Design Document

Any Time Medicine

Alok Asok
Kapil Sharma
Sagar Dhamija
Varun Khandelwal
Ramanpreet Singh Khinda

Contents

1. INTRODUCTION 5
1.1. PURPOSE 5
1.2. SCOPE 5
1.3. HIGH LEVEL DESIGN SUMMARY 5
1.4. REFERENCES 5
1.5. ACRONYMS AND GLOSSARY 6
2. GLOBAL DATA STRUCTURES AND SHARED DATA FUNCTIONS 7
2.1. TABLE NAME: ATM_USER_PROFILE_TB 7
2.2. TABLE NAME: ATM_LOGIN_TB 7
2.3. TABLE NAME: ATM_HEALTHCARE_CENTER_TB 7
2.4. TABLE NAME: ATM_INSURANCE_PROVIDER_TB 7
2.5. TABLE NAME: ATM_BODY_TB 8
2.6. TABLE NAME: ATM_REMEDY_TB 8
2.7. TABLE NAME: ATM_SYMPTOM_TB 8
2.8. TABLE NAME: ATM_HISTORY_TB 8
2.9. TABLE NAME: ATM_INSURANCE_PRO_HEALTHCARE_CENTER_JOIN_TB 9
3. HIGH LEVEL DESIGN 10
3.1. USE CASE DIAGRAM 10
3.2. SEQUENCE DIAGRAM 11
3.2.1 US_001: Register 11
3.2.2 US_002: Login 12
3.2.3 US_003: Logout 13
3.2.4 US_004: Get Medicine 14
3.2.5 US_005: Locate healthcare center: 15
3.2.6 US_006 & US_007: Calculate BMI & Get Diet Plan 16
3.2.7 US_008: View History 17
3.2.8 US_009: Change Password: 18
3.2.9 US_010: Reset Password: 19
3.2.10 US_011: Update Profile: 20
3.3. ER DIAGRAM 21
3.4. DATA FLOW DIAGRAM 22
4. USER INTERFACE DESIGN 23
4.1. USR_001: 23
4.1.1. DESIGN DETAILS 23
4.1.1.1. WIREFRAME DESIGN 23
4.1.1.2. PROCESSING DETAILS 23
4.1.1.3. ERROR TO BE CHECKED FOR: 24
4.1.1.4. OUTPUT: 24
4.1.1.5. ERROR CAPTURING 24
4.1.1.6. BOUNDARY AND NORMAL OPERATING CONDITION 26
4.1.1.7. GLOBAL DATA STRUCTURE REFERENCES: 26
4.1.1.8. MODULE SPECIFIC DATA STRUCTURE: 26
4.1.1.9. EXTERNAL INTERFACES: 26
4.1.1.10. ASSUMPTIONS: 26
4.2. USR_002: 27
4.2.2. DESIGN DETAILS 27
4.2.2.1. WIREFRAME DESIGN 27
4.2.2.2. PROCESSING DETAILS 27
4.2.2.3. ERROR TO BE CHECKED FOR: 27
4.2.2.4. OUTPUT: 28
4.2.2.5. ERROR CAPTURING 28
4.2.2.6. BOUNDARY AND NORMAL OPERATING CONDITION 29
4.2.2.7. GLOBAL DATA STRUCTURE REFERENCES: 29
4.2.2.8. MODULE SPECIFIC DATA STRUCTURE: 29

4.2.2.9. EXTERNAL INTERFACES: 29
4.2.2.10. ASSUMPTIONS: 29
4.3. USR_003: 30
4.3.1. DESIGN DETAILS 30
4.3.1.1. WIREFRAME DESIGN 30
4.3.1.2. PROCESSING DETAILS 30
4.3.1.3. ERROR TO BE CHECKED FOR: 30
4.3.1.4. OUTPUT: 30
4.3.1.5. ERROR CAPTURING 31
4.3.1.6. BOUNDARY AND NORMAL OPERATING CONDITION 31
4.3.1.7. GLOBAL DATA STRUCTURE REFERENCES: 31
4.3.1.8. MODULE SPECIFIC DATA STRUCTURE: 31
4.3.1.9. EXTERNAL INTERFACES: 31
4.3.1.10. ASSUMPTIONS: 31
4.4. USR_004: 32
4.4.1. DESIGN DETAILS 32
4.4.1.1. WIREFRAME DESIGN 32
4.4.1.2. PROCESSING DETAILS 32
4.4.1.3. ERROR TO BE CHECKED FOR: 33
4.4.1.4. OUTPUT: 33
4.4.1.5. ERROR CAPTURING 33
4.4.1.6. BOUNDARY AND NORMAL OPERATING CONDITION 33
4.4.1.7. GLOBAL DATA STRUCTURE REFERENCES: 33
4.4.1.8. MODULE SPECIFIC DATA STRUCTURE: 33
4.4.1.9. EXTERNAL INTERFACES: 33
4.4.1.10. ASSUMPTIONS: 34
4.5. USR_005: 35
4.5.1. DESIGN DETAILS 35
4.5.1.1. WIREFRAME DESIGN 35
4.5.1.2. PROCESSING DETAILS 35
4.5.1.3. ERROR TO BE CHECKED FOR: 36
4.5.1.4. OUTPUT: 36
4.5.1.5. ERROR CAPTURING 36
4.5.1.6. BOUNDARY AND NORMAL OPERATING CONDITION 36
4.5.1.7. GLOBAL DATA STRUCTURE REFERENCES: 36
4.5.1.8. MODULE SPECIFIC DATA STRUCTURE: 37
4.5.1.9. EXTERNAL INTERFACES: 37
4.5.1.10. ASSUMPTIONS: 37
4.6. USR_006 & USR_007: 38
4.6.1. DESIGN DETAILS 38
4.6.1.1. WIREFRAME DESIGN 38
4.6.1.2. PROCESSING DETAILS 38
4.6.1.3. ERROR TO BE CHECKED FOR: 39
4.6.1.4. OUTPUT: 39
4.6.1.5. ERROR CAPTURING 39
4.6.1.6. BOUNDARY AND NORMAL OPERATING CONDITION 39
4.6.1.7. GLOBAL DATA STRUCTURE REFERENCES: 39
4.6.1.8. MODULE SPECIFIC DATA STRUCTURE: 39
4.6.1.9. EXTERNAL INTERFACES: 39
4.6.1.10. ASSUMPTIONS: 39
4.7. USR_008: 40
4.7.1. DESIGN DETAILS 40
4.7.1.1. WIREFRAME DESIGN 40
4.7.1.2. PROCESSING DETAILS 40
4.7.1.3. ERROR TO BE CHECKED FOR: 40
4.7.1.4. OUTPUT: 41

4.7.1.5. ERROR CAPTURING 41
4.7.1.6. BOUNDARY AND NORMAL OPERATING CONDITION 41
4.7.1.7. GLOBAL DATA STRUCTURE REFERENCES: 41
4.7.1.8. MODULE SPECIFIC DATA STRUCTURE: 41
4.7.1.9. EXTERNAL INTERFACES: 42
4.7.1.10. ASSUMPTIONS: 42
4.8. USR_009: 43
4.8.1. DESIGN DETAILS 43
4.8.1.1. WIREFRAME DESIGN 43
4.8.1.2. PROCESSING DETAILS 43
4.8.1.3. ERROR TO BE CHECKED FOR: 43
4.8.1.4. OUTPUT: 44
4.8.1.5. ERROR CAPTURING 44
4.8.1.6. BOUNDARY AND NORMAL OPERATING CONDITION 45
4.8.1.7. GLOBAL DATA STRUCTURE REFERENCES: 45
4.8.1.8. MODULE SPECIFIC DATA STRUCTURE: 46
4.8.1.9. EXTERNAL INTERFACES: 46
4.8.1.10. ASSUMPTIONS: 46
4.9. USR_010: 47
4.9.1. DESIGN DETAILS 47
4.9.1.1. WIREFRAME DESIGN 47
4.9.1.2. PROCESSING DETAILS 47
4.9.1.3. ERROR TO BE CHECKED FOR: 47
4.9.1.4. OUTPUT: 48
4.9.1.5. ERROR CAPTURING 48
4.9.1.6. BOUNDARY AND NORMAL OPERATING CONDITION 48
4.9.1.7. GLOBAL DATA STRUCTURE REFERENCES: 48
4.9.1.8. MODULE SPECIFIC DATA STRUCTURE: 49
4.9.1.9. EXTERNAL INTERFACES: 49
4.9.1.10. ASSUMPTIONS: 49
4.10. USR_011: 50
4.10.1. DESIGN DETAILS 50
4.10.1.1. WIREFRAME DESIGN 50
4.10.1.2. PROCESSING DETAILS 50
4.10.1.3. ERROR TO BE CHECKED FOR: 50
4.10.1.4. OUTPUT: 51
4.10.1.5. ERROR CAPTURING 51
4.10.1.6. BOUNDARY AND NORMAL OPERATING CONDITION 51
4.10.1.7. GLOBAL DATA STRUCTURE REFERENCES: 52
4.10.1.8. MODULE SPECIFIC DATA STRUCTURE: 52
4.10.1.9. EXTERNAL INTERFACES: 52
4.10.1.10. ASSUMPTIONS: 52
5. TRACABILITY TO REQUIREMENTS 53

1. Introduction

1.1. Purpose

The document's chief purpose is to provide a detailed description of the design for "Any Time Medicine" (ATM) application software. It elaborates the purpose and features of the application software, what and how the application does (description of User Interfaces), the design approach used, the programming constructs and their interactions, data design and the architecture design. The document is primarily intended for the stakeholders, developers and quality assurance team and will be proposed to the instructor, teaching assistant and quality assurance team for review and feedback.

1.2. Scope

The "Any Time Medicine" is an android based mobile application that provides one stop medical solution to user's health concerns/ medical issues by recommending Over the Counter (OTC) medicines and home remedies without having to visit the physician.

The scope of the project will be to provide the functionality as described in the requirements document. The application will be developed on a LINUX machine using Android Studio, Oracle 11g and JDBC/ ODBC.

Requirements:

1. User – Register, Login, Logout, Change password, Update Profile, Reset password, Search Remedy, Get BMI and Diet Plan, Locate Health care centre, View history.

1.3. High Level Design Summary

Architecture: Client - Server
Front End: JAVA
Backend: Oracle

1.4. References

- https://en.wikipedia.org/wiki/Software_design_description
- http://www.iso-architecture.org/ieee-p1016/
- https://en.wikipedia.org/wiki/Data_flow_diagram
- https://en.wikipedia.org/wiki/Sequence_diagram
- http://www.justinmind.com/
- http://www.uml-diagrams.org/
- https://en.wikipedia.org/wiki/Class_diagram
- http://www.agilemodeling.com/artifacts/classDiagram.htm
- http://developer.android.com/guide/index.html
- https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model
- https://support.office.com/en-us/article/Create-a-data-flow-diagram-43623251-96c4-4ed5-8da3-70361afed88e
- https://erdplus.com/#/
- https://msdn.microsoft.com/en-us/library/dd409416.aspx

1.5. Acronyms and Glossary

Abbreviation:

Abbreviation	Meaning
N/A	Not Applicable
GPS	Global Positioning System
DOB	Date of Birth
OTC	Over The Counter
FK	Foreign Key
PK	Primary Key

Glossary:

TERM	DEFINITION
FOREIGN KEY	A foreign key is a column (or columns) that references a column (most often the primary key) of another table.
PRIMARY KEY	A primary key is a special relational database table column (or combination of columns) designated to uniquely identify all table records.
NOT NULL	The NOT NULL constraint enforces a column to NOT accept NULL values.
REFERENTIAL INTEGITY	Referential integrity is a property of data which, when satisfied, requires every value of one attribute (column) of a relation (table) to exist as a value of another attribute in a different (or the same) relation (table).
CONSTRAINT	Constraints are the rules enforced on data columns on table.
UNIQUE	The UNIQUE constraint uniquely identifies each record in a database table.
VARCHAR	A varchar or Variable Character Field is a set of character data of indeterminate length.
FIELD	A field is part of a record and contains a single piece of data for the subject of the record

2. Global Data Structures and Shared Data Functions

2.1. Table Name: ATM_USER_PROFILE_TB

FIELD	DATATYPE	CONSTRAINT
USER_ID	NUMBER(50)	PRIMARY KEY(PK_USER_PROFILE)
FIRST_NAME	VARCHAR2(200)	NOT NULL
LAST_NAME	VARCHAR2(200)	NOT NULL
DOB	DATE	CHECK CONSTRAINT
GENDER	VARCHAR2(20)	CHECK CONSTRAINT
HEIGHT	NUMBER(5)	NOT NULL
WEIGHT	NUMBER(10)	NOT NULL
EMAIL_ID	VARCHAR2(50)	UNIQUE & NOT NULL
PHONE_NUMBER	NUMBER(10)	UNIQUE & NOT NULL
INSURANCE_PROVIDER_ID	NUMBER(200)	FOREIGN KEY

2.2. Table Name: ATM_LOGIN_TB

FIELD	DATATYPE	CONSTRAINT
USER_ID	NUMBER(50)	FOREIGN KEY(FK_LOGIN_USER_PROFILE)
EMAIL_ID	VARCHAR2(50)	UNIQUE & NOT NULL
PASSWORD	VARCHAR2(32)	NOT NULL
LOGIN_STATUS	NUMBER(1)	0 - Not Logged in 1 - Already logged in

2.3. Table Name: ATM_HEALTHCARE_CENTER_TB

FIELD	DATATYPE	CONSTRAINT
HEALTHCARE_PROVIDER_ID	NUMBER(100)	PRIMARY KEY(PK_HEALTHCARE_CENTER)
HEALTHCARE_PROVIDER_NAME	VARCHAR2(200)	NOT NULL
ADDRESS	VARCHAR2(250)	NOT NULL
PHONE_NUMBER	NUMBER(10)	UNIQUE & NOT NULL
EMAIL_ID	VARCHAR2(50)	UNIQUE & NOT NULL
LATITUDE	NUMBER(20,10)	NOT NULL
LONGITUDE	NUMBER(20,10)	NOT NULL

2.4. Table Name: ATM_INSURANCE_PROVIDER_TB

FIELD	DATATYPE	CONSTRAINT

INSURANCE_PROVIDER_ID	NUMBER(100)	PRIMARY KEY(PK_INSURANCE_PROVIDER)
INSURANCE_PROVIDER_NAME	VARCHAR2(200)	NOT NULL

2.5. Table Name: ATM_BODY_TB

FIELD	DATATYPE	CONSTRAINT
BODY_PART_ID	NUMBER(100)	PRIMARY KEY(PK_BODY)
BODY_PART	VARCHAR2(200)	UNIQUE & NOT NULL

2.6. Table Name: ATM_REMEDY_TB

FIELD	DATATYPE	CONSTRAINT
REMEDY_ID	NUMBER(100)	PRIMARY KEY(PK_REMEDY)
SYMPTOM_ID	NUMBER(100)	FOREIGN KEY(FK_REMEDY_SYMPTOM)
BODY_PART_ID	NUMBER(100)	FOREIGN KEY(FK_REMEDY_BODY)
HOME_REMEDY1	VARCHAR2(255)	NOT NULL
HOME_REMEDY2	VARCHAR2(255)	NOT NULL
HOME_REMEDY3	VARCHAR2(255)	NOT NULL
OTC1	VARCHAR2(255)	NOT NULL
OTC2	VARCHAR2(255)	NOT NULL
OTC3	VARCHAR2(255)	NOT NULL

2.7. Table Name: ATM_SYMPTOM_TB

FIELD	DATATYPE	CONSTRAINT
SYMPTOM_ID	NUMBER(100)	PRIMARY KEY(PK_SYMPTOM)
SYMPTOM	VARCHAR2(255)	UNIQUE & NOT NULL

2.8. Table Name: ATM_HISTORY_TB

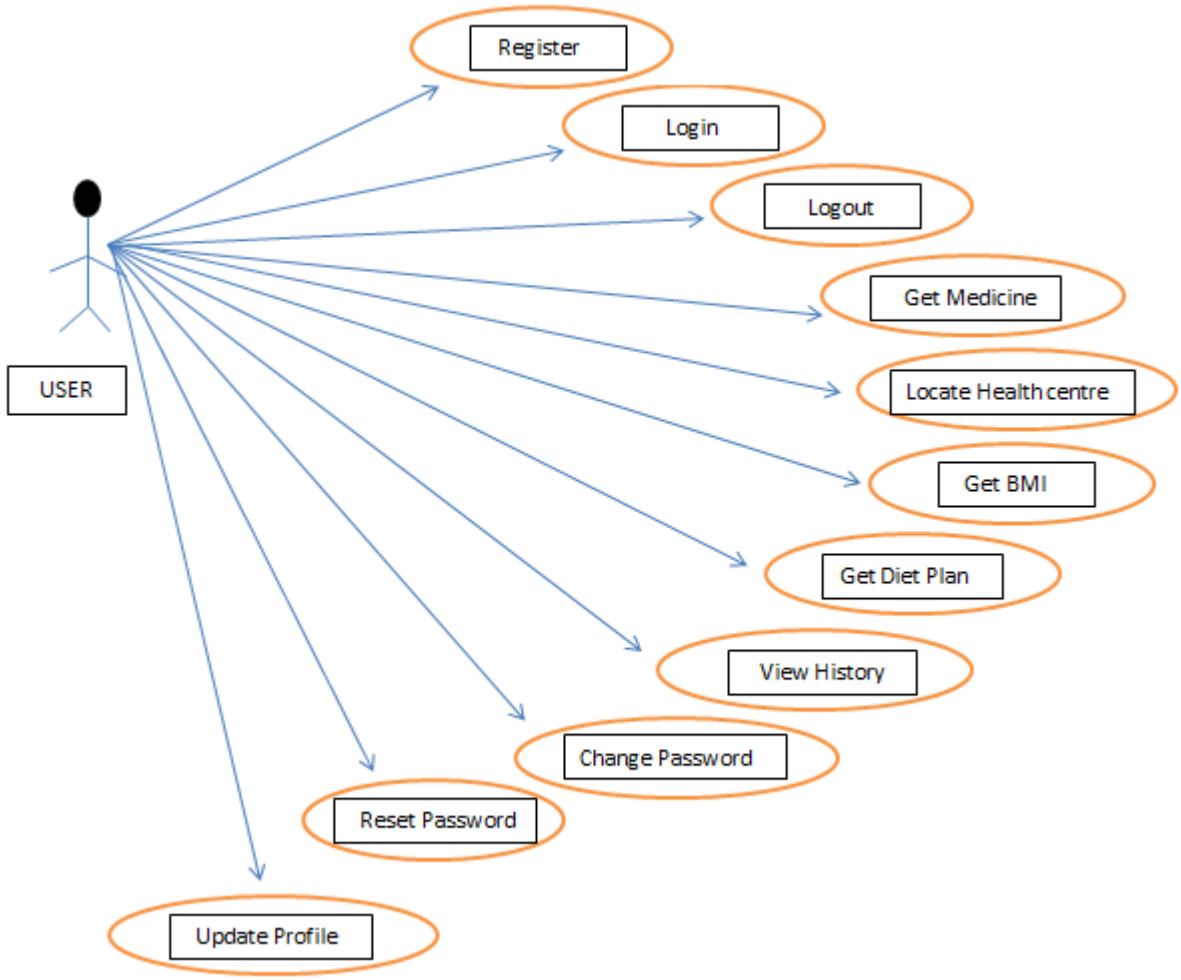
FIELD	DATATYPE	CONSTRAINT
DATE	DATE	NOT NULL
USER_ID	NUMBER(100)	FOREIGN KEY(FK_HISTORY_USER_PROFILE)
REMEDY_ID	NUMBER(100)	FOREIGN KEY(FK_HISTORY_REMEDY)

2.9. Table Name: ATM_INSURANCE_PRO_HEALTHCARE_CENTER_JOIN_TB

FIELD	DATATYPE	CONSTRAINT
INSURANCE_PROVIDER_ID	NUMBER(100)	FOREIGN KEY
HEALTHCARE_CENTER_ID	NUMBER(100)	FOREIGN KEY

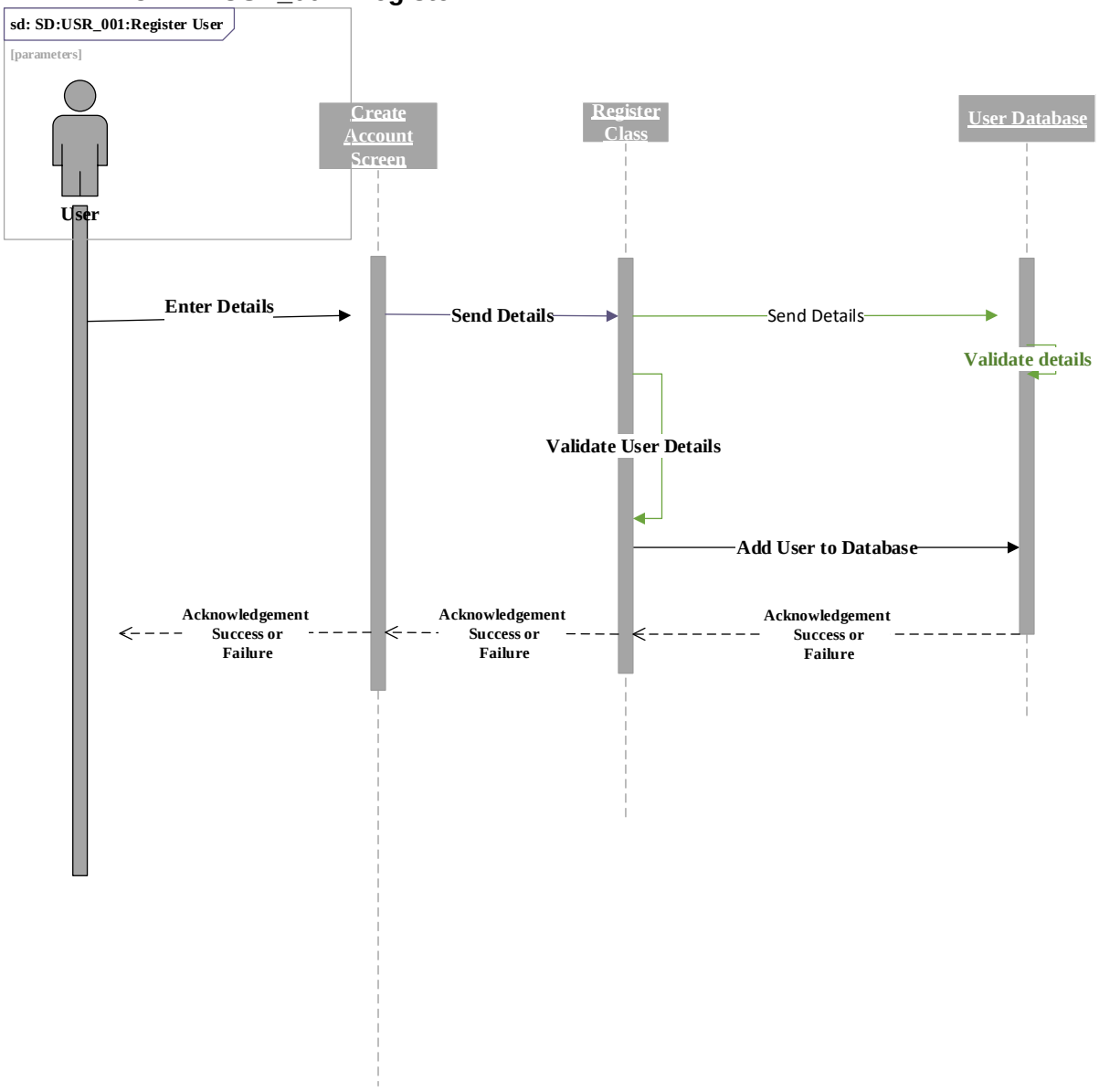
3. High Level Design

3.1. Use case Diagram

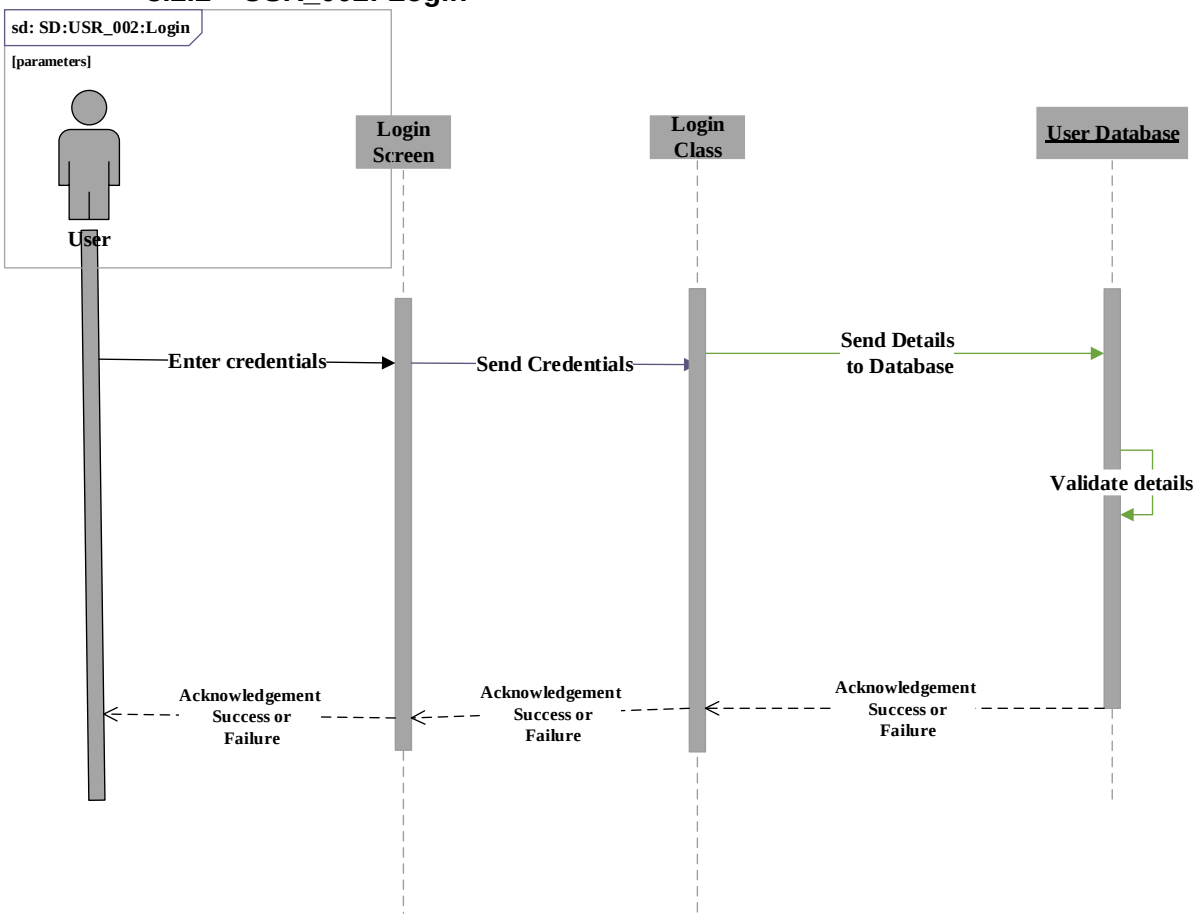


3.2.Sequence Diagram

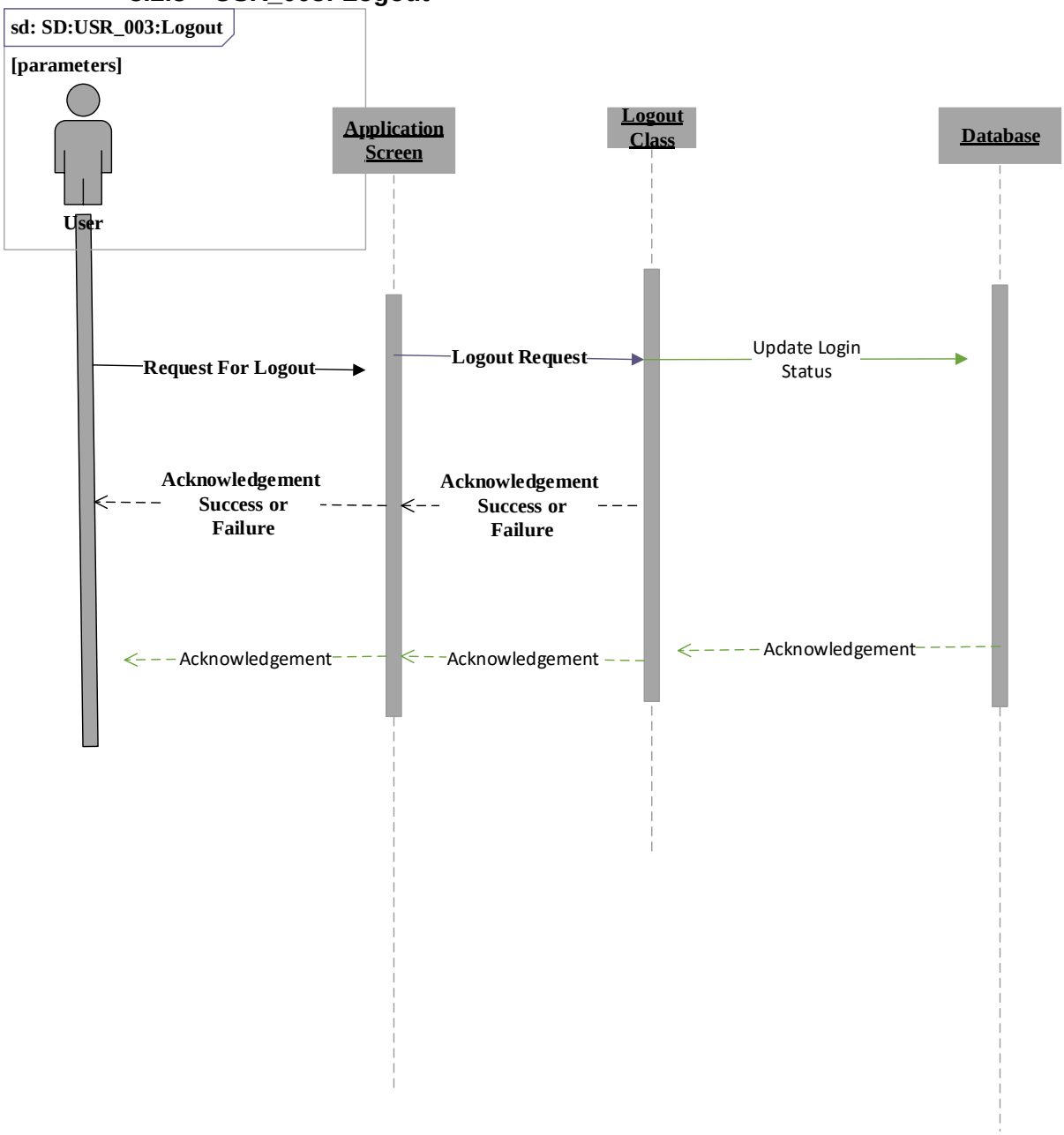
3.2.1 USR_001: Register



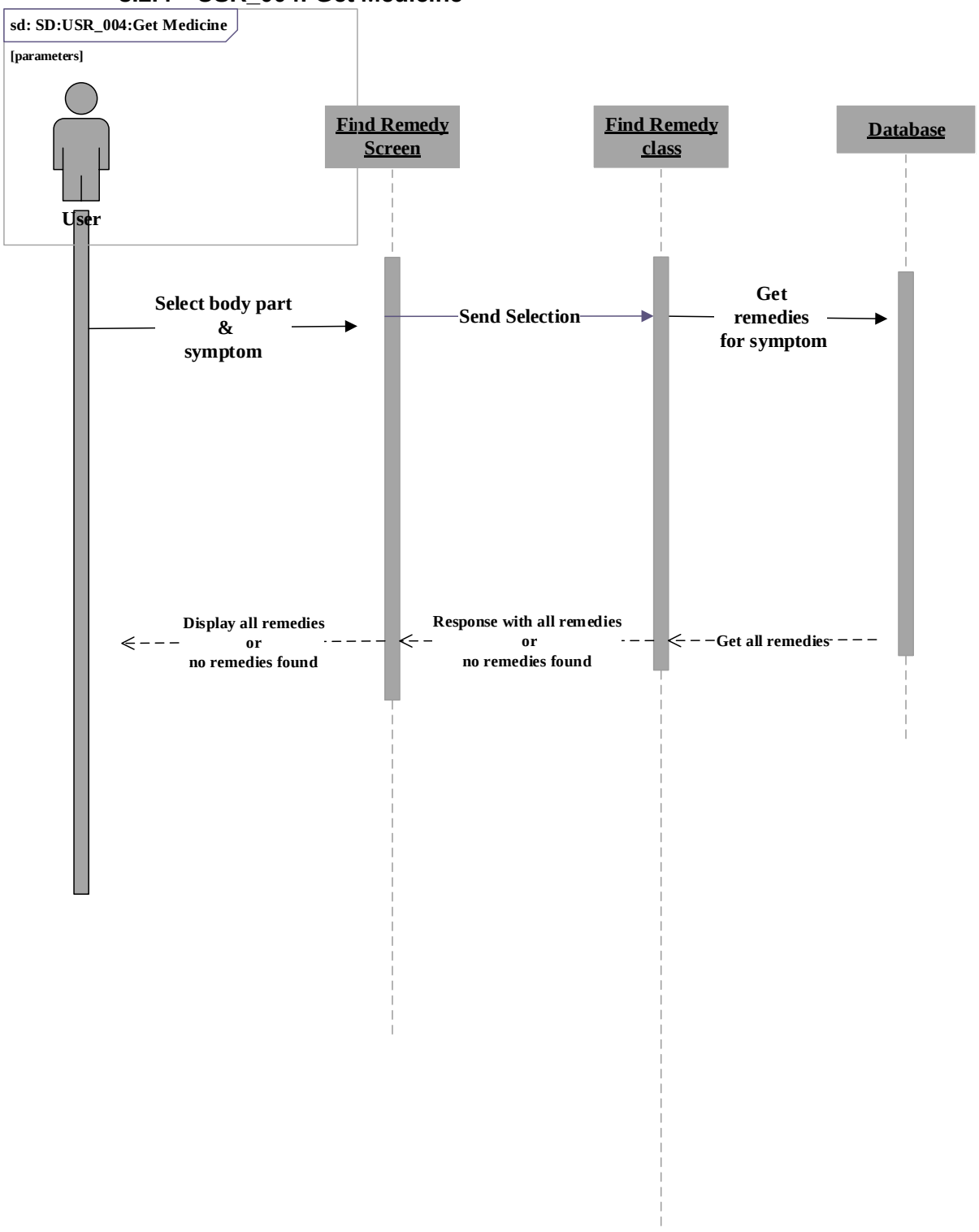
3.2.2 USR_002: Login



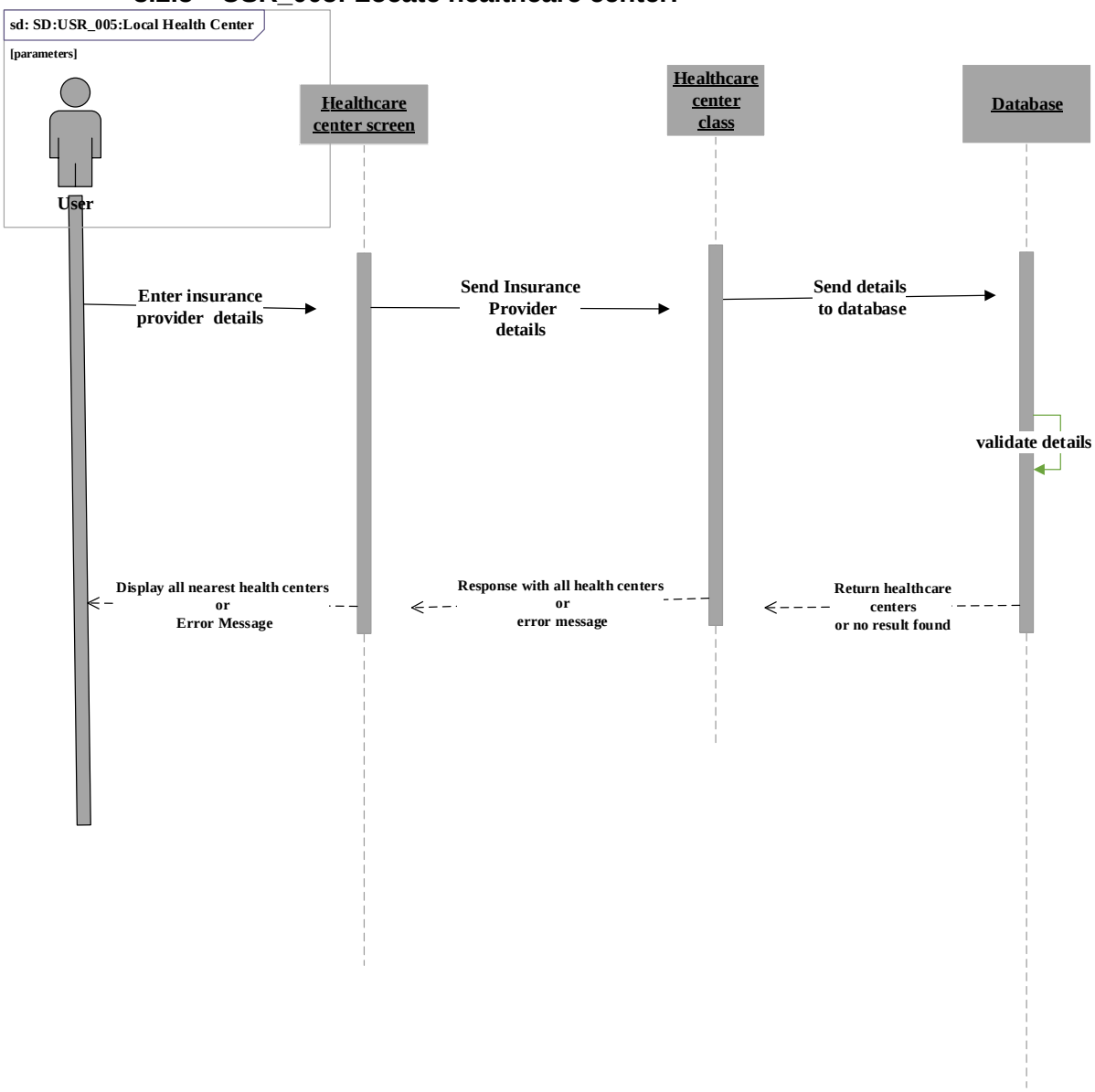
3.2.3 USR_003: Logout



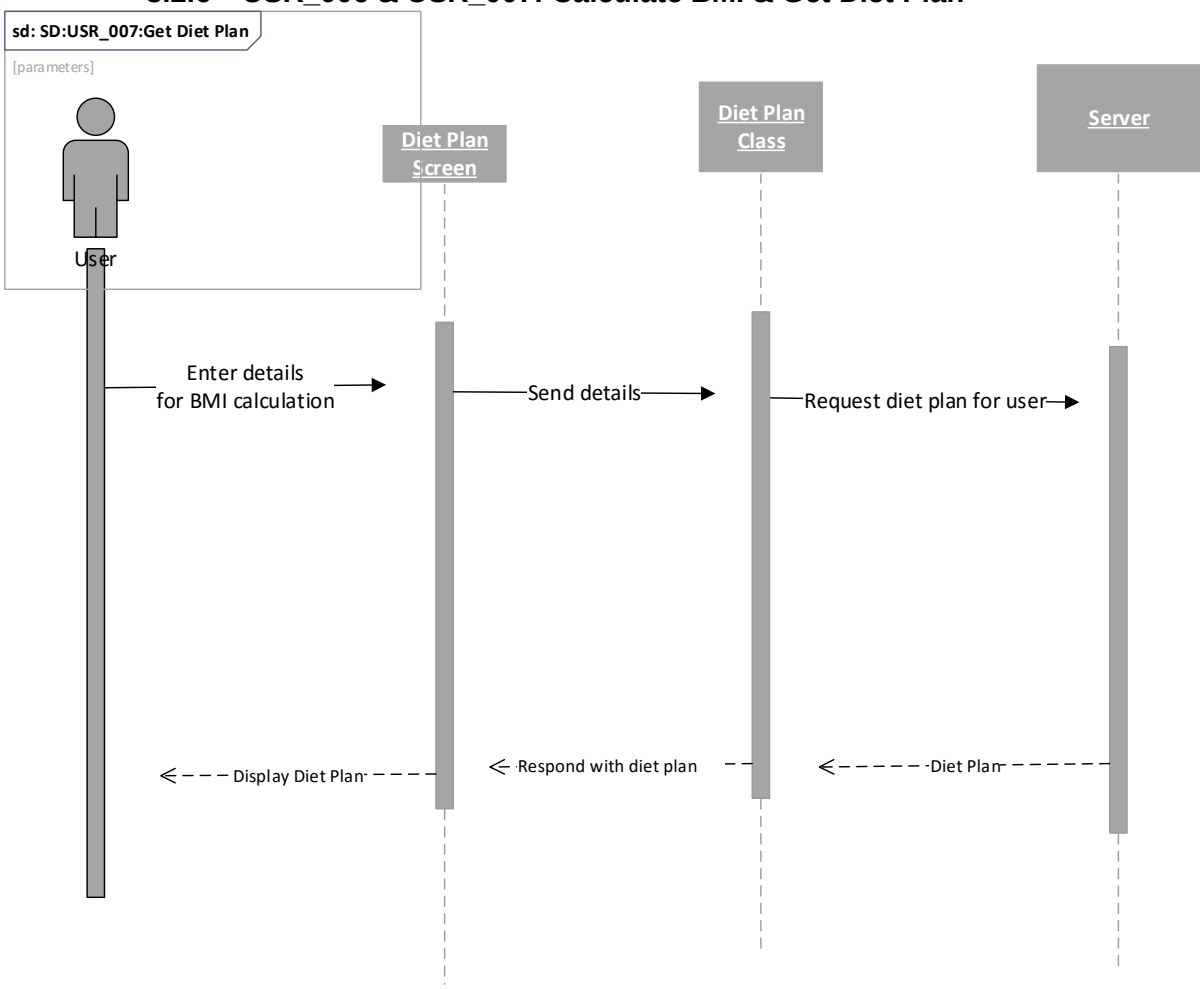
3.2.4 USR_004: Get Medicine



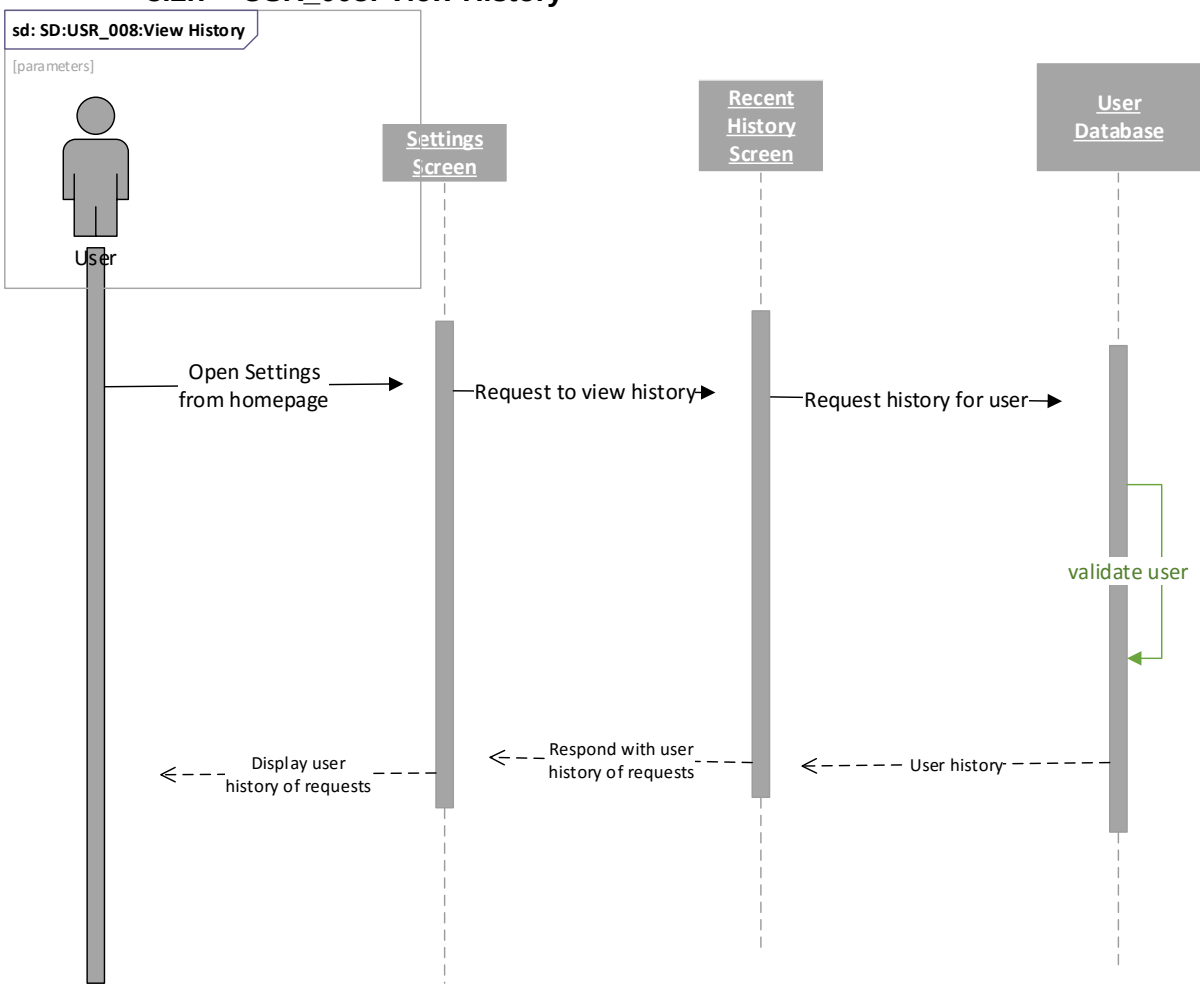
3.2.5 USR_005: Locate healthcare center:



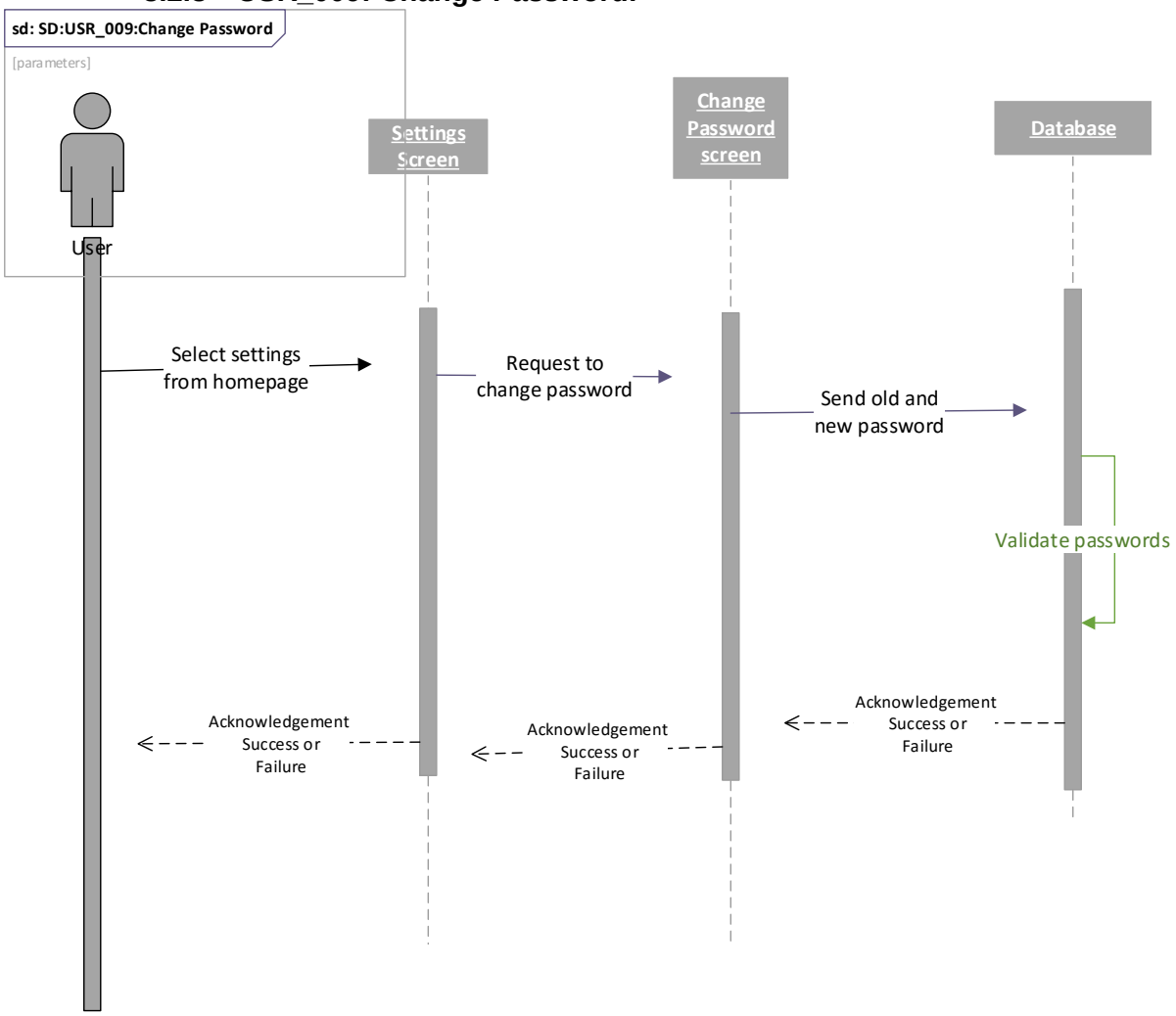
3.2.6 USR_006 & USR_007: Calculate BMI & Get Diet Plan



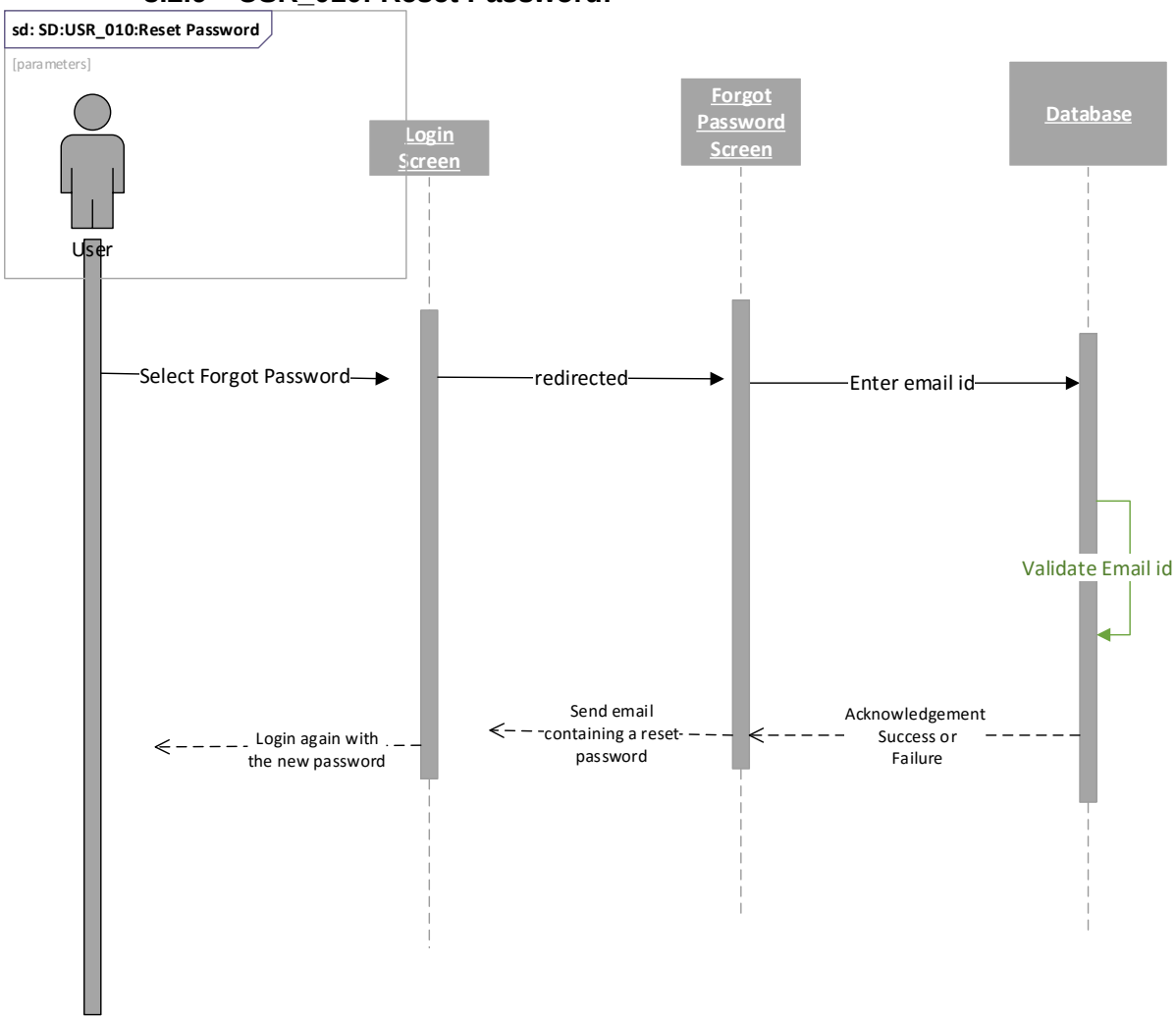
3.2.7 USR_008: View History



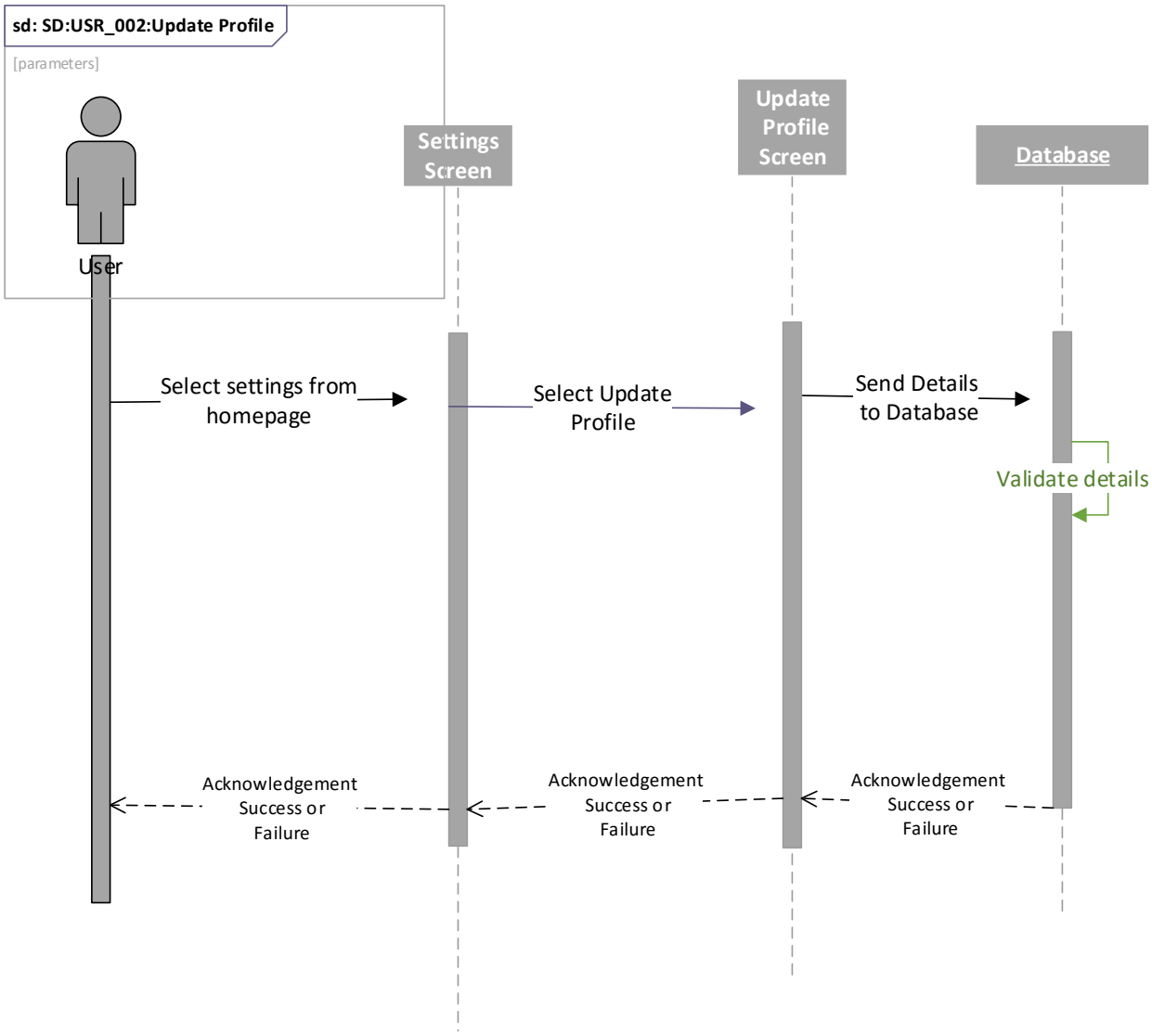
3.2.8 USR_009: Change Password:



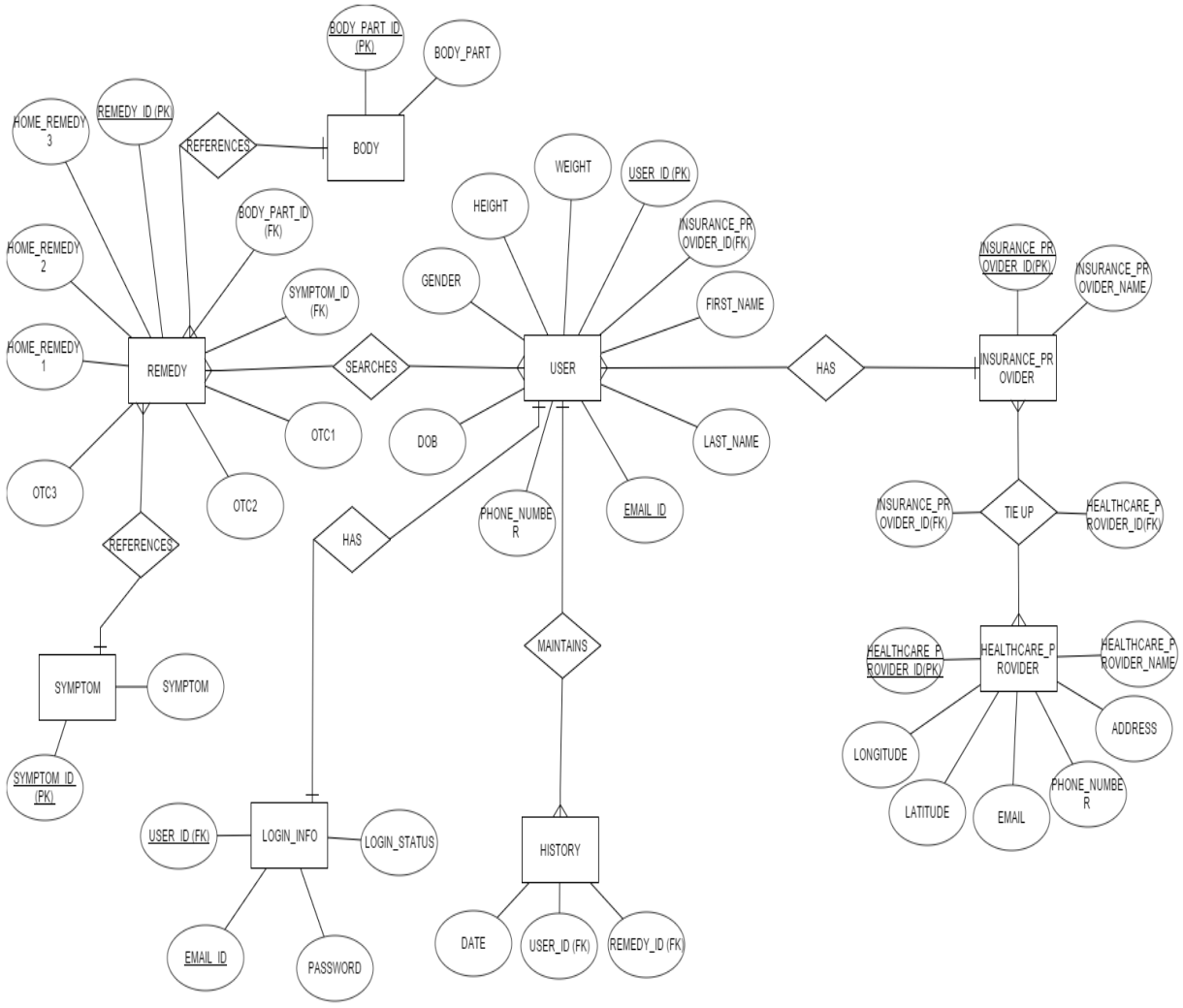
3.2.9 USR_010: Reset Password:



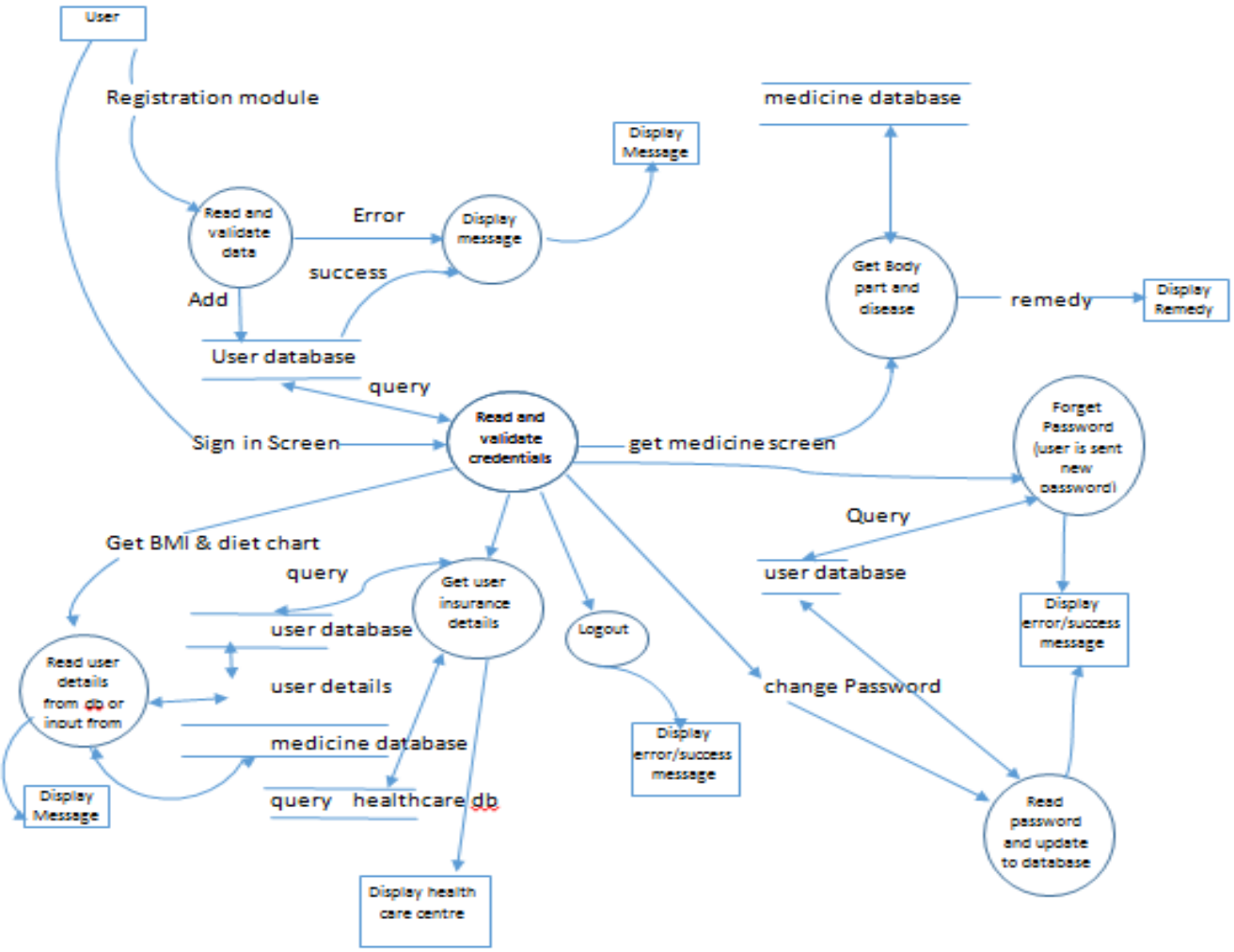
3.2.10 USR_011: Update Profile:



3.3.ER DIAGRAM



3.4.DATA FLOW DIAGRAM



4. User Interface Design

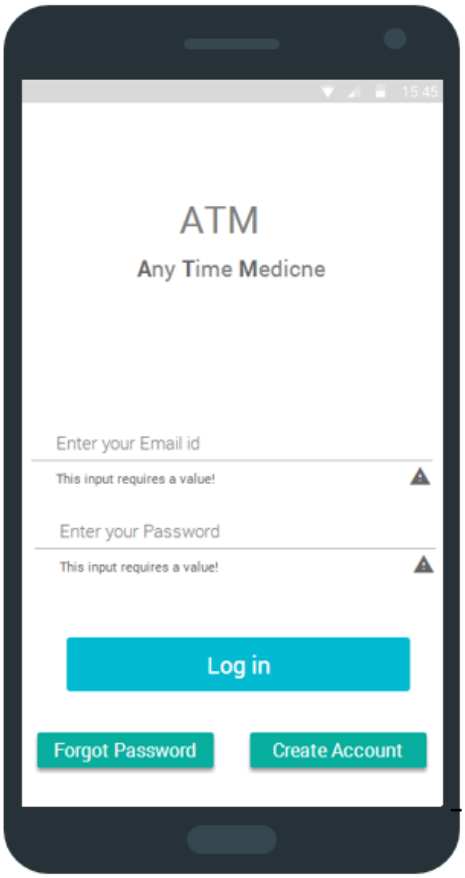
4.1.USR_001:

The user should be able to register with the application

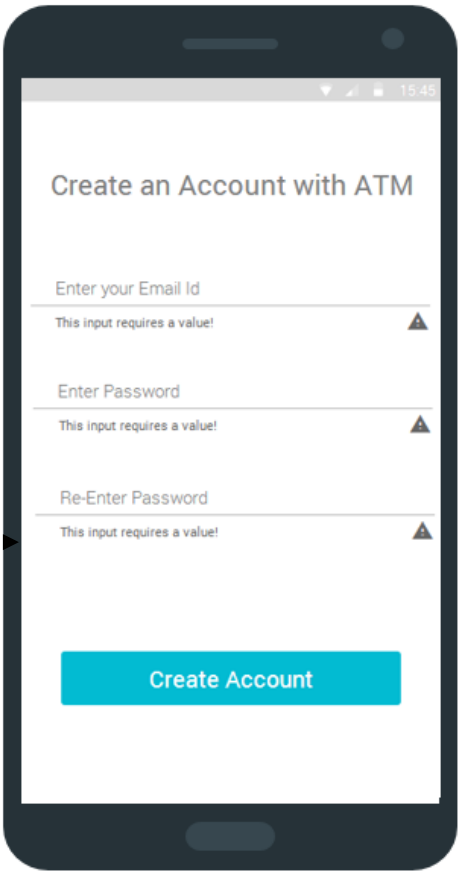
4.1.1. Design Details

4.1.1.1. Wireframe Design

Log in Screen



Create Account Screen



4.1.1.2. Processing Details

Upon opening the application user is presented with the ‘Log in’ screen. If the user is not registered with the application then he can register with the application by simply clicking on the “Create Account” button. Upon clicking the button, user is redirected to the registration

page where he can input his valid email id and password. Once the user has filled in the valid details, the “Create Account” button becomes available for clicking.

Once the user clicks on the “Create Account” button, the user information is stored on the database and a new entry for the user is created with the following fields:

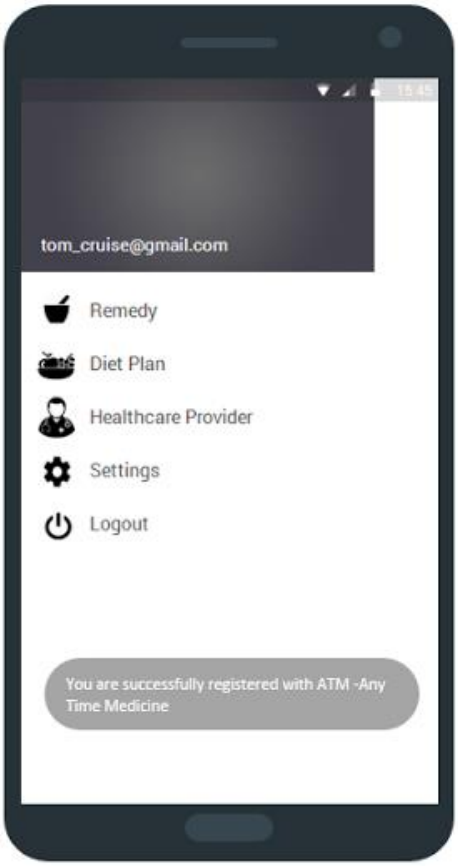
Field Name	Data Type
User_id	Number
Email_id	Varchar2
Password	Varchar2

4.1.1.3. Error to be checked for:

We should check that no field is null. The “Create Account” button is enabled only when all the fields are filled.

4.1.1.4. Output:

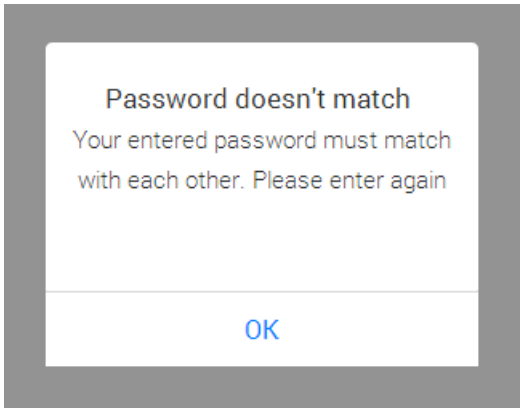
The user information is added to the database and is redirected to the homepage.



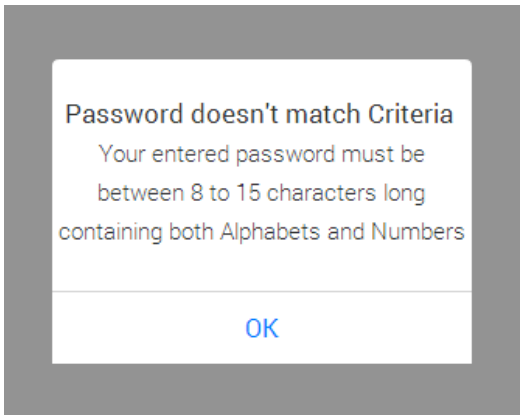
4.1.1.5. Error Capturing

Try catch block for null values. Failure message will be displayed.

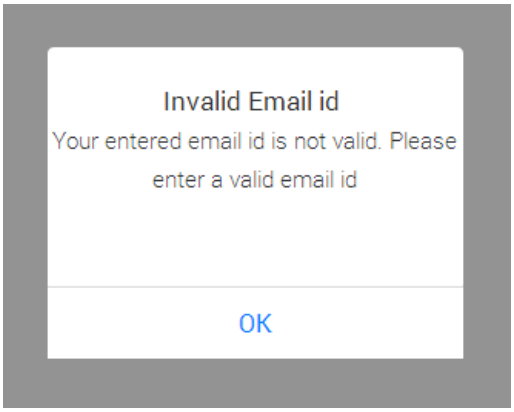
1. If upon re-entering the password, the password does not match with the previously entered password then the following error message is displayed:



2. If the password entered doesn't match the criteria then the following error message is displayed:



3. If INVALID/ ALREADY REGISTERED Email id is entered then the following error message is displayed:



4.1.1.6. Boundary and normal Operating condition

Email id length must not exceed 100 characters and Password length must be between 8 to 15 characters.

4.1.1.7. Global data structure references:

TABLE_NAME	SECTION
ATM_USER_PROFILE_TB	2.1

4.1.1.8. Module specific data structure:

Android EditText and Button

4.1.1.9. External Interfaces:

Android – Remote Database connection

4.1.1.10. Assumptions:

User has downloaded the application

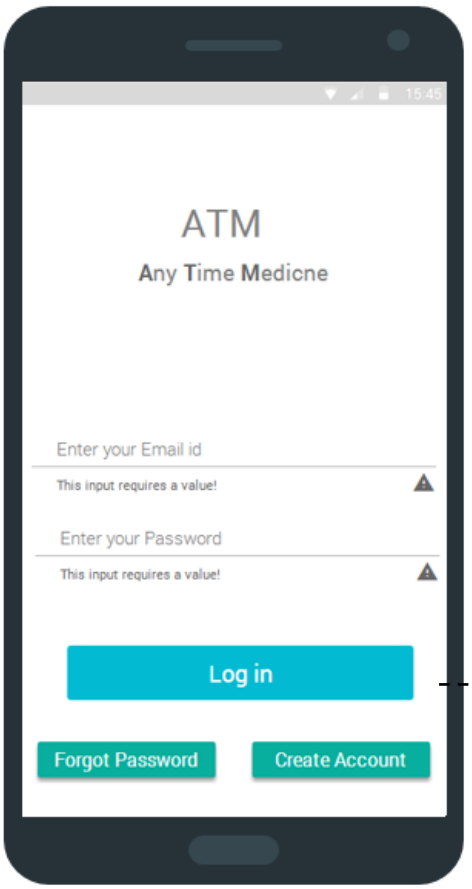
4.2. USR_002:

The user should be able to login to the application

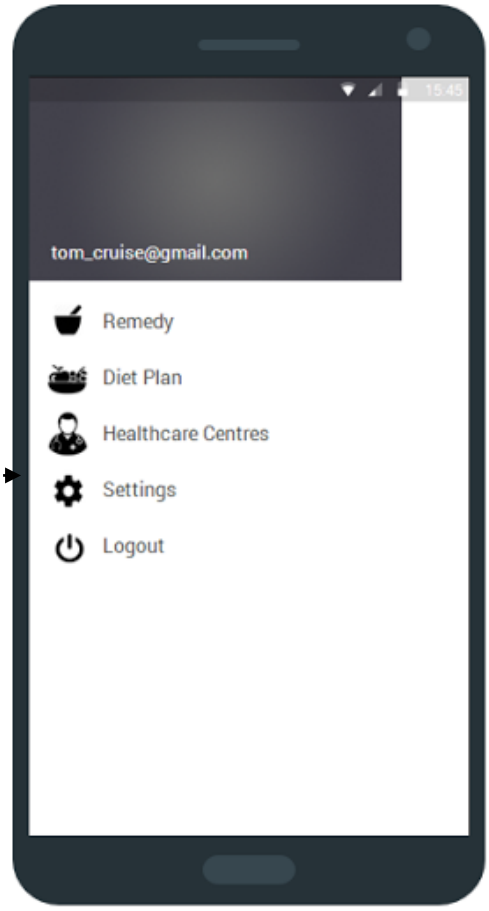
4.2.2. Design Details

4.2.2.1. Wireframe Design

Log in Screen



Homepage



4.2.2.2. Processing Details

Upon opening the application the user is presented with the “log in” screen. Once the user enters valid email id and password the log in button is enabled for clicking.

On clicking the log in button a background service runs which connects with the remote database and validates the user’s email id and password.

If the validation is successful, the user is redirected to his/her homepage.

4.2.2.3. Error to be checked for:

We should check that no field is null. The “Log in” button is enabled only when all the fields are filled.

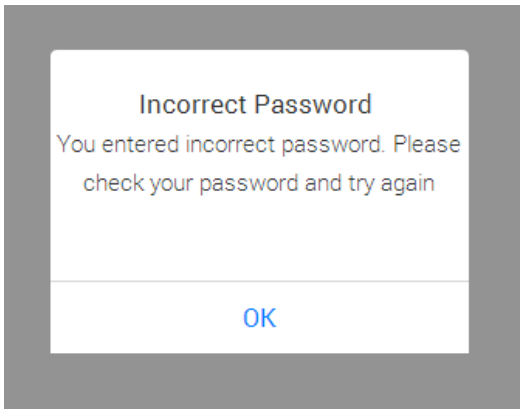
4.2.2.4. Output:

System validates the details and redirects the user to the homepage.

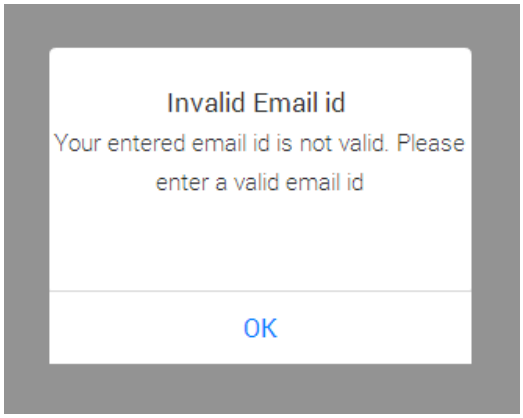
4.2.2.5. Error Capturing

Try catch block for null values. Failure message will be displayed.

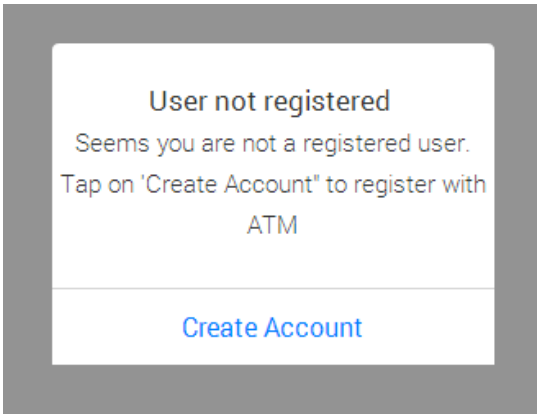
1. If invalid password is entered then the following error message is displayed:



2. If invalid email id is entered then the following error message is displayed :



3. If user tries to log in without registering with the application then the following error is displayed:



4.2.2.6. Boundary and normal Operating condition

Email id length must not exceed 100 characters and Password length must be between 8 to 15 characters.

4.2.2.7. Global data structure references:

TABLE_NAME	SECTION
ATM_LOGIN_TB	2.2

4.2.2.8. Module specific data structure:

Android EditText and Button

4.2.2.9. External Interfaces:

Android – Remote Database connection

4.2.2.10. Assumptions:

User has downloaded the application

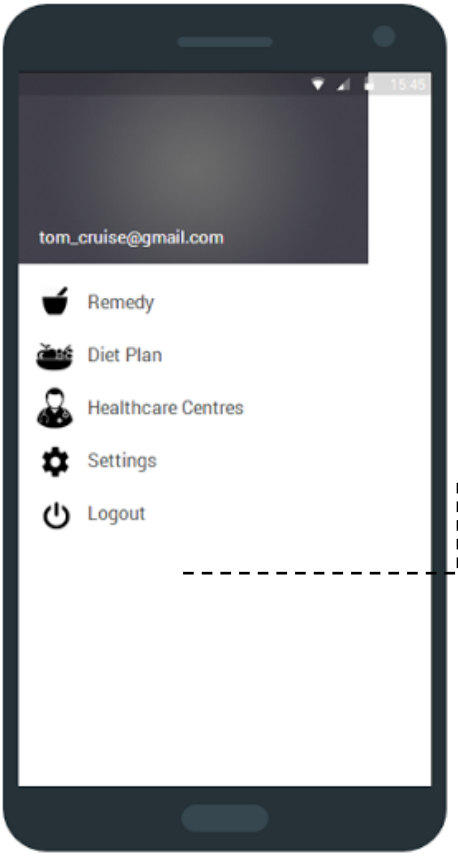
4.3.USR_003:

The user should be able to logout from the application

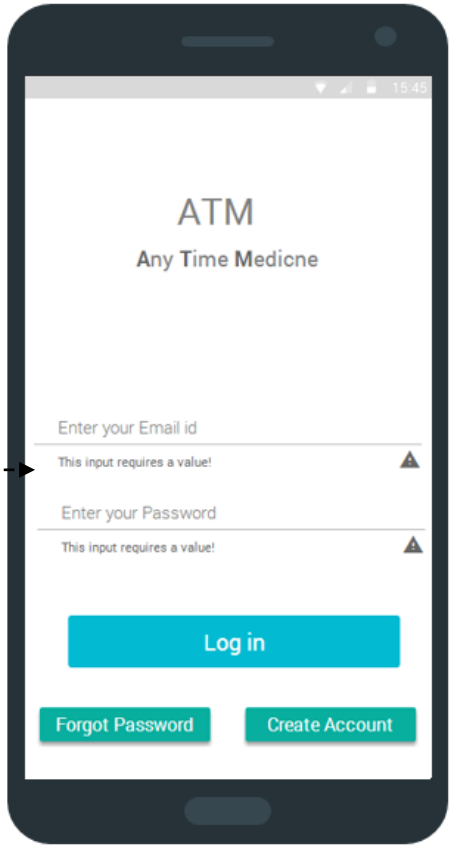
4.3.1. Design Details

4.3.1.1. Wireframe Design

Homepage



Log in Screen



4.3.1.2. Processing Details

Logout option is available on the homepage. Once the user clicks on the logout button, he is logged out and redirected to the login page.

4.3.1.3. Error to be checked for:

N/A

4.3.1.4. Output:

User is logged out and redirected to the login page

4.3.1.5. Error Capturing

N/A

4.3.1.6. Boundary and normal Operating condition

N/A

4.3.1.7. Global data structure references:

TABLE_NAME	SECTION
ATM_LOGIN_TB	2.2

4.3.1.8. Module specific data structure:

Android drawer

4.3.1.9. External Interfaces:

Android – Remote Database connection

4.3.1.10. Assumptions:

User has registered and logged into the application

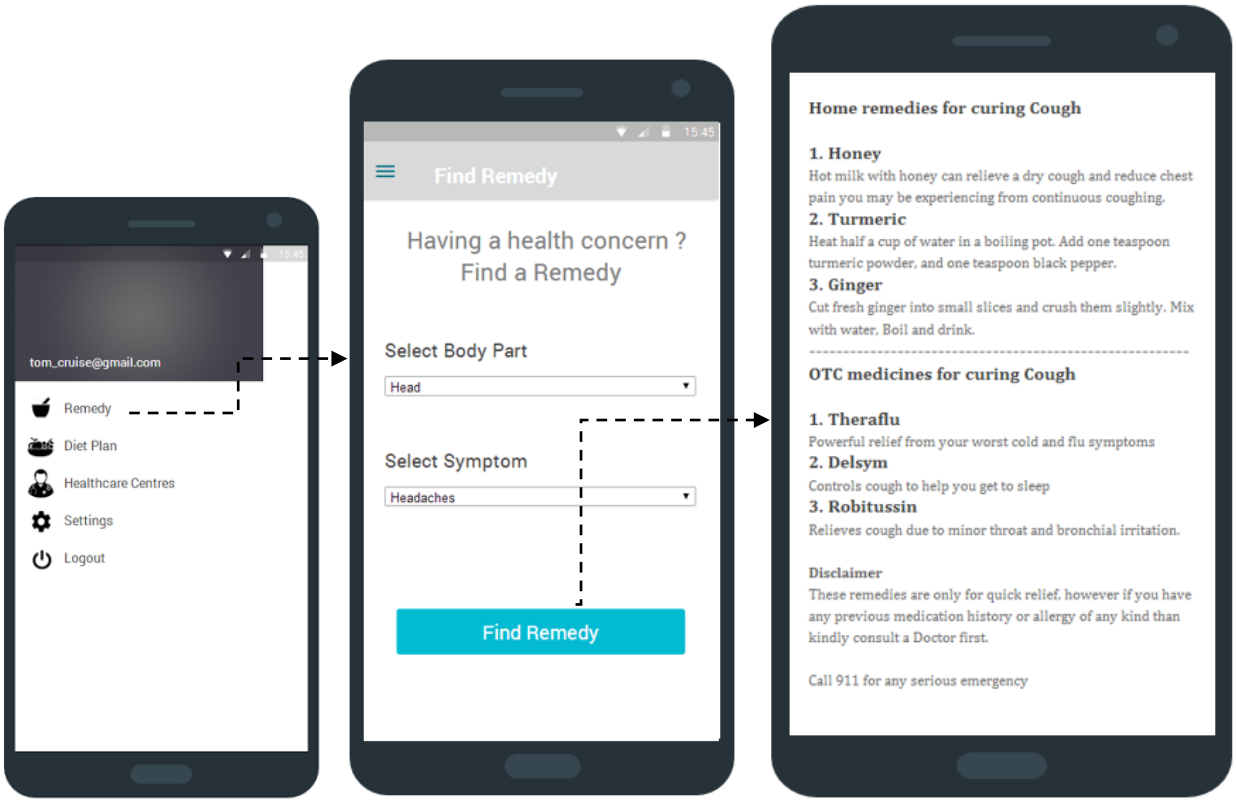
4.4.USR_004:

The user should be able to select a body part and problem associated with that body part and view medicines/remedies as per the selection

4.4.1. Design Details

4.4.1.1. Wireframe Design

Homepage Find Remedy Screen Remedy Screen



4.4.1.2. Processing Details

When the user clicks on list item “Remedy” on the homepage, he is redirected to the “Find Remedy” screen where he can select a body part and the associated symptom from the drop down list.

Once the user clicks “Find Remedy” button a background service will run that connects with remote database and fetches the remedy details associated with the specific selection made. The data is parsed at the client end and presented to the user.

4.4.1.3. Error to be checked for:

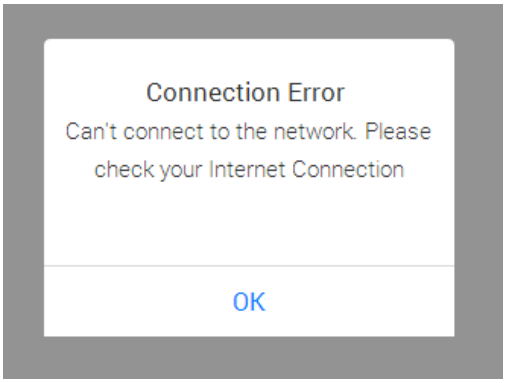
We should check for connection error.

4.4.1.4. Output:

Based on the selections made by user, remedies are displayed.

4.4.1.5. Error Capturing

If the system cannot connect to database due to connection error then the following error message is displayed:



4.4.1.6. Boundary and normal Operating condition

N/A

4.4.1.7. Global data structure references:

TABLE_NAME	SECTION
ATM_BODY_TB	2.5
ATM_REMEDY_TB	2.6
ATM_SYMPTOM_TB	2.7

4.4.1.8. Module specific data structure:

Android ListView and Button

4.4.1.9. External Interfaces:

Android – Remote Database connection

4.4.1.10. Assumptions:

User is logged into the application.

4.5. USR_005:

The user should be able to locate the nearest healthcare center covered under his insurance policy

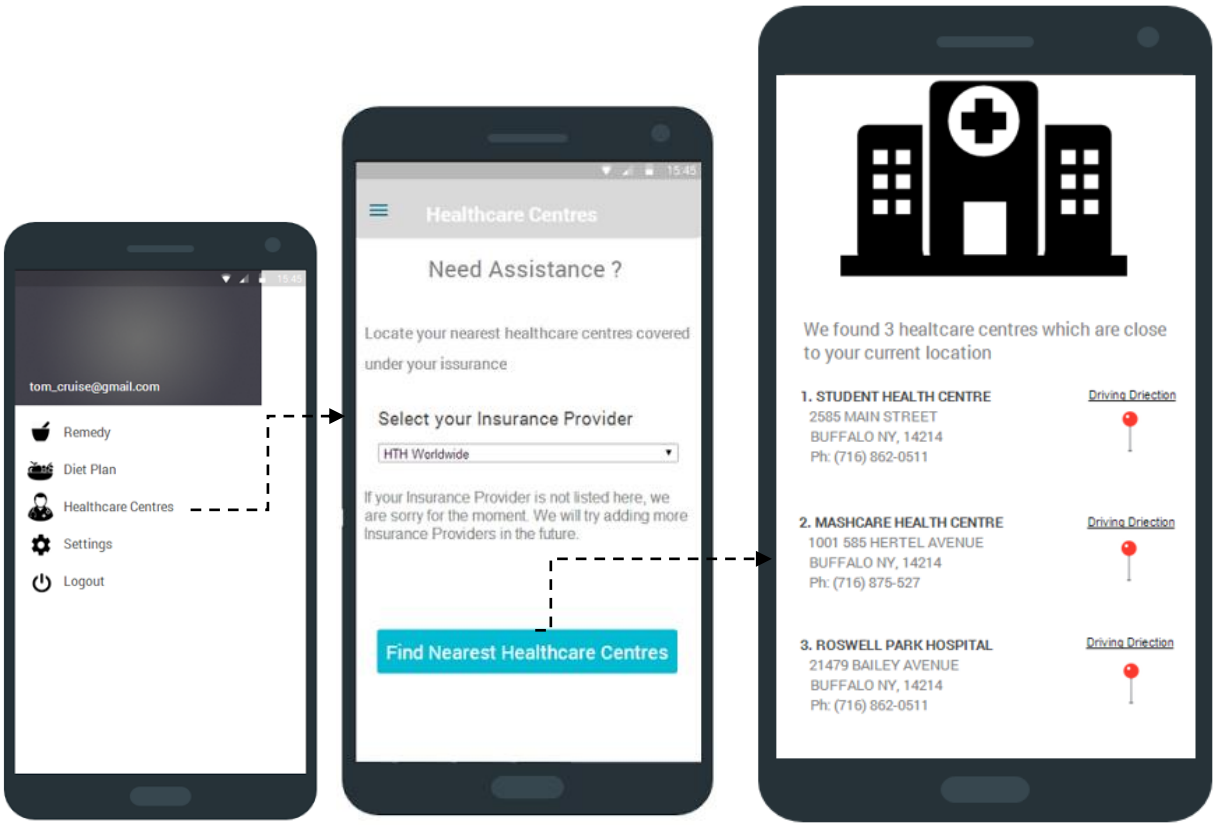
4.5.1. Design Details

4.5.1.1. Wireframe Design

Homepage

Search nearest Healthcare Center Screen

Healthcare Provider Screen



4.5.1.2. Processing Details

When the user clicks on list item “Healthcare Provider”, he is redirected to the Healthcare Provider screen where by just providing his/her insurance provider user can search for the nearest healthcare center.

A background service runs which sends user’s current GPS location information to the remote database and fetches the nearest healthcare center based upon latitude and longitude. The received location information is parsed at client end and displayed to the user.

4.5.1.3. Error to be checked for:

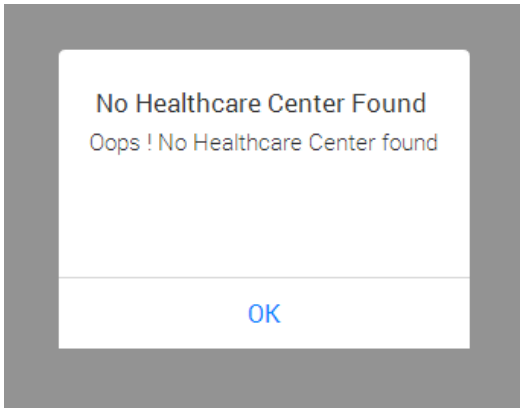
We should check for connection error. If user's GPS cannot track his location due to any error then the nearest healthcare center information will not be fetched.

4.5.1.4. Output:

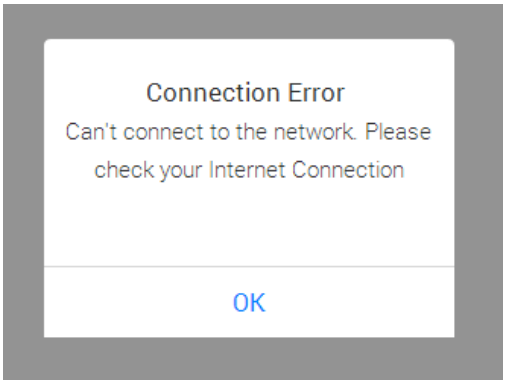
User is provided with the healthcare centers closest to his/her current location.

4.5.1.5. Error Capturing

1. If no healthcare center is found nearest to the user's location then the following error message is displayed



2. If the device cannot provide user's current location due to lack of internet connection the following error message is displayed:



4.5.1.6. Boundary and normal Operating condition

N/A

4.5.1.7. Global data structure references:

TABLE_NAME	SECTION
ATM_HEALTHCARE_CENTER_TB	2.3
ATM_INSURANCE_PROVIDER_TB	2.4
ATM_INSURANCE_PRO_HEALTHCARE_CENTER_JOIN_TB	2.9

4.5.1.8. Module specific data structure:

Android ListView and Button

4.5.1.9. External Interfaces:

Android – Remote Database connection

4.5.1.10. Assumptions:

User is logged into the application.

4.6.USR_006 & USR_007:

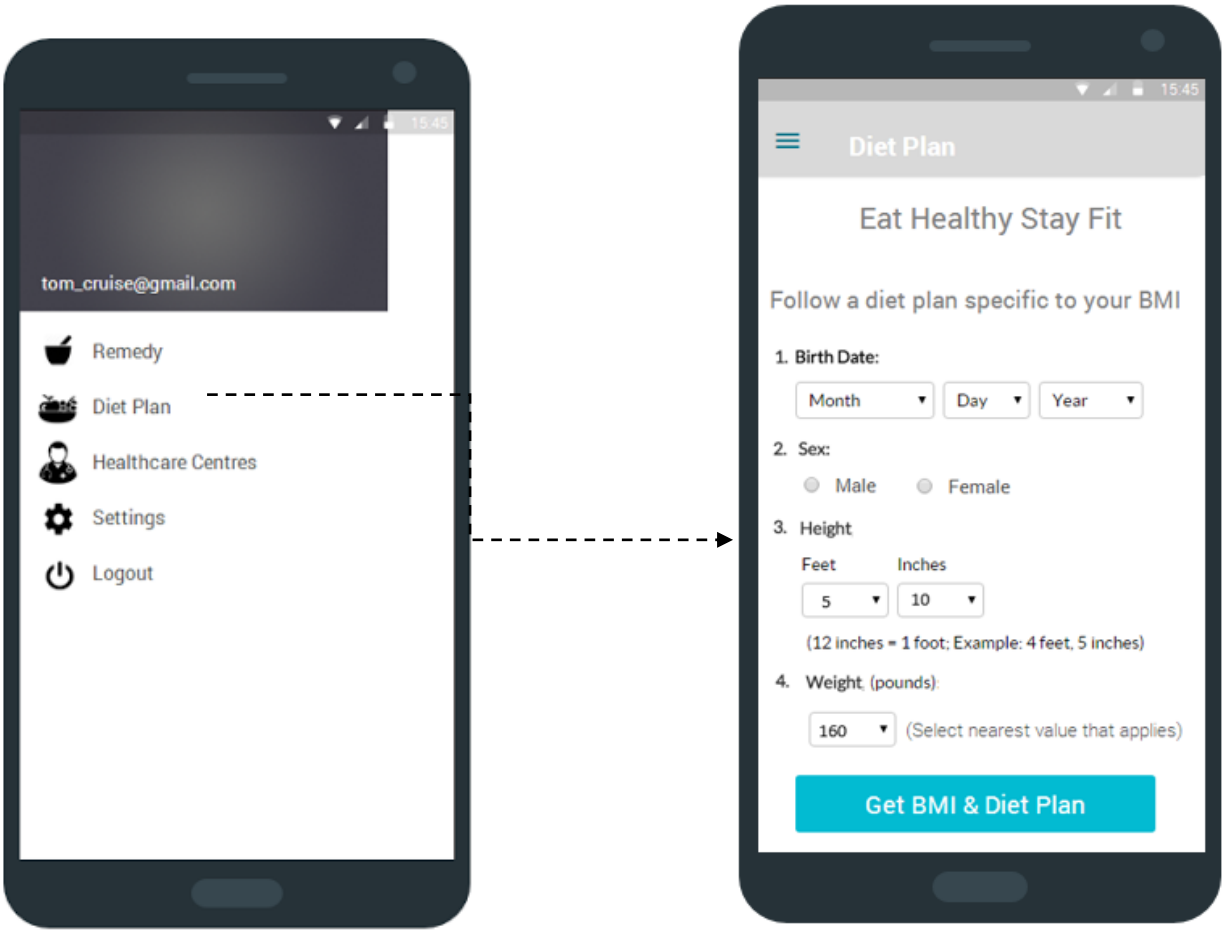
The user should be able to get the diet plan as per his/her BMI (Body Mass Index)

4.6.1. Design Details

4.6.1.1. Wireframe Design

Homepage

Diet Plan Screen



4.6.1.2. Processing Details

When the user clicks on list item “Diet Plan” on the homepage, he is redirected to the Diet Plan screen where he can input his Date of Birth, Sex, Height and Weight. Once all the required details are filled, the “Get BMI & Diet Plan” button is enabled.

When the user clicks on the button, his/her BMI is calculated and a background service runs that connects with remote database and fetches the matching available diet plan based on user’s BMI. The received information is parsed at client end and displayed to the user.

4.6.1.3. Error to be checked for:

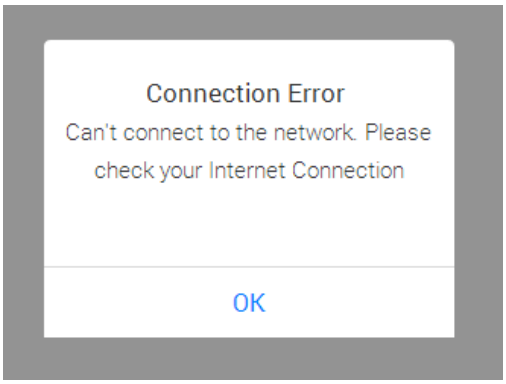
We should check for connection error.

4.6.1.4. Output:

User is provided with the BMI and diet plan according to his/her BMI.

4.6.1.5. Error Capturing

1. If the system cannot connect to database due to some connection error then following error message is displayed



4.6.1.6. Boundary and normal Operating condition

N/A

4.6.1.7. Global data structure references:

TABLE_NAME	SECTION
ATM_USER_PROFILE_TB	2.1

4.6.1.8. Module specific data structure:

Android ListView and Button

4.6.1.9. External Interfaces:

Android – Remote Database connection

4.6.1.10. Assumptions:

User is logged into the application.

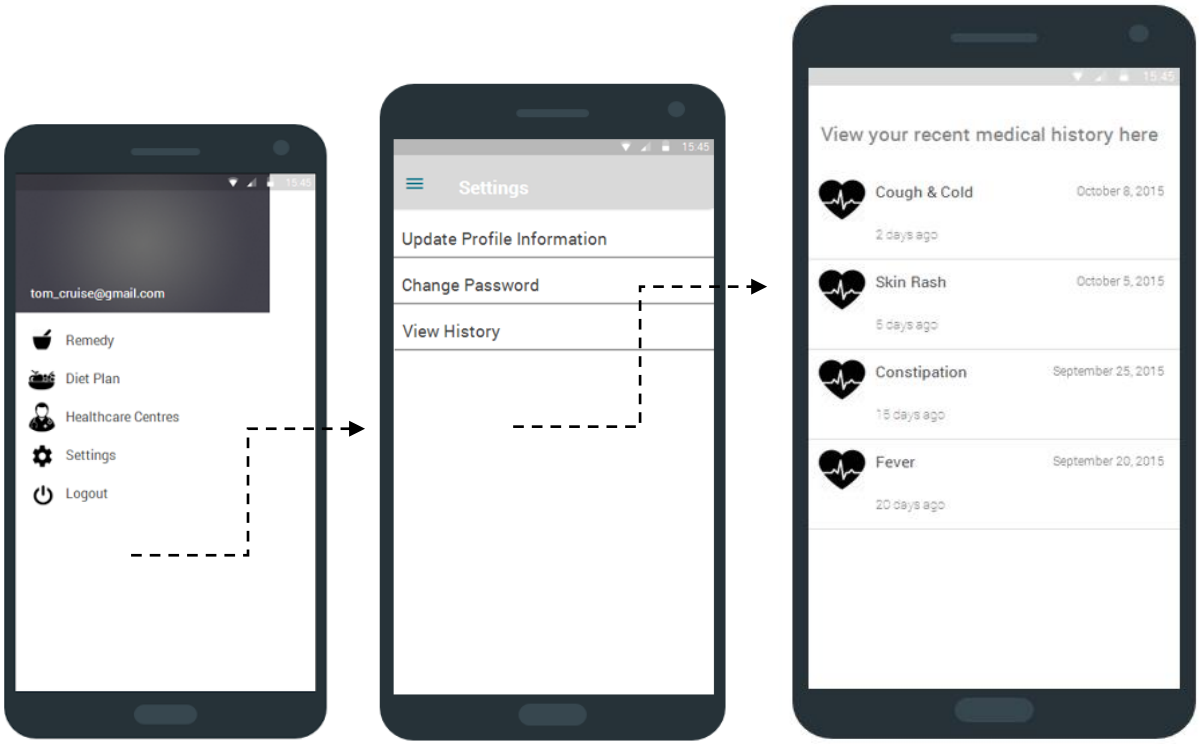
4.7.USR_008:

The user should be able to view his/her history of remedy searches

4.7.1. Design Details

4.7.1.1. Wireframe Design

Homepage Settings Screen Recent History Screen



4.7.1.2. Processing Details

When the user clicks on list item “Settings”, he is redirected to the “Settings” screen where he can click on “View History” to view his recent history of searches.

Once the user click on “View History” a background service runs that connects with remote database and fetches user’s recent history of remedy searches. The received information will be parsed at client end and shown to the user.

4.7.1.3. Error to be checked for:

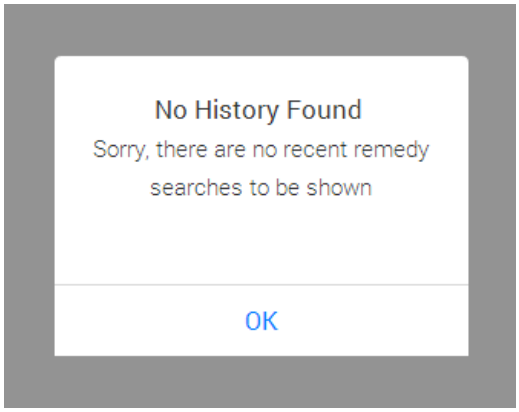
We should check for connection error.

4.7.1.4. Output:

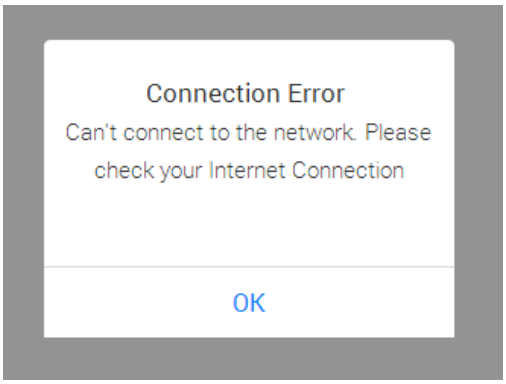
User is able to view his recent history of remedy searches.

4.7.1.5. Error Capturing

1. If no recent history is found then the following error message is displayed:



2. If the system cannot connect to database due to some connection error then following error message is displayed



4.7.1.6. Boundary and normal Operating condition

N/A

4.7.1.7. Global data structure references:

TABLE_NAME	SECTION
ATM_USER_PROFILE_TB	2.1
ATM_HISTORY_TB	2.8
ATM_REMEDY_TB	2.6

4.7.1.8. Module specific data structure:

Android ListView and Button

4.7.1.9. External Interfaces:

Android – Remote Database connection

4.7.1.10. Assumptions:

User must be logged in to perform this task.

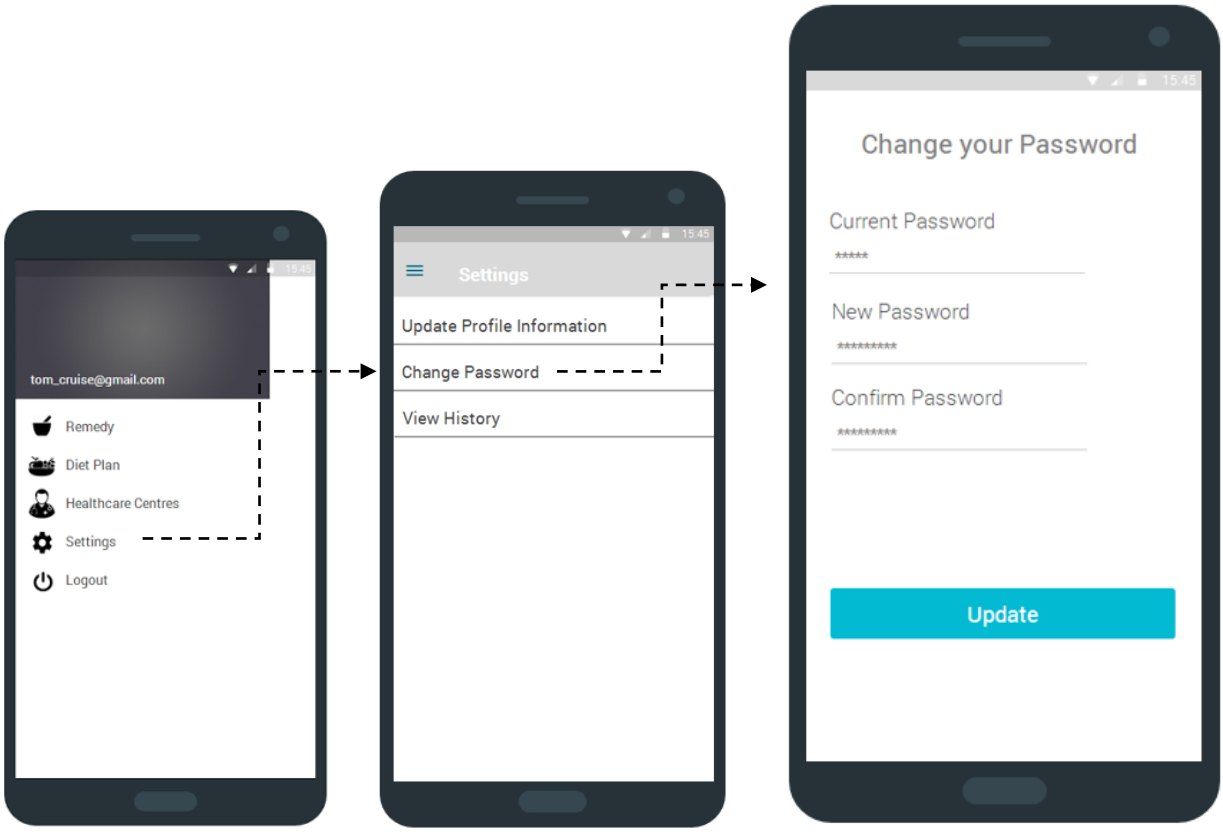
4.8. USR_009:

The user should be able to change his password

4.8.1. Design Details

4.8.1.1. Wireframe Design

Homepage Settings Screen Change Password Screen



4.8.1.2. Processing Details

When the user clicks on list item “Settings” he is redirected to the “Settings” screen where he can click on “Change Password” to change his/her password. Once the user clicks on “Change Password”, he is redirected to change password page where he enters his current password and provides a new password.

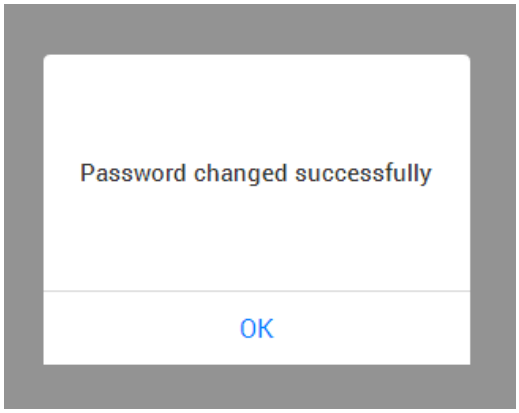
Once the user clicks on “Update” button a background service runs that connects with remote the database and validates the current password and updates the new password if validation is successful.

4.8.1.3. Error to be checked for:

We should check for connection error.

4.8.1.4. Output:

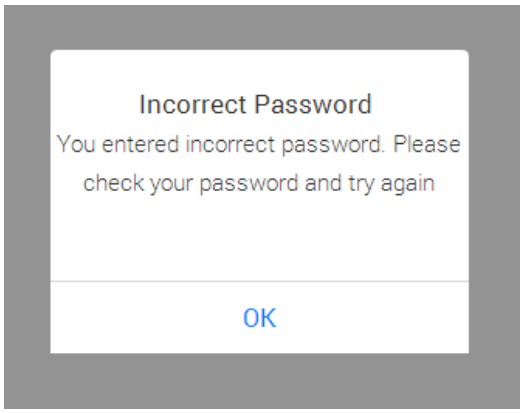
User is successfully able to change his/her password. Following success message is displayed:



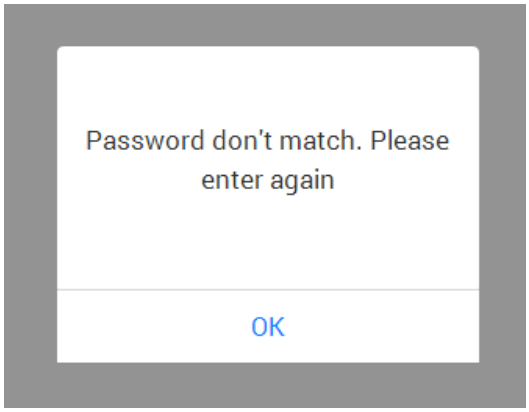
4.8.1.5. Error Capturing

Try catch block for null values. Failure message will be displayed.

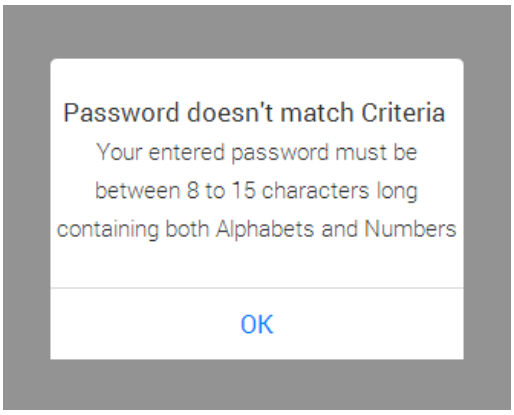
1. If INCORRECT "current password" is entered then the following error message is displayed:



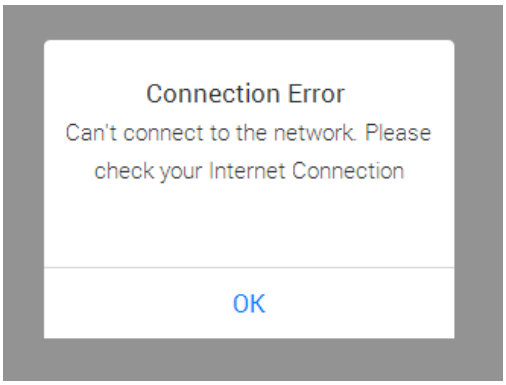
2. If upon re-entering the new password, the password does not match with the previously entered new password then the following error message is displayed:



3. If the new password doesn't meet the password criteria then the following error message is displayed:



4. If the system cannot connect to database due to some connection error then the following error message is displayed:



4.8.1.6. Boundary and normal Operating condition

Password length must be between 8 to 15 characters.

4.8.1.7. Global data structure references:

TABLE_NAME	SECTION
ATM_USER_PROFILE_TB	2.1
ATM_LOGIN_TB	2.2

4.8.1.8. Module specific data structure:

Android EditText and Button

4.8.1.9. External Interfaces:

Android – Remote Database connection

4.8.1.10. Assumptions:

User is logged into the application

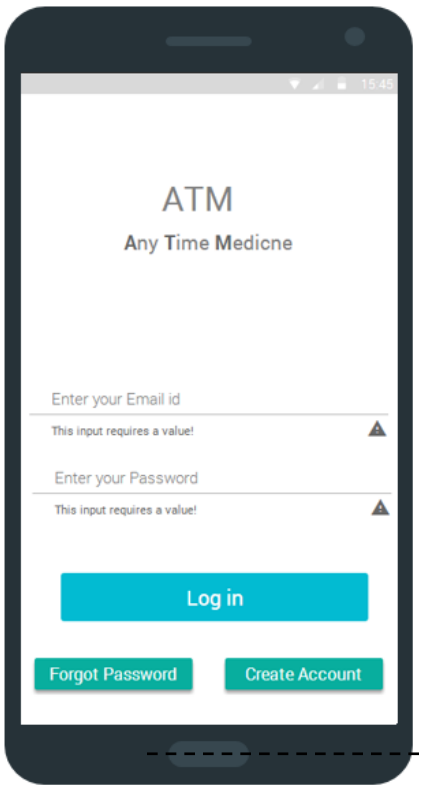
4.9.USR_010:

The user should be able to reset his password

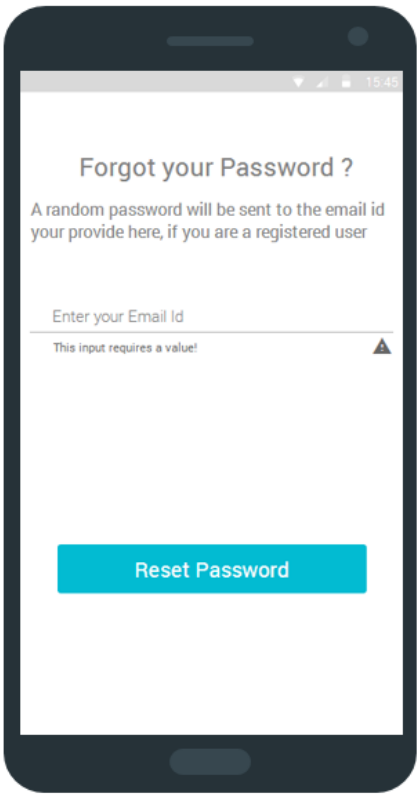
4.9.1. Design Details

4.9.1.1. Wireframe Design

Log in Screen



Forgot Password Screen



4.9.1.2. Processing Details

When the user clicks on “Forgot Password” button, he is redirected to “Forgot Password” screen where he can enter his email id to reset the password. Once the user enters his/her valid email id, the “reset password” button is enabled.

As the user clicks on the button a background service runs that connects with remote database and sends a random password on his/her email id.

User is redirected to the log in Screen after email is sent to login again with the new password.

4.9.1.3. Error to be checked for:

We should check for connection error.

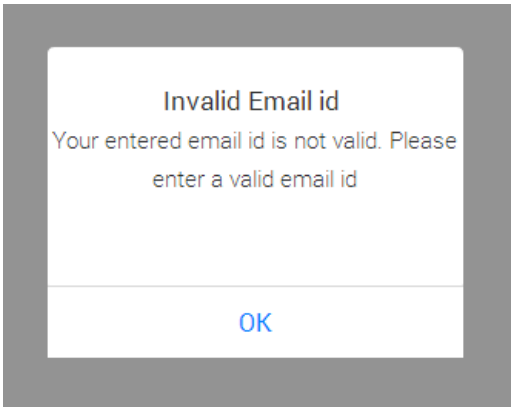
4.9.1.4. Output:

User is able to login after receiving the random password sent on his/her email id.

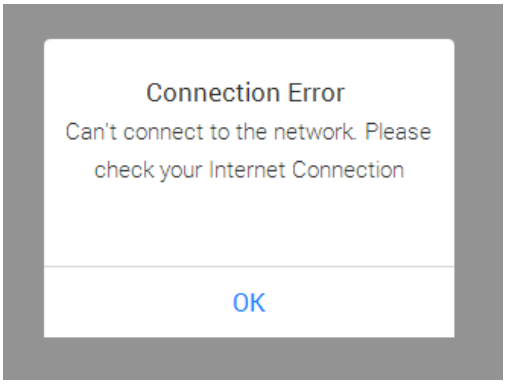
4.9.1.5. Error Capturing

Try catch block for null values. Failure message will be displayed.

1. If email id entered is not same as the one used for registration then the following error message is displayed to the user



2. If the system cannot connect to database due to some connection error then the following error message is displayed:



4.9.1.6. Boundary and normal Operating condition

Email id length must not exceed 100 characters.

4.9.1.7. Global data structure references:

TABLE_NAME	SECTION
ATM_USER_PROFILE_TB	2.1
ATM_LOGIN_TB	2.2

4.9.1.8. Module specific data structure:

Android EditText and Button

4.9.1.9. External Interfaces:

Android – Remote Database connection

4.9.1.10. Assumptions:

User knows his email id.

4.10. USR_011:

The user should be able update his profile information

4.10.1. Design Details

4.10.1.1. Wireframe Design

Homepage Settings Screen Update Profile Screen

4.10.1.2. Processing Details

When user clicks on list item “Settings”, he is redirected to the “Settings Screen” where he has the option to select “Update Profile Information” and get redirected to the “Update Profile Screen”.

Once on the “Update Profile’’ screen user can update his details like Date of Birth, Height, Weight and Name. User needs to enter his/her password to update the information provided.

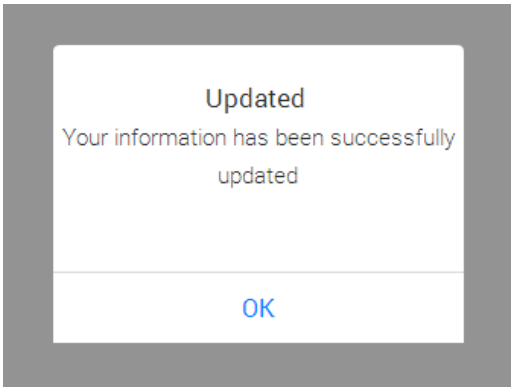
As user clicks on the “Update” button a background service runs that connects with remote database and updates user’s information.

4.10.1.3. Error to be checked for:

We should check for connection error.

4.10.1.4. Output:

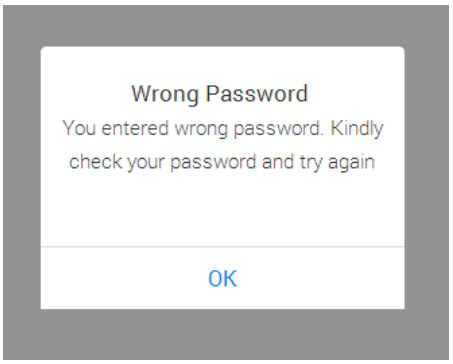
User is successfully able to update his/her information. A success message is displayed to the user:



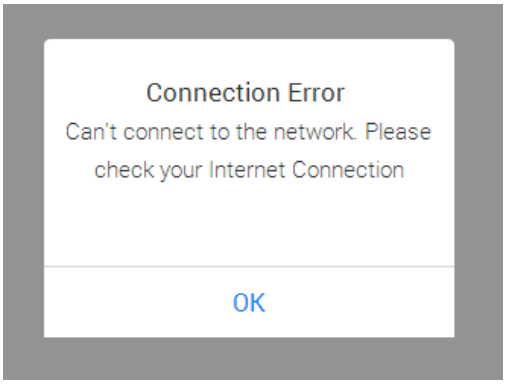
4.10.1.5. Error Capturing

Try catch block for null values. Failure message will be displayed.

1. If incorrect password is entered then the following error message is displayed to the user:



2. If the system cannot connect to database due to some connection error then the following error message is displayed:



4.10.1.6. Boundary and normal Operating condition

N/A

4.10.1.7. Global data structure references:

TABLE_NAME	SECTION
ATM_USER_PROFILE_TB	2.1

4.10.1.8. Module specific data structure:

Android EditText and Button

4.10.1.9. External Interfaces:

Android – Remote Database connection

4.10.1.10. Assumptions:

User is logged into the application.

5. TRACEABILITY TO REQUIREMENTS

Document reference Id & Description: (Doc Id from which this document is derived)		
Sl No.	**Reference document Requirement/Feature (Section ID/Name)**	**Design document (Section ID/Name)**
1	USR_001	4.1
2	USR_002	4.2
3	USR_003	4.3
4	USR_004	4.4
5	USR_005	4.5
6	USR_006	4.6
7	USR_007	4.6
8	USR_008	4.7
9	USR_009	4.8
10	USR_010	4.9
11	USR_011	4.10