**1.1. Purpose of the System**

This system was designed with the aim of providing easy, credible for teachers who want to take attendance, users can display their classes, and participation percentages, which class their missed and detailed view of classes. It makes the whole process an easy affair without too much hassle as follow a few easy steps.

**3. Proposed System**

Our online attendance tracking app allows users to see the their class details, system focused on improving teacher and student relations. The admin in the system can accept or reject the request that sent from student affair. Finally, the system allows students to enroll or drop their classes as given time. The online attendance application prevents the waste of unnecessary paper and the time lost in the virtual environment.

**Reliability:** The system must be running 100% of the time when teacher taking attendance.

**Implementation:** The system will be implemented on Angular CLI it is a typescript framework. In addition, firebase will be used as the database. User Interface should be responsive, bootstrap will be use so it will be smart phone friendly.

**Use Case 1**

|  |
| --- |
| **Use case name:** Listing Assigned Classes |
| **Participant actor instances:** İnitiated by Teacher |
| **Flow of Events:**   1. Opening the application page. 2. Application responded by showing login screen. 3. Application asks teachers email and password. 4. Teacher enters the related info's to the text-boxes. 5. Clicks to ‘login’ button to submitting. 6. System checks for the related data from database. 7. System redirecting teacher to main page and loads the classes data in to main page. |
| **Entry Condition:** User opens the application page. Clicks the login button on login page. User has not to be registered. |
| **Exceptional Cases:**  **Invalid Character:** If the App detects invalid characters into the text-boxes or not filled any text-box, it warns the User about the entering invalid characters and wants to retry from the User by displaying alert. |

**Use Case 2**

|  |
| --- |
| **Use case name:** Listing Assigned Classes Students |
| **Participant actor instances:** İnitiated by Teacher |
| **Flow of Events:**   1. Opening the application page. 2. Application responded by showing login screen. 3. Application asks teachers email and password. 4. Teacher enters the related info's to the text-boxes. 5. Clicks to ‘login’ button to submitting. 6. System checks for the related data from database. 7. System redirecting teacher to main page and loads the classes data in to main page. 8. Teacher clicks the ‘Details’ button of required class and system takes the ID of the class, redirects the detail page and load the related data of class. 9. For finding classes students system uses ID for related tables for filtering students and lists. |
| **Entry Condition:** If required class has no enrolled student NULL data will return. |
| **Exceptional Cases:**  **User Session:** If the proces cant be done in the required time, session will be expired and user will be redirected to the login page. |

**Use Case 3**

|  |
| --- |
| **Use case name:** Oppening Attandance |
| **Participant actor instances:** İnitiated by Teacher |
| **Flow of Events:**   1. Opening the application page. 2. Application responded by showing login screen. 3. Application asks teachers email and password. 4. Teacher enters the related info's to the text-boxes. 5. Clicks to ‘login’ button to submitting. 6. System checks for the related data from database. 7. System redirecting teacher to main page and loads the classes data in to main page. 8. Teacher clicks the ‘Details’ button of required class and system takes the ID of the class, redirects the detail page and load the related data of class. 9. Teacher clicks ‘open attendance’ button and system redirects the teacher into attendance creating page and the fields that needs to be filled. |
| **Quality Requirements:** If teacher leaves required fields empty, system displays a warning message, like "This area cannot be empty.” |
| **Exceptional Cases:**  **User Session:** If the process cant be done in the required time, session will be expired and user will be redirected to the login page. |

**Use Case 4**

|  |
| --- |
| **Use case name:** Taking Attandance |
| **Participant actor instances:** İnitiated by Teacher |
| **Flow of Events:**   1. Opening the application page. 2. Application responded by showing login screen. 3. Application asks teachers email and password. 4. Teacher enters the related info's to the text-boxes. 5. Clicks to ‘login’ button to submitting. 6. System checks for the related data from database. 7. System redirecting teacher to main page and loads the classes data in to main page. 8. Teacher clicks the ‘Details’ button of required class and system takes the ID of the class, redirects the detail page and load the related data of class. 9. Teacher clicks ‘opened attendance’ button and system redirects the teacher into attendance list page and loads the student list with it. 10. Teacher selects options(present, absent, excused) for each student and clicks save. 11. System takes the submitted data, saves it into attendance table, each tuple with related student. |
| **Quality Requirements:** If teacher leaves required fields empty, system displays a warning message, like "This area cannot be empty.” |
| **Exceptional Cases:**  **User Session:** If the proces cant be done in the required time, session will be expired and user will be redirected to the login page. |

**Use Case 5**

|  |
| --- |
| **Use case name:** Sending Request to Student Affair |
| **Participant actor instances:** İnitiated by Teacher |
| **Flow of Events:**   1. Opening the application page. 2. Application responded by showing login screen. 3. Application asks teachers email and password. 4. Teacher enters the related info's to the text-boxes. 5. Clicks to ‘login’ button to submitting. 6. System checks for the related data from database. 7. System redirecting teacher to main page and loads the classes data in to main page. 8. Teacher clicks the ‘open courses’ button, system redirects to the page and loads the related data. 9. Teacher clicks the ‘send reques’ for desired course. 10. System saves the request into ‘requests’ table. |
| **Exceptional Cases:**  **User Session:** If the proces cant be done in the required time, session will be expired and user will be redirected to the login page. |