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

Mobile Based Patient Navigation System for Rehabilitation of Paralyzed Patients

Pua Lee Ling and Dayang N. A. Jawawi
School of Computing, Faculty of Engineering
Universiti Teknologi Malaysia
81310 Johor Bahru, Johor, Malaysia
pualeeling@gmail.com, dayang@utm.my

***Abstract*—Paralyzed patients are people who unable or have difficulty in controlling a muscle or group of muscles in a part of their body. Hospitals or healthcare centers will provide rehabilitation treatments to these patients for recovery of their physical, sensory or mental ability. The current workflow of Rehabilitation Department in Hospital Sultanah Aminah (HSA) is in manual ways such as recording the information of patients, appointments, and rehabilitation assessment of paralyzed patients in paper documents. It is quite hard to ensure the correctness of information because human errors may occur during the recording process. Therefore, a mobile based Patient Navigation System for rehabilitation of paralyzed patients was proposed. In this system, the system admin can register users and manage users. This system allows the community (Pharmacist, Healthcare Center etc.) and nurse to register new patients. Besides, this system allows the nurse to set the appointment of the patient in specific date and time, manage appointment requests made by patients and make vital check-up of patient. Furthermore, the doctor can make general assessment and view notification on the new appointment set, appointment information, and assessment report. For the patients, they can manage profile, make appointment request, and view notification on the new appointment set, appointment information, and appointment request status. While for the physiotherapist and occupational therapist, they can make stroke assessment and progress assessment respectively. Agile-Kanban methodology was chosen to develop this system. During the development, the Ionic Vue Framework and programming languages such as HTML, CSS, and PHP were used. Lastly, the system was tested using Black Box Testing and User Acceptance Testing. For Black Box Testing, the results show that all functions of the system have passed. For User Acceptance Testing, five staffs work at IT Unit of HSA were invited to test the system. Most of the testers agreed that the system is user-friendly and is reliable to be implemented in Rehabilitation Department of HSA.**

Keywords- Patient Navigation System, Rehabilitation, Paralyzed Patients, Mobile Application

I. INTRODUCTION

Paralyzed patients are people who get paralysis. Paralysis causes these patients unable or have difficulty controlling a muscle or group of muscles in a part of their body. It occurs due to disorders of the central nervous system [1]. Paralyzed patients usually need rehabilitation treatment to recover from their paralysis. Hence, hospitals or healthcare centers will provide rehabilitation treatments to these patients for recovery of their physical, sensory or mental ability so they can live and work normally and independently.

The Patient Navigation Program (PNP) is a programme that aims to save patients' lives from cancer by reducing the obstacles of timely care between the discovery of disease and the resolution of the disease through diagnosis and treatment [2]. It is the first programme that implements the concept of patient navigation and was introduced by Dr. Harold Freeman in Harlem, New York in 1990. Although the patient navigation initially mainly focused on cancer patients, it is now also used for patients who suffer from other diseases and also paralyzed patients who undergo rehabilitation. The Patient Navigation System (PNS) which implements the concept of patient navigation was launched to help track and monitor the rehabilitation process of the paralyzed patients by recording patients' diagnosis and treatment details.

Nowadays, some hospitals and healthcare centers in Malaysia still use manual methods to keep and manage patients information [3]. For example, the current workflow of Rehabilitation Unit in Hospital Sultanah Aminah is in manual ways such as recording the information of patients, appointments, vital check-ups, rehabilitation assessment of paralyzed patients in paper documents. It is quite hard to ensure the correctness of information because human errors may occur during the recording process.

The aim of this project is to develop a mobile based Patient Navigation System for rehabilitation of paralyzed patients at Hospital Sultanah Aminah to manage patients' information,

manage appointments, and monitor the rehabilitation progress of the paralyzed patients in a computerized way.

II. Related Works

This section discusses on the current system analysis for the existing systems that applied the concept of patient navigation system. It will identify any problems or weaknesses of the systems. It is important as it helps to make sure that the identified weaknesses will not be included in the mobile based PNS.

A. Allevio Patient Navigator

"Allevio Patient Navigator" [4] is a mobile application that helps the patients of Allevio Pain Management clinic to be informed, connected and in a better control of their care. The patients can use this application to find all the information related to their appointments and get reminders on their mobile devices. "Allevio Patient Navigator" allows the patients to view the appointment only set by the clinic. While the patients are not able to make request for an appointment by using this application. Therefore, the patient cannot make an appointment with the doctor when needed, which brings inconvenience to the patient. Besides, the application is mainly designed to be used by patients, there is no functions available for the doctors and nurses.

B. Navigation Tracker

"Navigation Tracker" [5] is a web-based application that runs on PC, Mac or iPad. The users of this application are care managers and patient advocates. This system provides some features to the care managers and patient advocates, such as managing patients' information, recording service records, scheduling and tracking appointments, viewing email from patients and many more. "Navigation Tracker" is mainly designed for care managers and patient advocates. The system functions are only suitable for care managers and patient advocates. For example, the care managers and patient advocates can set the appointment, view the information of the appointment, and reschedule the appointment by using the application. While patients are not allowed to use this application. Hence, they cannot direct make request for an appointment when needed by using this application. If the patients want to book an appointment with the care managers or patient advocates, they can only do so by manually informing, emailing, or calling the care managers or patient advocates.

C. MaximEye

"MaximEyes" [6] is a web-based application that uses the idea of a patient navigation system. Patient encounter is the user of this application, and the patients are not allowed to use this application. This application provides some features for the patient encounter, such as registering a new patient, finding a patient, registering an encounter, viewing encounter information, and others. However, the user interface of the application is not well designed and not user-friendly, which may cause the user who is not familiar with the application to fail to use certain functions of the application.

D. Comparison between Existing System

There are some differences between the existing systems used by healthcare and the mobile based PNS. Table I shows the comparison between "Allevio Patient Navigator", "Navigation Tracker", "MaximEyes" and mobile based PNS based on the features or aspects.

TABLE I. Comparison of Features and Aspects

Features / Aspects	**Allevio Patient Navigator**	**Navigation Tracker**	**MaximEyes**	**Mobile Based PNS**
Account	Free	Free	Free	Free
Platform	iOS	iOS, web browser	Web browser	Android
User	Patient	Care managers and patient advocates	Patient encounter	Patient, doctor, nurse, community, physiotherapist, occupational therapist, system admin
Patient Type	Patient who suffers chronic pain conditions	General patient	Patient who needs vision care	Paralyzed patient
Patient Update Profile	Yes	Not applicable	Not applicable	Yes
Manage Patient	None	Yes	Yes	Yes
Set Appointment	None	Yes	Yes	Yes
View Appointment	Yes	Yes	Yes	Yes
Update Appointment	Yes	Yes	Yes	Yes
Make Appointment Request	None	Not applicable	Not applicable	Yes
Get Notification	Yes	None	None	Yes
Make Assessment	Not applicable	None	None	Yes

III. Methodology

The methodology that will be used in the development of mobile-based PNS is the Agile model. There are several types of Agile models, such as Scrum, Kanban and others. Kanban is a software development methodology that focuses on delivering functionality in a timely manner and managing the amount of work in progress (WIP). It provides an intuitive way of project representation so that the developer can easily track the current status of the project. The reason that Kanban was chosen is because it is suitable for small projects [7]. Mobile based PNS is considered a small project and will be developed by only one person. Besides, Kanban uses deadline as the measurement of productivity [8]. Therefore, the tasks during the development of mobile based PNS can be divided into multiple iterations and

each iteration can be set with a deadline. It is important because the development progress of mobile based PNS can be easily tracked and to make sure the mobile based PNS will be developed on time. Figure 1 shows the graphical illustration of the Kanban methodology.

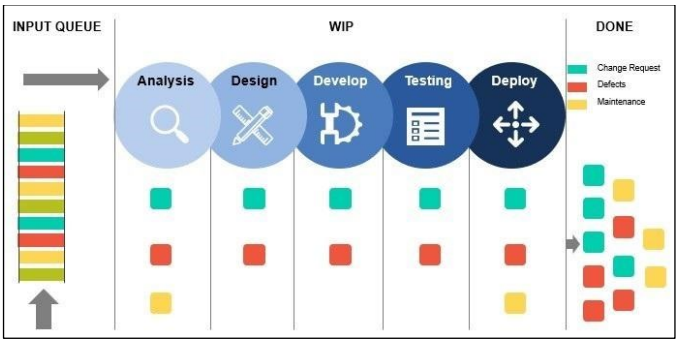

Figure 1. Kanban Methodology

The development of mobile based PNS will be divided into several iterations by using the Kanban board. Each iteration will be set with a deadline. Different iterations will focus on different modules in mobile based PNS. Then, the software development process will be divided into six phases, which are requirement analysis phase, design phase, development phase, testing phase, deployment phase, and review phase. The review phase will occur when the item is put in the done section.

E. Requirement Analysis Phase

In this phase, the requirements of mobile based PNS are collected and gathered from the sources and the stakeholder. The techniques that will be used to gather requirements are document analysis and interviews. Document analysis will be done on the thesis with the title "Patient Navigation System for Rehabilitation of Paralysis Patients" [9] done by Muhammad Syafiq to get the relevant information. Besides, an interview with the stakeholder will be conducted to obtain detailed information about the workflow in Hospital Sultanah Aminah and to know about the stakeholder's requirements. Another interview with Muhammad Syafiq will also be carried out to get information on the basic functions of the mobile based PNS he developed and get suggestions for improvement from him. After the interview, the user stories of PNS will be elicited based on the requirements gathered to obtain a clear idea of the system functions of mobile based PNS. The user stories will be put into the Backlog column and then be divided into multiple iterations using the Kanban board. Besides, the functional and non-functional requirements of the mobile based PNS will be identified in this phase. This is to ensure the PNS will be developed according to the stakeholder's expectation and needs.

F. Design Phase

In this phase, the system internal and external structure of mobile based PNS will be designed and then reported in Software Requirements Specification (SRS) and Software Requirements Specification (SDD) reports. The chosen architectural style to develop mobile based PNS is Model-View-Controller. The details of the functional and non-functional requirements together with the use case diagram, use case description, architecture diagram, domain diagram, activity diagrams for each use case, sequence diagram for each use case, database design and others will be included in these reports. Since the requirements can change from time to time, the content of SRS and SDD reports will also change from time to time.

G. Development Phase

In this phase, mobile based PNS will start to be developed according to the Kanban board for each iteration. Each iteration will focus on one module of PNS. A Kanban board will be created and divided into three parts, which are to-do, in progress, and done. Each functionality of the module will need to be placed in done section before the deadline to make sure the development of mobile based PNS can be done on time. The coding process will be carried out on code editor, Visual Studio Code. Besides, an Android emulator will be run to see the output of the source code. MySQL will be used as the database management system for mobile based PNS. Each iteration will need to produce a working and usable PNS. However, it is not the final version since it only includes certain functionality. The other functionality will be added in the future iteration. The final version of mobile based PNS will be produced in the last iteration.

H. Testing Phase

In this phase, the developer will carry out testing activities such as Black Box Testing on the developed PNS. The purpose of this phase is to ensure the system architecture of the developed PNS follows the proposed system architecture and to ensure the developed PNS is free of defects or bugs. The testing activities will follow the test cases designed in the Software Testing Documentatio (STD). If there are any defects or bugs detected in the developed PNS during the testing, the detected defects or bugs will be fixed to ensure the usability and quality of the developed PNS. Besides, User Acceptance Testing will be conducted to validate the functionality of the developed PNS and obtain the user experience of the stakeholder.

I. Deployment Phase

In this phase, each version of the developed PNS will be released and presented to the stakeholder. The functionality of each released PNS will be incremented for each iteration. When all the iterations are completed, the final version of mobile based PNS with all the functionality will be released. It will then be delivered to the stakeholder and it is ready to be used by the users, such as paralyzed patients, doctors, nurses, physiotherapists, and occupational therapists at Hospital Sultanah Aminah.

J. Review Phase

This is the last phase of each iteration. All the problems that happened during the iteration will be reviewed in this phase. The developer needs to have a plan to solve the problems before moving to the next iteration. For example, problems such as the functionality of view appointment not working properly may occur during the iteration. Hence, this problem needs to be reviewed before moving to the next iteration to ensure the

quality of the developed PNS and reduce the risk of system failure for the developed PNS.

IV. REQUIREMENT ANALYSIS AND DESIGN

Requirement analysis and design is the crucial phase in software development. The phase starts with analyzing the functional and non-functional requirements of the system. This process is to ensure the developer understands the activities or processes that should be performed in the system and determines the functionality that should be developed. After that, the developer can continue to design the system according to the collected requirements.

Use case diagram is a behavioral diagram in Unified Modeling Language (UML) diagram that shows the interactions of users with a system. It models the functionality of a system using actors and use cases. By using the use case diagram, the relationship between the actors and the use cases can be clearly understood. Figure 2 shows the use case diagram of mobile based PNS.

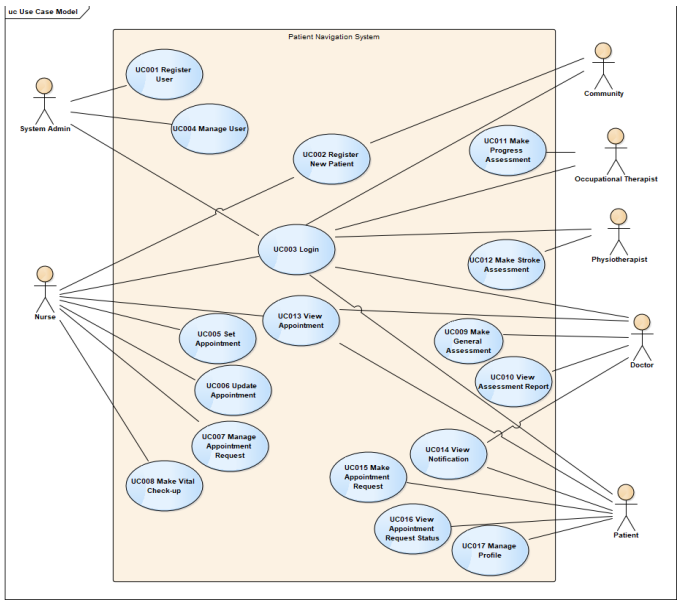

Figure 2. Use Case Diagram of Mobile Based PNS

This system has seven actors, which are patient, doctor, nurse, physiotherapist, occupational therapist, community, and system admin. All system users need to use their respective username and password to log into the system before using this system. By using this system, the doctor and patient can easily check their appointments. The patient may also request to make an appointment with the doctor on a specific date and time via this system. Besides, the system also enables the physiotherapist, occupational therapist, and doctor to easily make an assessment of the patient. It provides a convenient way for the therapists and doctors to track the rehabilitation progress of the patient.

Sequence diagram is an interaction diagram in Unified Modeling Language (UML). It shows the details on how the operations are carried out. Figure 3 shows the sequence diagram of Make Appointment Request use case.

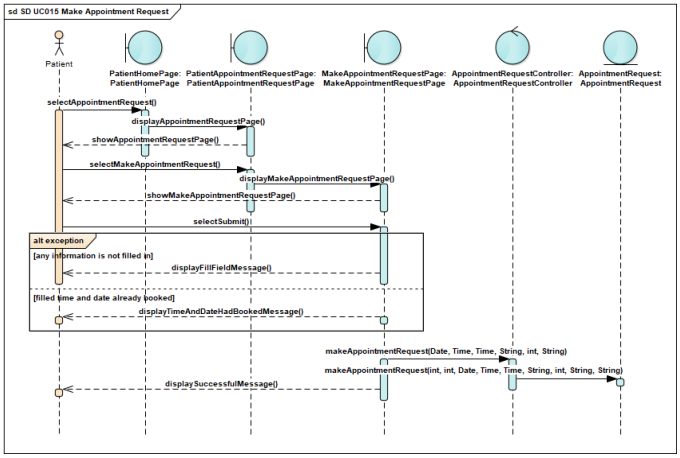

Figure 3. Sequence Diagram of Make Appointment Request

Activity diagram is a behavioral diagram in the Unified Modeling Language (UML). It is an advanced version of a flow chart that models the flow of a use case. Figure 4 shows the activity diagram of the Make Appointment Request use case.

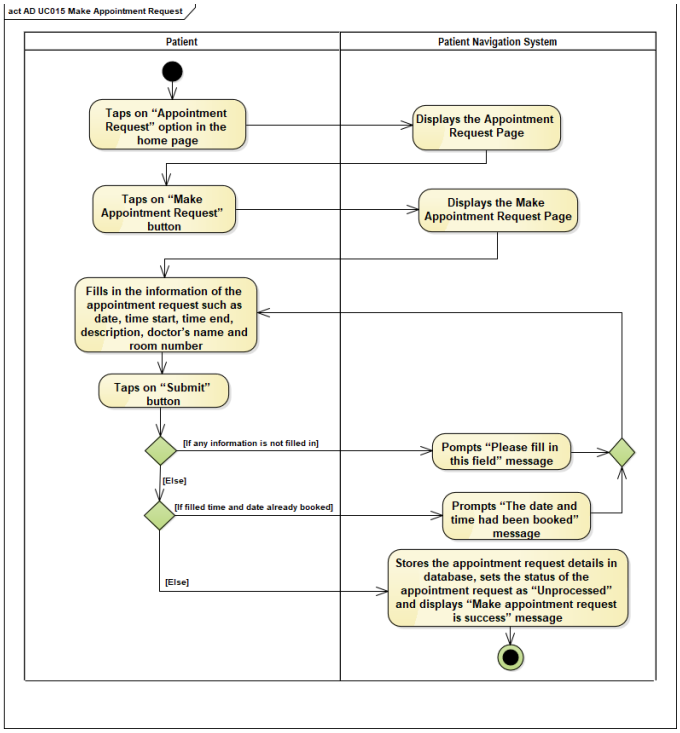

Figure 4. Activity Diagram of Make Appointment Request

For the system design, Model-View-Controller (MVC) is chosen to develop PNS. MVC is an architecture pattern that splits the application into three major components, which are View, Model, and Controller. The Model component represents the data to the end-user. It will be either the data transferred between the View and Controller components or any data related to the business logic. The View component represents the user interface that the end-user interacts with. It will display the model data to the end-user. The Controller component is the main operational component of the system. It connects the end-user and the system. It interacts with the View and Model

components to process all the business logic and handle the requests and responses from the end-user.

The reason that MVC was chosen is because it speeds up the development process of the application. By using this architectural style, PNS can be developed in a shorter time and hence the end-user can use PNS earlier. Besides that, this architectural style allows the data to be represented in different ways to the different end-users, such as doctors, nurses, and patients. Furthermore, since the View, Model, and Controller components are independent components, the modification can be easily made and will not affect the entire MVC model. Therefore, it allows the developer to easily modify the code and add new code if there are any functions of the system that need to be updated or added according to the end-users' needs. Figure 5 shows the implementation of MVC in the software architecture design of mobile based PNS.

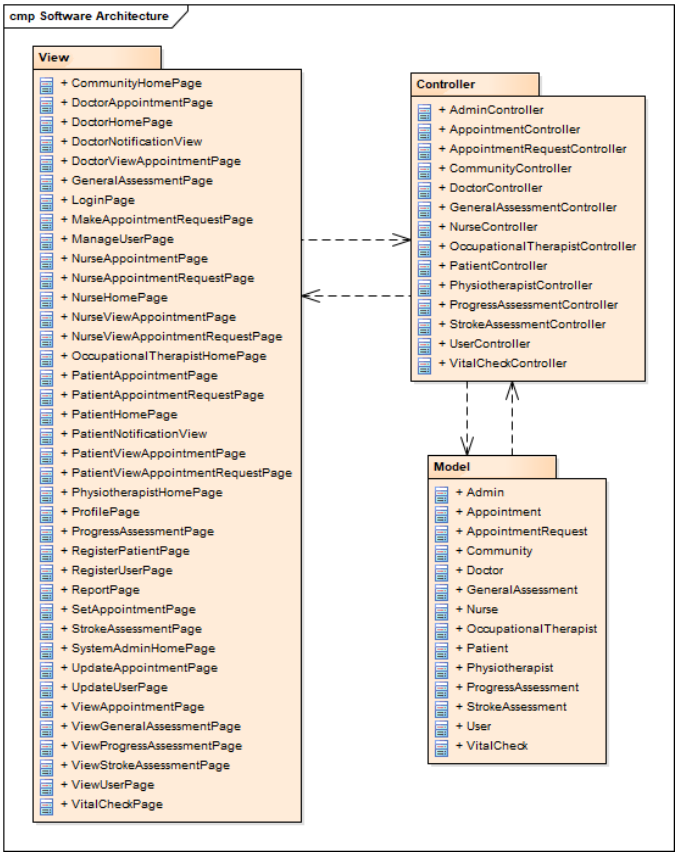

Figure 5. Software Architecture Design of Mobile Based PNS

V. IMPLEMENTATION

Mobile based PNS is developed using the Ionic Vue Framework for frontend development and PHP for the backend. The architectural style implemented in the system is MVC. MVC consists of three components, which are Model, View, and Controller. The View component handles the data presentation and user interaction of the system. It represents the user interface of the system. Figure 6 shows the code snippet of the Stroke Assessment View in Ionic Vue. When the physiotherapist taps on the "Make Stroke Assessment" button in the home page, the Stroke Assessment View will display a form for the physiotherapist to fill in the information for the stroke assessment done on the patient.

```
  <ion-list>
    <ion-list-header>
      <ion-label><b>Patient Information</b></ion-label>
    </ion-list-header>
    <ion-item class="first">
      <ion-label position="stacked" class="label">Full Name</ion-label>
      <ion-input type="text" name="fullname" v-model="state.user.fullname" disabled="true" />
    </ion-item>
    <ion-item>
      <ion-label position="stacked" class="label">IC Number</ion-label>
      <ion-input type="text" name="icNo" v-model="state.user.icNo" disabled="true" />
    </ion-item>
  </ion-list>
  <ion-list>
    <ion-list-header>
      <ion-label><b>Assessment Details</b></ion-label>
    </ion-list-header>
    <ion-item class="first">
      <ion-label position="stacked" class="label">Checked Date</ion-label>
      <ion-input type="date" name="date" v-model="state.strokeAssessment.date"
        required />
    </ion-item>
    <div class="error-message">{{ v$.strokeAssessment.date?.$errors[0]?.$message }}</div>
    <ion-item>
      <ion-label position="stacked" class="label">Progress Label</ion-label>
      <ion-input type="number" inputmode="numeric" name="progressLevel"
      v-model="state.strokeAssessment.progressLevel" required />
    </ion-item>
    <div class="error-message">
      {{ v$.strokeAssessment.progressLevel?.$errors[0]?.$message }}
    </div>
    <ion-item>
      <ion-label position="stacked" class="label">Current History</ion-label>
      <ion-input type="text" name="currentHistory"
        v-model="state.strokeAssessment.currentHistory" required />
    </ion-item>
    <div class="error-message">
      {{ v$.strokeAssessment.currentHistory?.$errors[0]?.$message }}
    </div>

    <ion-item>
      <ion-label position="stacked" class="label">General Condition</ion-label>
      <ion-input type="text" name="generalCondition"
        v-model="state.strokeAssessment.generalCondition" required />
    </ion-item>
    <div class="error-message">
      {{ v$.strokeAssessment.generalCondition?.$errors[0]?.$message }}
    </div>
    <ion-item>
      <ion-label position="stacked" class="label">Goal</ion-label>
      <ion-input type="text" name="goal"
        v-model="state.strokeAssessment.goal" required />
    </ion-item>
    <div class="error-message">{{ v$.strokeAssessment.goal?.$errors[0]?.$message }}</div>

    <ion-item>
      <ion-label position="stacked" class="label">Sensation</ion-label>
      <ion-input type="text" name="sensation"
        v-model="state.strokeAssessment.sensation" required />
    </ion-item>
    <div class="error-message">
      {{ v$.strokeAssessment.sensation?.$errors[0]?.$message }}
    </div>
    <ion-item>
      <ion-label position="stacked" class="label">Description</ion-label>
      <ion-textarea name="description" v-model="state.strokeAssessment.description"
        rows="3" required></ion-textarea>
    </ion-item>
    <div class="error-message">
      {{ v$.strokeAssessment.description?.$errors[0]?.$message }}
    </div>
  </ion-list>
</ion-list>
<div class="ion-text-center">
  <ion-button color="success" @click="selectSave">Save</ion-button>
</div>
</form>
```

Figure 6. Code snippet for Stroke Assessment View in Ionic Vue

Controller is the component that serves an interface between the Model and View components. It handles all the business logic and incoming requests. Figure 7 shows the code snippet for Stroke Assessment Controller in Ionic Vue. When the physiotherapist taps on the "Save" button in Stroke Assessment View, the Stroke Assessment Controller will send the form data from the View to the Stroke Assessment Model for storing the data in database.

```
import axios from 'axios';
import authHeader from './auth-header';

class StrokeAssessmentService {
  async makeStrokeAssessment(strokeAssessment){
    const response = await axios.post('makeStrokeAssessment', {
      patientID: strokeAssessment.patientID,
      date: strokeAssessment.date,
      progressLevel: strokeAssessment.progressLevel,
      currentHistory: strokeAssessment.currentHistory,
      generalCondition: strokeAssessment.generalCondition,
      goal: strokeAssessment.goal,
      sensation: strokeAssessment.sensation,
      description: strokeAssessment.description,
      physiotherapistID: strokeAssessment.physiotherapistID,
    },{
      headers: authHeader()
    });
    return response;
  }
```

Figure 7. Code snippet for Stroke Assessment Controller in Ionic Vue

Model is the component that stores data and its related logic. It responds to the instructions of the Controller if there are any user actions. Figure 8 shows the code snippet of the Stroke Assessment Model in PHP. It inserts the stroke assessment data that are passed by the Stroke Assessment Controller into the database.

```
$PatientID = $request->patientID;
$Date = $request->date;
$ProgressLevel = $request->progressLevel;
$CurrentHistory = $request->currentHistory;
$GeneralCondition = $request->generalCondition;
$Goal = $request->goal;
$Sensation = $request->sensation;
$Description = $request->description;
$PhysiotherapistID = $request->physiotherapistID;

try{
    $sql = "INSERT INTO strokeassessment (patientID, date,
    progressLevel, currentHistory, generalCondition, goal,
    sensation, description, physiotherapistID) VALUES
    (:PatientID, :Date, :ProgressLevel,
    :CurrentHistory, :GeneralCondition, :Goal,
    :Sensation, :Description, :PhysiotherapistID)";

    // Prepare statement and bind parameters
    $stmt = $conn->prepare($sql);
    $stmt->bindParam(':PatientID', $PatientID);
    $stmt->bindParam(':Date', $Date);
    $stmt->bindParam(':ProgressLevel', $ProgressLevel);
    $stmt->bindParam(':CurrentHistory', $CurrentHistory);
    $stmt->bindParam(':GeneralCondition', $GeneralCondition);
    $stmt->bindParam(':Goal', $Goal);
    $stmt->bindParam(':Sensation', $Sensation);
    $stmt->bindParam(':Description', $Description);
    $stmt->bindParam(':PhysiotherapistID', $PhysiotherapistID);

    // execute the query
    $stmt->execute();

    if($stmt->rowCount()>0){
        echo json_encode(array("message"=>"success"));
    }else{
        echo json_encode(array("message"=>"error"));
    }
```

Figure 8. Code snippet for Stroke Assessment Model in PHP

The system consists of seven types of users. Since the requirements of each interface depend on the type of user, the user interface will vary. Figure 9 shows the interface that allows the physiotherapist to make a stroke assessment for the patient.

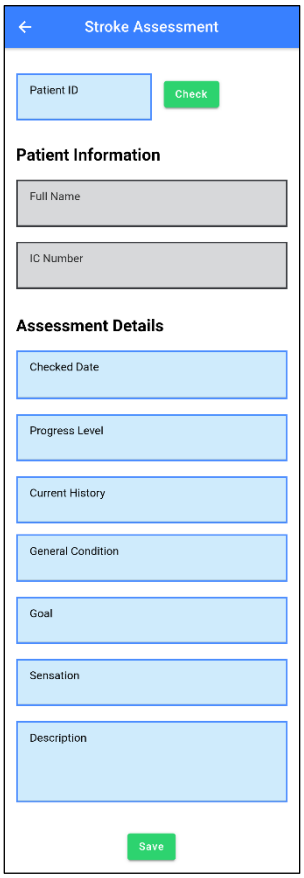

Figure 9. Main Interface for Physiotherapist to Make Stroke Assessment

Figure 10 shows the interface that allows the occupational therapist to make progress assessment for the patient.

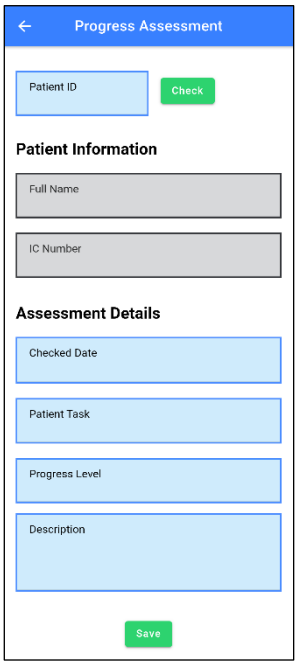

Figure 10. Main Interface for Occupational Therapist to Make Progress Assessment

The doctor can view assessment reports such as general report, stroke report and progress report. Figure 11 shows the interface that allows the doctor to view the details of stroke assessment.

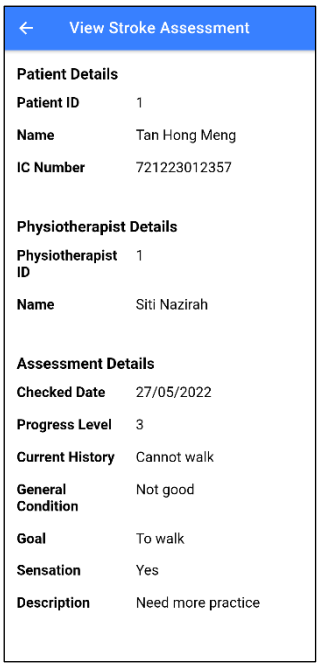

Figure 11. Main Interface for Doctor to View Stroke Assessment

The interface for viewing appointment will be accessed by the doctor, nurse and patient. However, the interfaces will be slightly different for these three types of user. For example, the doctor can only view appointments that are set to him or her, which is also the same as the patient. Figure 12 shows the interface that allows the doctor to view the details of the appointment.

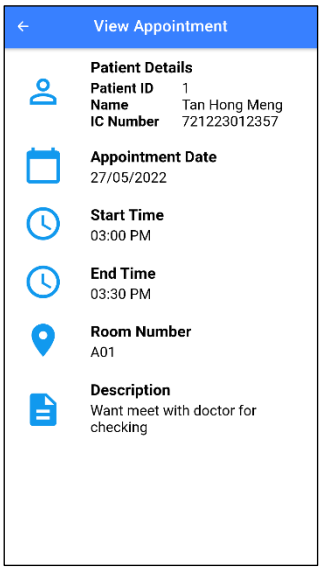

Figure 12. Main Interface for Doctor to View Appointment

VI. TESTING AND RESULTS

After the implementation of the system has been completed, system testing is carried out to validate the system is compliance with its specified requirements and meet the stakeholder's expectations.

K. Black Box Testing

Black Box Testing is one of the common testing techniques that involves testing the functionalities of the software system without knowing its internal code structure, implementation details, and internal paths. It fully focuses on input and output of the software system and also entirely based on software requirements and specifications. Table II shows one of the test cases for make stroke assessment functionality. The result of the test case is pass.

L. User Acceptance Testing

User Acceptance Testing (UAT) is a testing approach in which the end user or customer verifies that the software system meets the software requirements and specifications before it is moved to the production environment. At this phase, the system will be tested by actual users. They will need to complete the given tasks and some instructions will be provided for each task.

Five testers were invited to test the system. The testers are the staff who work at the IT Unit of HSA, including the stakeholder, Ushananthiny Suvelayutnan. The ages of the testers ranged from 40 to 45 years old. The work experience of the testers at the IT Unit of HSA ranges from about 12 to 20 years. Three of the testers are male, while the other two are female. Before starting the testing process, the procedure of task execution was briefly explained. The testing was mainly focused on the usability of the system. After the testers completed the tasks, a Google Form was sent to the testers to obtain their opinions or feedback on the developed system.

Figure 13 shows the graph of users' opinions on the user-friendliness of the system. The x-axis of the graph represents the degree of agreement, where 1 indicates "Strongly Disagree", 2 indicates "Disagree", 3 indicates "Neutral", 4 indicates "Agree", while 5 indicates "Strongly Agree". Based on the results of the testing, most of the users agreed that the system is user-friendly.

TABLE II. TEST CASE DESIGN FOR MAKE STROKE ASSESSMENT

Test Case ID	TC012_01	**Test Case Description**	Test the Make Stroke Assessment Functionality in PNS		
Created By	Pua Lee Ling	**Reviewed By**	Pua Lee Ling	**Version**	1.0
QA Tester's Log	Initial test case added				
Tester's Name	Pua Lee Ling	**Date Tested**	08/06/2022	**Test Case (Pass/Fail/Not Executed)**	Pass

S #	Prerequisites:
1	The physiotherapist must have a registered account and logged into the system.
2	There is active internet connection to the system
3	
4	
5	
6	
7	
8	

S #	Test Data
1	Patient ID = 1
2	Date = 12/05/2021
3	Progress level = 3
4	Current history = unable to walk
5	General condition = good
6	Goal = practice to walk
7	Sensation = yes
8	Description = still need more practice

Test Scenario	Verify on entering valid Patient ID, progress level, current history, general condition, goal, sensation, and description and selecting date, the physiotherapist can make stroke assessment

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Taps on "Make Stroke Assessment" option in the home page.	System displays the Stroke Assessment Page.	System displays the Stroke Assessment Page.	Pass
2	Fills in the valid Patient ID, progress level, current history, general condition, goal, sensation, and description and select a date	Valid Patient ID, progress level, current history, general condition, goal, sensation, and description are filled in and the date is selected	Valid Patient ID, progress level, current history, general condition, goal, sensation, and description are filled in and the date is selected	Pass
3	Taps on "Save" button	System stores the stroke assessment details in database and displays "Stroke assessment is completed" message	System stores the stroke assessment details in database and displays "Stroke assessment is completed" message	Pass

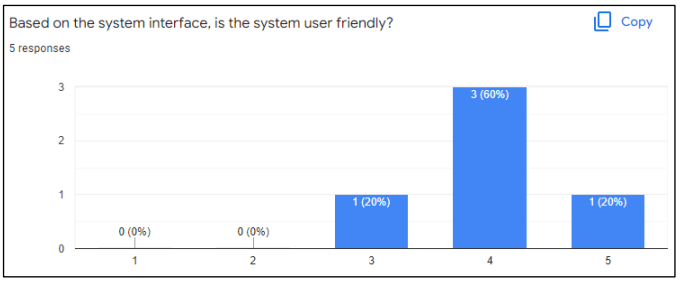

Figure 13. User-friendliness of The System

VII. CONCLUSION

Mobile based PNS has achieved the project objectives stated beforehand. The requirements for developing the mobile based PNS have successfully gathered and elicited from the stakeholder during the requirement analysis phase. Besides, this project has designed a mobile based PNS according to the requirements. A mobile application for PNS for rehabilitation of paralyzed patients at HSA was then developed based on the design according to the requirements. Black Box Testing and User Acceptance Testing were done on the developed mobile application for PNS to ensure the system is free of bugs and to verify that the functionalities of the system meet the stakeholder's requirements. There are some suggestions that can be considered for the future improvement of the system, which are allowing appointments to be set between the patient and the therapist, enhancing the notification functionality and the manage appointment request functionality, and adding the search functionality to the system.

ACKNOWLEDGMENT

We would like to express my sincere appreciation to my dearest stakeholder, Ushananthiny Suvelayutnan for giving detailed requirements of the system, guidance and support during the process of completing this project. Besides, a sincere thanks to staffs from IT Unit of Hospital Sultanah Aminah who participated in User Acceptance Testing.

REFERENCES

[1] Peckham, P. H. and Kilgore, K. L. Challenges and Opportunities in Restoring Function After Paralysis. IEEE Transactions on Biomedical Engineering, vol 60(3), March 2013, pp. 602–609.

[2] Freeman, H. P. The Origin, Evolution, and Principles of Patient Navigation. Cancer Epidemiology Biomarkers & Prevention, vol 21(10), October 2012, pp. 1614–1617.

[3] Desi Hertin, R., & Ismael Al-Sanjary, O. Performance of Hospital Information System in Malaysian Public Hospital: a Review. International Journal of Engineering & Technology, vol 7(4.11), October 2018, pp. 24-28.

[4] Allevio Ventures Inc. (2020). Allevio Patient Navigator. https://appadvice.com/app/allevio-patient-navigator/1521513319. Available online May 12, 2021.

[5] Navigation Tracker. (n.d.). Patient Navigation Case Management Software. http://www.navigationtracker.com/. Available online May 12, 2021.

[6] Add Addendum Note to a Locked Encounter. (n.d.). https://customer.first-insight.com/maximeyessqlehr/htmlpages/UserGuide_EncounterAddAddendumNote.html. Available online May 22, 2021.

[7] Rehkopf, M. (2021). Kanban vs. scrum: which agile are you? https://www.atlassian.com/agile/kanban/kanban-vs-scrum. Available online July 26, 2021.

[8] Kirovska, N. and Koceski, S. Usage Of Kanban Methodology At Software Development Teams. Journal of Applied Economics and Business, vol 3(3), September 2015, pp. 25-34.

[9] Muhammad Syafiq Abdul Manan. Patient Navigation System For Rehabilitation Of Paralysis Patients. Bachelor's thesis. Universiti Teknologi Malaysia; 2020.