|  |  |  |  |
| --- | --- | --- | --- |
| **Software Project Management (SE3080) : Assessment 2** | | | |
| **Interfaces of the Proposed System** | | | |
|  | | **Group No** | **2021S2\_REG\_WE\_02** |
| **Registration No** | **IT19004778** | | |
| **Student**  **Name** | **W.G.M.S.Wickramarathna** | | |
| **Function(s)**  **Note**: Include the functions required to complete for **sprint 1** & **sprint 2** | **Sprint 1**   * Teacher Mark Attendance (Attendance Management) * Teacher Edit Attendance (Attendance Management) * Admin view attendance (Attendance Management) * Login * Teacher Add Marks (Exam Marks Management)   **Sprint 2**   * Teacher Edit Marks (Exam Marks Management) * Admin View/Delete Marks (Exam Marks Management) * Admin or Teacher generate Attendance Report (Attendance Management) * Admin or Teacher generate Marks Report (Exam Marks Management) | | |
| **Sprint 1** | **Picture of interface 1** | **Brief description of interface 1** | |
| Graphical user interface  Description automatically generated | **Purpose**: This interface will be used to mark the attendance of students.  **Flow**: The user is required to insert a class id and press the search button. Then the user will be able to see the list of the class. After that, the user can mark attendance using the toggle button.  **Good practices**:   * The interface was designed according to the single responsibility principle. It means focuses on mark attendance tasks only. * All the buttons and labels are well named, have correct spellings, and start with a capital letter. * Input fields have given real-world metaphors or icons which will help the user to identify them easily. * Have a meaningful title for the interface * The interface is simple and does not have unnecessary information. * Words are easy to understand and haven’t used system-oriented words. | |
| **Picture of interface 2** | **Brief description of interface 2** | |
|  | **Purpose**: This interface will be used to update the attendance  **Flow**: The user is required to add class id and date to search. Then the attendance will appear. After that, the user can Update  **Good practices**:   * The interface was designed according to the single responsibility principle. It means focuses on edit attendance tasks only. * Input fields have given real-world metaphors or icons which will help the user to identify them easily. * Only necessary things are included to maintain user-friendliness * Have a meaningful title for the interface * The interface is simple and does not have unnecessary information. * Words are easy to understand and haven’t used system-oriented words. | |
| **Picture of interface 3** | **Brief description of interface 3** | |
| Table  Description automatically generated | **Purpose**: This interface will be used to view attendance dates vice in every grade.  **Flow**: In here the user can filter attendance in two different ways. If the user needs to get a particular class attendance, then the user can filter attendance by giving class id and date.  Otherwise, the user can get all attendance on a particular date providing just a date.  **Good practices**:   * The interface was designed according to the single responsibility principle. It means focuses on edit attendance tasks only. * Input fields have given real-world metaphors or icons which will help the user to identify them easily. * Only necessary things are included to maintain user-friendliness * Have a meaningful title for the interface * The interface is simple and does not have unnecessary information. * Words are easy to understand and haven’t used system-oriented words. | |
| **Picture of interface 4** | **Brief description of interface 4** | |
| **Graphical user interface  Description automatically generated** | **Purpose**: This interface will be used to view attendance dates vice in every grade.  **Flow**: In here the user can filter attendance in two different ways. If the user needs to get a particular class attendance, then the user can filter attendance by giving class id and date.  Otherwise, the user can get all attendance on a particular date providing just a date.  **Good practices**:   * The interface was designed according to the single responsibility principle. It means focuses on login tasks only. * Only necessary things are included to maintain user-friendliness * Have a meaningful title for the interface * The interface is simple and does not have unnecessary information. * Words are easy to understand and haven’t used system-oriented words. | |
| **Sprint 2** | **Picture of interface 5** | **Brief description of interface 5** | |
| **Graphical user interface, application, table  Description automatically generated** | **Purpose**: This interface will be used to navigate different pages and get a brief idea about the system.  **Flow**: The user can navigate to pages by clicking buttons.  **Good practices**:   * The interface was designed according to the single responsibility principle. It means focuses only on view general details. * Only necessary things are included to maintain user-friendliness * Words are easy to understand and haven’t used system-oriented words. | |
| **Picture of interface 6** | **Brief description of interface 6** | |
| **Graphical user interface, application  Description automatically generated** | **Purpose**: This interface will be used to add marks.  **Flow**: The user is required to search the student by providing the student id. If there are any student details which is matching to the searched id, then the system will display details. After that, the user can add marks.  **Good practices**:   * All the buttons and labels are well named, have correct spellings, and start with a capital letter. * Only necessary things are included to maintain user-friendliness * Provide the users with the option to select values so that it minimizes the number of times that the user has to type things. * The interface focuses on one function which is to add student marks. | |
| **Picture of interface 7** | **Brief description of interface 7** | |
| Graphical user interface, application  Description automatically generated | **Purpose**: This interface will be used to edit marks.  **Flow**: The user is required to search the student by providing the student id. If there are any student details which is matching to the searched id, then the system will display details. After that, the user can edit marks.   * **Good practices**: All the buttons and labels are well named, have correct spellings and start with a capital letter. * Only necessary things are /included to maintain user-friendliness * The interface focuses on one function which is to add student marks. * Provide the users with the option to select values so that it minimizes the number of times that the user has to type things. | |
| **Picture of the interface\_8** | **Brief description of interface\_8** | |
| **Graphical user interface  Description automatically generated with medium confidence** | **Purpose**: This interface will be used to view marks.  **Flow**: The user is required to provide the student id to search. Then the system will display all marks of the student.  **Good practices**:   * All the buttons and labels are well named, have correct spellings, and start with a capital letter. * Only necessary things are included to maintain user-friendliness * Input fields have given real-world metaphors or icons which will help the user to identify them easily. | |
| **Picture of the interface\_9** | **Brief description of interface\_9** | |
| **Graphical user interface  Description automatically generated** | **Purpose** : This interface will be used to generate reports of attendance.  **Flow**: The user is required to select report type and select the date required. The system will display the report. The user can print the report by clicking the print button.   * **Good practices**: All the buttons and labels are well named, have correct spellings and start with a capital letter. * Only necessary things are included to maintain user-friendliness   Input fields have given real-world metaphors or icons which will help the user to identify them easily | |
|  | **Picture of the interface\_10** | **Brief description of interface\_10** | |
| **Graphical user interface, application  Description automatically generated** | **Purpose**: This interface will be used to generate reports of marks.  **Flow**: The user is required to select a report type, grade, and the date required. The system will display the report. The user can print or download the report by clicking the print button or download button.   * **Good practices**: All the buttons and labels are well named, have correct spellings and start with a capital letter. * Only necessary things are included to maintain user-friendliness * Input fields have given real-world metaphors or icons which will help the user to identify them easily | |
| **Registration No** | **IT19006994** | | |
| **Student Name** | **K.H.K.L. De Silva** | | |
| **Function(s)**  **Note**: Include the functions required to complete for **sprint 1** & **sprint 2** | **Sprint 1**   * Admin Add New Student (Student Management) * Admin view / Delete Student Details (Student Management) * Admin Update Student Details (Student Management) * Admin or Teacher Add New Notice (Notices Management) * Teacher or Admin View / Delete Notices (Notices Management) * Admin or Teacher Update Notices (Notices Management) * Student View Notices (Notices Management)   **Sprint 2**   * Admin or Teacher Generate Reports for Student Details (Student Management) * Admin or Teacher Search Student Details (Student Management) * Search Notices (Notices Management) | | |
| **Sprint 1** | **Picture of interface 1** | **Brief description of interface 1** | |
| Graphical user interface, application  Description automatically generated | **Purpose** : This interface will be used to add a new Student to the system.  **Flow**: Initially the Admin must navigate to the Student Interface and Add the Student Details. The form has three main parts as Student Info, Contact Info and Parent/Guardian Details. Should enter the Admission Number, First Name, Last Name, Section, Class, Gender and Date of Birth Under Student Info. Should enter Mobile Number, Email and Address under Contact Info. And should complete the Guardian Name, Relationship, Guardian Mobile Number and Email under the Parent/Guardian Section. It is mandatory to fill all these input fields. Users can add a profile picture or can ignore it. When all the details are added the admin can click on the Save button or can click the Cancel button to ignore the task. When the Cancel button is clicked Admin will be navigated to the Admin Dashboard. If the save operation is success the admin will be navigated to the All-Student View Interface.  **Good practices**:   * The interface was designed according to the Single Responsibility Principle. That means focuses on one specific task which is Adding a new student to the system. * Date of Birth, Gender, Section and Class can be selected using the dropdowns, so the user does not have to type them. * All the buttons and labels are with correct spellings, Start with capital letters and have meaningful names. * Buttons have different colours based on the function they perform. This will help the user to identify which function they perform and will minimize confusion when all buttons are in the same colour. * All the buttons and input fields have given real-world metaphors or icons which will help the user to identify them easily. (Save and cancel button) * The form is designed side by side to avoid it getting too long. * Only the necessary things are included in the interface and kept as simple as possible. * The form has separated into sections based on the information that has to be entered and it’s easy for the user to identify areas when entering data. * If there are any empty input fields an error box/ an alert box will be prompted mentioning the error. | |
| **Picture of interface 2** | **Brief description of interface 2** | |
| Graphical user interface, application  Description automatically generated | **Purpose** :  This interface will be used to view all student details through the system.  **Flow**:  In Admin Dashboard when the students' section is clicked from the side navbar, the user will be navigated to this interface. Admin can click on the view button under each student and can get a detailed view on an individual student or click delete to remove a student from the system.  **Good practices**:   * All the buttons and labels are with correct spellings, start with capital letters and have meaningful names. * Buttons have different colours based on the function they perform. This will help the user to identify which function they perform and will minimize confusion when all buttons are in the same colour. * All the buttons and student details such as email and phone number have given real-world metaphors or icons which will help the user to identify them easily. (Ex: mail, phone, bin, view icons) * Due to the card view, users can easily understand what to do and give an attractive look to the interface than the traditional table view. * Only the necessary things are included in the interface and kept as simple as possible. * When a user clicked the delete box an alert box will be prompted to confirm the process to avoid the mistaken loss of sensitive information. * When the delete function is executed successfully an alert box will be prompted mentioning that the Student Deleted Successfully. | |
| **Picture of interface 3** | **Brief description of interface 3** | |
| Graphical user interface, application  Description automatically generated | **Purpose** :  This interface will be used to update student details.  **Flow**:  When Admin clicked on the All-Student Interface and click on the View Button of the particular student, the Admin will be navigated to this page. Here the user can view all details of the selected student and can update data by clicking update. Instead, the Admin can click on the cancel button to undo the operation and will be navigated to the All-Student interface again.  **Good practices**:   * The interface was designed according to the Single Responsibility Principle. That means focuses on one specific task which is updating student details through the system. * Date of Birth, Gender, Section and Class can be selected using the dropdowns, so the user does not have to type. * All the buttons and labels are with correct spellings, start with capital letters and have meaningful names. * Buttons have different colours based on the function they perform. This will help the user to identify which function they perform and will minimize confusion when all buttons are in the same colour. * All the buttons and input fields have given real-world metaphors or icons which will help the user to identify them easily. * The form is designed side by side to avoid it getting too long. * Only the necessary things are included in the interface and kept as simple as possible. * The form has separated into sections based on the information that has to be entered and it’s easy for the user to identify areas when entering data. * If there are any empty input fields an error box/ an alert box will be prompted mentioning the error. | |
| **Picture of interface 4** | **Brief description of interface 4** | |
| **Graphical user interface, application  Description automatically generated** | **Purpose** :  This interface will be used to add a new notice to the system.  **Flow**: Admin / Teacher must click on the Add Notices tab from the side navbar the user will be navigated to this page. Here the user can add an image and enter the Notice Title and Description and click the publish button. All these fields are mandatory to be filled except the image. When the Notice is added successfully the user will be navigated to the View All Notices Interface and when the cancel button is clicked the operation will be terminated and will be navigated to the Admin/ Teacher Dashboard again.  **Good practices**:   * All the buttons and labels are with correct spellings, start with capital letters and have meaningful names. * All the buttons and input fields have given real-world metaphors or icons which will help the user to identify them easily. (Cancel and publish button) * Buttons have different colours based on the function they perform. This will help the user to identify which function they perform and will minimize confusion when all buttons are in the same colour. * The interface was designed according to the Single Responsibility Principle. That means focuses on one specific task which Adding a new Notice to the system. * If there are any empty input fields an error box/ an alert box will be prompted mentioning the error. * Only the necessary things are included in the interface and kept as simple as possible. | |
| **Sprint 2** | **Picture of the interface\_5** | **Brief description of interface\_5** | |
| Text  Description automatically generated with medium confidence | **Purpose** :  This interface will be used to view and Delete the Notices.  **Flow**:  When the Notices tab is clicked from the side navbar by Admin/Teacher they will be navigated to this interface. Here the user can view or delete the notices. When the view button is clicked the user will be navigated to the detailed view of the notice. When the delete button is clicked the notice will be deleted.  **Good practices**:   * All the buttons and labels are with correct spellings, start with capital letters and have meaningful names. * All the buttons have given real-world metaphors or icons which will help the user to identify them easily. * Buttons have different colours based on the function they perform. This will help the user to identify which function they perform and will minimize confusion when all buttons are in the same colour. * Only the necessary things are included in the interface and kept as simple as possible. * Due to the card view, users can easily understand what to do and give an attractive look to the interface than the traditional table view. * When a user clicked the delete box an alert box will be prompted to confirm the process to avoid the mistaken loss of sensitive information. * When the delete function is executed successfully an alert box will be prompted mentioning that the Student Deleted Successfully. | |
| **Picture of the interface\_6** | **Brief description of interface\_6** | |
| Graphical user interface, application  Description automatically generated | **Purpose** :  This interface will be used to update the notices.  **Flow**:  When the user clicked on the view button of a specific notice in the View All Notices Page, the user will be navigated to this page. Through this page, a detailed view of the notice can be obtained and can do the changes and update the details as well. When the Update Button is clicked data will be updated and the cancel button will undo the changes and return the user to the Dashboard.  **Good practices**:   * The interface was designed according to the Single Responsibility Principle. That means focuses on one specific task which is updating Notices through the system. * All the buttons and input fields have given real-world metaphors or icons which will help the user to identify them easily. * All the buttons and labels are with correct spellings, start with capital letters and have meaningful names. * Buttons have different colours based on the function they perform. This will help the user to identify which function they perform and will minimize confusion when all buttons are in the same colour. * Only the necessary things are included in the interface and kept as simple as possible. * If there are any empty input fields an error box/ an alert box will be prompted mentioning the error. * A detailed view of the notices can be obtained in this interface. | |
| **Picture of the interface\_7** | **Brief description of interface\_7** | |
| Graphical user interface  Description automatically generated with low confidence | **Purpose** :  This interface will be used to view the Notices for everyone including students and guests.  **Flow**:  When the Notices tab is clicked from the side navbar the user will be navigated to this page. When the View more button is clicked can get a detailed view of the Notices.  **Good practices**:   * The interface was designed according to the Single Responsibility Principle. That means focuses on one specific task which is viewing Notices through the system. * All the buttons and labels are with correct spellings, start with capital letters and have meaningful names. * Buttons have different colours based on the function they perform. This will help the user to identify which function they perform and will minimize confusion when all buttons are in the same colour. * Due to the card view, users can easily understand what to do and give an attractive look to the interface than the traditional table view. * Only the necessary things are included in the interface and kept as simple as possible. * All the buttons have given real-world metaphors or icons which will help the user to identify them easily. | |
| **Picture of the interface\_8** | **Brief description of interface\_8** | |
| **Graphical user interface, application  Description automatically generated**  \*Student Search and Notices Search function will be developed in sprint 2. But they are just a part of sprint 1 interfaces therefore have been included in those interfaces. | **Purpose** :  This interface will be used to generate student detail reports.  **Flow**:  First, the Admin/Teacher must click on the Report generation Tab from the side navbar and then should select student detail reports. Then the user will be navigated to this page.  **Good practices**:   * The interface was designed according to the Single Responsibility Principle. That means focuses on one specific task which is generating student detail reports through the system. * All the buttons and labels are with correct spellings, start with capital letters and have meaningful names. * Buttons have different colours based on the function they perform. This will help the user to identify which function they perform and will minimize confusion when all buttons are in the same colour. * Only the necessary things are included in the interface and kept as simple as possible. * All the buttons have given real-world metaphors or icons which will help the user to identify them easily. * Dropdowns are given to filter data instead of manually typing and search data. | |
| **Registration No** | **IT19111766** | | |
| **Student Name** | **U.L.V.M. Lekamalage** | | |
| **Function(s)**  **Note**: Include the functions required to complete for **sprint 1** & **sprint 2** | **Sprint 1**   * Admin Add New Teacher (Teacher Management) * Admin view / Delete Teacher Details (Teacher Management) * Admin Update Teacher Details (Teacher Management) * Admin Add New Subject (Subject Management) * Admin View / Delete Subject (Subject Management) * Teacher Add Lecture Slide to Subject (Subject Management) * Student View Subjects/Lecture Slides (Subject Management Subject)   **Sprint 2**   * Admin Generate Reports for Teacher Details (Teacher Management) * Admin Generate Reports for Subject Details (Subject Management) * Search Teacher Details (Teacher Management) * Search Subject Details (Subject Management) | | |
| **Sprint 1** | **Picture of interface 1** | **Brief description of interface 1** | |
| Graphical user interface, application, website  Description automatically generated | **Purpose** :  This interface will be used to add a new Subject to the system.  **Flow**:  The admin must navigate to the subject interface and add new subject details. The form includes subject id, allocated grade, subject name, and subject description. It is mandatory to fill all these input fields. When all the details are added the admin can click on the Add subject button or can click the Cancel button to ignore the task. When the Cancel button is clicked Admin will be navigated back to the Admin Dashboard. If the save operation is success the admin will be navigated to the All-subjects View Interface.  **Good practices**:   * The interface was designed to the Single Responsibility Principle. That means focuses on one specific task which is Adding a new subject to the system. * All the buttons and labels are with correct spellings, start with capital letters and have meaningful names, and the button has different colours to minimize confusion. * The form is designed to side by side to avoid it getting too long and include only the necessary things for kept as simple as possible * If there are any empty or error input fields an alert box will be prompted mentioning the error. | |
| **Picture of interface 2** | **Brief description of interface 2** | |
| Table  Description automatically generated | **Purpose** :  This interface will be used to view all subject details through the system.  **Flow**:  In Admin Dashboard when the Subject section is clicked from the side navbar, the user will be navigated to this interface. Admin can see all the subject details and if he/she click the delete button to remove a subject from the system. And when the admin clicks the generate report button, he/she will be direct to generate the report page.  **Good practices**:   * All the buttons and labels are with correct spellings, start with capital letters and have meaningful names, and the button has different colours to minimize confusion. * All the necessary things about subjects are included in the interface and kept as simple as possible. So, no need to navigate to another page for more details. * When a user clicked the delete box an alert box will be prompted to confirm the process to avoid the mistaken loss of information | |
| **Picture of interface 3** | **Brief description of interface 3** | |
| Graphical user interface, application  Description automatically generated | **Purpose** :  This interface will be used to add a new Teacher to the system.  **Flow**:  The Admin has to navigate to the Teacher interface and add new teacher details. The form includes teacher id, allocated grade, teacher name, and allocated subjects, and teacher description. It is mandatory to fill all these input fields. When all the details are added the Admin can click on the Add details button or can click the Cancel button to ignore the task. When the Cancel button is clicked Admin will be navigated back to the Admin Dashboard. If the save operation is success the admin will be navigated to the All teachers View Interface.  **Good practices**:   * The interface was designed to the Single Responsibility Principle. That means focuses on one specific task which is Adding a new teacher to the system. * All the buttons and labels are with correct spellings, Start with capital letters and have meaningful names, and also the button has different colours to minimize confusion. * The form is designed to side by side to avoid it getting too long and include only the necessary things for kept as simple as possible * If there are any empty or error input fields an alert box will be prompted mentioning the error. | |
| **Picture of interface 4** | **Brief description of interface 4** | |
| **Table  Description automatically generated** | **Purpose** :  This interface will be used to view all teachers details through the system  **Flow**:  In Admin Dashboard when the teacher section is clicked from the side navbar, the user will be navigated to this interface. Admin can see all the teacher details and if he/she click the delete button to remove a teacher from the system. Or admin clicks the edit button he will be direct to edit the teacher details page. And when the admin clicks the generate report button, he/she will be direct to generate the report page.  **Good practices**:   * All the buttons and labels are with correct spellings, start with capital letters and have meaningful names, and the button has different colours to minimize confusion. * All the necessary things about teachers are included in the interface and kept as simple as possible. So, no need to navigate to another page for more details. * When a user clicked the delete box an alert box will be prompted to confirm the process to avoid the mistaken loss of information | |
|  | **Picture of the interface\_5** | **Brief description of interface\_5** | |
| Graphical user interface, application  Description automatically generated | **Purpose** :  This interface will be used to update teacher details  **Flow**:  When Admin clicked on the All-teacher Interface and clicks on the edit button of the teacher, the admin will be navigated to this page. Here the user can view all details of the selected teacher and can update data by clicking update. Instead, the Admin can click on the cancel button to undo the operation and will be navigated to the All-teacher interface again. If he/she clicks the delete button teacher will be deleted from the system.  **Good practices**:   * The interface was designed to the Single Responsibility Principle. That means focuses on one specific task which is Adding a new teacher to the system. * All the buttons and labels are with correct spellings, start with capital letters and have meaningful names, and the button has different colours to minimize confusion. * The form is designed to side by side to avoid it getting too long and include only the necessary things for kept as simple as possible * If there are any empty or error input fields an alert box will be prompted mentioning the error. * When a user clicked the delete box an alert box will be prompted to confirm the process to avoid the mistaken loss of information | |
| **Picture of the interface\_6** | **Brief description of interface\_6** | |
| Graphical user interface, application  Description automatically generated | **Purpose** :  Landing page for the teacher when he/she login. And teachers can do certain tasks using this dashboard.  **Flow**:  When the teacher logs in to the system using their credentials, they will be direct to this dashboard. In here they can see allocated students for him/her. And using the side navbar he/she can direct to the student, subject, attendance, and notices pages.  **Good practices**:   * The interface was designed uniquely for teachers. So, teachers can do their tasks easily. * All the buttons and labels are with correct spellings, start with capital letters and have meaningful names, and the button has different colours to minimize confusion. * And in the dashboard, they can see their details like name, profile image. | |
| **Picture of the interface\_7** | **Brief description of interface\_7** | |
| Graphical user interface  Description automatically generated | **Purpose** :  this interface will be used to see all allocated subjects to teachers.  **Flow**:  When after teachers log in, they can go to the subject details page here they can see the allocated subject to teach. Then they can click on the subject and go direct to the subject page.  **Good practices**:   * The interface was designed according to the Single Responsibility Principle. That means focuses on one specific task which is to view allocated subject details through the system. * All the buttons and labels are with correct spellings, start with capital letters and have meaningful names, and the button has different colours to minimize confusion. | |
| **Picture of the interface\_8** | **Brief description of interface\_8** | |
| **Graphical user interface  Description automatically generated** | **Purpose** :  this interface will be used to see inside of an allocated subject to teachers.  **Flow**:  When after the teacher goes to the subject details page here, they can see the allocated subject to teach. Then they can click on the subject and go direct to the subject page. In here they can see previously added lecture slides lessons, etc... and, they can add new lecture slides and delete unnecessary slides.  **Good practices**:   * The interface was designed according to the Single Responsibility Principle. That means focuses on one specific task which is to view single subject details through the system. * All the buttons and labels are with correct spellings, start with capital letters and have meaningful names, and the button has different colours to minimize confusion. | |
| **Picture of the interface\_9** | **Brief description of interface\_9** | |
| **Graphical user interface, application  Description automatically generated** | **Purpose** :  this interface will be used to add lecture slides for one subject.  **Flow**:  When the teacher clicked on the subject Interface and click on add lecture slide Button, the teacher will be navigated to this page. Here the user can add a new lecture slide. Instead, the teacher can click on the cancel button to undo the operation and will be navigated to the All allocated subject interface again.    **Good practices**:   * The interface was designed to the Single Responsibility Principle. That means focuses on one specific task which is Adding a new lecture slide to the system. * All the buttons and labels are with correct spellings, start with capital letters and have meaningful names, and the button has different colours to minimize confusion. * The form is designed to avoid it getting too long and include only the necessary things for kept as simple as possible * If there are any empty or error input fields an alert box will be prompted mentioning the error. | |
| **Picture of the interface\_10** | **Brief description of interface\_10** | |
| **Diagram  Description automatically generated** | **Purpose** :  this interface will be used to see all allocated subjects to the teacher.  **Flow**:  When after teachers log in, they can go to the subject details page here te can see the allocated subject to teach. Then they can click on the subject and go direct to the subject page.  **Good practices**:   * The interface was designed according to the Single Responsibility Principle. That means focuses on one specific task which is to view allocated subject details through the system. * All the buttons and labels are with correct spellings, start with capital letters and have meaningful names, and the button has different colours to minimize confusion. | |
| **Picture of the interface\_11** | **Brief description of interface\_11** | |
| **Graphical user interface, application  Description automatically generated** | **Purpose** :  this interface will be used to see inside of an allocated subject to the student.  **Flow**:  When after the student goes to the subject details page here, they can see the allocated subject to teach. Then they can click on the subject and go direct to the subject page. In here they can see previously added lecture slides lessons, etc... and, they can download lecture slide  **Good practices**:   * The interface was designed according to the Single Responsibility Principle. That means focuses on one specific task which is to view single subject details through the system. * All the buttons and labels are with correct spellings, start with capital letters and have meaningful names, and the button has different colours to minimize confusion. | |
| **Picture of the interface\_12** | **Brief description of interface\_12** | |
| **Sprint 2** | **Table  Description automatically generated with medium confidence** | **Purpose** :  This interface will be used to generate teacher detail reports.  **Flow**:  First, the Admin must click on the Report generation button from the view all teachers and then should select generate report button. Then the user will be navigated to this page.    **Good practices**:   * All the buttons and labels are with correct spellings, start with capital letters and have meaningful names, and the button has different colours to minimize confusion. * Only the necessary things are included in the interface and kept as simple as possible. * Dropdowns are given to filter data instead of manually typing and search data. | |
| **Picture of the interface\_13** | **Brief description of interface\_13** | |
| **Graphical user interface  Description automatically generated** | **Purpose** :  This interface will be used to generate teacher detail reports.  **Flow**:  First, the Admin must click on the Report generation button from the view all subject and then should select generate report button. Then the user will be navigated to this page.  **Good practices**:   * All the buttons and labels are with correct spellings, start with capital letters and have meaningful names, and the button has different colours to minimize confusion. * Only the necessary things are included in the interface and kept as simple as possible. * Dropdowns are given to filter data instead of manually typing and search data. | |