Software Requirements Specification

Submitted to:

Dr. Adil Raja
Faculty of Engineering and Computer Science
Namal College, Mianwali.

Submitted by:

Group #03

M.Ikram Ullah Khan (UoB # 15026431)

Muhammad Usman (UoB # 15026436)

Naeem Ullah Khan

Shafi Ullah Khan

Abdul Kareem

Arshad Majeed

Students of Computer Science department Year-3
Namal College, Mianwali.

November 03, 2017.

Table of Content

1. Introduction

Before going to details, we give an overview of the purpose of the document, scope of project and different features covered by that Document. Later on, we are looking forward to those aspects which is covered by the SRS Documents like Functional Requirements, Use Cases etc.

1.1 Purpose

It is universal rule that if you have a strong base, your product will also be more refine and strengthen. So following that rule, we also use SRS documentation method. The basic purpose of that document is to understand the requirements of the project and give you detail descriptions of every section which help you to understanding the overall projects. The document also helps us to negotiate on requirements and get permission from the client before going to design and implementation phase of project. It is a complete road map which give clear pictures of the project before moving ahead to next stage.

1.2 Scope of Project

The Scope of our project is to digitalize the current manually tutorials groups allocation system. So our **"Tutorial Group App"** is standalone app which is deployed in a school to fulfill the circumstance. It is run by "**school secretary"** and it helps to create tutorials group and allocate them to teacher.

The system should provide option to the admin to makes tutorials groups of student. It also give flexibility to edit and delete students and staff members allocated to the specific group. The sectary of the school can upload the students using the file. User can also add new student and teacher manually using GUI of system. You can also delete student, group and teacher from system. System provide some additional options like lists of students, groups and staff members in PDF format and sorting the students and staff members in alphabetical order of their names.

1.3 Attending Audience

This document helps the developers and different people who are associated with project to understand the different section of the product. It provides a helpful material to end user to understand the different functionality of the product. It will help the developer in the future, to versioned the product. There is a list of the audience which can use this document to understand the functionality of the product.

1.3.1 The Client

By getting the requirements, we tries to give organized and structured description of every sections. It will help the user or client to know about every section of the "Tutorial Group App". We try to use simpler language which could be understandable to the client. We try to give description of every technical term which can be understandable to the end user or client.

1.3.2 The Interface Designer

The document will also help the development team specially interface designer to understand the requirements and according to that requirements, they can design interface and class diagrams. It

also give a complete organization of the product and functionality which helps to design a good interface of the product.

1.3.3 Development Team

Reading this document, development team can easily transform the requirements into the class diagrams and structure code for the software or app. They can easily relate every section of the product. It is very helpful to the development team in the future, when they are going to change the some parts of the app or software.

1.3.4 Testing Team

One of the important phase of the software development is testing of the software .So this documents is useful for the testing team to understand every section and test according to given scenario and requirements. So it will be helpful to compile the results of the testing and give feedback of every section.

1.4 Definitions/acronyms/abbreviations

It is important to give description of the some of the terms which is used in this document. It will help the different audience to understand the meaning of the terms within the document. So here is details of the some of the term

- **Client/User/Admin:** The person who is running that app i.e. in our projects school secretary is the client/user or admin.
- **Android:** Android is mobile application developing platform, where you can develop the different apps for the mobile.
- **SRS (Software Requirements Specifications):** It is official statement or requirements document of the system which should be implemented by the developers.
- **Data Base:** Data Base is a well organize form of saving data which is used in the development of software.
- **GUI:** Graphical User Interface
- **OS:** Operating System i.e. like Windows or Linux or Android.
- **MB:** Mega Bytes
- **UoB:** University of Bradford
- **Project/Software/System/Product/App:** Our **"Tutorial Group App"** is a software, app or system which is providing required functionality to the user.

2. Overall Description

This section deals with the overall description of the whole product using different prospective. Here, we explained how system's different parts interact with each other's and how they show their functionality. We also working with different functionalities of the system .In the end of section, we talk about the constrains and dependencies of the system

2.1 Product Perspective

This software is a standalone app which has independent existence. The software is basically about the management system where user have different rights like adding student, adding mentor, creating groups and edit different information data of the system.

The software saves all the information within the database which gives a security insurance of the data. Software has capability of read and write from the database. The data base will be associated with in the product.

2.2 Product Functions

In our mobile application, user have multiple function which can be perform depends on the user's input. Some of the function are described here

- Adding the information of the students using File or Add manually using GUI.
- User can also add Mentor or teacher.
- User also has an option of creating group.
- Application also have capacity of editing different information like edit student information, edit group information or edit mentors etc.
- Application also have capability of delete information like delete group, delete Members etc.
- Application also have functionality of showing data like list of un-allocated students, groups and their members etc.

2.3 User Characteristics

From the requirements documents and meeting with clients, we confirm about one type user for system. So this app is design for one user, it could not be shared data or it have multiple user. Only one user, can managed data according to its own choice. In our case, School secretary is the only user who have rights of using application. This type of user can use the app but everybody has its own data, it will not share with any other user of such app.

2.4 Constraints

As required of saving data in the document, it is important to use the data base which eventually can create problem memory so we have to restrict our memory allocation at some limit. It also makes our app heavy which can be slow down and crash our system.

But overall, reading the documents and meeting with client, we do not find any constrains which limit our products.

2.5 Assumptions and Dependencies

Some of the assumption are described here,

- As clients does not force us of using any specific platform, so we try to provide modern technology, so we have choices **Android** Platform and we are assuming that our user have **Android O.S.**
- As data is saved with in the memory, we assume that user have enough memory and other resources which needed for the app, otherwise it will affect the app's performance.

- The user should basic knowledge of the mobile usage and computer usage.

2.6 Apportioning Of Requirements

Most of the requirements are fulfilled, but some requirements cannot be fulfilled due to limited time or any other issue. These requirements specifications could possibly be available in the updated version.

3. Specific requirements

This section of the document, covers all of the functional and quality requirements of the software. It will explain every section and give guidelines about different features of the system.

3.1 External interface Requirements

This parts deals with the externally working interfaces like User Interfaces, interaction with other external components like hardware Interface. This gives detailed information about all the inputs and outputs of the software. It also provides the prototype of user interface of software. It gives detail knowledge of communication interfaces, software interface and hardware.

3.1.1 User Interface

User interface helps user to use the functionality of the system's every functions without going to complex code. Here we have a list of User Interfaces for every function of the system.

3.1.1.1 Log in Interface

At the starts, app should shows a log in page before going to other sections. It have two input fields one for user name and other is for password. There is another option of resting password which helps the user when he forgets the password. Log in interface prototype is shown in Figure 3.1.1.1

Figure 3.1.1.1(Log in Interface)

3.1.1.2 Home Page Interface

After successfully log in to the system, user is redirected to "Home Page". Here user can see different option which the system possess like he/she can see list of different groups. The Home page also have options (which can be seen by clicking the three dots on home screen) which helps user to use different functionality of system like adding students, creating group, delete group etc. Home page interface can be seen in the figure 3.1.1.2

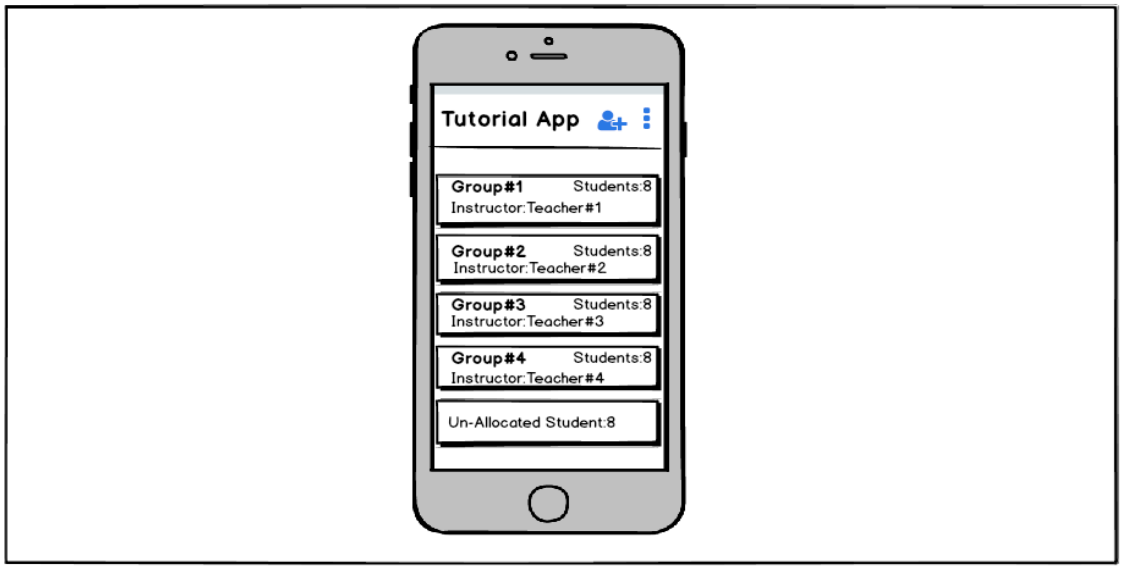


Figure 3.1.1.2(Home Page Interface)

.

3.1.1.3 Upload Student Interface

As required of file uploading, system have interface where you can choose a file and by clicking upload button all data of file is uploaded to the system. This can be seen in Figure 3.1.1.3.

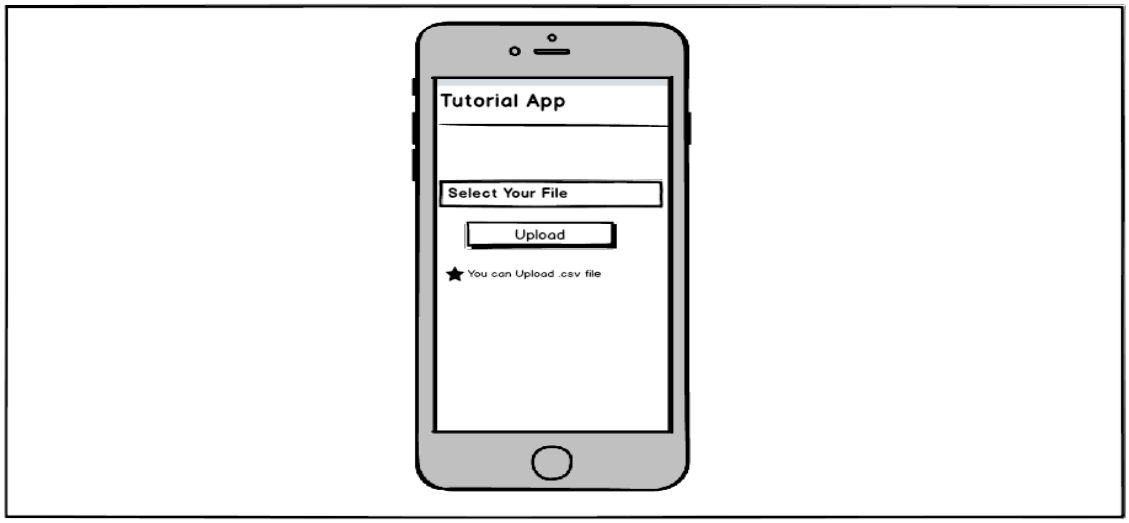


Figure 3.1.1.3(File Chooser)

3.1.1.4 Add Student Interface

Add student interface gives you flexibility of add students manually by filling different text fields like name, year, UOB and select group in which you want to add. You have add button which is used to add the student. This can be seen in Figure 3.1.1.4.

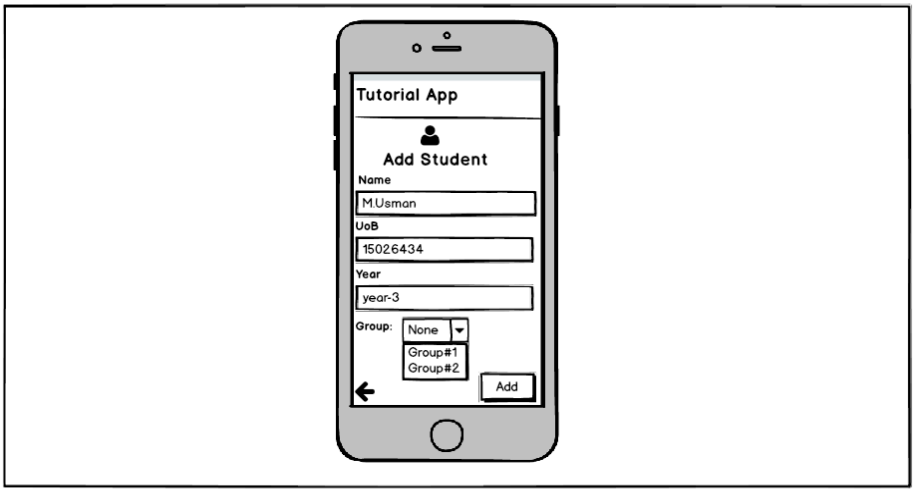


Figure 3.1.1.4 (Add Student Manually)

3.1.1.5 Add New Group Interface

If user want to add a new group within the system, he can do by clicking "Add new group" option which is at "Home page" which redirected to **Add New Group Interface** where you can named the group, select the instructor and members of the group. After clicking the save button, the new group has been created. This interface can be shown in the figure 3.1.1.5.1 and 3.1.1.5.2.

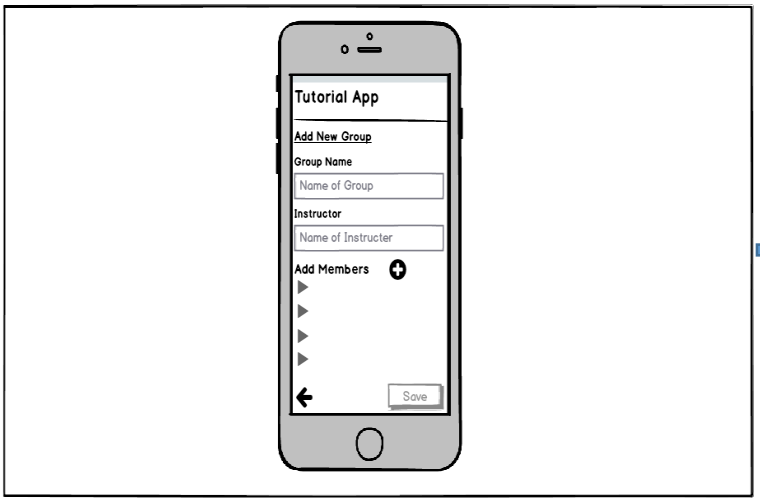


Figure 3.1.1.5.1 (Add new Group)

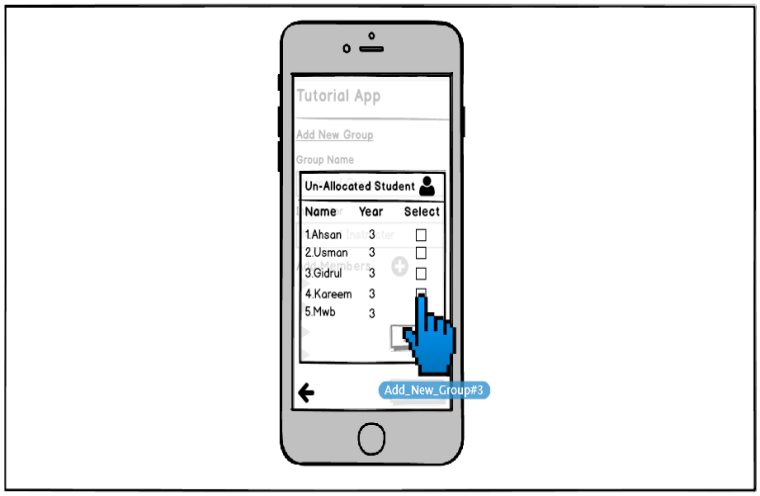


Figure 3.1.1.5.2 (Clicking (+) sign you get list of un-allocated students which can be added to group)

3.1.1.6 Edit Group Interface

On the "Home Page" of app, you can see list of all groups which already exists. System provides opportunity of editing the group. By tapping the name of group, you will see a confirmation box of edit group (Figure 3.1.1.6.1). By clicking, that you will see a form of editing group, where you can change the name of group, instructor of group and members of the group (Figure 3.1.1.6.2).

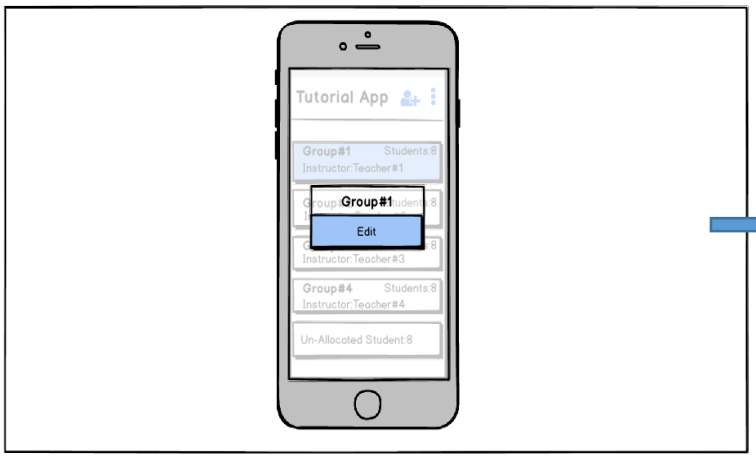


Figure 3.1.1.6.1

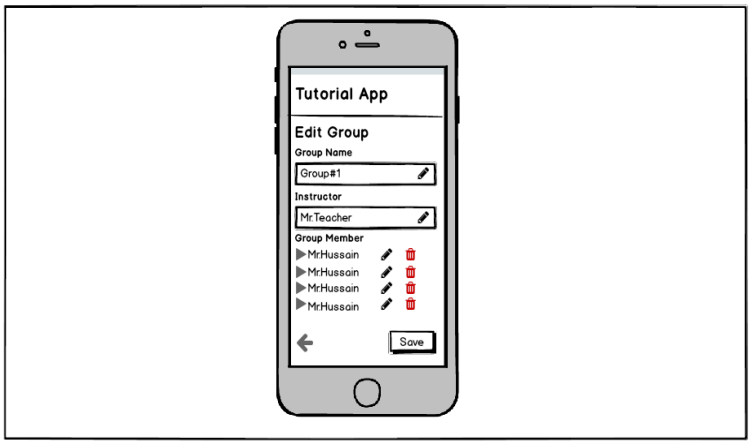


Figure 3.1.1.6.2

3.1.1.7 Delete Group Interface

If you want to delete any specific group or want to delete all the group, the system provides flexibility of doing that. On home page, you have section which leads to the deleting section of the system. You can also delete groups one by one by selecting them or you also have select all option in a while and delete all the groups. The design is very simple to use and you can delete the all groups easily. The Figure 3.1.1.7.1 shows deletion of one by one and Figure 3.1.1.7.2 shows deletion of all groups in one steps.

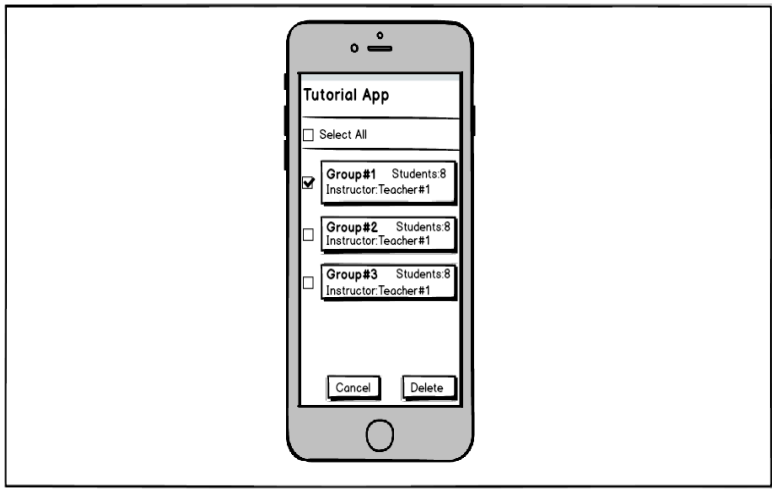


Figure 3.1.1.7.1

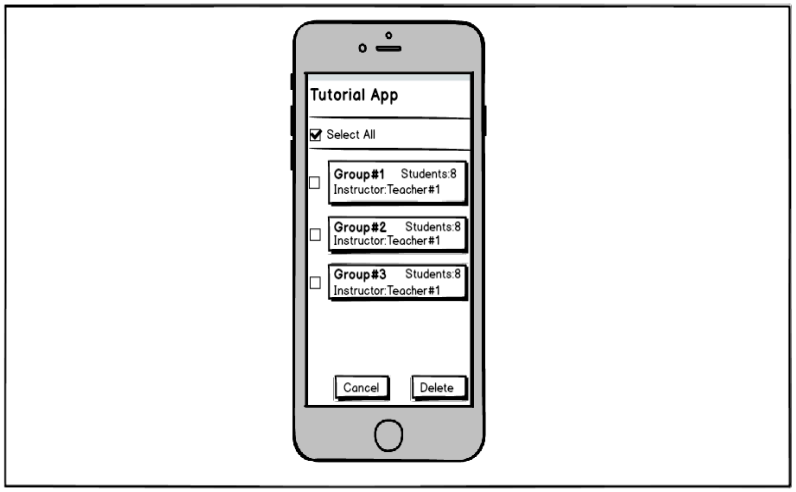


Figure 3.1.1.7.2

3.1.1.8 Edit Student Interface

System also allow to edit information of student. Here you can change the name, UoB and Allocated group. It will help to save record correctly and manage in correct manners. This can be done by using Edit Student Interface.

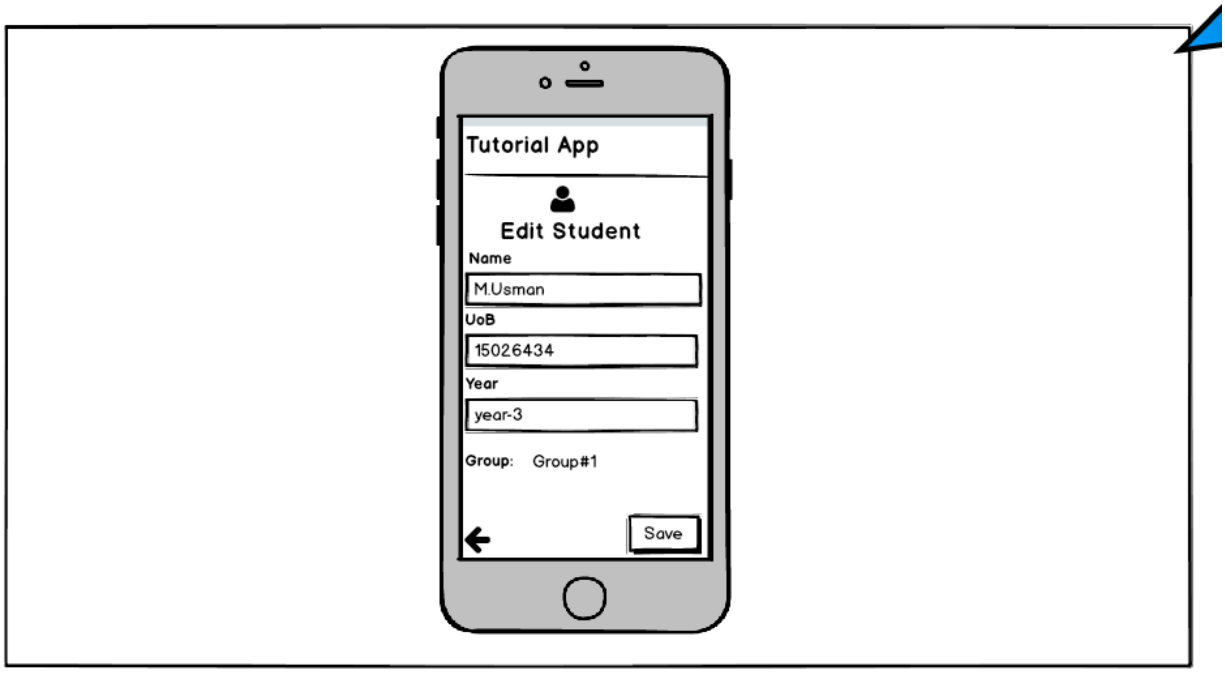


Figure 3.1.1.7.1

3.1.1.9 Add Teacher or Mentor Interface

If you want add a new Mentor or Teacher interface, System allow to perform such action. By filling all the inputs fields, you can add new teacher or mentor.

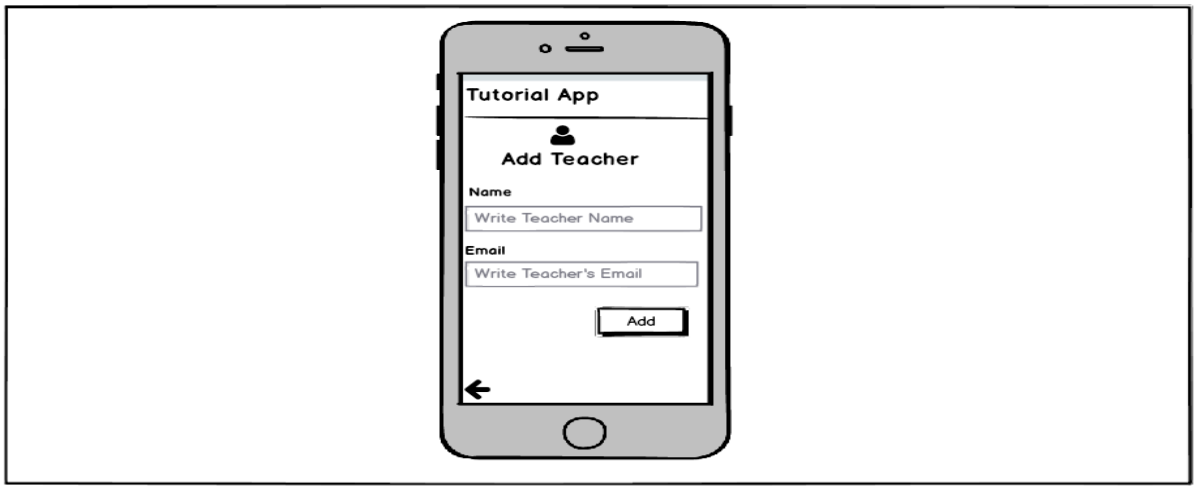
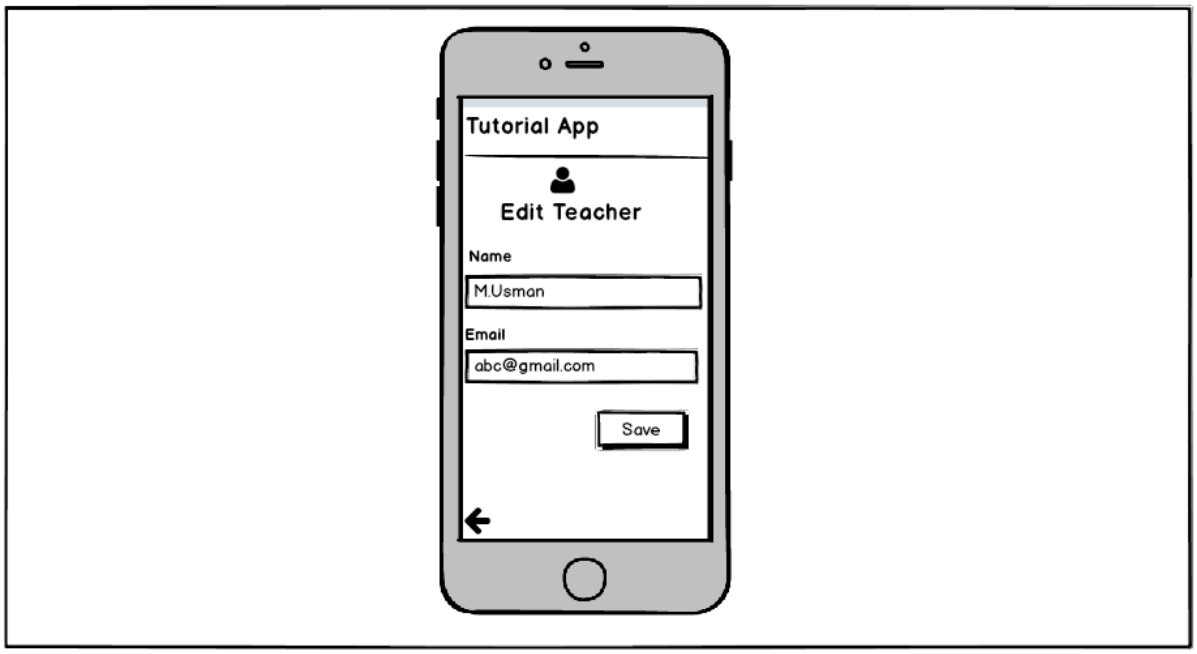


Figure 3.1.1.9

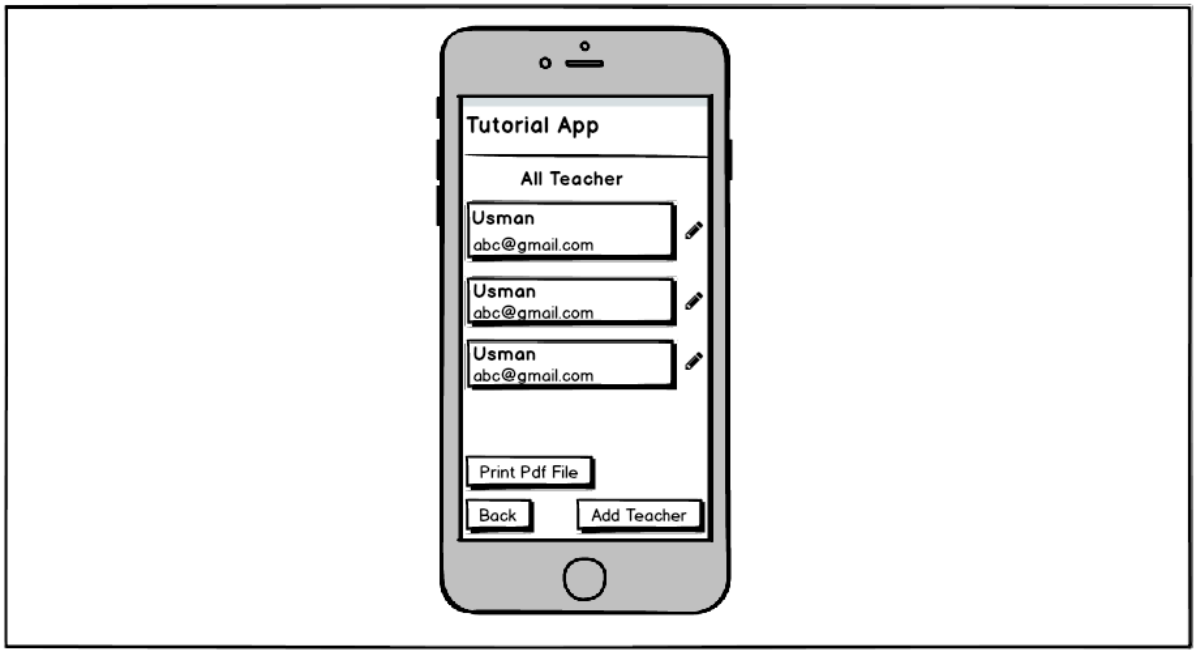
3.1.1.10 Edit Teacher or Mentor Interface

There is a section for editing the mentor or teacher. You can edit mentor information, by changing the text of input fields of name and E-mail of the mentor.



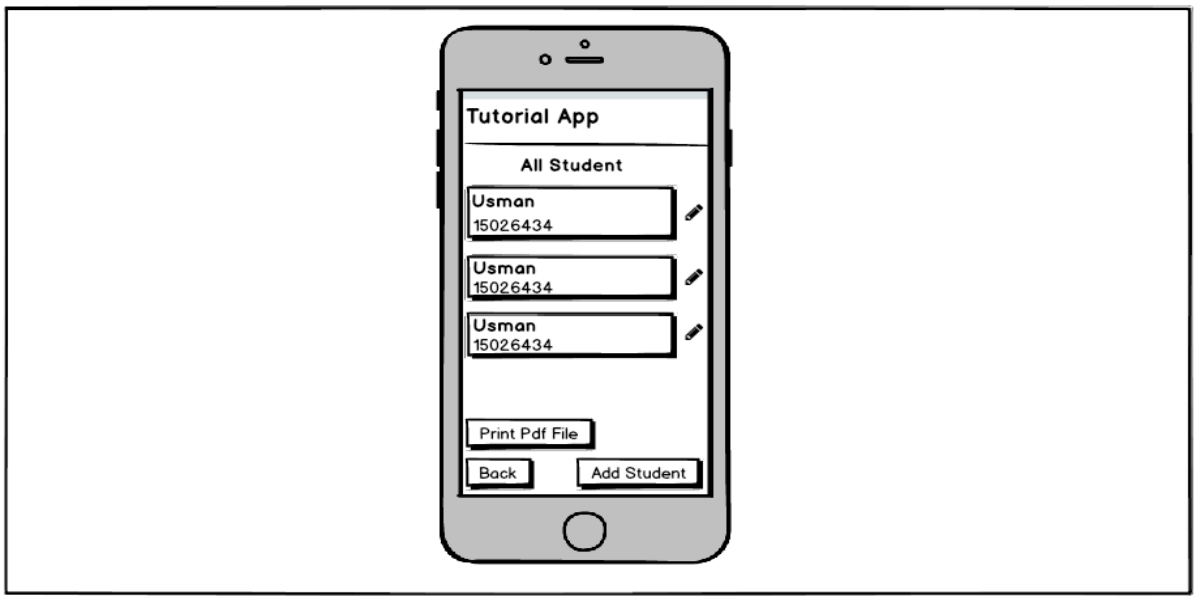
3.1.1.11 All Teacher or Mentor Interface

There is interface with in the system, where you can see a list of all the teacher. Here you can edit and add a new teacher within the list of all teachers. You can get PDF file of all teacher's list.



3.1.1.12 All Students Interface

System also provide separate interface, where you can see a list of all the students. Here you can edit and add a new teacher within the list of all teachers. You can get PDF file of all teacher's list.



3.1.1.13 Print PDF File

There is interface, where you can Print PDF list of students, groups, and teachers. This will helps to exports list from system.

3.1.2 Hardware interfaces

The system has no direct connection with the hardware or have not any designated hardware, but user should have a mobile or android device which can run that software or app. The App also needed some memory up to 10 MB to work in smoothly without any disturbance.

3.1.3 Software interfaces

As the app is developed on the Android platform, so to run this app smoothly, user should have android operating system otherwise it is not going to work. To run this app on windows, you should have an android emulator, which run that app.

3.1.4 Communication Interfaces

As the software or application is a standalone app, it is run by only one type user and it does not share any database or another information with any other, so it does not need any communication interface.

3.2 Functional Requirements and Use Cases

In this section, we will provide some brief functional requirements. Functional requirements identifies and deals with the basic functionalities done by the software.

3.2.1 Log in to system

Description: To ensure the security of the system, user should be log in before using all the features of the app.

Use Case:

Use case Name	Log in into the system
Priority	Essential
Trigger	The user should click the button to log in the system
Pre-Condition	The user should have a correct user name and password.
Task Sequence Or Basic Path	i. The user open the system. ii. The user write user name and password. iii. When he click the button, the user name and password is crossed checked within database. iv. If it is correct information then he will be log in to the system.
Alternate Path	The user can reset its password.
Post Condition	User is redirected to the **"Home Screen"** of the system
Exception Condition	If User cannot login, then system give flexibility of resetting password process.

3.2.2 List of Groups

Description: After the successful login, user get permission of **"Home Screen"** of app. At **"Home Screen"** user can see list of groups which exists within the system.

Use Case:

Use case Name	List of Groups
Priority	Essential
Trigger	N/A

Pre-Condition	The user should be login to the system and it is on home screen
Task Sequence Or Basic Path	i. The user should be log in to the system. ii. The User should be on the **"Home Screen"** of App. iii. The **"Home Screen"** can see multiple groups which already exists within the system.
Alternate Path	N/A
Post Condition	User can check and see the members of each group which exists within the groups. It also have options of edit and delete any specific group.
Exception Condition	N/A

3.2.3 Add Students

To add students data, we can use either do this by using file or we can add manually.

3.2.3.1 Add Students Using File

Description: In this method, you can add and update the information of student using the file. You choice the file which have required data and upload into the system. This data eventually save into the database.

Use Case:

Use case Name	Add Students Using File
Priority	Essential
Trigger	By Selecting file and clicking the upload button.
Pre-Condition	The user should have .csv extension file.
Task Sequence Or Basic Path	i. From Home screen, user select the **"Upload File".** ii. He/she select the file from its directory. iii. Then he click the upload button.
Alternate Path	N/A
Post Condition	Data retrieve from file and save into database. This can be seen with in the system.
Except. Condition	If there is no file found, then It will give error of file no-selection error.

3.2.3.2 Add Students Using System's GUI

Description: In this method, you can add the information of student using the GUI Form. By filling all the required fields, student can be added to the system. This data eventually save into the database.

Use Case:

Use case Name	Add Students Using System's GUI
Priority	Essential
Trigger	By clicking the Add button.
Pre-Condition	The user should filled all the fields properly.
Task Sequence Or Basic Path	i. From Home screen, user select the "add student/Instructor" ii. Then choice the GUI method. iii. He/she Fill all the fields. iv. Then he/she click the add button to submit.
Alternate Path	N/A
Post Condition	By submitting the button, all information save into database. This can be seen with in the system.
Exception Condition	If some fields are not properly filled, then it highlights the error or un-filled fields.

3.2.4 Add Instructor

Description: You can add new Instructor using System's GUI Form. By filling all the required fields, instructor can be added to the system. This data eventually save into the database.

Use Case:

Use case Name	Add Instructor
Priority	Essential
Trigger	By clicking the Add button.

Pre-Condition	The user should filled all the fields properly.
Task Sequence Or Basic Path	i. From Home screen, user select the “New Teacher/Instructor” tab ii. Then He/she Fill all the fields. iii. Then he/she click the add button to submit.
Alternate Path	i. From Home screen, user select the “All Teacher/Instructor” tab ii. To add new Instructor, click on the (+) sign. iii. Then He/she Fill all the fields. iv. Then he/she click the add button to submit.
Post Condition	By submitting the button, all information save into database. This can be seen with in the system and instructor will be available for the new group allocation.
Exception Condition	If some fields are not properly filled, then it highlights the error or un-filled fields and instructor is not added to the system.

3.2.5 Create New Groups

Description: One of the key functionality is to create groups of students and allocate instructor to the groups.

Use Case:

Use case Name	Create New Groups
Priority	Essential
Trigger	Press “Add” Button
Pre-Condition	The user should filled all required fields and there is un-allocated students.
Task Sequence Or Basic Path	i. From Home screen, user select the “Create New Group” tab. ii. You will get a form of different text fields. iii. User write the “Group name”. iv. Select Instructor and then add members. v. Then he click the add button
Alternate Path	N/A
Post Condition	Data is saved into database, User can see details of the group.
Exception Condition	If form is not properly filled then it will gives error message of empty fields and group has not be created.

3.2.6 Edit Information

Description: This application also provides facility to user to edit the information in multiple ways like

3.2.6.1 Edit Group Information

3.2.6.2 Edit Student Information

3.2.6.3 Edit Instructor Information

3.2.6.1 Edit Group Information

Description: Here you can edit the name of group, instructor of group and the members which are already existing with the group.

Use Case:

Use case Name	Edit Groups information
Priority	Essential
Trigger	Press “Save” Button
Pre-Condition	Group should already exists within the system, it should have name, instructor name and members with the group
Task Sequence Or Basic Path	i. On Home Screen you will see a list of different groups ii. By long tapping on any of these groups, a pop message of “Edit” will be shown on screen. iii. By clicking the “Edit” option, you will get details of that group which you have tapped. iv. Here you can change group name, instructor of group and members of group. v. Then click “Save” button to update information.
Alternate Path	N/A
Post Condition	Data is updated into database, User can see updated details of the group.
Exception Condition	If form is not properly filled then it will gives error message of empty fields and group has not be updated.

3.2.6.2 Edit Student Information

Description: Here you can edit the information of student like name, UoB and Group of the student.

Use Case:

Use case Name	Edit Student information
Priority	Essential
Trigger	Press “Save” Button
Pre-Condition	Student should exists within the system.
Task Sequence Or Basic Path	i. On Home Screen you will see a list of different groups ii. By clicking on any of these groups, you will see details of every member of group. iii. Clicking the “Member”, you will get details of that “Member” which you have clicked. iv. Here you have edit option for that member, clicking that you can change name, UoB and group. v. Then click “Save” button to update information.
Alternate Path	i. On Home Screen you will see an option of “All Student” ii. By clicking on this option, you will see a list where you can see details of every students. iii. Clicking the on any student, you will get details of that “student”. iv. Here you have edit option for that student, clicking that you can change name, UoB and group. v. Then click “Save” button to update information.
Post Condition	Data is updated into database, User can see updated details of the Person.
Exception Condition	If form is not properly filled then it will gives error message of empty fields and Person’s details does not be updated.

3.2.6.3 Edit Instructor Information

Description: Here you can edit the information of the instructor like name, email and it assigned group (if assigned).

Use Case:

Use case Name	Edit Instructor information
Priority	Essential

Trigger	Press “Save” Button
Pre-Condition	Student should exists within the system.
Task Sequence Or Basic Path	i. On Home Screen you will see an “All Teacher” Tab. ii. By clicking on this, you will see details of every Instructor. iii. Clicking the “Instructor”, you will get details of that “Instructor” which you have clicked. iv. Here you have edit option for that instructor, clicking that you can change name, email and assigned group. v. Then click “Save” button to update information.
Alternate Path	N/A
Post Condition	Data is updated into database, User can see updated details of the Person.
Exception Condition	If form is not properly filled then it will gives error message of empty fields and Instructor details does not be updated.

3.2.7 Delete

System provides functionality of delete some items on the basis of different

3.2.7.1 Delete Group

3.2.7.2 Delete Person

3.2.7.3 Delete Instructor

3.2.7.1 Delete Group

Description: Here you can delete group which are already existing.

Use Case:

Use case Name	Delete Group
Priority	Essential
Trigger	Press “Delete” Button
Pre-Condition	Group should already exists within the system.
Task Sequence Or Basic Path	i. On Home Screen, you will select a “Delete” Tab. ii. You will find a list of already exiting groups within the system. iii. You can delete groups in two ways,

	i. Select the group one by one. ii. Other is select all groups. v. Then click “Delete” button to Delete group.
Alternate Path	N/A
Post Condition	Group data is deleted from database. Students are listed into un-allocated categories.
Exception Condition	N/A

3.2.8 Print PDF File

Description: System also provides functionality of creating pdf file of the students and exports from the system.

3.2.8.1 Groups PDF File Creator

3.2.8.2 Students PDF File Creator

3.2.8.3 Teacher PDF File Creator

3.2.8.1 Groups PDF File Creator

Description: You have an option of getting list of groups in form of PDF File.

Use Case:

Use case Name	Groups PDF File Creator
Priority	Essential
Trigger	Press “Print” icon
Pre-Condition	A list of Group should be exists within the system.
Task Sequence Or Basic Path	i. On Home Screen, you will select a “Print” icon. ii. By Clicking this icon, list of Groups is converted into PDF Format. iii. This can be export as PDF file from the system.
Alternate Path	N/A
Post Condition	A list of groups can be copied into the file and you can see all groups within the file.

Exception Condition	N/A

3.2.8.2 Students PDF File Creator

Description: You can get the list of all Students using the Print PDF functionality of system.

Use Case:

Use case Name	Students PDF File Creator
Priority	Essential
Trigger	Press “Print” icon
Pre-Condition	A list of all students should be exists within the system.
Task Sequence Or Basic Path	i. On Home Screen, you will select an “All students” icon. ii. You will get to “All students” interface. ii. By Clicking this icon, list of students is converted into PDF Format. iii. This can be export as PDF file from the system.
Alternate Path	N/A
Post Condition	A list of students can be copied into the file and you can see all students within the file.
Exception Condition	If there is no student then it could not create file.

3.2.8.3 Teachers PDF File Creator

Description: You can get the list of all Teachers using the Print PDF functionality of system.

Use Case:

Use case Name	Teachers PDF File Creator
Priority	Essential

Trigger	Press “Print” icon
Pre-Condition	A list of all teachers should be exists within the system.
Task Sequence Or Basic Path	i. On Home Screen, you will select an “All Teachers”. ii. You will get to “All Teachers” interface. ii. By Clicking “Print” icon, list of teachers is converted into PDF Format. iii. This can be export as PDF file from the system.
Alternate Path	N/A
Post Condition	A list of teachers can be copied into the file and you can see all teachers within the file.
Exception Condition	If there is no teacher then it could not create file.

3.2.9 All Students

Description: System has an option of seeing all students together using the functionality of “All Students”.

Use Case:

Use case Name	List of all Students
Priority	Essential
Trigger	Click the All students button.
Pre-Condition	Students should be exists within the system.
Task Sequence Or Basic Path	i. On Home Screen, you will select an “All Students” option. ii. You will get to “All Students” interface. iii. You can see all students on that interface.
Alternate Path	N/A
Post Condition	A list of students can be seen in form of list and it can be edited and check the details of all students.
Exception Condition	If there is no student then it could not show students with in the list.

3.2.10 All Teachers

Description: System has an option of seeing all teachers together using the functionality of "All Teachers".

Use Case:

Use case Name	List of all Teachers
Priority	Essential
Trigger	Click the All Teachers button on Home Screen.
Pre-Condition	Teachers should be exists within the system.
Task Sequence Or Basic Path	i. On Home Screen, you will select an "All Teachers" option. ii. You will get to "All Teachers" interface. iii. You can see all Teachers on that interface.
Alternate Path	N/A
Post Condition	A list of teachers can be seen in form of list and it can be edited and check the details of all teachers.
Exception Condition	If there is no teachers then it could not show teachers with in the list.

3.3 Non-functional Requirements / Software Attributes

3.3.1 Performance

As our App is a standalone app and does not needs any external resources except memory, so it performance will be much better than any other app. It responds to every instruction smoothly without any problem. But if there is lack of memory by user, then it may affect the performance of app.

3.3.2 Scalability

As it is a standalone app which is designed for one type of user, so scale of its services limited to only one type of user. So it does not share services to any other user.

3.3.3 Flexibility and Usability

System is design in a way that, it is easy to understandable to every user. The icons and language is easy to understandable to common men. The design follows the modern conventions which is easy to understandable for every user.

3.3.4 Availability

As it is a standalone mobile, so it will provides all the required functionalities to users 24/7. User does not any specific time or any specific external resources to use its functionalities. She/he can use its functionalities at any time.

3.3.5 Recoverability

As our system is a standalone, so it can violate the Recoverability principle in case of hardware crash. Since app does not have connection to any external server, so in case of hardware crash of user device, it will loses all the information of the system and it cannot be recoverable.

3.3.6 Reliability

Our system is reliable to android system, user can use its all functionality .It will gives effective and accurate responds to every request of the user. The things are displayed in a reliable manners which makes use for user to use it easily.

3.3.7 Error Prevention

System ensure error free services to its user, but in case of error, it use error prevention techniques to give understandable message to its user.

3.3.8 Safety and Security

One of the leading issues in I.T world is the privacy and security. So our system will be dealing with such issues to protect the privacy and security issues. In our case, only those users can use this system, who have correct name and password.

4. Use Case Diagram

"A **use case diagram** is a representation of a user's interaction with the system that shows the relationship between the user and the different **use cases** in which the user is involved.

Actor: Actor is a person / user who is interacting with the system.

*In our case **"School Sectary"** is Actor.

System: The product/software/app is called **system** because Actor is interacting with its different functionalities.

*In our case our **"Tutorial Group App"** will be **system**.

5. Conclusion

At the end, we think this document will help the user to use the product properly and on the other it will be base for the developer team to versioned it to make more better product.