Title of the Project

by

Student’s Name…

This project is submitted to the Gannon University graduate faculty in

partial fulfillment for the degree Master of Science in Computer and Information Science.

Option: <Degree Option>

Approved:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| *Deacon Stephen T. Frezza, Ph.D., PSEM* |  | *Jeremy Cannell, MS* |
| Advising Professor in Charge of Research |  | Committee Member |
|  |  |  |
|  |  |  |
| *Mei-Huei Tang, PhD* |  | *Deacon Stephen T. Frezza, Ph.D., PSEM* |
| Committee Member |  | Chair, Computer and Information Science Department |
|  |  |  |

Gannon University

Erie, Pennsylvania 16541

December 2020

Acknowledgements

< The student thanks significant stakeholders and supporters of the research effort. Consider thanking mentors, colleagues, individuals or institutions supporting the research, family. If a explicit permission was given to reproduce or to use material in the research, those entities are named and the relevance of their permission identified and credited. >

Table of Contents

[Acknowledgements ii](#_Toc8392258)

[Abstract v](#_Toc8392259)

[List of Figures vi](#_Toc8392260)

[List of Tables vii](#_Toc8392261)

[1. Introduction 1](#_Toc8392262)

[1.1 Overview 1](#_Toc8392263)

[1.2 Curriculum Scope 1](#_Toc8392264)

[1.3 Key Stakeholder Needs 1](#_Toc8392265)

[1.4 Product Perspective 1](#_Toc8392266)

[1.5 Product Position Statement 2](#_Toc8392267)

[1.6 Summary of Capabilities 2](#_Toc8392268)

[1.7 Alternatives and Competition 2](#_Toc8392269)

[1.8 Project Management Plan 3](#_Toc8392270)

[1.9 References 3](#_Toc8392271)

[2. Requirements Management 4](#_Toc8392272)

[2.1 Requirements Development Perspective 4](#_Toc8392273)

[2.2 Use Characteristics 4](#_Toc8392274)

[2.2.1 User Classes and Characteristics 4](#_Toc8392275)

[2.2.2 Use-Case Model Survey 4](#_Toc8392276)

[2.2.3 User Documentation 5](#_Toc8392277)

[2.3 Feature Attributes 5](#_Toc8392278)

[2.3.1 Name\_of\_Attribute-1 6](#_Toc8392279)

[2.3.2 Name\_of\_Attribute-2 6](#_Toc8392280)

[2.3.3 Name\_of\_Attribute-3 6](#_Toc8392281)

[2.4 Key System Features 6](#_Toc8392282)

[2.4.1 Key System Feature 1 6](#_Toc8392283)

[2.4.2 Key System Feature 2 (*and so on*) 7](#_Toc8392284)

[2.5 Key Design and Implementation Constraints 7](#_Toc8392285)

[2.5.1 Operating Environment 7](#_Toc8392286)

[2.6 Interface Requirements 7](#_Toc8392287)

[2.6.1 User Interfaces 7](#_Toc8392288)

[2.6.2 Hardware Interfaces 8](#_Toc8392289)

[2.6.3 Software Interfaces 8](#_Toc8392290)

[2.6.4 Communications Interfaces 8](#_Toc8392291)

[2.7 Nonfunctional Requirements 8](#_Toc8392292)

[2.7.1 Performance Requirements 8](#_Toc8392293)

[2.7.2 Security Requirements 9](#_Toc8392294)

[2.7.3 Software Quality Attributes 9](#_Toc8392295)

[2.7.4 Safety Requirements 9](#_Toc8392296)

[2.7.5 Other Requirements 9](#_Toc8392297)

[2.8 Assumptions and Dependencies 9](#_Toc8392298)

[3. Design 10](#_Toc8392299)

[3.1 Introduction 10](#_Toc8392301)

[3.1.1 Goals and Objectives of Design 10](#_Toc8392302)

[3.1.2 Statement of Software Scope 10](#_Toc8392303)

[3.1.3 Software Context 10](#_Toc8392304)

[3.1.4 Major Design Constraints 10](#_Toc8392305)

[3.2 Data Design 10](#_Toc8392306)

[3.2.1 Major Internal Software Data Structure 10](#_Toc8392307)

[3.2.2 Global Data Structure 10](#_Toc8392308)

[3.2.3 Temporary Data Structure 10](#_Toc8392309)

[3.2.4 Database Description 11](#_Toc8392310)

[3.3 Architectural and Component-Level Design 11](#_Toc8392311)

[3.3.1 Program Structure 11](#_Toc8392312)

[3.3.2 Key Software Components 11](#_Toc8392313)

[3.4 User Interface Design 12](#_Toc8392314)

[3.4.1 Description of the User Interface 12](#_Toc8392315)

[3.4.2 Interface Design Rules 12](#_Toc8392316)

[3.4.3 Components Available 12](#_Toc8392317)

[3.4.4 User Interface Design Description 12](#_Toc8392318)

[3.5 Restrictions, Limitations, and Constraints 12](#_Toc8392319)

[4. Verification and Validation 13](#_Toc8392320)

[4.1 Test items 13](#_Toc8392322)

[5. Conclusion 14](#_Toc8392323)

[6. Bibliography 15](#_Toc8392324)

[Appendix A: Glossary 16](#_Toc8392325)

[Appendix B: Use Case Analysis 17](#_Toc8392326)

[Appendix C: Analysis Models 17](#_Toc8392327)

[Appendix D: Design Models 17](#_Toc8392328)

[Appendix E: Testing Log and Summary Status 17](#_Toc8392329)

[Appendix F: Screen Captures 17](#_Toc8392330)

[Appendix G: Project File Repository Definitions 17](#_Toc8392331)

Abstract

< Give a brief summary (100-300 words) of the key information presented in the document. Do not simply describe what each segment of the writing offers, but condense the significant aspect of each section into the key facts and primary end states, decisions made, or conclusions reached in each section. Succinctly state the aspects of what situation was addressed by the research, why was the situation significant to its stakeholder, what solution design was followed, what final development arose, what major difficulties were addressed, how successful was the research in achieving its goals.>

# List of Figures

[Figure 1. Maybe a Context Diagram 1](#_Toc1377697)

[Figure 2. Maybe something else 5](#_Toc1377698)

[Figure 3. This is another Caption 5](#_Toc1377699)

< *A listing of the figures or graphs displayed in the document are itemized here, including the page number of the appearance of the graphic. The following format pattern should be used in the listing and should be used as the graphic’s caption. The text of the caption should be boldface, all capital letters, the caption should appear under the graphic, and the numbering should be enabled through the* /References/Insert-Table-of-Figures *option of MS Word. “N” refers to the page number where the figure appears.*

*The List of Figures for this template has been generated above.>*

# List of Tables

[Table 1: Position Statement 2](#_Toc8392414)

[TABLE 2: Benefits and Supporting Features 2](#_Toc8392415)

< A listing of the tables displayed in the document are itemized here, including the page number of the appearance of the table. The following format pattern should be used in the listing and should be duplicated as the table’s caption. The text of the caption should be boldface, all capital letters, the caption should appear under the graphic, and the numbering should be enabled through the /References/Insert-Table-of-Figures option of MS Word. “N” refers to the page number where the table appears.

*The List of Tables for this template has been generated above.>*

>

1. Introduction

Overview

<Provide a **short description** of the software, system, process, or product – henceforth referred to as “product” -- being built or enhanced by the **technical directed project report (TeDPR)**. Describe its purpose, including relevant benefits, objectives, and goals. Relate the system to corporate goals or business strategies. This section is a relatively high-level, brief description.>

**Standard text is composed in Calibri, 11 pt. font, regular-face. Double space paragraphs. Paragraphs are denoted by a line separation. Justify the margins of standard text.** [1]

Throughout the template, guidance is given in italicized text, in angle-brackets. Remove the guidance text from your final manuscript.

Curriculum Scope

<Provide a description of how the learning experiences encountered in the development of TeDPR correlated to the curriculum option of your studies. >

Key Stakeholder Needs

<Describe the key problems or needs as perceived by the stakeholder(s). Clarify the issues of each problem by identifying its causes, solution approaches currently used, and users’ vision of the characteristics of the improved approach.>

Product Perspective

<Put the product in perspective to other related products and the user’s environment. If the product is independent and totally self-contained, state so. If the product is a component of a larger system, this subsection should relate how these systems interact and should identify the relevant interfaces among the systems. **Provide a context diagram here**. >

Figure . Maybe a Context Diagram



Product Position Statement

<Provide an overall statement summarizing, at the highest level, the unique position the product intends to fill in the marketplace. Communicate the intent of the application to all concerned.

**Include the table describing “For...Who...The...That...Unlike.”** Justify the contents of the cells to “flush left”.> But all tables and figures, like Table 1 should be cited in the text at least once. These tables and figures each require at least one paragraph describing what the reader should see in the table or figure.

|  |  |
| --- | --- |
| For | *(target customer)* |
| Who | *(statement of the need or opportunity)* |
| The *(****product name****)* | is a *(product category)* |
| That | *(statement of key benefit, that is, compelling reason to buy)* |
| Unlike | *(primary competitive alternative)* |
|  | *(statement of primary differentiation)* |

Table : Position Statement

Summary of Capabilities

<Summarize the major benefits and features the product provides. Organize and state the features so that the list is understandable to individuals initially becoming familiar with the product.

A simple table listing benefits and features may be sufficient. Justify the contents of the cells to “flush left”.>

***(product name)***

|  |  |
| --- | --- |
| **Benefit** | **Supporting Features** |
| *(benefit 1)* | *(feature 1)* |
| *(benefit 2)* | *(feature 2)* |
| *(benefit 3)* | *(feature 3)* |
|  | *<list may continue>* |
|  |  |

TABLE : Benefits and Supporting Features

Alternatives and Competition

<Identify alternatives perceived as available. These options can include buying a competitor’s product, building a homegrown solution, or simply maintaining the status quo. List any known competitive choices. List the major strengths and weaknesses of each option. State how the developed product compares. >

Project Management Plan

<Describe how you approached the development of the project. Explain difficulties you faced in any phase of the development. Discuss which aspects of the project management phases – problem identification, analysis, design, verification, or deployment -- were particularly challenging, why, and how did you address the challenges.>

References

<List **any other documents or Web addresses** to which this TeDPR refers. Include the “document-management/source-code control system” (DM-SCCS) address. These references may also include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

1. Requirements Management

*<Begin this major heading section and each subsequent “Heading 1” unit on a new page.* Normally, there is no text should appear here, between the headings 2 and 2.1.*>*

1. **Do not delete… Bug in document**

Requirements Development Perspective

<Describe the **context and origin** of the product being specified in this TeDPR. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the TeDPR defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.

Disclose challenges addressed in the definition of the requirements. Include a candid statement of aspects well-specified and areas where further work may need revision in later versions.>

Use Characteristics

User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be **differentiated based** on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. If applicable, distinguish the favored user classes from those which are less important to satisfy.>

Actor Survey

<*List actors identified in the use cases. Include a short description of the responsibilities and involvement of the actor to the system. Actors are listed in the user classes*.>

Use-Case Model Survey

<List and describe the use cases identified to relate the requirements of the system. Include a short description of its functionality to the system. The system use-case model appears here; other use-case models appear in the appendix. Remember to inline-cite all figures, like Figure 2 and Figure 3. Use the Insert Reference feature in Word.>

Figure . Maybe something else

<All figures should be referenced in the text, using the references feature. Normatively, these are only “Label and Number” style citations, like this reference to Figure 2. All figures need a paragraph illustrating what the reader should see in the figure. Figure captions could be centered below the figure. Table captions should be centered above the table. >

Figure . This is another Caption

User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that **are delivered** along with the software. Identify any known user documentation delivery formats or standards.>

Feature Attributes

*<Features have attributes that provide additional project information, used to evaluate, track, prioritize, and manage the product’s development. Identify each attribute, its meaning for this product, the expected values of the attribute, and the context when the value would be applied. A valid metaphor for this is to imagine each attribute as an enumerated set and its values as the values to the enumerated set. Keep the list of attributes manageable. NOTE: Most students don’t use these. >*

Name\_of\_Attribute-1

<Describe Attribute-1. List each of the potential values/states it may have. For each value/state, describe the circumstances for assigning that value to the attribute. >

Name\_of\_Attribute-2

<Describe Attribute-2. List ....>

Name\_of\_Attribute-3

<Describe Attribute-3. List ....>

Key System Features

<This template illustrates organizing the functional requirements for the product by system **features, the major services provided by the product**. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

Key System Feature 1

<Do not say “System Feature 1.” State the feature name in just a few words. Please include a reference label for it, but also include the name of the feature. >

Description

<Provide a short description of the feature.>

Attribute Classification

<Classify the feature according to the attribute classification described above.>

Key Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These items are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-x:

REQ-y:

Key System Feature 2 (*and so on*)

<Do not say “System Feature 2.” State the feature name in just a few words. Please include a reference label for it, but also include the name of the feature. >

Description

<Provide a short description of the feature.>

Attribute Classification

<Classify the feature according to the attribute classification described above.>

Key Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. >

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-r:

REQ-s:

Interface Requirements

User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This section may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification. If not applicable, include the section heading and for the text body use the phrase: Not applicable.>

Hardware Interfaces

<Describe the logical and physical characteristics **of each interface** between the software product and the **hardware components** of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used. If not applicable, include the section heading and for the text body use the phrase: Not applicable.>

Software Interfaces

<Describe the connections between this product and other specific software components (name and version), **including databases, operating systems, tools, libraries, and integrated commercial components**. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint. If not applicable, include the section heading and for the text body use the phrase: Not applicable.>

Communications Interfaces

<Describe the requirements **associated with any communications functions** required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms. If not applicable, include the section heading and for the text body use the phrase: Not applicable.>

Nonfunctional Requirements

<Each system has its own categories of nonfunctional requirements. If “Performance,” “Security,” “Software Quality,” or “Safety” are not applicable or meaningful to your system, then replace them as a category. While you may not have the explicit, following, categories, the product has nonfunctional requirements. Hence, the section has different headings for each TeDPR.

Within this section, explain the genesis and importance of the nonfunctional requirements to the project. >

Performance Requirements

<If various circumstances drive different performance requirements for the product, state them here and explain their rationale. Specify the timing relationships for real time systems. Make such requirements as specific as possible. **You may need to state performance requirements for individual functional requirements or features**.>

Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, **quantitative, and verifiable** when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

Safety Requirements

<Specify those requirements concerned with possible loss, damage, or harm that could result **from the use of the product**. Define any safeguards or actions to be taken, as well as actions to be prevented. Refer to any external policies or regulations which state safety issues affecting the product’s design or use. Define any safety certifications that must be satisfied.>

Other Requirements

<Define any other requirements not covered elsewhere in the TeDPR. These might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. **Add any new sections that are pertinent to the project**.>

Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the TeDPR. These factors could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. **The project could be affected if these assumptions are incorrect, are not shared, or change.** Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project.)

1. Design

Introduction

<This section provides an overview of the entire design document. This document describes all data, architectural, interface and component-level design for the software. Explain how the design decisions offered interesting or challenging consideration by you.>

Goals and Objectives of Design

<Overall goals and software objectives are described.>

Statement of Software Scope

<A description of the software scope is presented. Major inputs, processing functionality, and outputs are described without regard to implementation detail. Supporting diagrams appropriate for inclusion.>

Software Context

<The software is placed in a business or product line context. Strategic issues relevant to context are discussed. The intent is for the reader to understand the 'big picture'.>

Major Design Constraints

<Any business or product line constraints that will impact he manner in which the software is to be specified, designed, implemented or tested are noted here.>

Data Design

<A description of all data structures including internal, global, and temporary data structures.>

Major Internal Software Data Structure

<Data structures that are passed among components of the software are described.>

Global Data Structure

<Data structures that are available to major portions of the architecture are described.>

Temporary Data Structure

<Files created for interim use are described.>

Database Description

<Database(s) created as part of the application is(are) described. Include an ERD with structural rules governing its relationships and organization. Relational schema included in appendix. Major, unique, data definition challenges described here.>

Architectural and Component-Level Design

*<A description of the program architecture is presented. >*

Program Structure

<A detailed description the program structure chosen for the application is presented.

Architecture diagram

A pictorial representation of the architecture is presented.

Key Software Components

<A detailed description of key software components contained within the architecture is presented. The section is repeated for each of n components.>

Processing Narrative for <component name>

<Replace the <component name> in the title with the name of the component in the system. Provide a processing narrative for the component. Describe its role in the system.>

<Continue adding individual “Processing Narrative” sections for each key software component.>

<Address the following points in a degree appropriate to your project.>

* Component n interface description.

A detailed description of the input and output interfaces for the component is presented.

* Sub-Component n.m processing detail

A detailed algorithmic description for each sub-component within the component n is presented.

Repeat section; describe for each of the m sub-components of component n.

Optionally, choose to add the following elements, also.

* + **Interface description:** A description of sub-component m inputs and outputs is presented.
  + **Algorithmic model:** The pseudocode listing for sub-component m is presented.
  + **Restrictions/limitations:** The external environment and/or infrastructure that must exist for sub-component m to operate correctly is provided.
  + **Local data structures:** The data structures used within sub-component m are presented.
  + **Performance issues:** Information on topics that may affect the run-time performance, security, or computational accuracy of this sub-component are presented.
  + **Design constraints:** Attributes of the overall software design (including data structures, OS features, I/O, and interoperable systems) that constrain the design of this sub-component are presented.>
* External machine interfaces

Interfaces to other machines (computers or devices) are described.

* External system interfaces

Interfaces to other systems, products, or networks are described.>

User Interface Design

*<A description of the user interface design of the software is presented. In the following sections>*

Description of the User Interface

*<A detailed description of user interface including screen images or prototype is presented.*

Screen images

*<Discussion of the interface from the user's point of view.>*

Objects and actions

*<All screen objects and actions are identified. >*

Interface Design Rules

*<Conventions and standards used for designing/implementing the user interface are stated.>*

Components Available

*<GUI components available for implementation are noted.>*

User Interface Design Description

*<The user interface development system is described.>*

Restrictions, Limitations, and Constraints

*<Special design issues which impact the design or implementation of the software are noted here.>*

1. Verification and Validation

Test items

*<Summarize the software items (programs, modules, classes) and software features tested. The need for each item and its version/revision history may be included. Also specify characteristics of their transmittal media that impact hardware requirements or indicate the need for logical or physical transformations before testing can begin (e.g., programs must be transferred from tape to disk). Items that are to be specifically excluded from testing may be identified. >*

***<****In a narrative sections, focus on the key tested features. Address the points below.>*

***Features tested***

*Identify all software features and combinations of software features tested. Provide reference to requirements for each feature.*

***Features not tested***

*Identify all features and significant combinations of features that were not tested and the reasons.*

***Item pass/fail criteria***

*Specify the criteria used to determine whether each test item has passed or failed testing.*

***Testing tasks***

*Identify the set of tasks necessary to prepare for and perform testing. Identify all intertask dependencies and any special skills required.*

***Environmental needs***

*Specify both the necessary and desired properties of the test environment. This specification should contain the physical characteristics of the facilities including the hardware, the communications and system software, the mode of usage (e.g., stand-alone), and any other software or supplies needed to support the test.*

*Also specify the level of security that must be provided for the test facilities, system software, and proprietary components such as software, data, and hardware.*

*Identify special test tools needed. Identify any other testing needs (e.g., publications or office space). Identify the source for all needs that are not currently available to the test group.*

***Responsibilities***

*Identify the groups/person managed, designed, prepared, executed the test tasks.*

*These groups may include the developers, testers, operations staff, user representatives, technical support staff, data administration staff, and quality support staff.*

***Test Result Summary***

*Summarize the results of testing.*

*In an appendix, for each feature/combination of features tested, provide test cases used and the results of test cases, whether there are issues raised from the test result and the resolution of issues.>*

1. Conclusion

<The **Conclusion** should constitute at least four paragraphs. More are acceptable, but the material suggested is to be addressed.

<Paragraph: Identify the result of the research effort as either “successful” or “requiring further work.” Indicate major tests and their results to support the conclusion.

<Paragraph: Compare the final state of the work with the original intent sought when the project was proposed. Include how the project could be extended into further versions.

<Paragraph: Discuss the ethical elements of the product. Include such items as privacy aspects, stewardship needs, security concerns, data access and integrity perspectives. Describe how your product attended to these ethical elements. Include points where ethical features remain to be addressed.

<Paragraph: Describe realization or potential for realization of benefits to the organization. Indicate how the product enables these benefits.

*< Each appendix should appear. Each appendix should begin on a new page.*

*The text of an appendix should be in Calibri, 11 pt. font, double-spaced.*

*Even if no content should be included in an appendix, the appendix page should still appear. For instance, the project may not have a human interface component (Appendix F). Then the appendix title appears with the text: Not applicable.>*

1. Bibliography

|  |  |
| --- | --- |
| [1] | J. Doe, "Strange papers are worth citing," in *IEEE Conference on Strange Report Writing*, TimbukTu, 1995. |

< Using **IEEE citation format**, list all sources referenced to develop the work or explicitly cited in the writing. Online support for the format is available, EndNote supports it as does RefWorks. Note that you should reset the Style of the “Bibliography” section to “Heading 1” to ensure that it works with the TOC.>

Use the "Insert Citation" button to add citations to this document.

Appendix A: Glossary

< Define and alphabetize all the terms necessary to properly interpret the writing, including abbreviations, acronyms, and initialisms with their expanded terms and meanings. The reader should find any term – particularly abbreviations – used in the writing in the list. The format of each term should be similar to the following. Place sufficient tabs between the list of abbreviations and the terms so that the colons align; double space between abbreviations.

ACM : Association for Computing Machinery

: A professional organization of the computing field. ACM professionals provide guidance to standards-committees and to government in defining and understanding computing technology.

CIS : Computer and Information Science

: Department conferring graduate degree

TCP/IP : Transmission Control Protocol / Internet Protocol

: Communication protocol

>

Appendix B: Use Case Analysis

<Include use case model(s) and use case descriptions for the major, exemplary use cases developed in the project . Each separate figure and item presented in the appendix should be consecutively captioned as Figure B.N where “N” will increment beginning with 1.>

Appendix C: Analysis Models

<Include any pertinent analysis models such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams required for a complete understanding of the research. Each separate figure and item presented in the appendix should be consecutively captioned as Figure C.N where “N” will increment beginning with 1>

Appendix D: Design Models

< Include any pertinent design representations such as relational schemas, data dictionary, physical data flow diagrams, or navigation sequence diagrams Each separate figure and item presented in the appendix should be consecutively captioned as Figure D.N where “N” will increment beginning with 1.>

Appendix E: Testing Log and Summary Status

< Include any pertinent testing artifacts such as detailed test cases and test results. Each separate figure and item presented in the appendix should be consecutively captioned as Figure E.N where “N” will increment beginning with 1.>

Appendix F: Screen Captures

< Include screen images if the human interface components. Each separate figure and item presented in the appendix should be consecutively captioned as Figure F.N where “N” will increment beginning with 1.>

Appendix G: Project File Repository Definitions

< Include a listing of the directories and file designations used by the document management or source-code control system in order to reference the documentation and/or source code of the project.>