

SOFTWARE REQUIREMENT SPECIFICATION(SRS)

TUTLIFY (CROSS- PLATFORM APPLICATION)



**Dr. A.P.J. Abdul Kalam Tech. University
Uttar Pradesh**

Project Guide:
Miss. Kamle

Submitted By:

Niteesh dubey (CSE)-1604610055
Athar sameen khan (CSE)-1604610033
Mayank mani (CSE)-1604610049
Mayank Animesh (CSE)-1604610048

1) Introduction:

1.1 Purpose:

The purpose of this document is to present a detailed description of the Tutlify(Cross-Platform App). It will explain the purpose and features of the cross-platform app, the interfaces of the App, what the App will do, the constraints under which it must operate and how the App will react to users needs. This document is intended for both the stakeholders and the developers of the system.

1.2. Scope:

An cross-platform app will be applicable everywhere, and any one can use it 'who wants to learn and grow there knowledge'. It will be more efficient and easier way to have a content on the apps. As all the content will be present in a centralized way, user can learn, earn and grow by performing practices given by the app.

1.3. Definitions and Abbreviations:

Following are the definitions for the jargoned words.

TERMS	DEFINITION
No SQL	Non Structure query language for the database purposes.Used to define procedures to store and retrieve data.
User	A person who use this app in order to get all the Services, related to the content that want, main moto Of this app is Learn, Earn and Grow.
Database	Area or Platform where we stores the data, which is used by the App(Tutlify).
Firebase	Firebase gives you the tool to grow your app and Monetize yourself, it helps to run your app in each platform.

Google API	Google API is the platform created by the google As this platform use to authenticate the users before Accessing the content of the app(tutlify)
Vs Code	Vs code is the software which is used to, develop the Cross-platform app, this software uses the plugins And extensions to compile and interpret the app to check the output
Android and Iphone	These system used to access the app, to use that app, which will be used as clients to access the contents provided by the app.
Tutlify Features	All the benefits and characteristics that app provide. These features will be explained to the New users inside the app.
Administrator	A person that will be responsible for the addition and deletion of the content from the general database of the system..
SRS	A document that completely describes all of the functions of a proposed of cross-platform app and the constraints under which it must operate. For example, this document.
Stake Holders	Any person with an interest in the project who is not a developer.

1.4. References:

This cross-platform app has been prepared on the basis of discussion with Team members, faculty members and also taken information from following books & website.

1.4.1. Websites:

1.4.1.1. www.google.com

1.4.1.2. www.wikipedia.org

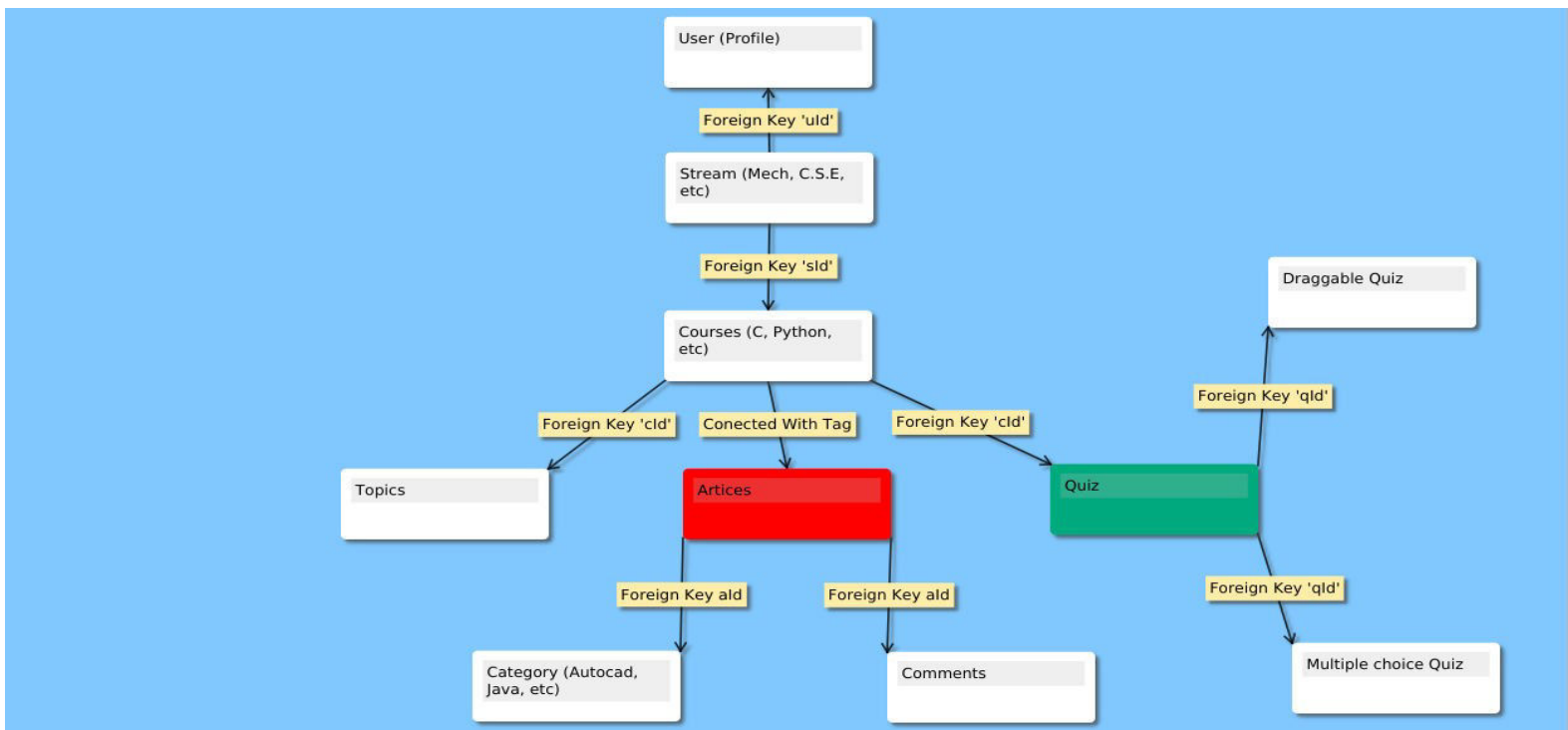
1.4.2. Documentation:

1.4.2.1. flutter.dev

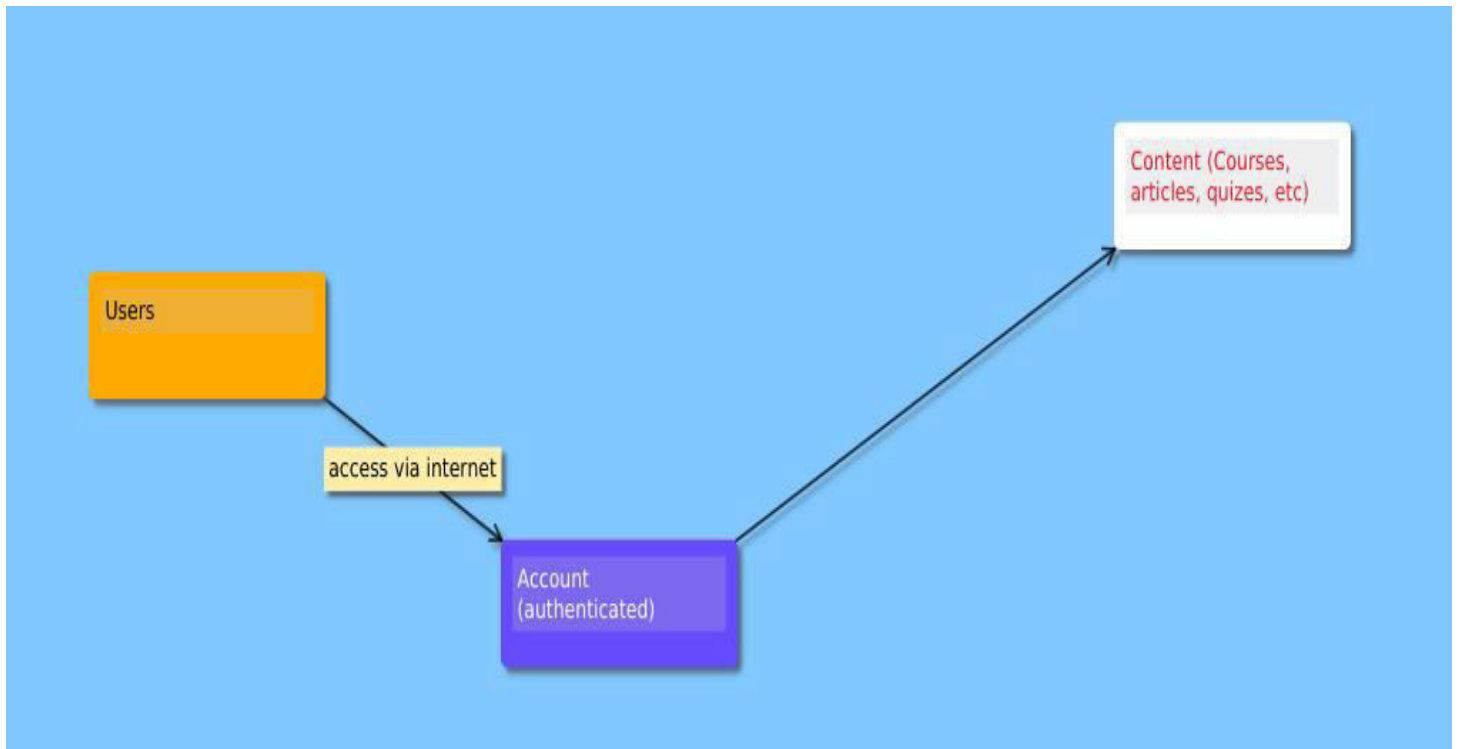
2) General Descriptions:

2.1. Product Perspective:

Following is the arrangement of the screen connected to each other and transferring the data and showing data to each screen from database by the help of foreign keys.



In earlier apps, user should have to download different apps for each different streams(mech, C.S.E, etc) in the mobile. It wastes time.



After implementing the Tutlify(cross-platform app) user will be able to connect to his account to access all the content from the app which is available in one single platform.

2.2. Functionalities:

This application will have following functionalities:

2.2.1. Provide Different Streams:

Users will be able to access all the streams from the single application by just clicking the selected streams provided in the app.

2.2.2. Object Scanner (Tutlify eye):

This is the features provided by the application in which a given application identifies the object and give the related information, related to that object.

2.2.3. Quizes:

A given application provides the quizzes to check the ability of the user and rank them according to their scores.

2.2.4. Notify You About the topics to learn:

Provides the notification based on the daily performance of the user and gives them a suggestion which topics to cover and provides the daily articles.

2.2.5. Competet with your friends to earn more badges :

Provides the feature to connect with your friend by using a device hotspot or internet and competet with each other by playing some the games and quizzes to earn the badges.

2.3 User Characteristics:

There are various kinds of users for the application. Usually cross-platform application is able to run on both devices android and iphone and they are visited by various users for different reasons. The users include :

2.3.1. The users who want to earn some money from the app by growing their knowledge which affects their growth.

2.3.2. Users which create the content by posting their blogs so that everyone can see their useful content and rate them , through the level of quality of the articles.

2.4 Generals Constraints:

Some general constraints should be defined which will have a great part in the overall succession of the Tutlify application.

2.4.1. Hardware Requirements:

As this system is an cross-platform application so a client server will be the most suitable Organizational style for this system. Android or Iphone systems will be needed by each of the actor as well as that user must be connected to the internet. So, concisely following hardware will be needed.

- 1) Android or Iphone systems
- 2) Internet availability

2.4.2. Safety and Security:

This Project application is safe and secure from hackers as it provide the highly authentication and the google API to register and login the app, to get inside the app content so the given application is safe and secure.

2.5 Assumptions and Dependencies:

Following are the assumptions and dependencies which are related to this online banking project.

- 1) This project is a stand-alone project so it will not affect the system where it will be embedded.
- 2) This project is a cross-platform project this mean that this project is easily runnable on any platfrom whethere it is android or iphone it can be easily deploy on any devices.

- 3) This system will not depend on any other module. As it uses the flutter framework for development of the application.
- 4) It is will not affect the environment at all.
- 5) Users will feel free to adopt it because it will be free and easy to use.
- 6) As this project contains valuable content and new features so it will probably remove the previous apps present in the app stores.

3) Specific Requirements:

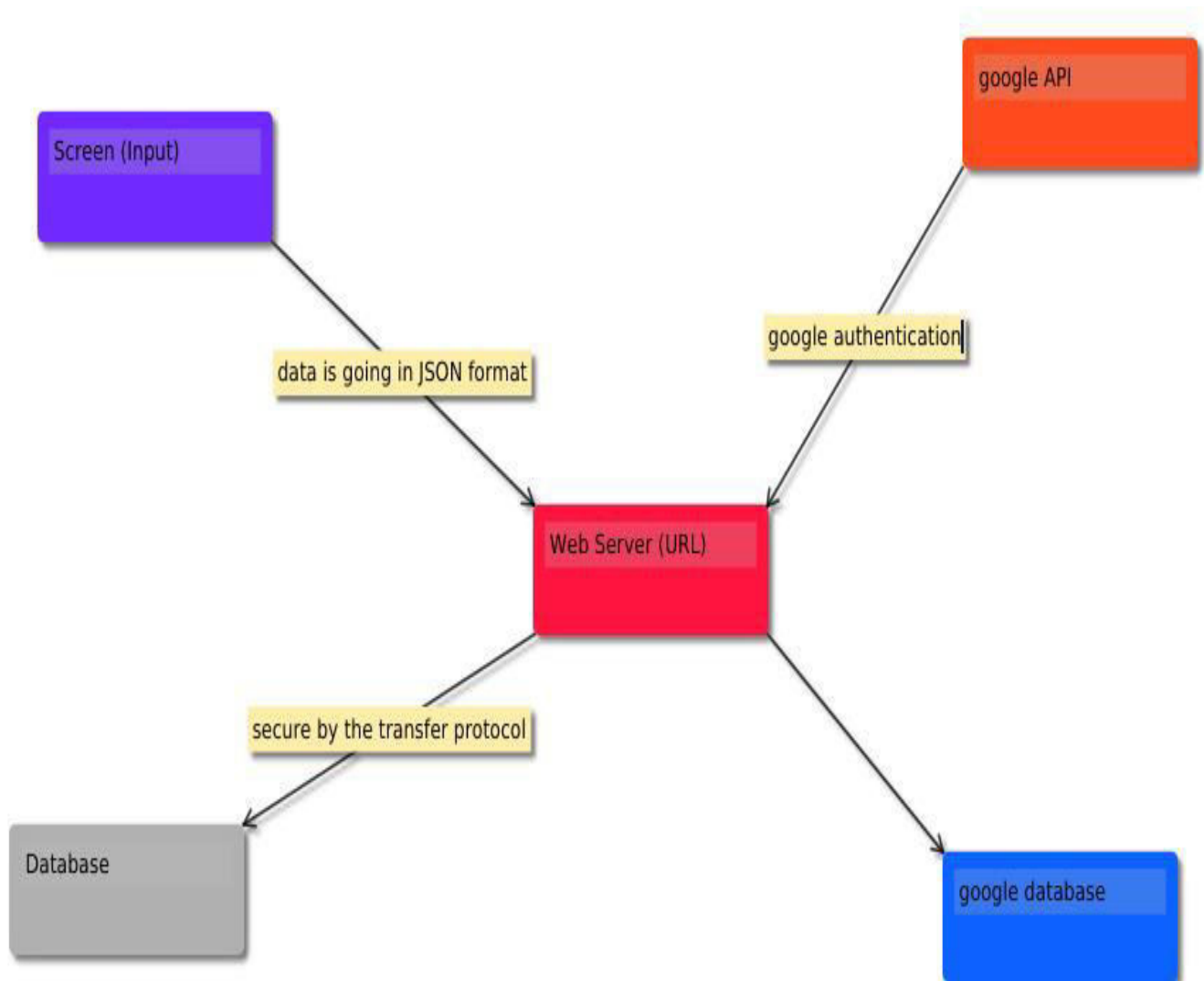
How the Tutlify application will interact with the environment, what will be the functional and non-functional requirement. These all the steps should be defined here for providing a powerful base to the design phase. The design of the project will completely depend on the functional and non-functional requirements. So these should be defined clearly and accurately for the effectiveness.

3.1 Functional Requirements:

Following are the services which this application will provide. These are the facilities and functions required by the users.

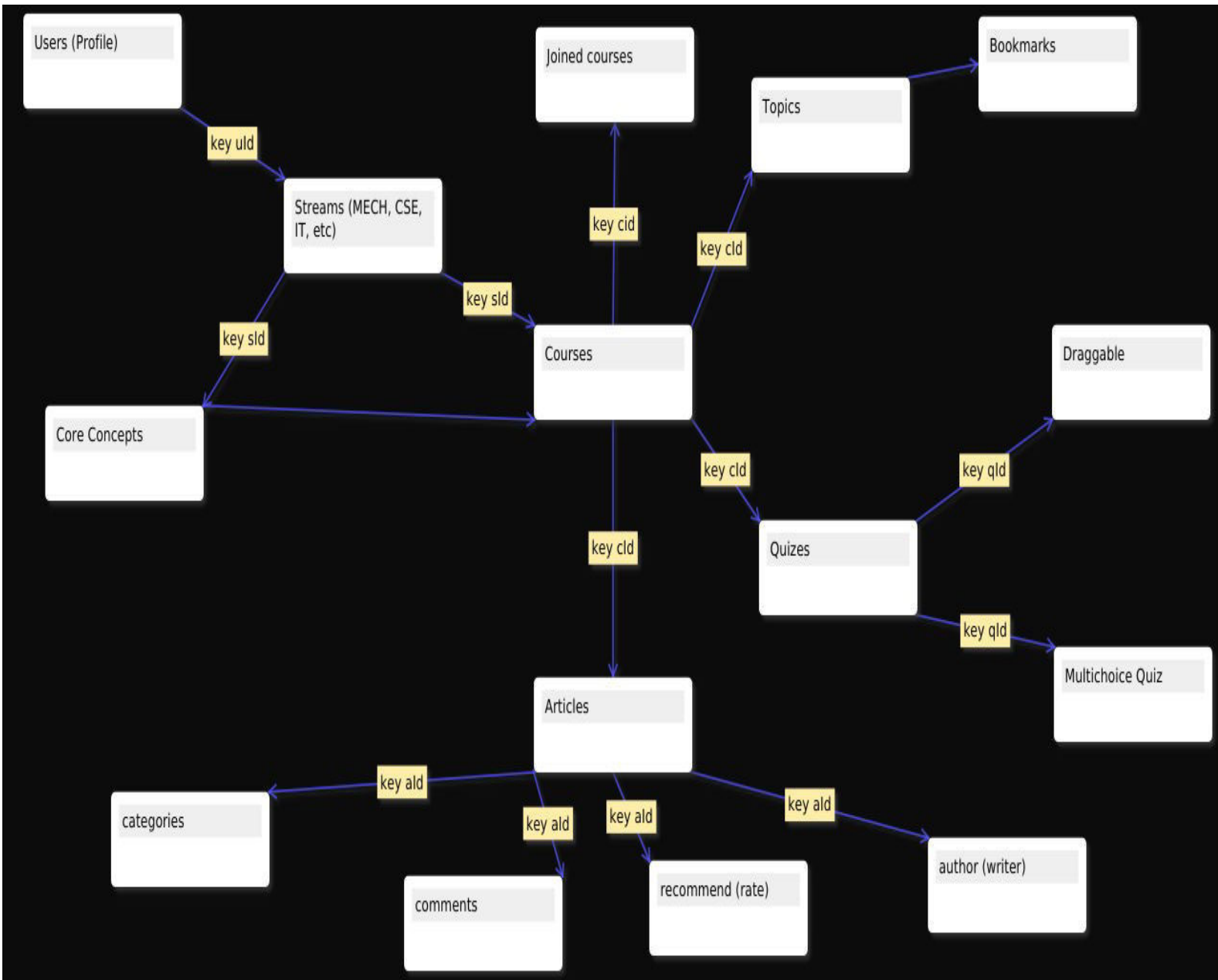
- a) Check the post.
- b) Check your current scores and performance.
- c) Create your own post or blog or articles.
- d) Update your blog or post.
- e) Check your blog comments.
- f) Check number of joined courses.

3.1.1) Context Diagram and extent list:



3.1.2) Dataflow Diagram:

Following is the data flow diagram for the online banking system.



3.1.3) Process Specification:

All the process mentioned in the DFD are described as below.

Users (Profile):

Each Users will have its account Id and password. This screen will require both of these attributes for them to access their account.

Streams:

Users can select there own streams to check there courses and all the content they related to that streams.

Courses:

Courses is provided for all differents streams selected by the users.

Joined Courses:

Number of courses you have joined in the given selected streams.

Bookmarks:

Set the list of all the bookmarks of the courses content in the given joined courses.

Articles:

It provides you a number of new articles based on the categories and related to the selected categories.

Quizes:

Provides the number of quizzes like drag and drop quiz and options related quiz and much more, competet to other users and win prizes.

Keys:

These are just the foreign keys that used to connect the different lists and datas of the different screens and there aragment in the given application.

3.2) External Interface Requirements:

These requirements are discussed under the following catagerisation.

3.2.1. User interface:

Application will be accessed through a android and iphone operating system . The interface would be viewed best using 1024 x 768 and 800 x 600 pixels resolution setting. The software would be fully compatible with all the versions of android and iphone and above.

No user would be able to access any part of the application without logging on to the system.

3.2.2. Hardware Interface:

3.2.2.1. Client side:

- a) Operating System: Android(5.0 <), Iphone.
- b) Processor: Pentium 3.0 GHz or higher.
- c) RAM: should not be less than 500mb.
- d) Hard Drive: not required.

3.2.2.2. Server side:

- a) In Server side it use the firebase provided by the google and renders the data from the cloud using no sql and JSON format via URL.

3.2.3. Software Interface:

3.2.3.1 Client Side:

Flutter , Dart.

3.2.3.2. Web Server:

Flutter , Dart, firebase, python, sql.

3.2.4. Communication Interface:

The Customer must connect to the Internet to access the Application :

- a) Dialup Modem of 52 kbps.
- b) Broadband Internet.
- c) Dialup or Broadband Connection with a Internet Provider.

3.3) Non-Functional Requirements:

Those requirements which are not the functionalities of a system but are the characteristics of a system are called the non-functionalities. Every software system has some non-functionalities. Just fulfilling the requirements of the user is not a good task, keeping the system accurate, easy to maintain, reliable and secure is also a basic part of software engineering. Online Banking System must have the following non-functional requirements so that it could be said as a complete system.

a) *Conformance to specific standards:*

b) *Performance constraints:*

This system must be fit according to the performance wise. It should use less memory and will be easily accessible by the user. Memory management should be done wisely so that none of the memory part goes wasted.

c) *Hardware limitations:*

It should be designed in such a way that cheap hardware must be installed to access and use it effectively. It should be platform independent. There should be no hardware limitations. It should be designed to work with the low specification hardware so that it could easily work with the high specification hardware.

d) *Maintainable:*

Each of the modules should be designed in such a way that a new module can easily be integrated with it.

e) *Reliable:*

f) *Testable:*

3.4) Other Requirements:

Software Quality Attributes:

The Quality of the System is maintained in such a way so that it can be very user friendly to all the users. The software quality attributes are assumed as under:

- a) Accurate and hence reliable.
- b) Secured.
- c) Fast speed.
- d) Compatibility.

4) Possible Product Evolution:

Not even one system maintains its stability for a very long period. Every system requires evolution according to the time and fashion introduced in the market as well as due to lot of competition companies have to change their system to provide more features to their customers to compete the society.

Following are some perspectives according to which this system can be maintained in the future. These are key points according to which it would need a great evolution soon.

4.1 Credit Card Management:

Credit cards are the key feature for the online shopping. These cards provide the easiest way to shop almost at all well known shopping malls and many other places. As it isn't so popular in the environment where this system is going to be installed but it might start working soon when this fashion (Requirement) will be needed. So it is the basic evolution which might be necessary soon.

4.2 Interface evolution:

As the user interface created by the software designers will be good looking and easy to use but according to the fashion and time, selection of colors usually change person to person. Style of the system will become old and it will surely need evolution to provide a new and cool look to the users.

4.3 Technology Evolution:

This system is going to be designed by using Flutter and Firebase. As these languages provide much security in the current situation but According to the most security Issues these languages might crash or slow down in the future and at that time this system might be replaced by other new technology that will better than flutter which provide more security to the users data and make the content easily accessible.

5) Index:

5.1. Introduction.

5.1.1. Purpose.

5.1.2. Scope.

5.1.3. Definition & Abbreviation.

5.1.4. References.

5.2. General Description.

5.2.1. Product Perspective.

5.2.2. Functionalities.

5.2.3. User Characteristics.

5.2.4. General Constraints.

5.2.5. Assumptions & Dependencies.

5.3. Specific Requirements.

5.3.1. Functional Requirements.

5.3.1.1. Context Diagram Extent List.

5.3.1.2. Data Flow Diagram.

5.3.1.3. Process Specification.

5.3.2. External Interface Requirements.

5.3.2.1. User Interface.

5.3.2.2. Hardware Interface.

5.3.2.3. Software Interface.

5.3.2.4. Communication Interface.

5.3.3. Non Functional Requirements.

5.3.4. Other Requirements.

Software Quality Attributes.

5.4. Possible Product Evolution.

5.4.1. Credit Card Management.

5.4.2. Interface Evolution.

5.4.3. Technological Evolution.