Software Requirements Specification Document

|  |
| --- |
| SMART INVENTORY |

Software Requirements Specification

1.0

09/09/2018

TEAM MEMBERS

Rakesh Varma Nadakudhiti

Snohitha Rakashi

Lokeswari Pittu

Nilantha Dambadeni Kalu Achchillage

Shivani Busireddy Niharika Gundala

Karthik Raja Vemula

Submitted in partial fulfilment Of the requirements of CSIS 44-691 Graduate Directed Project 1

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 09/09/2018 | 1.0 | Nilantha Dambadeniya | First Revision |
| 09/16/2018 |  | Shivani Busireddy | Second Revision |
| 09/23/2018 |  | Lokeswari Pittu | Third Revision |
| 9/30/2018 |  |  | Fourth Revision |

# Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  | Dr. Zhengrui Qin  Naveen Kumar Nuggu | Project Requirements | 09/14/2018 |
|  | Dr. Zhengrui Qin  Naveen Kumar Nuggu | ER Diagram | 09/21/2018 |
|  | Dr. Zhengrui Qin  Naveen Kumar Nuggu | Project Schedule(Gantt Chart) | 09/14/2018 |
|  |  | UI designs | pending |

**Table of Contents**

[**Revision History**](#_mz1c4fn372p) **2**

[**Document Approval**](#_azwvxbn61ix2) **2**

[**1. Introduction**](#_gdms1gwbks86) **4**

[1.1Purpose](#_h5wy22n9f52i) 4

[1.2 Scope](#_8pbkmyxwtnr2) 4

[1.3 Definitions, Acronyms, and Abbreviations](#_ba5eu3xzyht6) 6

[1.4 References](#_qsx4ekfdrcb7) 6

[1.5 Overview](#_4l45w6e1m5e4) 6

[**2. General Description**](#_bnnqlhs6cxvp) **7**

[2.1 Product Perspective](#_1wl4aewmchy8) 7

[2.2 Product Functions](#_p29brlz40swu) 7

[2.3 User Characteristics](#_jyxpdkjmg4co) 7

[2.4 General Constraints](#_hmgw3a23zsvg) 7

[2.5 Assumptions and Dependencies](#_ykt9xwt213x5) 7

[**3. Specific Requirements**](#_katmkgtix4ab) **8**

[3.1. External Interface Requirements](#_3pax3chhrj1e) 8

[3.1.1. User Interfaces](#_lzyoy12y4wga) 8

[3.1.2. Hardware Interfaces](#_pdpxljuw0c6h) 8

[3.1.3. Software Interfaces](#_mxor748s0ac0) 8

[3.1.4. Communications Interface](#_6ok44lb6xjgh) 8

[3.2. Functional Requirements](#_1ktkm1teqizx) 8

[Admin Requirements](#_26leral4uyd7) 8

[3.3. Use Cases](#_wuksj0wsrjnl) 11

[3.4. Class/Objects](#_3esgdzpwkugx) 11

[3.5. Non-Functional Requirements](#_4ig8oporm4xg) 11

[3.5.1. Performance](#_3xj5degh5rna) 11

[3.5.2. Reliability](#_fuebxjvjyiah) 11

[**4. Design**](#_9m71k783kmqr) **12**

[4.1 ER diagram](#_2k0odvuvg1fg) 12

[4.2 GUI](#_hodqvgqus4q1) 13

[**5. Analysis Models**](#_53qszc7ttj0a) **16**

[5.1 Gantt Chart](#_xfx45tb5bvqy) 16

**6.** [**Technical Manual**](#_vax0iq6gy2n) **17**

# 1. Introduction

## 1.1Purpose

This project requirement document contains the requirements for the Smart Inventory – Android (SMI) the inventory control and android based vendor – user interaction interface application system. These requirements are identified and defined by the client and the project team. These requirements are implemented in the development of the application and if there are any changes or new requirements, client and development team need to be agreed to include them in this document.

## 1.2 Scope

Vendor is the person who run the business and maintain the inventories. Users are the people who provide items to vendor for the price vendor asked for. The inventory will hold the information about users and items.

The inventory system should have two level of access, admin level and user level. Admin should be able to add, remove and edit the item list as well as accept, add, remove users and send messages to the users. In user level, users should be able to create a user account, view the item inventory, request approvals and shipping labels and track the transaction history.

Admin can access the inventory through his personal computer and on his android phone. Client can access the inventory through his android phone.

## 1.3 Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **Definition** | **Abbreviation** |
| SIA | Smart Inventory-Android |
| UI | User Interface |
| ERD | Entity Relationship Diagram |
| URD | User Requirement Document |
| SRD | Software Requirement Document |

## 1.4 References

## 1.5 Overview

This document is divided into five chapters, Introduction, General Description, Specific requirements, Design and Analysis Models.

In the Introduction it describe the purpose and scope of the project.

The General Description gives a clear and concise description of product prospective. It consists of a short view on the product, a short summary of the general capabilities of the product and more in-depth view of the main parts of the product.

The Specific requirements chapter will describe all the external interface requirements, functional requirements, use cases, class/object definitions and non-functional requirements. Moreover, here it will describe the user constraints, logical database requirements, prototype and use case diagrams.

The Design chapter will describe the ER diagram and the UI designs.

The last chapter will describe the Data Flow diagram and Sequence design.

# 2. General Description

## 2.1 Product Perspective

The client (Vendor) is running a business that he buys items for bargained price from registered users and sell them on regular price to make a revenue. Right now the client is maintaining the user records and item inventories manually on spreadsheets and accounting books. Smart Inventory system is an effort of computerized inventory system for the above business.

With the new system, vendor get to create a list of required items with the description and users can view them on real time and supply the items to the vendor. When vendor add or edit an item on the inventory, he can declare an expiring date for the item to be alive on the user list. User can inform the vendor the number of specific items he can supply and request for shipping labels. User will get 7 days to ship the items after receiving the shipping labels from the vendor, otherwise shipping labels will be expired. Once the vender receive the items he conforms it to the user and then user can bill the vendor for the items. Vender will pay using third party payment method.

## 2.2 Product Functions

## 2.3 User Characteristics

## 2.4 General Constraints

## 2.5 Assumptions and Dependencies

# 3. Specific Requirements

## 3.1. External Interface Requirements

## 3.1.1. User Interfaces

## 3.1.2. Hardware Interfaces

## 3.1.3. Software Interfaces

## 3.1.4. Communications Interface

## 3.2. Functional Requirements

### Admin Requirements

1. Admin login

Administrator should be able to log in to the system with admin privileges using his admin username and password

|  |  |
| --- | --- |
| Admin username | Any username |
| Password | Minimum of 8 characters  At least one uppercase letter  At least one lowercase letter  At least one number or symbol (!, #, %, &. \*) |

2. Add new items to the inventory

Administrator should be able to add new items to the list. Most recently added item should be displayed on the top of the inventory list. New item should have following fields. Items should be able to publish to the user’s list or hide it from the user’s list.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Picture of the item | Item | Description | Buying price | # of units needed | Added date | Expiring Date | Comments |

3. Edit/Remove items on the inventory

Administrator should be able to edit or remove any existing items on the list. When editing, he can edit all the fields except ID.

4. Accept or deny membership requests from new users

When a new user wanted to sign up for the Smart Inventory, he may fill up a sign up form and send the request to the vendor. Vendor should be able to accept or deny the request and send an email to the user informing the decision.

5. Accept or deny items supply requests from users

Once a user submit an item supply request, vendor should be able to accept or deny the request and inform the user the decision through an email.

6. Admin should be able to send shipping labels to the user

If vendor accept the item supply request, he should send shipping label to the user’s email for the accepted items.

7. Conform the received items from the users

Once vendor received the items form a user, vendor should be able to conform the received items and send a request for invoice from the user. These information will be sent to the user’s email.

8. Accept and pay user bills

Once user send the invoice for the items, vendor will pay the user through a third party money transfer system.

### User Requirements

1. Make request for sign up

When a potential user wanted to be a member of the smart inventory system, he should fill the sign up form and send it to the vendor for approval. When a user sign up for the system, he should provide following details.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| First Name | Last Name | Email Address | Password | Contact number | Date of birth | Gender | Selected security question | Security answer |

In addition he needs to agree to the user agreement and user privacy notice

2. Log in to the Smart Inventory - Android

Once vendor approves the signup request, user can log in to the Smart Inventory – Android using his email address and password.

3. Recover lost or forgotten password

If user lost or forgot the password, he can recover the password by answering to the security question. The new password will be sent to the user’s email.

4. View the items and details vender looking for

Once the user log in to the Smart Inventory – Android, he can view the list of items currently vender is looking for. In the initial list it will display the following fields

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Picture of the item | Item | Buying Price | # of units needed | Expiring date |

If user wants to see more details about a specific item, he can tap on the item and view the other information and methods of supplying the items.

5. Send requests to vendor for supplying items and request shipping labels

If user wish to supply any items to the vendor he can fill an item supply request and send it to the vendor for acceptance. In that request user need to provide the following details.

|  |  |
| --- | --- |
| Number of items | If shipping label required |

6. Send requests for canceling/editing supplying list

If user wants to cancel or change a supply request that sent to the vendor, he should be able to send a cancel/change supply request to the vendor.

7. Send invoice to vendor for payments

Once user receive the item received confirmation from the vendor, user can send an invoice to vendor for payments. In this invoice, user should provide the available methods for receiving money from the vendor.

8. View the user’s transaction history

User should be able to view his transaction history with vendor

9. Logout

## 3.3. Use Cases

## 3.4. Class/Objects

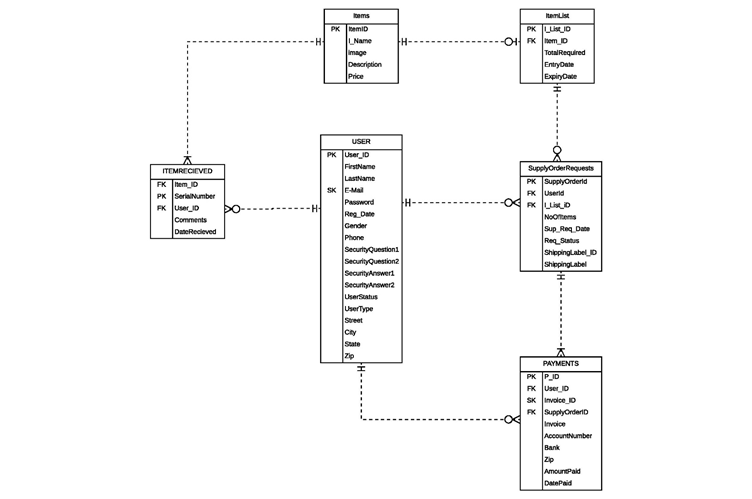
## 3.5. Non-Functional Requirements

### 3.5.1. Performance

### 3.5.2. Reliability

# 4. Design

## 4.1 ER diagram



## 

## 4.2 GUI

|  |  |
| --- | --- |
| Login Page | Password Recovery |
|  |  |
| Signup | Signup2- SecQuestions |
|  |  |
| User Home | Item Details |
|  |  |
| Admin View | Add Item |
|  |  |
| New user request | New user accept |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Chapter 4 to be continued as the project progress…………………

# **5.** **Analysis Models**

## 5.1 Gantt Chart

Please refer to the support document Gantt Chart PDF and Gantt Chart MPP.

Documentation continues as the project progress…………………

Smart Inventory - Android App

Version 1.0.0

# Technical Manual

Team : Android Team

Course: 44691-1

Class : GDP 01 Fall 2018

TEAM MEMBERS  
Rakesh Varma Nadakudhiti  
Snohitha Rakashi  
Lokeswari Pittu  
Nilantha Dambadeni Kalu Achchillage  
Shivani Busireddy  
Niharika Gundala   
Karthik Raja Vemula

# Table of Contents

[**1 Introduction**](#_f3rel8o5888f) **20**

[1.1. Document Identification](#_i14fwx8kd7sv) 20

[1.2. System overview](#_7q0n13x36i02) 20

[1.3. Tools and technologies](#_5yy8od98r9kb) 20

[1.3.1. Android Studio 3.2.1](#_hv89sf7ee95z) 20

[1.3.2 Firebase](#_qf120emkx9wl) 20

[1.3..3 GitHub](#_3zphodeam6d) 20

[1.4 Document overview](#_ro0b8ev1tziv) 21

[**2. Client Signup**](#_3c8cta8wg89z) **22**

[2.1. Requirement](#_rn22o6owqbvq) 22

[2.2 Technical information](#_hlp0iw4a6rnw) 22

# 1 Introduction

## 1.1. Document Identification

This document describe the technical aspect of design and implementation of SmartInventory App for android based mobile phone. This document is prepared by Android team of GDP 01 (44691-1) Fall 2018.

## 1.2. System overview

SmartInventory-Android is developed for android base mobile devices to make it possible for clients to communicate with the vendor for supplying vender required items. This app is consist of two parts as client application and admin application. Clients can download the client application through internet and use it to communicate with the vendor and supply items for vendor. Admin application can be used by vendor and it is not distributed through the internet. Admin application can be used to reply to client requests, add items, upload shipping labels and other administrative tasks.

## 1.3. Tools and technologies

### 1.3.1. Android Studio 3.2.1

Android Studio 3.2.1 is used to develop the User Interfaces (UIs) -Activities, and frontend and backend functionalities.

### 1.3.2 Firebase

Google Firebase is used for backend data handling, storing and manipulation. Also Authentication, Email verification and password recovery functionalities are implemented using firebase functionalities.

Firebase Account Credentials

User Name : [gdpsmartinvntory@gmail.com](mailto:gdpsmartinvntory@gmail.com)

Password : SmartInventory123

### 1.3..3 GitHub

Github repository is used for version control and collaborating purpose of the development process. All the implementation and documentation work were uploaded to the github by each member of the team as they work.

Github link : https://github.com/NilanthaD/SmartInventory\_Android.git

### 1.4 Document overview

This document will describe the technical aspect of each activity, functionality and associations separately. The list and order of the items will be as follows.

1. Sign Up for user
2. Login for user
3. Password recovery for user
4. User logout

Documentation will be continued as the functionalities are implemented….

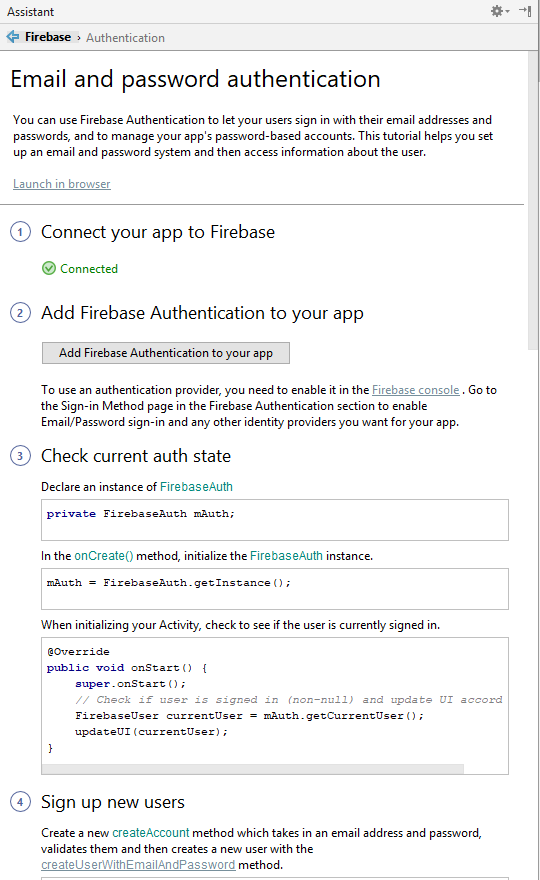
# 2. Client Signup

## 2.1. Requirement

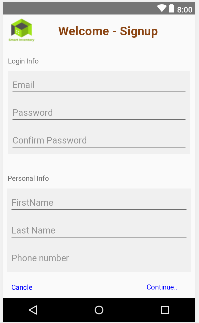
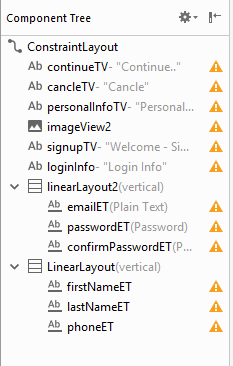
Once the user download the app, he can see the signup option on the login window. Once he click on the signup option, he will directed to the signup page. Then user can provide the necessary information and sign up for the app. Then the user will get a verification email to his email account and he must verify his email in order to get access to the app.

## 2.2 Technical information

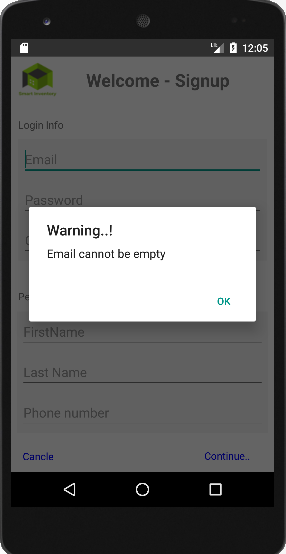
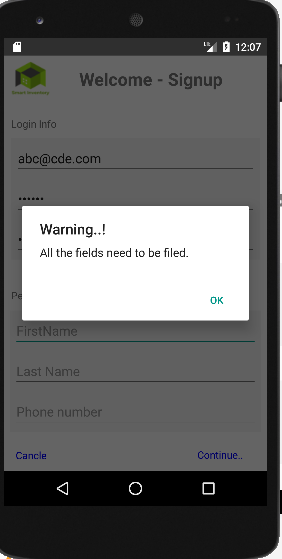
i. App is connected to the Firebase and activated the Authentication with email and password authentication.



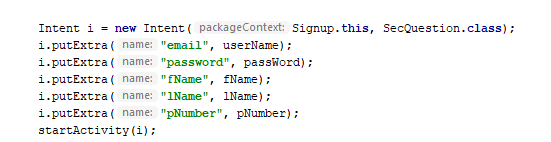
ii. Signup activity is created on Android studio

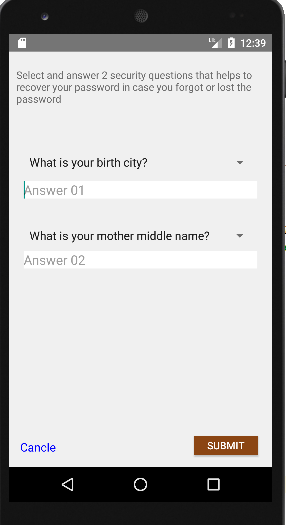
 

iii. User need to fill all the information on signup page, otherwise it will pop up a warning dialog asking to fill all the information.

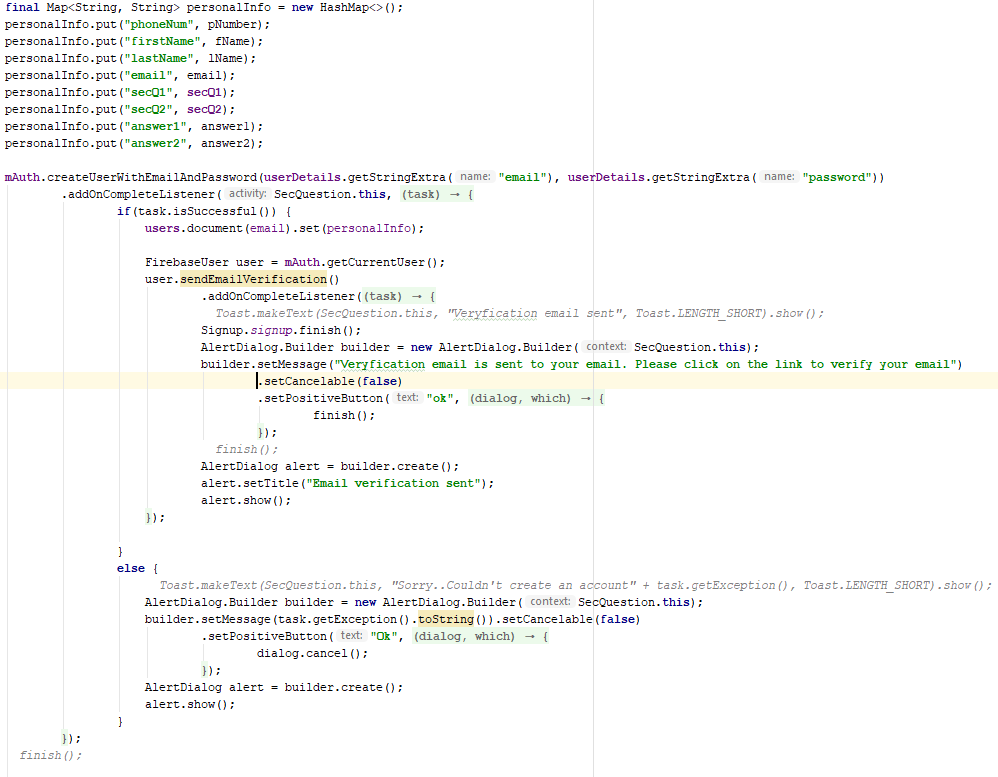
 

iv. Once user fill all the information and click on continue, it will create a new intent with all the user information and go to the SecQuestions activity.

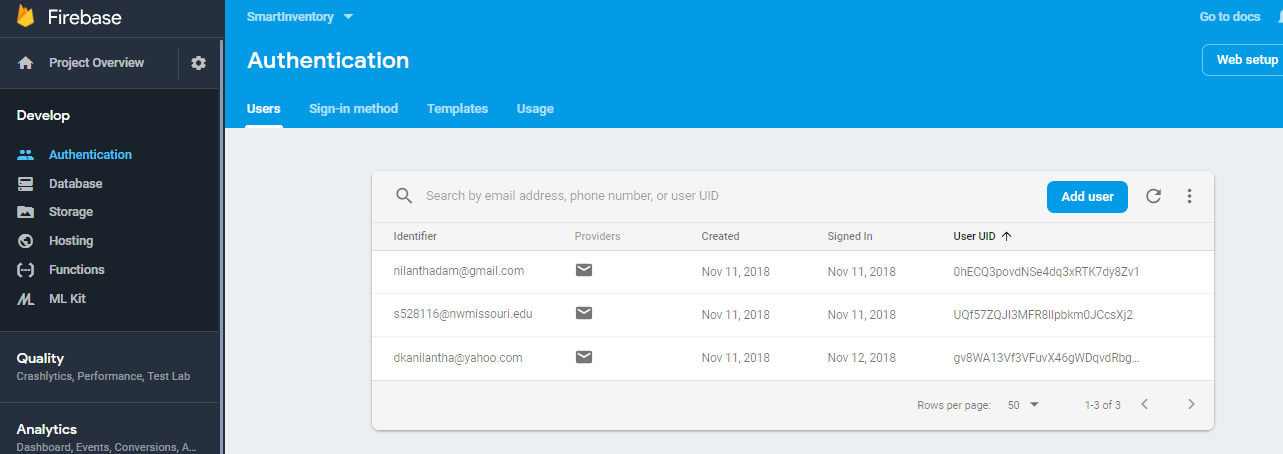




v. Once user anwer the both selected security questions, and click submit button, it will create a record on the firebase authentication and will send an email to the user for verification.

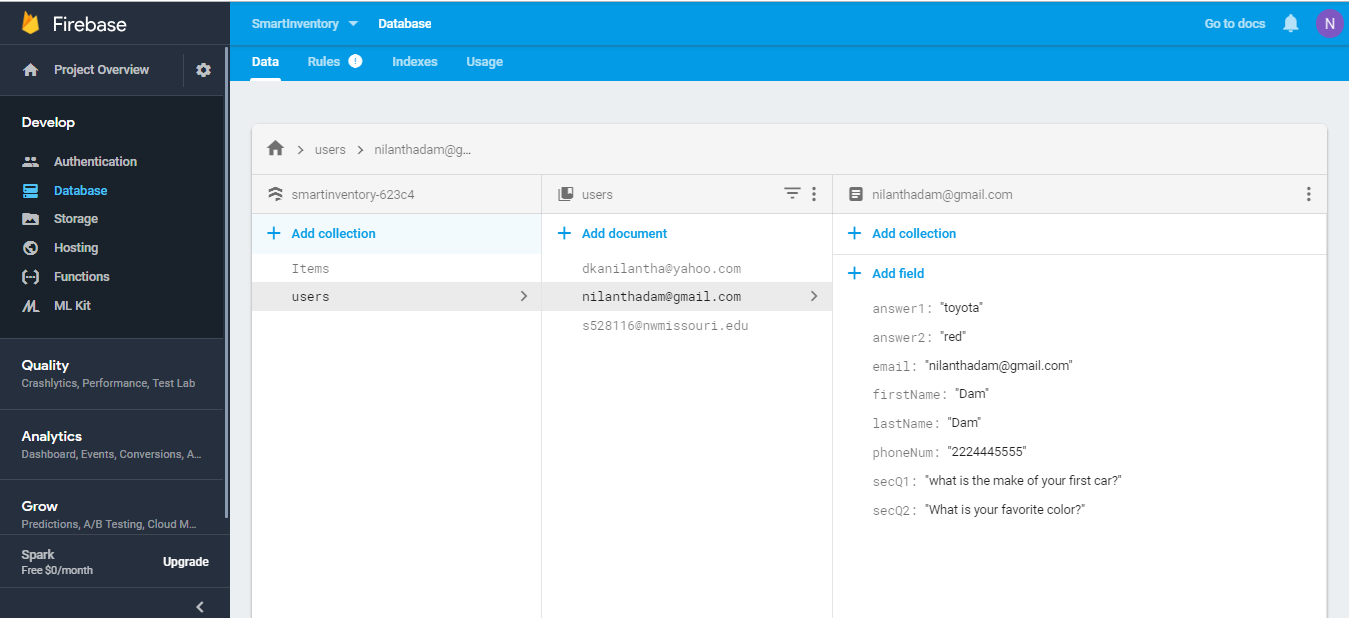


vi. Authentication data will store in the firebase authentication and personal data will store on firebase firesotre.



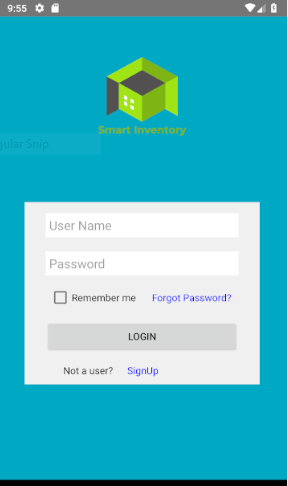
In firestore, user details will store as follows.

Here document id would be the user’s email address.



vii. At any time, if user click the cancel button, it will close all the activities except the login (main) activity.

3. Client Login



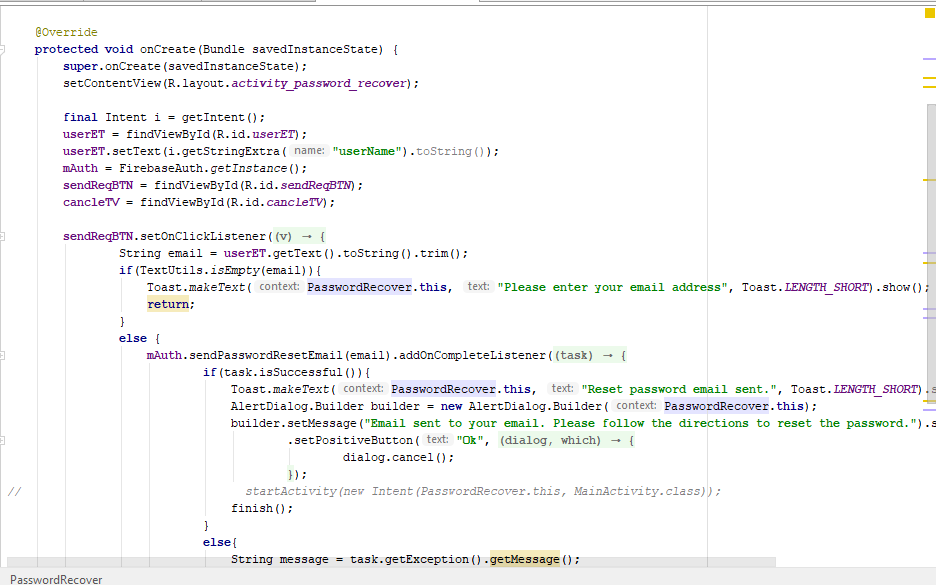
Login Functionality:

****

**ItemList Functionality:**

****

**Password Recovery Functionality:**

****