INFORMATICS INSTITUTE OF TECHNOLOGY

INFORMATICS INSTITUTE OF TECHNOLOGY

In collaboration with

UNIVERSITY OF WESTMINSTER, UK

Health Status Assessment in Remote Patient Monitoring Systems using Hybrid Machine Learning

A Dissertation by
Mr. Hammadh Yusuf Mohamed Arquil
w17617808 / 2018128

Supervised by
Ms. Divya Premanantha

May 2023

Submitted in partial fulfillment of the requirements for
the BSc (Hons) Computer Science degree at
the University of Westminster.

ABSTRACT

In the healthcare sector, the field of remote patient monitoring and telemedicine is expanding and being developed quickly, giving healthcare professionals a tool to remotely monitor patient health data however most machine learning models utilized to generate patient status warnings are inefficient and have issues with performance, which are two drawbacks of the existing approaches used for assessing patient health status in remote patient monitoring systems.

In the past, remote patient monitoring systems that monitor vital signs have not used hybrid machine learning models for health status assessments. The author has decided to implement a novel hybrid machine learning model to perform health status assessments on the patient's vital signs because hybrid models have an excellent track record of handling similar challenges. The author also implements a vital sign forecasting model in order to use the forecasted vitals on the hybrid model to forecast the future health status of the patient as well.

The author was able to experiment and develop a hybrid health status assessment model where the evaluation metric results were a classification **accuracy** of **94%** gained utilizing the prototype model for hybrid health status assessment. The **precision** and **recall scores**, which are **96%** and **97%** respectively with an **F1 score** of **96%**. **Cohen's kappa score** was also generated showcasing a result of **91.3%** for the hybrid model. For the vital sign forecasting model developed, the author received results of an **MAE score** of **0.11%** and an **RMSE score** of **0.03%** when tested on the dataset which showcased satisfactory improvement gained.

Keywords: Remote Patient Monitoring, Hybrid Machine Learning, Health Status Assessments, Vital Signs, Telehealth, Vital Sign Forecasting

Subject Descriptors:

- Computing methodologies — Machine learning
- Applied computing — Life and medical sciences — Health informatics.
- Computing methodologies — Machine learning — Machine learning algorithms
- Applied computing — Life and medical sciences — Consumer health.

DECLARATION

I hereby declare that this dissertation, and its associated sub-components, are the product of my own research and have not been, nor are currently being, presented as content for any degree or other qualification program to any other university or institution. Sources from which facts were taken have been cited properly.

Student Name: Hammadh Yusuf Mohamed Arquil
Registration Number: 17617804 // 2018128

Date: 05/05/2023

Signature:

ACKNOWLEDGEMENT

I truly appreciate the individuals in my life that have enabled me to go beyond my prior limitations and reach my current level of success, including completing this research project for my final year. I am grateful to my supervisor, Ms. Divya Premanantha, for her assistance in developing and refining my project idea.

My appreciation also goes to all the evaluators, who took their time to review my work and offer helpful advice for improvement. I am grateful to the survey respondents, including industry and academic professionals, as well as my peers at IIT, for their input during the project research and development. I am sincerely thankful to the Informatics Institute of Technology lecturers and the management for the opportunities and support they have provided me throughout my undergraduate degree. I am also grateful to my friends from the university, whose encouragement and positive influence have been essential in my development.

Lastly, my deepest gratitude goes out to my parents for their support in achieving my aspirations, and to my beloved family, friends, and relatives, whose support has been a source of strength during challenging moments.

Thank You!

- HAMMADH

CONTENTS

LIST OF TABLES

LIST OF FIGURES

TABLE OF ACRONYMS

Acronym	Description
RPM	Remote Patient Monitoring
ML	Machine Learning
DNN	Deep Neural Network
MIT	Massachusetts Institute of Technology
PHC	Portable Healthcare System
RNN:	Recurrent neural network
LSTM	Long Short-Term Memory
OOADM	Object-oriented Analysis and Design Methodology
RF	Random Forest
DT	Decision Trees
IOT	Internet of Things
MLP	Multi-layer Perceptron
GB	Gradient Boosting
SVM	Support Vector Machines
SSADM	Structured Systems Analysis and Design Methodology

CHAPTER 1: INTRODUCTION

1.1 Chapter Overview

In this chapter, the author provides an overview of the current state of remote patient monitoring systems in healthcare delivery proposing a hybrid machine learning model to automate the health status/triage risk assessment process of a patient to alert the healthcare providers. Also defined in this chapter by the author are the problems found, the aims and objectives of the project, challenges and the proposed contribution to the bodies of knowledge.

1.2 Problem Background

As monitoring and treatment of diseases require people to go frequently to hospitals, the healthcare industry dramatically needs to change and constantly advance towards a more technological transition in order to ease their difficulty. Remote monitoring and remote healthcare are rapidly growing fields of study and research (Malasinghe, Ramzan and Dahal, 2019). The use of technology and methods to keep track of the health information of individuals located away from the hospital is referred to as remote healthcare, which can be broken down into categories such as telehealth and mobile health. Under the telehealth domain, remote patient monitoring systems allows healthcare providers and caretakers to view and monitor patient health information remotely from a distance and allow them to take the necessary steps to prevent the patient from deteriorating (El-Rashidy et al., 2021). The usage of these monitoring systems contributes to reduced healthcare costs by further decreasing the number of hospital inpatient admissions and usage of facilities (Dhinakaran et al., 2022). With the usage of the remote patient monitoring systems, these resources can further improve the quality of life of its users (Salehi et al., 2020).

In the current age of healthcare delivery, there is a huge importance and need for analyzing and assessing medical data with better performance and greater accuracy of medical assessments by providing diagnoses and predictions done with the usage of machine learning models especially when humans cannot directly recognize certain abnormalities (Jayatilake and Ganegoda, 2021). Being able to assess a patient's health status ahead of time would heavily improve a patient's chances of recovery as a late diagnosis could lead to several serious complications on the patients' health (Nallakaruppan and Kumaran, 2020). It is therefore clear that the neglect of accurate or late diagnosis often places the patient at the risk of poor consequences, so early detection of such deterioration enables immediate involvement from medical personnel (Elliott and Endacott, 2022).

1.3 Problem Definition

Traditional healthcare delivery systems require doctors to view the patient, diagnose the disease, and advise the future course of action for the patient (Vinutha, Kavyashree and Raju, 2022), however these methods do pose some difficulty in terms of patient's mobility and daily workflow (Malasinghe, Ramzan and Dahal, 2019). Traditional remote patient monitoring further assists in that data transaction in healthcare delivery allowing doctors to assess the data and make decisions for the patient from afar while allowing patients to continue with their daily workflow. These systems too have their own limitations, one of them being due to their remote nature, certain emergencies cannot be accurately predicted or diagnosed prior (Albahri et al., 2018). With the usage of machine learning algorithms, this limitation can be resolved by allowing us to automate this process of analyzing the data by detecting abnormalities in the patient medical data early on in time and alerting the relevant healthcare providers of the patient's condition (Lata Sahu et al., 2021).

Considering the characteristics of the data, available machine learning model integrations with RPM systems point out a number of performance concerns. (Tabassum et al., 2020) proposed a hybrid encoding technique which they used in conjunction with the Random Forest algorithm, although they had received sufficient accuracy, they suffered from overall deficient performance in their solution for large datasets due to the large covariance between the techniques used. (Gontarska et al., 2021) developed a deep neural network model to calculate risk scores for cardiovascular diseases using vital signs collected where they also suffered from mediocre performance due to the time series nature of the data used. Having said that, we understand that one of the root issues is that most of the available solutions for integrating machine learning into RPM suffer from the performance of machine learning models leading to a lowered efficiency of such systems.

1.3.1 Problem Statement

Existing systems for classifying health status assessments and forecasting them with the use of machine learning models are faced with numerous model performance issues which lead to a decrease in precision and a substandard functioning of available remote patient monitoring systems.

1.4 Aims & Objectives

1.4.1 Research Questions

- **RQ1**: How can machine learning models be improved to increase the accuracy and efficiency of remote patient monitoring systems?
- **RQ2**: What are the benefits of using hybrid machine learning models for remote patient monitoring systems?
- **RQ3**: How can the proposed system assist medical professionals in delivering better healthcare?

1.4.2 Research Aims

This research aims to conduct the design, development, and evaluation of a hybrid machine-learning model to be used for acuity risk assessment and predict the future state of the health status of a patient using a vital sign time series forecasting model integrated in a remote patient monitoring system.

By utilizing the novel hybrid machine learning model that has been proposed, the research primarily intends to improve the performance and accuracy of health risk status assessments in remote patient monitoring systems.

1.4.3 Research Objectives

Table 1 - Research Objectives

Objective	Description	Learning Outcomes	Research Questions
Problem Identification	Identify and document the problem and justify its novelty. **RO1:** Examine the present condition of remote patient monitoring systems and recognize areas that need to be enhanced. **RO2:** Create an appropriate schedule, Gantt chart and identify deliverables required.	LO3, LO1	RQ3
Literature Review	Gather the necessary data by examining, understanding, and evaluating already existing publications. **RO3**: Conduct a thorough review of studies done on existing remote patient monitoring systems.	LO2, LO4, LO5	RQ1

	RO4: Investigate the need for specific prediction models in remote patient monitoring systems. **RO5**: Obtain and review insights on the architecture of systems in remote patient monitoring		
Data Gathering and Analysis	Collect and analyze the requirements of the project using relevant tools and techniques available. **RO6**: Gather requirements from the architecture of RPM systems. **RO7**: Collect available data on patient medical data and vital signs. **RO8**: Obtain views and recommendations from specialists in the respective field and technologies	LO1, LO2, LO3	RQ1, RQ2, RQ3
Research Design	Design a system that can successfully handle the issues that have been noted. **RO9**: Design an optimized hybrid framework capable of both diagnosis and prognosis for patient health status. **RO10**: Design a model with support for prognosis and diagnosis inputs.	LO1	RQ2, RQ3
Implementation	Implement the system that has been proposed to fill in the available research gaps. **RO11**: Implement a novel hybrid machine learning model capable of assessing patient health status efficiently with fewer false positives. **RO12**: Implement a remote patient monitoring system and integrate the developed model.	LO1, LO5, LO6, LO7	RQ1, RQ2

Testing and Evaluation	Effectively evaluate the implemented model and the RPM system using recommended techniques. **RO13**: Create a test plan with test cases for performance and integration testing. **RO12**: Test and evaluate the model against other studies	LO4	RQ2, RQ3
Documentation	Document the progression of the research project.	LO6, LO8	RQ1

1.6 Research Gap

Remote patient monitoring systems (RPMS) offer an advantage to patients by allowing for continuous monitoring of their health without having to be physically present in a medical facility. However, most systems currently use pre-set threshold limits that have been determined by doctors to assess the patient's state of health, which may not be suitable for every patient (Lata Sahu et al., 2021).

Existing research has identified a need for a more optimized and automated method of assessing patients' health status in remote patient monitoring systems using machine learning models as due to the characteristics of the dataset used, some studies have reported unsatisfactory performance (Gontarska et al., 2021). Therefore, the author intends to utilize a hybrid machine learning approach, which has not been widely used in remote patient monitoring systems, to fill in the gap and demonstrate the potential of such systems for enhancing patient outcomes in remote settings.

1.7 Project Contributions

1.7.1 Problem Domain Contribution

The author proposes the development of a comprehensive remote patient monitoring system that is capable of assessing a patient’s health status through the use of their vital signs. Without relying on physician set thresholds, the author plans on utilizing hybrid machine learning models to build a system for assessing a patient's health status. The author also plans to forecast the health status, in order to do this the author will develop a time series forecasting model capable of taking several days of vital signs and forecasting the vital signs of the next day allowing healthcare providers to take necessary measures to prevent deterioration.

1.7.1 Research Domain Contribution

Through the development of a remote patient monitoring system, the author plans to create a novel hybrid machine learning model to assess a patient's health status using the vitals added into the system, and also through the development of a time series forecasting model to forecast vital signs, the author will be able to utilize the same hybrid health status model on the forecasted vital signs in order to forecast the health status. Given the hybrid framework taken in this system's functionality, it is expected to perform more effectively than previous systems that offered similar outputs improving the efficiency of healthcare delivery in remote settings.

1.8 Research Challenge

The author faced several obstacles throughout the project timeline. One of the challenges was getting access to a healthcare dataset that was satisfactory and fit the needs of the project and could be utilized to effectively undergo training. Another difficulty was identifying the limitations of available literature, appropriate tools and technologies required, and handling patient data, in order to develop a novel framework in remote patient monitoring that could integrate and optimize assessments and forecasts. These challenges were mainly difficult due to the lack of literature around the specific domains.

1.9 Chapter Summary

This chapter presented the problem, the gap in the available literature, the aim and objectives of the project where its learning outcomes has been mapped to the learning outcomes of the university modules. the research challenges the author faces, and the contribution to the bodies of knowledge that the author would follow over the next several months to present the deliverables.

CHAPTER 2: LITERATURE REVIEW

2.1 Chapter Overview

In this chapter, the author will evaluate the body of literature that already exists in the field, investigate the technologies that have been used to implement health status assessments in monitoring systems, and consider potential advancements that could be made to integrate machine learning, triage health status assessments, and remote patient monitoring.

2.2 Concept Map

After surveying the available literature on a generalized scope of the domain, a concept map was constructed to discover the different possible fields the author could explore for this research under the different areas of challenges in healthcare services, limitations of available systems, the technologies used and the approaches that can be taken for evaluation.

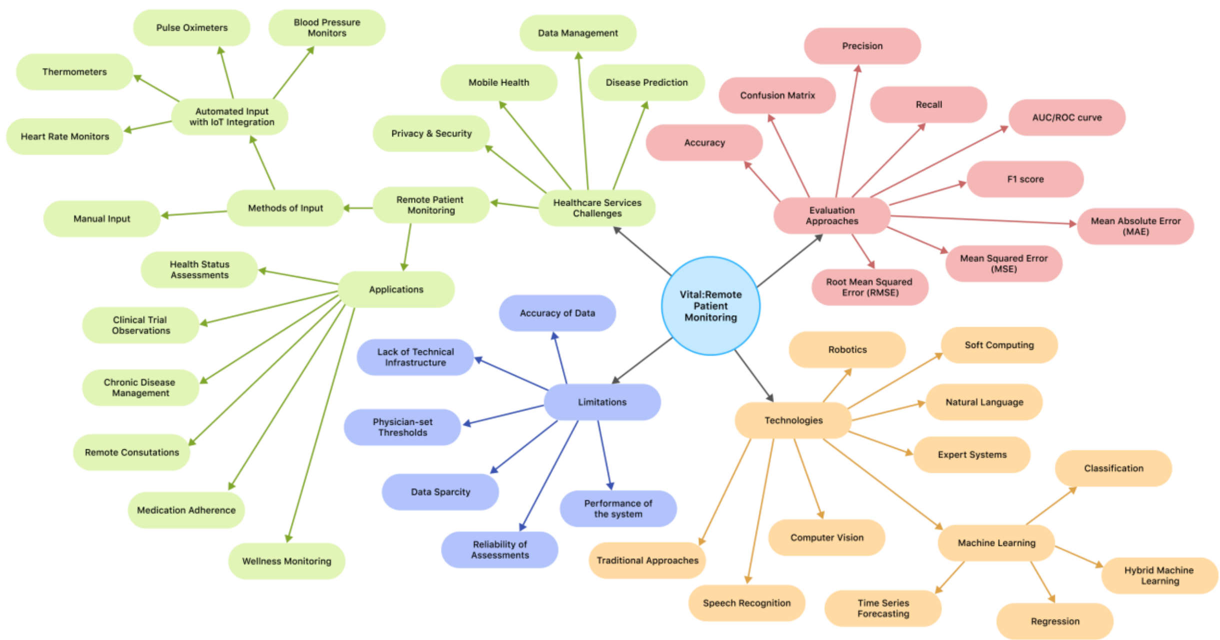


Figure 1 - Concept Map (Self-Composed)

2.3 Problem Domain

A study conducted by the World Health Organization showed that the number of elderly people, those aged 65 or over, is predicted to reach 1.5 billion by 2050, mostly from countries that are still developing, where chronic diseases are on the (World Health Organization, 2012).

Therefore, innovative solutions are necessary and should be reviewed and developed to assist and resolve the problems faced by patients, particularly those which incorporate advanced technologies such as artificial intelligence and machine learning into tele-healthcare systems (Bellini et al., 2022).

2.3.1 Tele-healthcare

The tele-healthcare domain is a rapidly emerging field of research and has become very popular in the over the past several months mainly cause of the compelling benefits and advantages available over its usage giving patients easier access to their health information and access to remote services within healthcare (Drerup et al., 2021). Telehealth systems can be beneficial in various ways. They can allow for early detection of illnesses and provide constant monitoring of patient data. Additionally, they can lead to cost savings for both patients and care providers as hospitalizations may be reduced (Kuthiala, Telivala and Karawadia, 2022).

These systems could further be moderated and improved with the usage of technologies such as machine learning to assist the healthcare providers with decision making in a clinical setting in nearly every expertise area along with the health status/triage assessments of patients as these technologies can study the data for abnormality and make judgements based on the inputs provided (Al-Turjman, Nawaz and Ulusar, 2020). It is predicted that by 2025, the demand for healthcare providers will increase by 94% due to the growing population. This could lead to problems with efficiency in the healthcare sector if there are more patients than care providers, creating a need for solutions to this problem (Kadum et al., 2023).

2.3.2 Remote Patient Monitoring Systems (RPMS)

Remote patient monitoring (RPM) is a field of research and a component under the tele-healthcare domain. Healthcare providers can use these systems to monitor a patient's vitals and other health information in real-time, typically with the assistance of integrated medical IoT devices. These systems can also be helpful in areas where there is restricted access to healthcare, when in-person care service can be expensive, or when a patient has difficulty with mobility (Timpano et al., 2013). Healthcare providers often use these systems to remotely monitor the patient's health for any signs of abnormality and make a clinical judgment for medical intervention in order to protect the patient's health from deteriorating any further (Sarierao and Prakasarao, 2018).

A survey of remote patient monitoring systems (RPMS) showed that research in this area is in-demand and is on the rise due to the increased healthcare needs of the aging population, the advancement of wireless technologies, and the improvements in monitoring technologies. This may result in a reduction of healthcare resources and expenses, resulting in a higher likelihood of hospitals, clinics and nursing homes utilizing remote patient monitoring systems. By having this system in place, it would enable care providers to monitor the vital signs of their patients to prevent any further decline in their health (Olivencia et al., 2022).

2.3.3 Patient Triage/Health Status Assessments

In healthcare, patient triage/health status assessment is a process that involves evaluating a patient's health and deciding the urgency of their treatment based on the severity of their condition. Its purpose is to make sure that patients receive the right care at the right time, according to their medical requirements (Napi et al., 2019). These systems were mainly created to monitor the patient's progress within the emergency department. However, they can also be applied to other medical environments to classify patients according to the severity of their condition (Phungoen et al., 2020)

The Emergency Severity Index (ESI) is a method of triaging patients which categorizes them into five levels, ordered from the most urgent (1) to least urgent (5), according to the urgency of the patient's condition and their need for resources. This method considers the patient's vitals, their chief complaint and their age, and this has proven to be a very reliable and effective way of prioritizing the patient and generating a status level (Phungoen et al., 2020)

2.3.4 The Role of Machine Learning in Telehealth

The usage of machine learning methods and tools in the majority of areas to analyze the data collected and construct models has been vital to understanding and improving the performance of systems in the tele-healthcare domain especially in triage systems during emergency situations where care providers need to sort the patients based on severity to take immediate action for those that require their services the most (Gao et al., 2022).

As the demand for telehealth services rises, care providers are feeling the pressure of insufficient time to examine and supervise every individual patient and make clinical decisions, especially with an increasing number of patients where with each new patient comes new data to analyze. This has necessitated the need for a solution in order to prioritize and identify those patients who require immediate attention, in order to guarantee quality care for all (Salman et al., 2021). Utilizing the vital signs of the patient, the author plans to assist in this process by triaging the remote patients using machine learning techniques to allow the healthcare providers to view the patients based on their status accordingly.

The rise in the usage of machine learning in the domain has given rise to new technologies such as AI chatbots that are being utilized for patients to access round the clock aid in identifying their symptoms and providing a diagnosis. Nevertheless, the accuracy of these systems may not always be reliable as the patient may be untruthful or deliberately misrepresenting their symptoms to the chatbot (Kandpal et al., 2020)

2.3.5 Time Series Forecasting

When multiple observations are taken at a fixed time interval over a certain period, the data is referred to as time series data (Ciaburro and Iannace, 2021). The research and implementation of time series forecasting models trained on historical data has been very popular amongst researchers as it can play a huge role in solving problems in day-to-day life (Verma et al., 2021).

As data and processing power become more accessible to the researchers and developers, it has become very clear that deep learning and machine learning models have both become essential in the recent years for the development of new timeseries models (Xuan et al., 2022).

These models have been applied in a variety of areas, including but not limited to forecasting the amount of traffic during a certain time of day, forecasting the climate during a certain period of the year by understanding the weather patterns, and predicting stock and cryptocurrency values (Elsayed et al., 2021).

2.4 Existing Work

There were several studies done around the domains relating this research project towards remote monitoring of patients, assessing the patients' health status/triage them in order to improve the performance of these systems due to their contribution to the field and lives of patients.

2.4.1 Remote Patient Monitoring & Patient Health Assessments

(Lata Sahu et al., 2021) developed a mobile application that collects vital signs in real time through the integration of IoT medical devices where the data could be stored locally and, on the cloud, however the system used a rule-based method for abnormality detection which required physician set thresholds in order to generate alerts.

A study by (Dhinakaran et al., 2022) analyzed the challenges in remote patient monitoring systems and found the usage of machine learning to be very efficient in removing flaws from loss of information and miscommunication between the care provider and patient, the study showed made use of a decision support component which was rule-based to detect abnormalities to notify the care provider.

(Aditya et al., 2020) proposed the usage of remote patient monitoring systems for the Intensive Care Unit (ICU) where doctors can monitor the patient's status while away. The system developed utilized image processing of the patients movements and a temperature sensor to alert the doctor of any abnormality in real time.

A study by (Ishtiaque et al., 2021) showed the significance of using remote patient monitoring systems where they proposed a low energy patient monitoring system to gather different vital signs from the patient where the doctor would get alerted through a GSM module in cases where the condition of the patient is critical, the author of that study also implemented a web application which included several additional functionalities to detect pneumonia using chest x-rays and arrhythmia using deep with the ECG data collected from the system.

(Mia et al., 2021) had gathered several vital signs through wearable devices allowing the system to monitor different parameters like blood pressure, body temperature, blood oxygen saturation, heart rate, while also making use of additional functionalities like patient fall detection, and breath analysis in order to feed it to the remote patient monitoring system for patients with chronic diseases during the pandemic.

(Anitha et al., 2022) developed a web-based application made use of several machine learning algorithms in order to provide predictions and diagnose the patient through remote monitoring systems. The application developed allowed the surveillance of the patients vitals, returned any alerts and also suggested certain drugs based on the disease detected.

A study done by (Zachariasse et al., 2019), reviewed the performance of available triage systems used in emergency care and evaluated the validity of each of the systems which led to understanding that there is no clear triage system that was good as each of them performed differently as it was based on the parameters used for the system to triage.

A study produced by (Gontarska et al., 2021) which was related to prediction of medical interventions through vital signs where the author had implemented a deep learning model to produce a risk score for cardiovascular diseases in order to prioritize patients for treatment which outperformed the rule-based model they had compared it with which showed the usage of deep learning models can improve the efficiency of triaging patients.

(Tabassum et al., 2020) developed a new approach for data enhancement to improve the performance of machine learning models on a patients vital signs collected through a Portable Healthcare (PHC) system to predict the status of the patients in order to decide on a date to follow up for visits.

In the study conducted by (Tan et al., 2022), the authors researched and developed a remote patient monitoring system with the integration of several sensors to identify and track the vital signs of a patient in real-time, the system developed on a real-time, the system developed on a rule-based system and only alerted the doctor when certain thresholds were passed.

2.4.1 Vital Sign Forecasting in Monitoring Systems

In the research done by (Phetrittikun et al., 2021) mentioned that vital signs are essential to identifying the risk of the patient's condition deteriorating over time therefore forecasting these parameters would allow care providers to optimize the treatment plan of the patient. Using time series vital sign measurements, they were able to capture and detect the irregular patterns in the vital sign data.

A study by (Chang, Chang and Pourhomayoun, 2019) had developed predictive models in order to predict critical vital signs an hour ahead of time for patients within the ICU to prevent their health from deteriorating further.

(Bhavani et al., 2022) developed a health monitoring system with the usage of IoT medical devices capable of performing stress classification and also vital sign forecasting. The study made use of several physiological vitals when classifying stress and used metrics like RMSE and MAE for vital sign forecasting.

2.5 Technological Review

There have been several implementations of remote patient monitoring systems, patients health status assessments/triage and vital sign forecasting over the years especially today where there is a huge demand, and more research is being done in the domains. Several different approaches were taken for the development of these systems.

2.5.1 Machine Learning Approaches

Various models were tested and evaluated in terms of integrating machine learning models for health status assessments and patient triages. In the study by (Souri et al., 2020) which goes into the development of a machine learning model to diagnose the condition of a student through a health monitoring system. The study made use of several models such as DT, SVM, RF and MLP neural networks where the experiment concluded that the SVM algorithm provided the best results in detecting the condition of a student using the evaluation metrics – accuracy, precision, recall and F1-score.

(Godi et al., 2020) suggested the usage of algorithms like SVM, DT, KNN and Regression, the study specifically implemented the SVM algorithm for the research based on its prediction performance for large scale data where the model was implemented into a healthcare monitoring system.

(Anitha et al., 2022) made use of data gathered through sensors in real-time where it was tested against various models like Logistic Regression, Random Forest and XGBoost in order to receive the best accuracy and sent an alert to the care provider of the patient if any criticality is observed where they found that the Random Forest model performed the best.

2.5.2 Deep Learning Approaches

In terms of vital sign forecasting, only few recent studies focused on using machine learning where most studies were mainly done specifically utilizing deep learning algorithms as DL models are capable of learning complex patterns and relationships within time series data as certain traditional machine learning models may not be able to capture temporal dependencies within the data required to make accurate predictions.

(Bhavani et al., 2022) worked on Recurrent Neural Network (RNN) specifically a type of RNN called Gated Recurrent Unit (GRU) which is similar to LSTM which it was trained on a combination of vital sign parameters in order to forecast the future values which performed better when compared to models like Auto Regression, ARIMA and LSTM

In the study by (Wang et al., 2021), A Convolutional Neural Network (CNN) model was developed to forecast vital signs using ballistocardiogram (BCG) signals where the model showed good performance showing increased accuracy when using DL models.

(Phetrittikun et al., 2021) made use of the Temporal Fusion Transformer deep learning model in order to perform multi horizon vital sign forecasting and concluded that the model was capable of understanding the patterns within the data.

The paper by (Liu, Yao and Motani, 2019) proposed the usage of generative boosting techniques on LSTM models to generate vital signs for a couple time steps. The model developed indicated that the method suggested provided good results was able to forecast certain vital signs ahead of time.

3.5.3 Data Preprocessing Techniques

Many of the datasets used in these studies contain missing measurement values which could affect the performances of the models. However, only a few studies mentioned the techniques used to preprocess the data.

To tackle missing values, (Phetrittikun et al., 2021) implemented forward filling with median values. (Bhavani et al., 2022) dealt with missing measurements by replacing the missing data with the previous value using forward filling, the data was resampled with the average where outliers were not removed as it may affect the performance due to it being time sensitive data.

(Srinivasulu and Gupta, 2022) ensured the measurement data was properly preprocessed to ensure good performance of the models through data mean imputation for the missing values, the study also dealt with data normalizing by identifying the min and max values.

2.6 Evaluation

2.6.1 Evaluation Approaches

The performance of the application can be measured by looking at the evaluation metrics when comparing the classifications and predictions to the test data.

Table 2 - Metrics Used for Evaluation in Health Status/Triage Classification

Metric	Description	Formula
Accuracy	The ratio of accurate predictions to the sum of all predictions generated by the model.	$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$
Precision	The ratio of true positive predictions to the total number of positive predictions.	$Precision = \frac{TP}{TP + FP}$
Recall	The ratio of true positives to the total of both true positives and false negatives.	$\text{Recall} = \frac{TP}{TP + FN}$
F1-Score	The harmonic mean of both precision and recall	$F_1 = \frac{2}{precision^{-1} + recall^{-1}}$

Table 3 - Metrics used for Evaluation in Vital Sign Time Series Forecasting

Metric	Description	Formula
MAE	The average absolute error between the actual and predicted values.	$MAE = \frac{1}{N} \sum_{i=1}^{N} \lvert y_i - \hat{y}_i \rvert$

MSE	The average squared difference between the estimated value and true value	$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \tilde{y}_i)^2$
RMSE	The square root of MSE, also known as L2 loss	$\text{RMSE} = \sqrt{\sum \frac{(y_{pred} - y_{ref})^2}{N}}$

2.6.2 Benchmarking

In order to perform a benchmarking analysis, a standard data set that is utilized by several research studies is required. Unfortunately, the majority of studies have used custom datasets sourced from locally, which are not accessible to the general public and therefore not accessible to the author for comparing the proposed system. Consequently, the author is unable to conduct a benchmarking study.

2.7 Chapter Summary

This chapter provides an overview of the research conducted to identify any potential gaps in knowledge. This was done by breaking down the research into various disciplines and displaying the results as a concept map. Each piece of literature was carefully analyzed and assessed to reveal any possible gaps that could be further looked into in the future.

CHAPTER 3: METHODOLOGY

3.1 Chapter Overview

This chapter provides an overview of the strategies and procedures the author has employed in order to ensure the research project is feasible and efficient.

3.2 Research Methodology

The following research methodologies were evaluated and selected from a predefined Research Onion Model by (Saunders, Lewis, and Thornhill, 2007).

Table 4 - Research Methodology

Research Philosophy	The research revolves around watching evaluation measures and feedback gathered through an iterative procedure, hence the author chose **Pragmatism** as the research philosophy.
Research Approach	The **Inductive** strategy has been chosen since it seeks to address the issue through trial and error.
Research Strategy	**Surveys, Interviews and Document Analysis** will be the primary research strategy to assist with evaluation, testing and feedback.
Research Choice	The **Mixed Method** would be used as it suits the proposed research for the required analysis
Time Horizon	Since the data will be collected all at once during the study analysis phase, the **Cross-sectional** time horizon was selected as the best choice.
Techniques and Procedures	The techniques used will be a form of **Data Collection and Analysis** through trial & error, analyzing similar solutions and literature.

3.3 Development Methodology

3.3.1 Life Cycle Model

Agile was chosen as the approach for the research and development life cycle because the project requires extensive iterative development.

3.3.2 Design Methodology

The author chose **Structured Systems Analysis and Design Methodology (SSADM)** as the design methodology during the development process instead of Object-Oriented Analysis and Design Methodology (OOADM) due to the simplicity of implementation and the fact that the core components of the system would not gain any specific benefit from the object-oriented design.

3.3.3 Evaluation Methodology

According to the proposed research, the **Classification Accuracy** level and **Root Mean Square Error (RMSE)** are the best metrics for measuring the proposed model's performance.

3.4 Project Management Methodology

As the project requires numerous iterations for ongoing changes, the author has chosen to combine **Agile** and **PRINCE2** while also partitioning the work into multiple chunks to focus on each with a plan-based approach.

3.4.1 Schedule

3.4.1.1 Gantt Chart

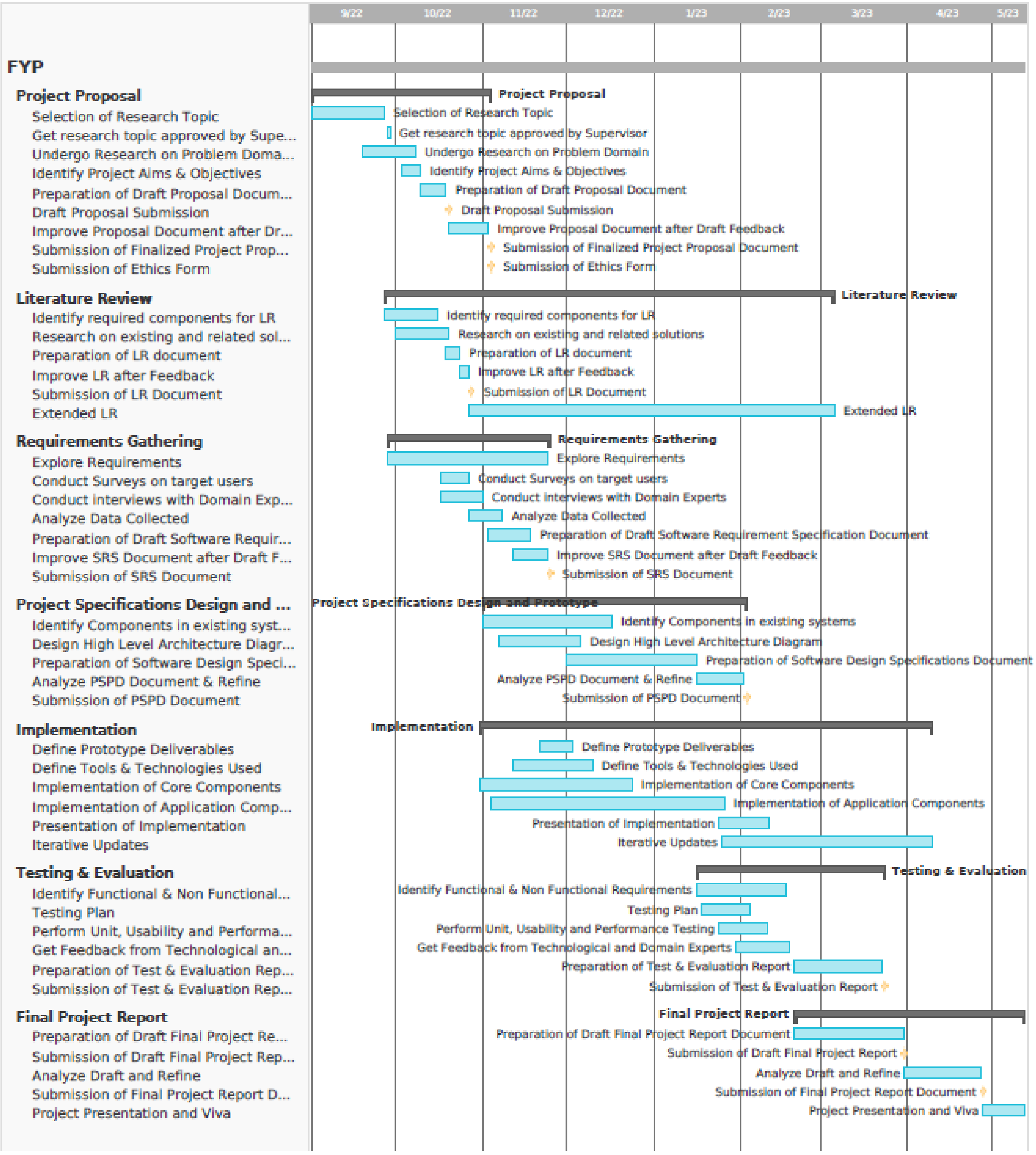


Figure 2 - Project Schedule Gantt Chart (Self-Composed)

3.4.1.2 Deliverables and Dates

Table 5 - Deliverables and Dates

Deliverable	Date
Literature Review Critical analysis of related work	27th October 2022
Project Proposal & Ethics Form	3rd November 2022

Initial document proposing the research to be conducted	
Software Requirement Specification (SRS) The guide that outlines the conditions that must be followed in order to develop the prototype and gather data	24th November 2022
Proof Of Concept & Implementation Presentation An initial Implementation of the proposed system.	23rd December 2022
Project Specifications Design & Prototype A prototype of the system with the core features and an attached document specifying the design followed & an overview of the implemented algorithm.	2nd February 2023
Final Thesis Final thesis submission with thorough project journey documentation.	27th April 2023

3.4.2 Resource Requirements

3.4.2.1 Hardware Resources

- **Core i7 Processor (8th gen) or above** – To be able to handle models with intensive workloads.
- **8GB RAM or above** – To be able to oversee multiple development environments, multitasking and model implementation.
- **Disk storage of 10GB or more** – To store application code and data.

3.4.2.2 Software Resources

- **Operating System (Windows/macOS/Linux)** – Windows will be utilized as the default operating system because all necessary tools are readily available.
- **Python** – Will be used to create the proposed model, it is an all-purpose language with different tools and libraries designed specifically for data science.
- **TensorFlow, Keras, and Scikit-learn** – Libraries to facilitate the data science modules in the model.

- **Flutter** – To develop the client side of the application, Flutter is a development framework capable of cross-platform functionality leading to a single codebase.
- **Firebase** – To facilitate the backend and storage module of the application with seamless integration to the client framework.
- **Visual Studio Code, and Android Studio** – Will be used to facilitate the development of the application.
- **Jupyter Notebook, and Google Colab** – Will be used as the development environment for building the prediction model.
- **Mendeley** – Used to manage references and research artifacts.
- **GitHub** – Will be used for Version Control during the implementation period of the proposed research.
- **Figma** – Will be used to create the high-fidelity design prototype of the application.
- **Microsoft Office, Draw.IO, Google Suite, and Notion** – Will be used in the documentation process to create reports, diagrams, figures, and backup files.

3.4.2.3 Technical Skills

- Academic research and writing skills.
- Work on the creation and implementation of a hybrid machine-learning model
- To be able to build scalable and optimal machine learning and deep learning models.
- Being able to create client-side software with dynamically updating user interfaces based on expected outcomes.

3.4.2.4 Data Requirements

- **Patient Vital Signs Dataset** – A dataset that includes but is not limited to the vital signs of a patient with the acuity triage assessment column for each row collected in a time series nature with the chart time column available.

3.4.3 Risk and Mitigation

The following table identifies and highlights the possible risks that could occur during the project timeframe and how the author plans to mitigate them.

Table 6 - Risk Mitigation Plan

Risk Item	Severity	Frequency	Mitigation Plan
Lose access to the code	5	2	The usage of Version Control solutions, and Cloud-based backups.
Corrupted Documentation	4	4	Backup and store all necessary files and documents on the cloud and external storage
Inability to deliver expected deliverables	4	2	Plan ahead and adhere to a list of priorities to deliver in accordance
Invalid hypothesis	3	2	Continue with the research as there will still be a contribution regardless
Lack of knowledge during the development of ML models	6	4	Learn from industry experts and gain insights

3.5 Chapter Summary

The chapter went over the methodologies the author had taken including the research methodologies, development methodologies and project management methodologies, the project schedule, the requirements of the research and a mitigation plan for any foreseeable risks.

CHAPTER 4: SYSTEM REQUIREMENTS SPECIFICATION

4.1 Chapter Overview

This chapter discusses the system stakeholders, requirement elicitation techniques, functional and non-functional requirements based on priority level, use case diagrams, and use case descriptions of the system.

4.2 Rich Picture Diagram

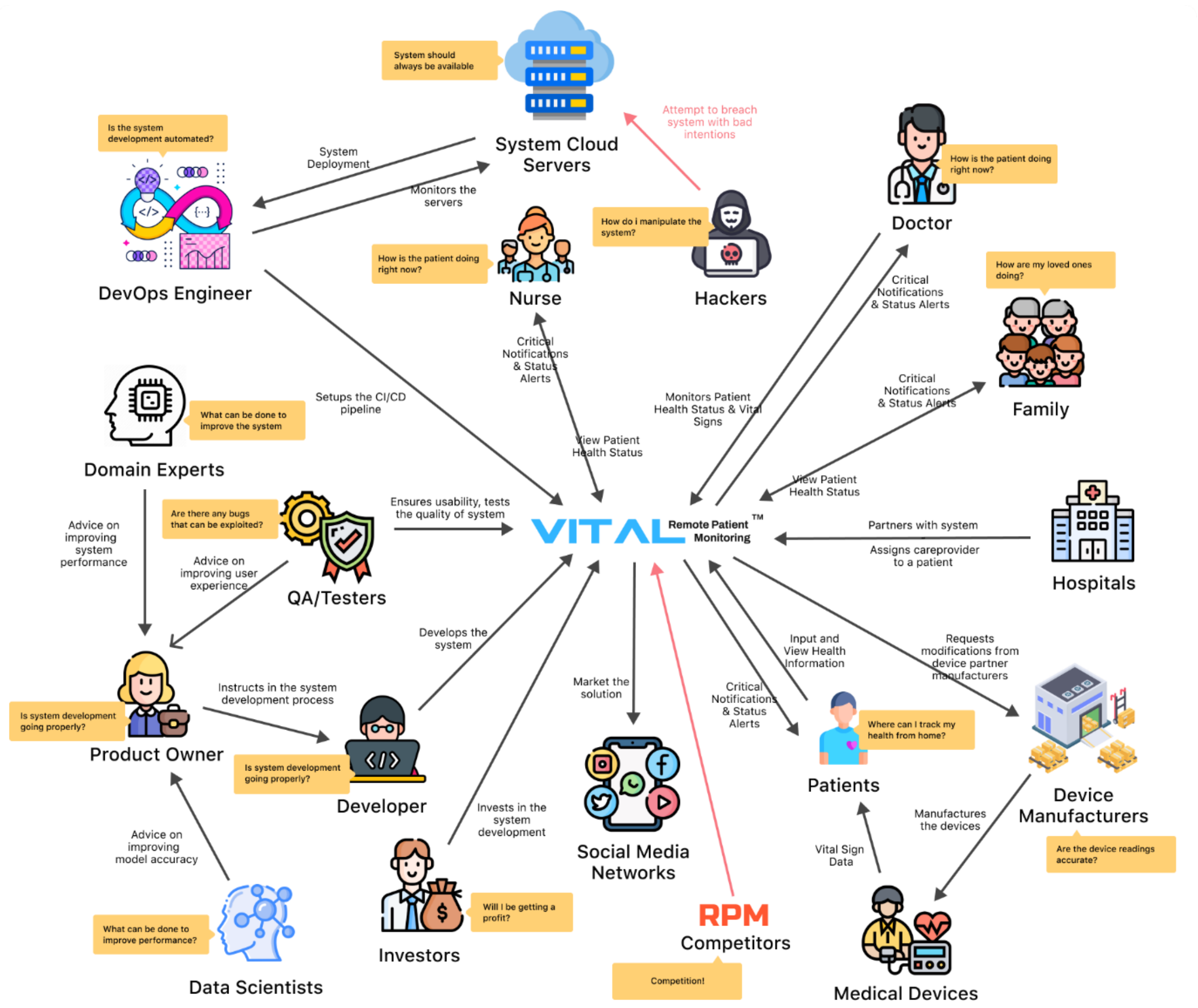


Figure 3 - Rich Picture Diagram (Self-Composed)

The rich picture diagram above showcases the flow of interaction within the system which would be used to understand who and what interacts with the system to make it functional.

4.3 Stakeholder Analysis

The onion model below showcases all the major stakeholders who can be associated with the system with a description of their involvement in the stakeholder viewpoints.

4.3.1 Stakeholder Onion Model

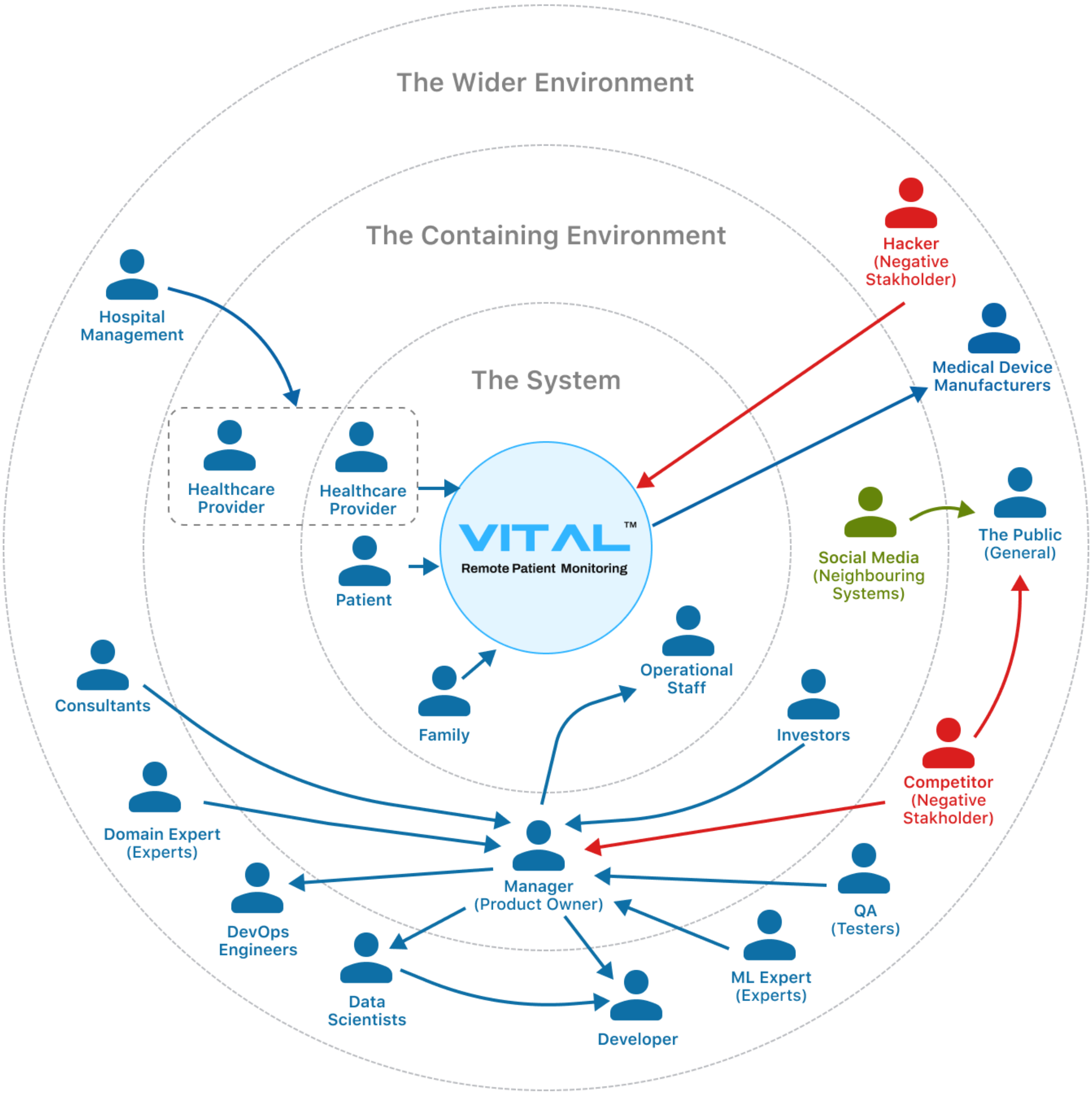


Figure 4 - Stakeholder Onion Model (Self-Composed)

The above figure represents the onion model of the VitalHealth system. Detailed description of all the system stakeholders is provided below.

4.3.2 Stakeholder Viewpoints

The table below briefly describes each of the stakeholders and their roles in the system.

Table 7 - Stakeholder Viewpoints

Stakeholder	Role	Description
Patient	Functional Beneficiary	It becomes easier for patients to receive medical monitoring and assessments from the comfort of their home, reduces financial burdens and ease of access to their healthcare providers
Family/Guardian	Functional Beneficiary	It becomes easier for family members to monitor the health of the patient while away and receives alerts during emergencies.
Healthcare Provider	Functional Beneficiary - Financial Beneficiary	it allows doctors to monitor patients from a distance and get alerted when there are irregularities allowing them to manage and monitor more patients tending to those in need of immediate services.
Developer	Financial Beneficiary, Maintenance	Develops the system and works on new features/functionalities
Manager/Product Owner	Financial Beneficiary, Administration	Oversees the team, providing direction to the programmers and support employees.
Domain & ML Experts	Expert & Quality Regulator	Offers expertise, guidance and insights into the domain and technologies to enhance system performance.
Data Scientists	Expert & Quality Regulator	Improves the performance of the system's models and algorithms.
DevOps Engineers	Operational, Deployment & Upkeep of the System	Ensures that the system is fully operational in the cloud and is serving users without being overloaded.
Testers	Operational, Quality Assurance	Tests the system and assesses whether it is acceptable for deployment in production

Stakeholder	Role	Description
Operational Staff	Operational, System Support	Maintains system functionality and responds to user concerns and requests.
Medical Device Manufacturers	Financial Beneficiary	Creates the medical equipment that patients use. works closely with the system and adjusts device output to suit system requirements
Competitor	Negative Stakeholder	May study the product and build competing applications that can outperform the system
Hackers	Negative Stakeholder	If proper security measures aren't met, it allows hackers to manipulate and access system information, highly confidential patient information with malicious intentions
Investors	Financial Beneficiary	Invests in the system development and benefits off the profit received.
Hospital Management	Operational, Administration	Works together with the system to add healthcare providers, and assign patients to the doctors

4.4 Requirements Gathering

4.4.1 Techniques for Gathering Requirements

Table 8 - Techniques for Gathering Requirements

Method 1: Literature Reviews
During the initial stages of the project, the author has performed thorough analysis of the literature available to identify the available research gaps in the chosen domain where similar systems and technologies that could be integrated into such systems were studied
Method 2: Survey Questionnaire
The author created a questionnaire to gather needs and opinions from potential users of the proposed system. This questionnaire would help the author better understand the target users and determine whether or not such a system would be beneficial to them. It would also help the author to determine what features and functionalities would be useful to them.
Method 3: Brainstorming
The requirements for the prototype and its design can be determined through brainstorming sessions. especially when identifying the key features, creating the appropriate diagrams, to turn the research component into a proof-of-concept solution.
Method 4: Prototyping
Since the author had decided to use Agile development, prototyping would allow him to test and evaluate the system while highlighting any areas that needed improvement.

4.5 Analysis of Gathered Results

4.5.1 Findings from Literature Review

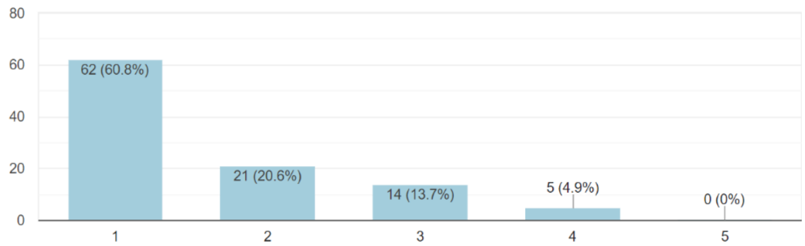
Table 9 - Findings from Literature Review

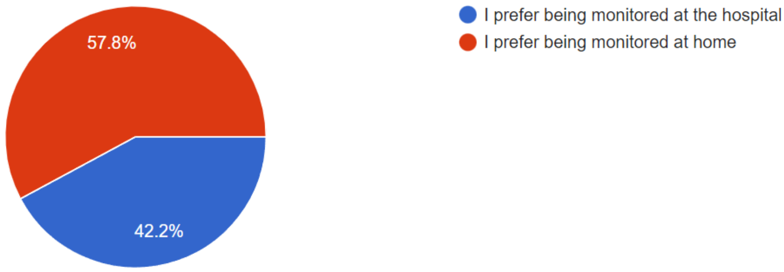
Finding	**Citation**
After reviewing the literature, the author had identified that automated health status reports generated in cloud based remote patient monitoring system would not only heavily benefit the patients but the care providers aswell by reducing costs on both sides while keeping a constant monitor on the patient vital signs.	(Lata Sahu et al., 2021; El-Rashidy, N. et al., 2021)
It was identified while exploring different technologies that could be applied to the domain that available machine learning models that were used in remote patient monitoring systems came back with several performance issues.	(Gontarska et al., 2021; Chang et al., 2019; Vinutha et al., 2022)
The author identified that the vital sign predictors that were often used in health status assessments of patients were the heart rate,	(Naemi, A. et al., 2021)

temperature, respiratory rate, oxygen saturation, systolic and diastolic blood pressure were essential to the patient monitoring process.	
Allowing the usage of physician set thresholds for the vital signs allows the models to be more patient specific for the prediction model.	(Dhinakaran, M et al., 2022)

4.5.2 Findings from Survey

Table 10 - Findings from Survey

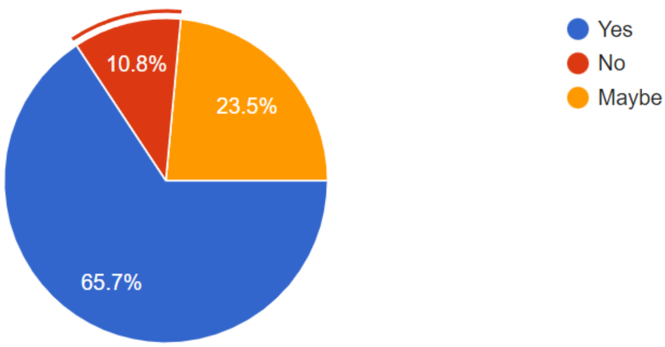
Question	**How often do you visit the hospital for medical checkups/assessments?**
Aim of Question	To understand how often a potential user might go to the hospital for assessments.
Findings & Conclusions	
	
Based on the responses, it is clear that most of the participants don't visit the hospital frequently; the data revealed that only 5% of them would do so often, while more than 60% visited the hospital just sometimes or hardly at all.	
Question	**If given the option, would you prefer your health being monitored at home instead?**
Aim of Question	To understand the preference of a potential user of whether they prefer being monitored at the hospital over being monitored at home.
Findings & Conclusions	



We might conclude from these findings that some of the participants are still somewhat hesitant about adopting home-based remote monitoring systems. Where 42.2% of individuals preferred being monitored at the hospital, only 57.8% of participants preferred being monitored at home.

Question	**Would you prefer an automated system where your doctors will be alerted when there are irregularities in your health?**
Aim of Question	To understand if such a system was available, would the participants prefer it where their doctor would be alerted during irregularities in their health.

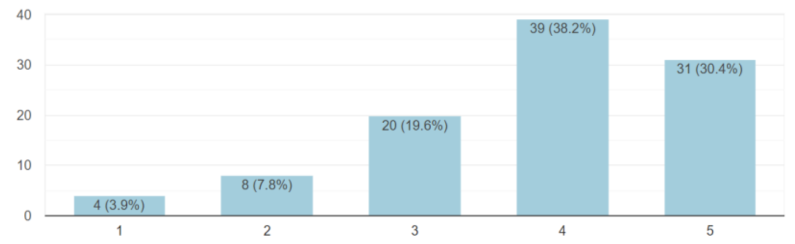
Findings & Conclusions



While 23.5% of respondents replied Maybe, which could indicate that they are unsure at the time of this study, 65.7% of participants stated they preferred having an automated system where their doctors will be alerted when there are deviations in their health. 10.8% of the participants thought that having a system like this was not something they preferred. Overall, it's encouraging to see that the majority of participants are eager to use such a system!

Question	**How beneficial would a system like this be for you?**
Aim of Question	To understand how beneficial the proposed system would be to them?

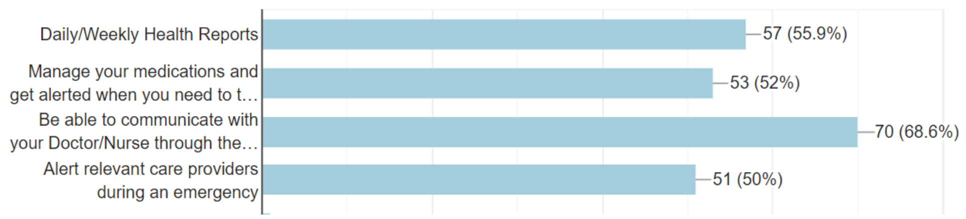
Findings & Conclusions



The majority of participants believe that the suggested system would be highly helpful to them, as shown by the histogram of the replies gathered from the participants. It is incredibly exciting to receive such favorable comments because they are the primary target audience.

Question	**What additional features would you like to see in such systems?**
Aim of Question	To identify any additional features that may be added to the system to improve usability and user experience.

Findings & Conclusions



Majority of the participants felt that being able to communication directly with their care provider through the system would help improve their user experience, where others suggested daily/weekly health reports, a medication management and reminder system and being able to alert care providers manually during an emergency.

4.5.3 Findings from Brainstorming

Brainstorming sessions were conducted to validate the research area and gather all the requirements for the proof of concept in order to understand the target users of solution, validate what functionalities are feasible to a good user experience when using the application.

4.5.3 Findings from Prototyping

The findings from the prototyping of the hybrid health status assessment machine learning model revealed the effectiveness of the model in accurately predicting the health status of patients based on their vital signs. During testing, the model was able to successfully take in inputs of heart rate, temperature, respiratory rate, oxygen saturation, systolic and diastolic blood pressure and produce the patient’s acuity level which is used as the health status level. The results of the prototyping confirmed the viability of the model and its potential to be integrated into a remote patient monitoring system for the minimum viable product.

4.6 Summary of Findings

Table 11 - Summary of Findings

ID	Finding	Literature Review	Survey	Brainstorming	Prototyping
1	The proposed system would heavily benefit the users which includes both the patients and care providers	✓	✓	✓	
2	Machine learning models used in remote patient monitoring systems have performance issues	✓			✓
3	Essential vital signs that are easily accessible to the public which are used in health status assessments (heart rate, temperature, respiratory rate, oxygen saturation, systolic, and diastolic blood pressure)	✓	✓	✓	✓
4	Having physician-set thresholds can make prediction models more patient specific	✓		✓	
5	The system's performance would depend on having a sufficient amount of well-cleaned and pre-processed data.			✓	✓
6	Not many people are aware of remote patient monitoring solutions, and many are still reluctant to use the available solutions		✓		
7	Some patients need to segment based on different diseases categories and vital tracking based on that		✓		

4.7 Context Diagram

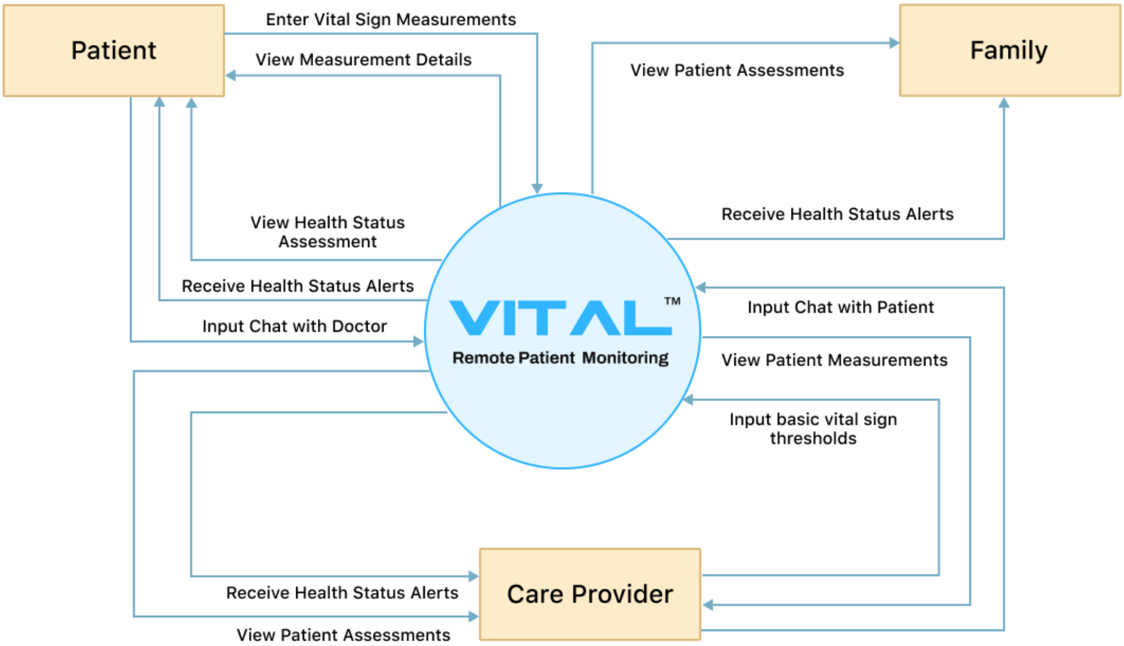


Figure 5 - Context Diagram (Self-Composed)

4.8 Use Case Diagram

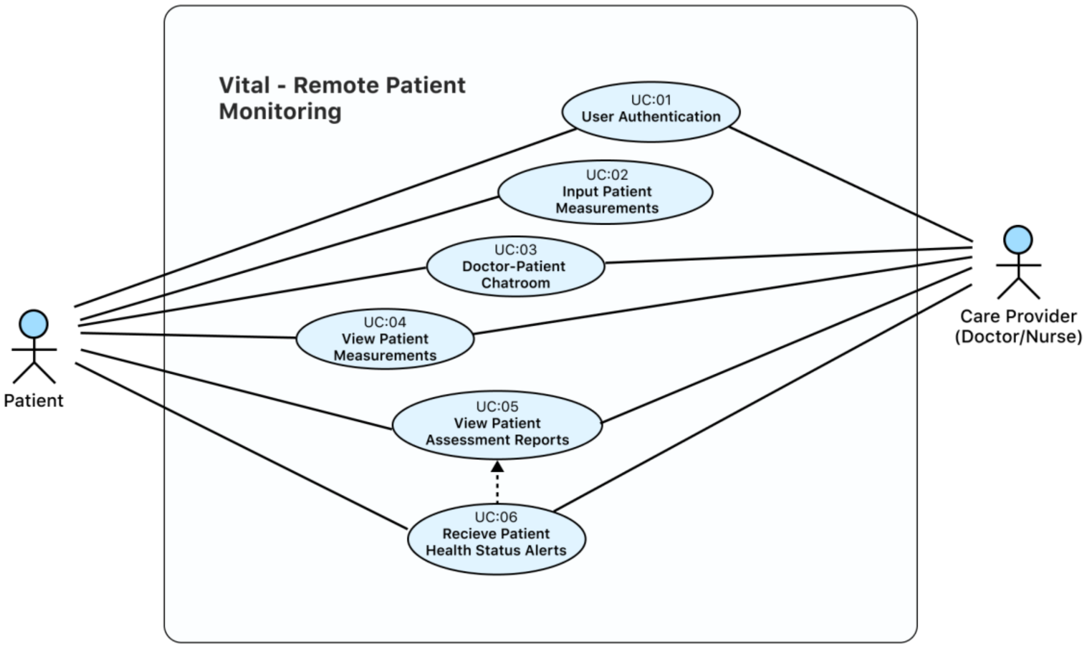


Figure 6 – Use case Diagram. (Self-Composed)

4.9 Use Case Description

The core use case descriptions that were identified are provided below.

4.9.1 Input Patient Measurements

Table 12 – Use case Description - Input Patient Measurements

Use Case	Input Patient Measurements
ID	UC:02
Description	The patient can input their vital sign measurements into the system which are required.
Primary Actor	Patient
Supporting Actors (if any)	None
Stakeholders (if any)	Doctor, Family
Pre-Conditions	Patient should be logged in to the correct account.
Post Conditions	Success scenario: The patient will be presented with a basic overview of the vital measurements added to the system.
Main Scenario	- Patient selects the add measurement option. - Patient selects measurement type and inputs the required fields based on that measurement. - The measurement is submitted and processed. - The system then navigates the patient to the vital measurement page containing a list of all the measurements added for that particular vital sign.
Alternative Scenarios	The patient can be navigated to the health status assessment screen with an overall assessment of the health status and vital signs used for it. However, this would require the patient to have added other required vital sign measurements previously.

4.9.2 View Patient Assessment Reports

Table 13 - Use case Description - View Patient Assessment Reports

Use Case	View Patient Assessment Reports
ID	UC:05

Description	Displays a report of the patient's health status generated automatically from the vital sign measurements added by the user.
Primary Actor	Patient
Supporting Actors (if any)	Doctor, Family
Stakeholders (if any)	None
Pre-Conditions	All measurements would need to be added previously. At least one health status assessment report will have needed to have been generated.
Post Conditions	Success scenario: The patient is presented with the health status assessment reports generated
Main Scenario	- The health status report is automatically generated once a patient enters all required vital signs. - The patient selects the reports screen option on the navigation bar. - The system loads all available health status assessments generated by the system. - The user selects the report he wishes to view. - The system opens the report in a new screen showing all the details on the report mentioning the status, the vital signs used to come to that conclusion and any alerts related to it.
Alternative Scenarios	If no previous health status report is generated, the user is presented with a screen containing all the pending measurements required to generate a report.

4.10 Requirements

4.10.1 Functional Requirements

Based on their significance, the MoSCoW approach was used to rank the system demands in order of priority.

Table 14 - Priority Descriptions - MoSCoW approach

Priority Level	Description
Must have (M)	This is a primary functional requirement for the system, and it must be carried out.
Should have (S)	While not strictly necessary for the desired prototype to function, these requirements do provide a lot of value.
Could have (C)	Requirements that are deemed as desirable and optional but aren't essential to the system
Will not have (W)	Requirements that will not be implemented due to certain constraints but could be integrated as future work.

Table 15 - Functional Requirements

FR ID	Requirement	Priority	Use Case
FR1	Patients must be able to add vital sign measurements into the system	M	UC:02
FR2	Doctors should be able to monitor patient vital signs and status through the system	S	UC:04
FR3	The system should be able to fetch the patient vital signs collected over a period of time and make a forecast	M	UC:05
FR4	The system should take the patient vital signs collected and provide an immediate diagnosis/status check	M	UC:05
FR5	The system should be able to generate daily/weekly health reports based on data collected/generated	C	UC:05
FR6	The system should alert healthcare providers when there is an irregularity in patient vital signs and health status level	S	UC:06
FR7	Integrating IOT medical devices into the system to automate the data collection process for continuous tracking	W	UC:02
FR8	Having a chatroom system, which would allow the patient and doctor to communicate directly	C	UC:03
FR9	Include multi-language translation support to the system to allow support for a wider variety of people	C	-

4.10.2 Non-functional Requirements

Table 16- Non-Functional Requirements

NFR ID	Requirement	Description	Priority Level
NFR1	Performance	The system should be capable of providing the assessment classifications as soon as possible once the required data is available.	Important
NFR2	User Experience	The usability and user experience of the system is one of the priorities of the system to ensure it is makes it easier for the stakeholders and that it is simple to use.	Important
NFR3	System Accuracy	It is very important to make sure the accuracy provided by the system is high.	Important
NFR4	Security	The system should make use of proper techniques to prevent any attackers from accessing the patient data.	Important
NFR5	Scalability	The system should be scalable to allow future growth within the system.	Desirable

4.11 Chapter Summary

This chapter discussed the various system stakeholders, different techniques for gathering requirements, the distinction between functional and non-functional requirements based on their priority level, and an overview of the system's use case diagrams and descriptions. The chapter's goal was to provide a thorough understanding of the process of gathering and organizing system requirements.

CHAPTER 5: SOCIAL, LEGAL, ETHICAL AND PROFESSIONAL ISSUES

5.1 Chapter Overview

This chapter aims to define the social, legal, ethical and professional issues that may happen during the project timeline and what steps will be taken to mitigate it.

5.2 Breakdown of Social, Legal, Ethical and Professional Issues & Mitigation

5.2.1 Social

- Since the answers to the questionnaires were kept anonymous, the responses were compiled as a summary without revealing any personal information.
- The names and titles of the evaluators were collected with consent as they were informed.

5.2.2 Legal

- The languages, frameworks and tools used within the project were under an open-source license or are provided to students with free usage.
- The dataset used for this research project is available publicly where all of its rows have been deidentified.
- The codebase of the research project is open source and available on GitHub.

5.2.3 Ethical

- The individuals who participated in filling the questionnaire were informed regarding the process and project they were contributing towards.
- The thesis is authentic; it is not fabricated, false, or copied. All the data and information presented were accurate and supported by appropriate citations and references.

5.2.4 Professional

- There was no illicit or pirated software or equipment used to make the prototype. Throughout the procedure, only open-source or student licenses were applied.
- The entire study process was conducted with strict adherence to research guidelines.
- The responses received have not been tampered with or falsified.

5.3 Chapter Summary

This chapter defines the social, legal, ethical, and professional issues that may arise during the project timeline. The chapter also breaks down each category with steps taken to mitigate these issues.

CHAPTER 6: DESIGN

6.1 Chapter Overview

This chapter describes the system's design, which includes a tiered architecture, a UML component diagram, level 1 and level 2 data flow diagrams, system sequence diagrams, the user interface design, and a diagram of the system's process flow.

6.2 Design Goals

Table 17 - Design Goals

Design Goal	Description
Performance	To ensure that the patient's health state is not critical, the system must be able to produce results as fast as possible, making system performance a top priority.
Quality & Correctness	A late or inaccurate assessment of the patient's health status could have fatal consequences; as a result, the system's accuracy and quality should be at the highest level possible with the data available at the moment.
Scalability	Due to the nature of the system, it must be easily expandable to accommodate future development and be able to manage hundreds or thousands of users so that inputs and generation can occur in the future with the least amount of work.
Usability	The primary goal of this system is to make it easier for patients to monitor their vital signs from the comfort of their own homes, hence it is crucial that it be simple to comprehend and use.
Adaptability	The system should enable the models used to be quickly swappable for improved performance and adaptation to the newer requirements based on the data currently available and anticipated requirements in the future. By ensuring the system can adapt to such changes, we can also ensure that it will function normally during the upgrading process.

6.3 High-Level Design

6.3.1 Three-Tiered Architecture

The system’s high level architecture design is showcased in the diagram below utilizing the tiered architecture containing 3 distinct tiers – The frontend (Presentation tier), The backend (Logic tier) and the Data tier.

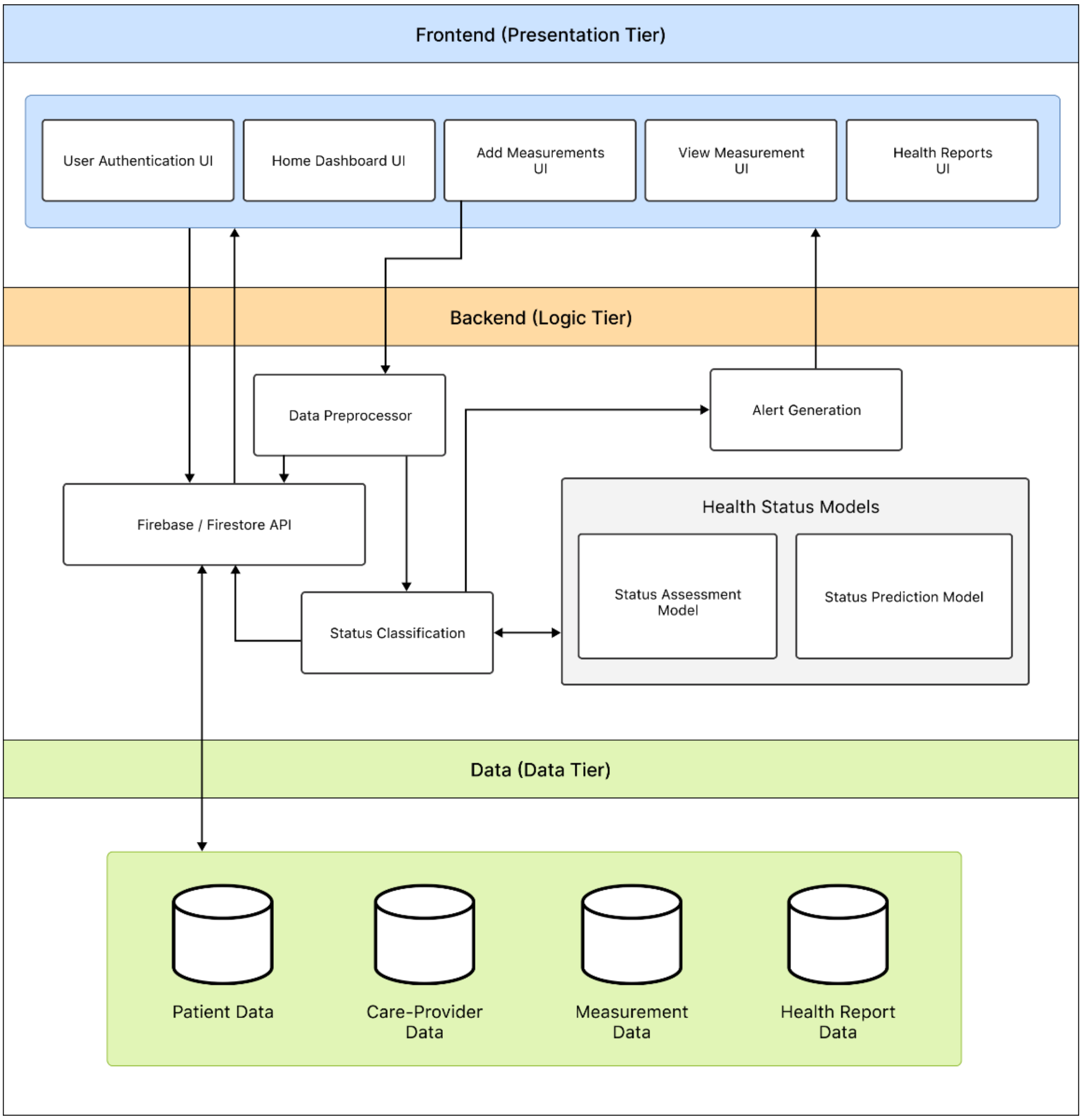


Figure 7 - High Level Architecture (Self-Composed)

6.3.2 Discussion of tiers

6.3.2.1 Frontend (Presentation Tier)

1. User Authentication UI – The user interface which allows users to login/register into the system.
2. Home Dashboard UI – The main dashboard homepage user interface which displays the patient's last recorded measurements, any pending tasks etc.
3. Add Measurements UI – This interface is very adaptive to the measurement types; it allows the patient to input the measurement and relevant information to the vital sign.
4. View Measurement UI – This UI allows the patient to view all statistics and information regarding the measurement added.

5. Health Reports UI – This user interface allows the patient to view a list of previous health status assessments including classifications and predictions done by the system.

6.3.2.2 Backend (Logic Tier)

1. Data Preprocessor – To ensure that all data received from the front-end user interfaces is suitable for the models and database by preprocessing it before it is used within the system.
2. Firebase/Fire-store API – The primary API would be utilized to establish a connection to the database for the purpose of reading and writing data.
3. Status Classifier – the main module that is activated by the system to utilize the models whenever an assessment is necessary.
4. Health Status Models – The models that the system uses to produce health status assessment classification and health status predictions for the patient.
5. Alert Generation - This module deals with the system's alert creation when an evaluation is available and there is a need to inform the patient and a healthcare professional immediately.

6.3.2.3 Data (Data Tier)

1. Patient Data – The data gathered and stored on the patient.
2. Care Provider Data – A collection of the care providers' data required by the system to function.
3. Measurement Data – Data which is retrieved from the patient when they input it into the system to be monitored.
4. Health Report Data – A collection of data containing health status assessment reports of both classification and prediction assessments of the patient's health status.

6.4 System Design

6.4.1 Choice of Design Paradigm

The author had decided to switch the design methodology from Object Oriented Analysis and Design Methodology (OOADM) which was mentioned initially during the project proposal to Structured Systems Analysis and Design Methodology (SSADM) naturally based on the factors that the main core machine learning components do not specifically gain any benefit from an object-oriented design, also for the ease of implementation of the minimum viable product.

6.5 Design Diagrams

6.5.1 Data Flow Diagrams

6.5.1.1 Data Flow Diagram – Level 1

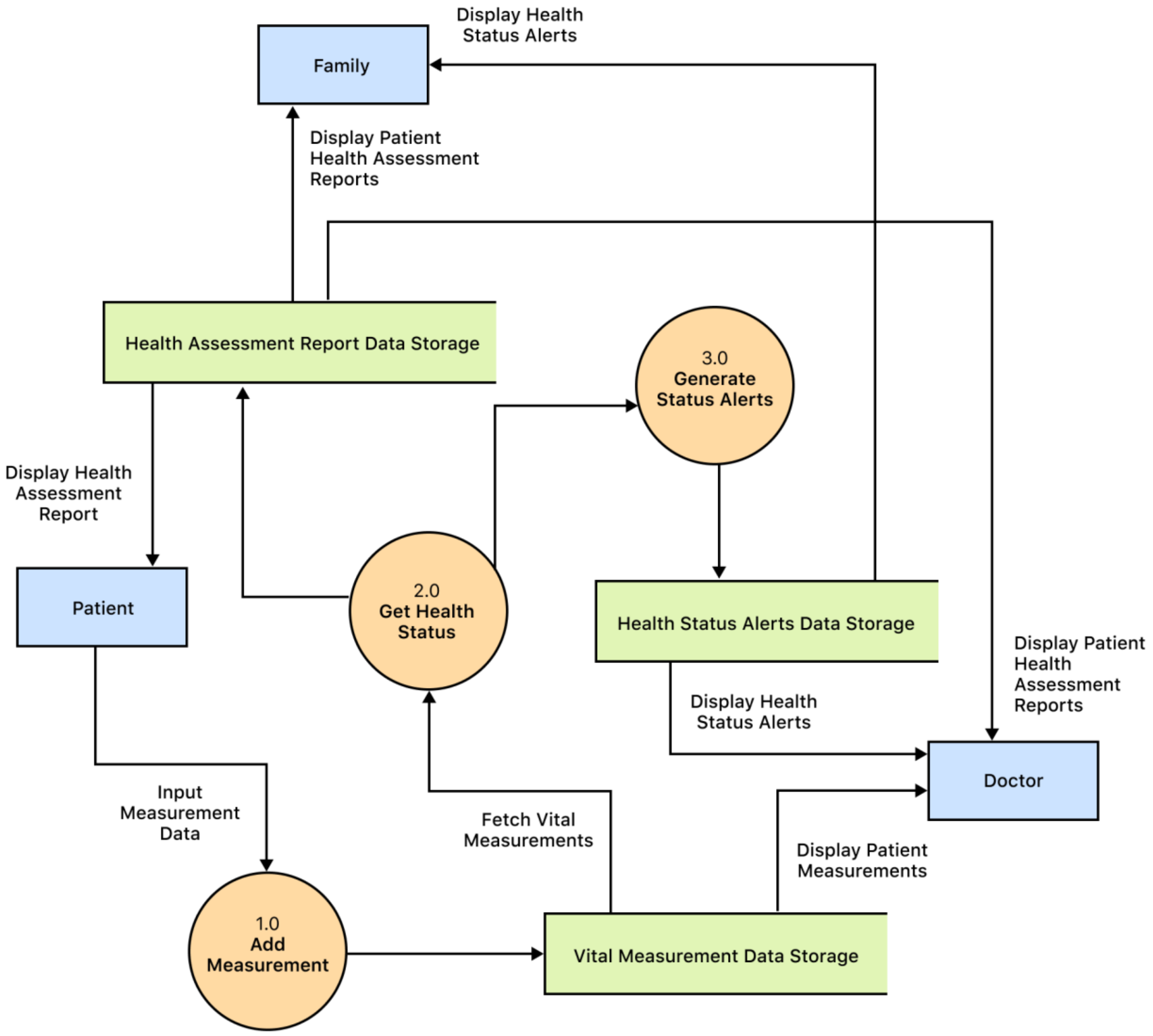


Figure 8 - Data Flow Diagram - Level 1 (Self-Composed)

6.5.1.2 Data Flow Diagram – Level 2

6.5.1.2.1 Add Measurements

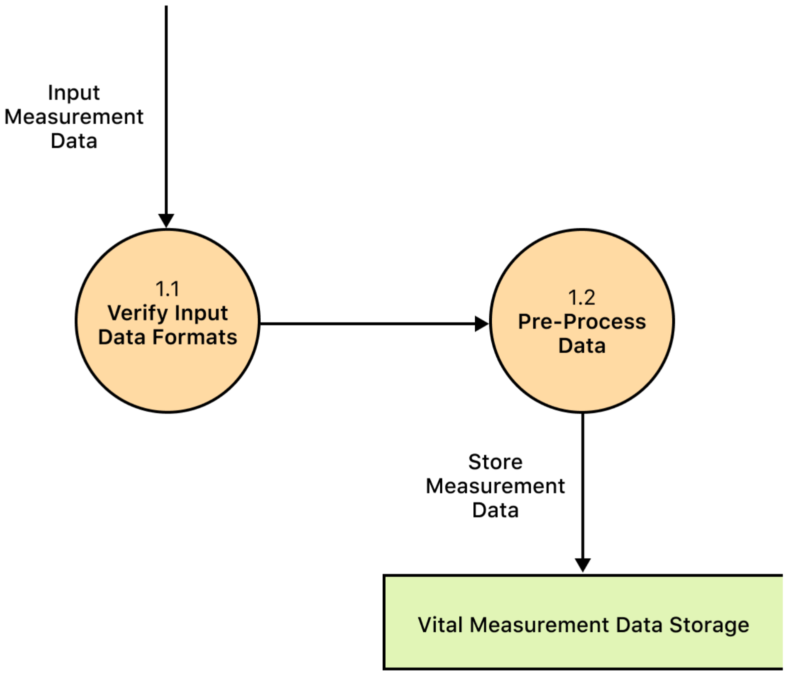


Figure 9 - Data Flow Diagram - Level 2 - Add Measurements (Self-Composed)

6.5.1.2.2 Get Health Status

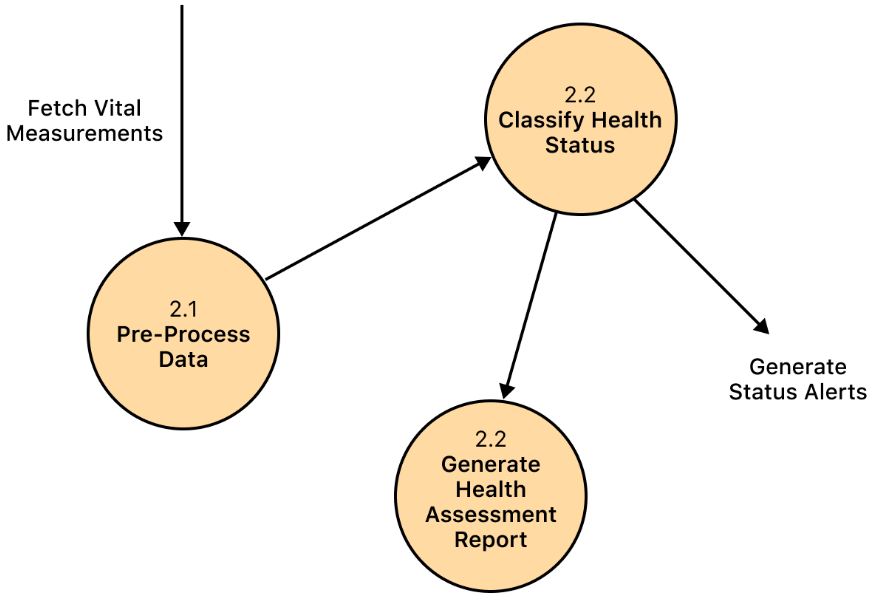


Figure 10 - Data Flow Diagram - Level 2 - Get Health Status (Self-Composed)

6.5.2 Sequence Diagram

6.5.2.1 Input Measurement

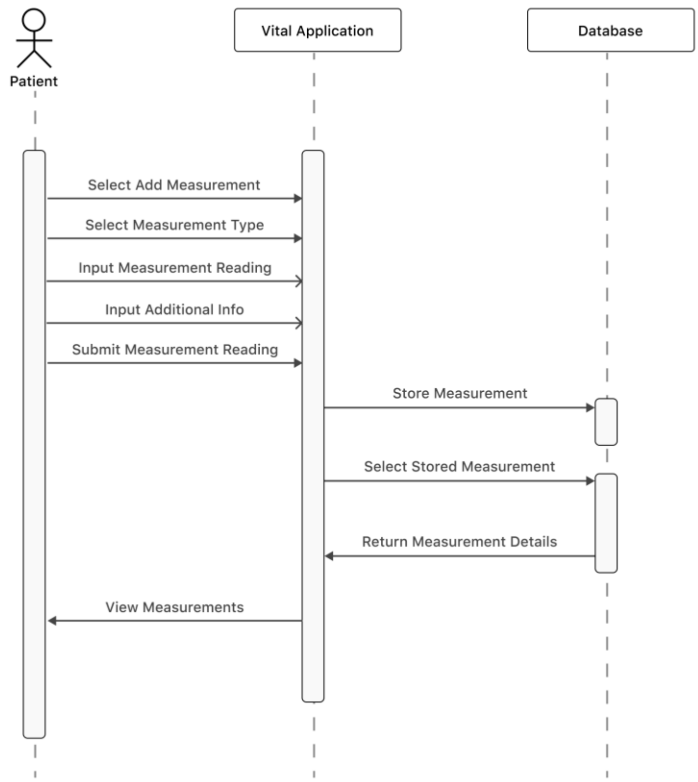


Figure 11 - Sequence Diagram - Input Measurements (Self-Composed)

6.5.2.2 View Patient Assessment Report

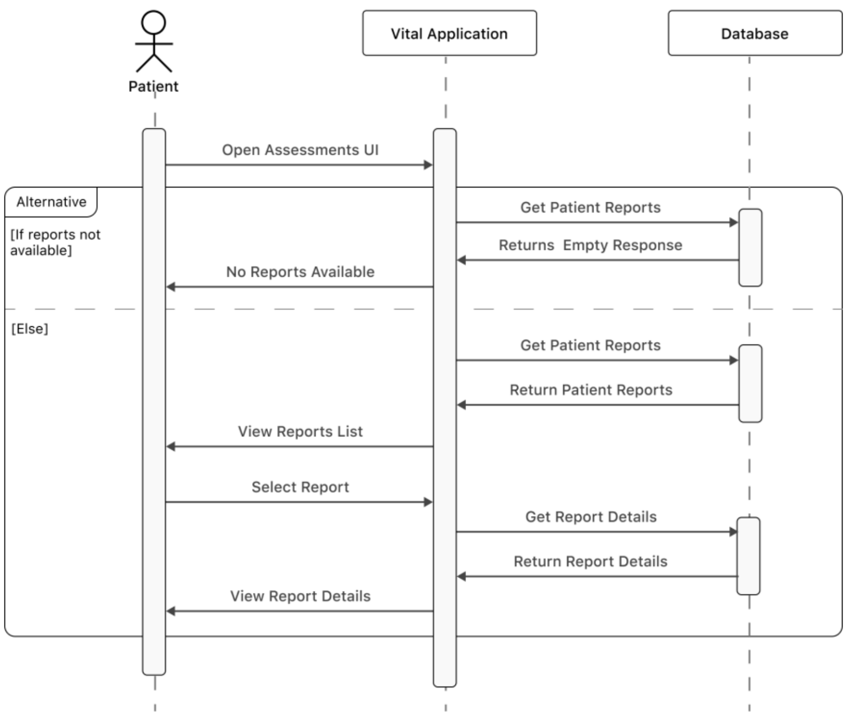


Figure 12 - Sequence Diagram - View Patient Assessment Report (Self-Composed)

6.5.3 UI Design

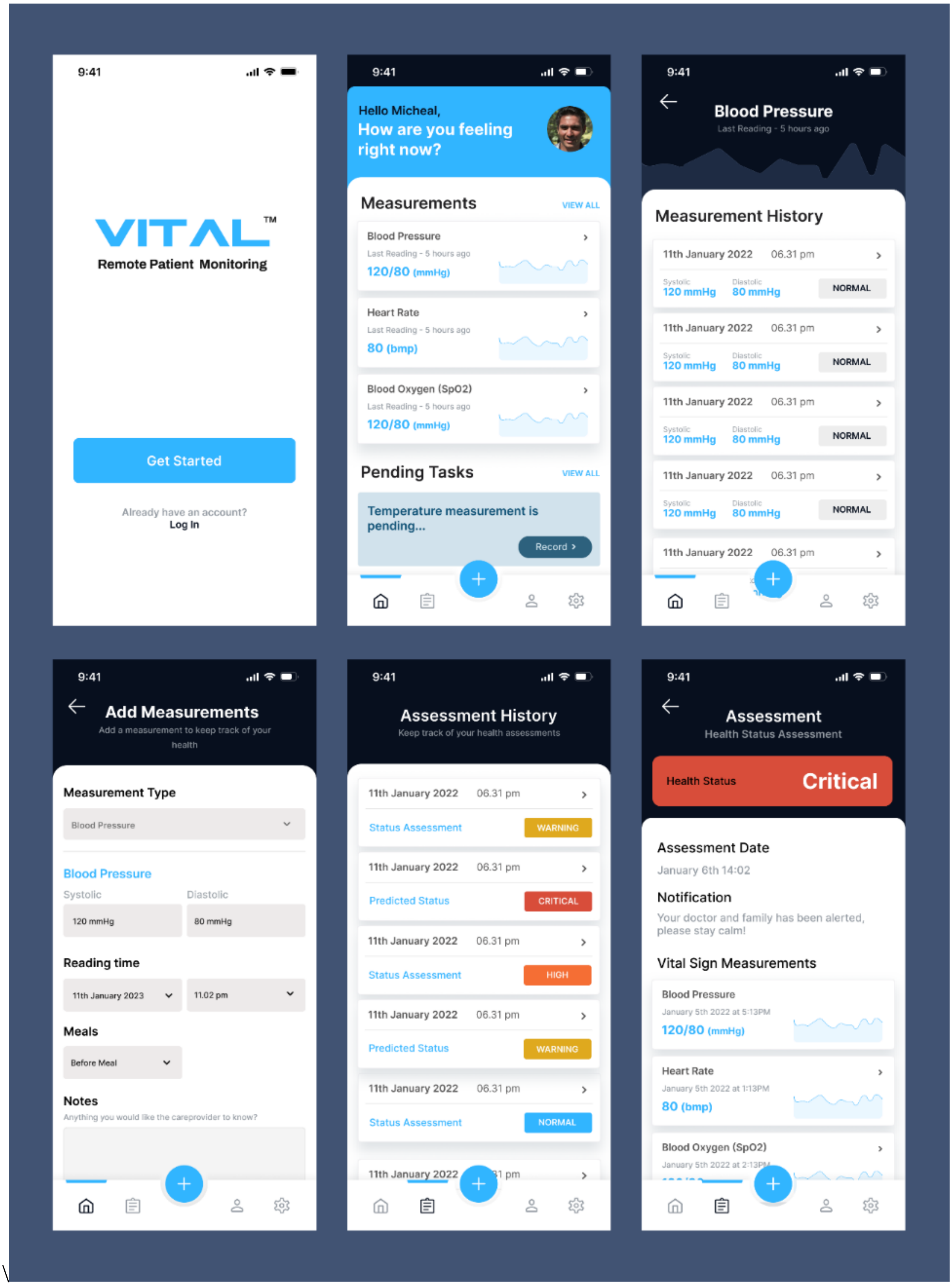


Figure 13 - User Interface Design (Self-Composed)

6.5.4 System Process Flow Diagram

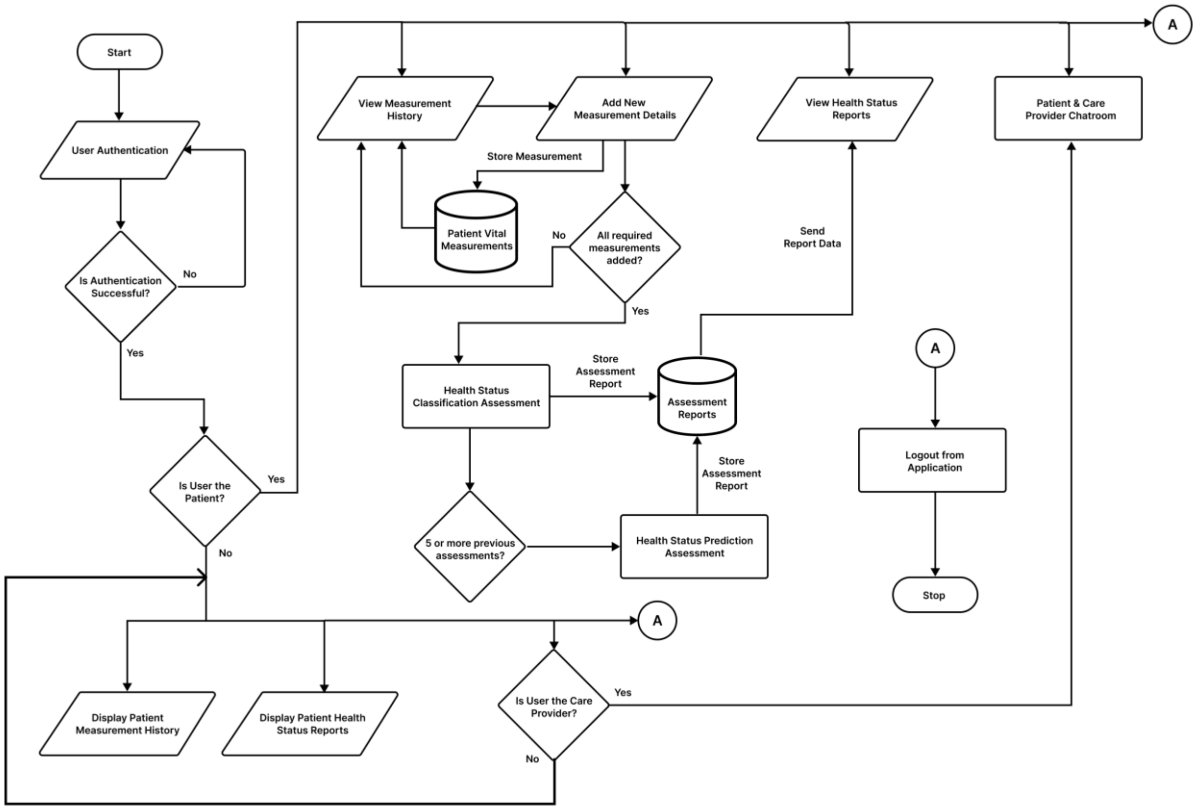


Figure 14 - System Process Flow Diagram (Self-Composed)

6.6 Chapter Summary

This chapter presented the design of the system, which includes a tiered architecture, level 1 and level 2 data flow diagrams, system sequence diagrams, and a detailed description of the user interface design. The chapter also included a diagram showing the process flow of the system. This chapter provided a comprehensive understanding of the system design and how the different components and processes fit together to form the system as a whole.

CHAPTER 7: IMPLEMENTATION

4.1 Chapter Overview

This chapter provides an overview of the core implementation of the research project, including the technologies, languages, and supporting tools used, and explains the rationale behind the author's choices.

7.2 Technology Selection

7.2.1 Technology Stack

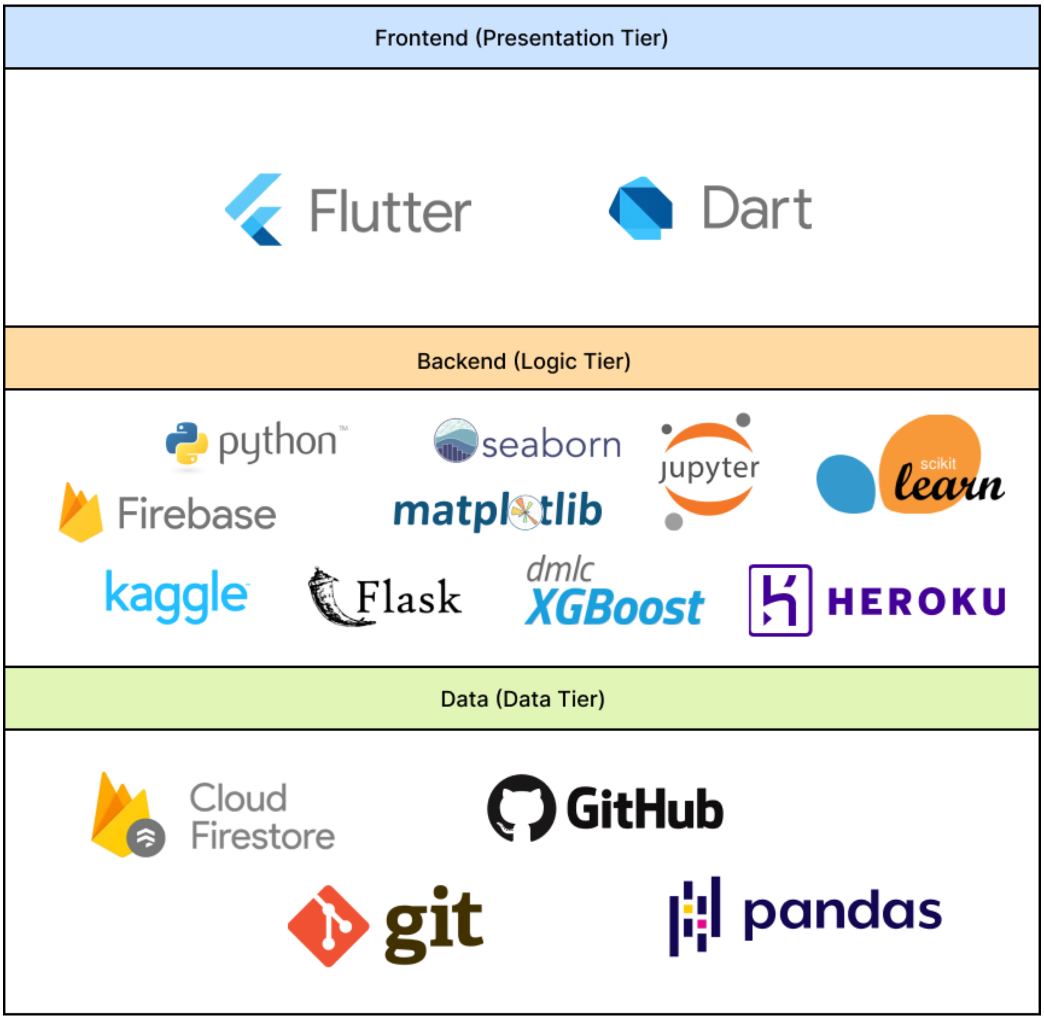


Figure 15 - Technology Stack (Self-Composed)

Windows Operating System will be utilized as the default operating system because all necessary tools are readily available.

7.2.2 Dataset Selection

Due to this being a data science project at its core, the author had to make sure that enough data were collected to be sufficient for the assessments alongside it being a widely trusted source for medical data.

The dataset needed to include triage assessments of a large number of patients for the system to be sufficiently trained. The assessments would include details of the patient's vital signs, as well as the acuity assessment by the doctor. The acuity was calculated using the Emergency Severity Index (ESI), which is a five-level triage algorithm which classifies the patient into five groups, ranging from (1) being most urgent to (5) being least urgent. This allows the risk level of the patient to be identified and care to be prioritized with limited resources.

The patient vital sign triage dataset used in this project is the **Medical Information Mart for Intensive Care IV (MIMIC IV) ED Dataset** (Johnson et al., 2022) which was provided by the Massachusetts Institute of Technology. To receive access to the dataset, the author was required to undergo Data or Specimens Only Research training course certification for human subjects' research and was also required to have credentialed access to the PhysioNet platform.

7.2.3 Development Frameworks Used

Table 18 - Development Frameworks Used

Framework	Usage Justification
Flutter	Used to create the user interface of the system, Flutter is an open-source framework developed by Google for building multi-platform applications using a single codebase and has the capability to create beautiful interfaces with its vast number of packages and widgets from the community.

7.2.4 Programming Languages Used

Table 19 - Programming Languages Used

Language	Usage Justification
Python	Used in the development of the core data science models. It is a general-purpose language however it is heavily used in many

	data science projects with a large collection of libraries capable of supporting the project development.
Dart	Used in front-end development for the flutter framework. It is a client-optimized language allowing people to create fast apps for any platform.

7.2.5 Libraries Used

Table 20 - Libraries Used

Library	Usage Justification
Pandas	Used during the model development mainly for data reading and manipulating. This library has a large number of functionalities that could be used for data cleaning, filtering and manipulating the data for data analysis
Scikit-learn	A python library, used during the model development mainly for preprocessing, training and evaluation of the model
Matplotlib	Used for data visualization and is highly customizable for data analysis
Seaborn	Used for data visualization, it's an extension of matplotlib library that also allows to plot various graphs from pandas library and NumPy library
NumPy	It is library that has a large collection of mathematical functions that can be used on large arrays/matrices
Joblib	Used for the serialization of the models developed to save the trained models making it easy to be reused for evaluation

7.2.6 IDEs Used

Table 21 - IDEs Used

IDE	Usage Justification
Jupyter Notebook	Used for the data science model development on the local environment

Kaggle	Kaggle provides 32GB of RAM and several GPUs which allowed easier and quicker model development on the cloud.
VSCode	Used during application development. it is very lightweight and has tons of features that assist during the development.
Android Studio	Used during application development. Contains inbuilt mobile emulators and works well with the flutter framework.

7.2.7 Summary of Technology Selection

Table 22 - Summary of Technology Selection

Component	**Tools Used**
Development Frameworks	Flutter
Programming Languages	Dart, Python
Libraries	Pandas, Scikit-learn, Matplotlib, Seaborn, NumPy
IDEs	Android Studio, Jupyter Notebook, VSCode
Version Control	Git, GitHub
Database & Hosting	Firebase, Azure

7.3 Implementation of Core Functionalities

The core functionalities of this research project include the implementation of the hybrid health status assessment, the forecasting of the health status through vital sign forecasting.

7.3.1 Hybrid Health Status Assessment

To implement the health assessment module of this research, the author used the triage table of the dataset selected. The vital sign columns are used as features and the acuity column is taken as the target.

```
# Normalization
scaler = MinMaxScaler()
columns = ["temperature", "heartrate", "resprate", "o2sat", "sbp", "dbp"]
dataset[columns] = scaler.fit_transform(dataset[columns])
```

```
import pickle
min_value, max_value = scaler.data_min_, scaler.data_max_
# Dump the model to a file
with open('status_scaler.pkl', 'wb') as file:
    pickle.dump((min_value, max_value), file)
```

Figure 16 - Implementation code segment: Data Normalization Scaler (Self-Composed)

Machine learning models usually perform best when all the data is trained across similar scale (Protić and Stanković, 2020). The code segment above is used to normalize the data using the MinMaxScaler in order to make it usable for the model to understand and train with easily to ensure the best performance. The min and max values of the scaler are then exported to be used when external input data is required in order to keep the same range.

```
from imblearn.combine import SMOTEENN
```

```
# Get the features and target columns
X = dataset2[['temperature','heartrate','resprate','o2sat','sbp','dbp']]
y = dataset2['acuity']
```

```
# Perform SMOTE and ENN
smote_enn = SMOTEENN(sampling_strategy='not majority')
X_resampled, y_resampled = smote_enn.fit_resample(X, y)
```

```
# Combine the resampled features and target into a new DataFrame
dataset_resampled = pd.DataFrame(np.column_stack([X_resampled, y_resampled]), columns=dataset.columns)
```

Figure 17 - Implementation code segment: Combination Sampling (Self-Composed)

Due to the effect of imbalance of the target class on the performance of the model, the author performed a combination of oversampling the majority classes and under sampling the minority classes as seen in the code segment provided above.

After this, the data was ready to be split for training and validation, where the author used 80% for training and 20% as test data.

```
#Ensemble method for combining multiple classifiers
from sklearn.ensemble import VotingClassifier

# Model Classifiers Used
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import RandomForestClassifier

#Initializing Classifiers
dt_model = DecisionTreeClassifier(random_state=42)
gb_model = GradientBoostingClassifier(random_state=42)
rf_model = RandomForestClassifier(random_state=42)
```

```
# Creating the hybrid model using the voting classifier
hybrid_model = VotingClassifier(estimators=[('rf', rf_model), ('dt', dt_model),('gb', gb_model)], voting='hard')
```

```
# Fitting the voting classifier on the training data
hybrid_model.fit(X_train, y_train)
```

```
VotingClassifier(estimators=[('rf', RandomForestClassifier(random_state=42)),
                             ('dt', DecisionTreeClassifier(random_state=42)),
                             ('gb',
                              GradientBoostingClassifier(random_state=42))])
```

Figure 18 - Implementation code segment: Hybrid Classifier Model (Self-Composed)

The above code segment implements the hybrid classification model using ensemble method of combining multiple models with the Voting Classifier. The model combines three classifiers with hard voting to ensure the predicted target value with majority votes.

```
@app.route('/status', methods=['POST'])
def status_predict():
    # Get the request data
    data = request.get_json()
    response = get_status(data)
    return jsonify(response)

def get_status(data):

    input_data = np.array(data['vitals'])  # Convert the data to a numpy array

    # Reshape the input data if it's a 1D array
    if input_data.ndim == 1:
        input_data = input_data.reshape(1, -1)

    with open('pickle/status_scaler.pkl', 'rb') as file:
        status_min_value, status_max_value = pickle.load(file)
    print('Loaded Status Scaler')

    normalized_input = (input_data - status_min_value) / (status_max_value - status_min_value)

    bucket = client.bucket('vitalrpm-models')
    blob = bucket.blob('hybridmodel.pkl')
    file_contents = blob.download_as_string()

    #Hybrid Health Status Model & Scaler
    status_model = pickle.loads(file_contents)
    print('Loaded Status Model')

    # Pass the scaled data to your model for inference
    acuity = status_model.predict(normalized_input)
    print("Patient Health Status Classification: ", acuity)

    # Map the prediction to a text status
    status = get_status_type(acuity)
    print("Patient Health Status: ", status)

    del status_model, bucket, blob, status_min_value, input_data, status_max_value, file_contents, normalized_input

    # Return the status as a JSON response
    response = {'status': status[0], 'vitals': data['vitals'], 'predicted_acuity': str(acuity[0])}
    return response
```

Figure 19 - Implementation code segment: Get Status Flask API (Self-Composed)

The code segment shown above is used in the Flask API, it takes the input vital sign data passed into it, loads the model from the cloud storage, performs a classification and returns the status level.

```
def get_status_type(prediction):
    print('Getting Patient Status')
    # Define a dictionary mapping each prediction value to a text status
    status_map = {
        1: 'status_critical',
        2: 'status_high',
        3: 'status_warning',
        4: 'status_normal',
        5: 'status_normal'
    }

    # Check if the prediction is a scalar value or an array of values
    if isinstance(prediction, (int, float)):
        return status_map[prediction]
    else:
        return [status_map[p] for p in prediction]
```

Figure 20 - Implementation code segment: Text Status Level Mapping (Self-Composed)

In order to map the predicted target value provided by the model and return a status level in text format to the API response, the below above segment is used.

7.3.2 Vital Sign Forecasting

To implement the health status forecasting module of this research, the author used the vital sign table of the dataset selected.

```
vitaldf  = vital_signs_df
# Resample the data to a daily frequency
vitaldf = vitaldf.resample('D').mean()
vitaldf
```

Figure 21- Implementation code segment: Data Frequency Resampled (Self-Composed)

In the above code segment, the author resamples the data to a daily frequency, as the data provided by the dataset did not have fixed time intervals and was heavily de-identified.

```
# Interpolate the missing values
vitaldf = vitaldf.interpolate()
vitaldf
```

Figure 22- Implementation code segment: Data Interpolation (Self-Composed)

To fill in the missing values, the author used the interpolate function as shown in the code segment above, this added new data points between the available values.

```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler(feature_range=(-1, 1))
transformer = scaler.fit(vitaldf)
```

Figure 23- Implementation code segment: Normalization Scaler (Self-Composed)

Similarly as like the status model, the data used was scaled and normalized using the above code segment ensuring the model performs the best.

```
from sklearn.metrics import mean_absolute_error as mae
from sklearn.metrics import mean_squared_error as rmse
```

```
train_size = int(len(series_transformed) * 0.8)
train_data = series_transformed[:train_size]
test_data = series_transformed[train_size:]
```

```
import xgboost as xgb
```

```
## Fitting models and checking the performance
def eval_model(model):
    model.fit(train_data[:-1], train_data[1:])
    forecast = model.predict(test_data[1:])
    print('model XGB obtains MAE: {:.2f}%'.format( mae(test_data[1:], forecast)))
    print('model XGB obtains RMSE: {:.2f}%'.format(rmse(test_data[1:], forecast)))
    return model,forecast

m_xgb, f_xgb = eval_model(xgb.XGBRegressor(objective='reg:squarederror',n_estimators=2000, max_depth=5))
```

Figure 24- Implementation code segment: Fitting the XGBoost Model (Self-Composed)

The above code segment shows how the data was split for training and testing, creating the model using XGBoost doing multivariate forecasting of the vital signs with slight parameter optimization in order to receive the best results for the data used. The model was then exported through the inbuilt method as shown in the code segment below.

```
# save to JSON
m_xgb.save_model("xgb_forecasting.json")
```

Figure 25- Implementation code segment: Saving the Model (Self-Composed)

The below code segment is used in the backend Flask API, it takes the input of a list of vital sign data sent through the POST request, loads the model file, preprocesses the input data and passes it onto the forecasting model which returns the forecasted vitals of the patient. The vitals are then passed back to the status model in order to receive the forecasted status level.

```
@app.route('/forecast', methods=['POST'])
def forecast_predict():
    # Get the request data
    data = request.get_json()

    array_data = np.array(data['vitals'])  # Convert the data to a numpy array

    # Reshape the input data if it's a 1D array
    if array_data.ndim == 1:
        array_data = array_data.reshape(1, -1)

    input_data = np.array(array_data)  # Convert the data to a numpy array

    with open('pickle/forecast_scaler.pkl', 'rb') as file:
        forecast_scaler = pickle.load(file)
    print('Loaded Forecast Scaler')

    normalized_data = forecast_scaler.transform(input_data)

    data = xgb.DMatrix(normalized_data) # Converting the numpy array to a DMatrix object

    print('Begin Prediction')
    forecast_model = xgb.Booster()
    forecast_model.load_model('xgb_forecasting.json')
    print('Loaded Forecast Model')

    forecast = forecast_model.predict(data) # Forecasting each row of the data
    forecast = forecast_scaler.inverse_transform(forecast)

    # Get the last value of your predictions array as the next day's value
    np_array = forecast[-1]
    print("Forecasted Vitals: ", np_array)

    # convert to a nested Python list
    nested_list = np.squeeze(np_array).tolist()

    rounded_list = [round(num, 2) for num in nested_list]
    print("Forecasted Vitals: ", rounded_list)

    status_response = get_status({'vitals': rounded_list})
    status = status_response['status']
    print("Patient Health Status: ", status)

    del forecast_model, forecast, status_response, normalized_data, array_data, input_data, data, nested_list, np_array

    # # Return the status and forecast as a JSON response
    response = {'forecasted_vitals': rounded_list, 'status': status}

    return jsonify(response)
```

Figure 26- Implementation code segment: Flask API for Status Forecasting (Self-Composed)

7.4 Chapter Summary

In this chapter, the author showcased the implementation of proposed project, highlighting the dataset that was selected for usage, the development frameworks, programming languages, libraries, and IDEs that were utilized. The chapter provided a justification for the usage of each technology and presented the initial core implementation code of the project, which was used during the development of the model & backend API. This chapter offered insight into the technical aspects of the project and how they were executed to implement the prototype.

CHAPTER 8: TESTING

8.1 Chapter Overview

This chapter describes the testing that was done to make sure all functionalities worked as expected. The objectives and goals of the testing phase alongside the model testing, benchmarking, functional testing, non-functional testing, module and integration testing are just a few of the testing objectives and techniques that will be covered.

8.2 Objectives & Goals of Testing

During the testing phase of a software application, the main goal is to ensure that the system functions as required. Some of the objectives during this phase are as follows:

- To ensure all developed models are working and have been tested thoroughly to achieve satisfactory results.
- To validate whether the system meets the functional requirements mentioned based on the MoSCoW technique assigned used.
- To ensure any important nonfunctional requirement is satisfied.
- To identify possible improvements that can be made and fix any bugs within the system to ensure good experience.

8.3 Testing Criteria

To meet the requirements of the proposed system, the author has set two criterion which will be test, they are as follows:

1. Functional Quality – Tests done that focus on the technical aspects of the system to see how well it fits the functional requirements of the system designed.
2. Structural Quality – Tests done to check whether the system meets the non-functional requirements ensure that the system’s performance is satisfactory.

8.4 Model Testing & Evaluation

8.4.1 Model Testing

Since two models were implemented for this research project, the author will go into detail how he performed testing for each.

8.4.1.1 Health Status Assessment Model Testing

The first model, which is the health status assessment model, the author had mainly focused on evaluation metrics for testing the accuracy and performance of the model.

```python
# Making predictions on the test set
predictions = hybrid_model.predict(X_test)
```

```python
print(predictions)
```

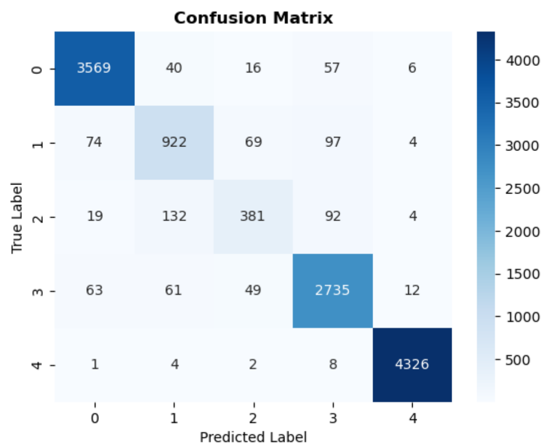
```
[1. 5. 5. ... 1. 4. 4.]
```

```python
# Evaluating the model on the test set
accuracy = hybrid_model.score(X_test, y_test)
print("Accuracy: ", accuracy)
```

```
Accuracy:  0.9364356901828454
```

Figure 27 - Getting Model Accuracy (Self-Composed)

As shown in the above code segment, predictions were done on the test data and an accuracy score was generated. A confusion matrix and classification report were also generated to retrieve the performance of the dataset as shown in the figure below.




```
Accuracy score: 93.64356901828455
Recall score: [0.96773319 0.79073756 0.6066879  0.93664384 0.99654457]
Precision score: [0.95786366 0.79551337 0.73694391 0.91502175 0.99402574]
              precision    recall  f1-score   support

         1.0       0.96      0.97      0.96      3688
         2.0       0.80      0.79      0.79      1166
         3.0       0.74      0.61      0.67       628
         4.0       0.92      0.94      0.93      2920
         5.0       0.99      1.00      1.00      4341

    accuracy                           0.94     12743
   macro avg       0.88      0.86      0.87     12743
weighted avg       0.93      0.94      0.94     12743
```

Figure 28 - Confusion Matrix & Classification Report (Self-Composed)

Also, since the model was trained where the target classes were still imbalanced, A Cohen's kappa score was generated, the below code segment shows how the author generated it.

```
from sklearn.metrics import confusion_matrix

# Calculate the confusion matrix
cm = confusion_matrix(y_test, predictions)

# Calculate the observed agreement
po = sum(np.diag(cm)) / np.sum(cm)

# Calculate the expected agreement
pe = np.sum(np.sum(cm, axis=0) * np.sum(cm, axis=1)) / np.sum(cm)**2

# Calculate Cohen's kappa score
kappa = (po - pe) / (1 - pe)

print("Cohen's kappa score: ", kappa)
```

```
Cohen's kappa score:  0.9135138176017596
```

Figure 29 - Generating Cohens Kappa Score (Self-Composed)

8.4.1.2 Vital Sign Forecasting Model Testing

For the vital sign forecasting model, the test data was saved separately and was used to perform forecasting the vital signs. Below are the figure plots for each of the vital signs forecasted.

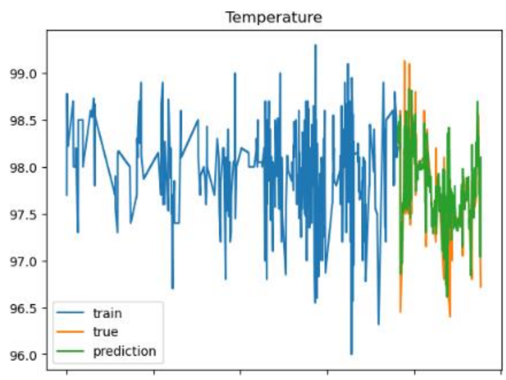


Figure 30 - Forecasting Temperature on Test Data (Self-Composed)

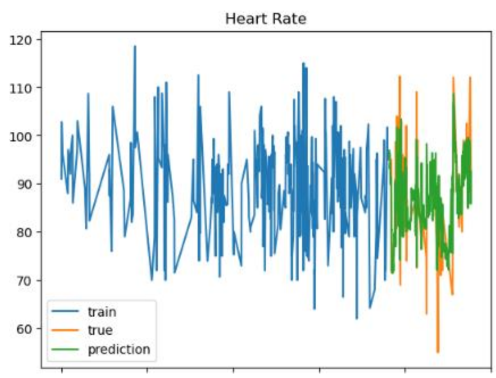


Figure 31 - Forecasting Heart Rate on Test Data (Self-Composed)

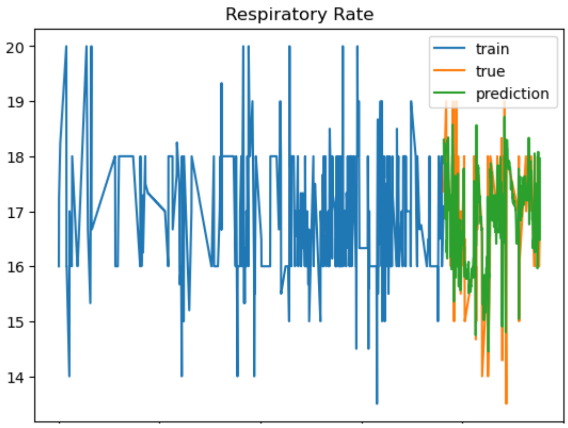


Figure 32 - Forecasting Respiratory Rate on Test Data (Self-Composed)

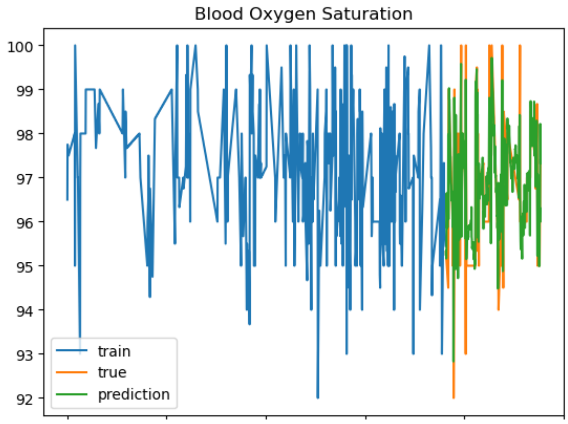


Figure 33 - Forecasting Blood Oxygen Saturation on Test Data (Self-Composed)

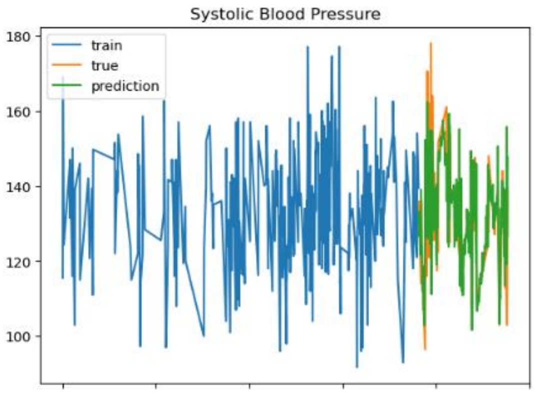


Figure 34 - Forecasting Systolic Blood Pressure on Test Data (Self-Composed)

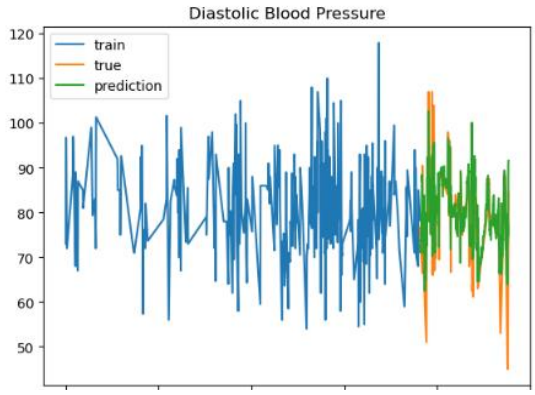


Figure 35 - Forecasting Diastolic Blood Pressure on Test Data (Self-Composed)

We can see that even with the heavy variance in the data due to irregular frequency of rows in the dataset, the model was still capable of studying and forecasting it almost accurately.

8.4.2 Model Evaluation

With the usage of the evaluation metrics highlighted during the literature review phase mentioned in **Chapter 2**, the following tables showcase the models performance metrics. The below table uses the metrics gathered for classification of the health status to evaluate the model.

Table 23 - Model Evaluation - Health Status Assessment

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Decision Tree Model	47.19%	33.37%	35.77%	35%
Random Forest Model	64.77%	61.88%	96.27%	75%
Gradient Boosting Model	66.54%	68.92%	85.56%	76%
Selected – Hybrid Model	**93.64%**	**95.78%**	**96.77%**	**96%**

The below table showcases the developed vital sign forecasting model in comparison to other time series models taken from the darts library which were developed and tested with the same data.

Table 24 - Model Evaluation - Vital Sign Forecasting

Model	**MAE**	**RMSE**
CatBoostModel	0.28%	0.34%
LightGBMModel	0.34%	0.44%
NHitsModel	0.36%	0.44%
Selected: XGBoost Model	**0.11%**	**0.03%**

With the metrics used, the lower the percentage gained the higher the accuracy and performance of the model however these metrics weren't the only thing taken into consideration for evaluation as when comparing the plots most models were not being trained properly and were not learning from the data therefore the selected model and approach is satisfactory.

8.5 Benchmarking

In order to do benchmarking as mentioned during the literature review, a standard dataset was meant to be used across multiple studies in order to provide comparisons between the systems available however most studies utilized datasets that were not available to the author therefore the author could not perform benchmarking.

8.6 Functional Testing

The system was tested against the functional requirements mentioned during the requirement gathering phase, the table below shows a breakdown of the tests performed.

Table 25 - Functional Requirement Testcases

Test Case	FR ID	Requirement	Expected Result	Actual Result	Result Status
1	FR1	Users must be able to add vital sign measurements into the system	Allow users to change app language	Allow users to change app language	Passed
2	FR2	Doctors should be able to monitor patient vital signs and status through the system	Doctor can view patient vital signs and status assessments and forecasts	Doctor can view patient vital signs and status assessments and forecasts	Passed
3	FR3	The system should be able to fetch the patient vital signs collected over a period of time and make a prognosis	System fetches vitals from status reports and generates a forecasted status level	System fetches vitals from status reports and generates a forecasted status level	Passed
4	FR4	The system should take the patient vital signs collected and provide an immediate diagnosis/status check	Users can input vital signs have the ability to generate an assessment based on it.	Users can input vital signs have the ability to generate an assessment based on it.	Passed
5	FR5	The system should be able to generate daily/weekly health reports based on data collected/generated	The users can generate the health status assessments once data requirements are met.	The users can generate the health status assessments once data requirements are met.	Passed
6	FR6	The system should alert healthcare providers	The healthcare providers view the	The healthcare providers view the	Passed

Test Case	FR ID	Requirement	Expected Result	Actual Result	Result Status
		when there is an irregularity in patient vital signs and health status level	health status assessment with the status level and forecast level to take preventive measures	health status assessment with the status level and forecast level to take preventive measures	
7	FR9	Include multi-language translation support to the system to allow support for a wider variety of people	Allow users to change app language	Allow users to change app language	Passed

8.7 Non-Functional Testing

8.7.1 Performance Testing

8.7.1.1 Model & Backend Performance

The model was deployed and stored with the usage of Heroku with the basic Dyno, GitHub Actions and Google Cloud Storage, and therefore does not utilize the device's processing power to stay online. If the models and API backend were used at a larger scale, the rest of the technologies utilized will also need to be scaled in order to meet requirements and handle multiple requests.

8.7.1.2 Frontend Performance

Since the frontend of the application uses the flutter framework, there are several tools provided by the framework to debug and test the application. The image below showcases the performance overlay which shows two graphs on the performance statistics of the UI in order to understand why and where the app could be skipping frames.

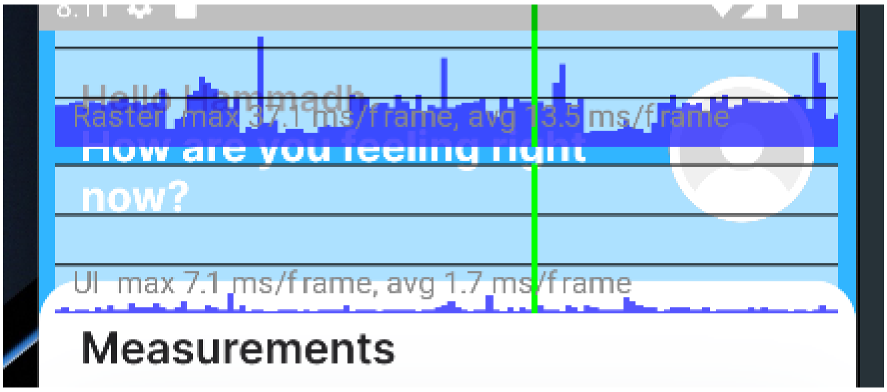


Figure 36 - Flutter Performance Overlay (Self-Composed)

8.7.2 Usability Testing

With the evaluations gathered from the experts & potential system users as mentioned during the evaluation phase of this research in **Chapter 9**, they found the system very easy to understand and navigate and felt that the design is very user friendly.

8.7.3 Code Quality Testing

As the project code and version control was hosted on GitHub, the author implemented CodeFactor to ensure that the repositories were maintained with no vulnerabilities. The below image shows the CodeFactor result for the project repository.

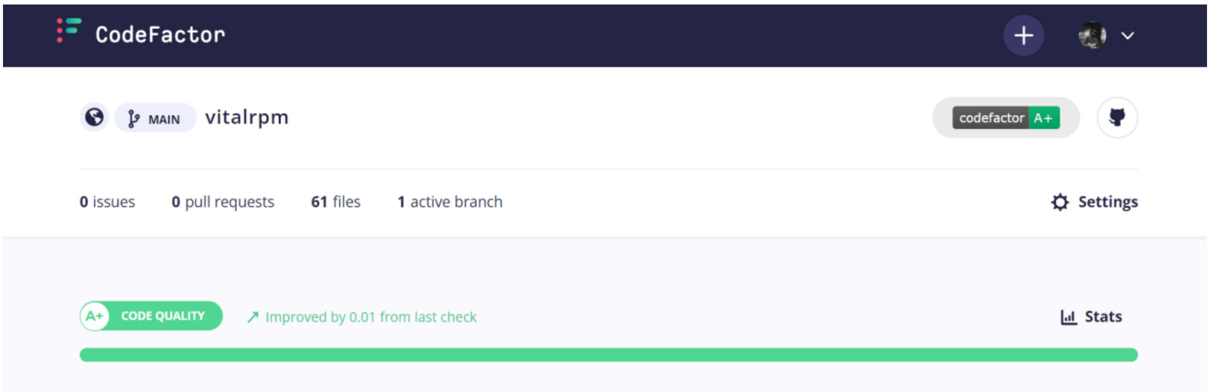


Figure 37 - CodeFactor - Vital:RPM Repository (Self-Composed)

8.7.4 Compatibility Testing

The system was developed on the flutter framework which allows the usage of a single codebase for multiple platforms. Currently the application developed and tested to be running smoothly on android devices but can also support IOS with some dependency modifications however this required a device that ran on MacOS, the application can also be used as a windows desktop app with some UI and dependency modifications and can be run on the web browser showcasing a wide range of compatibility with slight modification requirements.

8.7.5 Non-Functional Testcases

Table 26 - Non-Functional Testcases

Testcase	NFR ID	Result
1	NFR1	It was developed with some of the best technologies ensuring the system performs to standard.
2	NFR2	The system developed is built around user experience with great UI design & is very user friendly.
3	NFR3	The models created for the research after testing provide satisfactory accuracy.
4	NFR4	Firebase Authentication & Fire store was utilized ensuring good security. However further measures can be taken to improve the security.
5	NFR5	With the usage of cloud firestore and flutter, the system ensures scalability. However, the Heroku dyno package will need to be upgraded to make sure the system stays online in terms of usage for scalability

8.8 Module & Integration Testing

Table 27 - Module & Integration Testing

Module	Input	Expected Output	Actual Output	Status
Health Status Assessment Classifier	Patient vital signs (temperature, heartrate, respiratory rate, oxygen saturation, blood pressure - systolic and diastolic)	Acuity Health Status Level based on Emergency Severity Index	Acuity Health Status Level based on Emergency Severity Index	Passed
Vital Sign Forecaster & Forecasted Status Classifier	A list of patient vital signs over a period of time (minimum 7 rows)	Forecasted vitals & Forecasted Status Level	Forecasted vitals & Forecasted Status Level	Passed

8.9 Limitations of Testing Process

Even though the system was thoroughly tested, further testing could be performed through unit testing which the author did not conduct due to constraints in project delivery timeline however each phase was tested manually through debugging and validation.

8.10 Chapter Summary

In this chapter the author discussed the testing process performed for the research project outlining the objectives and goals of the testing phase, performed the model testing and benchmarking, performed black box testing on the functional requirements and module and integration testing, and used different analysis tools for the non-functional testing outlining the limitations of the testing phase.

CHAPTER 9: EVALUATION

9.1 Chapter Overview

In this chapter, the author will be going over the project evaluation based on the requirements gathered after a successful implementation of the system and testing phase. This chapter would also include self-evaluation of the project as well as evaluations from domain and industry experts.

9.2 Evaluation Methodology & Approach

The author has decided to carry out both quantitative and qualitative evaluation approaches based on the outputs of the system where techniques were extracted from the available literature around health status assessments and remote patient monitoring systems. A thematic analysis will be done on the feedback provided by the domain and industry experts.

The video presented to the evaluators demonstrating the research and solution which will be used for evaluations can be found here: ***https://youtu.be/ATf1hAAC7mI***

9.3 Evaluation Criteria

In order to evaluate and determine the value of the research undertaken, the following criteria will be used for analysis on the self-evaluation and expert feedback provided.

Table 28 - Evaluation Criteria

Criteria	**Purpose of Evaluation**
Choice of Research	To validate the importance of the research undertaken
Research Contribution	To identify the value of the contributions provided to the domain of health status assessments and remote patient monitoring, and other additional contributions done.
Quality of Documentation	To check and confirm whether the quality of the research undertaken throughout the research process is acceptable
Approach taken for Development	To determine whether the best approach was taken to solve the problem through the product implementation
Analysis of Results	To verify the metrics taken to test the system and analyze the results gained
Possible Improvements	To identify any improvements that can be implemented as future work relating to the product developed

Usability & User experience of MVP	To confirm whether the developed system provides good usability and ensures good user experience for the end users.

9.4 Self-Evaluation

Table 29 - Author's Self Evaluation

Criteria	Self-Evaluation
Choice of Research	The selected research area was selected in a domain that in-demand and has huge potential to be transformative to society
Research Contribution	The contributions provided through this research are very impactful to the domain allowing patients to automate the process of getting a health assessment on their vital signs. As this field is disruptive it opens pathways to future work within the field
Quality of Documentation	The research maintains the best possible standards ensuring the best structure of the research, the diagrams composed, and content of literature cited.
Approach taken for Development	Cutting edge technologies and tools were utilized to receive the best results throughout the development phase of this research to have the best approach was taken to ensure scalability for future development.
Analysis of Results	The testing phase of this research project provided satisfactory results of all the outputs received showcasing high accuracies overall throughout both models.
Possible Improvements	Based on the feedback gained from the experts, several improvements were noted as future work and limitations of the system as they could not be addressed due to time constraints.

Usability & User experience of MVP	The minimum viable product developed showcases a high level of consideration for the end users therefore ensuring a good system usability and overall good user experience through good design and support for multiple languages.

9.5 Selection of Evaluators

The evaluators selected can be broken down into the three categories as shown in the table below.

Table 30 - Categories of Evaluators Selected

CAT ID	Category Description
CAT1	Experts in the fields of Machine Learning, Data Science
CAT2	Experts from the domains of Remote Patient Monitoring, Telehealth and Healthcare
CAT3	The possible end users of the system

9.6 Evaluation Results & Expert Opinions

The following table includes a summary of opinions provided by the evaluators. Full evaluations which the author had received are available and can be found in the Appendix - ***Evaluator Feedback***

Table 31 - Summary of Evaluator Feedback

Criteria	CAT ID	Summary of Opinions
Choice of Research	CAT1, CAT2	Remote patient monitoring systems are currently in demand and have become increasingly popular recently. The adoption of machine learning and AI techniques is the future of healthcare. The author had made a good decision for selecting the area of research.

Research Contribution	CAT1, CAT2	The project provides significant contribution to improve patient outcomes. Most available systems lack the ability to give an overall status level warning considering multiple measurement parameters therefore contributions at this level are commendable
	CAT3	Good contribution to the domain allowing the monitoring of vitals for better health outcomes
Approach taken for Development	CAT1, CAT2	Approach taken is good and effective first step for evaluating the general health status using hybrid machine learning however further improvements can done to the model and system to improve on it and allow the usage of additional parameters which require more clinical knowledge.
Analysis of Results	CAT1, CAT2	The results are promising and if further tuned and optimized would be a great add-on for real use cases improving the delivery of care as there are no similar systems commercially available.
Usability & User experience of MVP	CAT1, CAT2, CAT3	The system is very user friendly focusing on the essential data; the design of the system is simple and intuitive which allows for easy navigation ensuring those without any background knowledge would take minimal time to understand it.
Possible Improvements	CAT1	The usage of additional techniques for preprocessing and parameter tuning can be employed to balance the dataset and model further to ensure best performance. Usage of edge computing concepts to improve support in cases of disconnection from the network.
	CAT2	Integration of medical devices can be included to ensure ease of automating the capture of data for the patients. Additional parameters such as age, gender and other medical conditions can be taken in to classify the patient status level.

	CAT3	Capability of getting more disease specific health markers with the data added such as diabetes, high blood pressure etc. Further support for multiple languages, video teleconsultation functionality with the doctors can be used as future improvements to the application.

9.7 Limitations of Evaluation

The author was unable to and could not show the application to the evaluators in person as most evaluators were out of the country and were unable to meet due to time constraints, the evaluations were done based on the video demonstration and additional documentation sent to them.

9.8 Evaluation of Functional Requirements

Table 32 - Evaluation of Functional Requirements

FR ID	Priority	Use Case	Evaluation
FR1	M	UC:02	Implemented
FR2	S	UC:04	Implemented
FR3	M	UC:05	Implemented
FR4	M	UC:05	Implemented
FR5	C	UC:05	Implemented
FR6	S	UC:06	Implemented
FR7	W	UC:02	Not Considered
FR8	C	UC:03	Not Considered
FR9	C	-	Implemented
Percentage of completion of Functional Requirements = 78%			

9.9 Evaluation of Non-functional Requirements

The following table showcases a breakdown of the evaluation of the non-functional requirements of the system.

Table 33 - Evaluation of Non-Functional Requirements

NFR ID	Requirement	Priority Level	Evaluation
NFR1	Performance	Important	Implemented
NFR2	User Experience	Important	Implemented
NFR3	System Accuracy	Important	Implemented
NFR4	Security	Important	Implemented
NFR5	Scalability	Desirable	Implemented
Percentage of completion of Non-Functional Requirements = 100%			

9.10 Chapter Summary

The evaluation chapter covered the approaches taken for evaluation of the Vital:RPM system, the criteria of evaluations, the authors self-evaluation of the research project, the categories of evaluators and a thematic analysis of their feedback based on the criteria defined and the evaluations of the functional and non-functional requirements.

CHAPTER 10: CONCLUSION

10.1 Chapter Overview

This chapter concludes the research project. The achievements, utilization of knowledge, deviations from the initial scope and limitations of the research, the future enhancements that could be done and the concluding remarks of the author are discussed within this chapter.

10.2 Achievement of Research Aim & Objectives

This research aims to conduct the design, development, and evaluation of a hybrid machine-learning model to be used for acuity risk assessment and predict the future state of the health status of a patient using a vital sign time series forecasting model integrated in a remote patient monitoring system.

This research project has successfully achieved its aim to design, develop and evaluate a hybrid machine learning model for acuity health status risk assessment and is capable of predicting the future status of the patient through vital sign forecasting integrated into a remote patient monitoring application.

10.3 Utilization of Knowledge from the Course

Table 34 - Knowledge utilized from the Course

Module(s)	**Knowledge Utilized**
Web Design and Development (L4), Advanced Client-side Development (L5), Advanced Server Side (L6)	The concepts gained from these modules helped provide an understanding on the client-server architecture, UI/UX guidelines, the hierarchy of frontend and backend frameworks in order to make important decisions with selecting tools, technologies and implementation of the APIs for the project.
Programming Principles 1 & 2 (L4), Object Oriented Programming (L5)	This module paved the way for the understanding design and flow of a program to utilize these coding concepts within an application to make certain decisions during the project development

Software Development Group Project (L5)	This gave the author the initial spark to create innovative solutions. From learning how to research a topic, design, develop and test a system from start to finish.
Database Systems (L5)	This module helped to understand how database systems function in order to make queries for read/write operations within the application
Machine Learning (L5)	Helped understand the process of training a machine learning model, from data collection to preprocessing which was very useful in developing the models utilized in the project.
Applied AI (L6)	Helped understand the theoretical concepts behind several machine learning models and how the algorithms interact during development.

10.4 Use of Existing Skills

- **Full stack Development –** The author completed his internship at HTWorks where he got to work and was exposed to several full-stack development technologies.
- **Tele-healthcare –** This research project required a deeper understanding of the domain of telehealth in order find solutions that would improve the user's experience, this deeper understanding of the workflow was gained through the author's involvement in several projects within the domain throughout his internship period at HTWorks.
- **Mobile App Development –** During the author's internship, he had gained knowledge on using Flutter for mobile application development, including API integration, IOT integration, code quality and structure, and gained useful information which helped with the development of this research project.
- **UI/UX Design –** The author had specialized in UI/UX, and graphic design having freelanced over 7 years prior, this allowed him to ensure really good user experience throughout the application usage.

- **Documentation skills –** During the course timeline, the author gained a certain level of expertise being exposed to creating project documentation, especially during the development of the second year SDGP module report.

10.5 Use of New Skills

- **Machine Learning/Deep Learning –** Although the author had learnt certain concepts around machine learning during the course, it was only during the implementation of the project where he understood most of the development practices and also watched several tutorials and online courses.
- **Time Series Forecasting** – The author did not have prior experience with time series forecasting therefore gaining new exposure and experience in developing a model to perform it.
- **Hyperparameter Optimization** – Several optimization techniques were explored in order to improve the performance of the models through the usage of online articles and documentation.
- **Model Deployment & Hosting** – The author had to learn several techniques and gained knowledge of implementation of backend APIs through Flask and model hosting through Heroku.

10.6 Achievement of Learning Outcomes (LOs)

Table 35 - Learning Outcome Achievements

Learnings	Learning Outcomes
The gap that was identified has been addressed by implementing suitable techniques with justifications	LO1
The author had worked on the development of the solution. Any SLEP issues that were related to the research were identified and mitigated through planning ahead and time management.	LO6, LO7
The project requirements were gathered through reviewing literature, creating questionnaires and conducting brainstorming sessions.	LO3

Milestones were created in advance and set during the project timeline in order to successfully complete the deliverables and manage the workload.	LO2, LO5
A lot of literature papers, articles and journals were critically reviewed.	LO4
All the research undertaken was documented to the best of standards and with a good structure.	LO8

10.7 Problems and Challenges Faced

Table 36 - Mitigations done to problems and challenges faced.

Problem/Challenge	Mitigation
Significant computational resource requirements	The health status model was initially trained on a very large dataset, this required a lot of RAM usage which would often slow down the authors laptop, therefore the author utilized Kaggle Notebooks to train and test the models.
Long hours of training time	The different models developed to be compared were very time consuming and inefficient. To combat this the author decided to create several Kaggle notebook files training several models at the same time in order to produce timely results
Model File size	The hybrid health status model after exporting ended up being 1.2GB which was huge feasibility cost to host and integrate for classifications. The author decided to reduce the dataset it was trained on to 10% providing better performance to the overall system and hosted the model on Google Cloud Storage.

10.8 Deviations

Initially the goal of the author was to have the forecasting model forecast the status directly instead of vital signs, however the dataset utilized for triage classifications did not have a timestamp field which as it was removed for de identification purposes, this meant creating one based on vital signs and integrating it with the status model which gave a unique approach to it. Due to certain time and feasibility constraints, some desirable functional requirements like doctor patient chatroom, IOT integration for automated input and a side of the application for family members could not be implemented.

10.9 Limitations of the Research

- The application does not support real time monitoring of the patients' vital signs, and only provides an assessment/forecast when the user clicks on the generate button.
- The model only used 10% of the triage dataset due to development feasibility.
- The vital sign forecasting model was trained on data having an inconsistent frequency.
- The author could not implement automated input of vital signs for data collection through IoT integration due to feasibility constraints.
- The models were trained on vital signs and did not consider other information that could be useful to the assessment process.
- The models are trained on data collected from the hospital emergency departments and would work and perform best for those patients, which means other use cases could perform differently.
- Due to the heavy performance requirements and limitation in feasibility, several models could not be explored for health status assessments.

10.10 Future Enhancements

- Integration of IoT devices to automate the input and reduce the chance for errors.
- Improve model performance further through hyperparameter optimization.
- Utilize different datasets to further train the model on different outcomes, including the capability to identify different chronic diseases.
- Since the models used were only trained on the vital signs, training the model with other data such as medical records to take in consideration the patient's age, gender, weight, allergies and other medical related issues could provide a better outcome and accuracy of classification of the patient.

10.11 Achievement of the Contribution to Body of Knowledge

The author was able to effectively contribute to the problem domain and research domain during the research project's allotted schedule in order to improve system performance.

10.11.1 Problem Domain Contribution

- An automated method of classifying the patients' health status with high performance using just their vital signs.
- Forecasting a patient's health status through time series vital sign forecasting.
- A remote patient monitoring system capable of taking patient vitals, using the integrated machine learning models and return status assessments and forecasts for the doctor to take necessary measures.

10.11.2 Research Domain Contribution

- A novel model utilizing hybrid machine learning to classify a patient's health status on using thier vital signs.
- A hybrid framework approach taken to forecast a patient's health status through time series forecasting of their vital signs and using it into the hybrid health status classification model to give a forecasted status.

10.12 Implementation Code

All the research conducted including the code used during the development of the implementation were conducted on GitHub and are available in the link below for convenience:

- Vital:RPM: Implementation Code & Research Documentation repository - ***https://github.com/hammvdh/vitalrpm***

10.13 Concluding Remarks

This would conclude the research undertaken to show the complete design, development and evaluation of health status assessments using hybrid machine learning in remote patient monitoring systems. This chapter determined whether the author accomplished all objectives, the existing skills utilized and the new skills gained throughout the project timeline. The author also mentions any deviations taken, the limitations and future enhancements of the system that can be done and showcases the contributions made to the bodies of knowledge with the author's findings.

REFERENCES

Johnson, A., Bulgarelli, L., Pollard, T., Horng, S., Celi, L. A., and Mark, R. (2022) 'MIMIC-IV' (version 2.0), PhysioNet. Available at: https://doi.org/10.13026/7vcr-e114.

Saunders, M., Lewis, P. and Thornhill, A. (2003), 'Research methods for business students, Essex: Prentice Hall: Financial Times. Available from https://toc.library.ethz.ch/objects/pdf_ead50/4/E57_7072090_TB-Index_005013522.pdfAditya, T.R. et al. (2020). Real Time Patient Activity Monitoring and Alert System. *Proceedings of the International Conference on Electronics and Sustainable Communication Systems, ICESC 2020*, 708–712. Available from https://doi.org/10.1109/ICESC48915.2020.9155602 [Accessed 1 February 2023].

Albahri, O.S. et al. (2018). Systematic Review of Real-time Remote Health Monitoring System in Triage and Priority-Based Sensor Technology: Taxonomy, Open Challenges, Motivation and Recommendations. *Journal of Medical Systems*, 42 (5). Available from https://doi.org/10.1007/s10916-018-0943-4.

Al-Turjman, F., Nawaz, M.H. and Ulusar, U.D. (2020). Intelligence in the Internet of Medical Things era: A systematic review of current and future trends. *Computer Communications*, 150, 644–660. Available from https://doi.org/10.1016/J.COMCOM.2019.12.030 [Accessed 3 May 2023].

Anitha, P. et al. (2022). Virtual Telemedicine System for Remote Health Monitoring of Patients. *2022 4th International Conference on Cognitive Computing and Information Processing, CCIP 2022*. Available from https://doi.org/10.1109/CCIP57447.2022.10058637 [Accessed 1 May 2023].

Bellini, V. et al. (2022). Artificial intelligence and telemedicine in anesthesia: potential and problems. *Minerva anestesiologica*, 88 (9), 729–734. Available from https://doi.org/10.23736/S0375-9393.21.16241-8 [Accessed 4 February 2023].

Bhavani, T. et al. (2022). Stress Classification and Vital Signs Forecasting for IoT-Health Monitoring. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. Available from https://doi.org/10.1109/TCBB.2022.3196151 [Accessed 4 February 2023].

Chang, Daniel, Chang, David and Pourhomayoun, M. (2019). Risk prediction of critical vital signs for ICU patients using recurrent neural network. *Proceedings - 6th Annual Conference on Computational Science and Computational Intelligence, CSCI 2019*. 1 December 2019. Institute of Electrical and Electronics Engineers Inc., 1003–1006. Available from https://doi.org/10.1109/CSCI49370.2019.00191.

Ciaburro, G. and Iannace, G. (2021). Machine Learning-Based Algorithms to Knowledge Extraction from Time Series Data: A Review. *Data*, 6 (6). Available from https://doi.org/10.3390/DATA6060055 [Accessed 8 January 2023].

Dhinakaran, M. et al. (2022). A System of Remote Patients' Monitoring and Alerting Using the Machine Learning Technique. Available from https://doi.org/10.1155/2022/6274092 [Accessed 4 October 2022].

Drerup, B. et al. (2021). Reduced No-Show Rates and Sustained Patient Satisfaction of Telehealth During the COVID-19 Pandemic. *Telemedicine journal and e-health : the official journal of the American Telemedicine Association*, 27 (12), 1409–1415. Available from https://doi.org/10.1089/TMJ.2021.0002 [Accessed 8 January 2023].

Elliott, M. and Endacott, R. (2022). The clinical neglect of vital signs' assessment: an emerging patient safety issue? *Contemporary nurse*. Available from https://doi.org/10.1080/10376178.2022.2109494 [Accessed 2 November 2022].

El-Rashidy, N. et al. (2021). Mobile health in remote patient monitoring for chronic diseases: Principles, trends, and challenges. *Diagnostics*, 11 (4). Available from https://doi.org/10.3390/diagnostics11040607.

Elsayed, S. et al. (2021). Do We Really Need Deep Learning Models for Time Series Forecasting? Available from http://arxiv.org/abs/2101.02118 [Accessed 3 May 2023].

Gao, F. et al. (2022). Predictive Models for Emergency Department Triage using Machine Learning: A Review. *Obstetrics and Gynecology Research*, 05 (02). Available from https://doi.org/10.26502/OGR082 [Accessed 6 January 2023].

Godi, B. et al. (2020). E-Healthcare Monitoring System using IoT with Machine Learning Approaches. *2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)*. Available from https://doi.org/10.1109/ICCSEA49143.2020.9132937 [Accessed 2 May 2023].

Gontarska, K. et al. (2021). Predicting Medical Interventions from Vital Parameters: Towards a Decision Support System for Remote Patient Monitoring. Available from http://arxiv.org/abs/2104.10085.

Ishtiaque, F. et al. (2021). IoT-Based Low-cost Remote Patient Monitoring and Management system with Deep Learning-Based Arrhythmia and Pneumonia detection. *2021 IEEE 4th International Conference on Computing, Power and Communication Technologies, GUCON 2021*. Available from https://doi.org/10.1109/GUCON50781.2021.9573620 [Accessed 5 January 2023].

Jayatilake, S.M.D.A.C. and Ganegoda, G.U. (2021). Involvement of Machine Learning Tools in Healthcare Decision Making. *Journal of Healthcare Engineering*, 2021. Available from https://doi.org/10.1155/2021/6679512.

Kadum, S.Y. et al. (2023). Machine learning-based telemedicine framework to prioritize remote patients with multi-chronic diseases for emergency healthcare services. *Network Modeling Analysis in Health Informatics and Bioinformatics*, 12 (1). Available from https://doi.org/10.1007/s13721-022-00407-w.

Kandpal, P. et al. (2020). Contextual Chatbot for Healthcare Purposes (using Deep Learning). *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, 625–634. Available from https://doi.org/10.1109/WORLDS450073.2020.9210351 [Accessed 3 May 2023].

Kuthiala, R., Telivala, B.P. and Karawadia, T. (2022). Remote patient monitoring (RPM) through smartphone application in community oncology practice. *Journal of Clinical Oncology*, 40 (16_suppl), e18828–e18828. Available from https://doi.org/10.1200/JCO.2022.40.16_SUPPL.E18828 [Accessed 3 January 2023].

Lata Sahu, M. et al. (2021). Cloud-Based Remote Patient Monitoring System with Abnormality Detection and Alert Notification. *Mobile Networks and Applications*, 1, 16. Available from https://doi.org/10.1007/s11036-022-01960-4 [Accessed 4 October 2022].

Liu, S., Yao, J. and Motani, M. (2019). Early Prediction of Vital Signs Using Generative Boosting via LSTM Networks. *Proceedings - 2019 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2019*, 437–444. Available from https://doi.org/10.1109/BIBM47256.2019.8983313 [Accessed 2 May 2023].

Malasinghe, L.P., Ramzan, N. and Dahal, K. (2019). Remote patient monitoring: a comprehensive study. *Journal of Ambient Intelligence and Humanized Computing*, 10 (1), 57–76. Available from https://doi.org/10.1007/S12652-017-0598-X/TABLES/6 [Accessed 2 November 2022].

Mia, M.M.H. et al. (2021). An Internet of Things Application on Continuous Remote Patient Monitoring and Diagnosis. *BioSMART 2021 - Proceedings: 4th International Conference on Bio-Engineering for Smart Technologies*. Available from https://doi.org/10.1109/BIOSMART54244.2021.9677715 [Accessed 1 January 2023].

Nallakaruppan, M.K. and Kumaran, U.S. (2020). Hybrid machine learning model for healthcare monitoring systems. *International Journal of Internet Technology and Secured Transactions*, 10 (5), 538–551. Available from https://doi.org/10.1504/IJITST.2020.109532 [Accessed 1 November 2022].

Napi, N.M. et al. (2019). Medical emergency triage and patient prioritisation in a telemedicine environment: a systematic review. *Health and Technology*, 9 (5), 679–700. Available from https://doi.org/10.1007/S12553-019-00357-W [Accessed 3 February 2023].

Olivencia, S.B. et al. (2022). Integration of Remote Patient Monitoring Systems into Physicians Work in Underserved Communities: Survey of Healthcare Provider Perspectives. *ArXiv*. Available from https://doi.org/10.48550/ARXIV.2207.01489 [Accessed 3 May 2023].

Phetrittikun, R. et al. (2021). Temporal Fusion Transformer for forecasting vital sign trajectories in intensive care patients. *BMEiCON 2021 - 13th Biomedical Engineering International Conference*. Available from https://doi.org/10.1109/BMEICON53485.2021.9745215 [Accessed 1 January 2023].

Phungoen, P. et al. (2020). Emergency Severity Index as a predictor of in-hospital mortality in suspected sepsis patients in the emergency department. *The American journal of emergency medicine*, 38 (9), 1854–1859. Available from https://doi.org/10.1016/J.AJEM.2020.06.005 [Accessed 3 May 2023].

Protić, D. and Stanković, M. (2020). Anomaly-Based Intrusion Detection: Feature Selection and Normalization Influence to the Machine Learning Models Accuracy European Journal of Formal Sciences and Engineering. *European Journal of Formal Sciences and Engineering*, 3 (1).

Salehi, S. et al. (2020). Assessment of remote patient monitoring (RPM) systems for patients with type 2 diabetes: a systematic review and meta-analysis. *Journal of Diabetes and Metabolic Disorders*, 19 (1), 115–127. Available from https://doi.org/10.1007/S40200-019-00482-3/TABLES/4 [Accessed 1 November 2022].

Salman, O.H. et al. (2021). A review on utilizing machine learning technology in the fields of electronic emergency triage and patient priority systems in telemedicine: Coherent taxonomy, motivations, open research challenges and recommendations for intelligent future work. *Computer Methods and Programs in Biomedicine*, 209. Available from https://doi.org/10.1016/j.cmpb.2021.106357.

Sarierao, B.S. and Prakasarao, A. (2018). Smart Healthcare Monitoring System Using MQTT Protocol. *2018 3rd International Conference for Convergence in Technology (I2CT)*. Available from https://doi.org/10.1109/I2CT.2018.8529764 [Accessed 3 May 2023].

Souri, A. et al. (2020). A new machine learning-based healthcare monitoring model for student’s condition diagnosis in Internet of Things environment. *Soft Computing*, 24 (22), 17111–17121. Available from https://doi.org/10.1007/S00500-020-05003-6/FIGURES/5 [Accessed 2 May 2023].

Srinivasulu, A. and Gupta, A.K. (2022). A Deep Convolutional Neural Network Framework based Model for IoT based Healthcare Monitoring System.

Tabassum, S. et al. (2020). A data enhancement approach to improve machine learning performance for predicting health status using remote healthcare data. *2020 2nd International Conference on Advanced Information and Communication Technology, ICAICT 2020*, 308–312. Available from https://doi.org/10.1109/ICAICT51780.2020.9333506 [Accessed 19 September 2022].

Tan, L.T. et al. (2022). Remote Patient Monitoring System. *2022 IEEE 5th International Symposium in Robotics and Manufacturing Automation, ROMA 2022*. Available from https://doi.org/10.1109/ROMA55875.2022.9915658 [Accessed 19 October 2022]

Timpano, F. et al. (2013). Tele-Health and neurology: what is possible? *Neurological Sciences*, 34 (12), 2263–2270. Available from https://doi.org/10.1007/S10072-012-1285-5 [Accessed 3 May 2023].

Verma, P. et al. (2021). Comparison of Time-Series Forecasting Models. *2021 International Conference on Intelligent Technologies, CONIT 2021*. Available from https://doi.org/10.1109/CONIT51480.2021.9498451 [Accessed October 2022].

Vinutha, D.C., Kavyashree and Raju, G.T. (2022). Machine Learning–Assisted Remote Patient Monitoring with Data Analytics. *Tele-Healthcare*, 1–26. Available from https://doi.org/10.1002/9781119841937.CH1 [Accessed 11 October 2022].

Wang, H. et al. (2021). Research on Model for Vital Signs Estimation Based on Convolutional Neural Network. *Chinese Control Conference, CCC*, 2021-July, 8550–8553. Available from https://doi.org/10.23919/CCC52363.2021.9549912 [Accessed 2 May 2023].

World Health Organization. (2012). Preface Overview Humanity's Aging Living Longer New Disease Patterns Longer Lives and Disability New Data on Aging and Health Assessing the Cost of Aging and Health Care Changing Role of the Family Suggested Resources. *Global Health and Aging*. Available from https://www.nia.nih.gov/sites/default/files/2017-06/global_health_aging.pdf [Accessed 10 February 2023].

Xuan, A. et al. (2022). A comprehensive evaluation of statistical, machine learning and deep learning models for time series prediction. *2022 7th International Conference on Data Science and Machine Learning Applications (CDMA)*, 55–60. Available from https://doi.org/10.1109/CDMA54072.2022.00014 [Accessed 10 October 2022].

Zachariasse, J.M. et al. (2019). Performance of triage systems in emergency care: a systematic review and meta-analysis. *BMJ Open*, 9 (5), e026471. Available from https://doi.org/10.1136/BMJOPEN-2018-026471 [Accessed 18 October 2022].

APPENDIX A - INTRODUCTION

A.1. Prototype Workflow Diagram

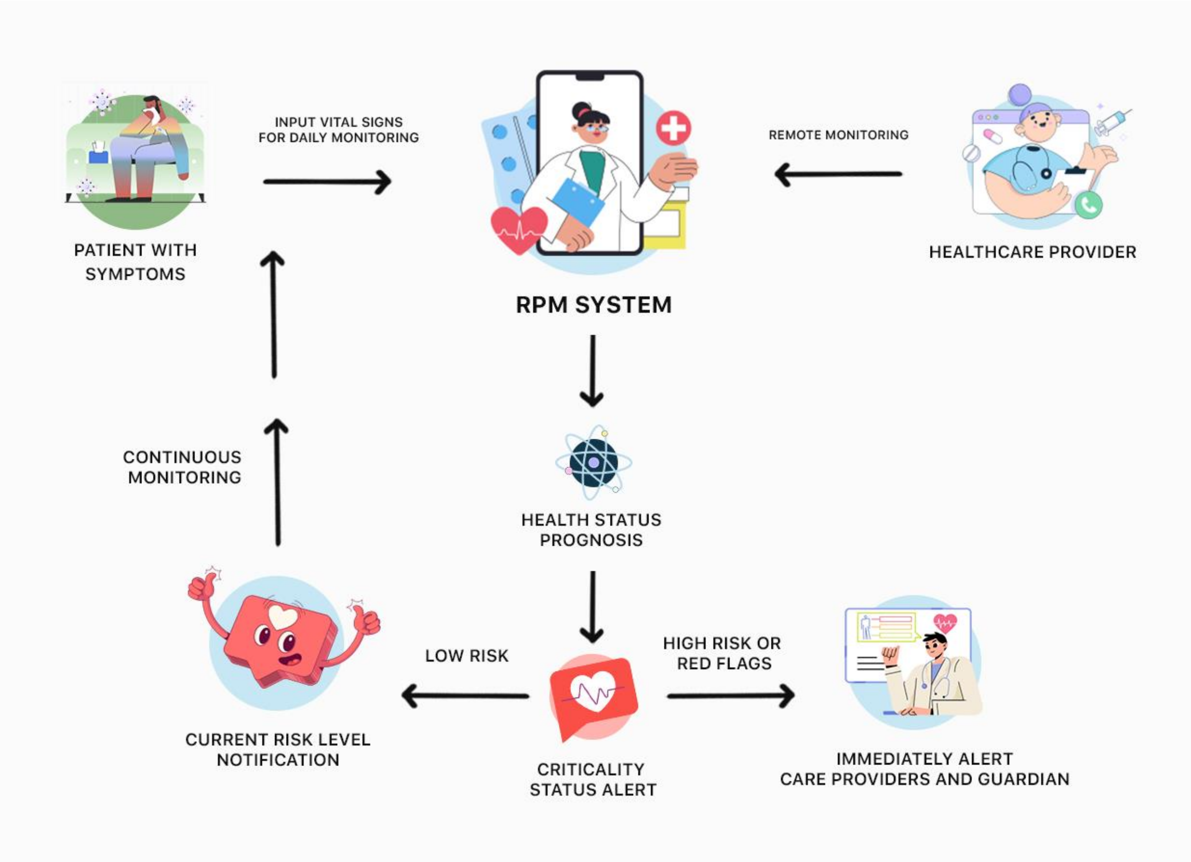


Figure 38 - Prototype Workflow Diagram (Self-Composed)

A.2. Project Scope

Based on the available timeframe for the research project, the scope is as follows:

A.2.1. In-scope

- Implementation of a hybrid machine-learning model capable of diagnosis and prognosis.
- Implementation of a remote patient monitoring system with a graphical user interface capable of taking patient input remotely.
- Integration of the proposed hybrid model with the RPM system to inform relevant healthcare providers of the patient's current health status and alerts when a prognosis is critical.
- Evaluation & comparison of the proposed system against existing solutions

A.2.2. Out-scope

- Integration of wireless IoT medical devices for automated input without relying on manual input from patients leading to improved accuracy of data collected.
- Ability to consider other additional external medical factors during prediction, such as previous health records, allergies, etc.
- Integration and support of the RPM system in the emergency department of hospital to monitor patients admitted.

APPENDIX B - EVALUATIONS

B.1. Evaluator Feedback

The below table showcases the quoted evaluations given by the experts and end users.

Evaluator	Feedback
Imran Haroon Health IT Consultant at Security Forces Hospital Riyadh	“In this digital era, the choice and novelty of this research is pretty popular in the western world and spreading across other regions as well. RPM is becoming mandatory specially with insurance companies to ensure Acute care patients are monitored and treated before they fall sick. Contributions by this research are Excellent at this stage as a student. The approach taken is fine but have to consider other stakeholders and make separate dashboard and access levels. For example, multiple care givers and relatives of the patient also access and monitor. The interface was nicely done. As mentioned above, this could be enhanced further to accommodate other participants with required screens and rules such as consent etc. for wider usage. Overall, its excellent. Since healthcare outcomes are very complicated, there is room for more integration and rules engine as we develop complicated system. Enabling video calls etc. with the same system is another enhancement to consider. ”
Mohamed Imtiaz Senior Technical Specialist, InterSystems	“The author absolutely selected a good field to research on and tapped into the future of telemedicine. Being able to remotely and proactively alert patients/physician of patients health condition is a dominant feature for any healthcare provider to have. The author has obtained datasets from known sources and validated results to near accuracy. This is the most needed outcome in any ML project, especially when it comes to patient safety. Application user interfaces look simple and easy to use. Considering the non-IT background users, time spent on training will be minimal. Most features/processing is happening in the background.”

Walid Bousmaha Software Project Manager & Senior Innovation Consultant at Hadi Clinic Kuwait	“This research project makes several contributions to the field of remote patient monitoring systems and healthcare. The authors have also performed extensive experiments and evaluations to demonstrate the effectiveness of their approach. They have used a variety of metrics to evaluate the performance of their model and have compared their results to other state-of-the-art models in the field. The results show that their approach outperforms other models in terms of accuracy and efficiency. Overall, the approach taken by the authors for the development of the research project appears to be sound and effective, and the results are promising. It seems that the user interface of the system is good. Overall, the project seems to be well-designed and implemented, with a clear focus on providing a useful tool. One potential improvement could be to expand the range of languages supported by the system, to make it more accessible to users who write or read in languages other than English. Additionally, it may be worth exploring ways to further streamline the user interface and make it more intuitive, to ensure a smooth and efficient user experience. Finally, continued development and refinement of the machine learning models used in the system could lead to even more accurate and effective capabilities. ”
Malik Ahmed Senior Software Engineer.	“The choice of the project ideally complies with this era due to technological evolution and medical attention to the individual patient. I believe you have covered the novelty part of the project. based on the patient past vitals you are training the model using hybrid ML model. According to what I understand, there is no similar application commercially available in the market. If you can fine-tune further and increase the accuracy, it will be great add-on to the society. Since your data is imbalanced try to make it balanced using MICE Imputation, Multivariate Visualization to eliminate the outliners, Correlation Matrix, SMOTE, precision, F1-Score, recall,

	oversampling, hyperparameter tuning and K-Folds. Even you can use a dummy data model, classifier is one that does not provide any new insights on the data but rather classifies the data that is provided by relying solely on straightforward principles. In addition to that you can use ensemble model to add your trained model to improve accuracy. As far as I can see, your software is straightforward and simple to use, so even those with little experience with mobile devices could use it after a demonstration.”
Ajwad Cader CEO at HT Works	“Remote Patient Monitoring is a growing field and bringing Machine Learning and AI into this is the future. There is still many possible use cases in this area to be explored and I believe this is a good attempt to focus on this area. This is a step in the right direction, and I believe it's a fair contribution at this level. Actual usage of these in the healthcare field would require more research from clinical sides. This is a good first step and the approach seems to be appropriate for this level. UI/UX of the solution is good and the developer seems to have a good idea of UI/UX and good taste for colors to keep the UI pleasing to the eye as well. For a undergraduate level project this is a very interesting subject and commendable achievement. Given this involves clinical components, real-life projects of this nature would require clinical domain expertise and regulatory involvement.”
Mohamed Ragab Product Specialist at InterSystems	“I like the idea of trying to implement a system similar to EWS (Early Warning Score) for patients not admitted to hospitals. Most patient monitoring systems can give warning based on a single healthcare measurement, but they lack the ability of combining different parameters that can be normal or low risk by itself but collectively can indicate a patient health issue. Adding ML models to get patient's health trend is an interesting approach that can help with a more proactive approach to address the underlying health condition. The application is easy

	to use. What was not clear is how the trends are generated and how the care providers can get alerted about patient case deterioration. More focus on completing the story and how a full workflow can be achieved from starting the patient on this program and then how the care provider can be prompted in case of deterioration. ”
Riyas Razik CIO at HT Works	“The choice and novelty of research is good because it is a trending field, the adoption of AI use in current RPM solutions are minimal. The approach taken looks good to me. The normal ranges can be different by region, age and sex. So it would be better to have provision to have multiple set of normal ranges. The UI is simple and intuitive. In real world scenario, the schedule to collect the readings might be defined by the doctor as this solution will also involve a care provider and a facility. For potential improvements, the questions section during adding measurements can be dynamic and link to vital items, as different items might have a different set of questions.”
Shazni Shiraz Software Engineer at Ascentic	“it's excellent to focus on the Remote Patient Monitoring System (RPMS) in the healthcare domain. With the shift towards virtual healthcare services, RPMS has the potential to improve patient outcomes and reduce healthcare costs by enabling remote monitoring and management of patients' health. As a relatively new area of research, there is much to be discovered about the implementation and optimization of RPMS technologies. Thus, the researcher's project could contribute significantly to the development and implementation of new RPMS systems that could revolutionize the healthcare industry so the choice to focus on RPMS is timely and relevant. The work could have significant implications for the healthcare industry, improving patient outcomes, reducing costs, and enhancing the overall patient experience. Based on the short demo of the application, I found the interface to be user-friendly and easy to navigate. The minimalist design and focus on

	essential data made it easy to understand and interpret the user's health status. The features and functionality of the system, including the ability to capture and analyze vital information and notify medical representatives in case of concerning changes in the user's health status, demonstrate the potential applicability of the system in real-world scenarios. I also believe that this application could be particularly useful for individuals with chronic conditions that require regular monitoring of their vital signs. The ability to capture and analyze vital information on a regular basis could allow for earlier detection of potential health issues and timely medical intervention when needed. Overall, the project sounds promising, and the benefits of the application are significant in improving health outcomes and reducing healthcare costs. One potential improvement that could be considered for future implementation is the integration of smart devices to automate the process of capturing vital information. This would not only reduce the burden on users but also enhance the accuracy and consistency of the captured data. It may be useful to consider the privacy and security concerns associated with the storage and sharing of sensitive health data. Ensuring that appropriate measures are taken to protect the privacy and security of the user's data should be a top priority when developing and implementing health monitoring applications. Overall, the project has the potential to improve patient outcomes, reduce healthcare costs, and enhance the overall patient experience. With ongoing development and improvements, it could become an essential tool for managing personal health and wellbeing.”
Hassan Anver CTO at HT Works	“RPM is an emerging area in healthcare and combining AI to solve a problem is current and relevant. The Hybrid module introduced in this project will improve delivery of care to patients in a remote setting. Use of HML model is innovative in this case but should cater for inherent bias in AI models. The

	UI/UX of the application is very appealing and intuitive. It will be very user friendly in a real-world setting. Should incorporate Edge computing concepts to improve latency and support in disconnected modes of operation”
Anonymous, Care Provider	“Choice and novelty of research is good, Remote health care control solutions can certainly help facilitate patients to access health care consultation more easily and conveniently. its the future of health and wellness management to track and maintain important health markers and get preventive and proactively consultation . A Robust solution that can track all the right markets and accurately predict health status and risk will be highly beneficial. I think the project is a good endeavor into an important area of value added service delivery within health care sector. And using modern app based technology to update and track and benchmark health parameters will facilitate better health outcomes more proactively. Its good starting point of evaluating general health status and proactive generating reports and projection based on historical input sequences. It will also help patients and practitioners to keep a track of heath status over a period time enabling them to better manage it. The design looks very modern and easy to understand and input sequences look easy to input and manage. Might be useful to see a time series view to understand how things are changing over time. I believe what is important is to be able conveniently input or getting automatic device based input of the respective health markers will be important. Also to be a little more disease specific health markers (Heart Health, Diebetes, HBP etc) as patients are often focused on specific diseases rather than general health. It will be also very useful to benchmark these health markers against established datasets to identify potential health issue patterns and predictive analytics of future health status, life expectancy etc.... lastly remote link to potential health testing laboratories where different health reports (blood

	test, scans etc) can be directly uploaded by the laboratory with reference to a patients will be useful to get the doctors to have a better reading on the patients health status beyond the general markers when required...”
Riyas Nizar Director - Product Development and Implementations at HT Works	“Combining AI and Machine learning to health care is a very high priority requirement. The patient care has to move from the reactive treatment to proactively predict the outcomes and take action. this will save lives as well reduce Emergency room and other ICU admission costs immensely. In that aspect this research is a very good starting point to help future healthcare industry and patient care. This is a very good initiation helping the healthcare industry in aligning it with the technology advancement. More investment and research is this area will definitely help positive healthcare outcomes and improve the patient care. It looks like a very simple and easy to use application. It can be enhanced specially to make it more usable by capturing data through connected devices. It will reduce data entry errors as well as the exact reading times. With the short product I didn't see other details when registering a patient like his Age, Gender and Clinical condition. These details are important when doing Assessments and future forecasting. Again this project is just a small step in addressing the greater changes needed in healthcare industry incorporating technology to improve patient care.”