

SOFTWARE REQUIREMENTS SPECIFICATION

for

Time Table Management

Indian Institute Of Information
Technology, Allahabad

Prepared by :-

1. Anshul Agarwal (IWM2017008)
2. Ritik Raj Gupta (IIM2017003)

Submitted to : Dr. Abhishek Vaish,
Assistant Professor

Date : December 3, 2019

Contents

1	Introduction	3
1.1	Purpose	3
1.2	Intended Audience and Reading Suggestions	3
1.3	Project Scope	3
1.4	Project Deadline	4
1.5	References	4
2	Overall Description	5
2.1	Software Process Model	5
2.2	Product Perspective	6
2.3	Product Functions	6
2.4	UML Diagrams	7
2.5	User Classes and Characteristics	8
2.6	Operating Environment	9
2.7	Design and Implementation Constraints	9
3	External Interface Requirements	10
3.1	User Interfaces	10
3.2	Hardware Interfaces	10
3.3	Software Interfaces	11
3.4	Communications Interfaces	12
4	System Features	13
4.1	User specific functionalities	13
4.2	Notification widget	13
4.3	Identity authentication	14
4.4	Check the faculty availability	14
4.5	Semester, Faculty, wise time table	14
5	Other Nonfunctional Requirements	15
5.1	Performance Requirements	15
5.2	Safety Requirements	15
5.3	Security Requirements	15
5.4	Software Quality Attributes	15

1 Introduction

1.1 Purpose

The purpose of this document is describing the project which is an android application for Online Time Table and Announcement. This project aims to update the schedule actively and display sudden announcements without losing time, eliminate the difficulty of updating the currently-used schedule tables, eliminate the risk of papers hung on the door getting damaged, prevent paper waste and ease the communication and interaction between teachers and students. This app also notifies you about the recent updates in the Time Table to its various users. This document includes detailed information about requirement of the project. It reflects the identified constraints and proposed software functionalities.

1.2 Intended Audience and Reading Suggestions

This SRS is for developers, project managers, users and testers. Further the discussion will provide all the internal, external, functional and also non-functional information about this project.

1.3 Project Scope

"Time Table Management" android application creates a space for Director and Teachers for scheduling the classes and providing important announcements without any collisions. Most of the students have difficulty in reaching teachers while they are out of their office, and in practice announcements and time tables of teachers hung on their doors. Although its practical to just have the printout of time table, but over the time it can easily be damaged and also, they are hard to update. Because of these reasons, the interaction between teachers and students may become difficult. In addition to this, at any time, a sudden update on a teacher's time table or condition may appear and it may not reach to the students in time. This project can solve these problems with real time interaction between the teacher and the touchscreen via an android application.

1.4 Project Deadline

Task	Duration
1. Time-table UI	10/10/2019 - 15/10/2019
2. Change Request UI	16/10/2019 - 25/10/2019
3. Firebase/Database	26/10/2019 - 5/11/2019
4. Testing & Enhancement	6/11/2019 - 10/11/2019

1.5 References

- Android studio documentation,
Online - <https://developer.android.com/reference/org/w3c/dom/Document>
Access date- 3/9/2019
- Firebase database documentation,
Online - <https://firebase.google.com/docs/database>
Access date- 3/9/2019
- Faculty related information, course structure,
Online - <https://www.iiita.ac.in/>
Access date- 3/9/2019
- Set up for Firebase real time database,
Online - <https://firebase.google.com/docs/android/setup>
Access date- 4/9/2019

2 Overall Description

2.1 Software Process Model

Initially we thought of implementing classical waterfall model , but then some issues got raised like what we will do if the client sought for some changes to be made. So we changed our mind and finally settled upon implementing iterative waterfall model. It is almost same as the classical waterfall model except some changes are made to increase the efficiency of the software development. The iterative waterfall model provides feedback paths from every phase to its preceding phases, which is the main difference from the classical waterfall model. When errors are detected at some later phase, these feedback paths allow correcting errors committed by programmers during some phase. The feedback paths allow the phase to be reworked in which errors are committed and these changes are reflected in the later phases. But, there is no feedback path to the stage – feasibility study, because once a project has been taken, does not give up the project easily. It is good to detect errors in the same phase in which they are committed. It reduces the effort and time required to correct the errors. In the classical waterfall model, there are no feedback paths, so there is no mechanism for error correction. But in iterative waterfall model feedback path from one phase to its preceding phase allows correcting the errors that are committed and these changes are reflected in the later phases. It may happen that during the project development we get some changes in the timetable of some faculty then we can simply implement them in this model, which would not have been possible in Waterfall model. It may be the case that we need to add or remove some faculty and this is provided by iterative waterfall model which we are using. We started with a simple implementation of a small set of the software requirements and iteratively enhances the evolving versions until our complete project is implemented and ready to be deployed.

Process of Iterative Model :- The process of Iterative Model is cyclic, unlike the more traditional models that focus on a rigorous step-by-step process of development. In this process, once the initial planning is complete, a handful of phases are repeated again and again, with the completion of each cycle incrementally improving and iterating on the software. Other phases of the iterative model are described below:

- **Planning Phase** - This is the first stage of the iterative model, where proper planning is done by the team, which helps them in mapping out the specifications documents, establish software or hardware requirements and generally prepare for the upcoming stages of the cycle. We all sat together and discussed what all tools and software we are going to use in future stages. We will contact the admin of

college for the details of the timetable of all the current semester running in the college as well as data of all the faculty and the courses they are deployed in.

- **Analysis and Design Phase** - Once the planning is complete for the cycle, an analysis is performed to point out the appropriate business logic, database models and to know any other requirements of this particular stage. Moreover, the design stage also occurs in this phase of iterative model, where the technical requirements are established that will be utilized in order to meet the need of analysis stage. We made a proper outline of the app we are about to make, the functionality it will perform, the way these functionality will be implemented. User interface format will be chosen and its designing will be done and we will also plan all the database and storage related attributes, relationship and its dependencies .
- **Implementation Phase** - This is the third and the most important phase of the iterative model. Here, the actual implementation and coding process is executed. All planning, specification, and design documents up to this point are coded and implemented into this initial iteration of the project. We implemented all the planned ideas in the android studio as per the design we made on paper and kept on checking it from time to time among ourselves if it is working fine or not.
- **Testing Phase** - After the current build iteration is coded and implemented, testing is initiated in the cycle to identify and locate any potential bugs or issues that may have been in the software. We will take help from a group of students and faculty who will use the application and will let us know the bugs or the problems they faced. We will then try to fix them and will repeat this process of testing until we get a satisfactory product.
- **Evaluation Phase** - The final phase of the Iterative life cycle is the evaluation phase, where the entire team along with the client, examine the status of the project and validate whether it is as per the suggested requirements.

2.2 Product Perspective

The project was envisioned taking in mind the difficulties faced by the students and faculties of college(s) regarding the access to the time-table .Whether it be semester wise time-table of students or faculty wise time table , access to these are difficult. If there is the case of change in time-table , it required a hectic procedure about which student were notified through mail which are prone to be missed. The project is an android app which brings all these requirements into a minimalistic one click application solving all these problems , Thus , providing faster and better interface. The project centralises it all and helps make these tasks at our fingertips.

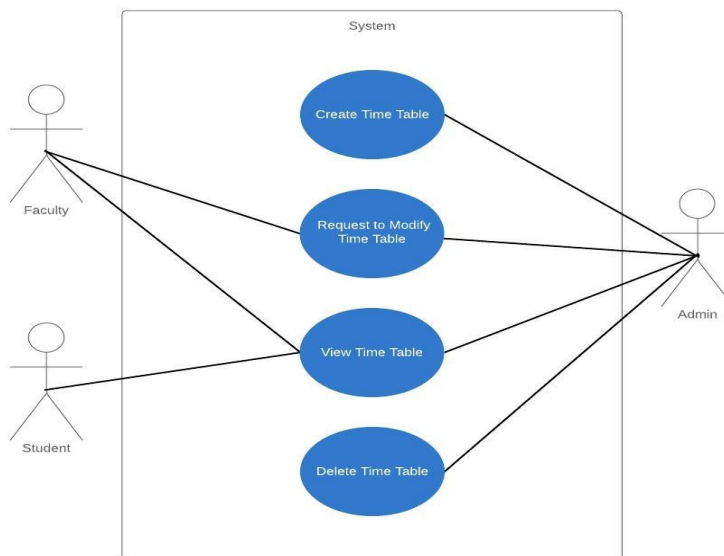
2.3 Product Functions

The project serves the following major functions:

- **Faculty wise timetable(whole)** - Faculty can access time-table as a whole.
- **Faculty wise timetable(semester)** - Faculty can access time-table semester wise.
- **Faculty request to change the slots** - Faculty can request admin to change the slots in case of any clashes with the existing time table.
- **Notification channel** - Student as well as faculties will be notified in case of any changes made.
- **Student view (semester wise time-table)** - Student can only view the current time-table of all the semester.
- **Student view (faculty wise time-table)** - Students can even view the time-table of the faculty so that they can get the idea of their free slots.
- **Admin role** - The admin will be notified for the requests sent by the faculties, and only he/she will be having the access rights to edit the database(the current time-table).

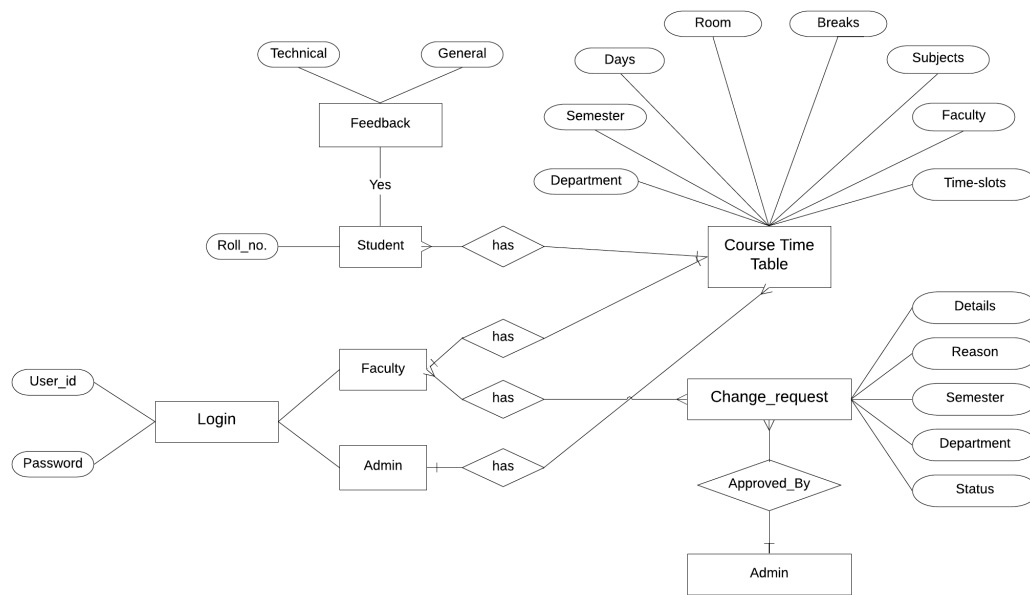
2.4 UML Diagrams

1. Use Case Diagram :-



2. ER Diagram :- The ER diagram constitutes a technique for representing the logical structure of a database in a pictorial manner. This analysis is then used to organize data as a relation, normalizing relation and finally obtaining a relation database.

- **ATTRIBUTES** - Attributes are the properties which define the entity type.
- **RELATIONSHIPS** - It connect entities and represent meaningful dependencies between them.
- **NORMALITY** - Normalization is the process of organizing the data in the database. The basic objective of normalization is to reduce redundancy from a relation or set of relations, which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of the data stored. Thus, the tables are designed in such a way to reduce redundancy.



2.5 User Classes and Characteristics

The user classes for this project are:

- **Students** - Students can see their time-table as well as can look through time-table of various faculties to check their availability. They can also send feedback or raise issue about any change in table or working of the app.
- **Faculties** - They can also look through the time-table of students as well as any other faculties in the campus. Additionally they can send request to admin about any changes in the time-table.
- **Admin** - Reviews request of the faculties and if possible update the time-table database with the required changes.

2.6 Operating Environment

The product is currently designed for various versions of Android. The database required for the app to store time-table entries and requests is done on Google firebase cloud platform.

2.7 Design and Implementation Constraints

1. For user identification and implementing individual usage, every user has to login into the app using college email-id. On the basis of login details the app will identify the type of user it is and let them to their respective window.
2. For seeing all different types of time-table , app provides list of options such as section , branch and semester for students' one and faculty name for faculties. This feature will be available in all type of users window.
3. Their will be a separate window for editing time-table for faculties. They have to similarly choose the time-table they want to change , then submit the changes for review.
4. Admin will have a window which will show all the requests which he/she goes through and can submit the change if accepted . This change will be reflected in cloud database platform instantly then in app within seconds.
5. The changes made then are notified to every users through notification widget of the app upon internet connection.

3 External Interface Requirements

3.1 User Interfaces

- Front-end software: Android studio
- Back-end software: Firebase cloud database

Android studio :-

- **Screen layout constraints :** Screen layout constraints will be adjusted according to the screen size of the phone. For example, the app will automatically adjust itself according to the screen size of the user's device. Size of the buttons, text field, widgets will be all relative.
- **Rotation of devices :** Users will get the perfect output screen while rotating the device by 180 degrees. Scrolling activity will also be introduced, to make sure that the whole information will be displayed in an easy scrolling manner. Back arrow will help to go to the previous activity.
- **Theme and color scheme :** Also, the theme color and design will be chosen according to a decent view.

Firebase cloud database :- Firebase will store and retrieve the data. The user (only faculty) can send a request to the admin for a change in the time table. Others will have view access to the respective database. Also, a notification widget will be there to notify the user.

3.2 Hardware Interfaces

- An Android phone

An Android phone :-

- **Sufficient memory :** The application will be installed on the android phone. The mobile must contain sufficient memory space to download and install the application.
- **Resistive apps should not be there :** It should not previously contain any resisting application which will resist the installation of our application.

- **Fit to any screen size :** There a wide array of screen sizes, hardware specifications and configurations because of intense competition in mobile software and changes within each of the platforms (although these issues can be overcome with mobile device detection).
- **Touch sensors :** The touch quality of the mobile should be proper otherwise, it may lead to improper timings for the opening and closing of the activities.
- **Hardware permissions :** User should grant all the hardware related permissions, for the smooth application.

3.3 Software Interfaces

- Android studio
- Firebase cloud Database
- Libraries and Dependencies

Android studio :- Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems.

Firebase cloud database :- It is a cloud computing service model using which the mobile app developers can connect their application with the backend cloud storage. It will be used as an interface to store and retrieve the data. It will store all the semester wise subject, faculties wise subjects, time slots for each faculty as well as for the subject. By using queries, we are pushing and retrieving the data from the cloud. Different people will have different accessing rights for the database respectively. Here admin will only have the editing rights, rest can only view the information.

It's a real-time database, along with the information related to the number of regular users, views per day, views within a month, crash number of users, technical feedback, server notifications. Also, firebase is having plenty of security rules to secure the data in cloud firebase. Firebase security rules stand between data and malicious users. By using these rules, it protects the app's data to the level of granularity that an application requires.

Libraries and Dependencies :- Libraries and dependencies are used in the manifest.XML file, to get the desired data values from them. They will be added according to the features to be implemented as well as the lowest Android version to be used. With the library and dependencies, we can perform actions/functions faster and in a more effective way.

Example:- Google play services dependency is used to keep the apps updated, ZXingView Scanner dependency is used to scan the documents, etc.

3.4 Communications Interfaces

- Notification widget
- Communication security
- Data synchronization

Notification widget :- On the request, faculties may send a message to the admin about the time-table related query. Admin having the database editing rights can edit the database.

For the same, there will be a notification widget which will notify both the sender and receiver about the query, either the query is accepted or rejected.

Communication security :- While submitting the query, the user id will be taken from the application for the communication security issues.

Data synchronization :- The application will be synchronized in such a way that consistency among the data from a mobile source device to a target data storage service will be maintained properly.

4 System Features

An android application which will provide easy access to the students to view their time table and faculty wise time-table to check their availability. The purpose of this project is to implement the features so as to increase user comfort and reliability. This app also notifies you about recent updates in the timetable to its various users. It will be having three types of users, i.e. Admin, Student, and Faculty. Also, the faculty will have a digital way to communicate with the admin, which not only saves time but also make the process more efficient. It will also inform the students about the changes or updates made in the time-table through notification.

4.1 User specific functionalities

This android application is designed in such a way that it provides appropriate access of data to a particular user. It is basically designed for three types of users.

- **Student** - It will provide students access to the time table of all the current semester. They will also have access to the faculty time table so that they are well aware of their free slots to catch them up in when in need.
- **Faculty** - They will have access to both the time table of the student's all semester as well as all faculties time table . They will also have an option to request admin for the changes they want to make in the schedule when required.
- **Admin** - They are the ones who are appointed by the college administration to govern over the academics calendar. So only they are the ones who have access to the database. They make the time table for all the semesters and give view-only access to the students as well as faculties. When faculty ask for some changes they do consider their request and respond accordingly.

4.2 Notification widget

Users may miss important updates . So this android application provides them with the notification window so that they don't miss them up.

- **Admin** - Whenever any faculty requests for any change then admin will be notified about the request made. They will act accordingly and will either accept the request or will decline it.

- **Faculty** - They will get notified about the action taken by the admin on their request. And they will also be informed about the usual changes made in the schedule.
- **Student** - They will only get a notification when some changes are made in the time table either temporary or permanent.

4.3 Identity authentication

- **Faculty** - Faculty will be having their own user-Id and password so that admin knows who has made a request for some changes in the timetable. This login feature for the faculty makes this application more safe and secure. As no one else will be able to ask for any change in the time table on someone's behalf. Thus the admin will have proper information about the person asking for the changes in the schedule.
- **Admin** - Login user-Id and password for admin is the most important feature as we can't give access to the database to anyone except the appointed admin by the administration.
- **Student** - As students will only have access to view all the time tables. So no requirement of any security as such.

4.4 Check the faculty availability

All students as well as faculty will have access to all the time table. This will help students to search for free slots of their respective faculty so that they can contact them if required. It will also be easier for the faculty to see if some slots available in student's schedule for some extra class when required. Thus it will make the whole process efficient and less tiring.

4.5 Semester, Faculty, wise time table

All the database related to the timetable is available to everyone. So that they can check each other's free slot and everything they wish to check.

5 Other Nonfunctional Requirements

5.1 Performance Requirements

The response time must be quick to extend convenience. Too, the associations between database, application and servers must be smooth without any idleness to avoid the misfortune of information and keep the stream of utilization going.

5.2 Safety Requirements

The database will not crash, or will not overload at any period of time. Managing the storage limit, along with the access rights will provide a safe environment for the product. It will be free from any kind of virus so that we will ensure the privacy of the user's data.

5.3 Security Requirements

In terms of security, database access is the main criteria. Identity authentication is important in terms of security. Here, only faculties have the right to send a request to the admin, no one other than the faculty can send. Hence ensuring the security criteria of the product. Technical issues will be solved by the developer team, only they can access the source code.

5.4 Software Quality Attributes

- **Availability** : It is designed for a campus's perspective. All the college students, as well as faculty, will get access to the product.
- **Adaptability** : It will be easy to adapt with a basic user interface.
- **Maintainability** : Low maintenance cost. User has to update the app after a certain period of time interval requiring a less amount of data.
- **Correctness** : It will feed the data to the database correctly, as well as it will also ensure that the right person will get the right details.
- **Reusability** : Code will be written in such a manner so that it can be used further for other similar projects.
- **Testing** : Before launching, the product will be tested by college students only.

- **Reliability :** It is safe to use the product under the privacy policy thus the users can easily rely on the security system of the product. Also, it will be free from any clashes either in terms of application or any server or database breakdown.

With all of these quality attributes, the software will ensure a good quality approach.