SOLVERE

Modern Software Requirements Specification

For Self Start System

Version 2.0



Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 7/11/2017 | 1.0 | Initial declaration of use cases | Everyone |
| 14/11/2017 | 1.1 | Further establishment of use cases | Everyone |
| 16/11/2017 | 1.2.1 | 2.1.1, 3 .1 | Aaryn Biro |
| 16/11/2017 | 1.2.2 | 3.1 | Riana McBride |
| 16/11/2017 | 1.2.3 | 3.8-3.13 | Peter Voege |
| 16/11/2017 | 1.2.4 | 1, 2.1.2 | Joshua Evans |
| 16/11/2017 | 1.2.5 | Use Case Diagram (2.1.4), 2.1.3 | Rachel Megidish |
| 16/11/2017 | 1.2.6 | 2.2, 3.2-3.7 | Kevin Freeman |
| 17/11/2017 | 1.3 | Final editing & revisions | Everyone |
| 17/11/2017 | 2.0 | Compiled all individual documents and components together. | Kevin Freeman |

Table of Contents

[Revision History 2](#_Toc498715705)

[Introduction 4](#_Toc498715706)

[Purpose 4](#_Toc498715707)

[Scope 4](#_Toc498715708)

[Definitions, Acronyms and Abbreviations 4](#_Toc498715709)

[References 5](#_Toc498715710)

[Overview 5](#_Toc498715711)

[Overall Description 6](#_Toc498715712)

[Use-Case Model Survey 6](#_Toc498715713)

[Introduction 6](#_Toc498715714)

[Survey Description 6](#_Toc498715715)

[Use-Case Model Hierarchy 6](#_Toc498715716)

[Diagrams of the Use-Case Model 10](#_Toc498715717)

[Assumptions and Dependencies 11](#_Toc498715718)

[Requirements 12](#_Toc498715719)

[Use-Case Specifications 12](#_Toc498715720)

[Functionality 25](#_Toc498715721)

[Usability 26](#_Toc498715722)

[Reliability 26](#_Toc498715723)

[Performance 27](#_Toc498715724)

[Supportability 27](#_Toc498715725)

[Design Constraints 27](#_Toc498715726)

[Online User Documentation and Help System Requirements 28](#_Toc498715727)

[Purchased Components 28](#_Toc498715728)

[Interfaces 28](#_Toc498715729)

[User Interfaces 28](#_Toc498715730)

[Hardware Interfaces 28](#_Toc498715731)

[Software Interfaces 28](#_Toc498715732)

[Communications Interfaces 28](#_Toc498715733)

[Licensing Requirements 28](#_Toc498715734)

[Legal, Copyright and Other Notices 28](#_Toc498715735)

[Applicable Standards 29](#_Toc498715736)

**Modern Software Requirements Specification**

Introduction

Purpose

The purpose behind the documentation of this Modern SRS is to properly gain an understanding of the requirements and functionalities of the Self Start system, which is a web application that allows physiotherapists to communicate with their patients. Currently, patients must meet with their physiotherapist to discuss their problems, and the physiotherapist must explain to them the exercises for remedy. This web application would allow the physiotherapists to diagnose patients by having the patients fill out forms regarding their symptoms, and sending in pictures of their posture. Physiotherapists would then be able to create custom exercise plans for the patients to follow, and report the plan’s effectiveness back to the physiotherapist.

Scope

Self Start will be a web application that contains web pages, as well as a database. The database contains all the custom exercise plans that can be given to a patient, as well as a patient’s treatment history, and general account information such as passwords and phone numbers.

The physiotherapist can query the database, and the patients are able to fill out forms and view their exercise plans through a simple UI. The effectiveness of the application will be determined by the ease of use by both the physiotherapists and the patients.

Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| Term | Definition |
| Exercise plan | A set of exercises given to a patient to help treat their physical issues. |
| Assessment test | Feedback from the patient to the physiotherapist, highlighting the progress of the patient and their thoughts on the exercises. |
| Database | Software solution to store information. The Self Start system will use a database to hold all information for patients, physiotherapists, and the admin. |
| Modern SRS | Describes the external behavior of the system, creating a conceptual model of the system to be built. |
| UI | A user interface allows users to interact with a system through input devices. |
| Web application | Client-server computer program that runs the client in a web browser. |

References

**Title**: Assignment 1: Developing the Software Requirements Specifications SRS

**Reference**: Ouda, A. “Assignment 1: Developing the Software Requirements Specifications SRS.” SE 3352 Software Requirements and Analysis. Western University, 8 November 2017. Web. 17 November 2017.

<<https://owl.uwo.ca/access/content/attachment/672f3446-c1d2-46a8-9426-90707ad34952/Assignments/8ebc5171-3538-47ba-adeb-85d4791f515c/SE3352a_assignment1%20_2017_.pdf>>.

Overview

The interactions between the patients, physiotherapists, and admin within the Self Start system are illustrated in section 2, where a use-case model allows anyone to understand what the system will be achieving.

The requirements of the system are described in section 3, and can be understood by technically able personnel. Use-case specifications provide more detail than the use-case diagram, and breakdown each use-case into its unique flow of events.

Overall Description

Use-Case Model Survey

Introduction

The main functionality of the Self-Start application involves creating a way for physiotherapists of Marcotte Physiotherapy to communicate and lend their services to clients digitally. With that in mind, the following use cases reveal a finer level of detail in terms of how the application will function to achieve this goal. These use cases will outline exactly *how* each type of user will interact with our system. By breaking down all the essential interactions between actors and the various components of the software that are needed for full functionality, one knows the purpose, function and requirements of each component and how they will all work together in the larger scale of the project. The attention given to these use cases is vital to gain insight into the flow and design of the proposed software, as they have a large impact on the final user experience and are essentially a blueprint for the Self-Start application’s design goals moving forward.

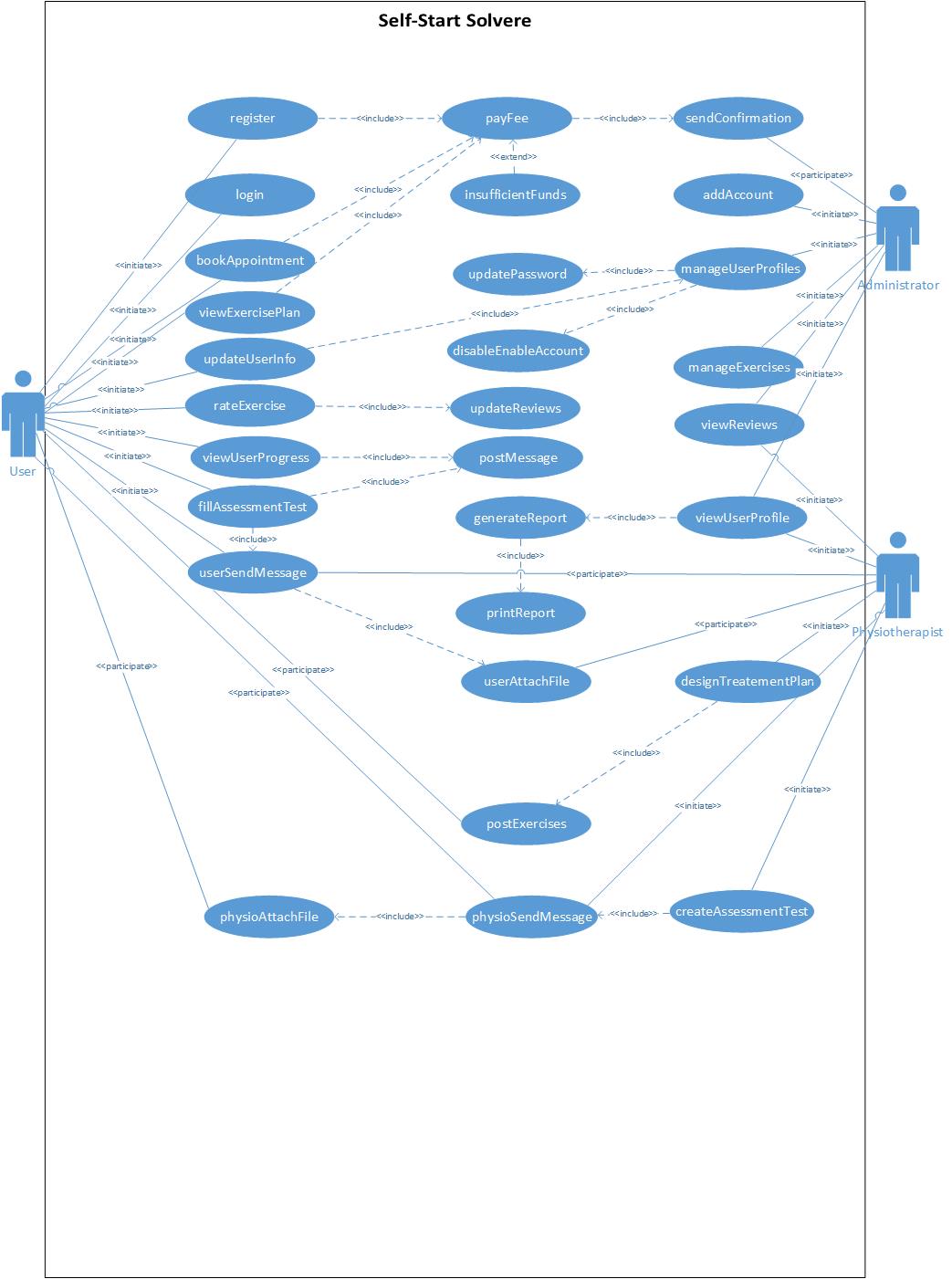
Survey Description

The Self Start system is a web application that allows physiotherapists and their patients to be connected, and share information about exercise plans and their assessment tests back and forth. The system is controlled by the admin, and allows physiotherapists and patients to update the state of the system through their interactions with one another. Users are authenticated upon logging in to assign distinct privileges to the type of user logged in, such as the admin, physiotherapist, and patient.

Use-Case Model Hierarchy

|  |  |
| --- | --- |
| **Actors** | **Description** |
| **Administrator** | The administrator is responsible for operation maintenance, and managing the system accounts, features and physiotherapy exercises. The administrator oversees payments and system reviews, and ensures high quality standards are upheld. |
| **Physiotherapist** | The physiotherapist is in charge of providing treatment plans and exercises to users. The physiotherapist can communicate with the user to clarify treatments and provide assistance, as well as gauge user progress. |
| **User** | The user is a patient who will use the system in order to treat physiological issues by receiving exercise plans from the physiotherapist. The user can communicate with the physiotherapist for assistance and in order to hone treatment effectiveness. |

|  |  |
| --- | --- |
| **Use-Cases** | **Description** |
| **Register** | The system will allow a user to register for an account by filling a registration form. This includes Pay Fee, where the user will make a payment to finalize the registration. |
| **Login** | A user who has already registered can securely log in to their account using their credentials. |
| **Book Appointment** | This allows the system to book an appointment to come in person to the physiotherapist at her practice. This includes pay fee as the user will be charged for the appointment. |
| **View Exercise Plan** | This allows the user to view his treatment plan of physiotherapy exercises given by the physiotherapist. This will include Pay Fee depending on whether the user has enough credits to receive more the treatment plan. |
| **Pay Fee** | Pay Fee allows the user to submit a payment to the system upon register in order to receive credits that can be used toward treatments. This is including from Book Appointment and View Exercise Plan as the user may need to pay again for an appointment, or receiving extra treatments. This extends to Insufficient Funds, a case where the online payment is unsuccessful due to low funds. |
| **Insufficient Funds** | This is extended from Pay Fee. If the user does not have sufficient funds to complete the purchase, the system will notify the user that the payment has not been able to process. |
| **Send Confirmation** | The system will send confirmations to the user and administrator that the payment has been received, along with the necessary information for the appointment or balance. This participates with the administrator and is saved to the system database. |
| **Add Account** | The administrator is able to add an account to the system. This can be in the form of a user or a physiotherapist. |
| **Update User Info** | The user is able to update their information. This covers personal and payment information. This is including to Manage User Profiles where the administrator will approve these changes and save them to the system database. |
| **Manage User Profiles** | The administrator is responsible for managing user profiles. This is included from Update User Info. The user will submit a form with the information to be updated, and the administrator will approve the changes for the database. This will include Update Password and Disable/Enable Account, as the administrator is able to change user passwords, and disable/enable accounts. |
| **Update Password** | This is included from Manage User Profiles. The administrator is able to change passwords per user request. |
| **Disable/Enable Account** | This is included from Manage User Profiles. The administrator is able to disable and enable accounts per user request. |
| **Manage Exercises** | The administrator is responsible for managing the exercises. The system allows the admin to be able add, remove and update exercise videos in the database. |
| **Rate Exercise** | The user can rate an exercise by filling a form. It can be rated on difficulty, pain, range of motion and general experience. This includes Update Reviews as the system will automatically save and update the reviews to the database. |
| **Update Reviews** | The system updates the reviews after a user rates an exercise. The physiotherapist and administrator may view the reviews in order to assess the users reactions to each exercise and hone specific treatment plans. |
| **View Reviews** | Both the administrator and the physiotherapist are able to access the exercise reviews made by the user. |
| **View User Profile** | Both the administrator and the physiotherapist are able to access any user’s profile and view their progress. This includes to Generate Report. |
| **Generate Report** | The administrator or physiotherapist is able to generate a report after viewing a user’s progress and profile. This includes the ability to Print Report. |
| **Print Report** | The system allows the report generated by the administrator or physiotherapist to be printed. |
| **Design Treatment Plan** | The physiotherapist is able to design a treatment plan for the user. This plan will consist of exercises, videos and descriptions. It includes Post Exercises, as the exercises will be posted directly on the user’s profile for the user to view. |
| **Post Exercises** | When the physiotherapist has completed designing a treatment plan for a user, the exercises are posted onto the user profile for the user to view. |
| **Create Assessment Test** | System allows physiotherapist to create a test which will assess the user’s progress. The user can rate their pain, range of motion and progress on this form. This includes to Physio Send Message. The system will send the test to the user. |
| **Physio Send Message** | The system allows the physiotherapist to send a message to the user. This can be in the form of an assessment test, or a general message. This includes attaching a file. It is participated with the user. |
| **Physio Attach File** | The system allows the physiotherapist to attach a video with the message to the user. This is participated with the user. The attachment can be in the form of a video to help explain the exercise. |
| **View User Progress** | The system will allow the user to be able to view their progress on their profile since they have been working with Self-Serve. This will include to Post Message, as the user will be able to post a message onto their profile to log their pain levels or progression at any specific time. |
| **Post Message** | When the user views their progress they are able to post a message to their profile. After a user completes an assessment test, it is also posted on to their profile and saved for future viewing. |
| **Fill Assessment Test** | The user will be able to fill an assessment test form given by the physiotherapist in order to gauge their progression. The assessment test will include User Send Message, as it will be sent to the physiotherapist to determine how the therapy will proceed. It will also include Post Message, as it will automatically save to the database and post onto the user’s profile for the user and physiotherapist to view at any time. |
| **User Send Message** | The system will allow the user to send a message to the physiotherapist. The message can be in the form of an inquiry or an assessment test. This includes to User Attach File as the user can also choose to attach a video demonstrating the exercise for the physiotherapist to critique. This is participated with the physiotherapist. |
| **User Attach File** | The system provides the user with the ability to attach a file with their message. The user can send a video demonstrating the exercise for the physiotherapist to ensure it is done correctly for best results. This is participated with the physiotherapist. |

Diagrams of the Use-Case Model

Assumptions and Dependencies

During the design phase of the project, one key assumption was made in attempt to simplify and retain the proper defined scope for the application's development. This assumption relates to the payment functionality for the application: it is assumed that the payment procedure will be handled by a third party, such as Paypal, or directly through a bank. It is important to note that this assumption is solely in relation to payment processing, and does not extend to payment confirmation delivery, nor accessing features on the application that are locked behind payment. Beyond this assumption, there were no other assumptions made for this project that could affect the viability of the project.

Requirements

Use-Case Specifications

|  |  |
| --- | --- |
| Use Case Name | Login |
| Participating Actors | Initiated by the User |
| Entry Condition | User’s username and password must be entered |
| Flow of Events | 1. The User must activate the log on function of the website. 2. Self-Start will encrypt the entered username and password and search through the database of customers to find the encrypted match. 3. Self-Start determines whether to let the user log on or to deny the request to log on because of a wrong username and/or password. 4. User will either receive a notification that their username/password is wrong or will be logged onto their account. |
| Exit Condition | The User must either be validated or denied. |
| Quality Requirements |  |

|  |  |
| --- | --- |
| Use Case Name | Register |
| Participating Actors | Initiated by the User |
| Entry Condition | User’s new username and password must be set |
| Flow of Events | 1. The User must activate the Register function of the website. 2. Self-Start will encrypt the entered username and password and post the new user with their username and password to the database. 3. Self-Start will request the user to pay for a program. The payFee use case is included here. 4. User will either receive a notification that their username/password has been created. 5. User, Physiotherapist, and Admin will be notified of the new account and a Physiotherapist will be assigned to the user. |
| Exit Condition | The User must either be validated or denied. |
| Quality Requirements | Physiotherapist, User, and Admin must be notified immediately that the account has been created. |

|  |  |
| --- | --- |
| Use Case Name | updateUserInfo |
| Participating Actors | Initiated by the User, participates with the Admin |
| Entry Condition | User must be logged in, and have their account paid for |
| Flow of Events | 1. The User must activate the Update Info function of the website. 2. Self-Start will direct the user to the User Info page. 3. User can then update their personal info, their payment information, or their password. 4. User will then save their changed information. 5. Self-Start will update the User Info in the system, and send a notification to the Admin to notify them of the change. |
| Exit Condition | -The User saves the User Info  -The User exits the User Info page |
| Quality Requirements | -the Admin must be sent a notification immediately  -the User info must be updated in the system |

|  |  |
| --- | --- |
| Use Case Name | bookAppointment |
| Participating Actors | Initiated by the User, participates with the Admin and Physiotherapist |
| Entry Condition | User must be logged in, and have their account paid for |
| Flow of Events | 1. The User must activate the Book Appointment function of the website. 2. Self-Start will direct the user to the Book Appointment page. 3. User will select a month, day, and time from the available list of appointments. 4. User will receive a notification that their appointment has been booked. 5. Self-Start will update the calendar to make sure that no other users can book an appointment during that time. |
| Exit Condition | -The User books an appointment  -The User exits the Book Appointment page  -The sendConfirmation use case is included, to then send the notification to the Physiotherapist and admin. |
| Quality Requirements | -the user must only be allowed to select available dates and times for their appointment. |

|  |  |
| --- | --- |
| Use Case Name | payFee |
| Participating Actors | Initiated by the User |
| Entry Condition | -user must be logged in  -user payment information must be valid |
| Flow of Events | 1. User activates the “Purchase Plan” function of the website. 2. Self-Start responds by presenting a form asking for the billing information as well as the shipping information. 3. Self-Start will prompt the user to confirm their order. Once the order is confirmed the user will pay the fee. 4. User must fill in information and accept the payment. 5. Self-Start will verify the billing information as valid. 6. The sendConfirmation use case is included here. Self-Start will use sendConfirmation to send an email to the Customer confirming the order. 7. Self-Start will save the order information to the database. |
| Exit Condition | -Customer cancels the order  -Billing information cannot be validated  -The order is completed |
| Quality Requirements | -The website should maintain the data of the transaction |

|  |  |
| --- | --- |
| Use Case Name | sendConfirmation |
| Participating Actors | Participates with Admin, User, and Physiotherapist |
| Entry Condition | -user must be logged in  -payment method must be validated  -payment must be completed |
| Flow of Events | 1. The Self-Start software will take the payment information and store it in the database. 2. Self-Start will send an email to the User to confirm their payment and thank them. 3. Self-Start will send an email to the assigned physiotherapist to notify them of their new patient, and send them the patient details after querying the database. 4. Self-Start will send an email to the Admin to let them know that there is a new client added to the system. |
| Exit Condition | -Self-Start has emailed User, Admin, and Physiotherapist. |
| Quality Requirements | Physiotherapist, User, and Admin must be notified immediately that the payment has been made. |

|  |  |
| --- | --- |
| Use Case Name | viewExercisePlan |
| Participating Actors | Initiated by User |
| Entry Condition | -user must be logged in |
| Flow of Events | 1. User must activate the View Exercises function of the software. 2. Self-Start will locate all the exercises that the Physiotherapist has assigned to the User, and display them in a list form on the page. 3. User can then select an exercise to view the video corresponding to that exercise. 4. Self-Start will display the video that corresponds to the exercise, for the User to view. |
| Exit Condition | -User exits the View Exercises page. |
| Quality Requirements | -Each exercise must have a video attached.  -Only exercises that are part of the User’s treatment plan must be displayed.  -The videos should be made of avataars  -While User is viewing an exercise, at any point in time rateExercises or updateRating use cases may be included, being initiated by the User. |

|  |  |
| --- | --- |
| Use Case Name | rateExercise |
| Participating Actors | Initiated by User |
| Entry Condition | -user must be logged in |
| Flow of Events | 1. User must activate the View Exercises function of the software. 2. Self-Start will locate all the exercises that the Physiotherapist has assigned to the User, and display them in a list form on the page. 3. User must then select an exercise to view the video corresponding to that exercise. 4. Self-Start will display the video that corresponds to the exercise, for the User to view. 5. Below the video for the exercise, there will be a rating section for the exercise that will prompt the user to enter their opinion of the exercise on a scale of 1-10. 6. Self-Start will POST their rating to the database. |
| Exit Condition | -User exits the View Exercises page.  -User rates the Exercise. |
| Quality Requirements | -The 1-10 scale must be made of smiley faces.  -User can only rate each exercise once, otherwise they must select the included updateReviews use case to change their review. |

|  |  |
| --- | --- |
| Use Case Name | updateReviews |
| Participating Actors | Initiated by User |
| Entry Condition | -user must be logged in  -the exercise must already be rated by the User |
| Flow of Events | 1. User must select the Change Rating function under a specific exercise. 2. Self-Start will display the scale from 1-10 for the user. 3. The User will then be prompted to select their new rating of the exercise, on a scale of 1-10. 4. Self-Start will PUT the changed rating on the database. |
| Exit Condition | -User exits the View Exercises page.  -User re-rates the Exercise. |
| Quality Requirements | -The 1-10 scale must be made of smiley faces. |

|  |  |
| --- | --- |
| Use Case Name | viewUserProgress |
| Participating Actors | Initiated by User |
| Entry Condition | -user must be logged in |
| Flow of Events | 1. User must activate the Progress function of the software. 2. Self-Start will query the database for the User’s reviews and past assessment tests. 3. Self-Start will chart all the exercise reviews that User has filled out, and based on their assessment tests and these reviews will display charts of the user’s progress in the treatment plan. 4. User can then track their progress. |
| Exit Condition | -User exits the Progress page. |
| Quality Requirements | -Progress page will update each time the user fills out a review or completes an assessment test |

|  |  |
| --- | --- |
| Use Case Name | postMessage |
| Participating Actors | Initiated by User |
| Entry Condition | -user must be logged in |
| Flow of Events | 1. User may have filled out assessment test and clicked submit or must activate the View Progress function of the software. 2. If included by the fillAssessmentTest use case, Self-Start will take the information from the assessment test and update the User Progress charts, and will post the assessment test to the User’s account. 3. If included by the viewUserProgress use case, the user must then activate the post message function. 4. User must enter a message, and can optionally enter a subject. 5. User must click “Post” to post their message to that page. 6. Self-Start will save this message so that the User can look back at their post at any time. |
| Exit Condition | -User exits the Post Message page.  -Self-Start saves the message. |
| Quality Requirements | -Self-Start must guarantee that the message will be saved, so that the post can be viewed at any time. |

|  |  |
| --- | --- |
| Use Case Name | userSendMessage |
| Participating Actors | Initiated by User, participates with Physiotherapist |
| Entry Condition | -user must be logged in |
| Flow of Events | 1. User must activate the Contact Physiotherapist function of the software. 2. Self-Start will bring the user to the message page, which automatically has the User’s Physiotherapist set as the recipient, and the User set as the sender. 3. User must enter a message, and can optionally enter a subject. 4. User must click send to send the message to their Physiotherapist. 5. Self-Start will deliver the message to the Physiotherapist’s inbox, indicating who sent the message. |
| Exit Condition | -User exits the Contact Physiotherapist page.  -User sends the message. |
| Quality Requirements | -Self-Start must guarantee that the message will reach the Physiotherapist  -The message function must automatically know which Physiotherapist to send to  -The user can initiate the userAttachFile use case before they send the message, which is included as part of this use case. |

|  |  |
| --- | --- |
| Use Case Name | fillAssessmentTest |
| Participating Actors | Initiated by User, participates with Physiotherapist |
| Entry Condition | -user must be logged in  -assessment test must be sent to the User from the Physiotherapist |
| Flow of Events | 1. User must activate the Contact Physiotherapist function of the software. 2. User will activate the Fill Assessment Test function. 3. Self-Start will load the Assessment test that has been provided by the Physiotherapist. 4. User will fill out the Assessment test, and then submit the form. 5. The postMessage use case is included here. The Assessment test will then be posted to the user profile so that the user progress can be updated. 6. Self-Start will then send the submitted Assessment test to the Physiotherapist’s inbox. |
| Exit Condition | -User exits the Assessment Test page.  -The User sends the Assessment test. |
| Quality Requirements | -The Assessment test must reach the Physiotherapist’s inbox.  -The Assessment test must be complete before it can be sent. |

|  |  |
| --- | --- |
| Use Case Name | userAttachFile |
| Participating Actors | Initiated by User, participate with Physiotherapist |
| Entry Condition | -user must be logged in |
| Flow of Events | 1. User must activate the Contact Physiotherapist function of the software. 2. User will select the Send Message. 3. Self-Start will load a page that will allow a user to upload filesand a message that they can then submit to the Physiotherapist. 4. User can upload multiple files to the message, and then press submit to send the attachment to the Physiotherapist. 5. Self-Start will then send the attachment to the User’s Physiotherapist. |
| Exit Condition | -User exits the Send Video Message page.  -User sends the video message. |
| Quality Requirements | -The video message must reach the Physiotherapist’s inbox.  -The video message must have a body of the message. |

|  |  |
| --- | --- |
| Use Case Name | insufficientFunds |
| Participating Actors | Communicates with User |
| Entry Condition | This extends the payFee use case. It is initiated by the system when the user tries to pay and has no more funds left in their account. |
| Flow of Events | 1. User tries to pay from their account. 2. Self-Start checks the User’s current balance and finds that they do not have enough funds to pay. 3. Self-Start displays an alert to the User stating that they have insufficient funds, with an option to redirect to the “Purchase Plan” page. |
| Exit Condition | -User exits page  -User goes to “Purchase Plan” page |
| Quality Requirements | Self-Start must check to verify that the User does not have enough funds. |

|  |  |
| --- | --- |
| Use Case Name | designTreatmentPlan |
| Participating Actors | Initiated by the Physiotherapist |
| Entry Condition | Physiotherapist must be logged in and navigate to the designTreatmentPlan form. |
| Must Flow of Events | 1. The form page opens. 2. Physiotherapist selects the exercises from the list which they believe suit the client’s needs. 3. The Physiotherapist can plan the exercise list for some time into the future. 4. Physiotherapist confirms their daily selection with a Submit button and the sendExercises use case is initiated. Or they can Cancel or Save the list for later revision using cancel/save button. |
| Exit Condition | The Physiotherapist must choose at least one exercise and one client before initiating the sendExercises use case included from this, must save and leave form or must cancel and leave form. |
| Quality Requirements | -Must be able to connect to database to draw exercises and create a daily ‘recipe’ list of exercises of undefined size as each patient may have varying number of exercises to complete on any given day.  -Exercises must contain and be sent with specific information (i.e. Unique identification code, Name, Description, Author, name, Objectives, Action Steps, Location, Frequency & Duration, Target Date, Multimedia URL). |

|  |  |
| --- | --- |
| Use Case Name | postExercises |
| Participating Actors | Initiated by the Physiotherapist |
| Entry Condition | Physiotherapist must be logged in and navigate to the designTreatmentPlan form and have the form filled out properly. |
| Must Flow of Events | 1. Physiotherapist has entered the designTreatment plan use case, which this use case is included from. 2. Physiotherapist confirms their selection of exercises to send on the current date and time to the client with a click on the Submit button. 3. The exercise list is sent to the client, which is then updated on their profile. |
| Exit Condition | -The Physiotherapist must choose at least one exercise and one client before sending the form.  -A success message will let the Physiotherapist know the exercises were sent successfully before redirecting to previous page. |
| Quality Requirements | Must have the valid User account to send exercise plan to and the User must receive their Exercises daily. |

|  |  |
| --- | --- |
| Use Case Name | physioSendMessage |
| Participating Actors | Initiated by the Physiotherapist  Participates with the User |
| Entry Condition | Physiotherapist clicks send message button beside their client list. |
| Flow of Events | 1. Self-Start will bring the Physiotherapist to the message page, which automatically has the User set as the recipient, and the Physiotherapist set as the sender. 2. Physiotherapist must enter a message or link to an Assessment form, and can optionally enter a subject. 3. Physiotherapist can optionally enter the PhysioAttachFile use case. 4. Physiotherapist must click send to send the message to their client. 5. Self-Start will deliver the message to the User’s inbox, indicating who sent the message. |
| Exit Condition | - Physiotherapist exits the Contact User page.  - Physiotherapist sends the message. |
| Quality Requirements | -Self-Start must guarantee that the message will reach the User.  -The message function must automatically know which User to send to.  -Must be able to send URL links for the Assessment test forms to be sent.  -At any time, the Physiotherapist can initiate the PhysioAttachFile use case, which is included as part of this use case. |

|  |  |
| --- | --- |
| Use Case Name | physioAttachFile |
| Participating Actors | Initiated by Physiotherapist  Participated by the User |
| Entry Condition | -Physiotherapist must be logged in and within the PhysioSendMessage use case. |
| Flow of Events | 1. Physiotherapist must activate the sendMessage use case function of the software. 2. Physiotherapist will select the file type to Upload from their computer (i.e. video file or text file). 3. Physiotherapist clicks upload button and the file attaches to the message. 4. Physiotherapist is redirected to the PhysioSendMessage use case with selected files attached to the message. |
| Exit Condition | - Physiotherapist exits the use case by pressing Upload once videos are loaded.  - Physiotherapist has videos attached to message. |
| Quality Requirements | -The video message must attach to the message and be uploaded in appropriate file format. |

|  |  |
| --- | --- |
| Use Case Name | createAssessmentTest |
| Participating Actors | Initiated by the Physiotherapist |
| Entry Condition | Physiotherapist must be logged in, and navigate to Create Assessment Test section of application. |
| Flow of Events | 1. The create test page opens to a dynamic form. 2. The Physiotherapist fills out all applicable details, adding the needed test questions, conforming to each patient or problem type assessment needs. 3. The assessment test is saved to the database using save or not saved by cancelling the form. |
| Exit Condition | - Physiotherapist exits the Create Test page by saving or cancelling. |
| Quality Requirements | -All mandatory fields must be filled out before the form is saved.  -Physiotherapist can save and edit or delete the form once it is created. |

|  |  |
| --- | --- |
| Use Case Name | viewUserProfile |
| Participating Actors | Initiated by the Physiotherapist or Admin |
| Entry Condition | - Initiator must be logged in and navigate to User Profile section of the application. |
| Flow of Events | 1. Initiator views the specific client’s Profile, which includes client info, progress and rehabilitation plan on the page (updated as Users move through the exercises daily). 2. Once done clicks exit button to return to client list. |
| Exit Condition | -Presses exit button. |
| Quality Requirements | -The website should maintain the data of the client progress and rehabilitation plan and display it in a clean and informative way on their profile.  - The Physiotherapist must be able to edit the client rehabilitation plan at any time. |

|  |  |
| --- | --- |
| Use Case Name | generateReport |
| Participating Actors | Initiated by Physiotherapist or Admin |
| Entry Condition | - Physiotherapist or Admin initiates the ViewUserProfile use case, which this use case is included from. |
| Flow of Events | 1. Initiator presses the Generate Report button in the User Profile. 2. Initiator redirected to PDF document with the User’s progress report and stats filled out automatically. 3. Initiator is given the option to click save, cancel or print on this document window. |
| Exit Condition | - Initiator cancels, saves or prints the document and is redirected the ViewUserProfile use case. |
| Quality Requirements | - The app must use current user data to generate the report.  -The report must be in PDF form.  -All mandatory fields are filled out and the report must be insightful, drawing on appropriate data, and easy to read with nice layout.  - Initiator must be able to print the document. |

|  |  |
| --- | --- |
| Use Case Name | printReport |
| Participating Actors | Initiated by Physiotherapist or Admin |
| Entry Condition | - Initiator initiates the GenerateReport use case, which this use case is included from. |
| Flow of Events | 1. Initiator presses the Print Report button in the User Profile. 2. The PDF document is printed and the Initiator is redirected to the User Profile. |
| Exit Condition | - Initiator prints the document and is redirected the viewUserProfile use case. |
| Quality Requirements | -The report must be in PDF form, and there must be a print button visible on the report. |

|  |  |
| --- | --- |
| Use Case Name | viewReviews |
| Participating Actors | Initiated by the Physiotherapist or Admin |
| Entry Condition | - Initiator must be logged in.  - Initiator navigates to view reviews section of Self-Start application. |
| Flow of Events | 1. Initiator navigates to page. 2. Self-Start will show the different data points about all the different exercises and how all users have responded to them, individually and as a whole. 3. Once done clicks exit button to return to previous page. |
| Exit Condition | -Presses exit button. |
| Quality Requirements | -The application should maintain the precise and accurate data of the client reviews and display it in a clean and informative way. |

|  |  |
| --- | --- |
| Use Case Name | addAccount |
| Participating Actors | Initiated by Admin |
| Entry Condition | -Admin must be logged in. |
| Flow of Events | 1. Admin must activate the Add Account function of the software. 2. Admin selects which type of user to create, Physiotherapist user or Client user. 3. Form opens with appropriate field to fill out. 4. Create User button saves the instance of the new user in system. |
| Exit Condition | -Admin cancels the process by exiting or the new user is created with a success message and redirect to previous page. |
| Quality Requirements | -The Admin must be able to choose which type of user to add (Physiotherapist or Client).  -The new user must be saved to the database.  -All appropriate mandatory field must be filled out on form to proceed, otherwise warning will be given. |

|  |  |
| --- | --- |
| Use Case Name | manageUserProfiles |
| Participating Actors | Initiated by Admin |
| Entry Condition | -Admin must be logged in and click the Profile they wish to manage. |
| Flow of Events | 1. Admin views the profile details of the user that was previously created. 2. Admin can update all fields of profile (Contact Details, Billing Information, etc.) if needed, using combination of text fields and dropdowns. Admin can optionally enter the included Enable/Disable Account or UpdatePassword use cases here with corresponding buttons. 3. The update, cancel or delete profile button is clicked. |
| Exit Condition | -Admin saves the updated profile, cancels the changes made or deletes it. |
| Quality Requirements | -Admin must be able to cancel or save changes made, enable/disable accounts and change passwords of all Users.  -Admin must be able to delete the profile, an alert must pop up to confirm that this is what they want to change/delete before it is carried out.  -All mandatory fields are filled out and a warning is generated if they are not. |

|  |  |
| --- | --- |
| Use Case Name | disableEnable Account |
| Participating Actors | Initiated by Admin |
| Entry Condition | -Admin initiates the manageProfile use case. |
| Flow of Events | 1. Admin presses the Enable/Disable Account button in the User Profile. 2. An alert pops up asking if the Admin wants to Enable or Disable the account and a confirm button s pressed. 3. The value of the isEnabled value on this user is toggled and the visual flag is set that this account is Disabled or Enabled. The corresponding functionality of the new state is applied to this account. |
| Exit Condition | -Admin saves the updated profile or deletes it. |
| Quality Requirements | -Admin must be able to enable/disable accounts.  -Admin must be able to see clearly a visual indication of the current state (enabled or disabled) of the account.  -All new functionality/permissions updated depending on updated User state. |

|  |  |
| --- | --- |
| Use Case Name | updatePassword |
| Participating Actors | Initiated by Admin |
| Entry Condition | - Admin initiates the ManageUserProfiles use case. |
| Flow of Events | 1. Admin presses the Update Password button in the User Profile. 2. Admin sent to form with Reset password fields available to fill out. 3. Admin clicks save or cancel to update the password. |
| Exit Condition | -Admin cancels or updates the User password.  -An email is sent to the User indicating their password has changed. |
| Quality Requirements | - Admin must be able to cancel or save changes made.  -New password is updated in the database.  -All mandatory fields are filled out and a warning is generated if they are not.  -Email indicating password changed must be sent to the User. |

|  |  |
| --- | --- |
| Use Case Name | manageExercises |
| Participating Actors | Initiated by Admin |
| Entry Condition | -Admin must be logged in.  -Admin navigates to the specific exercise details. |
| Flow of Events | 1. Admin enters the section of the Self-Start app which contains the list of the exercises. 2. Admin clicks a specific exercise to update or delete, or can add a new exercise. 3. The edit details page of the exercise opens and the fields can be updated. 4. Admin clicks save, cancel or delete button to exit. |
| Exit Condition | -Admin cancels or updates or deletes the exercise details, and it is deleted from database.  -Exercise is saved to database if updated or added. |
| Quality Requirements | - Admin must be able to cancel or save changes made.  -Admin must be able to delete the exercise, an alert must pop up to confirm that this is what they want to change/delete before it is carried out.  -All mandatory fields are filled out and a warning is generated if they are not.  -Each exercise must contain a set of instructions along with a video when created. |

Functionality

* A new user must have functionality to create their own account, with a standard username and password. The password must be protected by encryption on the back end of the database.
* The system administrator must be able to add both a user account, and a physiotherapist account, to the database. This functionality must be integrated such that accounts added by the admin have the same functionality as accounts created by user registration.
* The user must be able to log in to their account using their determined username and password.
* The user must be able to change their user details independently through the application. This includes the ability to chage password, billing information, name, and any additional profile information.
* The user must be able to book an appointment with a physiotherapist. Where applicable, they should be able to book during an available timeslot of their choosing, and with any valid physiotherapist.
* Any physiotherapist should be able to generate reports that are able to see detailed information about clients, such as graphs corresponding to the ratings given for exercises by users.
* Users should be able to add a rating to any exercises prescribed to them. This rating follows a standard scale (1 to 10), and the rating is able to be updated at any time.
* Any physiotherapist should be able to submit a treatment plan, acting as a template to be sent to clients.
* The system administrator should be able to manage user profiles, including functionality to change user passwords, profile information, and also delete/suspend any profiles.
* The system administrator must be able to add, delete, or change any existing exercises within the exercise database. These exercises can be incorporated by any physiotherapist, and viewed by any user.
* The user must be able to add payment to their account. This payment functions via credits, and works on a subscription basis. The payment itself is to be handled by an involved third party, such as Paypal or directly through a bank. The system administrator must be notified of any payments made.
* The system must permit functionality to send messages between users and physiotherapists. These messages may include text, treatment plans, and any attachments such as pictures or videos.
* The system must include functionality for physiotherapists to create assessment tests specifically for a particular user. As well, these assessment tests must be both viewable and able to be filled in by the user in question.

Usability

* The Self Start system's top priority in design is ensuring that the application is simple enough for an inexperienced user to navigate and use to a high degree, fulfilling all required needs. This will be achieved using modern design implementation, with any user input options being clearly defined.
* The application should also be quick and painless to navigate throughout, ideally requiring no more than a few clicks to reach one end of the application from another. This is to be achieved through a simple yet thorough application layout and design, complete with a homepage that clearly leads to any required pages, and quick traversal from one page to another.
* For any user inputs within the application, ranging from entering a text field to clicking a button, it is worth noting that the user must always feel in control. This will be achieved by attaching a response to any action executed by any user, such as a loading icon or ensuring text entry is responsive.
* Throughout the application design, the specified Microsoft GUI standards will be followed, ensuring an optimal user experience. This entails ensuring sizing and positioning is ideal, and that all features that the user can interact with are intuitive and well defined. Any buttons will clearly describe their function to reduce confusion, and empty space will be kept to a minimum.

Reliability

* The availability goal of the site is to be functional and available to users and physiotherapists 24/7, as the goal for the application itself is to allow clients to access support during hours where support is otherwise unavailable. When there is maintenance required, notice will be given on the home page of the site as soon as possible, ideally at least 4 business days in advance. The maintenance will ideally be performed at determined low activity hours, such as early in the morning, to reduce any potential interruptions.
* The mean time between failures (MTBF) will ideally be as high as possible, with intention to be as high as two days (48 hours). This will help keep downtime to a minimum, allowing for the highest work output possible. Any detected failures will ideally be dealt with as quickly as possible, to help further reduce downtime.
* Due to the complexity of the site, the mean time to repair (MTTR) is planned to be kept below 10 hours, including both the diagnostics and repair phases. This ensures that the repair will not occupy an entire work day, and can even be finished during a single night where applicable.
* Though not applicable to a large range of features, the system's accuracy is important where relevant. This will notably be when generating reports for calculating ratings and exercise effectiveness, to which any calculated averages will be rounded to the nearest hundredth decimal place.
* A low defect rate is of high priority for the application, particularly with functions pertaining to payment and registration. The goal for these two sections is to reduce the bugs/KLOC to no more than 1, while for other, less dire functions, up to 3 bugs/KLOC are to be expected.
* For the system, any bugs are to be distributed by three classifications: Minor bugs, major bugs, and critical bugs.

***Minor:*** Any bugs pertaining to the UI, or the calculation of data.

***Major:*** Any bugs that directly affect messaging or data generation, which would specifically interfere with the user's treatment effectiveness. These are of a higher priority than any graphical bugs, and should be dealt with promptly.

***Critical:*** Any bugs that directly interfere with authentication, or which have the capability to cause a system-wide failure. These bugs must be dealt with immediately for the sake of security and usability, respectively.

Performance

* In tandem with the predefined user experience priority, response time is to be kept as low as possible, ideally no more than 3 seconds for any operation requested by the user. Some functions, such as messaging, may require slightly more time, up to a permitted 5 seconds.
* The initial deployment of the application will rest upon a simple server infrastructure, not designed for more than a hundred concurrent users. This will help reduce server costs while interest is gauged. Should the traffic exceed this low estimation, then the server can easily be updated to permit any amount of concurrent users.
* With regards to resource allocation and distribution, a bulk of operations will be executed either server-side, or with a back-end database. This helps to reduce strain on the user, and allows even users without modern computing devices to receive high quality access to the application. Should this cause excess strain on the system, it can easily be adjusted to require more client-side processing.

Supportability

* During the implementation phase, standard coding practice will be followed precisely. This includes proper capitalization for all variables and functions, as well as all variables and functions having a clear name that accurately describes their purpose. This will help with reducing maintenance time.
* Development for the application will rely on widely used libraries particularly, and libraries that are compatible with any browsers and any operating systems. There should also be mobile accommodations made, such that any client can access the system from anything and anywhere. A prominent example of permitting this would be to utilize JQuery rather than JavaScript, to ensure the cross-browser functionality. For any issues that are not supported by the designated libraries, further functionality may be added specifically for the clients that have issue using the application.
* To optimize available support, modern practices will be used, however this only includes practices that are well-known. Given the choice between a slightly older methodology or a sparsely used new methodology, the older methodology will be selected to maximize the available support for the system.

Design Constraints

* During implementation and setup, a bureaucratic method will be followed to optimize supportability and reliability, as well as for ease of access. This primarily includes each page of the application possessing its own folder, with its associated files and images belonging solely to this folder. File names will clearly define their purpose and relevance, and will follow standard naming and templating conventions.
* The application itself will follow a strict and consistent design scheme, including a consistent colour palette, and stylistic decisions that are easy to follow. The colours used will be such that colourblind users are not negatively impacted, and the overall design will be clear enough that any user with vision deficiency can easily follow the flow of events required.
* The application itself will be dynamically created where applicable, such that each user has an individualized experience, and so any information logged is not outdated upon viewing. Exceptions to this will lie within the messaging functionality, which will merely follow standard messaging protocol.

Online User Documentation and Help System Requirements

The SOLVERE team will be documenting the provided code to enable basic troubleshooting capabilities as offline support. To provide online support, every member of SOLVERE will provide their Western email for contact in case of major unexpected errors. Every member of SOLVERE is also expected to return to Western next year, and as such one year of in-person support can be provided.

Purchased Components

All parts of the product provided are proprietary software of SOLVERE. To the best of the SOLVERE team’s understanding, there will be no need for any third-party APIs. As a result, there will be no problems relating to third-party legal constraints or purchasing the right to use such software.

Interfaces

User Interfaces

User accounts, physiotherapist accounts, and admin accounts will all have different privileges and subsequently different views per those privileges. This will allow for easy and streamlined functionality even as the functionality differs based on the user’s level of permissions.

Hardware Interfaces

The product will require a hardware interface to host the server that will be running. Communication to the server, which will run on Ember.js will take place over HTTPS connections on port 80, though it is possible a back-end server with its own port will be utilized.

Software Interfaces

To the best of the SOLVERE team’s knowledge, there is no external software to interface with, nor are any components of the system being developed for use in other projects.

Communications Interfaces

As touched on above in 3.10.2, Hardware Interfaces, the server will need to communicate with various actors. The client machine, used by a user, physiotherapist, or admin, will connect to the server over HTTPS to ensure secure and efficient communication.

Licensing Requirements

The final product is the proprietary creation of SOLVERE, and it will be sold to Marcotte Physiotherapy Clinic. For the product to be used outside of that scope, written consent from SOLVERE must be obtained.

Legal, Copyright and Other Notices

All patient information, including but not exclusive to treatment plans, assessment tests, and user progress are private and cannot be shared to people who are not Marcotte Physiotherapy Clinic staff without the consent of the patient in question.

Applicable Standards

The product that SOLVERE provides will be held to certain standards, outlined below:

* The product can be held to certain standards regarding data integrity. To remove risk of foreign agents compromising the accuracy of the data, the database will have security measures limiting access to the data to easily controlled paths. The product will also have precautions against corrupted data to ensure accuracy of database contents.
* The product can be held to a standard of compatibility with any browser. As it will be built with the Node.js framework, any browser supporting HTTP and JavaScript will be able to access and use the product. Cookies are also expected to be used to help streamline the user experience, and while the product will be usable without cookies it is recommended that they be available.