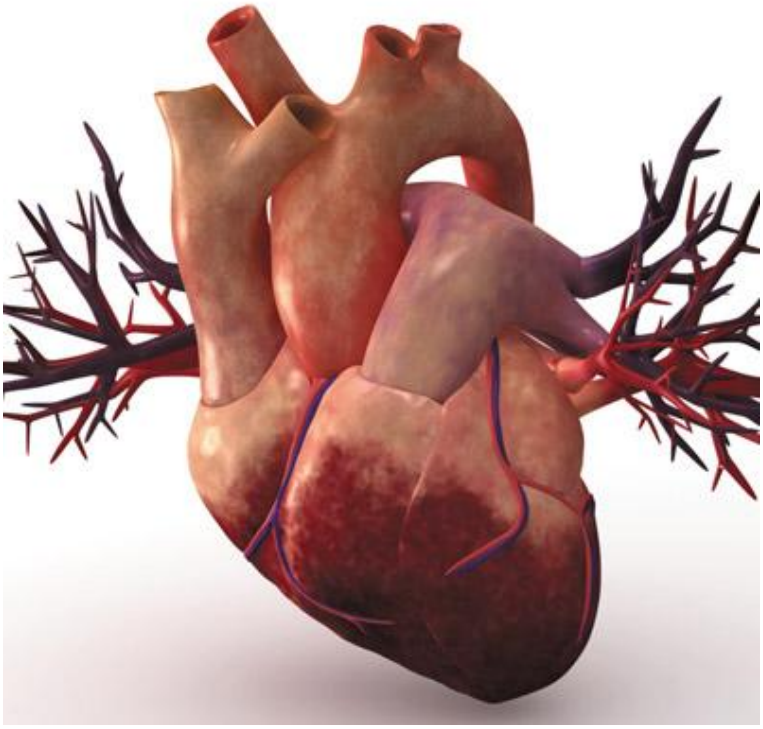
UNCC



Virtual Cardiac Rehabilitation Nurse

For Coronary Bypass Surgery Patients

Requirements Specification

Pioneers Group – (Team 4) Aneesh Garg, Elham Mahmoudi, Evan Mcclure, Philip Andrews, Rajini Mamidi

Contents

1. Introduction

Coronary bypass surgery is a procedure that restores blood flow to the patient's heart muscle by using a healthy blood vessel taken from his/her leg, arm, chest or abdomen and connects it to the other arteries in the heart. This diverts the flow of blood around a section of a blocked artery. However, patients are at potential risk to discontinue care after hospital discharge. According to a Brandeis study in *Circulation: Journal of the American Heart Association*, despite strong evidence that cardiac rehabilitation reduces disability and prolongs life, fewer than one in five people receives rehabilitation services after a heart attack or coronary bypass surgery. Cardiac rehabilitation reduces the overall risk of dying from a heart attack, the risk of future heart problems, decreases pain and the need for medicines to treat heart or chest pain, lessens the chance that the patients have to go back to the hospital or emergency room for a heart problem, improves overall health by reducing the risk factors for heart problems and improves the quality of life, making it easier for the patients to work, take part in social activities, and exercise. An absence of appropriate care and monitoring can lead to deterioration of the patient's health and safety, hospital readmission, and excessive costs to the health care system.

Technology can be utilized to contend with the nursing shortage and improve the quality of care. Virtual nursing can provide services for multiple facilities or individuals, and provide accessible, low cost nursing interventions for the patients, anytime and anywhere. The key objective of this technology is to provide the patients with guidance and information whenever they need it during their daily lives and not just during their scheduled visits. The patients can benefit from this application which offers a cybernetic nurse system at a relatively lower cost.

1.1 Project Overview and Statement of Proposal

Virtual Cardiac rehabilitation nurse will be a web based system. It will store the patients' data (medical and health history - medications, vital signs: blood pressure, heart rate, respiration rate, activity tolerance and exercise etc.) in a database before the surgery. Patients will have the ability to update data after the surgery, as per the planned rehabilitation interventions. The application will compare and track the patients' progress as they complete the recovery plan. We will create a web application where a patient will login with the username and password. The web application will allow patient to access his/her cardiac rehabilitation plan and have various categories like exercise, diet, medication, etc. Each category will have instructions which should be followed by the patient according to the calendar set for them. After completion of the calendar the patient and medical professional will get the feedback related to his health condition

compared to the patient's data before the surgery and continue with the plan. If the patient is not following the instructions, the application will generate an alert for the patient or medical professional. To implement the application we will create dummy rehabilitation plans and health data in database.

Statement of Proposal: *We propose to create a web application for patients undergoing cardiac rehabilitation after coronary bypass surgery.*

1.2 Project Scope and Objectives

The objective of the project is to provide virtual nurse services to a patient who has undergone a cardiac or coronary bypass surgery. The patient will be able to receive advice through the online system of the medical institution where he/she underwent surgery. The advice will be given to the user after collecting medical history and current information of the patient.

The scope of the project is proposed as below.

1. Patients will have online access to their medical center/hospital website where they had surgery through a secure access system, which uses their user name and password. We plan to create an imaginary hospital website, database and patient data as model for this purpose.
2. When the patient logs in, he/she goes to his/her personal rehabilitation plan page which has been designed by medical professionals who are responsible for the patient's rehabilitation plan. The plan includes exercise and activity, medication, diet, vital signs assessment etc. The feedback section under each weeks rehab plan will detail if he/she completed that part of the rehabilitation plan successfully or what areas need improving to optimize recovery. Moreover, the patient will be notified if she/he did not follow the instructions or the health conditions needs to be reported.
3. Her/his current health conditions can be compared to the baseline data (before surgery) and feedback can be provided as well. This feedback can be presented on the patient's webpage as text or by animation or diagram.

2. Non-Functional Requirements

Below are the non-functional requirements (grouped according to type) identified for this project.

2.1 Product Requirements

Req. No	Product Non – Functional Requirements
PR.1	The user interface of the application shall be implemented using HTML/CSS.
PR.2	The application /web server should be an Apache Tomcat version 6.
PR.3	The database should be a version of Microsoft SQL server 2008.
PR.4	Application and Database Server maintenance schedule should be between 11.00PM Saturday to 5.00AM Sunday every week.
PR.5	Rehabilitation plans must be simple enough to view that a patient can understand them with no external training.
PR.6	Progress reports must be generated within 120 seconds so that the data is relevant.
PR.7	Application must be available to Patients/Medical Staff 24/7 with the exception of scheduled outages with a minimum of 98% uptime.
PR.8	SQL Database should be able handle at least 2 GB of data, scale and maintain data integrity.
PR.9	The application should require no more than 1 hour of training for the typical user to be proficient with the system. Once trained, the typical user should be able to fill out their daily progress report in under 15 minutes.
PR.10	SQL Database will store and archive historical data (over 180 days old).
PR.11	Nightly backups will be performed on SQL Database.
PR.12	Multiple users should be able to login to the application at one time.
PR.13	Appropriate validation, escaping, and handling of special characters to avoid XSS and other server side related application vulnerabilities.
PR.14	The database should have logs of who last modified the patient logs and rehabilitation plan.

PR.15	The response time per click should be under 5 seconds for navigation and under 10 seconds if a query or request is sent.
PR.16	Percentage of events causing failure should be minimal (<5%).
PR.17	A log of changes to the rehabilitation plan must be maintained in the application

2.2 Organizational Requirements

Req. No	Organizational Non – Functional Requirements
OR.1	The virtual cardiac rehabilitation nurse system development process and deliverable documents must conform to the standards set by Dr. Tolone.
OR.2	The deliverables must be submitted using Moodle.
OR.3	The deliverables must be in the English language.

2.3 External Requirements

Req. No	External Non – Functional Requirements
ER.1	Virtual cardiac rehabilitation nurse application must be accessible from all HTML5/CSS3 compatible web browsers.

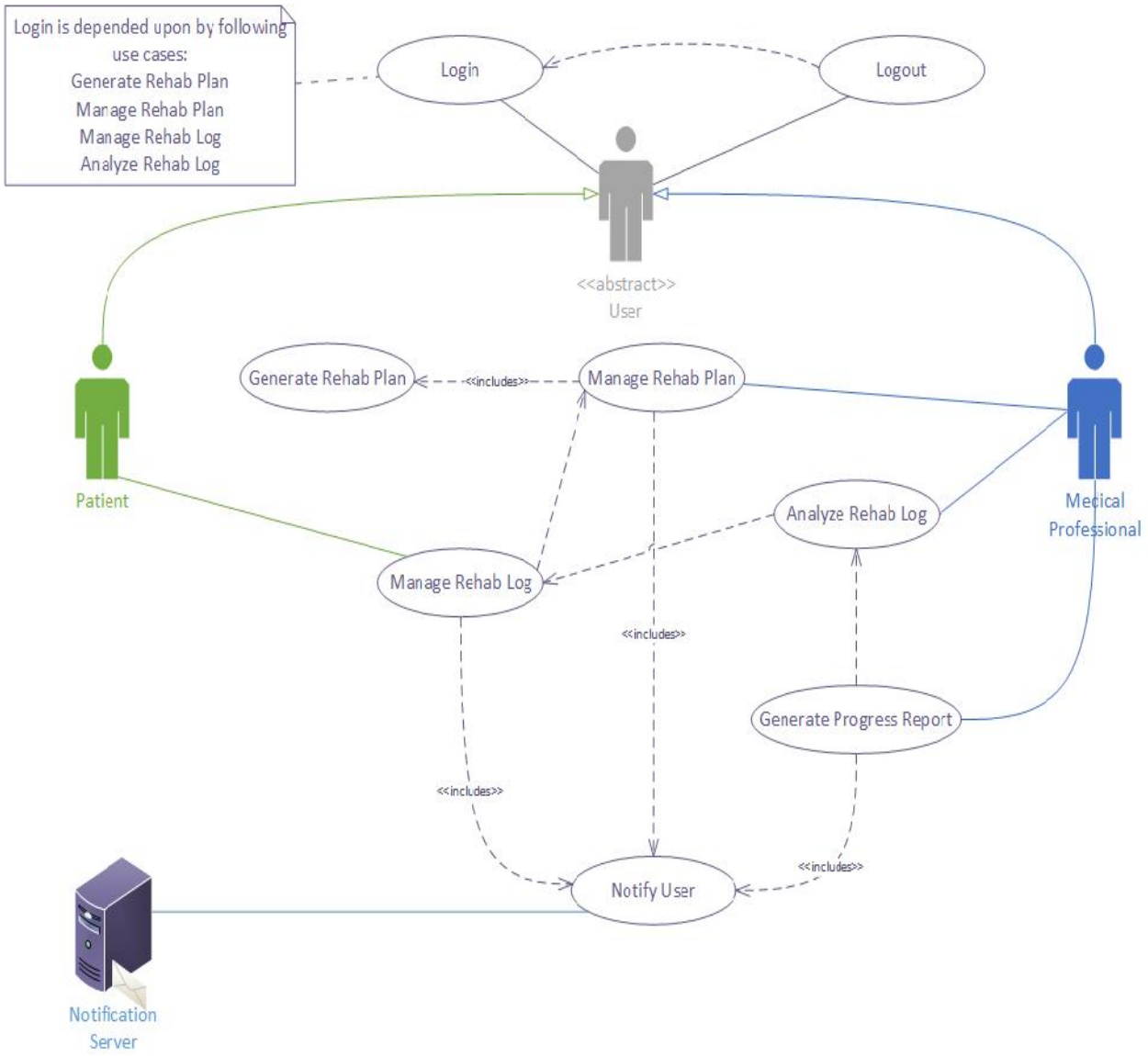
2.4 Security Requirements

Req. No	External Non – Functional Requirements
SR.1	The patients record must be visible only the patient and his medical professionals.
SR.2	Patient record must be inaccessible to anyone except patient and medical professionals.
SR.3	The patient should not be able to edit the rehabilitation plan.
SR.4	Doctors should have access to edit the information of their own patients only.
SR.5	Doctors should be able to see anybody's medical history or progress.
SR.6	The database should store password in encrypted form only.
SR.7	Data must be stored in a secure persistent data source.

SR.8	The password of the user should be encrypted at the client end before passing it to the server.
SR.9	User information should not be sent to server in a query string.
SR.10	Users can not login without a valid username/password.
SR.11	Strong passwords required (6-10 characters, Upper/Lower Case required, Min. 1 number).

3. Functional Requirements

3.1 Use Case Diagram



The Use Case Diagram consists of 7 use cases, 3 real actors and 1 abstract user as an actor. The actors Patient and Medical Professional are specialized Users. The diagram contains the Manage Rehab Plan use case which includes Generate Rehab Plan use case. The Manage Rehab Plan use case helps the Medical Professional to manage rehabilitation plan and is depended upon by the Manage Rehab Log use case. Manage Rehab Log use case helps patient enter his daily log of activities and is depended upon by Analyze Rehab Log use case. Analyze Rehab Log use case helps medical professionals analyze activities entered by the Patient and is depended upon by Generate Progress Report use case. Generate Progress Report use case generates the progress report of the Patient to be seen by medical professionals and patient. The use cases Manage Rehab Plan, Manage Rehab Log and Generate Progress Report includes Notify User use case in which Notification Server notifies corresponding user of any changes made due to the three use cases. The Use Case Diagram also contains Login and Logout use cases which helps user login and logout of the application. Logout use case depends upon Login use case while Login use case is depended upon by Generate rehab Plan, Manage Rehab Plan, Manage Rehab Log and Analyze Rehab Log use cases.

3.2 Actor Description

3.2.1 User

User is an abstract actor which is specialized by Patient and Medical Professional.

3.2.2 Patient

Patient is an actor who has undergone a cardiac surgery and has signed up for Virtual Cardiac Rehabilitation Nurse application for rehabilitation after surgery.

3.2.3 Medical Professional

Medical professional is an actor either a doctor or nurse who supervises rehabilitation of a patient.

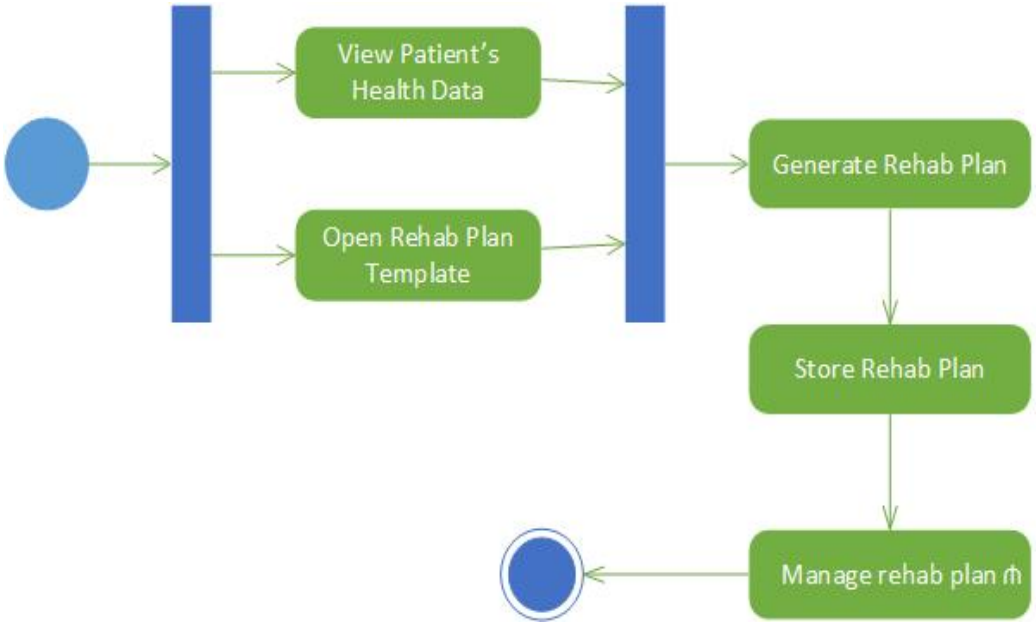
3.2.4 Notification Server

Notification Server is an external server which send notifications to all users.

3.3 Use Case Description

3.3.1 Generate Rehab Plan

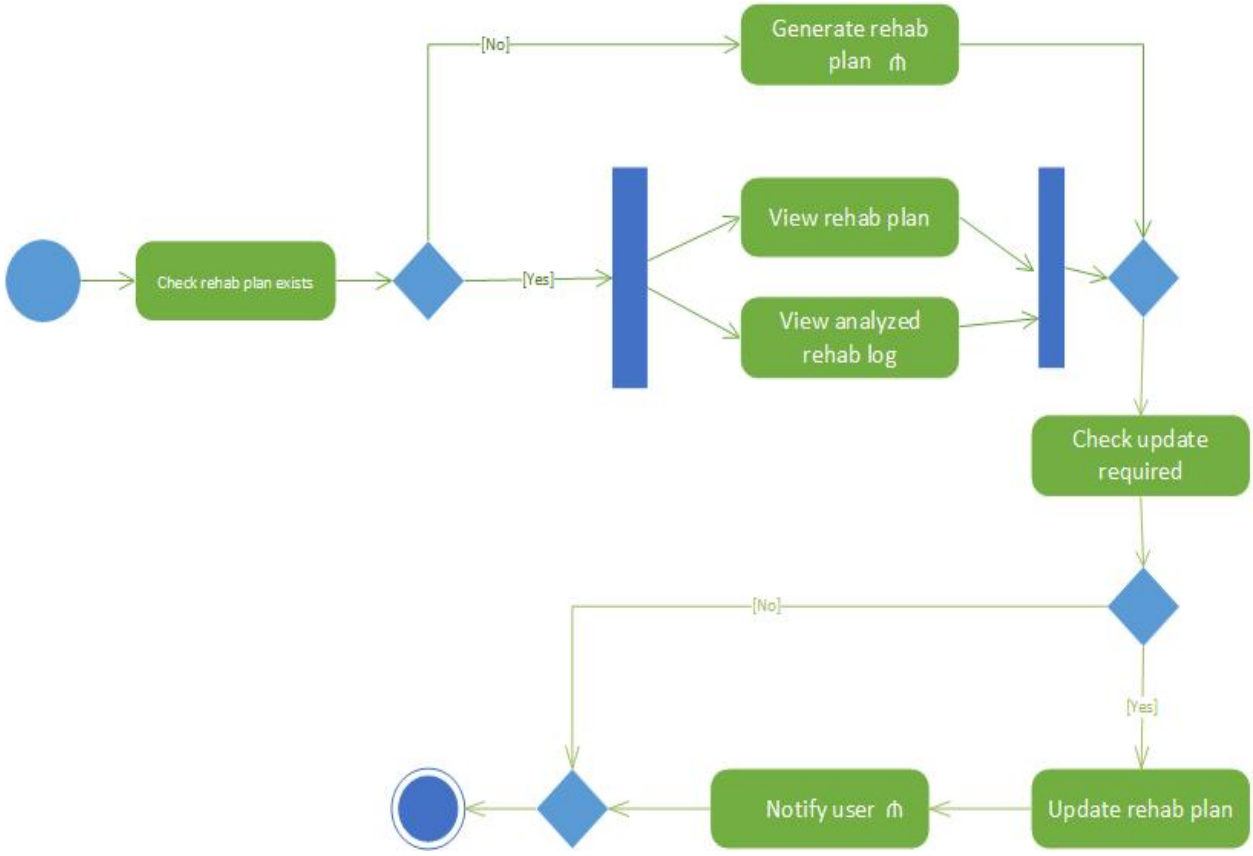
Elements	Description
Use case	Generate Rehab Plan
Participating Actors	Medical Professional
Goal	Medical Professional should be able to generate rehab plan
Pre-Conditions	• The system has patient's health data • The system has rehab plan template • Medical Professional is logged in
Post-Conditions	• Rehab plan is available for patient notification
Triggers	• New patient is admitted • Rehab plan is generated and manual action on rehab template is done
Primary and Alternative Flow of Events	1. The use case begins when Medical Professional views patient's health data and opens rehab plan simultaneously. 2. He then generates the rehab plan using above data. 3. He then stores generated rehab plan. 4. Manage Rehab Plan use case is triggered 5. The use case ends successfully.



3.3.2 Manage Rehab Plan

Elements	Description
Use case	Manage Rehab Plan
Participating Actors	Medical Professional
Goal	Medical Professional should be able to view/update rehab plan.
Pre-Conditions	• The system has patient's health data • The system has rehab plan template • Medical Professional is logged in
Post-Conditions	• Rehab plan is available for patient notification
Triggers	• New patient is admitted • Analyzed Rehab Log is available • Manual action on rehab template is done

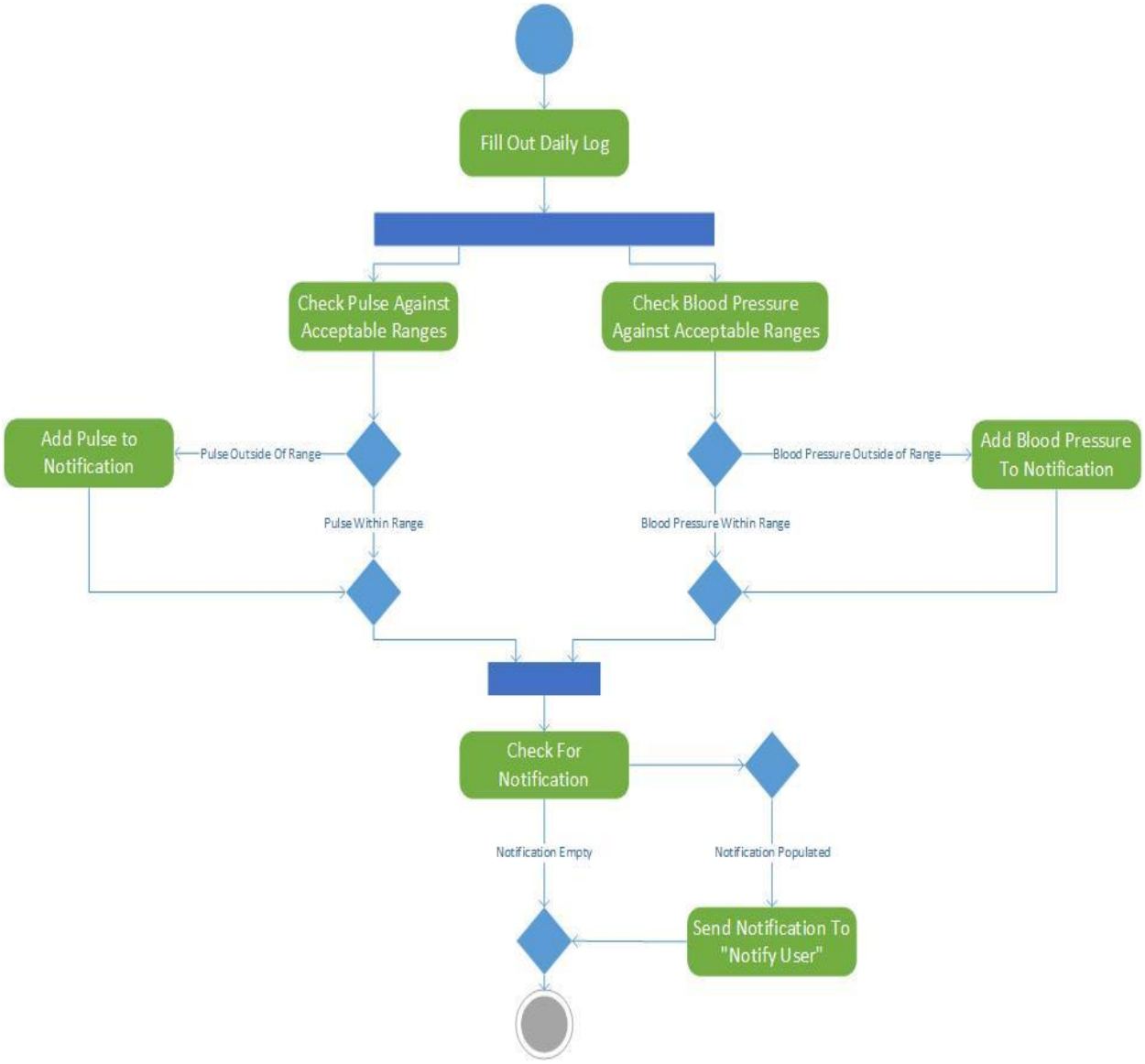
Primary and Alternative Flow of Events	**Primary Flow:** 1. The use case begins with Medical Professional checking the existence of rehab plan. 2. If plan exists Medical Professional views rehab plan and analyzed rehab log simultaneously. 3. He then checks whether update is required. 4. If update is required then he updates the rehab plan. 5. After update Notify User use case is triggered. 6. The use case ends successfully. **Alternative Flows:** • If in step 1 Rehab plan does not exist then Generate Rehab Plan use case is triggered after the use case resumes at step 3. • If in step 3 update is not required the use case ends successfully.



3.3.3 Manage Rehab Log

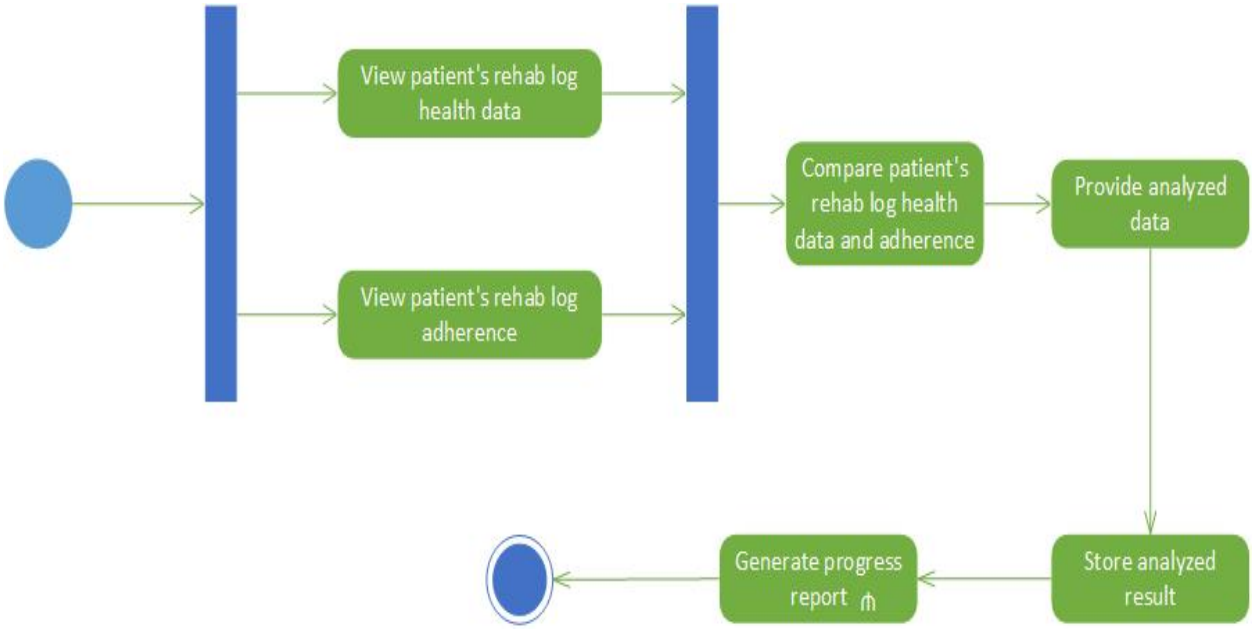
Elements	Description
Use case	Manage Rehab Log
Participating Actors	Patient
Goal	Allow the patient to fill out a daily log of activities and other metrics that will be used to track their progress during their rehabilitation.
Pre-Conditions	• Rehab Plan must be created via the "Manage Rehab Plan" use case. • Patient must be registered for application. • Patient must be logged in.
Post-Conditions	• Rehab log is available to be used in the "Analyze Rehab Log" use case • If the data entered into the report falls outside of predefined ranges specified in the "Manage Rehab Plan" use case, the "Notify User" use case is activated.
Triggers	• Manual selection by Patient
Primary and Alternative Flow of Events	1. A user will, either on their own volition, or because they have been reminded to by a notification, sent from the "Notify User" use case, will use the application to fill out a rehab log of their daily activities. 2. This log will include both critical and non-critical data. Critical data will consist of blood pressure and pulse, and all other metrics will be considered non-critical. 3. Upon saving blood pressure and pulse, the values entered into the log will be compared with the acceptable ranges previously defined in the "Manage Rehab Plan" use case. 4. If the values entered fall outside of the acceptable range, a notification addressed to both the medical professional and patient will be created containing the information relevant to this exception.

	5. If a notification has been generated when comparison to acceptable ranges is complete, the notification will be sent to both the patient and the medical professional via the “Notify User” use case and the process terminates. If no notification has been generated, the process terminates without sending anything.



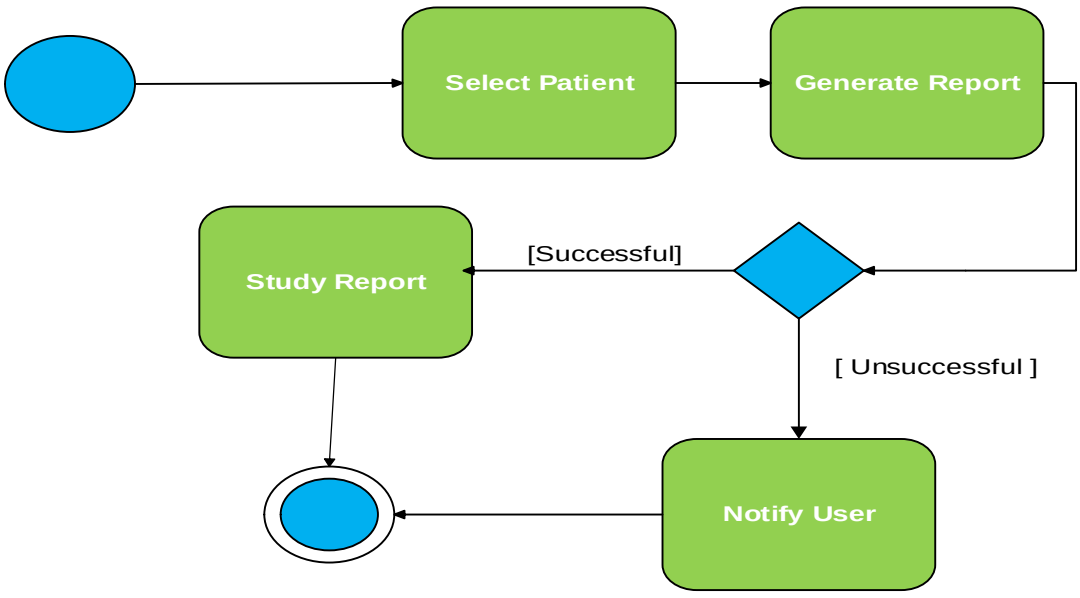
3.3.4 Analyze Rehab Log

Elements	Description
Use case	Analyze Rehab Log
Participating Actors	Medical Professional
Goal	Medical Professional should be able to analyze rehab log.
Pre-Conditions	• Rehab log is available for analyzing
Post-Conditions	• Patient's rehab log's analysis is available for generating progress report.
Triggers	• Manual selection by Medical Professional
Primary and Alternative Flow of Events	1. The use case starts with Medical Professional viewing patient's rehab log health data and patient's rehab log adherence simultaneously. 2. He then compares patient's rehab log health data and adherence. 3. Medical Professional provides analyzed data. 4. He then stores the analyzed result. 5. This use case ends successfully with the triggering of Generate Progress Report use case.



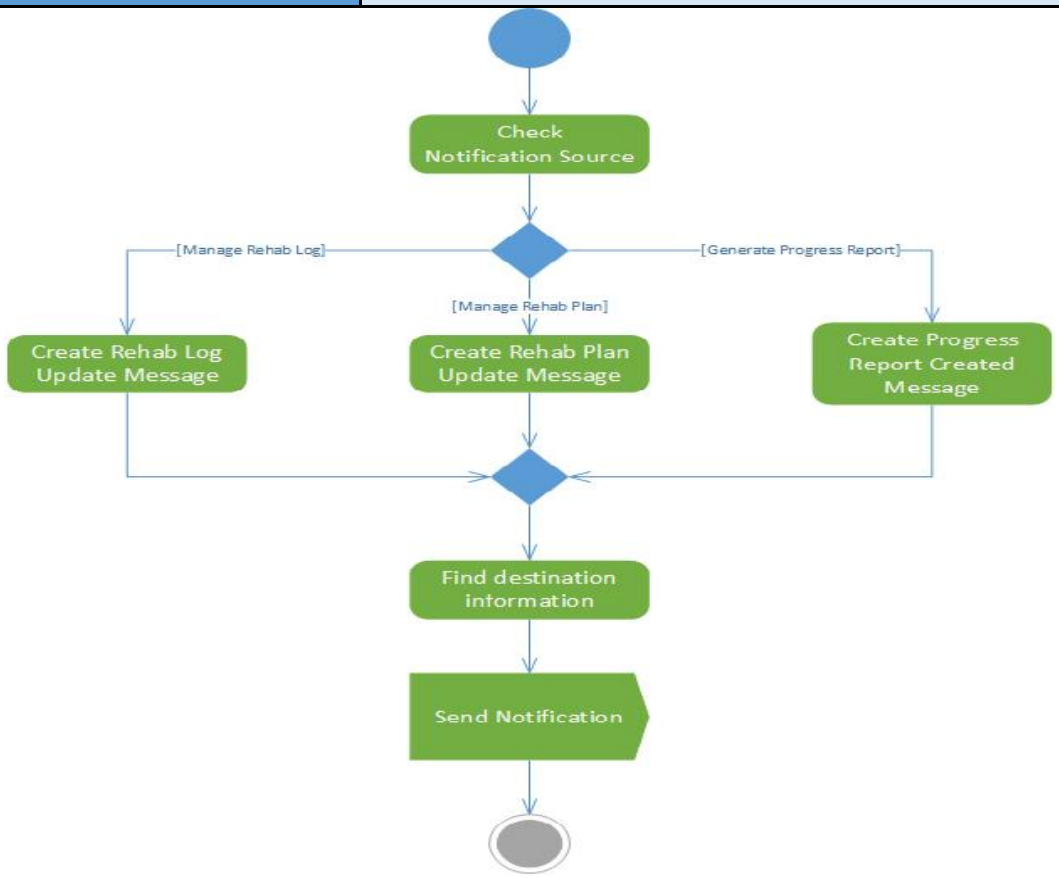
3.3.5 Generate Progress Report

Element	Description
Use Case:	Generate Progress Report
Participating Actors:	Medical Professional
Goal:	Produce documents which present the progress of the patient's recovery to the medical professional.
Pre-Conditions:	• Access to the application • Access to the patient record • Rehabilitation Plan Existence • Analyzed Rehab log information is available
Post Conditions:	If necessary Medical professional updates Rehabilitation plan
Triggers:	• Manually triggered by the Medical Professional.
Primary and Alternative Flow of Events:	1. Medical Professional selects the patient for which he wants to see the report. 2. He then runs the report for the selected patient. 3. If application is not able to generate a report, notifications will be sent to medical professional with the error message. (The notify user use case will be initiated). 4. The Medical Professional studies the report



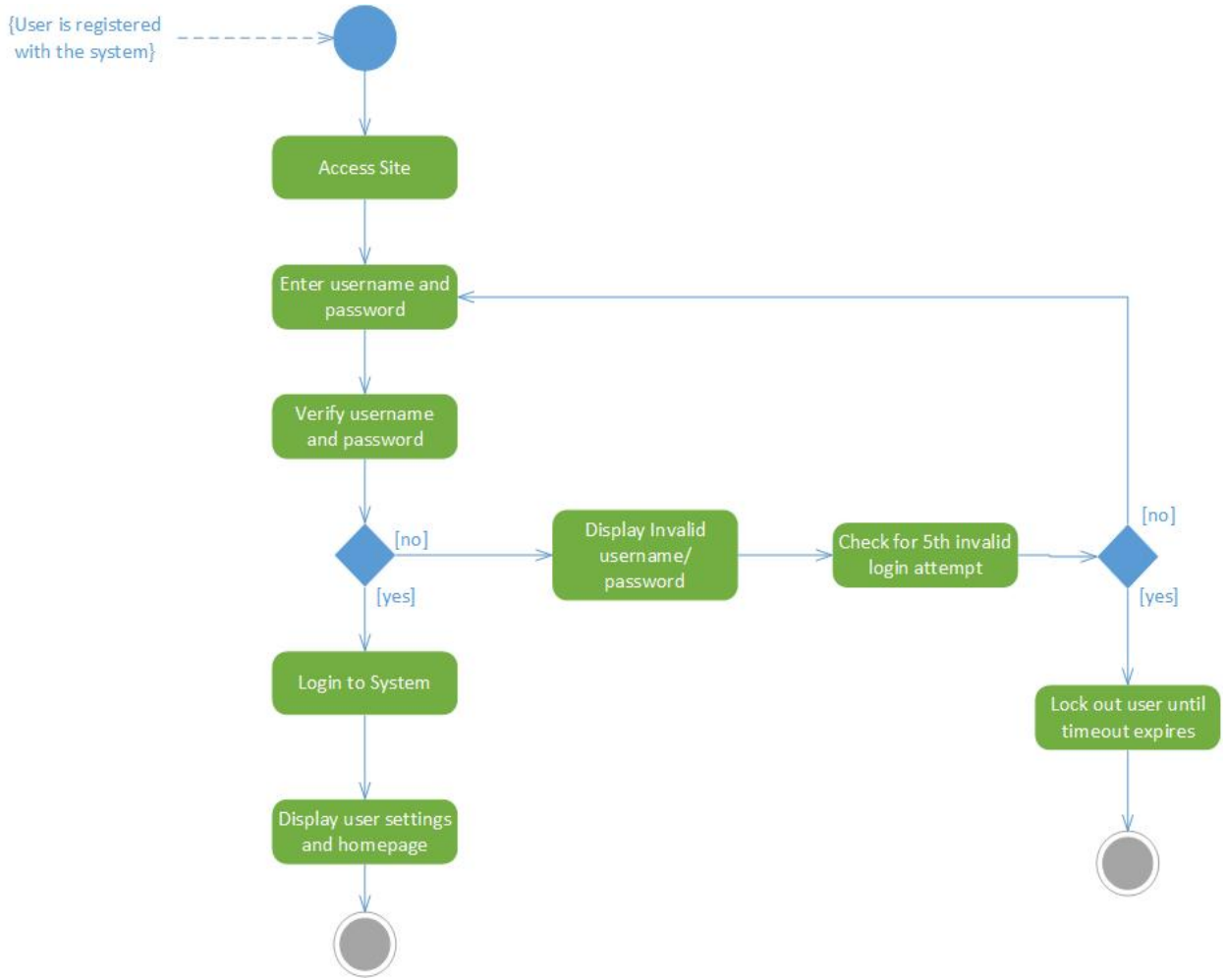
3.3.6 Notify User

Element	Description
Use Case:	Notify User
Participating Actors:	Notification Server
Goal:	Send notifications to Patient or Medical Professional.
Pre-Conditions:	• Notification server is online
Post Conditions:	Notification is sent to user to take further action
Triggers:	• Changes are made to rehab plan • Changes are made to Rehab Log • Progress Report is Generated
Primary and Alternative Flow of Events:	1. Check for the source of notification message 2. Create the corresponding message. 3. Find the destination information where notification needs to be sent. 4. Send the notification to the destination.



3.3.7 Login

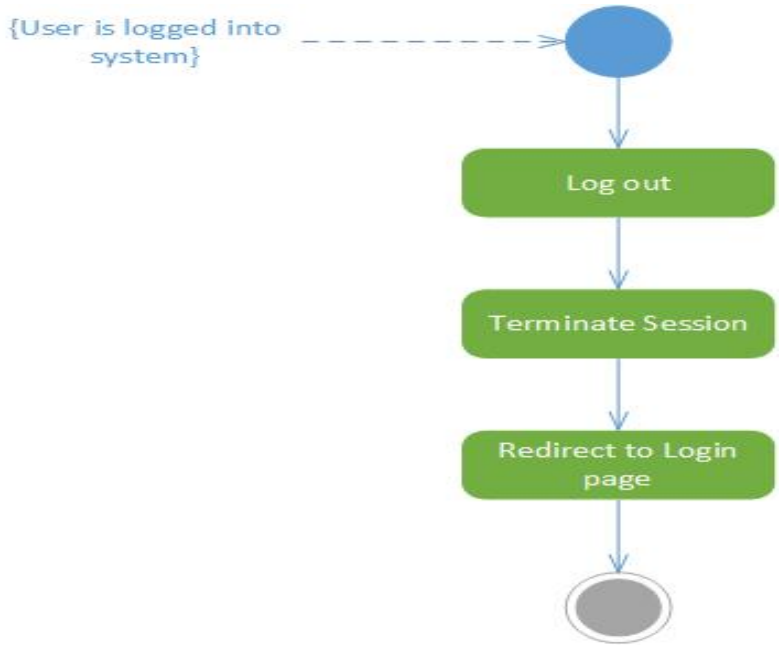
Element	Description
Use Case:	Login
Participating Actors:	User
Goal:	Users should be able to access the web application and login with valid username and password in order to access rehab plan.
Pre-Conditions:	• User has Internet access • User is already registered • Application is available (no outage)
Post Conditions:	• Homepage content is displayed to patient, doctor, or medical staff • System pulls user specific information from Database and displays to end user • Additional application functionality is available to user.
Triggers:	• Manual triggering
Primary and Alternative Flow of Events:	**Primary Flow:** 1. User accesses the web application by visiting URL 2. User enters username 3. User enters password 4. User submits information for authentication 5. System validates information 6. User logs into system and is directed to homepage **Alternative Flow:** 1. User accesses the web application by visiting the URL 2. User enters incorrect username or 3. User enters incorrect password 4. System displays error message 5. After 5 incorrect tries system locks user out



3.3.8 Logout

Element	Description
Use Case:	Logout
Participating Actors:	User
Goal:	A User should be able to logout of the web application.
Pre-Conditions:	• User has Internet access • User is logged into account (UC 3.3.7) • Application is available (no outage)
Post Conditions:	• User is redirected out of application • All functionality of application is disabled
Triggers:	• Manual triggering • Session Expired
Primary and Alternative Flow of Events:	**Primary Flow:** 1. User logs out of application

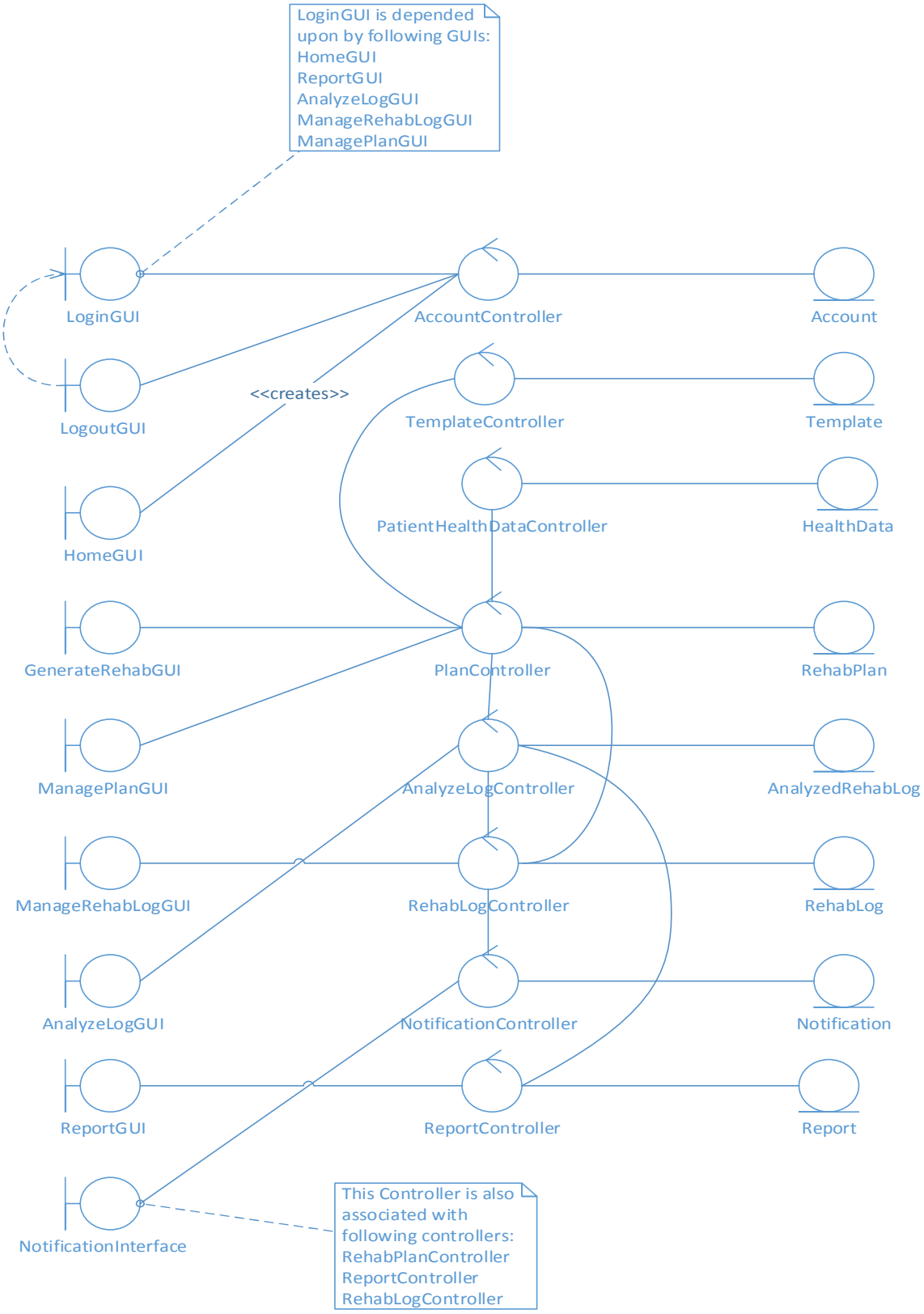
	2. System terminates user session 3. Redirected to login screen for user to login **Alternative Flow:** 1. User continues navigation of web application content and functionality



4. Requirement Analysis

4.1 Structural Analysis

4.1.1 Analysis Class Diagram



4.1.2 Boundary Classes

Class	Description
LoginGUI	Login GUI is the first interface that user will face once he/she starts the application. On this page user who is already registered will enter username and password to log into the application and access other functionalities. LoginGUI is depended upon by all other boundary classes except NotificationInterface.
LogoutGUI	Logout GUI is the interface which allows authenticated user to log out of the application and terminate ongoing session.
HomeGUI	Home GUI is the interface where user lands after he logs in the application. This interface displays all the notifications for the user.
GenerateRehabGUI	Generate Rehab GUI interface allows Medical Professional to generate rehab plan for patient considering template and patient's health data.
ManagePlanGUI	Manage Plan GUI interface allows Medical Professional to manage a patient's rehab plan. If plan does not exist then it will take user to GenerateRehabGUI to create a rehab plan. If rehab plan already exists then it will show analyzed log data and the rehab plan using which Medical Professional can update patient's rehab plan.
ManageRehabLogGUI	Manage Rehab Log GUI interface allows patient to enter his/her daily log of exercises, test results etc.
AnalyzeLogGUI	Analyze Rehab Log GUI interface will allow Medical Professional to view patient's log entries and analyze patient's condition based on it.
ReportGUI	Report GUI interface will allow Medical Professional to generate and view report of a patient.
NotificationInterface	NotificationInterface will send notification to Notification Server to distribute it to users.

4.1.3 Control Classes

Class	Description
AccountController	AccountController is used to handle information related to user's account. It is used to login or logout from the application.
TemplateController	TemplateController is used to get rehab plan templates from database.
PatientHealthDataController	PatientHealthDataController is used to handle patient's health data in database.
PlanController	PlanController is used to generate or manage patient's rehab plans. It is associated with TemplateController and PatientHealthDataController to get templates and patient's health

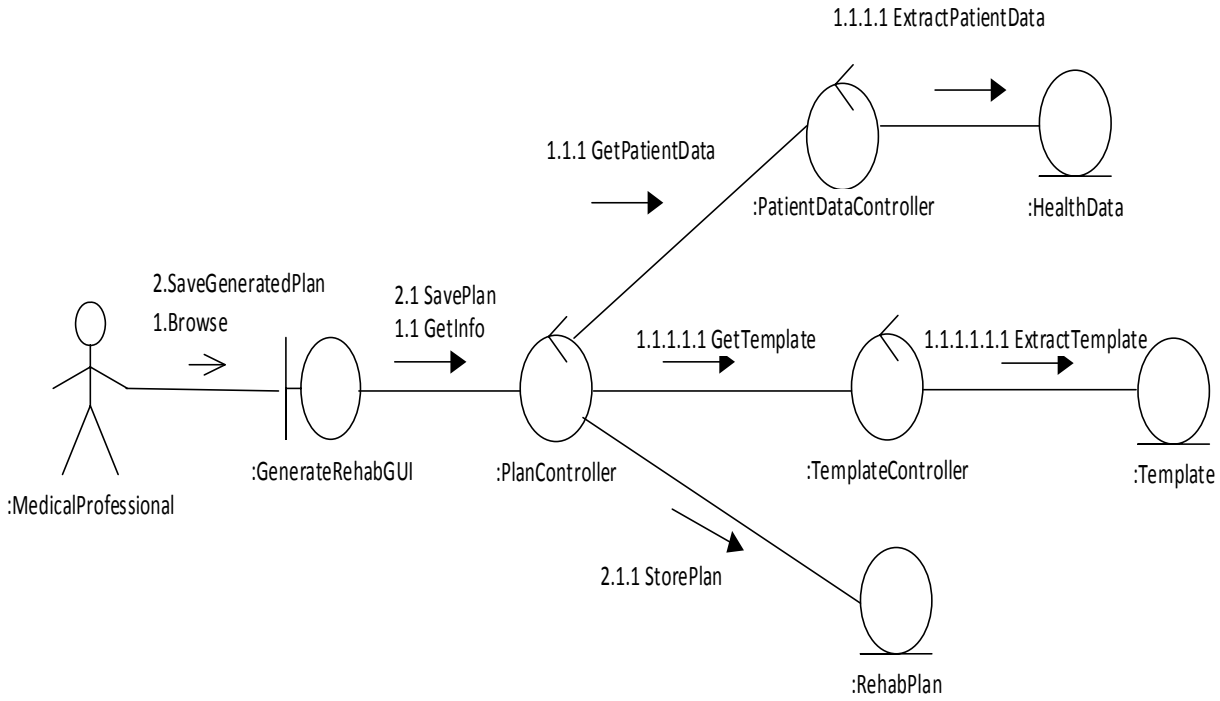
	data.
AnalyzeLogController	AnalyzeLogController will help medical professional to review patient's daily logs and allow storing analyzed log entries. It is associated with RehabLogController and PlanController to get data helpful in analyzing logs.
RehabLogController	RehabLogController will allow Patient to store daily logs in database. It is associated with Plan Controller to check for inconsistencies in daily logs and rehab plans. It is also associated with NotificationController to send alerts to appropriate users.
NotificationController	NotificationController helps to send notifications to users.
ReportController	ReportController helps in creating Reports. It is associated with AnalyzeLogController since it uses analyzed log data to create reports.

4.1.3 Entity Classes

Class	Description
Account	Account class is used to represent user account information.
Template	Template class is used to represent rehab plan template.
HealthData	HelthData class is used to represent patient's heatlh data
RehabPlan	RehabPlan class is used to represent rehab pan.
AnalyzedRehabLog	AnalyzedRehabLog class is used to represent analysis result of rehab logs.
RehabLog	RehabLog is used to represent rehab log entry.
Notification	Notification class is used to represent notification message.
Report	Report class is used to represent generated report.

4.2 Behavioral Analysis

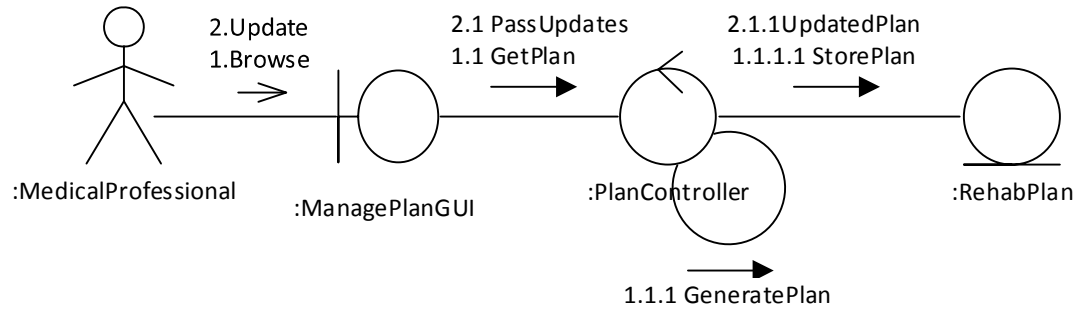
4.2.1 Generate Rehab Plan



1. Medical Professional browses the GenerateRehabGUI to read the patient's health data and view the template for the plan
   1.1. PlanController accepts the message to get the information that is to be displayed by the GenerateRehabGUI
      1.1.1. The PlanController then messages the PatientDataController to get the patient data
         1.1.1.1. PatientDataController extracts the relevant patient's data from the HealthData entity
            1.1.1.1.1. The PlanController messages the TemplateController to obtain the rehab plan template , that is used to create the rehab plan
               1.1.1.1.1.1. The TemplateController pulls the template from the Template entity which holds them
2. The Medical Professional saves the plan he generated using the template
   2.1. The message to save the plan is passed to the PlanController
      2.1.1. The PlanController saves the plan in the entity object RehabPlan

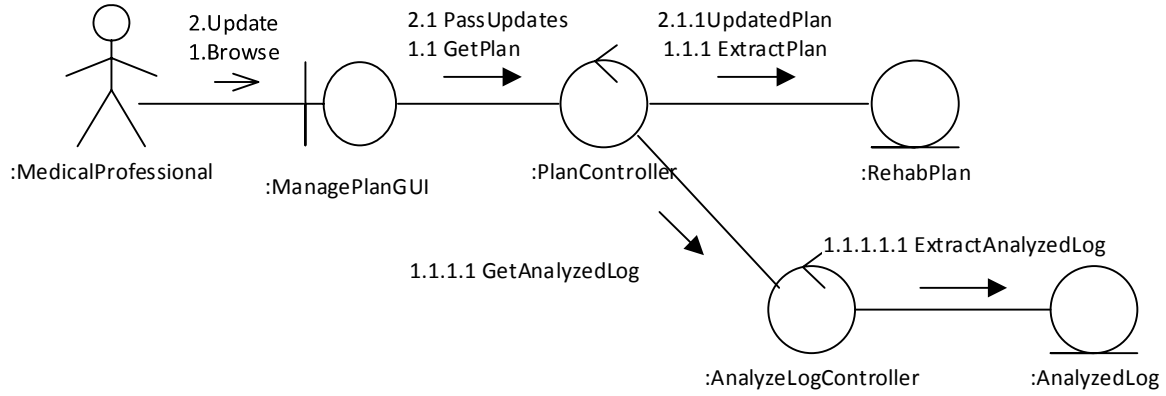
4.2.2 Manage Rehab Plan

No Plan Exists



1. Medical Professional browses the ManagePlanGUI to review the RehabPlan
   1.1. A message "GetPlan" is sent to 'RehabPlanController"
      1.1.1. Since the plan does not already exist for the patient, the PlanController creates one.
         1.1.1.1. The Generated plan is stored in the RehabPlan entity by the PlanController
2. The Medical Professional can update the plan using the ManagePlanGUI
      2.1.1. The updates given by the Medical Professional in the ManagePlanGUI are passed to the PlanController
         2.1.1.1. The updates to the plan are stored in the RehabPlan entity by the PlanController

Plan Exists



1. Medical Professional browses the ManagePlanGUI to review the RehabPlan
   1.1. A message "GetPlan" is sent to 'RehabPlanController"
      1.1.1. Since the plan already exists for the patient, the PlanController extracts it from the RehabPlan Entity.

1.1.1.1. The PlanController messages the AnalyzeLogController to get the Analyzed logs

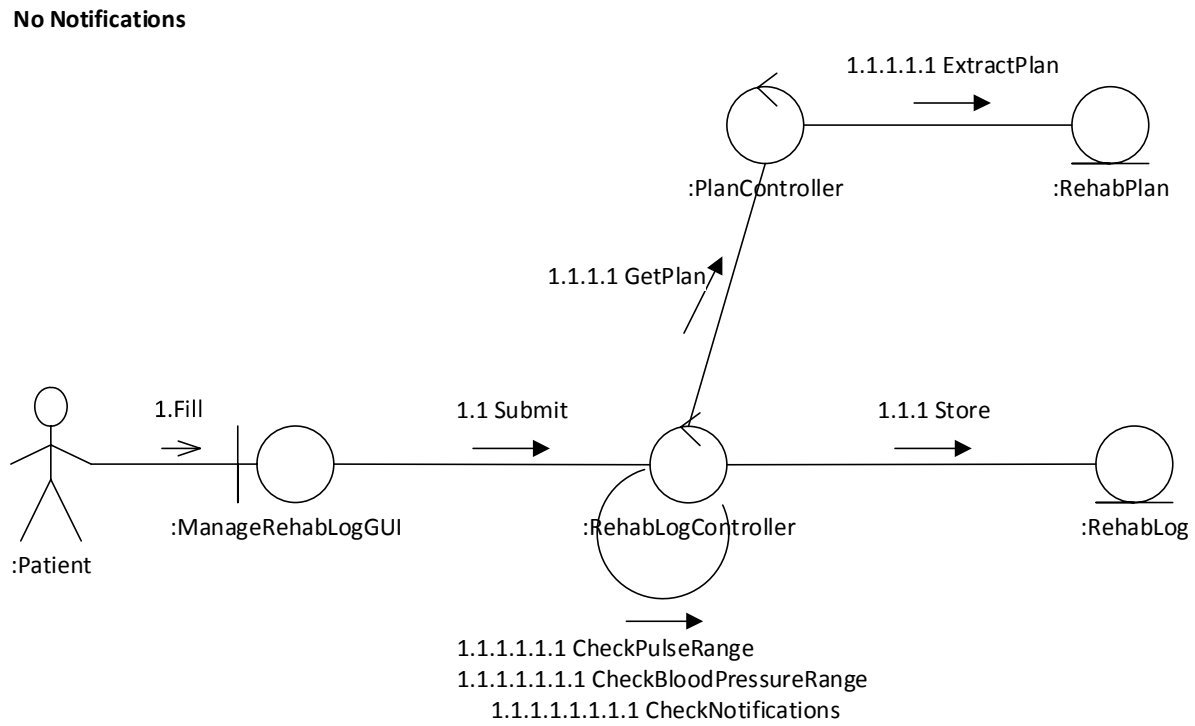
1.1.1.1.1. The AnalyzeLogController extracts the logs from the AnalyedLog entity

2. The Medical Professional can update the plan using the ManagePlanGUI

2.1.1. The updates given by the Medical Professional in the ManagePlanGUI are passed to the PlanController

2.1.1.1. The updates to the plan are stored in the RehabPlan entity by the PlanController

4.2.3 Manage Rehab Log



1. The Patient fills out all categories of log to record his vitals

1.1. The RehabLogController accepts the submitted vitals

1.1.1. The log is stored in the RehabLog entity

1.1.1.1. The RehabLog controller gets the plan from the plan controller

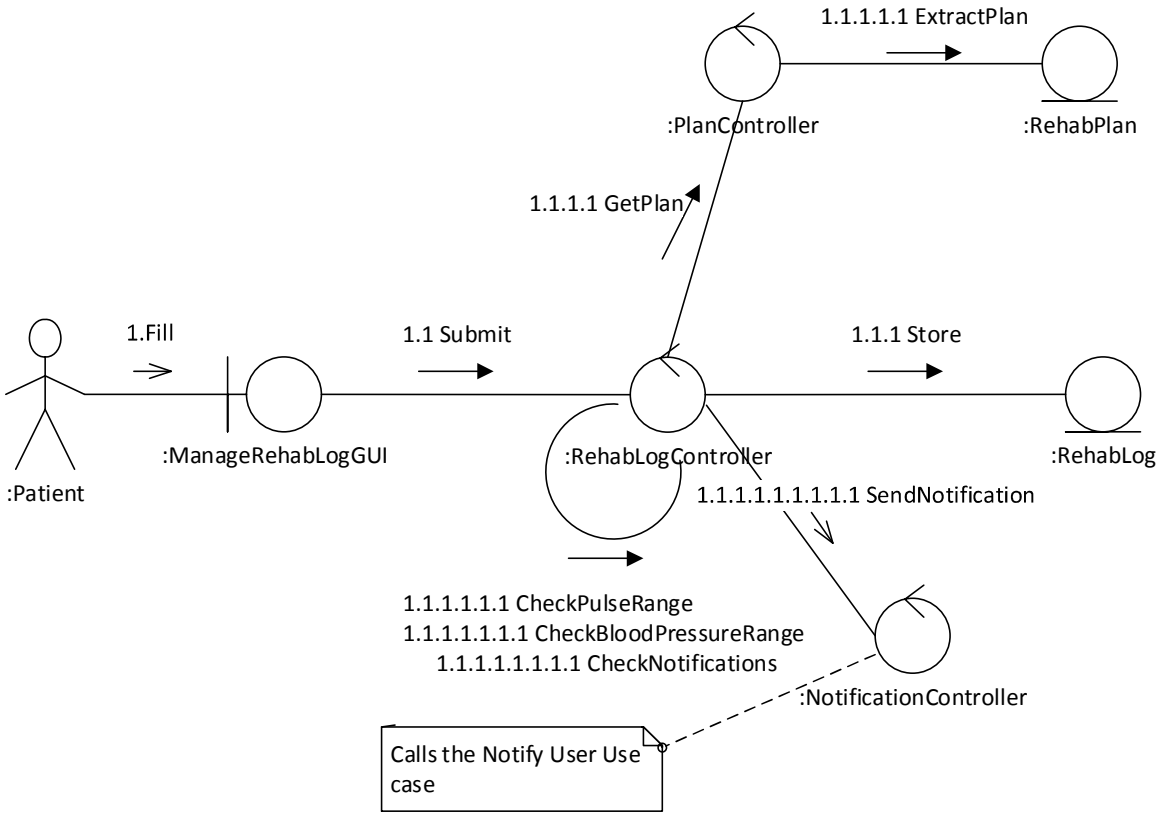
1.1.1.1.1. The plan is extracted from the RehabPlan entity

1.1.1.1.1.1. Pulse of the patient is checked against the baseline value in the rehab plan, for this flow; it is within limits

1.1.1.1.1.1.1. Pulse of the patient is checked against the baseline value in the rehab plan, for this flow; it is within limits

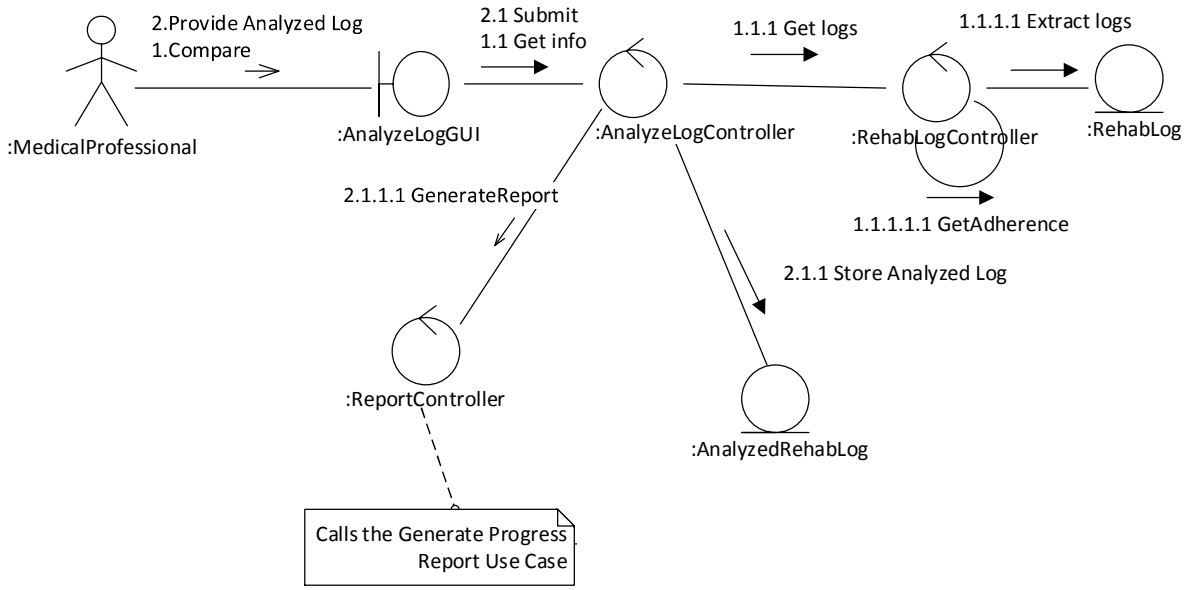
1.1.1.1.1.1.1.1. RehabLogController checks if there are any notifications, for this flow; there are no notifications

With Notifications



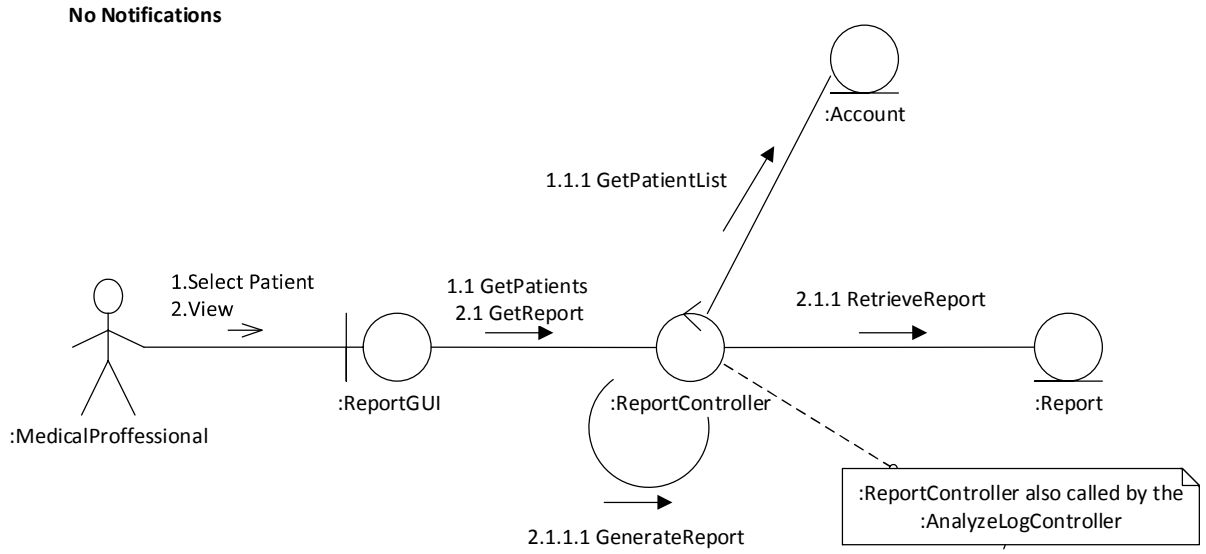
1. The Patient then fills out all categories of log to record his vitals
   1.1. The RehabLogController accepts the submitted vitals
      1.1.1. The log is stored in the RehabLog entity
         1.1.1.1. The RehabLog controller gets the plan from the plan controller
            1.1.1.1.1. The plan is extracted from the RehabPlan entity
               1.1.1.1.1.1. Pulse of the patient is checked against the baseline value in the rehab plan, for this flow; it is within limits
                  1.1.1.1.1.1.1. Pulse of the patient is checked against the baseline value in the rehab plan, for this flow; it is within limits
                     1.1.1.1.1.1.1.1. RehabLogController checks if there are any notifications, for this flow; there is a notification for either the pulse or blood pressure
                        1.1.1.1.1.1.1.1.1. Relevant message is sent to the NotificationController which distributes the message to the necessary users

4.2.4 Analyze Rehab Log

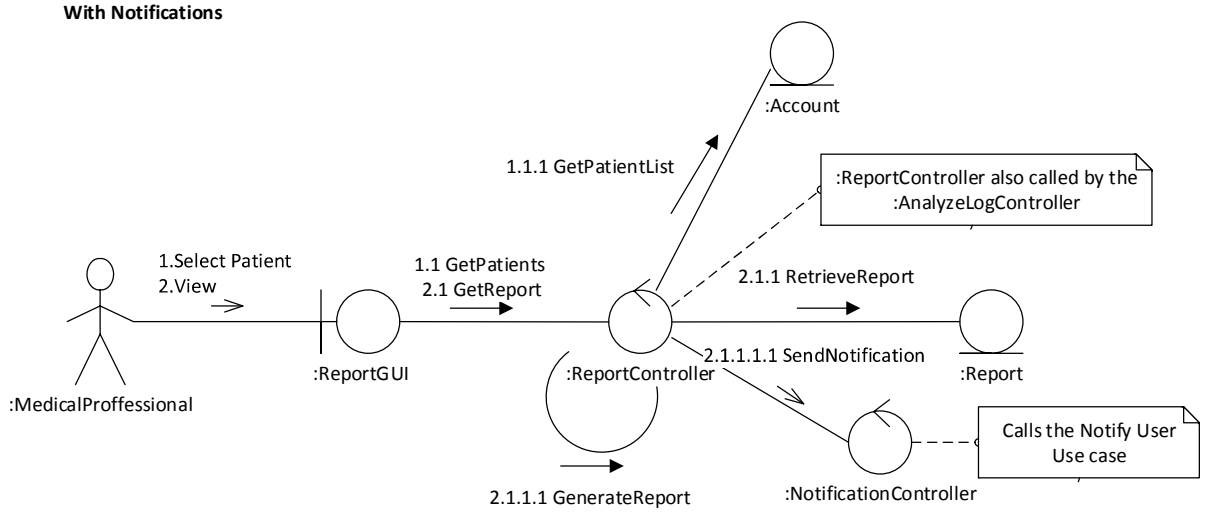


1. Medical professional compares patients rehab log through "AnalyzedLogGUI"
   1.1. A message "GetInfo" is sent to "AnalyzedLogController"
      1.1.1. A message "Get logs" is sent to "RehabLogController"
         1.1.1.1. A message "Extract logs" is sent to "RehabLog"
            1.1.1.1.1. The adherence of the patient is computed by the RehabLogController
2. Medical professional provides analyzed data through "Analyzed log GUI"
   2.1. A message "submit" is sent to "AnalyzedLogController"
      2.1.1. Then Analyzed data is stored in "AnalyzedRehabLog"
         2.1.1.1. Then a message "GenerateReport" is sent to "ReportController"

4.2.5 Generate Progress Report

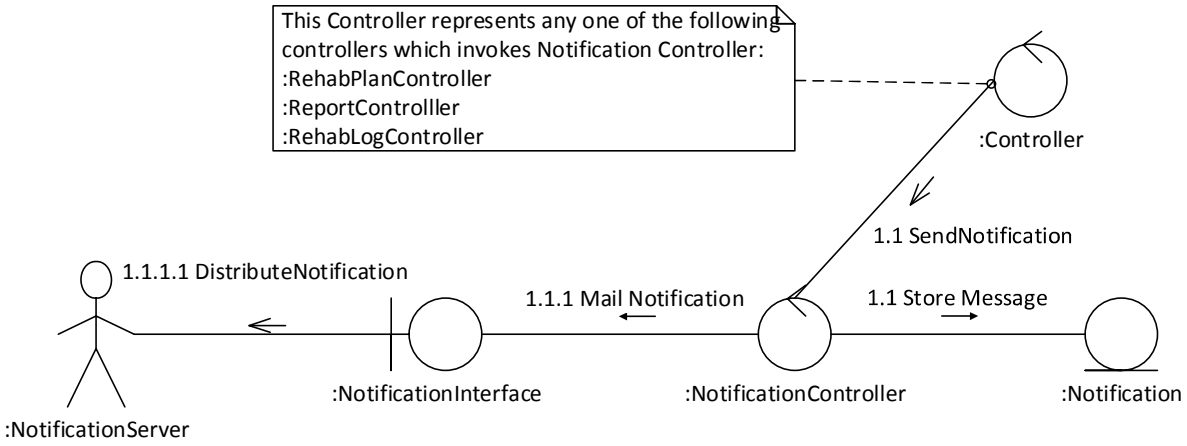


1. The Medical Professional goes into list of patients and selects a patient from the list presented by the ReportGUI boundary class retrieve
   1.1. The ReportController accepts the signal to get the list of patients
      1.1.1. The ReportController gets the list of patients to display from the Accounts entity
2. The Medical Professional chooses to view the report he/she wants to view for the patient
   2.1. The ReportController accepts the signal to get the report instance for the selected patient
      2.1.1. The Report features are extracted for the controller to generate the report
         2.1.1.1. The report is generated for the given patient



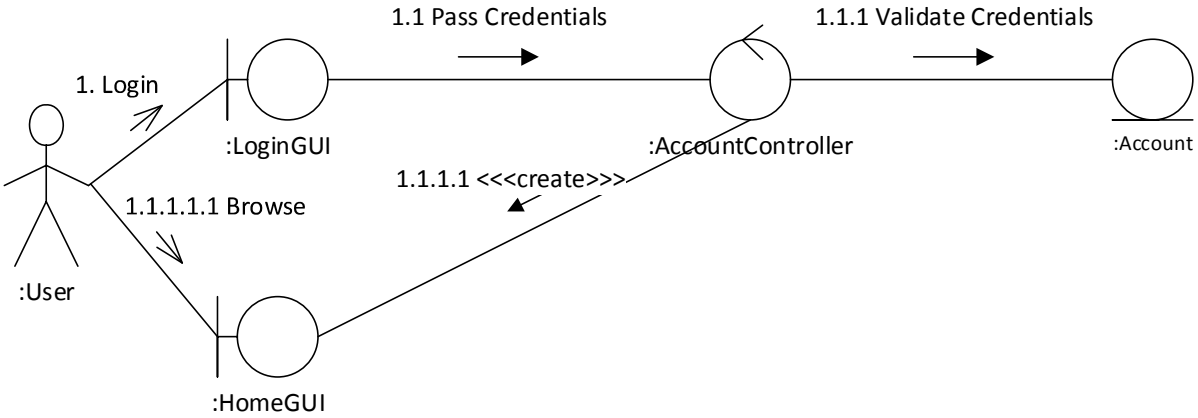
1. The Medical Professional goes into list of patients and selects a patient from the list presented by the ReportGUI boundary class
   1.1. The ReportController accepts the signal to get the list of patients
      1.1.1. The ReportController gets the list of patients to display from the Accounts entity
2. The Medical Professional chooses to view the report he/she wants to view for the patient
   2.1. The ReportController accepts the signal to get the report instance for the selected patient
      2.1.1. The Report features are extracted for the controller to generate the report
         2.1.1.1. The report is generated for the given patient
            2.1.1.1.1. Notifications is sent id there is an issue when generating the report

4.2.6 Notify User



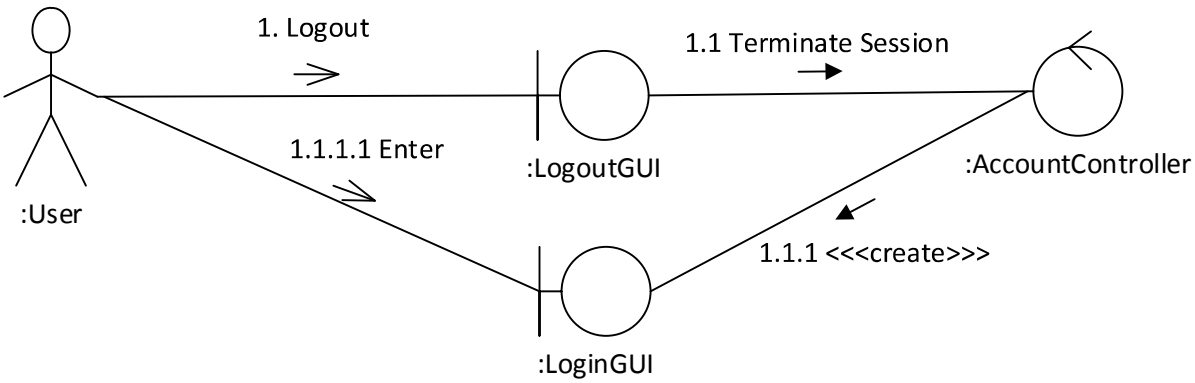
1. The Notification Controller is invoked by the message "SendNotfication" from the "RehabPlanController", "ReportController" or "RehabLogController" of the other use cases.
   1.1. The NotificationController stores the message to be send in the Notification entity
      1.1.1. The NotificationController then sends a signal to the Notification Interface to send the notification
         1.1.1.1. The NotificationServer accepts the message to distribute it to the recipients.

4.2.7 Login



1. The User logins in using the LoginGUI boundary class
   1.1. The AccountController accepts the credentials entered by the User in the GUI
      1.1.1. The AccountController validates the credentials of Medical Professional/Patient using the stored values in the Account Entity.
         1.1.1.1. Once validated, depending on validation result, User is redirected to the HomeGUI which displays relevant information if successful or error message is shown if unsuccessful
            1.1.1.1.1. User Browses the HomeGUI

4.2.8 Logout



1. The User logouts of application using the LogoutGUI boundary class
    1.1.1. The AccountController accepts the to terminate the session of the User
        1.1.1.1. The User is then redirected to the LoginGUI page
            1.1.1.1.1. The User can enter his credentials again if he wants to login to the application

4.3 Analysis Packages

4.3.1 Application General Packages:

User Package – This package contains three subsystems and describes the interactions an abstract user has with the system. Within the package is an authentication view class which depicts the interaction with the Login and Home pages that will be displayed when a user logs in or out of the system. The Authentication Service contains a controller which handles the authentication and validation of the user's credentials and the account entity class determines the role of the user.

Application Specific Packages:

Patient Package – This package contains a boundary which represents the interaction a patient will have when they login to the system. A Patient is a specialized user. They will see the GUI used to manage the rehab log in which they can update their progress.

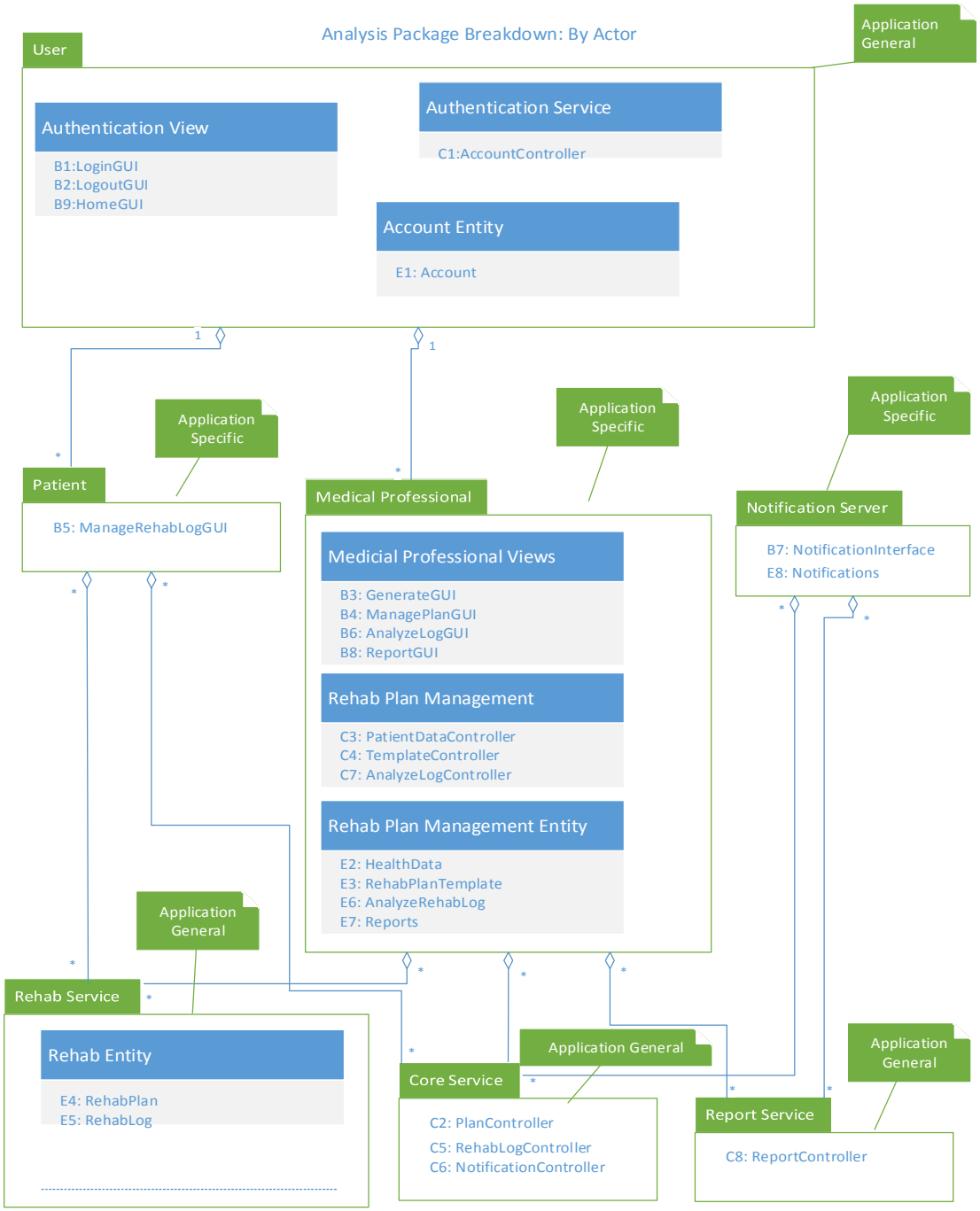
Medical Professional Package – This package contains the boundaries, controllers, and entities which allow a medical professional to interact with the system. A medical professional is a specialized user. A medical professional can interact with the system and has several different views or GUI's which allow them to manage patient data, select rehab plan templates for a patient, and analyze logs.

Notification Server Package – This package contains a boundary interface and entity object which allows for the management of notifications and alerts. Notifications fire based on set conditions or triggers.

Rehab Service Package – This package contains entities and is used to retrieve or display data that is stored on the backend database based on calls or requests that come from a user such as a medical professional or patient. The RehabLog or RehabPlan can be extracted.

Core Service Package – This package contains several controllers which provide core functionality and allows users to interact with the system. These controllers assist with the processing and updating of rehab plans, rehab logs, and notifications. All users interact with the core services package.

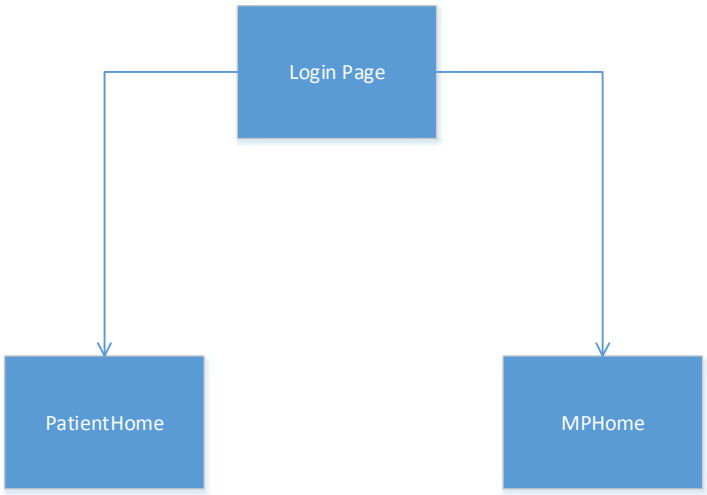
Report Service Package – This package contains a controller to handle the reporting capabilities of the system. Notification and medical professionals can interact with this controller.



5. Use-Case Storyboards

5.1 Login

5.1.1 Actor (User) Interface Flow Diagram



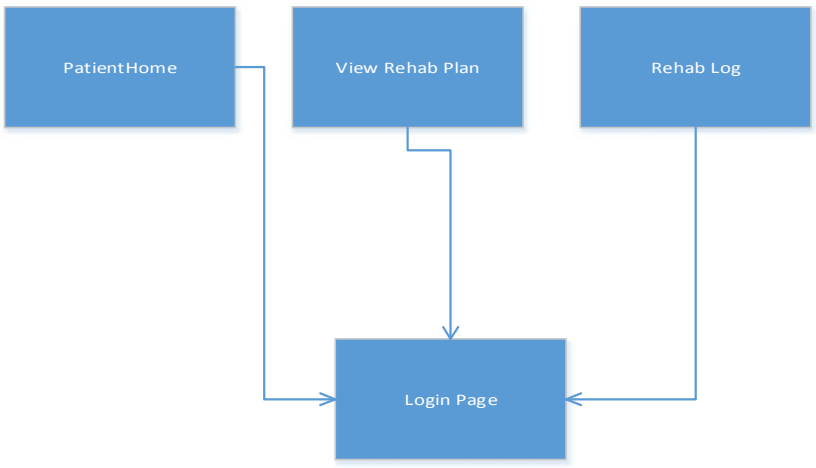
5.1.2 Storyboard

1. User (Patient or Medical Professional) navigates to application by visiting URL and lands at the Login page.
2. User enters their credentials on the Login page and submits the information
3. User gets redirected to either MPHome or PatientHome page depending on their role if the User has given correct credentials
4. In case the User has not given correct credentials the Login Page displays a message saying so.
5. If the credentials are incorrect the splash page shows related message to the user.

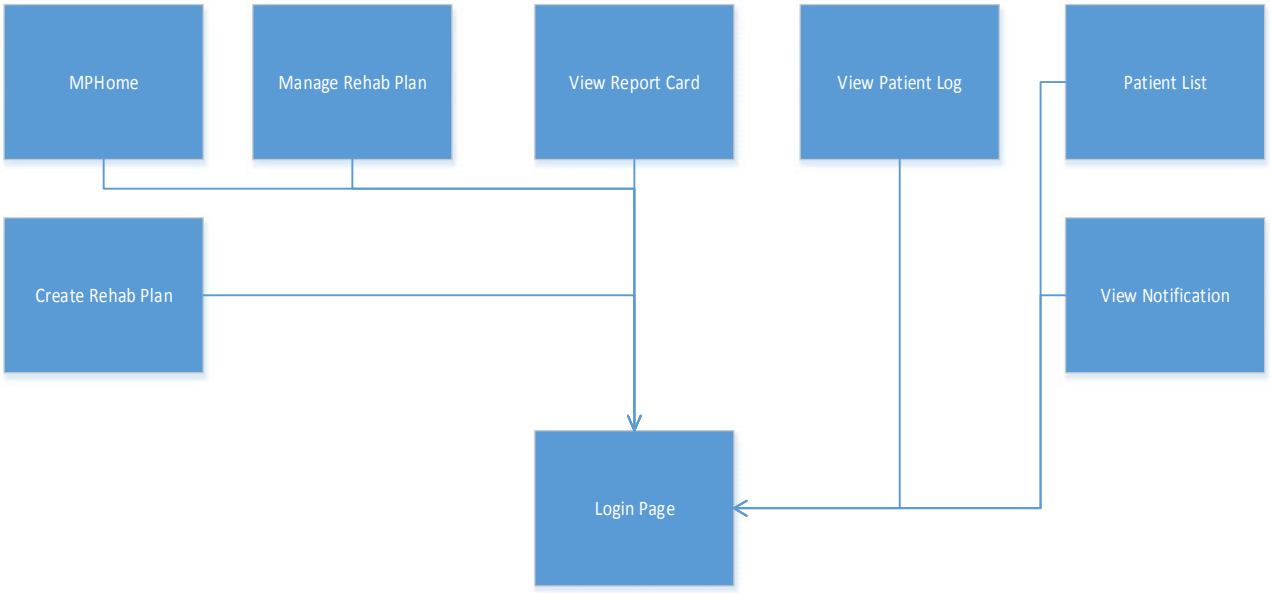
5.2 Logout

5.2.1 Actor Interface Flow Diagram

Patient:



Medical Professional:



5.2.2 Storyboard

Patient

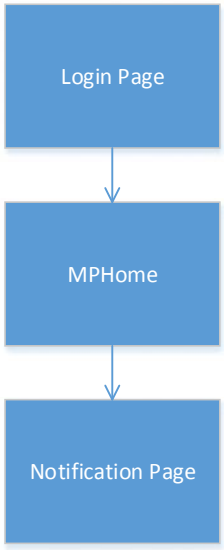
1. The patient can be in one of the pages (i.e PatientHome, View Rehab Plan, Enter Daily Log) when he decides to log out and clicks on the logout button on the page.
2. Once the logout button is clicked, the application terminates user session
3. Redirected to Login Page for User

Medical Professional

1. The medical professional can be in one of the pages (i.e MPHome, Manage Rehab Plan, Create Rehab Plan, View Report Card, View Patient Log, Patient List, Send Notification) when he decides to log out and clicks on the logout button on the page
2. Once the logout button is clicked, the application terminates user session
3. Redirected to Login Page for user to login

5.3 Notify User

5.3.1 Actor (Notification Server) Interface Flow Diagram

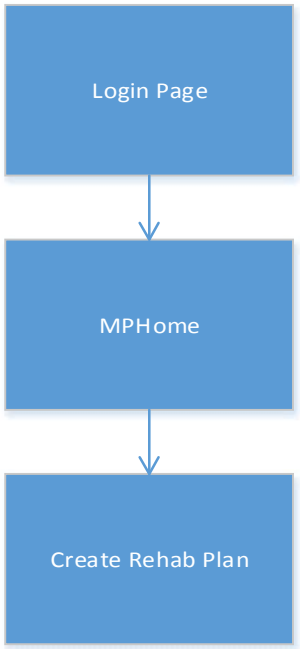


5.3.2 Storyboard

1. The notification server sends the notifications as a result of the Medical Professional facing issues in generating a report, when the Patient enters vitals in the log which exceed the accepted range of values
2. The Medical Professional can login to the application using the Login page
3. After launching the home page, Medical Professional can view all the notifications sent by the notification server on the Notifications Page.

5.4 Generate Rehab Plan

5.4.1 Actor (Medical Professional) Interface Flow Diagram

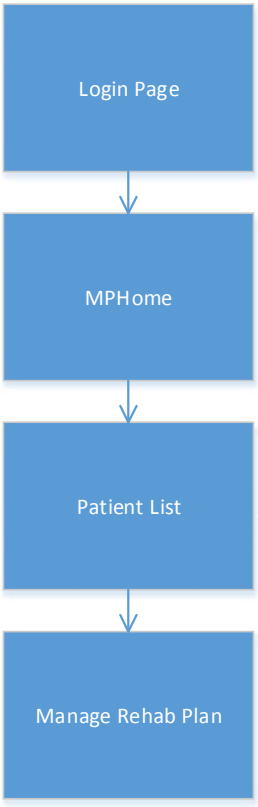


5.4.2 Storyboard

1. Medical Professional completes the “Login” use case from the Login page and lands on the MPHome page.
2. Medical Professional may have a notification on their home page that they have a new patient assigned to them.
3. Medical Professional clicks on “Create Rehab Plan” link on their new patient notification.
4. Medical Professional is redirected to the “Create Rehab Plan” page for the selected patient.
5. A rehab plan template along with the Patient’s Health data will be available to the Medical Professional.
6. Medical Professional reviews and completes the rehab plan by filling out all fields present on the page.
7. If additional exercises are required, Medical Professional can click “Add New Exercise” to be presented with another line of exercise fields.
8. If additional medications are required, Medical Professional can click “Add New Medicine” to be presented with another line of medication fields.
9. Once all fields are filled out, medical professional clicks “Save Plan” to save their changes.

5.5 Manage Rehab Plan

5.5.1 Actor (Medical Professional) Interface Flow Diagram

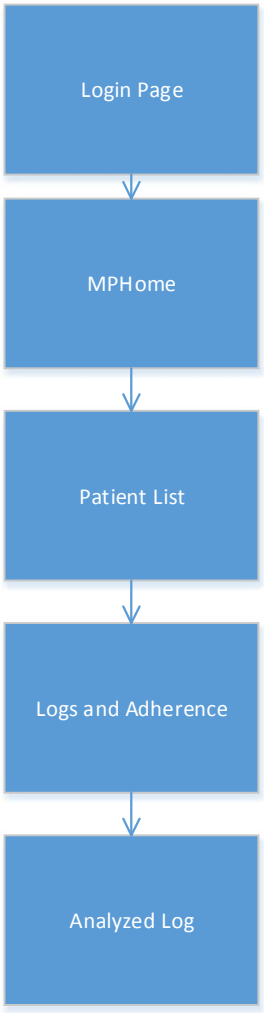


5.5.2 Storyboard

1. Medical Professional completes the “Login” use case and launches the MPHome page
2. Medical Professional clicks on “View Patient List” in the navigation menu.
3. Medical Professional is redirected to the Patient List.
4. Medical Professional finds the patient they are currently working with in the patient list.
5. Medical Professional clicks the “Manage Rehab Plan” link next to their chosen patient.
6. If the Plan does not already exist , the user is taken to the Create Rehab Plan page
7. If the plan already exists Medical professional is redirected to the “Manage Rehab Plan” page which is pre-filled with the existing rehab plan assigned to their chosen patient.
8. Medical Professional reviews and completes the rehab plan by filling out all fields present on the page.
9. If additional exercises are required, Medical Professional can click “Add New Exercise” to be presented with another line of exercise fields.
10. If additional medications are required, Medical Professional can click “Add New Medicine” to be presented with another line of medication fields.
11. Once all fields are filled out, medical professional clicks “Save Plan” to save their changes.

5.6 Analyze Rehab Log

5.6.1 Actor (Medical Professional) Interface Flow Diagram

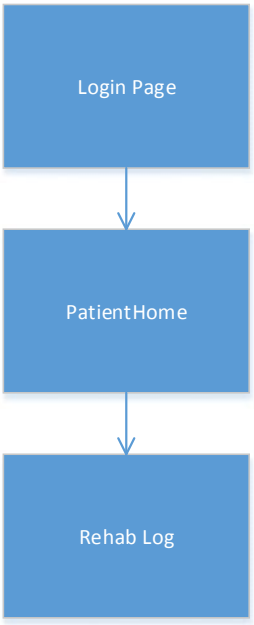


5.6.2 Storyboard

1. Medical Professional completes the “Login” use case and lands on the MPHome page.
2. Medical Professional clicks on “View Patient List” in the navigation menu and is redirected to the Patient List page
3. Medical Professional finds the patient they are currently working with in the patient list.
4. Medical Professional clicks the “Analyze Logs” link next to their chosen patient.
5. Medical Professional is redirected to the “Logs and Adherence” page where he/she reviews the overview of the patient’s information that is presented.
6. If desired, Medical Professional can click on “Update Log” to be presented with a form to fill the analyzed information i.e the Analyzed Log Page

5.7 Manage Rehab Log

5.7.1 Actor (Patient) Interface Flow Diagram

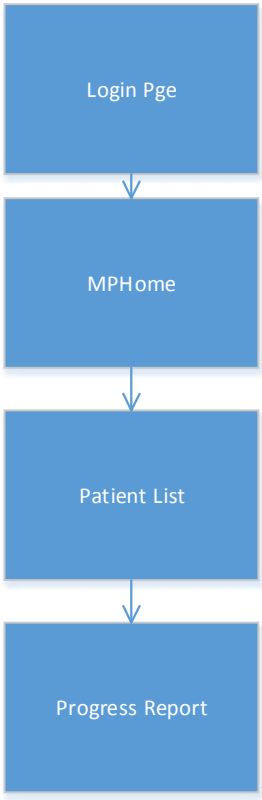


5.7.2 Storyboard

1. Patient completes the “Login” use case and lands on the PatientHome page.
2. Patient clicks on “Enter Daily Log” in the navigation menu
3. Patient is redirected to the “Rehab Log” page.
4. Patient fills out all fields to log their vital signs, diet, exercise, and medication.
5. If additional food items are required, Patient can click “Add New Food” to be presented with another line of dietary fields.
6. Once all fields are filled out, patient clicks “Save Log” to save their changes.
7. Once the log is saved, the application internally checks for any vitals that are not in acceptable range and sends a notification to the medical professional if any such vitals exist.

5.8 Generate Progress Report

5.8.1 Actor (Medical Professional) Interface Flow Diagram



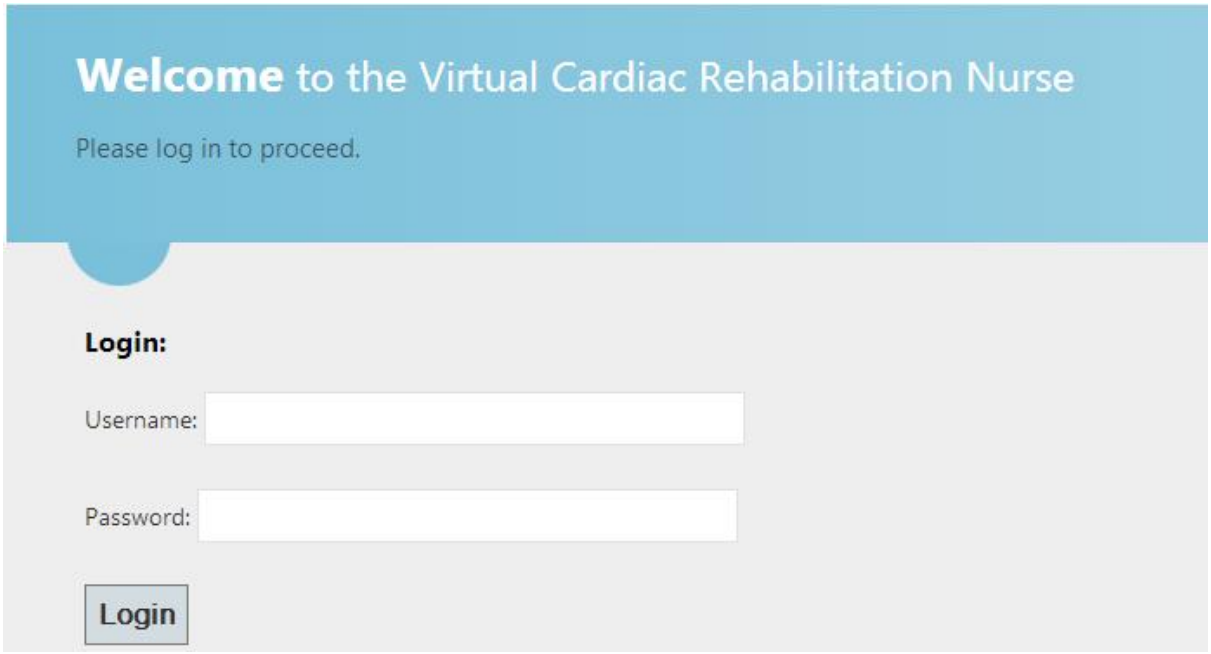
5.8.2 Storyboard

1. Medical Professional completes the “Login” use case.
2. Medical Professional clicks on “View Patient List” in the navigation menu on the MPHome page
3. List of Patients are displayed on the Patient List Page
4. Medical Professional finds the patient they are currently working with in the patient list.
5. Medical Professional clicks the “View Report Card” link next to their chosen patient.
6. Medical Professional is redirected to the “Progress Report” page.
7. Medical Professional reviews the overview of the patient’s status that is presented to them on the Progress Report.

6. Prototype User Interfaces

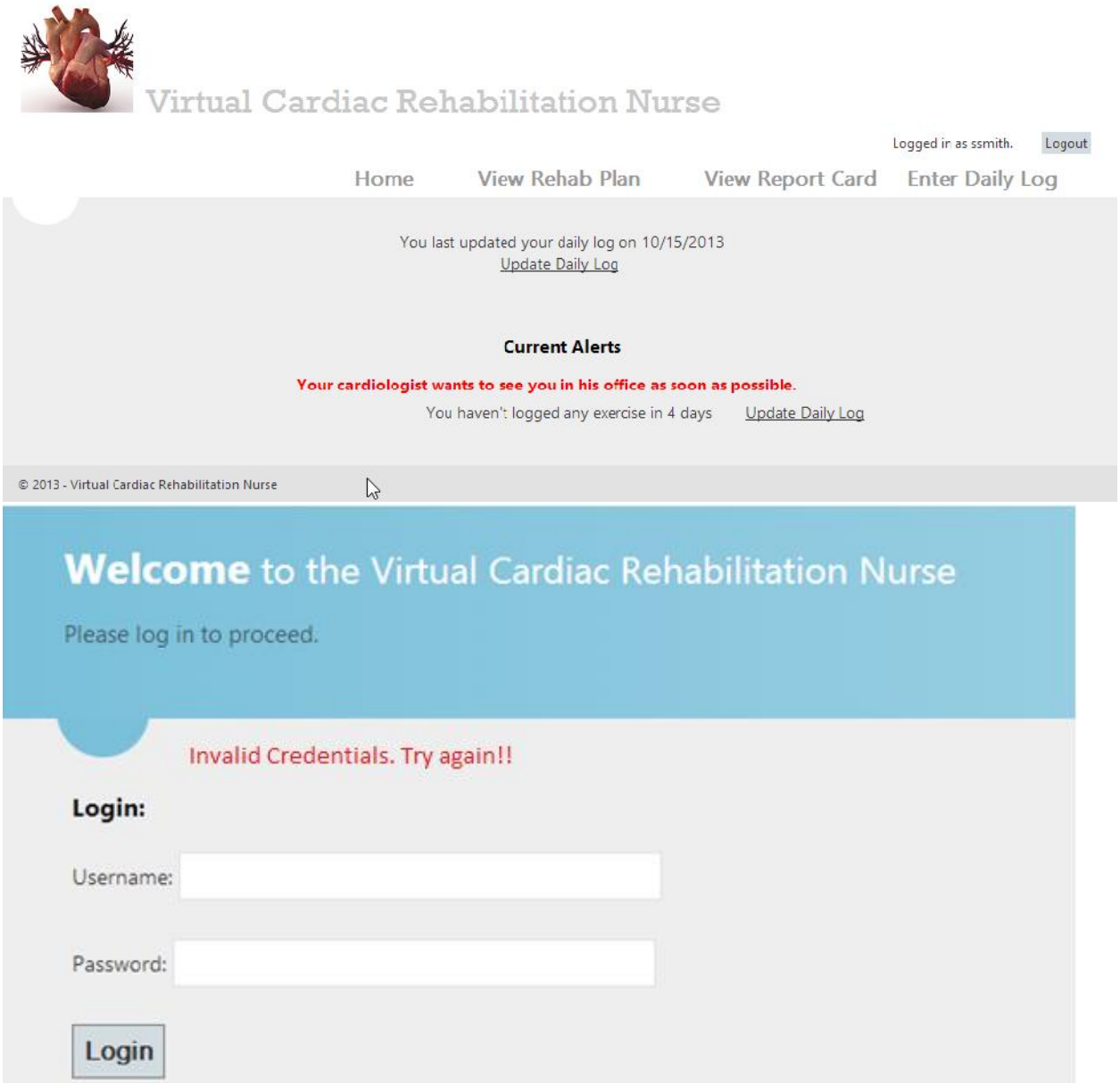
6.1 Login

Login Page



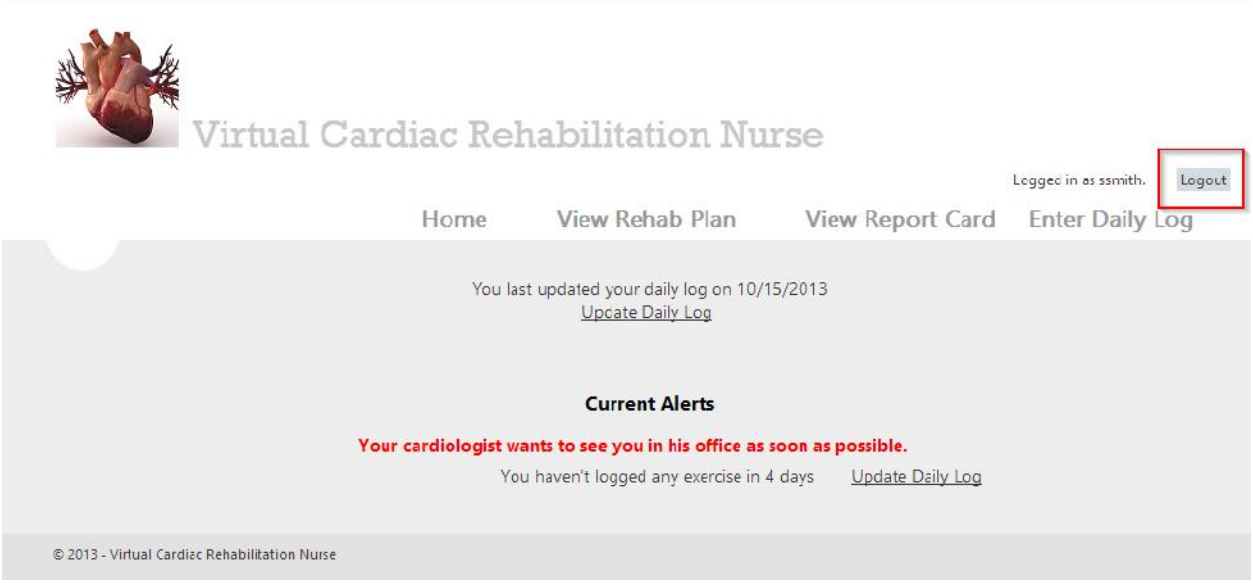
Medical Professional Home Page:

Patient Home Page



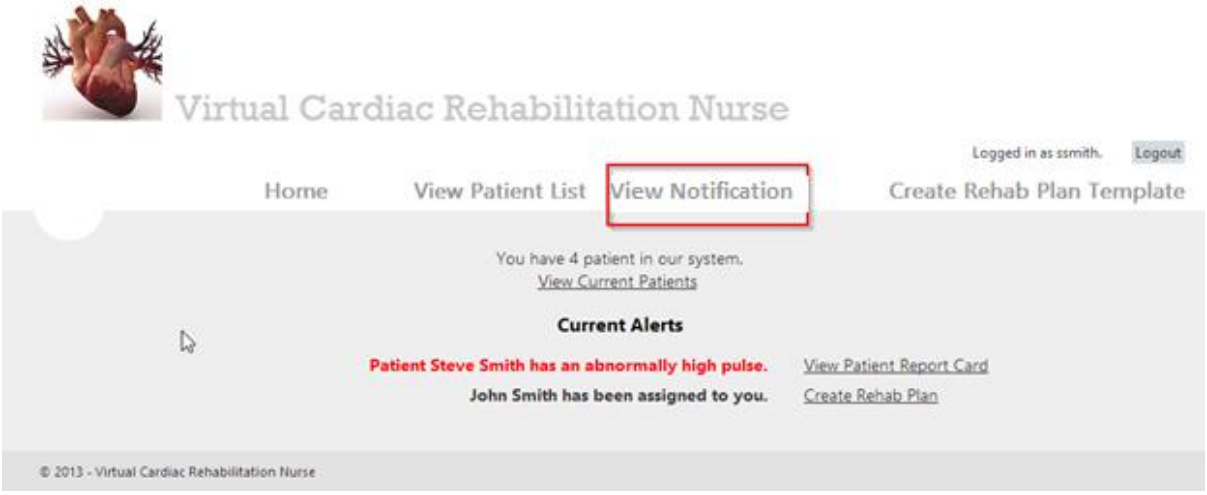
6.2 Logout

1. User clicks the “Logout” link in the upper right corner of the page.



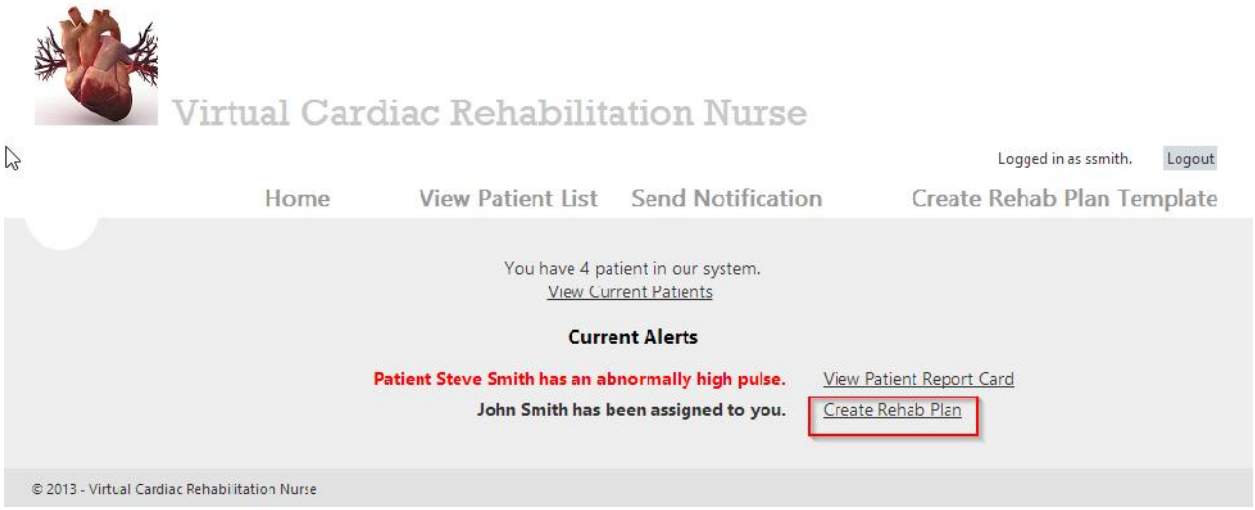
6.3 Notify User

Notifications Page

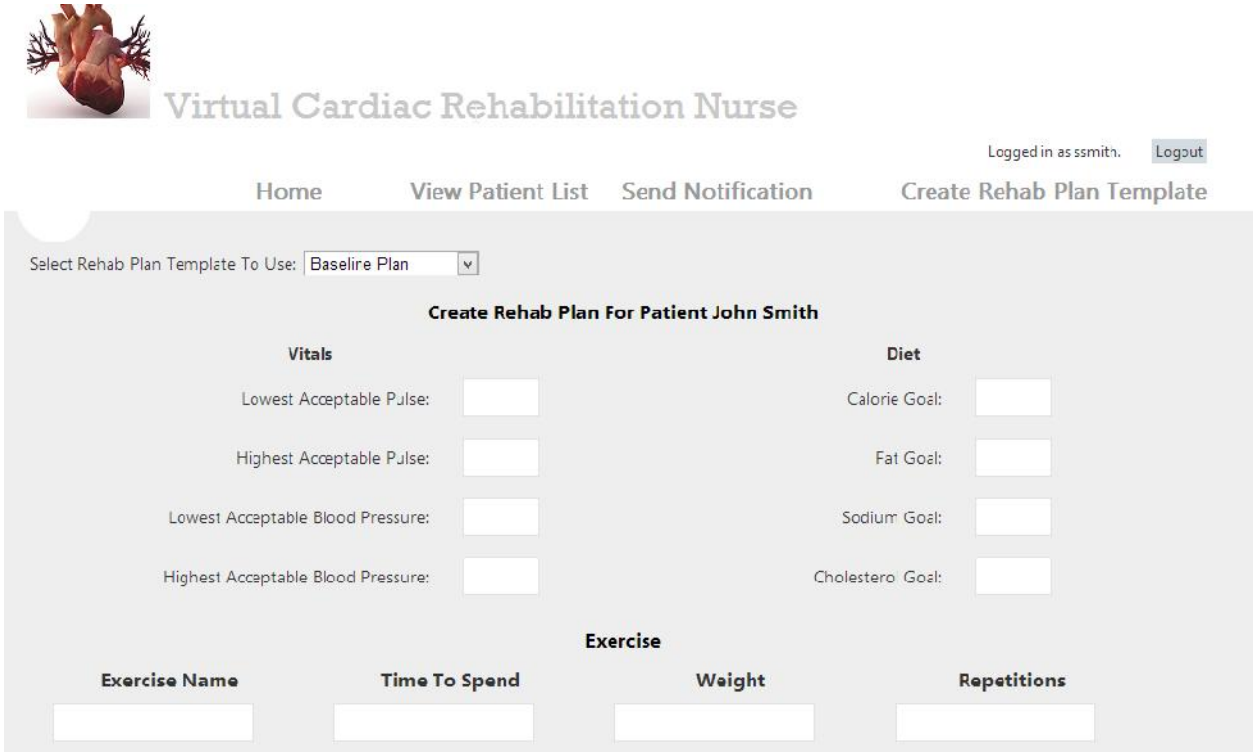


6.4 Generate Rehab Plan

MPHome



Create Rehab Plan page



pre-defined templates

Virtual Cardiac Rehabilitation Nurse

Logged in as ssmith. Logout

Home View Patient List Send Notification Create Rehab Plan Template

Select Rehab Plan Template To Use: Baseline Plan

Baseline Plan
Obese Patient Plan
Elderly Patient Plan

e Rehab Plan For Patient John Smith

Vitals

Lowest Acceptable Pulse:

Highest Acceptable Pulse:

Lowest Acceptable Blood Pressure:

Highest Acceptable Blood Pressure:

Diet

Calorie Goal:

Fat Goal:

Sodium Goal:

Cholesterol Goal:

Exercise

Exercise Name	Time To Spend	Weight	Repetitions

Virtual Cardiac Rehabilitation Nurse

Logged in as ssmith. Logout

Home View Patient List Send Notification Create Rehab Plan Template

Select Rehab Plan Template To Use: Baseline Plan

Create Rehab Plan For Patient John Smith

Vitals

Lowest Acceptable Pulse:

Highest Acceptable Pulse:

Lowest Acceptable Blood Pressure:

Highest Acceptable Blood Pressure:

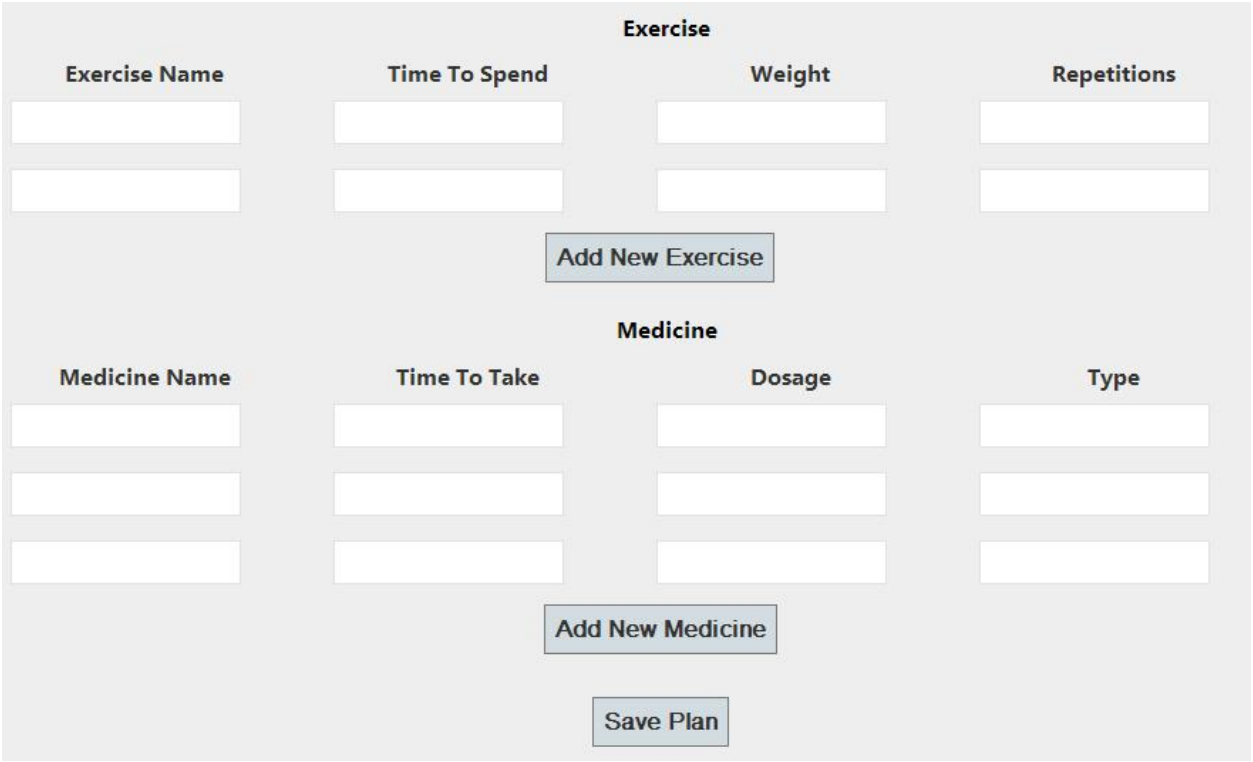
Diet

Calorie Goal:

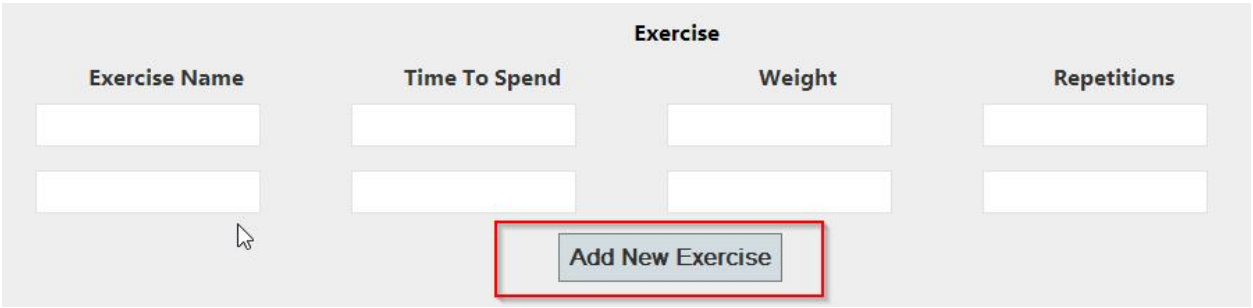
Fat Goal:

Sodium Goal:

Cholesterol Goal:

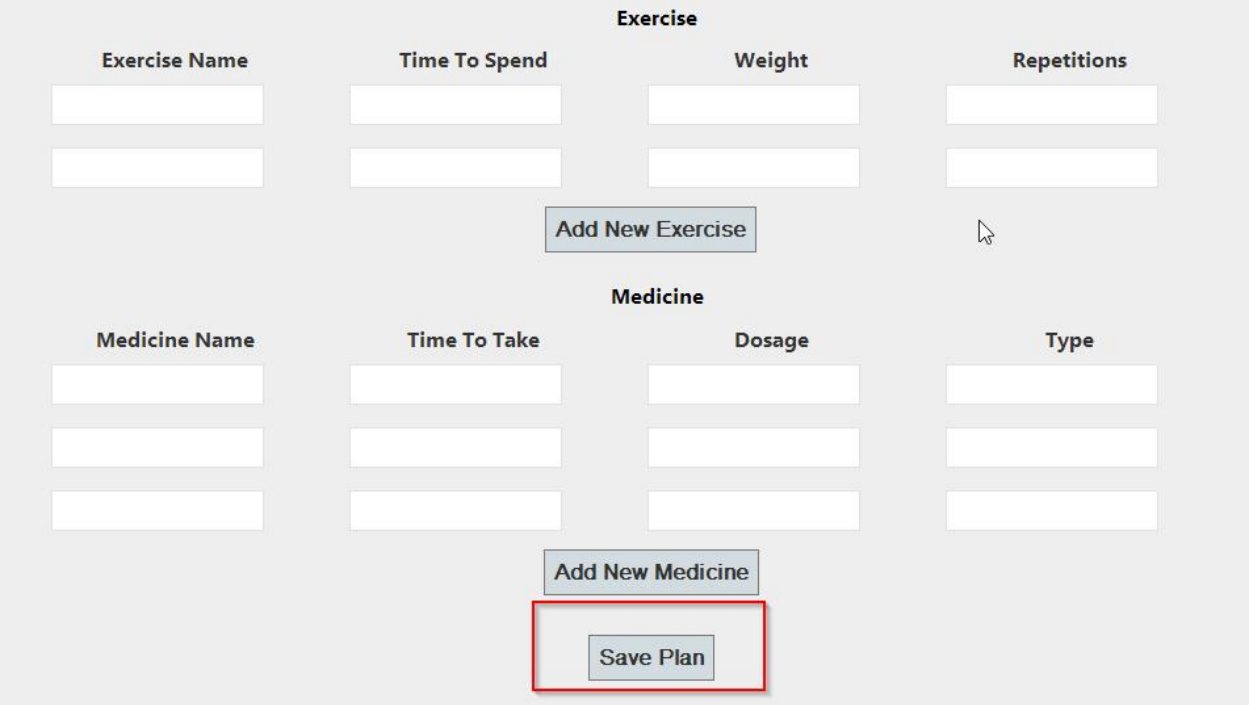


Additional exercises .



Additional medications

Once all fields are filled out, medical professional clicks “Save Plan” to save their changes.

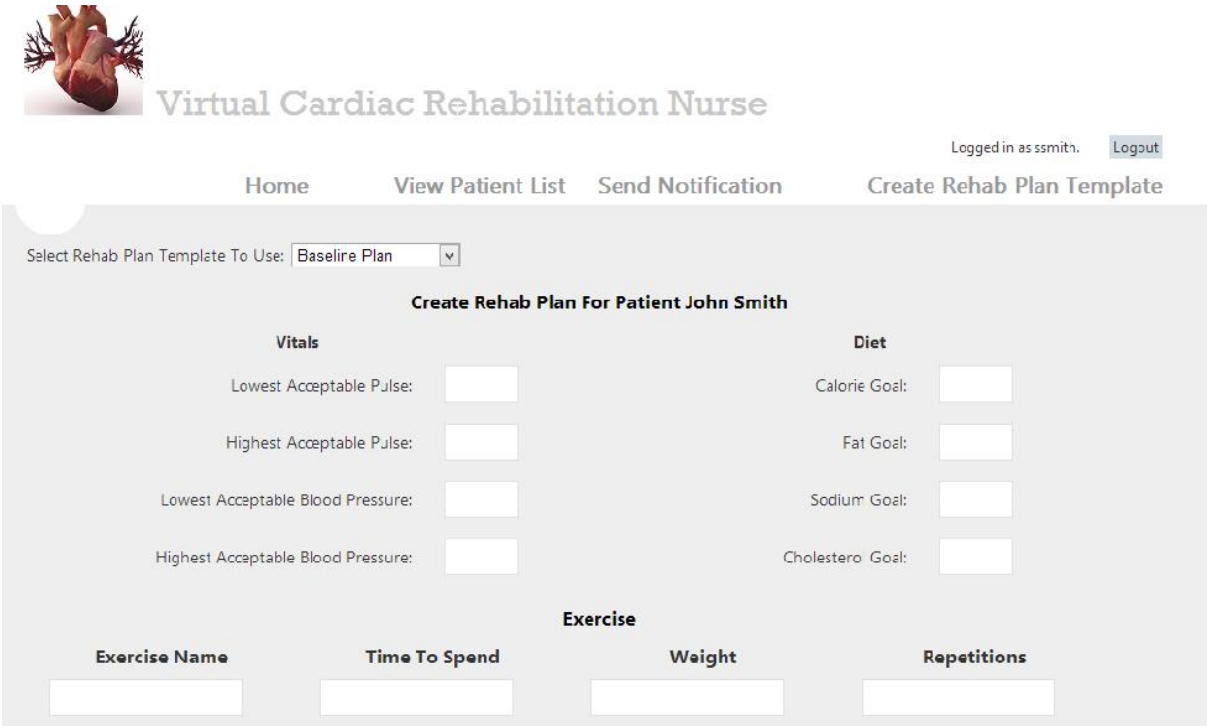


6.5 Manage Rehab Plan

MPHome

Patient List

Create Rehab Plan page



Manage Rehab Plan page

Virtual Cardiac Rehabilitation Nurse

Logged in as ssmith. Logout

Home View Patient List Send Notification Create Rehab Plan Template

Manage Rehab Plan For Patient John Smith

Vitals		Diet	
Lowest Acceptable Pulse:	70	Calorie Goal:	1600
Highest Acceptable Pulse:	100	Fat Goal:	51
Lowest Acceptable Blood Pressure:	85/55	Sodium Goal:	2500
Highest Acceptable Blood Pressure:	160/10	Cholesterol Goal:	200

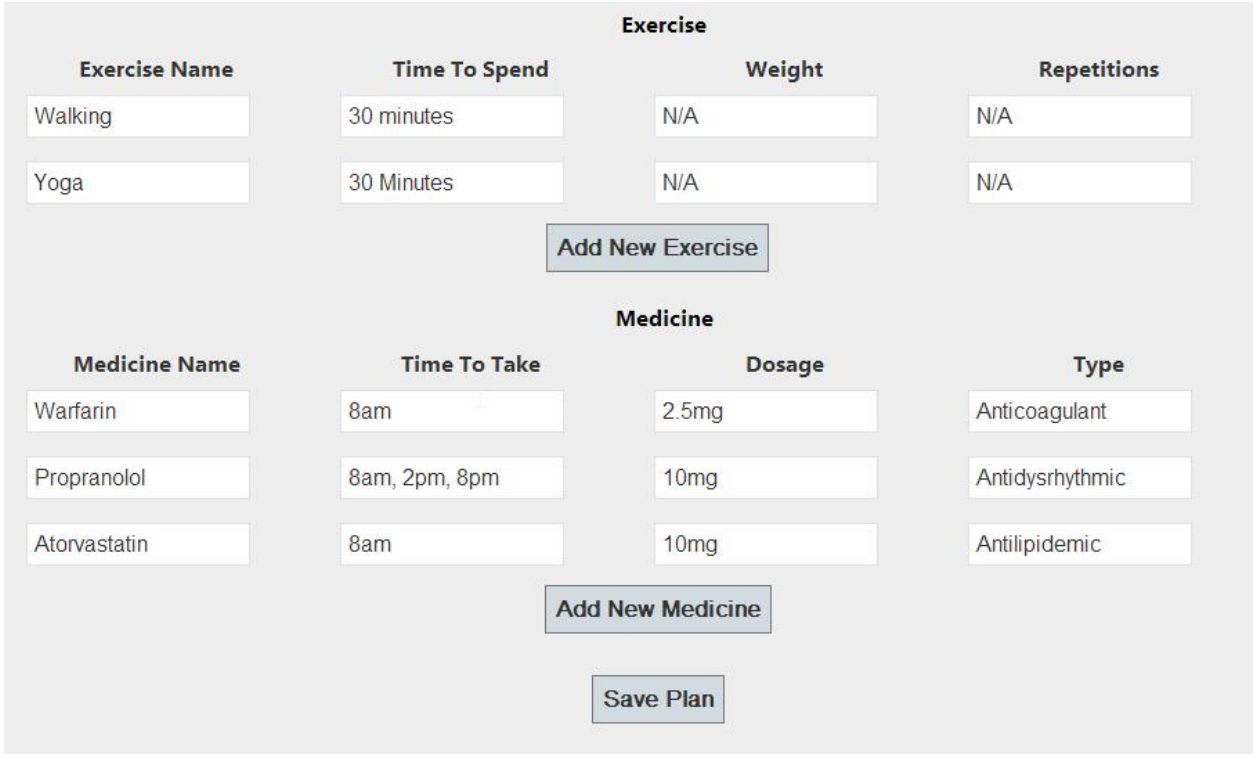
Virtual Cardiac Rehabilitation Nurse

Logged in as ssmith. Logout

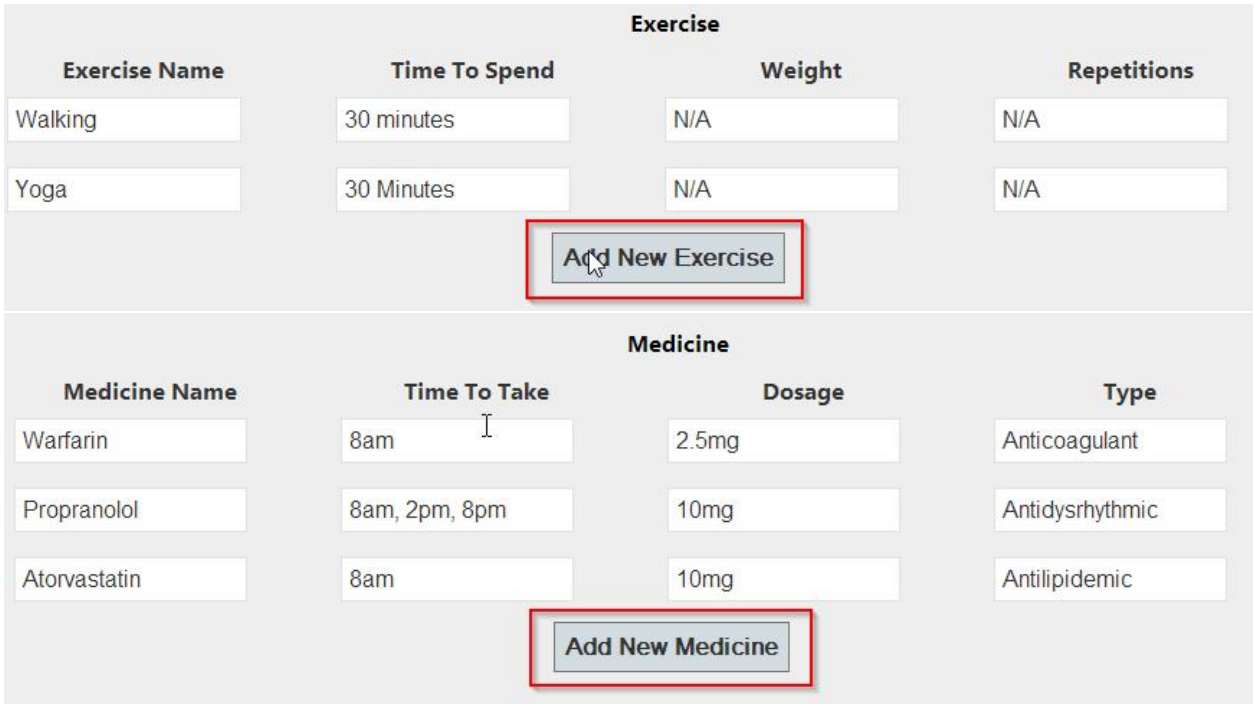
Home View Patient List Send Notification Create Rehab Plan Template

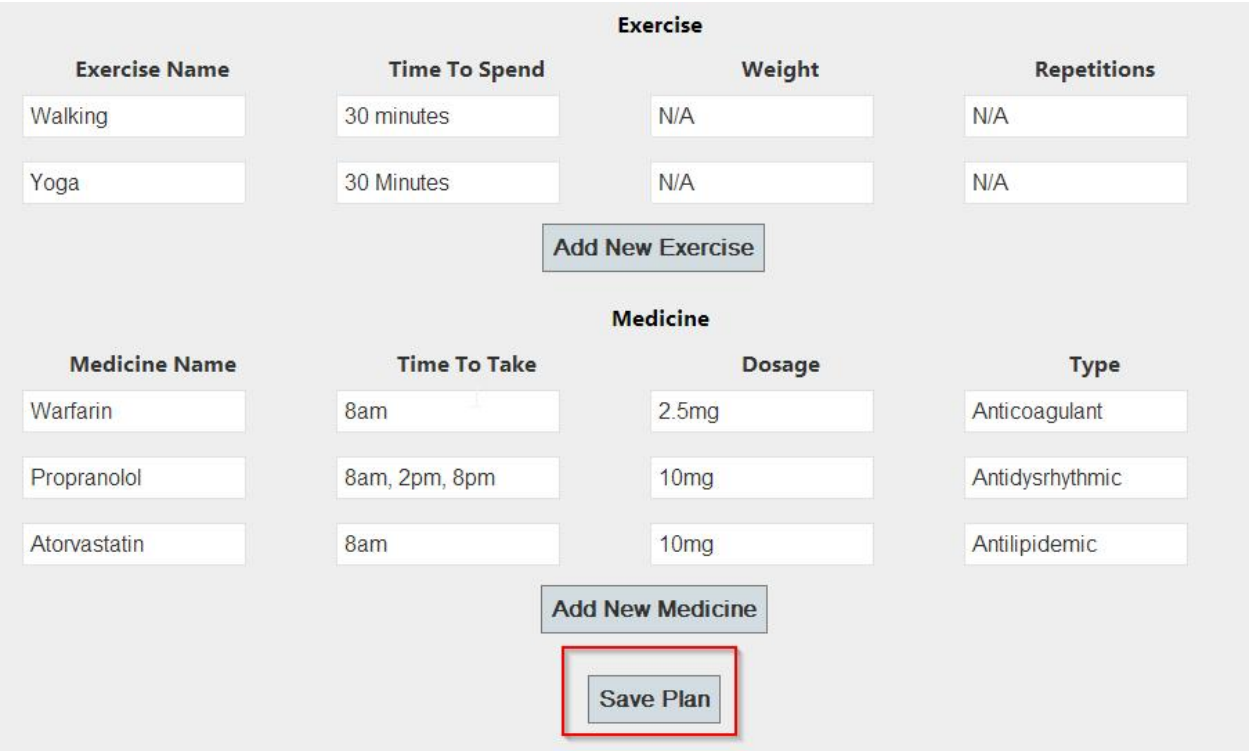
Manage Rehab Plan For Patient John Smith

Vitals		Diet	
Lowest Acceptable Pulse:	70	Calorie Goal:	1600
Highest Acceptable Pulse:	100	Fat Goal:	51
Lowest Acceptable Blood Pressure:	85/55	Sodium Goal:	2500
Highest Acceptable Blood Pressure:	160/10	Cholesterol Goal:	200



Additional Exercises





6.6 Analyze Rehab Log

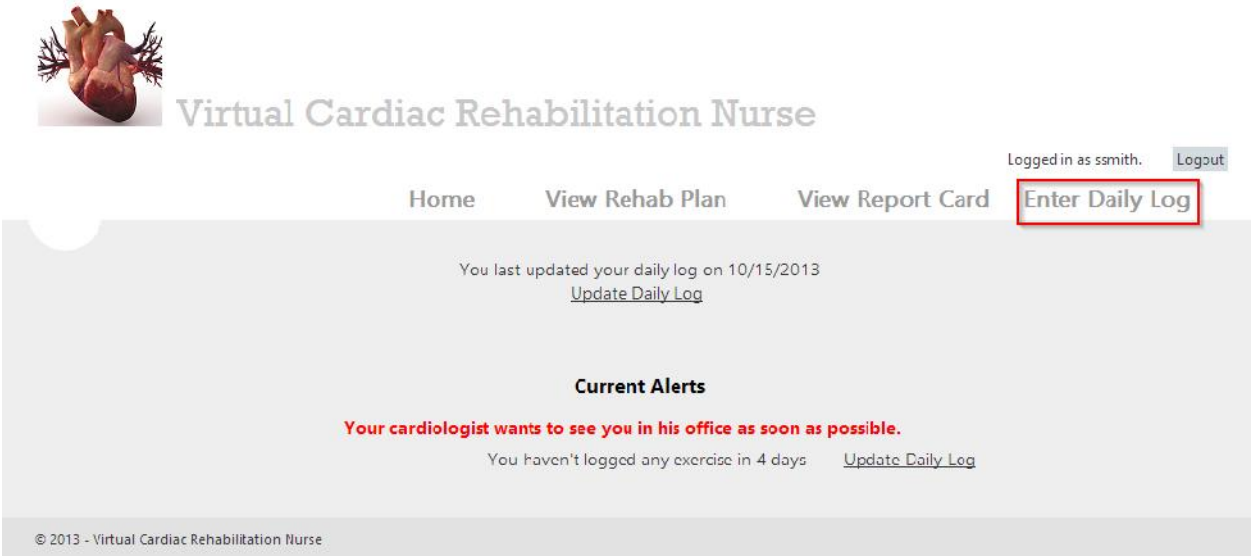
MPHome.

Patient List

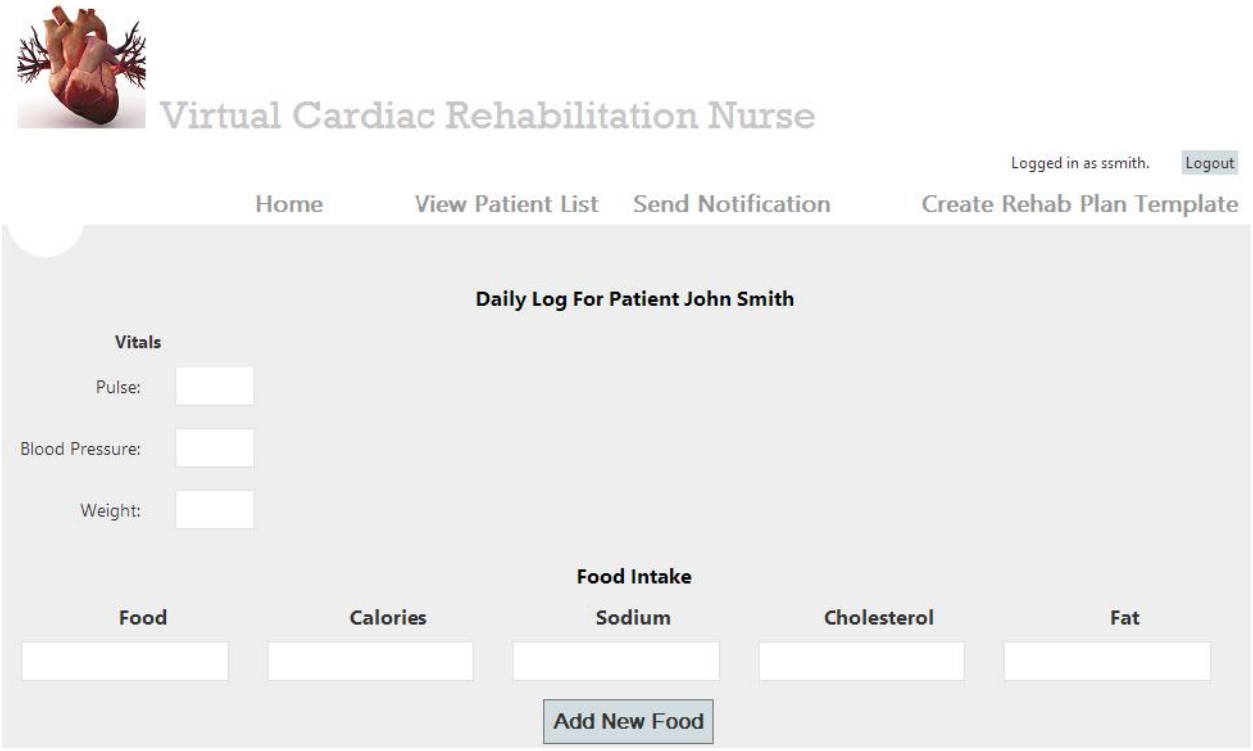
Logs and Adherence Page
Analyzed Log Form

6.7 Manage Rehab Log

Patient Home



Rehab Log" page.



Vital signs, diet, exercise, and medication module of the Rehab Log

Virtual Cardiac Rehabilitation Nurse

Logged in as ssmith. Logout

Home View Patient List Send Notification Create Rehab Plan Template

Daily Log For Patient John Smith

Vitals

Pulse:	70
Blood Pressure:	85/55
Weight:	220

Food Intake

Food	Calories	Sodium	Cholesterol	Fat
Onion Bagel	400	300	300	20

Add New Food

Exercise

Exercise Name	Time To Spend	Weight	Repetitions
Walking	30 minutes	N/A	N/A
Yoga	30 Minutes	N/A	N/A

Medicine

Medicine Name	Type	Time(s) Taken	Dosage
Warfarin	Anticoagulant	8am	2.5mg
Propranolol	Antidysrhythmic	8am, 2pm, 8pm	10mg
Atorvastatin	Antilipidemic	8am	10mg

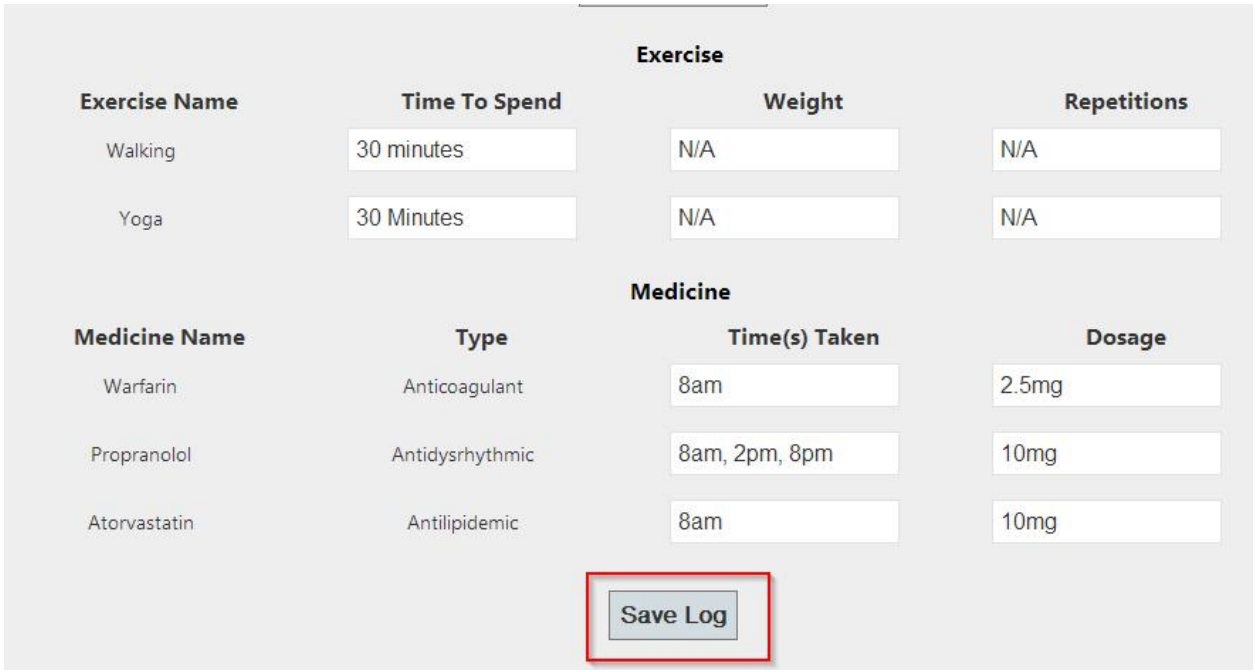
Save Log

Additional food items module of Rehab Log

Food Intake

Food	Calories	Sodium	Cholesterol	Fat
Onion Bagel	400	300	300	20

Add New Food



6.8 Generate Progress Report

MPHome page

Patient List Page

Progress Report Pages

Virtual Cardiac Rehabilitation Nurse

Logged in as ssmith. Logout

Home View Patient List Send Notification Create Rehab Plan Template

Report card for Patient Steve Smith for 10/15/2013
View Full Daily Log

Overall Score: 72.8

Dietary Goals

Calories:	Goal Met
Sodium:	Goal Met
Cholesterol:	Goal Exceeded
Fat:	Goal Met

Exercise Goals

Calories Burned:	Goal Met

Report card for Patient Steve Smith for 10/15/2013
View Full Daily Log

Overall Score: 72.8

Dietary Goals

Calories:	Goal Met
Sodium:	Goal Met
Cholesterol:	Goal Exceeded
Fat:	Goal Met

Exercise Goals

Calories Burned:	Goal Met
Frequency:	Goal Met
Duration:	Goal Exceeded
Variety:	Goal Met

Medication

Type:	Goal Met
Frequency:	Correct
Dosage:	Exceeded

7. Appendices

7.1 Project Status

The “Requirements Specification Part 1& 2” phase is complete. The project is on track and is 34% of the work is accomplished. The latest plan is uploaded as “Virtual Cardiac Rehabilitation Nurse Project Plan v1.5.mpp”

7.2 Analysis Package Diagram Index

7.3 Initial Analysis Package Breakdown by Actor