.

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

# Introduction

[The introduction of the Modern SRS should provide an overview of the entire Modern SRS. It should include the purpose, scope, definitions, acronyms, abbreviations, references and overview of the Modern SRS.]

## Purpose

[Specify the purpose of this Modern SRS. The Modern SRS should fully describe the external behavior of the application or subsystem identified. It also describes nonfunctional requirements, design constraints and other factors necessary to provide a complete and comprehensive description of the requirements for the software.]

## Scope

[A brief description of the software application that the Modern SRS applies to; the feature or other subsystem grouping; what Use Case model(s) it is associated with, and anything else that is affected or influenced by this document.]

## Definitions, Acronyms and Abbreviations

[This subsection should provide the definitions of all terms, acronyms, and abbreviations required to interpret properly the Modern SRS.  This information may be provided by reference to the project Glossary.]

## References

[This subsection should provide a complete list of all documents referenced elsewhere in the Modern SRS. Each document should be identified by title, report number (if applicable), date, and publishing organization. Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document.]

## Overview

[This subsection should describe what the rest of the Modern SRS contains and explain how the Modern SRS is organized.]

# Overall Description

## Use-Case Model Survey

### Introduction

A use case is a fundamental tool used to describe the functionality of a system. It highlights the basic functionality flow and the interaction between the system and external entities, referred to as actors. The use cases in this project work in unison to bring the treatment of the physiotherapist into the home. The system is going to be used by Marcotte Physiotherapy and will provide them with tools to improve the traditional approaches of the patient’s progression assessment. Each use case in this model is going to serve one main functionality which is to allow Marcotte Physiotherapy the ability to offer clients treatment from home.

### Survey Description

Self Start is a web-based system that will provide two-way communication between physiotherapists and clients. The purpose of this system is to automate the assessment, diagnosis and treatment activities that are currently carried out in person. The system will provide patients with access to the same services without having to book an in person appointment. The system is maintained by an administrator and allows physiotherapists to assess injuries, create treatment plans and monitor patient progress. The system also handles client payment. Additionally, the system authenticates each user at login and loads an appropriate user-interface based on the user’s role.

### Use-Case Model Hierarchy

|  |  |
| --- | --- |
| **Actors** | **Description** |
| Admin | The administrator of the system is responsible for ensuring 24/7 maintenance of the system. Controls creation and deletion of user accounts. |
| Physiotherapist | The physiotherapist uses the system to assess patient injuries, create exercises, assign treatment plans and monitor patient rehabilitation. |
| Client | The client utilizes the system to submit injuries, book appointments and view exercises and treatment. |

|  |  |
| --- | --- |
| **Use Case** | **Description** |
| Book Appointment | System allows the client to book appointments with a physiotherapist on-line with a self-confirmation message. |
| Create Account | System provides the ability for the client to create an online account. The account includes basic personal information like name, gender and date of birth etc. |
| View Treatment Exercises | System provides the ability for the client to view their treatment exercises. |
| Submit Self Assessment | System provides online forms to let the client respond to assessment test questions. |
| Process Payment | System accepts and processes the client’s online payments. |
| Contact Clinic | System provides the ability for unregistered clients to request an appointment or submit an online form explaining their injury. |
| Submit Introduction Form | System provides an online form that helps the client describe their injury and submit the form for assessment by the physiotherapist. |
| Change Password | System allows the client to change their password. |
| Create Rehab Plan | System provides the ability for the physiotherapist to create clinic-defined standard rehabilitation plans that include exercises and self-assessment activities. |
| Assign Rehab Plan | System provides the ability for the physiotherapist to assign a set of exercises to custom rehabilitation plans. |
| Send Treatment Plan | System provides the ability for the physiotherapist to generate and print or send a written report to the client that includes examination finds and a treatment plan. |
| Follow Up With Patient | System allows the physiotherapist to assign a follow-up session with the client or close the case. |
| Modify Introduction Form | System provides the ability for the administrator to customize the introduction form’s questions without changing system codes. |
| Manage Account | System allows the administrator to reset the user passwords and disable and enable existing user accounts. |
| Create Single Exercise | System provides the ability for the physiotherapist to create a single exercise. |
| Generate Single Self Assessment | System provides the ability for the physiotherapist to create an assessment test that includes test. |
| Assess Test Results | System provides the ability for the physiotherapist to generate data analysis that describes the impact of rehabilitation plans. |
| View Patient Summary | System provides the ability for the physiotherapist to display and print a summary report for each client. Report includes the clients personal information, the diagnosed case, treatments, appointments, invoices and final outcome. |

### Diagrams of the Use-Case Model

[Diagrams, primarily use-case diagrams, of the entire use-case model are included here.]

## Assumptions and Dependencies

[This section describes any key technical feasibility, subsystem or component availability, or other project related assumptions on which the viability of the software described by this Modern SRS may be based.]

# Requirements

[This section of the Modern SRS should contain all the software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements.   When using use-case modeling, the majority of these requirements are captured in the use cases.]

## Use-Case Specifications

|  |  |
| --- | --- |
| **Use Case Name** | Assign Rehab Plan |
| **Participating Actors** | Initiated by Physiotherapist |
| **Entry Condition** | Physiotherapist is logged into the system and on the Create Rehabilitation Plan page |
| **Flow of Events** | 1. The physiotherapist double clicks an exercise(s) from the Exercise side bar. System adds the exercise under the Exercise heading on the Rehabilitation Plan form. 2. The physiotherapist double clicks on an assessment test(s) from the Assessment Test side bar. System adds the assessment test under the Assessment Test heading on the Rehabilitation form. 3. The physiotherapist clicks the Create button. |
| **Exit Condition** | System sends an acknowledgment that the Rehabilitation Plan has been successfully created. |
| **Quality Requirement** | Rehabilitation Plan form must be simple and easy to use. |

|  |  |
| --- | --- |
| **Use Case Name** | Send Treatment Plan |
| **Participating Actors** | Initiated by Physiotherapist. Client participates. |
| **Entry Condition** | Physiotherapist is logged into the system |
| **Flow of Events** | 1. The physiotherapist presses the Send button. 2. System sends an acknowledgment that the treatment plan was sent to the client. |
| **Exit Condition** | Client receives a notification on their dashboard that treatment plan is available to view. |
| **Quality Requirement** | Client notification will appear on their dashboard within 1 minute of submission. |

|  |  |
| --- | --- |
| **Use Case Name** | Follow Up With Patient |
| **Participating Actors** | Initiated by Physiotherapist. Client participates. |
| **Entry Condition** | Physiotherapist is logged into the system |
| **Flow of Events** | 1. The physiotherapist presses the Book Appointment button or Close Case button. 2. System displays a message acknowledging the appointment was booked successfully or the case was closed successfully. |
| **Exit Condition** | Confirmation email is send to the client with the appointment information or acknowledging the case is closed. |
| **Quality Requirement** | Confirmation email will be send within 1 minute of submission. |

## Functionality

[This section describes the functional requirements of the system for those requirements that are expressed in the natural language style. For many applications, this may constitute the bulk of the Modern SRS Package and thought should be given to the organization of this section. This section is typically organized by feature, but alternative organization methods, for example organization by user, or organization by subsystem may also be appropriate. Functional requirements may include: **feature sets, capabilities and security**.

Where application development tools (requirements tools, modeling tools, etc) are employed to capture the functionality, this section document will refer to the availability of that data and indicate the location and name of the tool which is used to capture the data.]

### <Functional Requirement One>

[The requirement description.]

## Usability

[This section should include all of those requirements that affect usability. Examples:

1. Specify the required training time for a normal users and power users to become productive at particular operations.
2. Specify measurable task times for typical tasks, or
3. Base usability requirements of the new system on other systems that the users know and like.
4. Specify requirements to conform to common usability standards – e.g., IBM’s CUA standards, or the GUI standards published by Microsoft for Windows 95.]

### <Usability Requirement One>

The requirement description.

## Reliability

[Requirements for reliability of the system should be specified here. Suggestions:

1. Availability – specify % of time available ( xx.xx%), hours of use, maintenance access, degraded mode operations etc.
2. Mean Time Between Failures (MTBF) – this is usually specified in hours, but it could also be specified in terms of days, months, or years.
3. Mean Time To Repair (MTTR) – how long is the system allowed to be out of operation after it has failed?
4. Accuracy – specify precision (resolution) and accuracy (by some known standard) that is required in the systems output.
5. Maximum bugs or defect rate – usually expressed in terms of bugs/KLOC (thousands of lines of code), or bugs per function-point.
6. Bugs or defect rate – categorized in terms of minor, significant, and critical bugs: the requirement(s) must define what is meant by a “critical” bug (e.g., complete loss of data, complete inability to use certain parts of the functionality of the system).]

#### *<Reliability Requirement One*>

[The requirement description.]

## Performance

[The performance characteristics of the system should be outlined in this section. Include specific response times. Where applicable, reference related Use Cases by name.

1. Response time for a transaction (average, maximum)
2. Throughput (e.g., transactions per second)
3. Capacity (e.g., the number of customers or transactions the system can accommodate)
4. Degradation modes (what is the acceptable mode of operation when the system has been degraded in some manner)
5. Resource utilization: memory, disk, communications, etc.]

### <Performance Requirement One>

[The requirement description.]

## Supportability

[This section indicates any requirements that will enhance the supportability or maintainability of the system being built, including coding standards, naming conventions, class libraries, maintenance access, maintenance utilities.]

### <Supportability Requirement One>

[The requirement description.]

## Design Constraints

[This section should indicate any design constraints on the system being built. Design constraints represent design decisions that have been mandated and must be adhered to. Examples include software languages, software process requirements, prescribed use of developmental tools, architectural and design constraints, purchased components, class libraries, etc.]

### <Design Constraint One>

[The requirement description.]

## Online User Documentation and Help System Requirements

[Describes the requirements, if any, for on-line user documentation, help systems, help about notices, etc.]

## Purchased Components

[This section describes any purchased components to be used with the system, any applicable licensing or usage restrictions, and any associated compatibility/interoperability or interface standards.]

## Interfaces

[This section defines the interfaces that must be supported by the application. It should contain adequate specificity, protocols, ports and logical addresses, etc, so that the software can be developed and verified against the interface requirements.]

### User Interfaces

[Describe the user interfaces that are to be implemented by the software.]

### Hardware Interfaces

[This section defines any hardware interfaces that are to be supported by the software, including logical structure, physical addresses, expected behavior, etc.]

### Software Interfaces

[This section describes software interfaces to other components of the software system. These may be purchased components, components reused from another application, or components being developed for subsystems outside of the scope of this SRS, but with which this software application must interact.]

### Communications Interfaces

[Describe any communications interfaces to other systems or devices such as local area networks, remote serial devices, etc.]

## Licensing Requirements

[Defines any licensing enforcement requirements or other usage restriction requirements that are to be exhibited by the software.]

## Legal, Copyright and Other Notices

[This section describes any necessary legal disclaimers, warranties, copyright notices, patent notice, word mark, trademark, or logo compliance issues for the software.]

## Applicable Standards

[This section describes by reference any applicable standards, (and the specific sections of any such standards that apply to the system being described). For example, this could include legal, quality and regulatory standards, industry standards for usability, interoperability, internationalization, operating system compliance, etc.]