**[Product]**

**MTM Program Product**

**Software Requirements Specification**

*[SRS Version Number]*

*[SRS Version Date]*

*[Applying MTM SRS Template Version Number]*

Standard Version Number: 3.5

Standard Version Date: March 10, 2018

Version History

|  |  |  |  |
| --- | --- | --- | --- |
| *Version* | *Date* | *Authors* | *Comment* |
|  |  |  |  |

Template Version History

|  |  |  |  |
| --- | --- | --- | --- |
| *Version* | *Date* | *Authors* | *Comment* |
| 3.0 | 7/21/2012 | Frank Ackerman | Initiating standards versions |
| 3.1 | 8/2/2012 | Frank Ackerman | Some non-functional requirements definitions .Added Adaptability, Enhanceability, and Portability |
| 3.2 | 1/17/2013 | Frank Ackerman | Added usability comment |
| 3.3 | 3/6/2013 | Frank Ackerman | Added a bit more explanatory text and final section 8. |
| 3.5 | 3/10/2018 | Celia Schahczenski | Changed format of dates, rearranged, renamed items, removed Illustrative Use Cases, increased some explanations, added appendices including data and report sections. |

**Montana Tech Software Engineering Students:**

These Montana Tech Method software engineering standards encapsulate Dr. Ackerman’s decades of experience in the software industry, the IEEE software engineering standards, and many suggestions from various texts. They have gone through many revisions and additions over the last several years. They are part of your software engineering studies so that (1) you may have the experience of developing software to a standard (which you may find you need to do if you take a job that requires high reliability software), and so that (2) you will have the experience of developing high quality software. You are also invited to participate in the continuing evolution of these standards by studying them critically and making suggestions for their improvement and correction.

TABLE CONTENTS

[1 Introduction 5](#_Toc508609754)

[1.1 Software Purpose and Scope 5](#_Toc508609755)

[1.2 Document Purpose and Contents 5](#_Toc508609756)

[1.3 References 6](#_Toc508609757)

[2 General Factors 6](#_Toc508609758)

[2.1 Product Perspective 6](#_Toc508609759)

[2.2 Product Functions 6](#_Toc508609760)

[2.3 Environmental Conditions 6](#_Toc508609761)

[2.4 User Characteristic 6](#_Toc508609762)

[2.5 Dependencies 7](#_Toc508609763)

[2.6 Assumptions 7](#_Toc508609764)

[3 Use Cases 7](#_Toc508609765)

[3.1 Actor 7](#_Toc508609766)

[3.2 Use Cases 8](#_Toc508609767)

[3.2.1 [Use Case Name] 8](#_Toc508609768)

[3.2.2 [ Use Case Name 2] 9](#_Toc508609769)

[4 Specific Requirements 9](#_Toc508609770)

[4.1 Functional Requirements 9](#_Toc508609771)

[4.2 Quality Attributes 10](#_Toc508609772)

[4.2.1 Availability 10](#_Toc508609773)

[4.2.2 Human Factors 10](#_Toc508609774)

[4.2.3 Usability 10](#_Toc508609775)

[4.2.4 Performance 10](#_Toc508609776)

[4.2.5 Security 10](#_Toc508609777)

[4.2.6 Reliability 10](#_Toc508609778)

[4.2.7 Maintainability 10](#_Toc508609779)

[4.2.8 Enhanceability/Extendibility 10](#_Toc508609780)

[4.2.9 Portability 11](#_Toc508609781)

[4.2.10 V&V Activities 11](#_Toc508609782)

[4.2.11 Adaptability 11](#_Toc508609783)

[4.3 Non-Functional Requirements Which Are Not Quality Attributes 11](#_Toc508609784)

[4.3.1 External Interface Requirements 11](#_Toc508609785)

[4.3.2 Development Environment 11](#_Toc508609786)

[4.3.3 Delivery Environment 11](#_Toc508609787)

[4.3.4 Design Constraints 11](#_Toc508609788)

[4.3.5 Database 12](#_Toc508609789)

[4.3.6 Deliverable Items, Dates and Conditions 12](#_Toc508609790)

[4.3.7 Cost 12](#_Toc508609791)

[4.3.8 Standards 12](#_Toc508609792)

[5 Future Enhancements 12](#_Toc508609793)

[Appendices 13](#_Toc508609794)

[Appendix A: Definitions, Acronyms, and Abbreviations 13](#_Toc508609795)

[Definitions 13](#_Toc508609796)

[Acronyms and Abbreviations 13](#_Toc508609797)

[Appendix B: Analysis Models 13](#_Toc508609798)

[Appendix C: Data Dictionary 13](#_Toc508609799)

[Appendix D: Report Specification 14](#_Toc508609800)

[Appendix E: Business Rules 15](#_Toc508609801)

[Appendix F: Sample User Interface 15](#_Toc508609802)

[Appendix G: Issues 15](#_Toc508609803)

*[Steps to turn this template into a product SR:*

* *Change the file name to a name that refers to the product and contains “SRS”*
* *Substitute the name of the product for* [Product] *in both the title and the heading.*
* *Change [VersionNumber] to “Version 1.0” for the first complete version or to “0.x” for initially incomplete versions. Do this in the middle of the title page, the heading, and the version table.*
* *Change [VersionDate] to the release date of this version and also change this item on the heading.*
* *Make an initial entry in the title page version table. The version number and date in the table should match the entry on the center of the page*
* *Either replace all of this bracketed, italic text with actual SRS text or delete it.*
* *When your document is complete re-set the table of contents so that the headings there match the document.]*

# Introduction

*[This Software Requirements Specification template is designed to facilitate the definition of processes and procedures relating to software requirements specification activities. This template was developed using IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications.*

Information displayed in brackets is explanatory. Delete the bracketed text items and add your project-specific input. These items are food for thought on the section they address.

*The introduction section should explain the purpose and scope of the project software requirements specification (SRS), as well as, provide clarification of definitions, acronyms, and references. This section should also provide an overview of this document.*

*Place any material here that is not specific to any of the sub-sections below.]*

## Software Purpose and Scope

*[This subsection should:*

1. *Identify the software products to be produced, by name*
2. *Explain what the software products will, and if necessary, will not do*
3. *Describe the application of the software being specified including all relevant goals, objectives, and benefits from producing the software.]*
4. *The purpose of the software is to allow prisoners to read to their child. It will not allow prisoners to just talk to their child. It is an idea from Dr. Swain.*

## Document Purpose and Contents

*[This subsection should explain the purpose for writing an SRS for this project and describe the intended audience for the SRS. This subsection should describe the information that will be presented in each of the subsections from §2 on.]*

*This Document lists all the requirements and details of the logicality of the reading prisoner's prototype.*

## References

*[This subsection should list all important references used within the SRS. If there are no pertinent references for this product that fact should be stated here.]*

# General Factors

*[The General Factors section should describe the general factors that affect the product and its requirements. Place any material here that is not specific to any of the sub-sections.*

*In this and each subsequent major section, briefly describe the purpose of this section from the reader’s perspective.]*

## Product Perspective

*[This subsection should put the product into perspective with other related products or projects. If the product to be produced from this SRS is totally independent, it should be clearly stated here. If the product to be produced from this SRS is part of a larger system, then this subsection should describe the functions of each component of the larger system or project and identify the interfaces between this product and the remainder of the system or project. This subsection should identify all principle external interfaces for this software product (Note: descriptions of the interfaces will be contained in another part of the SRS).]*

## Product Functions

*[This subsection should provide a summary of the functions to be performed by the software produced as a result of this SRS. Functions listed in this subsection should be organized in a way that will make it understandable to the intended audience of the SRS. (Note: this subsection is an overview; details of the specific requirements will be contained in section 4.)]*

## Environmental Conditions

*[This subsection should provide a summary of the environment in which the software must operate. (Note: this subsection is an overview; details of the specific requirements will be contained in the remainder of the document.)]*

*The Jail has specific times for calls. They control who the prisoner* *can contact.*

## User Characteristic

*[This subsection should describe the general characteristics of the eventual users of the product that will affect the specific requirements. Eventual users of the product will include end-product customers, operators, maintainers, and systems people as appropriate. For any users that impact the requirements, characteristics such as education, skill level, and experience levels will be documented within this subsection as they impose constraints on the product.]*

The primary users of the system are incarcerated parents and children/families.

The children are going to be young, and have an adult with them.

The adult on the other end is going to be a parent.

Program is meant for users with any level of computer experience, users have to be literate.

## Dependencies

*[This subsection should list all external system dependencies on which the software resulting from the SRS will depend. This subsection should be the source for recognizing the impact of any changes to systems on the SRS and resulting software depends. This section can highlight unresolved requirement issues that should be recorded on the Project Manager’s Open Issues List.]*

*Not needed*

## Assumptions

*[This subsection should list all assumptions that on which the software resulting from the SRS will depend that have not been covered above. This subsection should be the source for recognizing the impact of any changes to these assumptions on the SRS and resulting software.. This section can highlight unresolved requirement issues that should be recorded on the Project Manager’s Open Issues List.]*

Not needed

# Use Cases

*[Use cases describe possible interactions between an actor and a system that results in an outcome that provides value to the actor. Develop these use cases with the client. This section may begin with a use case diagram, an analysis model that identifies the actors who can interact with a system, along with the various use cases with which each actor might be involved.*

*If no use cases exist for this product, this section should read “*Use Cases were not developed for this specification”*.]*

*Scenarios*

## Actor

[This subsection lists the various actors that will interact with the proposed system, along with the interactions that these actors may perform. An actor is a person, or other entity external to the software system, who may interacts with the proposed system to accomplish tasks. Actors may represent roles, identified from the customer community that will use the product.

*The following is a template for documenting user classes and the associated use cases.*]

|  |  |
| --- | --- |
| Primary Actor | Use Cases |
| *Actor 1* | 1. *Use case 1* 2. *Use case 2*   *…* |
| *Actor 2* | 1. *Use case 1* 2. *Use case 2*   *…* |
| *…* | *…* |

## Use Cases

*[This subsection contains use cases of the proposed system. Use cases can be “casual” or “fully dressed”. The template given is for a fully dressed use case. Omit portions of this template for casual use cases.*

### *[Use Case Name]*

|  |  |  |  |
| --- | --- | --- | --- |
| *Created By:* | *Name 1* | *Last Updated By:* | *Name 2* |
| *Date Created:* | *Month dd, yyyy* | *Date Last Updated:* | *Month dd, yyyy* |
| *Actors:* | *User class name* | | |
| *Description:* | *Description of use case* | | |
| *Preconditions:* | 1. *Precondition 1 or “none”* 2. *…* | | |
| *Postconditions:* | 1. *Postcondition 1 or “none”* 2. *…* | | |
| *Normal Flow:* | *1.0 Description phrase*   1. *Step 1.* 2. *Step 2.* 3. *…* | | |
| *Alternative Flows:* | *1.1 Description phrase for alternative flow (branch after step n)*   1. *Step 1.* 2. *Step 2.*   *…*  *n. Return to Step m.*  *1.2 Description phrase for 2nd alternative flow (branch after step l)*   1. *…* | | |
| *Exceptions:* | *1.0.E.1 Description phrase for exception (at step k)*   1. *Step 1.* 2. *Step 2.*   *…*  *1.0.E.2 Description phrase for 2nd exception (at step j)*  *1. …* | | |
| *Includes/Extends:* | *Name of included use cases, name of the use cases that this use case extends, or “none”* | | |
| *Priority:* | *Low, medium or critical* | | |
| *Frequency of Use:* | *How often it is expected that this use case will be performed.* | | |
| *Business Rules:* | *Business rules can be listed, or another document can be referenced, or “none”* | | |
| *Special Requirements:* | 1. *Special requirements 1 concerning this use case.* 2. *…*   *or “none”* | | |
| *Assumptions:* | 1. *Assumption 1 concerning this use case.* 2. *…*   *or “none”* | | |
| *Notes and Issues:* | 1. *Note 1 concerning this use case.* 2. *…*   *or “none”*  *This can serve as a placeholder for extraneous information.* | | |

### *[ Use Case Name 2]*

*…*

Inmate Harbor Scenario

Enters Username and password

clicks submit button

Picks contact

Clicks connect

Clicks book from history

Clicks next page when finished reading

Clicks Back to Menu when done reading

Searches book and picks it

Clicks end call

Clicks feedback emoji

Click submit

Child Scenario

Click accept

watch book get read

press end call

Clicks feedback emoji

Click submit

# Specific Requirements

*[The Specific Requirements section should contain all the requirements for the subject software. The details within this section should be defined as individual, specific requirements. Each specific requirement should be stated such that its achievement can be objectively verified by observation, inspection, usability testing, functional testing, analysis, or a combination of these. The method verification must be described. Each requirement should be clearly identified for tracking.]*

## Functional Requirements

*[This subsection should specify how the software product will react to every possible input situation. It describes all the actions that must take place in the software in response to every input. Pertinent changes in the environment are considered to be inputs.*

*Care must be taken to avoid dropping into design details. In the user cannot directly experience the effect of a requirement it probably crossed the line into design.*

*Functional requirements should be logically grouped. Each group should have a short, unique (within the SRS) abbreviation and a number. The word processing section number will probably change as the SRS is developed.*

*For each identified requirement an optional rationale for that requirement may be given.*

*Most modern software should provide at least a modicum of user help. For very complex applications in situ help may be supplemented by a user’s manual (or manual page) but for many simple applications comprehensive in situ help is sufficient.]*

*Drop written requirements.*

*Adult can Sign in and type their password. ADULT CAN Selects contact. Adult can select book, read book. If adult finishes one book, they can choose another book. Either party can end the call child can accept or deny call, child can listen to book,*

## Quality Attributes

*[This subsection specifies criteria used to judge the operation of a system, rather than specific behaviors of the system. Specify the specific behavior of the system in the functional requirements.]*

*Make stuff up when not applicable*

### Availability

Available to prisoners and their families who qualify for the program

### Human Factors

*[Not everyone has the same inherent mental and physical capabilities vis-à-vis a given computer application. For example if sound is part of the application, will other clues be given that will enable a hard of hearing user to use the proposed application as well as person with normal hearing; similarly for color blindness. Define these factors, if necessary, with validation criteria.]*

*Accessibility settings (such as colorblind mode)*

### Usability

It needs to be brain-dead-easy to use so that the child and adult who both likely have little tech-literacy can use it.

### Performance

Capable of streaming video footage with little lag.

### Security

Should be secure enough that inmates can only access their approved contacts as well as secure enough that it doesn’t get hacked often by outside sources.

### Reliability

*[Reliability is specified as mean-time-to failure of an operational item. An operational profile must be specified.]*

It needs to be available during times where a prisoner could usually have visiting hours.

### Maintainability

Parts of the program should be made modular to increase maintainability of the overall program

### Enhanceability/Extendibility

*[If the future it might be necessary to change the Functional requirements in specified ways, what is the maximum estimated effort required to make such changes and what is the rationale for this estimate?]*

TBA

### Portability

*[If in the future it might be necessary to change the above Development or Delivery Environments (DV or DL) to other specified environments, what is the maximum estimated effort required to implement such changes and what is the rationale for this estimate]*

In the future it might need to be available on common OSes on the family side. This is because a prisoner’s family is likely to not use the native OS this software was designed for.

### V&V Activities

Username and password.

### Adaptability

*[If it is specified that in the future it might be necessary to change any of the above Non-Functional requirements, what is the maximum estimated effort required to implement such changes and what is the rationale for this estimate.]*

## Non-Functional Requirements Which Are Not Quality Attributes

*[This subsection specifies non-functional criteria such as platform, deployment, interface, design and document requirements. If there is not a document describing project requirements, those requirements (cost, schedule, etc.) can be placed here.]*

*Needs to be able to work within facility*

### External Interface Requirements

#### Hardware

Requires a touch screen, speaker, camera, and microphone.

#### Software

Some functioning operating system, database of user info.

#### Communications

Internet connection.

### Development Environment

Ignore

### Delivery Environment

#### Site

*[This subsection should specify any requirements for installation or operation of the software that might change the pre-existing configuration of the user site.]*

#### Operations

*[This subsection should specify normal and special operations required by the user to include:*

* *Various modes of operation within the user organization*
* *Periods of interactive operations and unattended operations*
* *Data processing support functions*
* *Backup and recovery operation.]*

*Has to work with facility procedures*

### Design Constraints

*[Sometimes a client will require certain design constraints, for example the use of a certain system configuration or the use of particular algorithm. Such constraints are described in this subsection.]*

### Database

*[This optional subsection specifies requirements for any database to be developed as part of the product. The information in this section may include:*

* *Types of information to be stored*
* *Table attributes (queried, supporting, updated)*
* *Frequency of access*
* *Accessing capabilities and requirements*
* *Data elements and file descriptors*
* *Retention requirements for data.]*

*Take care to avoid design details. Unless so requested by the client, this section should only contain as much information about saved data as is necessary to fully document any of the requirements given above.]*

Database should contain prisoner username and password, as well as their contact list.

### Deliverable Items, Dates and Conditions

### Cost

Should be relatively low cost, after supplying the technology required to run the software

### Standards

# Future Enhancements

*[This section should describe any future enhancements that are contemplated at the time this SRS completed. If there is no known possibility that this product will be enhanced in the future this section should read :* It is not expected that there will be any future enhancements to this product.*]*

# Appendices

*[In some cases, it is helpful to move items out of the main portion of the Software Requirements and Specification Document. These items can appear here. Alternatively, move these items into the main part of the document.]*

# Appendix A: Definitions, Acronyms, and Abbreviations

*[This appendix should provide the definitions of all terms, acronyms, and abbreviations required to fully understand your SRS.]*

## Definitions

|  |  |
| --- | --- |
| software failure | a failure will be attributed to this software product whenever one of the delivered work products does not meet the requirements specified in this SRS, or does not meet ordinary and reasonable customer/user expectations. |

## Acronyms and Abbreviations

|  |  |
| --- | --- |
| DB | Database |
| HW | Hardware |
| SDD | Software Design Description |
| SRS | Software Requirements Specification |
| SW | Software |

# Appendix B: Analysis Models

*[Optionally, include any pertinent analysis models, such as activity diagrams, state-transition diagrams, entity-relationship diagrams, or a formal specification.]*

# Appendix C: Data Dictionary

*[The data dictionary defines the composition of data structures and the meaning, data type, length, format, and allowed values for the data elements that make up those structures. In many cases, storing the data dictionary as a separate artifact, rather than embedding it in an SRS is beneficial. This also increases its reusability potential in other projects.*

*List data items alphabetically. Make each name a bookmark so each time the name occurs in this SRS it can be link to this entry via a hyperlink. Choose names with care. The expectation is that these names will persist in the design and implementation.]*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Element | Description | Composition or Data type | Length | Values |
| *Name of data item being defined* | *Textual description of the business meaning of the data element* | *For primitive data elements: data type (integer, floating point, alphabetic, date, etc.) and, as appropriate, format (e.g. date as MM/DD/YYYY).*  *For data structures show the components that comprise the structure. ,* | *Maximum number of characters for primitives; blank for structures* | *List of allowed values, default, rules governing legal values, and any other description of the data values* |
| *…* | *…* | *..* | *…* | *…* |

# Appendix D: Report Specification

*[This optional appendix contains descriptions of reports that the system needs to generate. Many applications involve generating reports from one or more databases, files or other information sources. Exploring the content and format of the reports needed is an important aspect of requirements develop. Describe the contents and layouts of each report, including changes being made in an existing version of the report. Indicate the conditions that will trigger generating the report (e.g., manual or automatic) the timing of report generation, and the disposition of the report, such as to whom it is sent or where it is stored.*

*Use the following template to document business rules.*

|  |  |
| --- | --- |
| Report ID: |  |
| Report Title: |  |
| Report Purpose: |  |
| Data Sources: |  |
| Frequency and Disposition: |  |
| Latency: |  |
| Visual Layout: |  |
| Header and Footer: |  |
| Report Body: |  |
| End-of-Report Indicator: |  |
| Interactivity: |  |
| Security Access Restrictions: |  |

If appropriate, provide a mock-up or a sample of the report, or an illustration of a similar existing report, showing the desired layout. ]

# Appendix E: Business Rules

*[This optional appendix describes business rules that are relevant to the proposed system. Use the following template to document business rules.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Rule Definition | Type of Rule | Static or Dynamic | Source |
| *BR-1* | *Definition 1* | *Fact, constraint, computation* | *Static or dynamic* | *Name, role or document* |
| *…* | *…* | *..* | *…* | *…* |

*]*

# Appendix F: Sample User Interface

*[If a sample user interface exists, place it here. Make it clear that this user interface is only an example. If something is required in the user interface, state that earlier in this document.]*

# Appendix G: Issues

*[This optional appendix is a dynamic list of the open requirements issues that remain to be resolved, including TBDs, pending.]*