**Table of Contents**

[1. Introduction 1](#_Toc519808796)

[**1.1 Purpose** 1](#_Toc519808797)

[**1.2 Existing System** 2](#_Toc519808798)

[**1.3 Problem Statement** 2](#_Toc519808799)

[**1.4 Proposed Solution** 3](#_Toc519808801)

[2. Problem / System Analysis 6](#_Toc519808802)

[**2.1 Features:** 6](#_Toc519808803)

[**2.2 Non-Functional Requirements** 8](#_Toc519808804)

[**2.3 Use Case** 9](#_Toc519808805)

[3. System Design 28](#_Toc519808806)

[**3.1 Class Diagram** 28](#_Toc519808807)

[**3.2 Sequence Diagram** 30](#_Toc519808808)

[**3.3 Data Base Schema** 34](#_Toc519808809)

[4. Implementation 36](#_Toc519808810)

[**4.1 Tools and Technologies** 36](#_Toc519808811)

[**4.2 Code** 40](#_Toc519808812)

[5. System Testing 66](#_Toc519808813)

[***5.1 Testing Objectives*** 66](#_Toc519808814)

[***5.2 Testing Cases*** 67](#_Toc519808815)

[6 Conclusion 77](#_Toc519808816)

[6.1 Limitations of the System 77](#_Toc519808817)

[A2.References 79](#_Toc519808818)

**CHAPTER 1**

**INTRODUCTION**

1. **Introduction**

A school teacher has to handle a lot of things like answer the emails, make notes, schedule meetings with the parents, answers parents question and top of it manage the class and engage students. All these things become hectic and juggling act for the teacher. Not only that we also want to provide effective communication between teacher and parents, parents want to monitor and keep on updated about the activities that are going in the class room. So to cover all these things we want to create a plate form that provides all these features and some other features on one single plate form. Teacher can use all the tools of plate form to communicate effectively with the parents, manage and coordinate the classes and meetings. We want to create this App on higher scale that’s why we are making ‘Teacher Mate’, an app that provides all the above features on one single Platform.

### Purpose

Parents can keep track of student's attendance records and upcoming events in their child's school using this app. It’s platform for communicating all class activities with parents, as teacher has a lot of responsibilities on her, so to manage the class room and to keep in touch with parents we need ‘Teacher Mate’, it’s easy to use like an social media app but in terms of features it more like academic application, parents can see their students profiles, grading graphs are there to show their child progress, more over you can see the behavior of your child through the student portfolio that’s a unique feature which was missing in previous applications. Alerts for meetings and conferences all are handled through this app, also you can see your student activities through its time line feature where all the class activities are posted in form of pictures and videos. So like these kind of features parents are more aware of their child progress and learning.

### 1.2 Existing System

Existing system that is running in Beacon House School Lahore, has been implemented on low level, it’s still running on Lahore Branch. Existing System has niche features in it and that’s why we need to advance it. Features in Existing System are as teacher can only post Quizzes and assignments and its notifications are sent to the parents. Another feature of existing system isteacher can share Student Grades to notify their parents about their result.

### 1.3 Problem Statement

These are the main features, so for school teacher it’s not enough at all, because school teachers have to do a lot of task like parents can see their students profiles, grading graphs are there to show their child progress, more over you can see the behavior of your child through the student portfolio that’s a unique feature which was missing in previous applications. Because existing system has niche features which includes just grading students, teacher has to keep them motivated by doing activities in class and most importantly to let know the parents about their child.

|  |  |
| --- | --- |
| Problem | These are the main features, so for school teacher it’s not enough at all, because school teachers have to do a lot of task other than just grading students, teacher has to keep them motivated by doing activities in class and most importantly to let know the parents about their child. |
| Affects | Involves extra time and space for items |
| Impact | Existing system that is running in Beacon House School Lahore, has been implemented on low level, it’s still running on Lahore Branch. Existing System has niche features in it |
| Solution | ‘Teacher Mate’, it’s easy to use like a social media app but in terms of features it more like academic application, parents can see their students profiles, grading graphs are there to show their child progress, more over you can see the behavior of your child through the student portfolio that’s a unique feature which was missing in previous applications. Alerts for meetings and conferences all are handled through this app, also you can see your student activities through its time line feature where all the class activities are posted in form of pictures and videos. So like these kind of features parents are more aware of their child progress and learning. |

**Table 1.1 Problem Statement Table**

### 1.4 Proposed Solution

After Explaining all the problems related to existing system now we are progressing towards our new proposed solution that is ‘Teacher Mate’ so what is ‘Teacher Mate’? It’s platform for communicating all class activities with parents, as discussed above teacher has a lot of responsibilities on her, so to manage the class room and to keep in touch with parents we need ‘Teacher Mate’, it’s easy to use like an social media app but in terms of features it more like academic application, parents can see their students profiles, grading graphs are there to show their child progress, more over you can see the behavior of your child through the student portfolio that’s a unique feature which was missing in previous applications. Alerts for meetings and conferences all are handled through this app, also you can see your student activities through its time line feature where all the class activities are posted in form of pictures and videos. So like these kind of features parents are more aware of their child progress and learning.

**CHAPTER 2**

**SYSTEM ANALYSIS**

## Problem / System Analysis

Analysis emphasizes an investigation of the problem and requirements, rather than a solution. For example, if a new online system is desired, how will it be used? What are its functions? An investigation of a domain that results in models describing its static and dynamic characteristics. It emphasizes questions of "what," rather than "how."

"Analysis" is a broad term, best qualified, as in requirements analysis (an investigation of the requirements) or object-oriented analysis (an investigation of the domain objects). So we can say that system analysis is a task that bridges the gap between system level requirements and software design.

**2.1 Requirements Elicitation:**

This phase involves a high level of client input. For gathering correct requirements communication with the client is the essential part in case when the project is being developed by an organization. Following are the techniques used which were necessary for the requirement elicitation:

* Interviewing and Questioning
* Brain storming and idea reduction

**2.2 Requirements Analysis**

This is the process by which elicited requirements are evaluated for consistency. Requirements are triaged so placed in 3 separate categories:

* Must-have
* Should-have
* Nice-to-have

Through this process the most relevant, mission critical requirements are emphasized and developed first. This ensures the final product satisfies all of the important requirements, and does so in the most time and cost efficient manner possible. Other “should-have” and “nice to- have” requirements can be relegated to future phases, should they be necessary.

**2.3 Requirements Specification:**

This is the process by which requirements and their analysis are committed to some formal media. For this project, we have adopted a use case driven approach to organize and systematize this software development process and produce a document that will define system’s requirement.

* 1. **Functional Requirements**

Functional requirements of a system define the internal workings of the software. The functional requirements of a apraxia speech therapy application are as follows:

***2.4.1* *Class Room Environment (like Google Class Room):***

Teacher can create classroom environment like google classroom and send the invitation code to all teachers, as invitation is only send from teacher so its secure and private. Here teacher can add all the class tasks, assignments, pending home works and deadlines etc.

***2.4.2* *Class Activities Sharing on Class Room TimeLine:***

We also want to provide the unique feature that distinct our app from other apps is that we are adding the feature like sharing class activities by uploading photos or video on class room time line to share with all the parents who are member of this class room. Class Room time line is like Facebook time line were each shared posts are view by all the members, just like this we want to provide the parents monitoring of class room activities to keep them updated about class room.

***2.4.3 Maintaining Student Behavior:***

Each Student has its own profile which is connected to parent profile, so we are giving the feature of maintaining student behavior by adding awards to student’s profile when a student performs good in class room, there are many types of awards, each award resembling the quality of student so in this way behavior of student is maintained and shared with parents.

***2.4.4 Sharing Students Portfolio:***

Teachers creates the student portfolio by uploading the pictures of videos of particular student, in this student record of doing certain activity is maintained and shared to their parents.

***2.4.5 Private Messaging Between Parent and Teacher:***

Parents can contact directly with teachers through messaging on our app, all these messaging feature is completely secure and private, providing this feature allows the parents to know when the teacher is active on app so that parents can ask the questions.

***2.4.6 Student Study Planning:***

This feature is for the student’s our app creates the study plan for each student. There will be weekly study plan for each student, once a student done with the task of study plan he/she will go through the quiz based on that task, if a student gets reasonable marks then another task comes up otherwise our study plan will show students their mistakes and retake the quiz again

***2.4.6 Online Quizzes:***

Teacher can also create online quiz for their students, its purpose is to make students analyze that what are their mistakes so that they won’t make mistake on paper quiz at school.

***2.4.7 Parents Teachers Conferences Outcome and Results Sharing:***

If parents somehow cannot come to the meetings or conferences then teacher can add the results and outcomes of conferences on time line to share with the members, so that parents are one page with the teacher and if they have any issue regarding the results then they can ask parents and resolve the issue.

* 1. **Non Functional Requirements**

These are quality requirements that stipulate the performance levels required of the system for various kinds of activities. Numerical lower and upper limits set conditions on the response times, access times etc of the system. Sometimes, tradeoffs are necessary among various non- functional requirements. Some Non-Functional Requirements are as follow:

### *Language*

The system supports English language only.

### *Efficiency*

Teacher Mate application is concerned with the minimum processing time as it will as the Optimal use of system resource in designing the proposed system. The efficiency factor has been taken into account.

***2.5.3 Flexibility***

This application is capable of making changes and enhancement in accordance with the future need. It has been designed to meet future requirements of the user. It is flexible enough to make necessary adjustment and readjustment instantly

***2.5.4 Performance***

Response time of the Teacher Mate application will be 1 minute. Response time refers to the waiting time while the system accesses, queries and retrieves the information from the connecting device (mobile phone).

The proposed application will reduce time and efforts required for retrieving information from system. And will have the capability to achieve the various queries efficiently and in time.

***2.5.5 User Friendly Interface***

The interface of the system will be user friendly in order to allow users an easy interaction with the system. User will communicate with the system easily and simply. No specific staff will be required to operate the system. Because many of the users do not possess high computer knowledge they usually are at basic level. So interface must be simple. Any notification or error messages generated by application shall be clear, succinct, polite and free of jargon.

***2.5.6 Reliability***

Teacher Mate application shall be available at the time of need. This application shall always provide real time information about students consulting with teachers. Teacher Mate application shall be robust enough to have a high degree of fault tolerance. This application shall be able enough to recover from hardware failures, power failures and rollback data to their most recent valid state.

* + 1. ***Feasibility Study***

The main purpose of feasibility study is to consider the proposed system with a view to decide whether the development of the proposed system is feasible for the organization in the underlying circumstances. It is considered that whether the proposed system will meet all the requirements of the user with lesser cost and with the improved efficiency than the existing system if exists. The feasibility of a system is considered from the following three aspects.

* Financial feasibility
* Technical feasibility
* Operational feasibility
  + - 1. **Financial Feasibility**

Since the system is being developed as a BSSE degree requirement project. So no development expenditure will be charged for the whole development and the system will be free to use for the users. The tools being used for the development are android studio, blender. Software and hardware is used. So project is very much feasible economically and financially.

* + - 1. **Technical Feasibility**

Full technical supervision will be available for the system. On the other hand system does not have any extra ordinary technical equipment. So project is technically feasible too.

* + - 1. **Operation Feasibility**

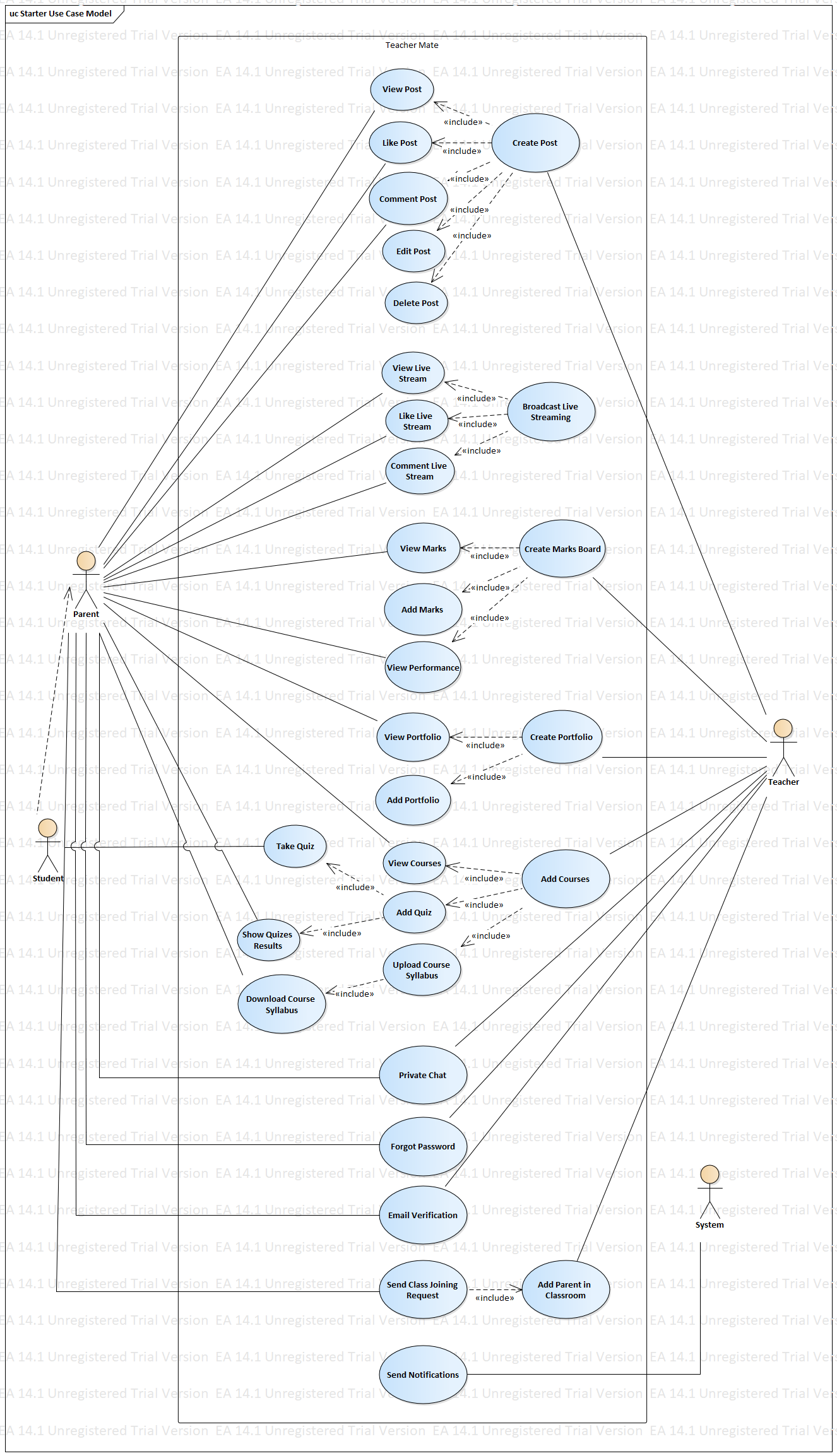
The proposed system will be operationally feasible became it will be efficient, accurate and reliable and will be provide more facilities than other existing systems. The proposed system will be user friendly and easy to understand, so average person would be able to understand and use this system easily.

### 2.3 Use Case

A use case is a sequence of actions that provide a measurable value to an actor. Another way to look at it is a use case describes a way in which a real-world actor interacts with the system. In a system use case you include high-level implementation decisions. System use cases can be written in both an informal manner and a formal manner.

Use case Diagrams are created to visualize the relationship between actors and use cases. A use case diagram is also used to capture the system functionality as seen by the user.Use cases are modeled using unified modeling language and are represented by ovals containing the names of the use case. Actors are represented using lines with the name of the actor written below the line. To represent an actor's participation in a system, a line is drawn between the actor and the use case. Boxes around the use case represent the system boundary. There are three basic elements that make up a use case:

* + - Actors: Actors are the type of users that interact with the system.
    - System: Use cases capture functional requirements that specify the intended behavior of the system.
    - Goals: Use cases are typically initiated by a user to fulfill goals describing the activities and variants involved in attaining the goal.



**Figure 2.3.1 Use Case Diagram**

#### 2.3.2 Fully Dressed Use Cases

* **Use case:** Create Account

**Actor:** Teacher, Parents

**Type:** Primary

**Description:** User when launch the app then first thing was to create an account by clicking if you are parent or teacher

* **Use case:** Create Classroom

**Actor:** Teacher

**Type:** Primary

**Description:** Teacher after successfully login, he/she can create the classroom by clicking on the create button then providing the classroom name and section classroom can be created

* **Use case:** Create Post

**Actor:** Teacher

**Type:** Primary

**Description:** Teacher starts the application and clicks on create post button to create the post, teacher can create post by adding an image or file in the post

* **Use case:** View Post

**Actor:** Parent, Teacher

**Type:** Primary

**Description:** Teacher or parent starts the application and after selecting the classroom the post can be seen on the timeline

* **Use case:** Like Post

**Actor:** Parent, Teacher

**Type:** Primary

**Description:** Teacher or parent starts the application and after selecting the classroom the post can be seen on the timeline, teacher can like his/her own post and parent can also like the post

* **Use case:** Comment Post

**Actor:** Parent, Teacher

**Type:** Primary

**Description:** Teacher or parent starts the application and after selecting the classroom the post can be seen on the timeline, teacher can comment his/her own post and parent can also comment on the post

* **Use case:** Add Student

**Actor:** Teacher, System

**Type:** Primary

**Description:** Teacher after selecting the classroom, clicks on the add student option from home menu button. Add Student page opens, teacher fill out the required fields and submit button. System now adds the student into the database

* **Use case:** Edit Post

**Actor:** Teacher

**Type:** Primary

**Description:** Teacher can edit the post by clicking on the edit post, then after updating the post the updated post can be seen on timeline

* **Use case:** Delete Post

**Actor:** Teacher

**Type:** Primary

**Description:** Teacher can delete the post by clicking on delete post and after that the post will be deleted and cannot be seen to parent and teacher

* **Use case:** Broadcast Live Streaming

**Actor:** Teacher

**Type:** Primary

**Description:** Teacher after successfully login and selecting a classroom, he/she can create live streaming by clicking on live broadcast then mobile camera starts rolling and the live broadcast starts streaming

* **Use case:** View Live Stream

**Actor:** Parent, Teacher

**Type:** Primary

**Description:** Teacher or parent after selecting the classroom, can view the live stream by clicking on live broadcast then after selecting the broadcasted video

* **Use case:** Like Live Stream

**Actor:** Parent, Teacher

**Type:** Primary

**Description:** Teacher or parent after selecting the classroom, can view the live stream by clicking on live broadcast then after selecting the broadcasted video parents and teacher can like the broadcasted video

* **Use case:** Comment Live Stream

**Actor:** Parent, Teacher

**Type:** Primary

**Description:** Teacher or parent after selecting the classroom, can view the live stream by clicking on live broadcast then after selecting the broadcasted video parents and teacher can comment on the broadcasted video

* **Use case:** Create Marks Board

**Actor:** Teacher

**Type:** Primary

**Description:** Teacher after selecting the classroom, can create the marks board by clicking on the home menu, then he/she can create the marks board by adding the total marks and weightage of each test on the marks board

* **Use case:** Add Marks Board

**Actor:** Teacher

**Type:** Primary

**Description:** Teacher after selecting the classroom, can add the student marks by clicking on home button menu option

* **Use case:** View Marks Board

**Actor:** Teacher, Parent

**Type:** Primary

**Description:** Teacher or parent after selecting the classroom, can view the student marks by clicking on home button menu option, for parents they can only see their child marks

* **Use case:** Create Student Portfolio

**Actor:** Teacher

**Type:** Primary

**Description:** Teacher after selecting the classroom, can create the student portfolio by clicking on the home menu option, then after providing the required the info portfolio can be created

* **Use case:** Add Student Portfolio

**Actor:** Teacher

**Type:** Primary

**Description:** Teacher after selecting the classroom, can add the student portfolio by clicking on the home menu option, then after selecting the child and clicking on the portfolio button teacher can provide the required the info to add portfolio

* **Use case:** View Student Portfolio

**Actor:** Teacher, Parent

**Type:** Primary

**Description:** Teacher or parents after selecting the classroom, can view the portfolio of student but parents can only view the portfolio of their own child

* **Use case:** Add Courses

**Actor:** Teacher

**Type:** Primary

**Description:** Teacher after selecting the classroom, can add courses by clicking on add courses option on home menu, then after adding the required info courses can be added

* **Use case:** View Courses

**Actor:** Teacher, Parents

**Type:** Primary

**Description:** Teacher or Parents after selecting the classroom, can view courses by clicking on the home menu option

* **Use case:** Add Quiz

**Actor:** Teacher

**Type:** Primary

**Description:** Teacher after selecting the classroom, click on the courses option from home menu then select the course topic and add quiz by providing the required fields data

* **Use case:** Take Quiz

**Actor:** Parents, Students

**Type:** Primary

**Description:** Parents after selecting the classroom then click on the courses from home menu and after selecting the particular course topic, quiz against the topic is displayed now student can take quiz

* **Use case:** Show Quizzes Results

**Actor:** Parents

**Type:** Primary

**Description:** Parents after selecting the classroom then click on the quizzes results option. List of quizzes results are displayed, parent can click on any particular result to see the detail results

* **Use case:** Upload Course Syllabus

**Actor:** Teacher, System

**Type:** Primary

**Description:** Teacher after selecting the classroom then click on the add courses. Teacher can fill out the required fields and upload any file. System then uploads the file on the cloud database

* **Use case:** Download Course Syllabus

**Actor:** Teacher, System, Parents

**Type:** Primary

**Description:** Teacher or parents after selecting the classroom then click on courses option from home menu, here all the courses will be displayed and parent or teacher can download the course file by clicking on the cloud icon

* **Use case:** Private Chat

**Actor:** Teacher, Parents

**Type:** Primary

**Description:** Teacher or parents after selecting the classroom then click on chat icon, then list of parents are displayed for teacher and for parent only teacher will be shown in the list. On clicking the particular parent or teacher chat page open for chatting

* **Use case:** Forgot Password

**Actor:** Teacher, Parents, System

**Type:** Primary

**Description:** parent or teacher can reset their password by clicking on the forgot password on signup page. System sent the Reset password email will be the particular user

* **Use case:** Email Verification

**Actor:** Teacher, Parents, System

**Type:** Primary

**Description:** After signup, email for verification of the user account will be sent to the user by the system

* **Use case:** Send Class Joining Request

**Actor:** Parents, System

**Type:** Primary

**Description:** parent after login can send the request to any teacher for request him/her to add the parent in the classroom

* **Use case:** Add Parent in Classroom

**Actor:** Parents, System

**Type:** Primary

**Description:** parent after login can send the request to any teacher for request him/her to add the parent in the classroom

* **Use case:** Send Notifications

**Actor:** System, Teacher

**Type:** Primary

**Description:** Teacher after successful login and selecting a classroom, system will send notifications to other parents in classroom when teacher add a post

**Iteration Plan:**

**Iteration Plan 1 (ITP -1):**

* **Use case: Create Account**
* **Scope:** Teacher Mate
* **Level:** User goal
* **Primary Actors:** Parent and Teacher
* **Stakeholders and Interests:** User creates account when the app launches
* **Preconditions:**

1. Application is running
2. Internet is running
3. Cloud database is integrated with the app

* **Success Guarantee (or Post-conditions):** New User account is created and the desired info is saved on cloud database
* **Main success Scenario:**

|  |  |
| --- | --- |
| **Actor Action** | **System Responsibilities** |
| 1. User starts the application |  |
| 1. User selects the option to choose whether he/she is Teacher or Parent |  |
|  | 1. System opens the signup page if user is not login before |
| 1. User fill the required fields then press signup button |  |
|  | 1. System then verify that user credentials are unique and then save the information of user and opens the home page. 2. If user with same email already exits then System will pop up the alert that email already exists |

* **Extension:**
* At any time, application will fails
* Restart the application
* Internet connection might get lost
* At any time, application crashes
* Restart the application
* Repeat the process
* **Specific Requirements:** Internet must be running and application will be running
* **Technology and Data Variations List:** Good speed internet will be required
* **Frequency of Occurrence:** Once
* **Use case: Create Classroom**
* **Scope:** Teacher Mate
* **Level:** User goal
* **Primary Actors:** Teacher

**Stakeholders and Interests:** Teacher after successfully login, he/she can create the classroom by clicking on the create button then providing the classroom name and section classroom can be created

* **Preconditions:**

1. Application is running
2. Internet is running
3. Cloud database is integrated with the app
4. Teacher is login

* **Success Guarantee (or Post-conditions):** New User account is created and the desired info is saved on cloud database
* **Main success Scenario:**

|  |  |
| --- | --- |
| **Actor Action** | **System Responsibilities** |
| 1. Teacher starts the application |  |
| 1. User selects the option on header to create classroom |  |
|  | 1. System open a pop up to fill out the required info for classroom |
| 1. User fill the required fields then press create |  |
|  | 1. System then verify that classroom name does not exists already 2. If classroom name is unique then System will create the classroom and saves it in database |

* **Extension:**
* At any time, application will fails
* Restart the application
* Internet connection might get lost
* At any time, application crashes
* Restart the application
* Repeat the process
* **Specific Requirements:** Internet must be running and application will be running
* **Technology and Data Variations List:** Good speed internet will be required
* **Frequency of Occurrence:** Once
* **Use case: Forgot Password**
* **Scope:** Teacher Mate
* **Level:** User goal
* **Primary Actors:** Teacher, Parent

**Stakeholders and Interests:** Teacher or Parent can reset password by clicking on log out button, then click on the user option and click on forget password. Page open which take email as input and reset password email will be sent to particular user email account

* **Preconditions:**

1. Application is running
2. Internet is running
3. Cloud database is integrated with the app
4. User is login

* **Success Guarantee (or Post-conditions):** New User account is created and the desired info is saved on cloud database
* **Main success Scenario:**

|  |  |
| --- | --- |
| **Actor Action** | **System Responsibilities** |
| 1. Teacher or parent starts the application |  |
| 1. Teacher or Parent clicks on log out button |  |
|  | 1. System opens the start page of app |
| 1. Teacher or Parent will chose the user option whether he/she is Parent or Teacher |  |
|  | 1. System will open the Login Page |
| 1. User will select the forgot password |  |
|  | 1. System opens the Forgot Password page with input field of email |
| 1. User now fills the input field and press submit button |  |
|  | 1. System will send the reset password email to the user email account to reset the password |

* **Extension:**
* At any time, application will fails
* Restart the application
* Internet connection might get lost
* At any time, application crashes
* Restart the application
* Repeat the process
* **Specific Requirements:** Internet must be running and application will be running
* **Technology and Data Variations List:** Good speed internet will be required
* **Frequency of Occurrence:** Once
* **Use case: Email Verification**
* **Scope:** Teacher Mate
* **Level:** User goal
* **Primary Actors:** Teacher, Parent

**Stakeholders and Interests:** Teacher or Parent

* **Preconditions:**

1. Application is running
2. Internet is running
3. Cloud database is integrated with the app
4. User is login

* **Success Guarantee (or Post-conditions):** New User account is created and the desired info is saved on cloud database
* **Main success Scenario:**

|  |  |
| --- | --- |
| **Actor Action** | **System Responsibilities** |
| 1. Teacher or parent starts the application |  |
|  | 1. System opens the start page of app |
| 1. Teacher or Parent will chose the user option whether he/she is Parent or Teacher |  |
|  | 1. System will open the Signup Page |
| 1. User fills the required input fields and press register |  |
|  | 1. System now verifies that account of user does not exists already 2. After verification System creates account of user and saves the user credentials in database 3. After successfully creating account, System opens the verification page |
| 1. User click on the send verification link button |  |
|  | 1. System will send the email verification link to the user email account |

* **Extension:**
* At any time, application will fails
* Restart the application
* Internet connection might get lost
* At any time, application crashes
* Restart the application
* Repeat the process
* **Specific Requirements:** Internet must be running and application will be running
* **Technology and Data Variations List:** Good speed internet will be required
* **Frequency of Occurrence:** Once
* **Use case: Private Chat**
* **Scope:** Teacher Mate
* **Level:** User goal
* **Primary Actors:** Teacher, Parent

**Stakeholders and Interests:** Teacher or Parent

* **Preconditions:**

1. Application is running
2. Internet is running
3. Cloud database is integrated with the app
4. User is login

* **Success Guarantee (or Post-conditions):** New User account is created and the desired info is saved on cloud database
* **Main success Scenario:**

|  |  |
| --- | --- |
| **Actor Action** | **System Responsibilities** |
| 1. Teacher or parent starts the application |  |
|  | 1. System opens the home page |
| 1. Teacher or Parent will chose the classroom |  |
|  | 1. System will open the Timeline page |
| 1. User clicks on the chat icon on header |  |
|  | 1. System opens the chat page containing list of users |
| 1. User click on particular user in the list of users |  |
|  | 1. System opens the page that contains the messages that was exchanged between the users before |
| 1. User types the message and press send button |  |
|  | 1. System then sends the message to user through cloud database |

* **Extension:**
* At any time, application will fails
* Restart the application
* Internet connection might get lost
* At any time, application crashes
* Restart the application
* Repeat the process
* **Specific Requirements:** Internet must be running and application will be running
* **Technology and Data Variations List:** Good speed internet will be required
* **Frequency of Occurrence:** Once
* **Use case: Create Post**
* **Scope:** Teacher Mate
* **Level:** User goal
* **Primary Actors:** Teacher

**Stakeholders and Interests:** Teacher

* **Preconditions:**

1. Application is running
2. Internet is running
3. Cloud database is integrated with the app
4. User is login

* **Success Guarantee (or Post-conditions):** Teacher starts the application and clicks on create post button to create the post, teacher can create post by adding an image or file in the post
* **Main success Scenario:**

|  |  |
| --- | --- |
| **Actor Action** | **System Responsibilities** |
| 1. Teacher or parent starts the application |  |
|  | 1. System opens the home page |
| 1. Teacher or Parent will chose the classroom |  |
|  | 1. System will open the Timeline page |
| 1. User clicks on the create post icon from footer |  |
|  | 1. System opens the create post page |
| 1. User adds the image or file and fill out the description then clicks on post |  |
|  | 1. System then saves the post data on the database 2. If post saves successfully system will pop up the successful alert 3. If post was not saved successfully then system will pop up the error alert |

* **Extension:**
* At any time, application will fails
* Restart the application
* Internet connection might get lost
* At any time, application crashes
* Restart the application
* Repeat the process
* **Specific Requirements:** Internet must be running and application will be running
* **Technology and Data Variations List:** Good speed internet will be required
* **Frequency of Occurrence:** Once
* **Use case: View Post**
* **Scope:** Teacher Mate
* **Level:** User goal
* **Primary Actors:** Teacher, Parent

**Stakeholders and Interests:** Teacher, Parent

* **Preconditions:**

1. Application is running
2. Internet is running
3. Cloud database is integrated with the app
4. User is login

* **Success Guarantee (or Post-conditions):** Teacher or parent starts the application and after selecting the classroom the post can be seen on the timeline
* **Main success Scenario:**

|  |  |
| --- | --- |
| **Actor Action** | **System Responsibilities** |
| 1. Teacher or parent starts the application |  |
|  | 1. System opens the home page |
| 1. Teacher or Parent will chose the classroom |  |
|  | 1. System will open the Timeline page 2. System retrieves the posts that are saved in database related to particular classroom and class teacher 3. If system finds any post, then the post can be seen on the time line page 4. If not found then time line page remains empty |

* **Extension:**
* At any time, application will fails
* Restart the application
* Internet connection might get lost
* At any time, application crashes
* Restart the application
* Repeat the process
* **Specific Requirements:** Internet must be running and application will be running
* **Technology and Data Variations List:** Good speed internet will be required
* **Frequency of Occurrence:** Once
* **Use case: Like Post**
* **Scope:** Teacher Mate
* **Level:** User goal
* **Primary Actors:** Teacher, Parent

**Stakeholders and Interests:** Teacher, Parent

* **Preconditions:**

1. Application is running
2. Internet is running
3. Cloud database is integrated with the app
4. User is login

* **Success Guarantee (or Post-conditions):** Teacher or parent starts the application and after selecting the classroom the post can be seen on the timeline, teacher can like his/her own post and parent can also like the post
* **Main success Scenario:**

|  |  |
| --- | --- |
| **Actor Action** | **System Responsibilities** |
| 1. Teacher or parent starts the application |  |
|  | 1. System opens the home page |
| 1. Teacher or Parent will chose the classroom |  |
|  | 1. System will open the Timeline page 2. System retrieves the posts that are saved in database related to particular classroom and class teacher 3. If system finds any post, then the post can be seen on the time line page 4. If not found then time line page remains empty |
| 1. User clicks on the view icon on the post |  |
|  | 1. System open the view post page |
| 1. User clicks on like icon button on the post |  |
|  | 1. System checks that if particular post is not already liked by same user 2. If not like by same user, then post like is updated on database and successful alert is generated 3. If like by same user then alert of already liked the post will be generated |

* **Extension:**
* At any time, application will fails
* Restart the application
* Internet connection might get lost
* At any time, application crashes
* Restart the application
* Repeat the process
* **Specific Requirements:** Internet must be running and application will be running
* **Technology and Data Variations List:** Good speed internet will be required
* **Frequency of Occurrence:** Once
* **Use case: Comment Post**
* **Scope:** Teacher Mate
* **Level:** User goal
* **Primary Actors:** Teacher, Parent

**Stakeholders and Interests:** Teacher, Parent

* **Preconditions:**

1. Application is running
2. Internet is running
3. Cloud database is integrated with the app
4. User is login

* **Success Guarantee (or Post-conditions):** Teacher or parent starts the application and after selecting the classroom the post can be seen on the timeline, teacher can comment his/her own post and parent can also comment on the post
* **Main success Scenario:**

|  |  |
| --- | --- |
| **Actor Action** | **System Responsibilities** |
| 1. Teacher or parent starts the application |  |
|  | 1. System opens the home page |
| 1. Teacher or Parent will chose the classroom |  |
|  | 1. System will open the Timeline page 2. System retrieves the posts that are saved in database related to particular classroom and class teacher 3. If system finds any post, then the post can be seen on the time line page 4. If not found then time line page remains empty |
| 1. User clicks on the view icon on the post |  |
|  | 1. System open the view post page |
| 1. User adds the comment and clicks on submit |  |
|  | 1. System updates the post 2. If updated successfully then successful alert is generated 3. If not updated successfully then error alert is generated |

* **Extension:**
* At any time, application will fails
* Restart the application
* Internet connection might get lost
* At any time, application crashes
* Restart the application
* Repeat the process
* **Specific Requirements:** Internet must be running and application will be running
* **Technology and Data Variations List:** Good speed internet will be required
* **Frequency of Occurrence:** Once
* **Use case: Add Student**
* **Scope:** Teacher Mate
* **Level:** User goal
* **Primary Actors:** Teacher

**Stakeholders and Interests:** Teacher

* **Preconditions:**

1. Application is running
2. Internet is running
3. Cloud database is integrated with the app
4. User is login

* **Success Guarantee (or Post-conditions):** Teacher after selecting the classroom, clicks on the add student option from home menu button. Add Student page opens, teacher fill out the required fields and submit button. System now adds the student into the database
* **Main success Scenario:**

|  |  |
| --- | --- |
| **Actor Action** | **System Responsibilities** |
| 1. Teacher starts the application |  |
|  | 1. System opens the home page |
| 1. Teacher will chose the classroom |  |
|  | 1. System will open the Timeline page |
| 1. User clicks on the add student option from home menu options |  |
|  | 1. System opens the add student page |
| 1. User fills the required data and then click on submit |  |
|  | 1. System check if child already exists or not 2. If not exists, then add new child 3. If exists then show the alert that child already exists |

* **Extension:**
* At any time, application will fails
* Restart the application
* Internet connection might get lost
* At any time, application crashes
* Restart the application
* Repeat the process
* **Specific Requirements:** Internet must be running and application will be running
* **Technology and Data Variations List:** Good speed internet will be required
* **Frequency of Occurrence:** Once
* **Use case: Edit Post**
* **Scope:** Teacher Mate
* **Level:** User goal
* **Primary Actors:** Teacher

**Stakeholders and Interests:** Teacher

* **Preconditions:**

1. Application is running
2. Internet is running
3. Cloud database is integrated with the app
4. User is login

* **Success Guarantee (or Post-conditions):** Teacher after selecting the classroom, System displays the post on the time line page. User clicks on the more icon on the post, system displays the pop up menu. User clicks on the edit post and edit post page open, user fill out the required data and submit.
* **Main success Scenario:**

|  |  |
| --- | --- |
| **Actor Action** | **System Responsibilities** |
| 1. Teacher starts the application |  |
|  | 1. System opens the home page |
| 1. Teacher will chose the classroom |  |
|  | 1. System will open the Timeline page 2. System displays the post on the time line page |
| 1. User clicks on the more icon on the post, system displays the pop up menu 2. user fill out the required data and submit. |  |

* **Extension:**
* At any time, application will fails
* Restart the application
* Internet connection might get lost
* At any time, application crashes
* Restart the application
* Repeat the process
* **Specific Requirements:** Internet must be running and application will be running
* **Technology and Data Variations List:** Good speed internet will be required
* **Frequency of Occurrence:** Once
* **Use case: Delete Post**
* **Scope:** Teacher Mate
* **Level:** User goal
* **Primary Actors:** Teacher

**Stakeholders and Interests:** Teacher

* **Preconditions:**

1. Application is running
2. Internet is running
3. Cloud database is integrated with the app
4. User is login

* **Success Guarantee (or Post-conditions):** Teacher after selecting the classroom, System displays the post on the time line page. User clicks on the more icon on the post, system displays the pop up menu. User clicks on the delete post
* **Main success Scenario:**

|  |  |
| --- | --- |
| **Actor Action** | **System Responsibilities** |
| 1. Teacher starts the application |  |
|  | 1. System opens the home page |
| 1. Teacher will chose the classroom |  |
|  | 1. System will open the Timeline page 2. System displays the post on the time line page |
| 1. User clicks on the more icon on the post, system displays the pop up menu 2. User clicks on the delete post |  |
|  | 1. System deletes the post from the database |

* **Extension:**
* At any time, application will fails
* Restart the application
* Internet connection might get lost
* At any time, application crashes
* Restart the application
* Repeat the process
* **Specific Requirements:** Internet must be running and application will be running
* **Technology and Data Variations List:** Good speed internet will be required
* **Frequency of Occurrence:** Once
* **Use case: Add Courses**
* **Scope:** Teacher Mate
* **Level:** User goal
* **Primary Actors:** Teacher, System

**Stakeholders and Interests:** Teacher, System

* **Preconditions:**

1. Application is running
2. Internet is running
3. Cloud database is integrated with the app
4. User is login

* **Success Guarantee (or Post-conditions):** Teacher after selecting the classroom, can add courses by clicking on add courses option on home menu, then after adding the required info courses can be added
* **Main success Scenario:**

|  |  |
| --- | --- |
| **Actor Action** | **System Responsibilities** |
| 1. Teacher starts the application |  |
|  | 1. System opens the home page |
| 1. Teacher will chose the classroom |  |
|  | 1. System will open the Timeline page |
| 1. User clicks on the add courses option from home menu button |  |
|  | 1. System displays the add courses page |
| 1. User fill out the required fields and submit button |  |
|  | 1. System add the course into the database and display success alert if data is saved successfully |

* **Extension:**
* At any time, application will fails
* Restart the application
* Internet connection might get lost
* At any time, application crashes
* Restart the application
* Repeat the process
* **Specific Requirements:** Internet must be running and application will be running
* **Technology and Data Variations List:** Good speed internet will be required
* **Frequency of Occurrence:** Once
* **Use case: View Courses**
* **Scope:** Teacher Mate
* **Level:** User goal
* **Primary Actors:** Teacher, Parent

**Stakeholders and Interests:** Teacher, Parent

* **Preconditions:**

1. Application is running
2. Internet is running
3. Cloud database is integrated with the app
4. User is login

* **Success Guarantee (or Post-conditions):** Teacher or Parents after selecting the classroom, can view courses by clicking on the home menu option
* **Main success Scenario:**

|  |  |
| --- | --- |
| **Actor Action** | **System Responsibilities** |
| 1. Teacher starts the application |  |
|  | 1. System opens the home page |
| 1. Teacher will chose the classroom |  |
|  | 1. System will open the Timeline page |
| 1. User clicks on the view courses option from home menu button |  |
|  | 1. System displays the list of courses in course view page |

* **Extension:**
* At any time, application will fails
* Restart the application
* Internet connection might get lost
* At any time, application crashes
* Restart the application
* Repeat the process
* **Specific Requirements:** Internet must be running and application will be running
* **Technology and Data Variations List:** Good speed internet will be required
* **Frequency of Occurrence:** Once

**CHAPTER 3**

**SYSTEM DESIGN**

## System Design

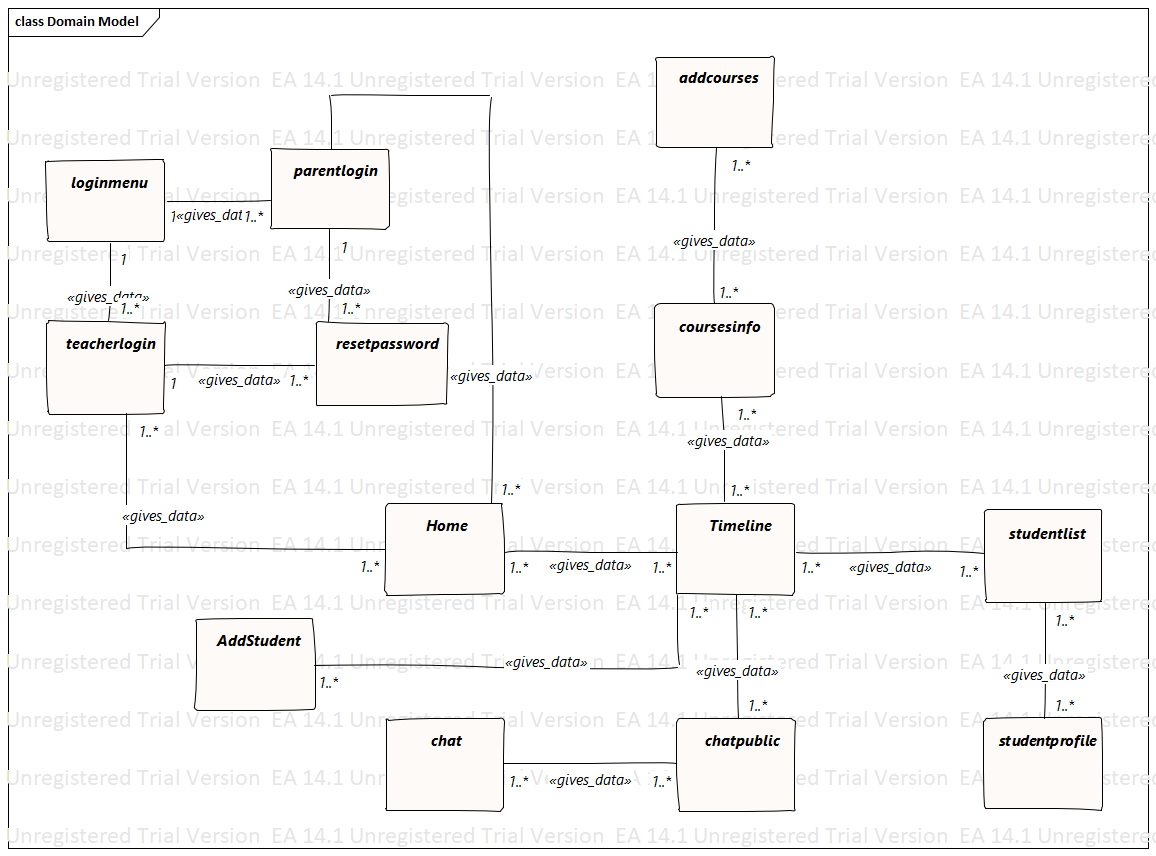
The next step which is system design. System design is the first step in development phase. System design is the process of defining the components, modules, flow charts, block diagrams and interface for a system.

One has to think before implementing that in how many parts he has to split his project for that purpose he use to make flow charts, block diagrams and pseudo codes. A picture is a better way to convey the idea just like a story.

One of the famous phrase is “A picture is worth a thousand words”. It means these flow charts, block diagrams and snap shots of the project are not even helpful for the programmer but also very informative for a non-programming person or a user.

## 3.1 Domain Model

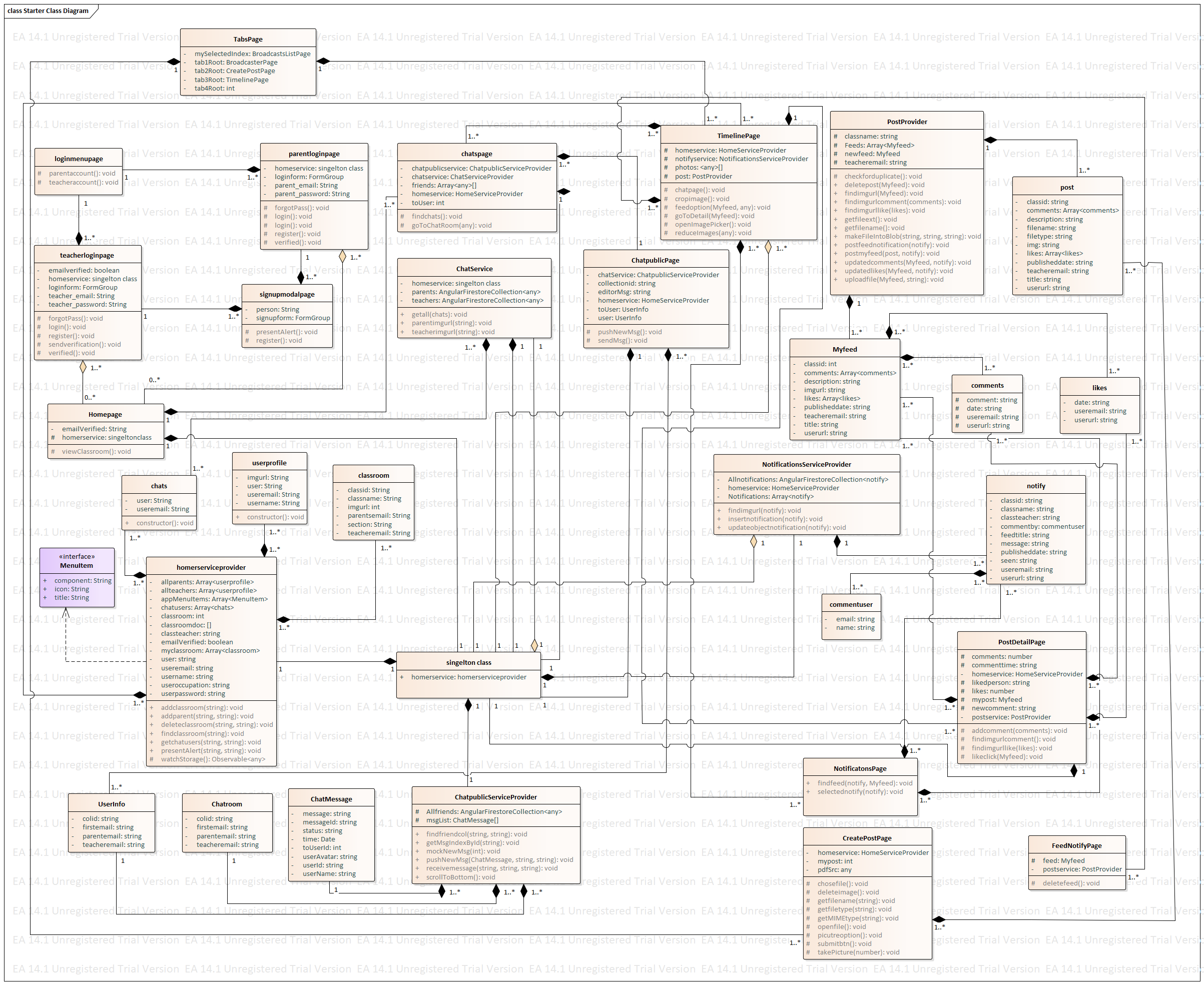
A domain model is a conceptual model of all the topics related to a specific problem. It describes the various entities, their attributes, roles, and relationships, plus the constraints that govern the domain. Domain Model for Teacher Mate is shown here:

****

**Figure 3.1 Domain Modal**

### 3.2 Class Diagram

**Class diagram** is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. The class diagram is the main building block of object-oriented modeling. It is used both for general conceptual modeling of the systematics of the application, and for detailed modeling translating the models into programming code. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed. Following are the corresponding classes of our system:

****

**Figure 3.2 Class Diagram**

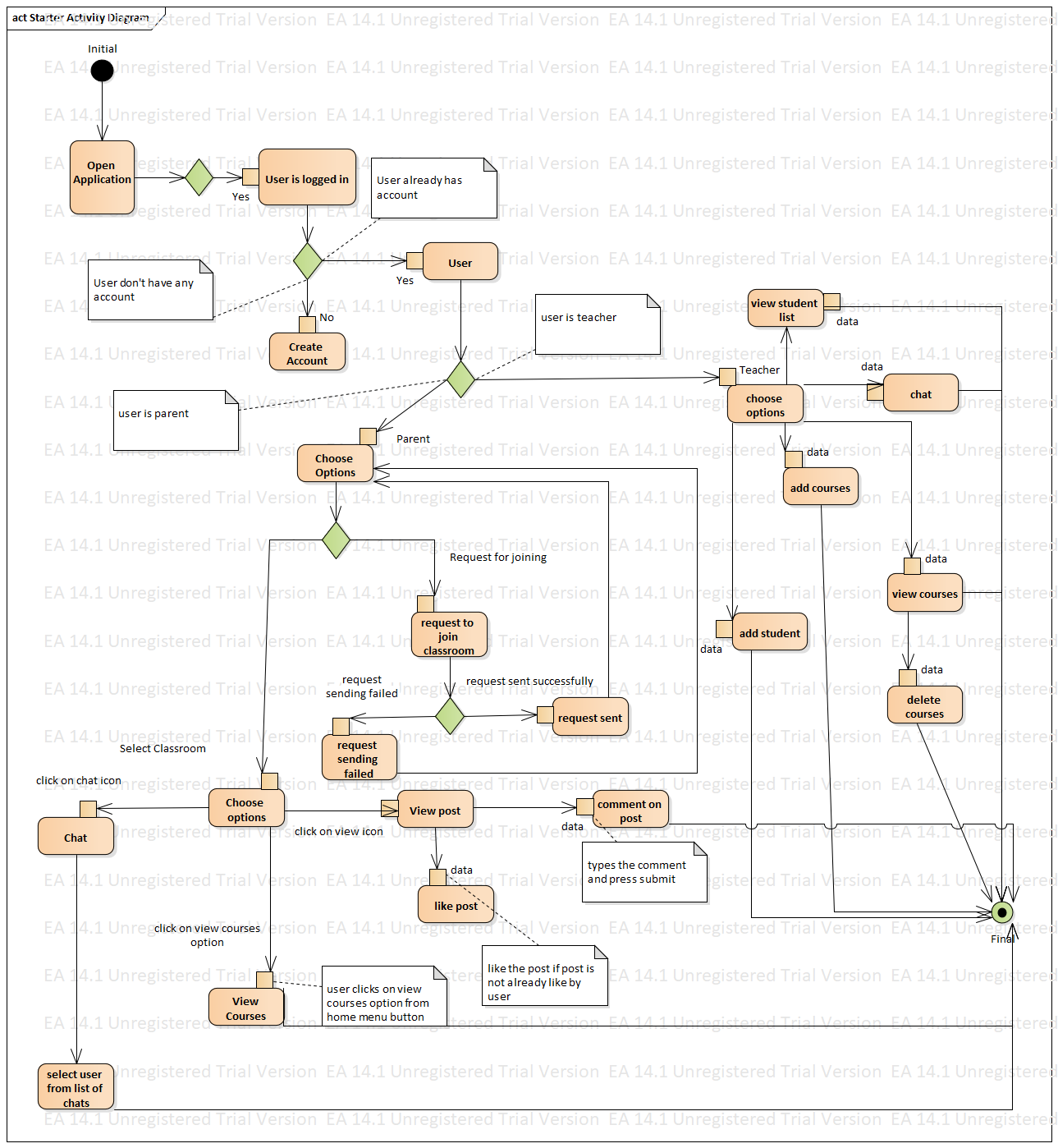
## 3.3 Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

****

**Figure 3.3 Activity Diagram**

**3.3 Sequence Diagram**

Sequence diagram is one of the type of interaction diagram that are used to show the dynamic behavior of the system, that how objects communicate with each other within a system. Here system is created as a white box. System dynamic design is made through Sequence diagram. In it objects are assigned responsibilities by applying different design principles and patterns. Sequence diagram shows how actors and objects interact to realize a Use case scenario.

**CHAPTER 4**

**IMPLEMENTATION**

## Implementation

Implementation often called as deployment or installation, refers to the final process of moving the solution from development mode to production mode .The implementation phase handles the issue of quality, performance, libraries and debugging. The main goal of implementation phase is to implement a system correctly and efficiently by using specific tools and techniques .It deals with the languages, compilers, system etc. in order to translate design in working code. The implementation phase is set of input (design, environmental, performance requirements), output (working system) and techniques (transformation, conversion, testing and monitoring). This chapter discusses the tools and techniques and the reason to select these tools for the implementation of design.

**4.1 Tools and Technologies**

#### 4.1.2 HTML

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. HyperText is the method by which you move around on the web by clicking on special text called hyperlinks which bring you to the next page. The fact that it is hyper just means it is not linear i.e. you can go to any place on the Internet whenever you want by clicking on links there is no set order to do things in.

Markup is what HTML tags do to the text inside them. They mark it as a certain type of text (italicised text, for example).

HTML is a Language, as it has code-words and syntax like any other language.

HTML consists of a series of short codes typed into a text-file by the site author — these are the tags. The text is then saved as a html file, and viewed through a browser, like Internet Explorer or Netscape Navigator. This browser reads the file and translates the text into a visible form, hopefully rendering the page as the author had intended. Writing your own HTML entails using tags correctly to create your vision. You can use anything from a rudimentary text-editor to a powerful graphical editor to create HTML pages.

using HTML and designing good websites is a different story, which is why I try to do more than just teach you code here at HTML Source I like to add in as much advice as possible too. Good website design is half skill and half talent, I reckon. Learning techniques and correct use of your tag knowledge will improve your work immensely, and a good understanding of general design and the audience you’re trying to reach will improve your website’s chances of success. Luckily, these things can be researched and understood, as long as you’re willing to work at it so you can output better websites.

#### 4.1.3 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

The style sheets define the colour, size and position of text and other HTML tags, while the HTML files define the content and how it is organised. Separating them allows you to change the colour scheme without having to rewrite your entire web site.

The cascading means that a style applied to a parent element will also apply to all children elements within the parent. For example, setting the colour of body text will mean all headings and paragraphs within the body will also be the same colour..

Changes to the graphic design of a document (or hundreds of documents) can be applied quickly and easily, by editing a few lines in the CSS file they use, rather than by changing markup in the documents.

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs,variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

#### 4.1.3 Bootstrap

Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web. Additionally, Bootstrap requires [**jQuery**](https://jquery.com/) to function. jQuery is an extremely popular and widely used JavaScript library, that both simplifies and adds cross browser compatibility to JavaScript.

***4.1.4 Firebase***

Firebase is for providing effective and fast message facility in our app, as messaging needs to be send and receive quickly without any delay so Firebase plays an important role here because firebase is cloud based database, Angular will send the response to Firebase and Firebase will update the database, in this fashion one to one messaging our app will be implemented. Most databases require you to make HTTP calls to get and sync your data. Most databases give you data only when you ask for it.When you connect your app to Firebase, you’re not connecting through normal HTTP. You’re connecting through a WebSocket. WebSockets are [much, much faster than HTTP](http://www.websocket.org/quantum.html). You don’t have to make individual WebSocket calls, because one socket connection is plenty. All of your data syncs automagically through that single WebSocket as fast as your client’s network can carry it. Firebase sends you new data as soon as it’s updated. When your client saves a change to the data, all connected clients receive the updated data almost instantly.

***4.1.5 AngularJs***

AngularJS is a JavaScript-based open-source front-end web framework mainly maintained by Google and by a community of individuals and corporations to address many of the challenges encountered in developing single-page applications. It aims to simplify both the development and the testing of such applications by providing a framework for client-side model–view–controller (MVC) and model–view–viewmodel (MVVM) architectures, along with components commonly used in rich Internet applications. (This flexibility has led to the acronym MVW, which stands for "model-view-whatever" and may also encompass model–view–presenter and model–view–adapter.) In 2014, the original AngularJS team began working on the Angular web framework.

### 4.2 Code

**CHAPTER 5**

**SYSTEM TESTING**

## System Testing

System testing is an empirical investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software Testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks at implementation of the software. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs.

Testing is the process of detecting errors. Testing plays a critical role in assuring quality and reliability of software. The results of testing are used later on during maintenance also.

We have performed Black Box Testing.

### 5.1 Testing Objectives

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say:

1. Testing is process of executing a program with intention of finding an error.
2. A successful test is note that uncovers an as yet undiscovered error.
3. A good test case is one that has a high probability of finding error, if it exists.
4. The tests are inadequate to detect possibly present errors.
5. The software more or less confirms to the quality and reliable standards.
6. Testing of the system makes it error free; hence the system is more reliable and efficient.

### 5.2 Testing Cases

The tests cases are performed during the project development which is verified successfully and are as follows

**CHAPTER 6**

**CONCLUSION**

# 6 Conclusion

## Limitations of the System

**A2: REFERENCES**