[Project Name]

MTM Program Product

Software Design Description

Version [*versionNmbr*]

Issue Date: *Month X, YYY*

**Project Director:** *Xxxxxx Yyyyyyyyyyy*

**Project Manager:** *Xxxxxxx Yyyyyyyyyyy*

**Project Team:** *Xxxxxx Yyyyyyy Xxxxxx Yyyyyyy Xxxxxx Yyyyyy*

**Document Authors:** *Xxxxxxx Yyyyyyyy Xxxxxxx Yyyyyyyy*

**A Montana Tech Method Standard**

|  |  |  |  |
| --- | --- | --- | --- |
| *Version* | *Date* | *Authors* | *Comment* |
| 0.1 |  |  | First draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**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 quality 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 of Contents

1 Introduction 1

1.1 Purpose 1

1.2 Scope 1

1.3 Definitions, Acronyms, and Abbreviations 2

1.3.1 Definitions 2

1.3.2 Acronyms and Abbreviations 2

1.4 References 2

2 Design Overview 2

2.1 Background Information 2

2.2 Alternative Designs Considered 2

3 User Characteristics 2

4 System Architecture Error! Bookmark not defined.

5 Data Architecture 3

5.1 Data Analysis 3

5.2 Output Specifications 3

5.3 Logical Database Model 3

5.4 Data Conversion 3

6 Interface Requirements Error! Bookmark not defined.

6.1 Required Interfaces **Error! Bookmark not defined.**

6.2 External System Dependencies **Error! Bookmark not defined.**

7 User Interface Error! Bookmark not defined.

7.1 [Module X] Interface Design **Error! Bookmark not defined.**

8 Detailed Design 3

8.1 Description for Component n 4

8.1.1 Processing for Component n 4

8.1.2 Component n Interface Description 4

# Introduction

*[This Software Design Document (SDD) template is designed to facilitate the definition of processe, procedures, and work products relating to software design activities. This template was developed using IEEE Std 1016-1998, IEEE Recommended Practice for Software Design Descriptions which has been adapted for an undergraduage software engineering teaching environment. “The SDD shows how the software system will be structured to satisfy the requirements identified in the software requirements specification (SRS, IEEE Std 830-1998). It documents how its associated SRS requirements map into classes functions, interfaces, and data that when implemented will satisfy the requirements. In essence, the SDD becomes a detailed blueprint for the implementation activity.”*

*Information displayed in brackets is explanatory. Delete the bracketed text items and add your project-specific text. Bracketed information is food for thought on the section they address. The document organization shown here is the organization that will be used for MTM SDDs.*

*This introduction section should explain the purpose and scope of project software design, as well as, provide clarification of definitions, acronyms, and references. This section should also provide an overview of the project.]*

## Purpose

*[This subsection should explain the purpose for writing an SDD for this project and describe the intended audience for the SDD.]*

## Scope

*[If an SRS exists this subsection should just cite that document, otherwise this section 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.*

*Provide a description of the dominant design methodology. Provide a brief overview of the product architecture. Briefly describe the external systems with which this system must interface. Also explain how this document might evolve throughout the product lifecycle.]*

## Definitions, Acronyms, and Abbreviations

### Definitions

|  |  |
| --- | --- |
| xxx | yyy |

### Acronyms and Abbreviations

|  |  |
| --- | --- |
| SDD | Software Design Description |

## References

*[This subsection should list all references used within the SDD. All relationships to other plans and policies should be described as well as existing design standards.]*

# Design Background

## Background Information

[*This section should briefly present any background information relevant to the development of the software design. All stakeholders should be identified and their contact information provided. A description of associated project risks and issues may also be presented along with assumptions and dependencies critical to project success. Describe the business processes that will be modeled by the system. If this information has already been provided in an SRS for this product the information in the SRS may simply be referenced here.]*

## Alternative Designs Considered

*[All design alternatives considered, and the rationale for their non-acceptance, should be briefly addressed in this section.]*

# User Characteristics

*[Identify the potential system users. Specify the levels of expertise needed by the various users and indicate how each user will interact with the system. Describe how the system design will meet specific user requirements.]*

# Design Overview

*[This section should provide a description of the architectural design. All entities should be described as well as their interdependent relationships. A top-level diagram may be provided.]*

# Data Architecture

*[This section should describe the data structures to be used in support of the implementation. If these include databases, define the table structure including full field descriptions, relationships, and critical database objects. Graphical languages are appropriate. If this information is often provided in a separate Database Design Document, if this is the case simply refer to this document and omit the remainder of section 6.]*

## Data Analysis

*[A brief description of the procedures used in support of data analysis activities should be described in this section. Any analysis of the data that resulted in a change to the system design, or that impacted system design, should be noted.]*

## Output Specifications

*[All designs supporting requirements for system outputs should be described in this section. These may include designs to support reporting, printing, email, etc.]*

## Logical Database Model

*[Identify specific data elements and logical data groupings that are stored and processed by the design components in 6.  Outline data dependencies, relationships, and integrity rules in a data dictionary.  Specify the format and attributes of all data elements or data groupings.  A logical model of data flow, depicting how design elements transform input data into outputs should be developed and presented here. ]*

## Data Conversion

*[This section should describe all design requirements in support of data conversion activities. This may be covered in a separate data migration plan, but if it is not, it should be documented in the design documentation. The migration and validation of any converted legacy data should be described.]*

# Design

*[Decompose the system into design components that will interact with and transform data to perform the system objectives.  Assign a unique name to each component, and group these components by type, e.g., class, object, procedure.  Describe how each component satisfies system requirements.  In user terminology, specify the inputs, outputs, and transformation rules for each component.  Depict how the components depend on one another. Each component should be described as shown by the headers listed in section 6.1.]*

## Description for Component n

### Processing for Component n

### Component n Interface Description

Appendix A// Requirements Traceability Matrix

|  |  |  |
| --- | --- | --- |
| **Requirement Identifier** | **Requirement Description** | **SDD**  **Ref.** |
|  |  |  |
|  |  |  |
|  |  |  |