UTM Computing Proceedings
Innovations in Computing Technology and Applications
Volume: 5 | Year: 2022 | eISSN No. : 2976-2278

Mobile Application of Monitoring High Blood Pressure for Senior Citizens

Janice G. Balan, Dayang N. A. Jawawi
School of Computing, Faculty of Engineering
Universiti Teknologi Malaysia
81310 Johor Bahru, Johor, Malaysia
janice.balan18@gmail.com, dayang@utm.my

***Abstract*—High blood pressure has always been a common illness especially among the senior citizens for years now. One of the most common risk of having this problem is due to age. The older a person gets, the higher the risk of getting this illness. As human gets older too, they tend to be more forgetful. With these two statements, if senior citizens were to have a high blood pressure, they may forget to take their medicine in addition to not knowing how to search beneficial information regarding blood pressure on the internet as they are not as tech-savvy. Therefore, monitoring high blood pressure is needed as without the right care, problems from high blood pressure may occur. A mobile application would especially be helpful as in these days, everyone has a smartphone. Therefore, an application called "HyperRemedy" is proposed. The aim of this project is to develop the "HyperRemedy" mobile application for senior citizens to track and monitor their blood pressure reading, receiving monthly graph that is easy to read, receiving extra information on their food intake based on the readings and also to set a reminder for their medicine intake in an aim to aid the senior citizens with high blood pressure illness. This application would be beneficial for the senior citizens who are not tech-savvy to receive extra information such as the food intake. Not only that, the project also would be able to help senior citizens to not forget on their medicine intake and aid them to record their daily blood pressure readings. As for academic importance, this project would be able to test the skills and knowledge learned over the past year on software development. The methodology that was chosen in order to develop this proposed project is Agile-Kanban methodology because it suits small projects like the proposed application. The software used during the development phase is Flutter Framework and Firebase with dart language. After finishing the full implementation of the project, HyperRemedy was tested using the black-box testing and user acceptance test (UAT). Overall, user gave good feedback from the UAT. The minor bugs that were found were fixed before releasing it to the users.**

Keywords- high blood pressure, senior citizen, monitoring record, data analysis, mobile application, system development, medicine reminder

I. Introduction

High blood pressure has always been a common illness especially among the senior citizens for years now. It is a common condition in which the long-term force of the blood against the artery walls is high enough that it may eventually cause health problems such as heart disease. A person can have high blood pressure for years without any symptoms. However, high blood pressure can easily be detected with the right knowledge. The symptoms of high blood pressure include headaches, shortness of breath or nosebleed. However, the symptoms may vary based on different people. Therefore, it is important to constantly be aware of your health.

The meaning of monitoring is to observe and check the progress. Therefore, monitoring high blood pressure refers to observing and checking the progress of the user with high blood pressure. Monitoring high blood pressure is needed as without the right care, problems from high blood pressure reading may arises. By monitoring the blood pressure reading, doctors or patients are able to prevent any harmful situation from happening such as heart failure or stroke.

Having a mobile application that serves the purpose of monitoring blood pressure reading can be beneficial to many. Everyone in this century, no matter the age, owned a mobile. It is small, handy and are able to keep many data in it. By having a mobile application that can monitors the blood pressure reading, user will be able to observe their readings monthly and are able to take the extra precautions if needed from the reading. The data will also not be lost compared to writing it on a piece of paper. It is especially useful for the senior citizen as senior citizen mostly are forgetful. With the mobile application, the problem of forgetting to record their readings can be solved.

One of the main risk factors for having a high blood pressure is age. The risk of getting the illness increases the more a person age. Not only that, as people get older, they might experience more frequent of instances of memory loss, and this is a normal part of aging. Combining these both problems, most senior citizens with high blood pressure tend to be more forgetful.

Having such dangerous and harmful illness with the tendency of being forgetful is extremely risky. One of the problems that arises from this is they may forget to take their medicines on time. They may also forget to get their new sets of medicines from the pharmacists before it runs out for a long period.

Not only that, most of them also need advice on what to eat and what to not eat based on their blood pressure readings. However, most senior citizens have a hard time searching more information on the google about the illness as they are not as tech-savvy. They would always need help from other person like their children to help them browse the internet or they would travel and go to see the doctor to receive advice even though the information can be found on the internet.

II. Related Works

This subsection of the chapter will discuss on the current system analysis for monitoring the high blood pressure using existing mobile application like “MyTherapy” and “Blood Pressure” which both each had one million plus downloads overall.

A. MyTherapy

MyTherapy is a mobile application that reminds its user on taking their medication or pills [1]. The app is useful for patients who know their medications well and need reminders throughout the day. This app has many useful features such as medication tracking history, pill reminder, reports for sharing with invitation, and simple overview of the areas of treatment which the users need for improvement.

Overall, from this application, although it serves many useful features for senior citizens with high blood pressure, however the interfaces are not catered to users of old age. Not only that, there also need to be an indication on which information needed to be entered and the meaning of each of the input such as the PRN. Most people may not be aware of the meaning of this word. PRN means Pro re nata which translate to as needed.

B. Blood Pressure

Blood pressure is a mobile application with over ten million downloads on the play store [2]. This app has multiple features such as which it focuses on saving the user’s measurement for blood pressure readings. User can also export the data of measurements.

Overall, from this application, its main purpose is to monitor the blood pressure application. However, improvement can be made especially on the interface of the app. The interface is not suitable for senior citizens as it is hard to identify what each symbol meant. Next, the filter function also must be added. Furthermore, more features can be added into the system so it can serve more than just monitoring the blood pressure.

C. Comparison between Existing System

Table 1 shows the comparison between MyTherapy, Blood Pressure and HyperRemedy based on the features. The features are account, ads, reminder function, adding blood pressure readings, reports, graph filter, additional information on the blood pressure, inventory of medicine reminder and input symptoms.

TABLE I. Comparisons of Features

Features	**MyTherapy**	**Blood Pressure**	**HyperRemedy**
Account	Free Account	Free Account	Free Account
Ads	**None**	Yes	**None**
Reminder function	Yes, various reminders with multiple built in inputs available	Only normal reminder with time and description	Yes, various reminders with multiple built in inputs available
Adding blood pressure readings	Yes	Yes	Yes
Reports	Pdf	CSV, Pdf	Pdf
Graph filter	**None**	Yes, but limited only up to 30 days	Yes, various necessary filters like weekly, last 30 days, last year and overall
Additional information on the blood pressure	**None**	**None**	Yes, such as the food suggestion based on blood pressure readings
Inventory of medicine reminder	Yes	**None**	Yes
Input Symptoms	Yes, but limited to the choices given by the app	**None**	Yes, unlimited choices where user can input themselves how they are feeling

III. Methodology

Agile Kanban is chosen as the methodology in the development of the Mobile Application of Monitoring High Blood Pressure for Senior Citizens. Agile Model refers to a software development approach that is based on iterative development [3]. In this model, the tasks that were elicit from the requirement phases are broken into smaller iterations. The plans on the number of iterations and its time are fixed since the beginning [4].

A. Requirement Phase

Requirement analysis is an important phase in the Software Development Life Cycle as it is the phase where the developer will be able to identify the functional and the non-functional requirement for the HyperRemedy mobile application.

A survey among 20 senior citizens was done as one of the technique use to gather requirements details from the users. Figure 1 shows the use case diagram that was designed based on the survey done and the requirements gathered.

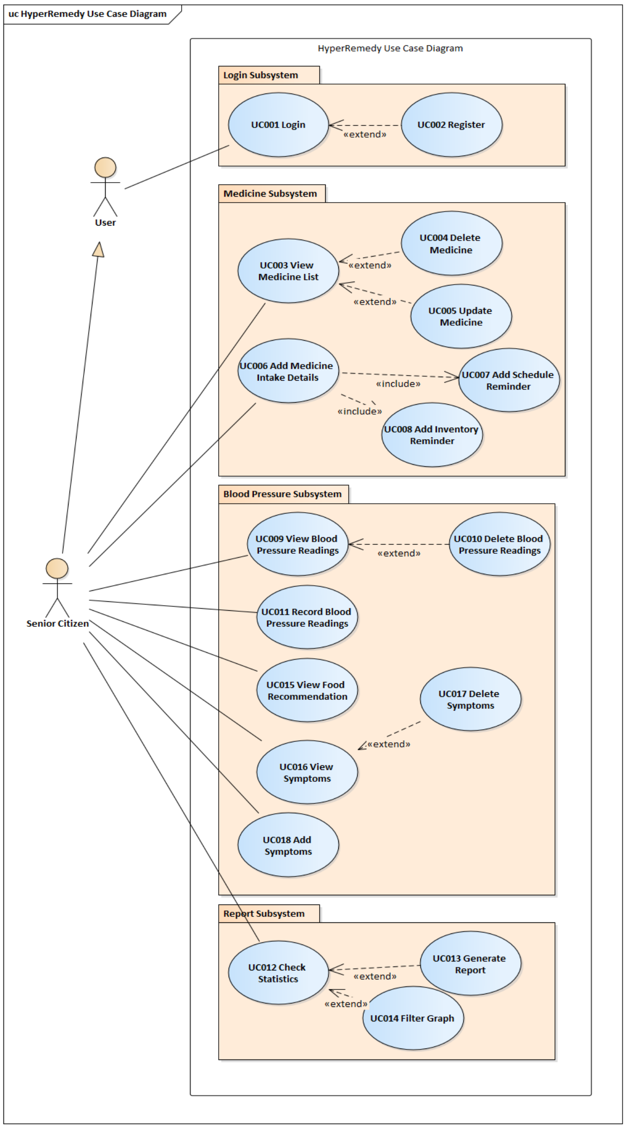


Figure 1. Use Case Diagram

B. *System Design Phase*

The architectural style that will be applied to the HyperRemedy application is Model-view-viewmodel (MVVM) for the client side or the front-end and Model View Controller (MVC) for the backend or the server side. The View in MVVM handles what the user sees and interacts with, View Model synchronizes between UI and what is going on behind the scenes and Model will take requests from View and forward them to the service. In this case, the Model will communicate with the controller on the backend to process the request such as POST or GET. The controller will then interact with the model to retrieve or send data to the database. Figure 2 shows the implementation of both MVVM and MVC on the HyperRemedy application software.

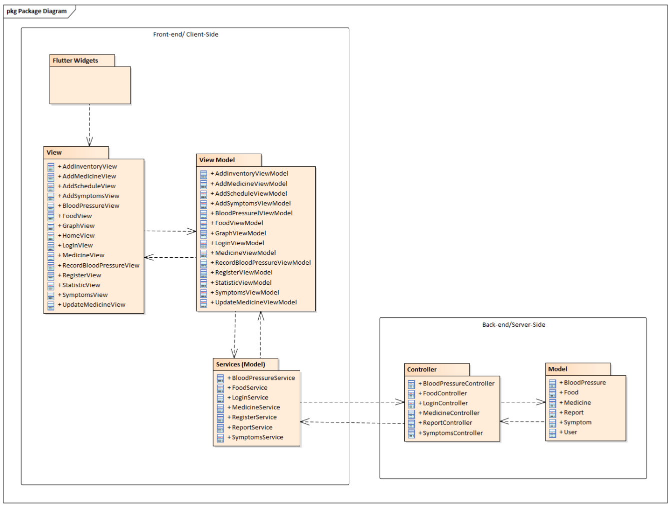


Figure 2. Software Architecture

For the system architecture, the server side will be implementing MVC structure. The flutter widgets will be located in the mobile. When a user interacts with the mobile screen, a request will be sent. From here, the REST Services which represents the API Gateway will send request to the controllers on the server side based on the subsystem of the controller. The controller will then call the model which will update the database table in the Firebase database. The controllers consist of four subsystem controllers which are Blood Pressure Subsystem controllers, Report Subsystem controllers, Medicine Subsystem controllers and Login Subsystem controllers. The models consist of four subsystem models which are Blood Pressure Subsystem models, Report Subsystem models, Medicine Subsystem models and Login Subsystem models. Figure 3 shows the System Architecture model of the HyperRemedy application.

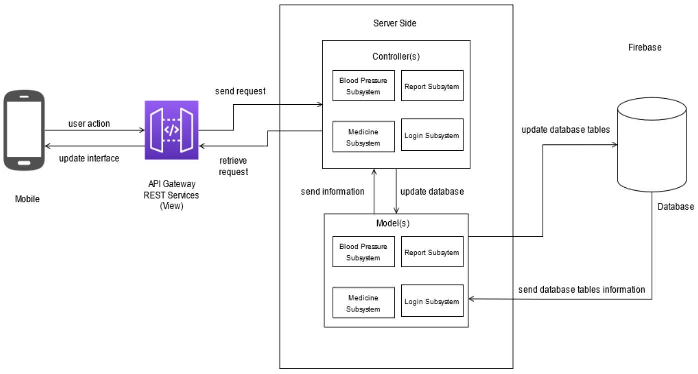
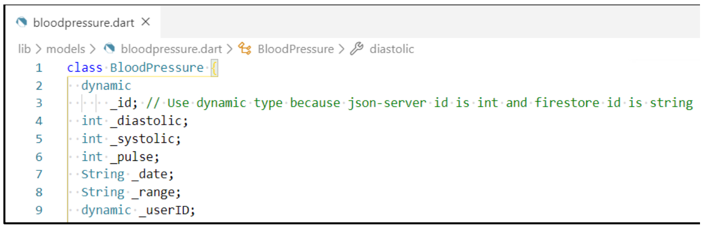


Figure 3. System Architecture

IV. Implementation

The system is developed using Flutter Framework for the front-end development and Firebase and Express JS for the back-end development.

MVVM architecture style is used for the client-side. It consists of 3 components which are model, view, and view model. MVVM allows the code to be reusable and easily changed. The code of the application is broken based on the MVVM architecture. Figure 4 shows the snippet of the Blood Pressure model class. This model class defined all the attributes of the blood pressure model based on the firebase database. This model class is also used to convert json data to dart language.

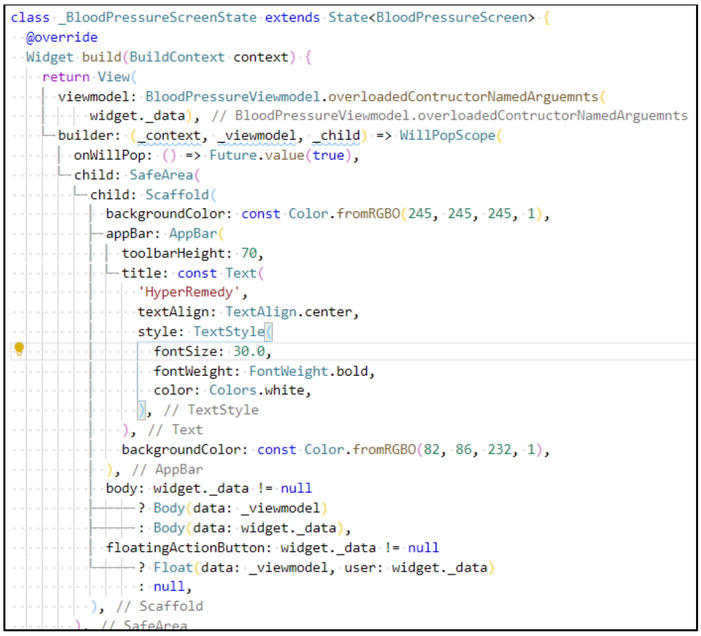


bloodpressure.dart ×
lib > models > bloodpressure.dart > BloodPressure > diastolic

```dart
class BloodPressure {
  dynamic
      _id; // Use dynamic type because json-server id is int and firestore id is string
  int _diastolic;
  int _systolic;
  int _pulse;
  String _date;
  String _range;
  dynamic _userID;
```

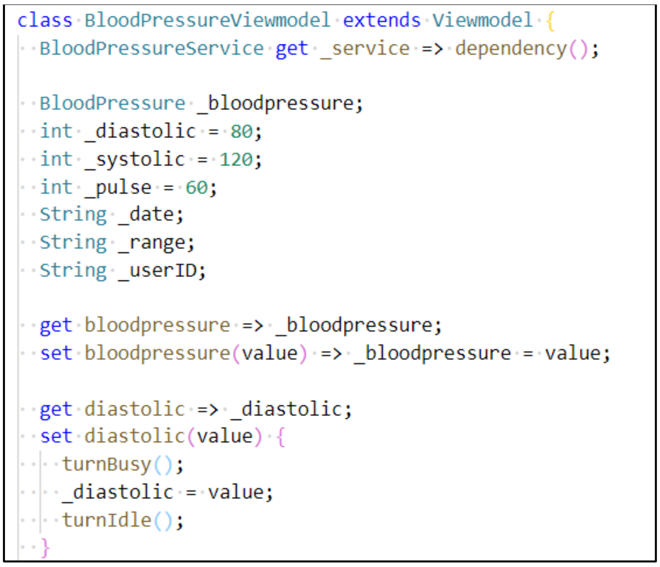
Figure 4. Blood Pressure Model

Figure 5 shows the snippet code of the Blood Pressure View class which is where the code for the interface of the system is coded. Figure 6 shows the Blood Pressure View Model which holds all the functions and methods required for the blood pressure functionality.



```dart
class _BloodPressureScreenState extends State<BloodPressureScreen> {
  @override
  Widget build(BuildContext context) {
    return View(
      viewmodel: BloodPressureViewmodel.overloadedContructorNamedArguemnts(
          widget._data), // BloodPressureViewmodel.overloadedContructorNamedArguemnts
      builder: (_context, _viewmodel, _child) => WillPopScope(
        onWillPop: () => Future.value(true),
        child: SafeArea(
          child: Scaffold(
            backgroundColor: const Color.fromRGBO(245, 245, 245, 1),
            appBar: AppBar(
              toolbarHeight: 70,
              title: const Text(
                'HyperRemedy',
                textAlign: TextAlign.center,
                style: TextStyle(
                  fontSize: 30.0,
                  fontWeight: FontWeight.bold,
                  color: Colors.white,
                ), // TextStyle
              ), // Text
              backgroundColor: const Color.fromRGBO(82, 86, 232, 1),
            ), // AppBar
            body: widget._data != null
                ? Body(data: _viewmodel)
                : Body(data: widget._data),
            floatingActionButton: widget._data != null
                ? Float(data: _viewmodel, user: widget._data)
                : null,
          ), // Scaffold
        ), // SafeArea
```

Figure 5. Blood Pressure View



```dart
class BloodPressureViewmodel extends Viewmodel {
  BloodPressureService get _service => dependency();

  BloodPressure _bloodpressure;
  int _diastolic = 80;
  int _systolic = 120;
  int _pulse = 60;
  String _date;
  String _range;
  String _userID;

  get bloodpressure => _bloodpressure;
  set bloodpressure(value) => _bloodpressure = value;

  get diastolic => _diastolic;
  set diastolic(value) {
    turnBusy();
    _diastolic = value;
    turnIdle();
  }
```

Figure 6. Blood Pressure View Model

The system main functions of this mobile application can only be tested on the interface using Emulator of any Android phones. There are total of five cards in the main interface of the system. The medicine card allows user to add medicine, add medicine reminder, add inventory reminder, update medicine, delete medicine and view medicine lists. The symptoms card allows user to view, add and delete symptoms. The food suggestions card allows user to view the recommended food based on blood pressure range. The blood pressure card allows user to add, view and delete blood pressure readings. Last but not least, the graph and statistics card allow user to view their statistic for blood pressure and symptoms, export graph to pdf and to filter the graph. Figure 7 shows the main page of the HyperRemedy application which can access all of the system main functions.

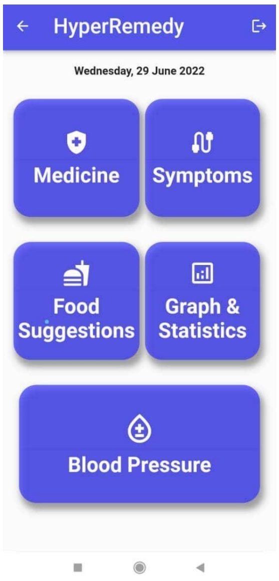


Figure 7. Main Interface of System

V. Testing and Result

Software testing phase is the final and most important phase in the software development life cycle. The testing is important because it helps discovers defects or bugs before the software is delivered to the client, which will guarantee the quality of the mobile application. This would make the app more reliable and easy to use. Thoroughly tested software ensures the reliable and high-performance software operation. The two software testing technique that were used are black-box testing and User Acceptance Testing (UAT).

A. Black-box Texting

Black box testing involves testing a system with no prior knowledge of its internal workings [5]. A tester will provide an input and observes the output generated by the system under testing. This allows to identify how the system responds to expected and unexpected user actions, its response time, usability issues and reliability issues. Table 2 shows the black box testing for add medicine use case.

TABLE II. Black-box Test Case Example

Test Case ID	**Input data**	**Expected result**	**Actual result**	**Pass / Fail**
TC006_01_01	14	Input accepted	Input accepted	Pass
TC006_01_02	abd	Invalid input	Invalid input	Pass
TC006_01_03	Abd12	Invalid input	Invalid input	Pass
TC006_01_04	!@##	Invalid input	Invalid input	Pass
TC006_01_05	!@!#@1232	Invalid input	Invalid input	Pass

B. User Acceptance Testing (UAT)

User acceptance testing (UAT), also known as application testing or end-user testing, is a step of software development testing in which the product is tested by its intended audience in the real world. UAT is frequently conducted at the conclusion of the software testing process, before the tested programme is delivered to its target market [6]. UAT ensures that software can handle real-world activities and execute to development criteria.

For the HyperRemedy system, the end users is senior citizens. The UAT was done to two senior citizens where they are given the list of task to be done. Next, the feedback from them is observed. Overall, the two senior citizens gave good feedback. Some of the feedback includes the mobile application has simple interface which would be easy for senior citizens to use. Table 3 shows the lists of task that was asked to be done by two of the senior citizens.

TABLE III. List of Task for UAT

Tasks	**Description**
1	Please create new account
2	Please login into the account using the newly created account
3	Please add a medicine and inventory reminder
4	Please update medicine
5	Please delete on a medicine
6	Please add a medicine reminder
7	Please add blood pressure readings
8	Please delete blood pressure readings
9	Please add symptoms
10	Please delete symptoms
11	Please view food recommendations
12	Please add multiple blood pressure readings and symptoms, then view the graph
13	Please filter the graph

Table 4 shows the user profile of the two senior citizens that tested the application. Senior Citizen 1 is a male age 62 years old while senior citizen 2 is a female aged 62 years old. Both of the senior citizen have high blood pressure. As for the experience on technologies and mobile application, senior citizen 1 is in the intermediate level while senior citizen 2 is in the beginner level.

Table 5 shows the result and the comments and suggestions between two of the senior citizens when testing the application based on the instructions that were given. However, from the User Acceptance Test, it can be observed that senior citizen 2 takes longer time to do each instruction compared to senior citizen 1 who has more experience. It can also note that senior citizen 2 are able to complete most of the tasks with little to zero help which shows that the application is simple and straightforward for technology illiterate people.

TABLE IV. User Profile of Tester

Details	**Senior Citizen 1**	**Senior Citizen 2**
Age	62 years old	62 years old
Gender	Male	Female
Experience on technologies	Intermediate	Beginner
Experience on mobile application	Intermediate. Have experience in using complicated mobile app like Banks app. However, struggles with readings words on screen.	Beginner. Uses application like Whatsapp, Facebook and Youtube only.
Job (current or before retire)	Manager in Business	Babysitter/ Housewife
High Blood Pressure	Yes	Yes

TABLE V. RESULT FROM UAT

No	Description	Result	Comments/Suggestions
1	Please create new account	Pass	Easy to register because not too much input
2	Please login into the account using the newly created account	Pass	NA
3	Please add a medicine and inventory reminder	Pass	Straightforward and input that were asked to be keyed in is simple
4	Please view the newly added medicine	Pass	After adding new medicine, can straight away view. Easy
5	Please update medicine	Pass	Takes longer time to do
6	Please delete on a medicine	Pass	Straightforward with the delete icon
7	Please add a medicine reminder	Pass	Feels like normal alarm, easy to add
8	Please add blood pressure readings	Pass	Can change the color and position of the input readings button
9	Please view the newly added blood pressure readings	Pass	The colors can maybe change so that only the status is colored
10	Please delete blood pressure readings	Pass	Easy
11	Please add symptoms	Pass	Straightforward because very little input
12	Please view the newly added symptoms	Pass	Straightforward
13	Please delete symptoms	Pass	Straightforward
14	Please view food recommendations based on range	Pass	Good description on the food
15	Please add multiple blood pressure readings and symptoms, then view the graph	Pass	The graphs shows clear depiction and is simple enough to understand. The average, highest and lowest is good way of knowing the overall condition
16	Please filter the graph	Pass	Straightforward
17	Please export the graph to pdf	Pass	Straightforward

VI. CONCLUSION

The mobile application called HyperRemedy has achieved all the objectives stated in the project objectives. This system can help senior citizens to monitor their blood pressure readings. As can be seen from the users' feedbacks from the user acceptance testing and the black-box testing that the functional and non-functional requirements of the project are complied with the Software Requirements Specification (SRS) document. While the existing application contains the functionality for monitoring blood pressure or setting reminder for medicine, HyperRemedy managed to integrate these two functions together with analysis of blood pressure readings in graph and recommendation of food based on the blood pressure range. The application was developed using Flutter Framework, Firebase and Express JS. Overall, there are improvements that can be made in order to make the system higher quality such as the system can be improved to be able to work in offline as right now, in order for the system to function, an internet connection is needed since it is connected to the live Firestore. By allowing offline, user would still be able to use the application even when they are at a place with no internet coverage.

ACKNOWLEDGMENT

In completing this project, we was in contact with many people, researchers and academicians. They have contributed towards my understanding and thoughts. Next, we would also like to show gratitude to our fellow course mates who have also helped me in completing this project on proposed development.

REFERENCES

[1] Mirkov, S. (2018). MyTherapy Meds & Pill Reminder app review | Health Navigator NZ. [online] Health Navigator New Zealand. Available at: https://www.healthnavigator.org.nz/apps/m/mytherapy-meds-pill-reminder-app/

[2] Szymon, K. (n.d). *Blood Pressure*. [online] Play.google.com. Available at: https://play.google.com/store/apps/details?id=com.szyk.myheart&hl=en&gl=US

[3] Dingsøyr, T., Nerur, S., Balijepally, V. and Moe, N.B. (2012). A decade of agile methodologies: Towards explaining agile software development.

[4] Kirovska, N. and Koceski, S. (2015). Usage of Kanban methodology at software development teams. *Journal of applied economics and business*, *3*(3), pp.25-34.

[5] Nidhra, S., 2012. Black Box and White Box Testing Techniques - A Literature Review. International Journal of Embedded Systems and Applications, 2(2), pp.29-50.

[6] Otaduy, I. and Diaz, O., 2017. User acceptance testing for Agile-developed web-based applications: Empowering customers through wikis and mind maps. Journal of Systems and Software, 133, pp.212-229.