+

**TEAM ANODYNE**

**SRS**

SOFTWARE REQUIREMENTS SPECIFICATION

1.0

**JANUARY 2013**

**TEAM ANODYNE**

Lead Software Engineer

**PREPARED FOR**

Software Engineering

Module Code: MOD003263

**INSTRUCTOR**:

Dr. Cristina Luca

Bryant 110, Cambridge Campus

Cristina.luca@anglia.ac.uk

# REVISION HISTORY

|  |  |  |  |
| --- | --- | --- | --- |
| **DATE** | **DESCRIPTION** | **AUTHOR** | **COMMENTS** |
| October 2013 | 1.0 | Team Anodyne | Successful! |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# DOCUMENT APPROVAL

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **SIGNATURE** | **PRINTED NAME** | **TITLE** | **DATE** |
|  | DR. CRISTINA LUCA | INSTUCTOR | OCTOBER 2014 |
|  |  |  |  |
|  |  |  |  |

**TABLE OF CONTENTS**

Revision History ii

Document Approval ii

1. Introduction 1

1.1 Purpose 1

1.2 Scope 1

1.3 Definitions, Acronyms, and Abbreviations 1

1.4 References 1

1.5 Overview 1

2. General Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Characteristics 2

2.4 General Constraints 2

2.5 Assumptions and Dependencies 2

3. Specific Requirements 2

3.1 External Interface Requirements 3

3.1.1 User Interfaces 3

3.1.2 Hardware Interfaces 3

3.1.3 Software Interfaces 3

3.1.4 Communications Interfaces 3

3.2 Functional Requirements 3

3.2.1 <Functional Requirement or Feature #1> 3

3.2.2 <Functional Requirement or Feature #2> 3

3.3 Use Cases 3

3.3.1 Use Case #1 3

3.3.2 Use Case #2 3

3.4 Classes / Objects 3

3.4.1 <Class / Object #1> 3

3.4.2 <Class / Object #2> 3

3.5 Non-Functional Requirements 4

3.5.1 Performance 4

3.5.2 Reliability 4

3.5.3 Availability 4

3.5.4 Security 4

3.5.5 Maintainability 4

3.5.6 Portability 4

3.6 Inverse Requirements 4

3.7 Design Constraints 4

3.8 Logical Database Requirements 4

3.9 Other Requirements 4

4. Analysis Models 4

4.1 Sequence Diagrams 5

4.3 Data Flow Diagrams (DFD) 5

4.2 State-Transition Diagrams (STD) 5

5. Change Management Process 5

A. Appendices 5

A.1 Appendix 1 5

A.2 Appendix 2 5

# 1. INTRODUCTION

This SRS document is written in a self-explanatory form that will give a great insight to the system’s development process.

## 1.1 PURPOSE

The aim of this document is to give a detailed description of the Electronic System of the Over Surgery designed to replace the existing Paper-Based System currently in use.

This which will make working hours more productive and efficient

*.*

## 1.2 SCOPE

This Designed Electronic System **WILL** allow its **authorized** user(s)

* Check the Availability of GP(s) and Nurses
* Register new patient(s)
* Book, Change and Cancel Appointments,
* Extend Prescriptions (where necessary)
* Check and Print out Patient’s Test Result(s).

The System **WILL NOT** allow its **authorized** user(s)

* Generate (Bio-) Data of Patients that are not registered on the system
* Extend Prescription(s) of patients without a GP’s approval
* See a GP(s) or Nurse(s) availability **expect** the GP(s) or Nurse(s) has logged in/registered his/her presence/availability into a “connected” secondary database

## 1.3 Definitions, Acronyms, and Abbreviations

|  |  |  |
| --- | --- | --- |
| **TERM** | **SECTION USED** | **MEANING** |
| Authorized | 1.2 | To give someone a legitimate access |
| (Bio) - Data | 1.2 | Primary Details of a person such as Name, Age |
| SRS | 1 | Software Requirement Specification |
|  |  |  |

## 1.4 REFERENCES

**[1]** Dr. Orest Pilskalns (WSU, Vancover) and Jack Hagemeister (WSU, Pullman: The SRS templates used as guides in developing the template for the WSU-TC Spring 2005 CptS 322 course that is used in this document

**[2]** Sections of this document are based upon the IEEE Guide to Software Requirements Specification (ANSI/IEEE Std. 830-1984).

## 1.5 OVERVIEW

The rest of this SRC document will state the General description and Specific Requirements as well as the Non-Functional Requirements of the Designed System.

It will describe the user’s characteristics and the system’s perspective, functions and its general constraints, assumptions and dependencies

It will also describe the user/ hardware/ software/ communications interfaces

For a better understanding of this documents pictorial explanations will used

…all these will be in **plain language!**

# 2. GENERAL DESCRIPTION

## 2.1 Product Perspective

The system will allow the user to carry out a wide variety of tasks, which the user will use at the Over Surgery. The user uses a paper based system to carry out tasks such as booking appointments and canceling appointments and registering patients and many more. With each task been carried out using a paper system, this is very hard to manage and keep tack of, due to the size of the system. When the system uses a computer it will allow the user to carry out the exact same tasks but on an electronic system, which will be easier to use and manage. The product will also be used by itself and not run along side any other product. The function of the product will be discussed bellow in section 2.2 Product function.

## 2.2 Product Functions

This subsection of the SRS should provide a summary of the functions that the software will perform.

The software will perform will perform a wide variety of tasks which will allow the user to carry out the same tasks that they do on the current paper based system. The system should be easier to use and quicker to use while at the same time allowing the user to keep all the functionality of the current system. The system will need to be able to:

* Handle large amounts of data
* Run without crashing
* Provide useful error messages
* Be upgradeable – Add new parts to the system
* Easy to maintain
* Scalable
* Perform quickly even when processing large amounts of data
* Have a simple easy to use GUI

To ensure that each part of the system meets the product function set out in this documentation. It will be tested to make sure that no problems arise. The product function will also be communicated to the user.

## 2.3 User Characteristics

The user is of a non technical nature who wants to enter as little information into the system as possible. The user needs to know only about patient information and GP/nurse information to use the system effectively. The user needs aspect to all parts of the system with the expectation of that database itself, so the user will not require a high technical knowledge. The user is also slow at typing so minimum input by the user will be the best option of the system.

## 2.4 General Constraints

The computer that the system will be running on will be a standalone machine with an age of 3 years, so the system will have to use very few resources in order to function on this old computer. The system will only have to interface with other applications when printing patient prescriptions. The system will also be accessed by one user and must ask for a secure username and password to make sure that the database can only be accessed by the receptionist.

The user will have access to all patient and GP/nurse information but will have no control over the physical database therefore eliminating any threat of data been stolen by the user.

## 2.5 Assumptions and Dependencies

A number of factors that may affect the requirements specified in the SRS include:

* The operating system been changed through the project to OSX or a never Windows operating system. This intron may cause the application to crash or not work correctly, resulting in a loss of data.
* The user maybe changed which may affect how the system is intended to work as the user may have different requirements to the previous user, resulting in a change of the system which could affect the surgery due to system downtime.
* The SRS may change also if the hardware is out of date or not functioning correctly, such as faulty USB ports so not been able to connect a printer, resulting in the system not been able to print prescriptions.

# 3. SPECIFIC REQUIREMENTS

This will be the largest and most important section of the SRS. The customer requirements will be embodied within Section 2, but this section will give the D-requirements that are used to guide the project’s software design, implementation, and testing.

Each requirement in this section should be:

* Correct
* Traceable (both forward and backward to prior/future artifacts)
* Unambiguous
* Verifiable (i.e., testable)
* Prioritized (with respect to importance and/or stability)
* Complete
* Consistent
* Uniquely identifiable (usually via numbering like 3.4.5.6)

Attention should be paid to the carefuly organize the requirements presented in this section so that they may easily accessed and understood. Furthermore, this SRS is not the software design document, therefore one should avoid the tendency to over-constrain (and therefore design) the software project within this SRS.

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces

### 3.1.2 Hardware Interfaces

### 3.1.3 Software Interfaces

### 3.1.4 Communications Interfaces

## 3.2 Functional Requirements

This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.

### 3.2.1 <Functional Requirement or Feature #1>

3.2.1.1 Introduction

3.2.1.2 Inputs

3.2.1.3 Processing

3.2.1.4 Outputs

3.2.1.5 Error Handling

### 3.2.2 <Functional Requirement or Feature #2>

…

## 3.3 Use Cases

### 3.3.1 Use Case #1

### 3.3.2 Use Case #2

…

## 3.4 Classes / Objects

### 3.4.1 <Class / Object #1>

3.4.1.1 Attributes

3.4.1.2 Functions

<Reference to functional requirements and/or use cases>

### 3.4.2 <Class / Object #2>

…

## 3.5 Non-Functional Requirements

Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second, system downtime may not exceed 1 minute per day, > 30 day MTBF value, etc).

### 3.5.1 Performance

### 3.5.2 Reliability

### 3.5.3 Availability

### 3.5.4 Security

### 3.5.5 Maintainability

### 3.5.6 Portability

## 3.6 Inverse Requirements

State any \*useful\* inverse requirements.

## 3.7 Design Constraints

Specify design constrains imposed by other standards, company policies, hardware limitation, etc. that will impact this software project.

## 3.8 Logical Database Requirements

Will a database be used? If so, what logical requirements exist for data formats, storage capabilities, data retention, data integrity, etc.

## 3.9 Other Requirements

Catchall section for any additional requirements.

# 4. ANALYSIS MODELS

List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable the SRS’s requirements.

## 4.1 Sequence Diagrams

## 4.3 Data Flow Diagrams (DFD)

## 4.2 State-Transition Diagrams (STD)

# 5. Change Management Process

Identify and describe the process that will be used to update the SRS, as needed, when project scope or requirements change. Who can submit changes and by what means, and how will these changes be approved.

# A. Appendices

Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS’s overall set of requirements.

*Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.*

## A.1 Appendix 1

## A.2 Appendix 2