<Team Name>

Modern Software Requirements Specification

For Self Start System>

Version <1.0>

[Note: Text enclosed in square brackets and displayed in blue italics (style=InfoBlue) is included to provide guidance to the author and should be deleted before publishing the document. A paragraph entered following this style will automatically be set to normal (style=Body Text).]

[To customize automatic fields (which display a gray background when selected), select File>Properties and replace the Title, Subject and Company fields with the appropriate information for this document. After closing the dialog, automatic fields may be updated throughout the document by selecting Edit>Select All (or Ctrl-A) and pressing F9, or simply click on the field and press F9. This must be done separately for Headers and Footers. Alt-F9 will toggle between displaying the field names and the field contents. See Word help for more information on working with fields.]

[Note: The Software Requirements Specification (SRS) captures the complete software requirements for the system, or a portion of the system.  The Modern SRS is a typical SRS outline for a project **using use-case modeling**. This artifact consists of a package containing use cases of the use-case model and applicable Supplementary Specifications and other supporting information. For a template of an SRS **not** using use-case modeling, which captures all requirements in a single document, with applicable sections inserted from the Supplementary Specifications (which would no longer be needed), see[\\program](file:///\\program) \program files\Rational\ RequisitePro\Outlines\ rup\_srs.dot.]

Many different arrangements of an SRS are possible. Refer to [IEEE93] for further elaboration of these explanations, as well as other options for SRS organization.]

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mmm/yy> | <x.x> | <details> | <name> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms and Abbreviations 4

1.4 References 4

1.5 Overview 4

2. Overall Description 4

2.1 Use-Case Model Survey 4

2.1.1 Introduction 4

2.1.2 Survey Description 4

2.1.3 Use-Case Model Hierarchy 5

2.1.4 Diagrams of the Use-Case Model 5

2.2 Assumptions and Dependencies 5

3. Requirements 5

3.1 Use-Case Specifications 5

3.2 Functionality 5

3.2.1 <Functional Requirement One> 5

3.3 Usability 6

3.3.1 <Usability Requirement One> 6

3.4 Reliability 6

3.5 Performance 6

3.5.1 <Performance Requirement One> 6

3.6 Supportability 6

3.6.1 <Supportability Requirement One> 7

3.7 Design Constraints 7

3.7.1 <Design Constraint One> 7

3.8 Online User Documentation and Help System Requirements 7

3.9 Purchased Components 7

3.10 Interfaces 7

3.10.1 User Interfaces 7

3.10.2 Hardware Interfaces 7

3.10.3 Software Interfaces 7

3.10.4 Communications Interfaces 7

3.11 Licensing Requirements 7

3.12 Legal, Copyright and Other Notices 7

3.13 Applicable Standards 7

Modern Software Requirements Specification

## Use-Case Specifications

|  |  |
| --- | --- |
| Use case name | Authenticate User  This use case covers: Objective 10 |
| Participating actor | Initiated by User |
| Entry condition | The user (Patient, Admin, Physiotherapist) is on the SelfStart website |
| Flow of event | 1. The user initiates “Authenticate User” from the website interface by logging in.   2. The SelfStart website responds by presenting a login screen that includes username and password fields.   1. The user fills the username and password fields. Once complete, the user submits the form by clicking on log in.   4. The SelfStart website authenticates the submitted form from the database and displays the acknowledgment |
| Exit condition | * The user receives confirmation of authentication and continues to the home page of SelfStart while logged into their account. * The user receives an error message that the username and password submitted was incorrect and to try again. |
| Quality requirement | * Not applicable |

|  |  |
| --- | --- |
| Use case name | Change Password  This use case covers: Feature 7 |
| Participating actor | Initiated by User |
| Entry condition | The user (Patient, Admin, Physiotherapist) is logged on the SelfStart website |
| Flow of event | 1. The user initiates “Change Password” from the website’s interface.  2. The SelfStart website responds by presenting an interface that includes an old password field (to confirm that it is the proper person making the changes) and a new password field.  3. The user inputs the appropriate information regarding their account in their respective fields.  4. User confirms password update.  6. The SelfStart system updates the changes made by the user in the database.  7. The SelfStart system sends an email to them telling them that a new password has been set. |
| Exit condition | * User is done changing their password. |
| Quality requirement | * Not applicable |

|  |  |
| --- | --- |
| Use case name | Ask A Physio  This use case covers: Feature 2 |
| Participating actor | Initiated by Patient |
| Entry condition | The patient is on the SelfStart website, either logged in or not. |
| Flow of event | 1. The user initiates “Ask A Physio” from the website’s interface.  2. The SelfStart website responds by presenting an interface that includes a form asking the patient to input some general information about themselves and their comment explaining their injury.  3. The user inputs the appropriate information in the form.  4. User submits form.  6. The SelfStart system adds the form into the database.  7. The SelfStart system sends an email to them telling them that they have successfully submitted an “Ask A Physio” form. |
| Exit condition | * Patient submits the “Ask A Physio” form. |
| Quality requirement | * If this is the patient’s first time submitting an “Ask A Physio”, they don’t need to make an account, but to make more in the future, they must make an account (This is an assumption, subject to change if requirement is undesired). |

|  |  |
| --- | --- |
| Use case name | Create Account  This use case covers: Feature 6 |
| Participating actor | Initiated by Patient |
| Entry condition | The patient is on the SelfStart website. |
| Flow of event | 1. The user initiates “Create Account” from the website’s interface.  2. The SelfStart website responds by presenting an interface that includes personal information fields.  3. The patient inputs the appropriate information in the appropriate fields.  4. Patient confirms that they want to create an account.  5. The SelfStart system updates the database.  6. The SelfStart system sends an email to them telling them that they have successfully created an account.  7. System logs in patient automatically after the account has been confirmed. |
| Exit condition | * User creates account. |
| Quality requirement | * Email must be unique (Assumption, subject to change if needed to) |

|  |  |
| --- | --- |
| Use case name | Book Appointment  This use case covers: Feature 2, Feature 3, Feature 4 |
| Participating actor | Initiated by Patient |
| Entry condition | The patient is logged onto their account on the SelfStart website. |
| Flow of event | 1. The user initiates “Book Appointment” from the website’s interface.  2. The SelfStart website responds by presenting the book appointment form to the patient to have them fill it out.  3. The patient inputs the appropriate information in the appropriate fields and answers the questions in the online evaluation form.  4. Patient has the option of an uploading image.  5. The SelfStart system updates the database.  6. The SelfStart system sends an email to them telling them that they have submitted a book appointment request. |
| Exit condition | * Patient submits book appointment. |
| Quality requirement | * None |

|  |  |
| --- | --- |
| Use case name | Make Payment  This use covers: Feature 30 |
| Participating actor | Initiated by Patient |
| Entry condition | The patient is logged onto their account on the SelfStart website. |
| Flow of event | 1. The user initiates “Make Payment” from the website’s interface.  2. The SelfStart website responds by presenting the payment form to the patient to have them fill it out.  3. The patient inputs the appropriate information in the appropriate fields.  4. The SelfStart system contacts the bank API to authenticate the user and make the payment.  5. The SelfStart system updates the database.  6. The SelfStart sends an email to the patient confirming that the payment has been confirmed and gone through. |
| Exit condition | * Patient confirms payment. |
| Quality requirement | * Payment method used has to be Paypal. |

|  |  |
| --- | --- |
| Use case name | View Exercises  This use covers: Feature 18 |
| Participating actor | Initiated by Patient |
| Entry condition | The patient is logged onto their account on the SelfStart website. |
| Flow of event | 1. The user initiates “View Exercises” from the website’s interface.  2. The SelfStart website responds by presenting an interface that shows guides on what exercises to do and how to do them. |
| Exit condition | * Patient is done viewing exercises. |
| Quality requirement | * Exercises are presented in the form of text and/or simple animations. |

|  |  |
| --- | --- |
| Use case name | Fill-in Assessment Test  This use covers: Feature 19, Feature 20 |
| Participating actor | Initiated by Patient |
| Entry condition | The patient is logged onto their account on the SelfStart website. |
| Flow of event | 1. The user initiates “Make Payment” from the website’s interface.  2. The SelfStart website responds by presenting the payment form to the patient to have them fill it out.  3. The patient inputs the appropriate information in the appropriate fields.  4. The SelfStart system contacts the bank API to authenticate the user and make the payment.  5. The SelfStart system updates the database.  6. The SelfStart sends an email to the patient confirming that the payment has been confirmed and gone through. |
| Exit condition | * Patient confirms payment. |
| Quality requirement | * Payment method used has to be Paypal. |

|  |  |
| --- | --- |
| Use case name | Manage Accounts  This use covers: Feature 7, Feature 27, Feature 28, Feature 29 |
| Participating actor | Initiated by Admin |
| Entry condition | The Admin is logged onto SelfStart using the admin account. |
| Flow of event | 1. The admin initiates “Manage Accounts” from the admin’s account interface.  2. The SelfStart website responds by presenting an interface that contains a variety of different options to manage accounts of patients and physiotherapist.  3. The admin can reset a patient’s forgotten password.  4. The admin can create patient and physiotherapist accounts.  5. The SelfStart system saves all changes to the database. |
| Exit condition | * Admin updates changes. |
| Quality requirement | * None. |

|  |  |
| --- | --- |
| Use case name | Manage Questions  This use covers: Feature 5 |
| Participating actor | Initiated by Admin |
| Entry condition | The Admin is logged onto SelfStart using the admin account. |
| Flow of event | 1. The admin initiates “Manage Questions” from the admin’s account interface.  2. The SelfStart website responds by presenting an interface that contains the current introduction form.  3. The admin can add, change, or delete questions from the introduction form.  4. The admin, after updating the introduction form, clicks on the update button.  5. The SelfStart system saves all changes to the database. |
| Exit condition | * Admin updates changes. |
| Quality requirement | * None. |

|  |  |
| --- | --- |
| Use case name | Manage Patient Profiles  This use covers: Feature 8 |
| Participating actor | Initiated by Physiotherapist |
| Entry condition | The Physiotherapist is logged onto SelfStart using the physiotherapist account. |
| Flow of event | 1. The physiotherapist initiates “Manage Patient Profiles” from the physiotherapist’s account interface.  2. The SelfStart website responds by presenting an interface that contains a list of the physiotherapist’s patients.  3. The physiotherapist can add, change, or delete information regarding their patient’s profile.  4. The physiotherapist, after updating the patient’s account, clicks on the update button.  5. The SelfStart system saves all changes to the database. |
| Exit condition | * Physiotherapist updates changes. |
| Quality requirement | * None. |

|  |  |
| --- | --- |
| Use case name | Assign Rehab Plans  This use covers: Feature 16, Feature 22 |
| Participating actor | Initiated by Physiotherapist |
| Entry condition | The Physiotherapist is logged onto SelfStart using the physiotherapist account. |
| Flow of event | 1. The physiotherapist initiates “Assign Rehab Plans” from the physiotherapist’s account interface.  2. The SelfStart website responds by presenting an interface that contains the rehab plans and patients (subject to change).  3. The physiotherapist can assign one or more rehab plans to one or more patients.  4. The physiotherapist, after assigning the rehab plan, clicks on the update button.  5. The SelfStart system saves all changes to the database.  6. The SelfStart system sends emails to all the patients that received rehab plans that they have received rehab plans. |
| Exit condition | * Physiotherapist updates changes. |
| Quality requirement | * None. |

|  |  |
| --- | --- |
| Use case name | Assign Rehabilitation Plans  This use covers: Feature 16, Feature 22 |
| Participating actor | Initiated by Physiotherapist |
| Entry condition | The Physiotherapist is logged onto SelfStart using the physiotherapist account. |
| Flow of event | 1. The physiotherapist initiates “Assign Rehabilitation Plans” from the physiotherapist’s account interface.  2. The SelfStart website responds by presenting an interface that contains the rehab plans and patients (subject to change).  3. The physiotherapist can assign one or more rehab plans to one or more patients.  4. The physiotherapist, after assigning the rehab plan, clicks on the update button.  5. The SelfStart system saves all changes to the database.  6. The SelfStart system sends emails to all the patients that received rehab plans that they have received rehab plans. |
| Exit condition | * Physiotherapist updates changes. |
| Quality requirement | * None. |

|  |  |
| --- | --- |
| Use case name | Manage Rehabilitation Plan  This use covers: Feature 9, Feature 11, Feature 12, Feature 13 |
| Participating actor | Initiated by Physiotherapist |
| Entry condition | The Physiotherapist is logged onto SelfStart using the physiotherapist account. |
| Flow of event | 1. The physiotherapist initiates “Manage Rehabilitation Plans” from the physiotherapist’s account interface.  2. The SelfStart website responds by presenting an interface that includes the current rehab plans made and some other functionalities.  3. The physiotherapist can create rehab plans.  4. The physiotherapist can add exercises and self-assessment activities to the rehabilitation plans.  5. The physiotherapist, after managing the rehabilitation plans, clicks on the update button.  6. The SelfStart system saves all changes to the database. |
| Exit condition | * Physiotherapist updates changes. |
| Quality requirement | * None. |

|  |  |
| --- | --- |
| Use case name | Manage Assessment Tests  This use covers: Feature 15 |
| Participating actor | Initiated by Physiotherapist |
| Entry condition | The Physiotherapist is logged onto SelfStart using the physiotherapist account. |
| Flow of event | 1. The physiotherapist initiates “Manage Assessment Tests” from the physiotherapist’s account interface.  2. The SelfStart website responds by presenting an interface that includes the current assessment test made and some other functionalities.  3. The physiotherapist can create, edit, and delete assessment tests.  4. The physiotherapist, after managing the assessment tests, clicks on the update button.  6. The SelfStart system saves all changes to the database. |
| Exit condition | * Physiotherapist updates changes. |
| Quality requirement | * None. |

|  |  |
| --- | --- |
| Use case name | Manage Exercises  This use covers: Feature 10, Feature 14 |
| Participating actor | Initiated by Physiotherapist |
| Entry condition | The Physiotherapist is logged onto SelfStart using the physiotherapist account. |
| Flow of event | 1. The physiotherapist initiates “Manage Exercises” from the physiotherapist’s account interface.  2. The SelfStart website responds by presenting an interface that includes the current exercises made and some other functionalities.  3. The physiotherapist can create, edit, and delete exercises.  4. The physiotherapist, after managing the exercises, clicks on the update button.  6. The SelfStart system saves all changes to the database. |
| Exit condition | * Physiotherapist updates changes. |
| Quality requirement | * None. |

|  |  |
| --- | --- |
| Use case name | Generate Reports  This use covers: Feature 17, Feature 21, Feature 23 |
| Participating actor | Initiated by Physiotherapist |
| Entry condition | The Physiotherapist is logged onto SelfStart using the physiotherapist account. |
| Flow of event | 1. The physiotherapist initiates “Generate Reports” from the physiotherapist’s account interface.  2. The SelfStart website responds by generating a report and downloading it to the physiotherapist’s computer.  3. The physiotherapist can print, email the report to the patient, or generate data analysis. |
| Exit condition | * Physiotherapist either prints, sends. |
| Quality requirement | * None. |