



MANIPAL UNIVERSITY

SOFTWARE REQUIREMENTS SPECIFICATION

For

CARDIGRAM

Version 1.0

Prepared by

<i>NAME</i>	<i>REG_NO</i>	<i>E-MAIL ID</i>
Aman Chopra	140911358	amanchopra64@gmail.com
Rishabh Drolia	140911316	drolia.rishabh94@gmail.com
Suvimal Yashraj	140911394	suvi.yashraj@gmail.com

Submitted to:

Instructor	Ms.Sumana , Mrs. Anuradha Rao, Ms. Vibha
Course	Advanced Technology (AT Lab)
Lab Section	IT-A2
Date	12-Mar-2017

Contents

REVISIONS.....	II
1 INTRODUCTION	1
1.1 DOCUMENT PURPOSE	1
1.2 PRODUCT SCOPE	1
1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW	1
1.4 REFERENCES AND LIBRARIES USED.....	ERROR! BOOKMARK NOT DEFINED.
2 OVERALL DESCRIPTION	3
2.1 PRODUCT PERSPECTIVE.....	3
2.2 PRODUCT FUNCTIONALITY	4
2.3 USERS AND CHARACTERISTICS	4
2.4 OPERATING ENVIRONMENT	4
2.5 DESIGN AND IMPLEMENTATION CONSTRAINTS	5
2.6 USER DOCUMENTATION.....	5
2.7 ASSUMPTIONS AND DEPENDENCIES	5
3 SPECIFIC REQUIREMENTS.....	6
3.1 EXTERNAL INTERFACE REQUIREMENTS	6
3.2 FUNCTIONAL REQUIREMENTS.....	8
3.3 BEHAVIOUR REQUIREMENTS	9
4 OTHER NON-FUNCTIONAL REQUIREMENTS.....	11
4.1 PERFORMANCE REQUIREMENTS	11
4.2 SAFETY AND SECURITY REQUIREMENTS.....	11
4.3 SOFTWARE QUALITY ATTRIBUTES.....	11
.....	

1 Introduction

With the country turning smarter every day, it's only fair that its citizen turn even smarter. In the time where carrying documents has become almost obsolete, when it comes to carrying prescriptions it becomes even more annoying. "CARDIGRAM" aims to give its users a wide range of functionality in the medi world, making it their most utilitic companion. The platform conferences between the doctors and their patients from the very start to ensure full time assist for those in need. The main objective of "CARDIGRAM" is to reduce the communication distance in the medi world where even the slightest of seconds count.

1.1 Document Purpose

The intended readers of this document are the developers of the system, the testers and the consumers or clients that will use the tool on a day to day basis. Any changes incorporated in the document will be added in the last section for the reference of the developers and its users.

1.2 Product Scope

The client will get the following functionalities from the product:

- In hand portable database of patients for the reference of a large database of doctors
- Personalized hub for each patient with records of his/her visitations and all the prescriptions and reports.
- Setting up appointments with doctors.
- Scalable charts and graph representation for test values of the patients to facilitate easier knowledge outlook and more robust overview.
- Real time patient-doctor, doctor-doctor, group chat.

1.3 Intended Audience and Document Overview

The set of stakeholders include the team members, the overseeing instructors and potential user of the tool.

1. Team members can use this document for detailed understanding of the requirements that are needed to be met by the application from both user centric and design centric viewpoints.
2. Instructors can use this document to access the level of detail the group is working from to guide development.
3. The document can be used as an agreement form for the tool being developed.

1.4 References and Libraries Used

- Android Design Library 25.1.1
- Firebase Realtime Database
- Firebase Cloud Storage
- Firebase Authentication
- Firebase Analytics
- MP Android Charts
- Glide and Picasso For Image Rendering
- Google Places API

References:

- www.stackoverflow.com
- <https://firebase.google.com>
- Youtube Channels : Firebase, Simplified Coding, Firecosts for Firebase Developers

2 Overall Description

2.1 Product Perspective

This product is an extension to already existing primitive Hospital Management Systems in the market. It is not a component of a larger system. The highlights added in the product are the first of its types to have been included in an integrated platform.

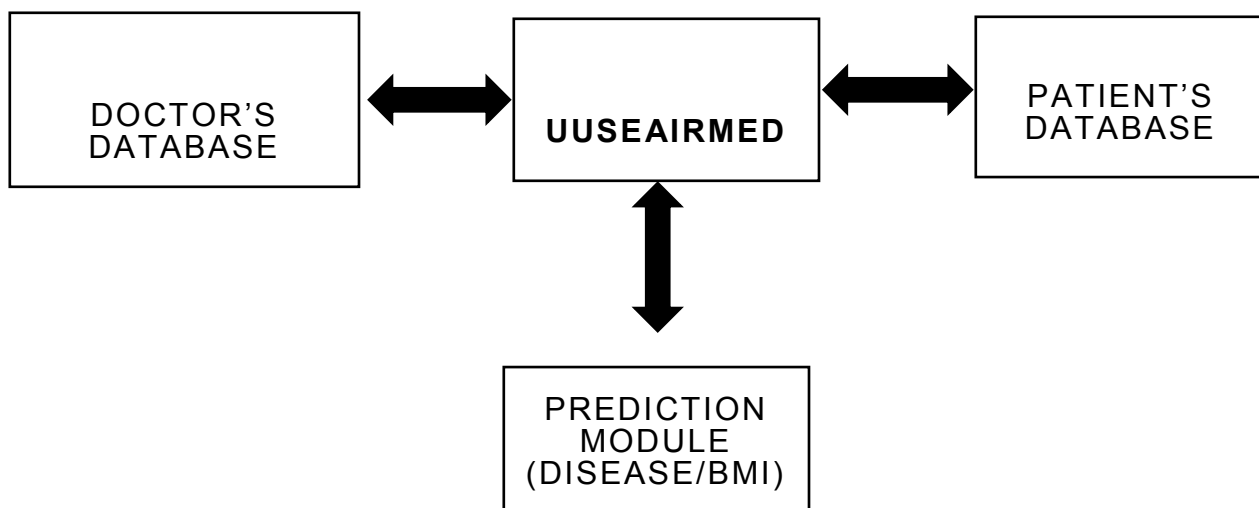
The idea behind the product is to build a tool that helps accessing patient's documentations on the go along with creating a one stop platform for the doctors and the patients to create an automated world for themselves which is always in there touch space. The product will help the hospitals, clinics and working professionals in this field stay connected with their patients and also the patients to stay updated about their physical well being without misinterpreting the data they usually find in their reports and prescriptions

The product with its existing features and most importantly the vast database on the cloud i.e Firebase will be able to revolutionize the medi world, with its users asking for nothing less. After its completion; your prescriptions, reoprts, doctors, References and others will just be a click away on your phone. So no more hogging on internet for hours to find the best possible treatment for your disease, with "Cardigram" finding yourself the best ever you deserve will never be easier.

Cardigram uses cloud storage and server space from Google's Firebase which intern facilitates the chat module integrated within. The system will also interact with the profile of each individual user and give information on the recent activities performed using the tool and the past records.

It will also give user a statistical and subjective analysis of the reports uploaded in the patient's personalized dashboard in the form of scalable charts which will help in providing a more robust overview of the reports to its owner.

Cardigram is very user-friendly, 'quick to understand' and reliable appliaction for the above purpose and it's simple.



2.2 Product Functionality

- The user can sign up on Cardigram using his/her name, email, password and display picture.
- Email verification has been added as means of verifying the user which would be causal to registering the user onto Cardigram.
- The user can either login by giving their email and password.
- The users can upload their medical data i.e. reports, prescriptions etc onto their profile which would in turn be available on the go using google Firebase cloud storage.
- Patients can use the application to set appointment with doctors which would be shown as an upcoming event in their dashboard.
- The users can use the integrated chat module to chat with doctors, patients in an individual chat or a group chat which is an amalgamation of all the patients registered under Cardigram's database. The chat module uses google's Firebase cloud server to facilitate the chat servers.

2.3 Users and Characteristics

The users of the product will be ones in the medical industry along with the patients. A registered user would be able to create an account which will contain the information about his/her visitations, the prescriptions, lab reports, medicines. Along with this the registered users can take advantage of all the product functionalities mentioned in clause 2.2.

The application is made for the general mass, keeping simplicity in mind. The user doesn't need to have any prior technical knowledge for operating the application. The user has to sign up for the application by providing their credentials and he/she is good to go. The user credentials are encrypted and stored on the remote database securely. The layout of the application is very simple and follows the WYSIWYG paradigm.

Since it is reasonable to assume that the potential user has basic knowledge about the working of an online commercial service, we assume that our users will already be informed about basic functionality of the product. Also clear documentation and tutorials about the product feature will be provided.

2.4 Operating Environment

The application is designed for the Android operating system and supports all the android devices with minimum SDK version 15. Thus, it supports all the android versions from 'Ice Cream Sandwich' to 'Nougat'. It requires a minimum of around 100 MB of storage on the disk drive after installation. The app functions smoothly on any dual core processor device with a minimum of 1GB RAM and require a constant high speed internet connection, as

the data is retrieved from the cloud. For the app to run properly, the mobile device should have GPS capabilities, as it retrieves the location of the hospital from a map view.

The program uses a custom made back-end for getting requested information about the patients, doctors etc. from its database stored on the cloud.

2.5 Design and Implementation Constraint

The application is cloud based, so it requires a constant high speed internet connection for its proper functioning. This is one of the major limiting factors of the application. This also limits the application from working in offline mode. Even, if the internet connection is slow, the application won't function properly and would take longer time to retrieve data from the database.

2.6 User Documentation

At the time of uploading of the application onto the store a specific writeup would be provided acting as a user manual for better understanding of the application. Cardigram will have a user friendly GUI, and the user will easily be able to use this application, and understand each option and navigate around.

2.7 Assumptions and Dependencies

The application uses Google Firebase (PaaS) as the backend service provider. Two of the Firebase modules are used in the application, namely, the authentication module, and the real time database module. The authentication module is used for storing the user's credentials in an encrypted format in the cloud. The real time database is used to store the user profile details, event details and the group messages of an event. The application also uses an external font library for custom fonts.

The success of the system will depend on:

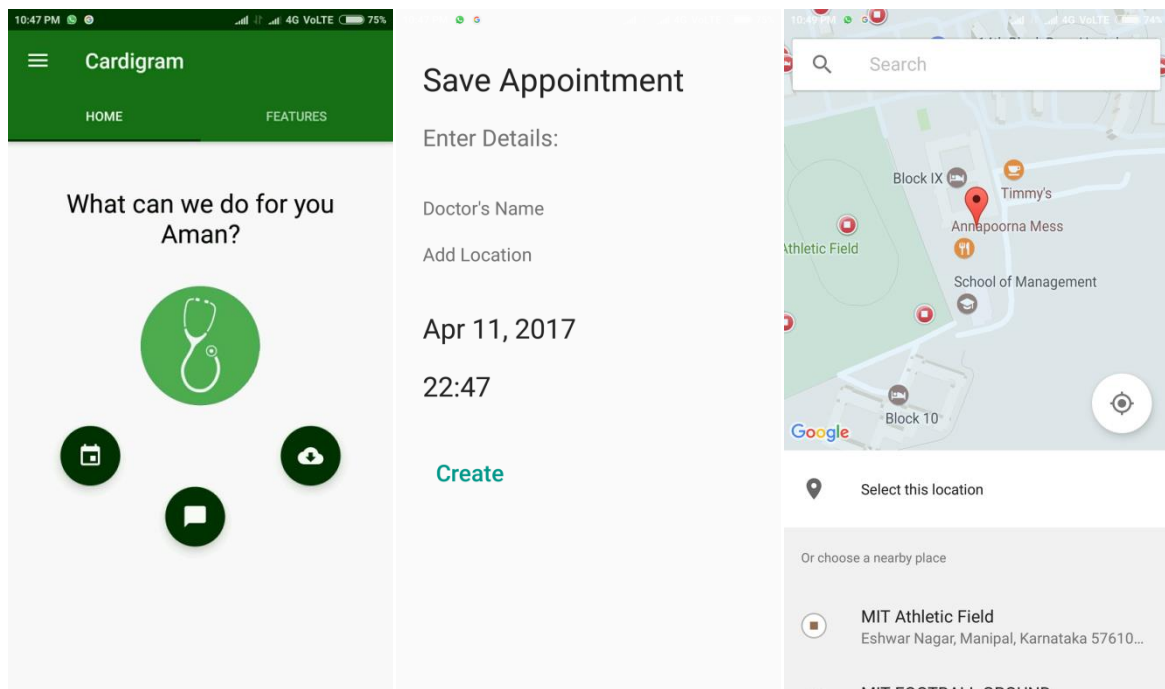
- Existence of internet service.
- Users must be comfortable with android apps and have enough conation to work with the application.

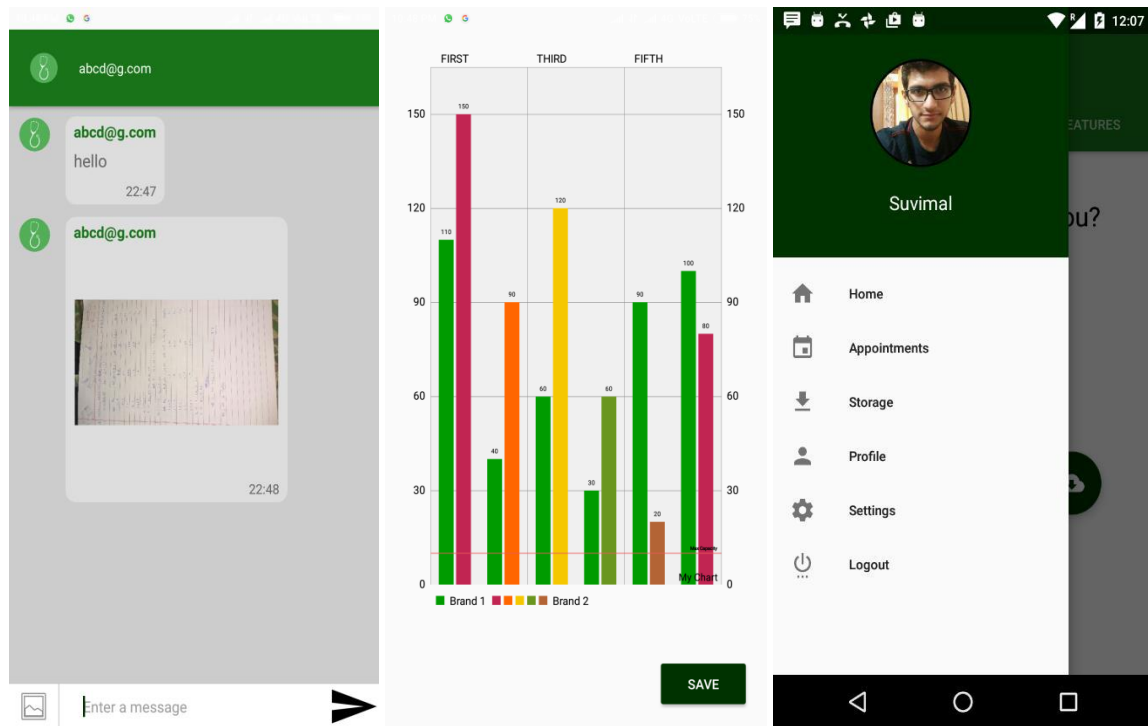
3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

- The application shall start with a splash activity which waits for the app to load and will prompt user for confirming all the main operations like termination, invalid log-in, etc.
- The UI/UX incorporates shared preferences which saves the state of the user and help avoid the same user to log in again and again.
- It will follow the basic GUI and error display standards.
- The first start includes an introduction to the application and tells the user what the application is meant for. The UI is consistent across the application.





3.1.2 Hardware Interfaces

The hardware requirements include a compatible Android device running Android Lollipop or above. There are no constraints on the screen size. However, it is required to have 1GB or above of RAM for smooth functioning of the application. The application requires constant access to the internet since the data gets updated in real time.

3.1.3 Software Interfaces

The product is designed for Android operating system for smart phones. It uses the standard libraries provided by Google for Android. We have additionally used a library for introduction cards and fonts. The messages transferred through and forth the product are data items pertaining to the details of users and events and the chat messages between users. We have used the API provided by Firebase for communication between client and server. No data is shared between other applications of the system. The Glide and Picasso Library has been used for rendering the images uploaded by the users. The MP Android chart library facilitates the generation of charts and graphs in the application in the analysis module of the patients reports.

3.1.4 Communications Interfaces

The product requires constant access to the internet, since it access the backend database - Firebase. The data is updated in real time. The data is transferred encrypted and uses HTTPS for communication. It is recommended to have a high speed connection.

3.2 Functional Requirements

3.2.1 System Initialization

3.2.1.1 Description and priority

This shows the user the front page of the Cardigram which is a splash activity after which is an activity which includes the create account for patients. The home screen consists of a typewriter text welcoming the user with a customized personalized text.

Priority: **High**

3.2.1.2 Stimulus/Response sequences

The action is triggered on the starting of the application.

The user can select either to log in if he/she is already registered or create new account.

3.2.1.3 Functional Requirements

Req 1. The system should be working.

3.2.2 Create Account

3.2.2.1 Description and priority

Upon clicking the create account button the user is directed to a page where he/she will have to register by entering a unique username, password, email and other registration fields accordingly.

Priority: **High**

3.2.2.2 Stimulus/Response sequences

The action is performed when the user wants to create an account.

The account is created after the validation has been done from the verification email sent by the system on the registered email id. The authentication is done by google Firebase Authentication library.

3.2.2.3 Functional Requirements

Req1. The username should be unique and cannot be left empty.

Req2. The password should not be left empty and should be of at least 8 characters.

Req3. The email provided should be valid.

Req4. The display picture provided by the user should be in valid format.

3.2.3 Set Appointments

3.2.3.1 Description and Priority

This functionality can be used by user to setup an appointment with the doctor. The functionality makes use of the Google Places API.

Priority: **Medium**

3.2.3.2 Stimulus/Response sequences

The action is performed when the user clicks on the floating button corresponding to setting up appointment on the home screen or clicks on the appointment event in the Navigation Bar.

The window for setting up appointment is opened up and after setting up the appointment the event is shown in the users list of upcoming events.

3.2.3.3 Functional Requirements

Req1: The user must select a doctor.

Req2: The user should get a location from the google maps window prompt(Active internet connection and GPS must be enabled)

Req3: The user must select a date and time for the appointment.

3.2.4 Uploading Reports

3.2.4.1 Description and priority

The users can upload all their medical data i.e. the reports, medicine prescriptions, and other visitation details on the application's cloud storage. The functionality uses glide and Picasso for dynamic rendering of the images uploaded by the user.

Priority: Medium

3.2.4.2 Stimulus/Response sequences

This functionality is rendered upon choosing the floating button corresponding to the documents upload or by clicking upon the upload event in the navigation bar. User can choose from "Taking A Picture" or "Choose from Gallery" options. The user can input the title and description and save the upload. The image is not compressed to maintain the quality. The images are stored in the firebase storage for higher quality. The image URL is stored for each user.

There is a download card view for each uploaded document

3.2.4.3 Functional Requirements

Req1. The documents uploaded must be in valid format.

Req2. The size of the documents uploaded must be within permissible limits.

3.2.5 Chat Interface For Patients

3.2.5.1 Description and priority

A chat interface for each individual patient to chat with other patients or doctors individually or in a group chat. The chat server utilizes the online cloud server of google Firebase. The user can send images and select images from gallery.

Priority: Medium

3.2.5.2 Stimulus/Response sequences

This functionality appears on pressing the chat float button on the home screen or by selecting the chat option from the navigation bar. The option results in opening of a chat window with the display picture and name of the user on the top.

The images to be sent by the user are selected from the gallery converted to bitmap, compressed and scaled. The compressed and scaled bitmap is then converted to a string which is stored in the database, while on the receiver side the string is decoded and rendered to bitmap.

3.2.5.3 Functional Requirements

Req1. The user must be connected to internet at all times.

Req2: System must be able to detect that the user has joined the specific event and display the chat button accordingly.

Req3: System must coordinate all other users' updates and display them in a sequential order that is common for everyone involved.

3.2.6 Generate Charts

3.2.6.1 Description and priority

This function will generate the charts and graphs taking in the values entered by the user. The type of charts can be chosen from a variety of charts and can be saved into gallery also.

Priority: Medium

3.2.6.2 Stimulus/Response sequences

Whenever a generate chart floating button is pressed or the generate chart event is selected from the navigation bar, the charts window is pop up. The window prompts the user to select from a variety of graphs and the user can also zoom in and out of the generated charts. The charts generated are in or user interaction as the width, height and other parameters can be varied by the user.

3.2.6.3 Functional Requirements

Req1: The user must be a registered user.

Req2: The values entered must be within the permissible limits i.e. the threshold.

3.3 Behavior Requirements

3.3.1 Use Case View

4 Other Non-functional Requirements

4.1 Performance Requirements

The system must be interactive and the delays involved must be less. So in every action-response of the system, there are no immediate delays. In case of opening windows forms, of popping error messages and saving the settings or sessions there is delay much below 2 seconds, In case of opening databases, sorting questions and evaluation there are minor delays based on the amount of data that needs to be fetched and stored.

System must be able to fetch chat message updates within an order of 100ms. This is important due to the real-time nature of the chat interface.

System must be able to fetch events in map view within an order of 2s. Such a big value is given to allow for a lot of events being taken place in the user's vicinity. System must be able to fetch the user profile within an order of 1s. Further changes made to the profile should propagate back to Firebase within 100ms in case the user kills the activity.

All the views should be rendered almost instantaneously and further loading of items should be done later on.

4.2 Safety and Security Requirements

- Information transmission should be securely transmitted to cloud without any changes in information
- The passwords are not directly stored in the database, rather MD-5 (Message-Digest 5 algorithm) is used for generation of hashes thus ensuring the safety of the stored passwords and the privacy of the user.
- Keeping in mind the Doctor-Patient confidentiality terms, the patients records can be seen only by the patient himself and the appointed doctor.
- Unauthenticated use of this application is not to be permitted.
- User authentication should be done using firebase's authentication feature such as private user info and passwords are not visible to system administrators.

4.3 Software Quality Attributes

The cardigram's tools are reusable and can be extended to other such applications. The testing of this product requires traversing of all forms and ensuring any arbitrary error does not occur or is properly handled.

Availability: The system should provide **99%** uptime as it's dependent only on Firebase and Google Maps, which are very reliable services.

Correctness: Firebase data should **at no point** be at an inconsistent state.

Learnability: App interface should be easy to use and learn.

Scalability: The app should be build to be scalable, i.e, able to handle large amount of people and events, and minimize data traffic of users to permit future growth. Scalability factor of at least **logn** is preferred.

Affordability: The app designed is meant to be Open Source and **completely free** to use.

5 Other Requirements

5.1 Database Requirements

Firebase real time database forms a hierarchical structure of nodes. It basically forms a JSON tree of objects. So, to insert values into the database, a child of a specific node is created and the instance of a class is stored in it. Even, while retrieving the values, an object is retrieved from the JSON tree, by specifying the class of the object, and hence the required object is retrieved from the data snapshot.

5.2 Reuse Objectives

The project is hosted on GitHub and can be reused as per the GNU GPL 3.0 license.