin clicks the Delete Student Button | System displays success message |
| Alternative scenario | |
| Admin clicks the Delete Student and it will not work. | System is not able to delete student |

#### Add Subject:

|  |  |
| --- | --- |
| UC- 009 |  |
| Title | Add Subject |
| Primary Actor | Administration |
| Preconditions | Admin must have a login id and should be logged in successfully. |
| Post Condition | Admin successfully adds the Subject |
| Main Success scenario | |
| Admin clicks the add subject | System displays the form where admin can enter subject ’s information | |
| Alternative scenario | | |
| Admin clicks the add subject and it will not work | System is not able to add subject | |

#### Update Subject:

|  |  |
| --- | --- |
| UC- 010 |  |
| Title | Update Subject |
| Primary Actor | Administration |
| Preconditions | Admin must have a login id and should be logged in successfully. |
| Post Condition | Admin successfully updates the subject details |
| Main Success scenario | |
| Actor’s Action and Intension | System Responsibility |
| Admin clicks the Update Subject option | Systems opens the page where admin can update Subject details |
| Alternative scenario | |
| Admin clicks the Update Subject option and it will not work | System is not able to open the page from where admin can update subject details |

#### Delete Subject:

|  |  |
| --- | --- |
| UC- 011 |  |
| Title | Delete Subject |
| Primary Actor | Administration |
| Preconditions | Admin must have login id and must be logged in. |
| Post Condition | Admin successfully deleted Subject |
| Main Success scenario | |
| Actor’s Action and Intension | System Responsibility |
| Admin clicks the Delete Subject Button | System displays success message |
| Alternative scenario | |
| Admin clicks the Delete Subject and it will not work. | System is not able to delete subject |

### 

### Manage Class:

|  |  |
| --- | --- |
| UC- 012 |  |
| Title | Manage Class |
| Primary Actor | Administration |
| Preconditions | Admin must have login id and must be logged in. |
| Post Condition | Admin successfully add/edit/delete class |
| Main Success scenario | |

|  |  |
| --- | --- |
| Actor’s Action and Intension | System Responsibility |
| Admin clicks the add/edit/delete Button | System displays success message |
| Alternative scenario | |
| Admin clicks the add/edit/delete Subject and it will not work. | System is not able to add/edit/delete subject |

##### Manage Events:

|  |  |
| --- | --- |
| UC- 013 |  |
| Title | Manage Events |
| Primary Actor | Administration |
| Preconditions | Admin must have login id and must be logged in. |
| Post Condition | Admin successfully add/edit/delete Event |
| Main Success scenario | |
| Actor’s Action and Intension | System Responsibility |
| Admin clicks the add/edit/delete event Button | System displays success message |
| Alternative scenario | |
| Admin clicks the add/edit/delete and it will not work. | System is not able to add/edit/delete subject |

##### Manage Holiday:

|  |  |
| --- | --- |
| UC- 014 |  |
| Title | Manage Holiday |
| Primary Actor | Administration |
| Preconditions | Admin must have login id and must be logged in. |
| Post Condition | Admin successfully Manage holiday |
| Main Success scenario | |
| Actor’s Action and Intension | System Responsibility |
| Admin clicks the Add/Edit/Delete holiday Button | System displays success message |
| Alternative scenario | |
| Admin clicks the Add/Edit/Delete holiday and it will not work. | System is not able to add/edit/delete holiday |

#### Manage Attendance:

|  |  |
| --- | --- |
| UC-015 | |
| Title | Manage Attendance |
| Primary Actor | Teacher |
| Preconditions | Teacher must have login id and must be logged in. |
| Post Condition | Teacher successfully marked attendance |
| Main Success scenario | |
| Actor’s Action and Intension | System Responsibility |
| Teacher clicks the present/absent/leave Subject Button | System displays success message and hide the button to again submit attendance |
| Alternative scenario | |
| Admin clicks the present/absent/leave Subject Button and it will not work. | System is not able to mark attendance |

#### Manage Appointment:

|  |  |
| --- | --- |
| UC- 016 |  |
| Title | Manage Appointment |
| Primary Actor | Parent, Teacher, Admin |
| Preconditions | User must have login id and logged in success fully. |
| Post Condition | User should be able to respond to the request /made request |
| Main Success scenario | |
| Actor’s Action and Intension | System Responsibility |
| User click to make appointment  User should be able to respond to the request made | System displays the available time and date. System should display the request made. |

|  |  |
| --- | --- |
| Alternative scenario | |
| User clicks make appointment and system does not perform any action. | System is not able to display the appointment form. |

#### Manage Exam:

|  |  |
| --- | --- |
| UC- 017 |  |
| Title | Manage Exam |
| Primary Actor | Teacher, Admin |
| Preconditions | User must have login id and logged in success fully |
| Post Condition | User successfully add exam details. |
| Main Success scenario | |
| Actor’s Action and Intension | System Responsibility |
| User clicks the exam icon. | System displays the exam form. |
| Alternative scenario | |
| User clicks the exam portion and it will not work. | System is not working properly. |

#### Manage Feedback:

|  |  |
| --- | --- |
| UC- 018 |  |
| Title | Manage Feedback |
| Primary Actor | Administration, Teacher, Parent |
| Preconditions | Application must be installed in user’s system and user must visit be logged in |
| Post Condition | User successfully give the feedback |
| Main Success scenario | |
| Actor’s Action and Intension | System Responsibility | |
| User clicks the feedback portion. | System displays the feedback dialog box where he can view/give their feedback. | |
| Alternative scenario | | |
| User clicks the feedback portion and it will not work. | System is not able to display the feedback dialog box. | |

#### Add Fee:

|  |  |
| --- | --- |
| UC- 019 |  |
| Title | Add Fee |
| Primary Actor | Administration |
| Preconditions | Application must be installed in user’s system and user must be logged in |
| Post Condition | Fee is added successfully |
| Main Success scenario | |
| Actor’s Action and Intension | System Responsibility |
| User clicks the Add Fee option. | System show the form to add fee |
| Alternative scenario | |
| User clicks the Add Fee option. and it will not work. | System is not able to show form to add fee | |

#### Leave Application:

|  |  |
| --- | --- |
| UC- 020 |  |
| Title | Leave Application |
| Primary Actor | Parent |
| Preconditions | Application must be installed in user’s system and user must visit be logged in |
| Post Condition | User successfully submits the Leave Application |
| Main Success scenario | |
| Actor’s Action and Intension | System Responsibility |
| User clicks the Leave Application portion. | System displays the dialog box where parent can select the student and write application |
| Alternative scenario | |
| User clicks the Leave Application portion and it will not work | System is not able to display the dialog box to send leave application |

#### Student Report:

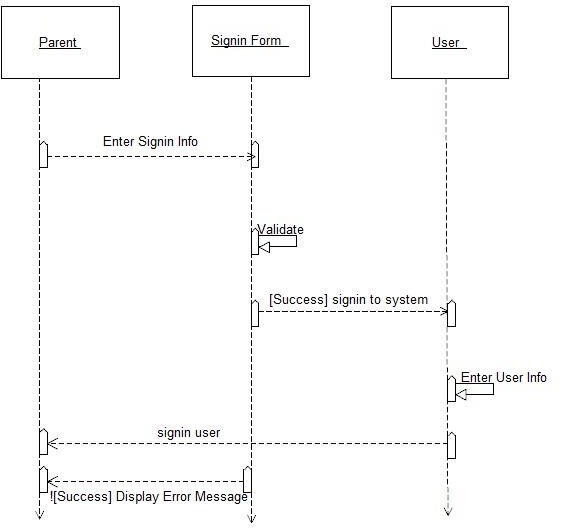
|  |  |
| --- | --- |
| UC-021 | |
| Title | Student Report |
| Primary Actor | Parent |
| Preconditions | Application must be installed in user’s system and user must be logged in |
| Post Condition | User successfully viewed student’s report |
| Main Success scenario | |
| Actor’s Action and Intension | System Responsibility |
| User clicks the Student Report option. | System shows the list of student/students with report option |
| User selects the student/child to see report | System displays the report of selected student |
| Alternative scenario | |
| User clicks the Student Report option and it will not work. | System is not able to show the report of student |
|  | |

#### Sign Out:

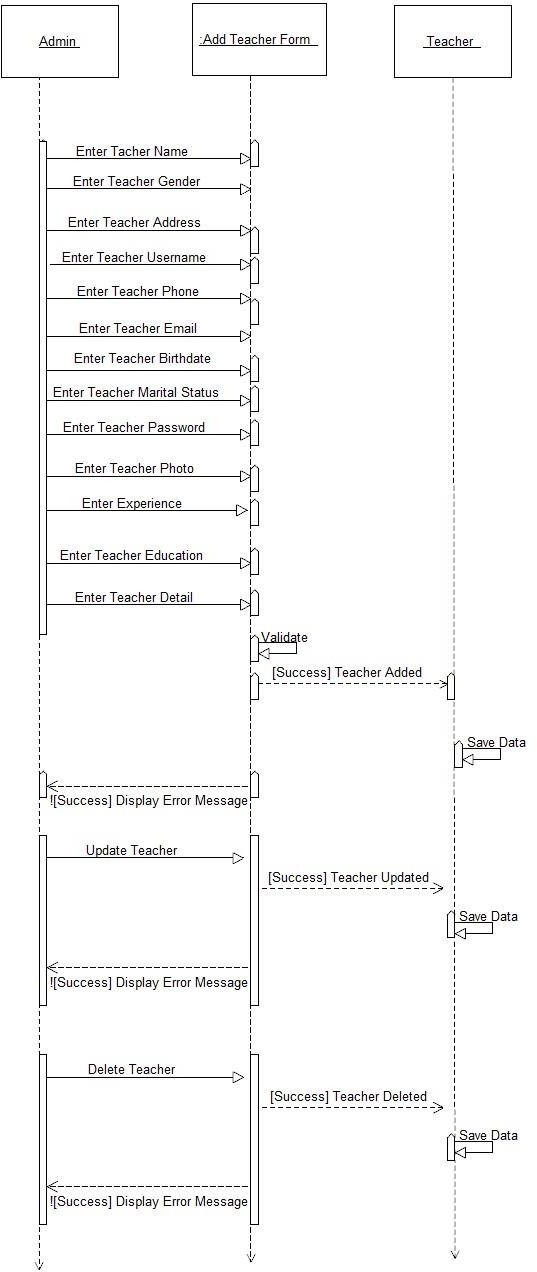
|  |  |
| --- | --- |
| UC- 025 |  |
| Title | Sign out. |
| Primary Actor | Parent, Teacher, Administration |
| Preconditions | Application must be installed in user’s system and user must be logged in |
| Post Condition | User successfully sign out of the application. |
| Main Success scenario | |
| Actor’s Action and Intension | System Responsibility |
| User clicks the sign out option. | System sign out the user and he/she is not able to use the features of app until and unless sign in again. |
| Alternative scenario | |
| User clicks the sign out option and it will not work. | System is not able to sign out the user out of the application. |

#### System Sequence Diagrams:

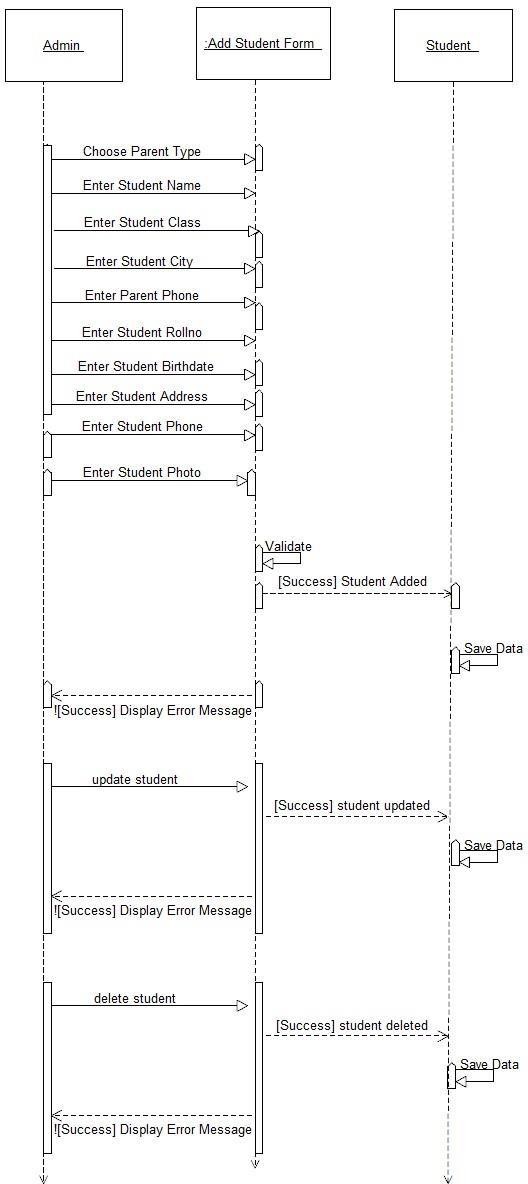
#### Sign In (Admin,Teacher,Parent)



#### Manage Teacher

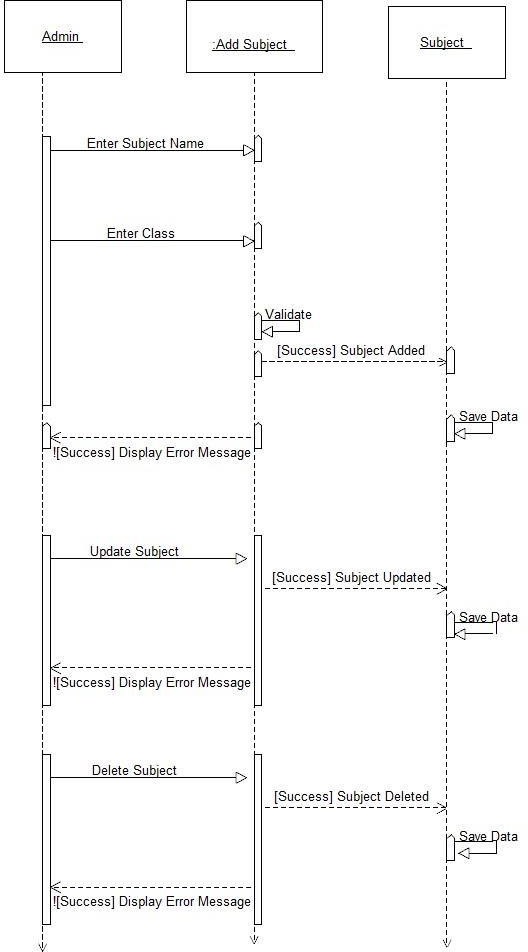


#### Manage Student

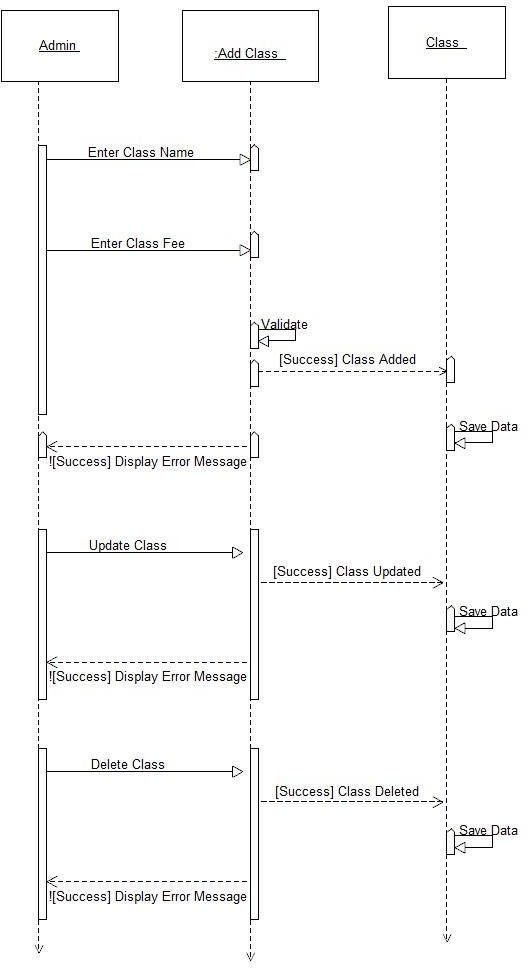


Enter Parent Email

#### Manage Subject



#### Manage Class



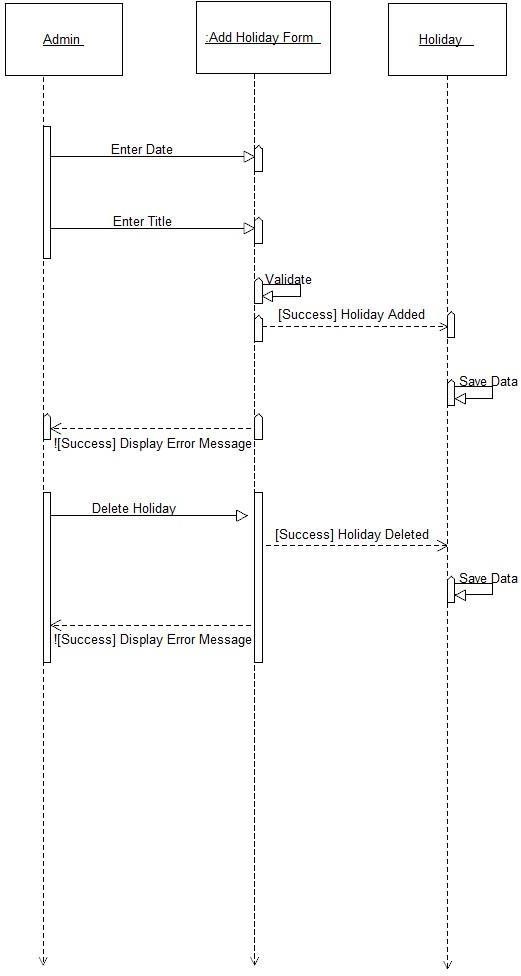
#### Manage Exams

##### 

#### Manage Fee

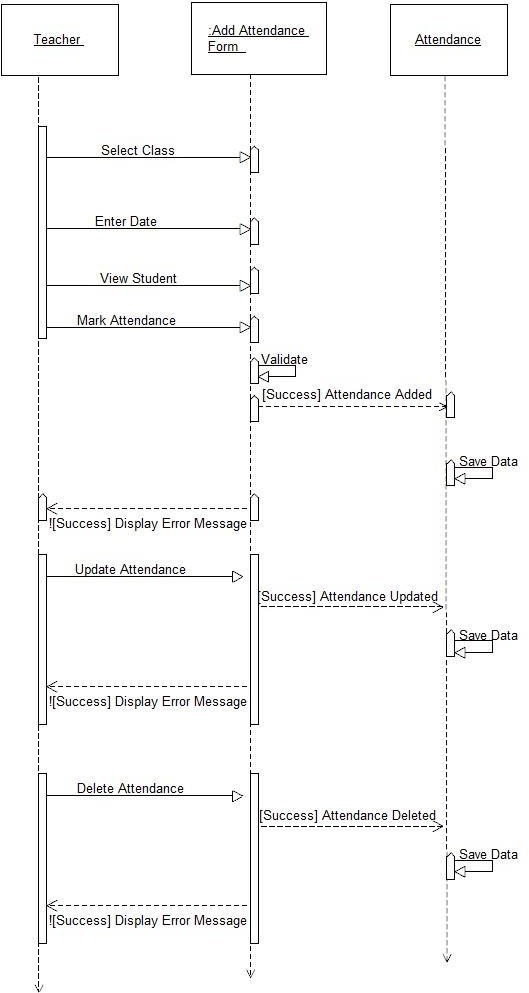
#### Manage Feedbacks

#### Manage Holidays



#### Manage Events

#### Manage Attendance



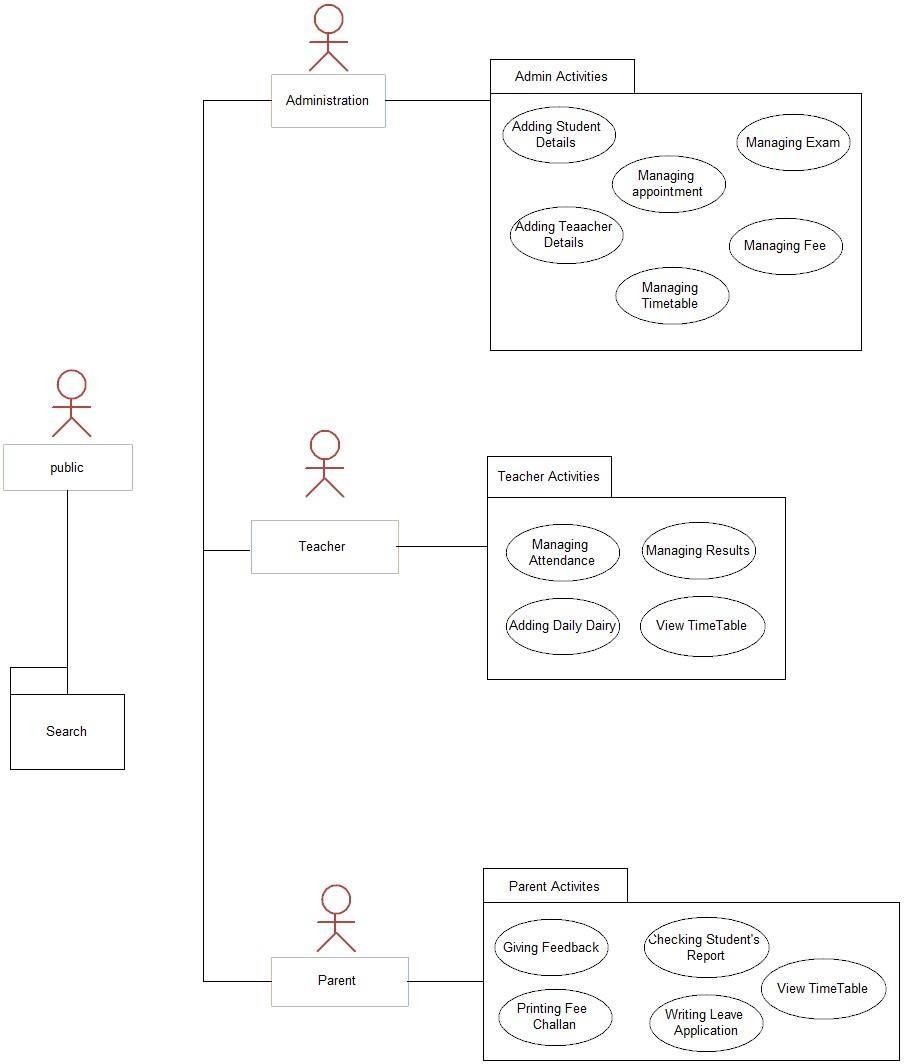
#### Sign Out (Admin,Teacher,Parent)

### Class Diagram

### 

### Package Diagram:

Package diagram, a kind of structural diagram, shows the arrangement and organization of model elements in middle to large scale project. Package diagram can show both structure and dependencies between sub-systems or modules, showing different views of a system.



### Software Testing Principle

* + - It is not possible to test a program completely.
    - It is a risk-based exercise.
    - Testing cannot show that bugs don't exist but they show that it exists.
    - No of bugs are not limited if there are bugs you find then there must be more bugs.
    - All the bugs you find cannot be fixed.
    - Software specifications are never final.

### Verification and Validation

Verification shows that system is according to customer needs or not, for this purpose we allowed our customer to use the system by themselves using dummy data and verify the needs and we found them satisfied with the functionalities which includes parent & staff’s communication ease, parent having report of his/child so that progress can be checked easily.

### White Box, Black Box and Grey Box Testing

While designing test cases, to describe the point of view software engineer use following terms for testing which are described below:

**Black box** testing considers the software as a black-box without any understanding as to how the internal codes behave. As a result, the tester inputs data and only sees the output from the object. This level of testing usually requires thorough test cases to be provided to the tester who then can simply verify that for a given input, output value, and verify that is it the same value as the expected value mentioned in the test case.

**White box** testing, is the testing in which the tester has access to the internal data structures, code, and algorithms. Due to this reason, [unit testing](http://en.wikipedia.org/wiki/Unit_testing) [a](http://en.wikipedia.org/wiki/Unit_testing)nd debugging can be categorized as white-box testing and it usually requires writing code and thus it requires more skill than the black-box tester.

**Grey box** testing could be used in the situation of testing a client-server environment when the tester has control over the input, checks the value in a NoSQL database, and the output value, and then compares all three (the input, NoSQL value, and output), to determine if the data got affected on the database insertion or retrieval.

#### Levels of Testing

* + - **Unit testing** is used to tests the software component, or module. Each unit (basic component) of the software is tested to verify that the detailed design for each unit has been correctly implemented. In an Object-oriented environment, this is usually at the class level, and the unit tests include the constructors and destructors.
    - **Integration testing** identify defects in the interfaces and interaction between integrated components (modules).
    - **Functional testing** tests at any level (class, module, interface, or system) for proper functionality as defined in the requirements.
    - **System testing** tests a completely integrated system to verify that it meets its requirements.
    - **System integration testing** verifies that a system is integrated to any external or third party systems defined in the system requirements.
    - **Acceptance testing** can be conducted by the end-user, Customer, or client to validate whether or not to accept the product. Acceptance testing may be performed as part of the process between any two phases of development. Following are the *Development* stage of **Alpha testing** is a testing done by potential users/Customers or an independent test team at the developers' site. Alpha testing is often done for off-the-shelf software as a form of internal acceptance testing, before the software goes to beta testing.

o **Beta testing** comes after alpha testing. Versions of the software, known as beta versions, are released to a limited audience outside of the company. The software is released to groups of people so that further testing can identify that the product has few faults or bugs. Sometimes, beta versions are available to the open public to increase the feedback field to a maximal number of future users.

## TEST CASES:

Test cases of the project are shown under this section.

#### Test Case 1: Registration

|  |  |  |
| --- | --- | --- |
| Test Case ID: | TC-01 | |
| Test Case Name: | Registration of teacher/student | |
|  |  | |
| Functionality to be tested: | This scenario covers the functionality of registering new users | |
| Actor: | Admin | |
| Normal Flow | **Expected Results** | Actual Results (Pass/Fail) |
| 1. Enter empty value in required fields. 2. Enter existing value for User Name. | 1. Display error message “please fill the required fields” 2. Display warning message “User Name already exists” | 1. Pass 2. Pass 3. Pass |
| 1. Enter empty value for Password. 2. Enter all the required fields with correct data. | 1. Display error message “Password required.” 2. Display success message “User added successfully.” | 4.Pass |
| Executed by: | Abdullah Hameed Khan  Muhammad Mustafa  Talal Bin Khizer Hayat  Muhammad Atif Javed | |
| Date: | 14-November, 2022 | |

#### Test Case 2: Search

|  |  |  |
| --- | --- | --- |
| Test Case ID: | TC-02 |  |
| Test Case Name: | Search | |
| Functionality to be tested: | Successfully search the information required by user. | |
| Actor: | User | |
| Normal Flow | **Expected Results** | Actual Results (Pass/Fail) |
| 1. On the click of search bar. 2. On typing in search field. 3. After finish writing keyword in search field. | 1. Searched results should be according to search keyword 2. Exact match or similar match should be displayed. 3. Searched results should be displayed | 1. Pass 2. Pass 3. Pass |
| Executed by: | Abdullah Hameed Khan  Muhammad Mustafa  Talal Bin Khizer Hayat  Muhammad Atif Javed | |
| Date: | 14-November, 2022 | |

#### Test Case 3: Update Admin Profile

|  |  |  |
| --- | --- | --- |
| Test Case ID: | TC-03 |  |
| Test Case Name: | Update Admin profile | |
| Functionality to be tested: | Admin selects Update Profile from sidebar to update | |
| Actor: | Admin | |
| Normal Flow | **Expected Results** | Actual Results (Pass/Fail) |
| 1. Admin click the profile | 1. A window open in which admin profile information is mentioned. 2. Admin can update their information in opened window 3. Information should be updated. | Pass Pass |
| button.  2. Admin click on the | Pass |
| profile information to |  |
| update. |  |
| 3. Admin click on the update button. |  |
| Executed by: | Abdullah Hameed Khan  Muhammad Mustafa  Talal Bin Khizer Hayat  Muhammad Atif Javed | |
| Date: | 14-November, 2022 | |

#### Test Case 4: Update Student/Teacher Profile

|  |  |  |
| --- | --- | --- |
| Test Case ID: | TC-04 | |
| Test Case Name: | Update Student/Teacher profile | |
| Functionality to be tested: | Admin selects update profile from list of users. | |
| Actor: | Admin | |
| Normal Flow | **Expected Results** | Actual Results (Pass/Fail) |
| 1. Admin click the profile button. 2. Admin click on the profile information to update. 3. Admin click on the update button. | 1. A window open in which user profile information is mentioned. 2. Admin can update the information in opened window. 3. Information should be updated. | 1. Pass 2. Pass 3. Pass |
| Executed by: | Abdullah Hameed Khan  Muhammad Mustafa  Talal Bin Khizer Hayat  Muhammad Atif Javed | |
| Date: | 14-November, 2022 | |

#### Test Case 5: Manage Class

|  |  |  |
| --- | --- | --- |
| Test Case ID: | TC-05 |  |
| Test Case Name: | Manage Class | |
| Functionality to be tested: | Admin select and manage class management from sidebar | |
| Actor: | Admin | |
| Normal Flow | **Expected Results** | Actual Results (Pass/Fail) |
| 1. Admin click the class management. 2. Enter empty value in class field. 3. Enter empty value in class fee field. 4. Enter existing value in class name field 5. Click on update button 6. Click on delete button | 1. A window opens which contains details of existing classes with a form to add new class. 2. Display error message “class name required.” 3. Display error message “class fee required.” 4. Display error message “class already exists” 5. A window opens with existing data where admin can edit the details and save them. | 1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass |
|  | 6. Delete the class and display success message “Selected class deleted successfully” |  |
| Executed by: | Abdullah Hameed Khan  Muhammad Mustafa  Talal Bin Khizer Hayat  Muhammad Atif Javed | |
| Date: | 14-November, 2022 | |

#### Test Case 6: Manage Subjects

|  |  |  |
| --- | --- | --- |
| Test Case ID: | TC-06 |  |
| Test Case Name: | Manage Subjects | |
| Functionality to be tested: | Admin select and manage subject management from sidebar | |
| Actor: | Admin | |
| Normal Flow | Expected Results | Actual Results (Pass/Fail) |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 1. Admin click the subject management. 2. Enter empty value in subject name field. 3. Enter existing subject with class in subject name value. 4. Click on update button 5. Click on delete button | 1. A window opens which contains details of existing classes with a form to add new class. 2. Display error message “subject name required.” 3. Display error message “subject already exists in the class” 4. A window opens with existing data where admin can edit the details and save them. 5. Delete the subject and display success message “Selected Subject deleted successfully” | 1. Pass 2. Pass 3. Pass 4. Pass 5. Pass |
| Executed by: | Abdullah Hameed Khan  Muhammad Mustafa  Talal Bin Khizer Hayat  Muhammad Atif Javed | |
| Date: | 14-November, 2022 | |

#### Test Case 7: Mark Student Attendance

|  |  |  |
| --- | --- | --- |
| Test Case ID: TC-07 | | |
| Test Case Name: | Mark Student Attendance | |
| Functionality to be tested: | User select and mark Student attendance from sidebar | |
| Actor: | Admin, Teacher | |
| Normal Flow | **Expected Results** | Actual Results (Pass/Fail) |
| 1. press view attendance button without selecting class and date 2. Mark attendance and save data. | 1. Display error message “please select class and date.” 2. Display success message “Student Attendance added successfully.” | 1. Pass 2. Pass |
| Executed by: | Abdullah Hameed Khan  Muhammad Mustafa  Talal Bin Khizer Hayat  Muhammad Atif Javed | |
| Date: | 14-November, 2022 | |

#### Test Case 8: Manage Exam

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case ID: | TC-08 | | | |
| Test Case Name: | | Manage Exam | |
| Functionality to be tested: | | Admin select and manage exam management from sidebar | |
| Actor: | | Admin, Teacher | |
| Normal Flow | | **Expected Results** | Actual Results (Pass/Fail) |
| 1. User click the exam management. 2. Enter empty value in required field. 3. Click on update button 4. Click on delete button | | 1. A window opens which contains details of existing exams with a form to add new exam 2. Display error message “please fill the required fields.” 3. A window opens with existing data where admin can edit the details and save them. 4. Delete the exam and display success message “Selected exam deleted successfully” | 1. Pass 2. Pass 3. Pass 4. Pass |
| Executed by: | | Abdullah Hameed Khan  Muhammad Mustafa  Talal Bin Khizer Hayat  Muhammad Atif Javed | |
| Date: | | 14-November, 2022 | |

#### Test Case 9: Manage Events

|  |  |  |
| --- | --- | --- |
| Test Case ID: | TC-9 |  |
| Test Case Name: | Manage Events | |
| Functionality to be tested: | Admin select and manage event management from sidebar | |
| Actor: | Admin | |
| Normal Flow | **Expected Results** | Actual Results (Pass/Fail) |
| 1. User click the events management. 2. Enter empty value in required field. 3. Click on update button | 1. A window opens which contains details of existing events with a form to add new event 2. Display error message “please fill the required fields.” | 1. Pass 2. Pass 3. Pass | |
| 4. Click on delete button | 1. A window opens with existing data where admin can edit the details and save them. 2. Delete the event and display success message “Selected event deleted successfully” | Pass | |
| Executed by: | Abdullah Hameed Khan  Muhammad Mustafa  Talal Bin Khizer Hayat  Muhammad Atif Javed | | |
| Date: | 14-November, 2022 | | |

#### Test Case 10: Manage Holiday

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case ID: | TC-10 | |  |
| Test Case Name: | Manage Holiday | | |
| Functionality to be tested: | Admin select and manage Holiday management from sidebar | | |
| Actor: | Admin | | |
| Normal Flow | **Expected Results** | | Actual Results (Pass/Fail) |
| 1. User click the Holiday management. 2. Enter empty value in required field. 3. Click on update button 4. Click on delete button | | 1. A window opens which contains details of existing Holiday with calendar and form to add new Holiday 2. Display error message “please fill the required fields.” 3. A window opens with existing data where admin can edit the details and save them. 4. Delete the holiday and display success message “Selected holiday deleted successfully” | 1. Pass 2. Pass 3. Pass 4. Pass | |
| Executed by: | | Abdullah Hameed Khan and TEAM | | |
| Date: | | 14-November, 2022 | | |

#### Test Case 11: Manage Fee

|  |  |  |
| --- | --- | --- |
| Test Case ID: | TC-11 |  |
| Test Case Name: | Manage Fee | |
| Functionality to be tested: | Admin select and manage Fee management from sidebar | |
| Actor: | Admin | |
| Normal Flow | Expected Results | Actual Results (Pass/Fail) |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 1. User click the Manage Fee types or add student fee 2. Enter empty value in required field. 3. Click on update button for manage fee types 4. Click on delete button | 1. A window opens which contains existing Fee details with a form to add new entry. 2. Display error message “please fill the required fields.” 3. A window opens with existing data where admin can edit the details and save them. 4. Delete the selected Fee details and display success message “Selected Fee   deleted successfully” | 1. Pass 2. Pass 3. Pass 4. Pass |
| Executed by: | Abdullah Hameed Khan  Muhammad Mustafa  Talal Bin Khizer Hayat  Muhammad Atif Javed | |
| Date: | 14-November, 2022 | |

#### Test Case 12: View Student Report

|  |  |  |
| --- | --- | --- |
| Test Case ID: | TC-12 |  |
| Test Case Name: | Manage Noticeboard | |
| Functionality to be tested: | Parent can select and view children report from sidebar | |
| Actor: | Parent | |
| Normal Flow | **Expected Results** | Actual Results (Pass/Fail) |
| 1. User click Children’s Report 2. Selects the required field 3. Clicks print option in result field | 1. A window opens which contains list of children 2. Display the details of student 3. Open a new window in which printing options can be selected. | 1. Pass 2. Pass 3. Pass |
| Executed by: | Abdullah Hameed Khan  Muhammad Mustafa  Talal Bin Khizer Hayat  Muhammad Atif Javed | |

|  |  |
| --- | --- |
| Date: | 14-November, 2022 |

#### Test Case 13: Send Leave Application

|  |  |  |
| --- | --- | --- |
| Test Case ID: TC-13 | | |
| Test Case Name: | Send Leave Application | |
| Functionality to be tested: | Parent select Leave application from sidebar | |
| Actor: | Parent | |
| Normal Flow | **Expected Results** | Actual Results (Pass/Fail) |
| 1. User click send leave application. | 1. A window opens which contains fields for leave application. | Pass |
| 1. Enter empty value of leave application. 2. Clicks send application | 1. Display warning message “please fill the required fields” 2. Display success message “leave application sent successfully” | 1. Pass 2. Pass |
| Executed by: | Abdullah Hameed Khan  Muhammad Mustafa  Talal Bin Khizer Hayat  Muhammad Atif Javed | |
| Date: | 14-November, 2022 | |

Table 6.13 Test case of Send Leave Application

#### Test Case 14: View Leave Application

|  |  |  |
| --- | --- | --- |
| Test Case ID: | TC-14 | |
| Test Case Name: | View Leave Application | |
| Functionality to be tested: | User select Leave application from sidebar | | |
| Actor: | Admin, Teacher | | |
| Normal Flow | **Expected Results** | Actual Results (Pass/Fail) | |
| 1. User click view leave application. 2. Click see leave application to open details. | 1. A window opens which contains list of received leave applications. 2. Show all the details of received leave application | 1. Pass 2. Pass | |
| Executed by: | Abdullah Hameed Khan  Muhammad Mustafa  Talal Bin Khizer Hayat  Muhammad Atif Javed | | |
| Date: | 14-November, 2022 | | |

Table 6.14 Test case of View Leave Application

#### Test Case 15: Request Appointment

|  |  |  |
| --- | --- | --- |
| Test Case ID: | TC-15 |  |
| Test Case Name: | Request Appointment | |
| Functionality to be tested: | Parent select Request Appointment from sidebar | |
| Actor: | Parent | |
| Normal Flow | **Expected Results** | Actual Results (Pass/Fail) |
| 1. User click request | 1. A window opens which contains fields to request appointment. 2. Display warning message “please select time and date” 3. Display warning message “please fill the required fields” | Pass |
| appointment  2. Clicks send button | Pass |
| without selecting date  or time | Pass |
| 3. Clicks send button |  |
| with empty field for |  |
| appointment detail |  |
| Executed by: | Abdullah Hameed Khan  Muhammad Mustafa  Talal Bin Khizer Hayat  Muhammad Atif Javed | |
| Date: | 14-November, 2022 | |

Table 6.15 Test case of Request Appointment

#### Test Case 16: View Appointment Request

|  |  |  |
| --- | --- | --- |
| Test Case ID: TC-16 | | |
| Test Case Name: | View Appointment Request | |
| Functionality to be tested: | User select Request Appointment from sidebar | |
| Actor: | Admin, Teacher | |
| Normal Flow | **Expected Results** | Actual Results (Pass/Fail) |
| 1. User click view Appointment Request 2. Click reject appointment 3. Click approve appointment | 1. A window opens which contains list of received appointment requests. 2. A feedback text box appears to add rejection feedback, status changed to   “rejected”   1. status changed to “approved” | 1. Pass 2. Pass 3. Pass |
| Executed by: | Abdullah Hameed Khan  Muhammad Mustafa  Talal Bin Khizer Hayat  Muhammad Atif Javed | |
| Date: | 14-November, 2022 | |

#### Test Case 17: Send Feedback

|  |  |  |
| --- | --- | --- |
| Test Case ID: | TC-17 |  |
| Test Case Name: | Send Feedback | |
| Functionality to be tested: | Parent select Send Feedback from sidebar | |
| Actor: | Parent | |
| Normal Flow | **Expected Results** | Actual Results (Pass/Fail) |
| 1. User click Send Feedback 2. Clicks send button with empty field for Feedback. | 1. A window opens which contains fields to request appointment. 2. Display warning message “please fill the required fields” | 1. Pass 2. Pass |
| Executed by: | Abdullah Hameed Khan  Muhammad Mustafa  Talal Bin Khizer Hayat  Muhammad Atif Javed | |
| Date: | 14-November, 2022 | |

Table 6.17 Test case of Send Feedback

#### Test Case 18: View Feedback

|  |  |  |
| --- | --- | --- |
| Test Case ID: | TC-18 |  |
| Test Case Name: | View feedback | |
| Functionality to be tested: | User select View feedback from sidebar | |
| Actor: | Admin, Teacher | |
| Normal Flow | **Expected Results** | Actual Results (Pass/Fail) |
| 1. User click view feedback 2. Click on response field 3. Clicks send response | 1. A window opens which contains list of received feedbacks 2. Can enter response for received feedback 3. Display success message “response sent successfully” | 1. Pass 2. Pass 3. Pass |
| Executed by: | Abdullah Hameed Khan  Muhammad Mustafa  Talal Bin Khizer Hayat  Muhammad Atif Javed | |
| Date: | 14-November, 2022 | |

#### Test Case 19: Press logout Button

|  |  |  |
| --- | --- | --- |
| Test Case ID: | TC-19 | |
| Test Case Name: | Press logout Button | |
| Functionality to be tested: | User presses logout Button to logout from application | |
| Actor: | User | | |
| Normal Flow | **Expected Results** | Actual Results (Pass/Fail) | |
| 1. User presses logout button on screen. | 1. Upon button selection of sign out option user will log out. | Pass | |
| Executed by: | Abdullah Hameed Khan  Muhammad Mustafa  Talal Bin Khizer Hayat  Muhammad Atif Javed | | |
| Date: | 14-November, 2022 | | |