*Florida International University*

*School of Computing and Information Sciences*

CIS 4911 Senior Project – **Software Engineering Focus**

**Format of Deliverable 4 – Final Deliverable**

**The bullets represent the sections before chapter 1 “Introduction”, numbers on the left represent the corresponding chapters and sections.**

· Cover page – Name of course and section, name of system, project team number, group member names, date, and name of professor.

· Page with copyright and trademark notices, restrictions on copying or distributing the documentation, information for contacting the issuing organization (reader’s comments), warranties, contractual obligations or disclaimers, and general warnings and cautions.

· Executive Summary – one or two paragraphs giving a brief overview of the document.

· Table of Contents

1. Introduction

Introduce the introduction (one or two paragraphs)

1.1. Problem definition.

1.2. Scope of system.

1.3. Over all development methodology.

1.4. Definitions, acronyms, and abbreviations (at most one page).

1.5. Overview of document

2. Feasibility Study

2.1. Description of current system. Identify limitations and constraints

2.2. Description of alternative solutions considered.

2.3. Recommendation with explanation of why the solution was selected.

3. Project Plan

3.1. Project Organization

3.1.1. Project Personnel

3.1.2. Hardware and Software Resources

3.2. Identification of Tasks, Milestones and Deliverables

3.3. Cost of the Project

4. System Requirements

Introduce the proposed system (one or two paragraphs).

4.1. Functional and Nonfunctional Requirements – similar to RD

4.2. Requirements Analysis - similar to RD

5. System Design (i.e., overall system design)

Introduce the system design chapter (one or two paragraphs). State the uses cases you are implementing.

5.1. Overview – overview of system decomposition. Identify the architectural patterns used and state why they were selected.

5.2. Subsystem Decomposition - describe each of the major subsystems. Identify the use cases (or parts of use cases) associated with each subsystem. Refer to use cases in appendix B.

5.3. Hardware and Software Mapping – map subsystems to h/w and s/w.

5.4. Persistent Data Management – identify data that needs to be stored e.g., attributes of objects, and primary attributes (may use a table format).

5.5. Security/Privacy – describe user authentication processes, encryption of data, and use of firewalls or security servers.

6. Detailed Design

Introduce the detailed design chapter (one or two paragraphs).

6.1. Overview – briefly describe the behavior and structure of each subsystem. Describe the design patterns used and why they were selected.

6.2. Static model – detailed description of the structure for each subsystem. May include detailed class diagrams. Appendix C.

6.3. Dynamic model – state machine diagram for the main control object in each subsystem. Include the design of the algorithms used in the problem solution. Refinement of the sequence diagram from the analysis model. Appendix D.

6.4. Code Specification - describe the class interfaces (attributes and method signatures) and constraint (invariants, pre-condition and post-conditions). Code should be in Appendix E.

7. System Validation

Introduce the system validation chapter (one or two paragraphs).

7.1. Subsystem Tests – test each of the subsystems. This will involve the creation of a test drivers and stubs. Include the code for the test drivers and stubs in Appendix G.

7.2. System Tests - For each use case create at least 3 test cases, 2 sunny day and one rainy day, should include security test cases. Each test case should include: test case id, purpose, test setup environment, test inputs, and expected outputs.

7.3. Evaluation of Tests – evaluate how successful the tests were. Use a tabular form.

8. Glossary - define terms used in document, especially domain specific terms.

9. Appendix

9.1. Appendix A - Project schedule (Gantt chart or PERT chart).

9.2. Appendix B – All use cases with nonfunctional requirements.

9.3. Appendix C – User Interface designs.

9.4. Appendix D – Analysis models (static and dynamic)

9.5. Appendix E – Design models (static and dynamic)

9.6. Appendix F – Documented Class interfaces (code) and constraints.

9.7. Appendix G – Documented code for test drivers and stubs.

9.8. Appendix H – Diary of meeting and tasks for the **entire semester**.

10. References

**User’s Guide:**

1. Cover Page
2. Copyright and trademark information.
3. Introduction – brief description of application.
4. Hardware and Software Requirements.
5. Installation and setup
6. Getting Started - How to run application. Include user screens where appropriate and the description to run the two most important use cases.
7. Quick reference
8. Accessing online help
9. References

**Formatting Stuff:**

* **Start each chapter on a new page.**
* **Use 1.5 spacing.**
* **Chapter headings 16 pt.**
* **Section headings 13 pt.**
* **Text 11pt.**
* **Label each diagram.**
* **Use block format for test**

**You should hand in two (2) separate documents (System Documentation and User’s Guide) both of them properly formatted and bound.**

**You are also required to hand in a CD with ALL the artifacts for the semester, including: all presentations, documents, uml diagrams and source code.**

**Also include contact information (Tel #, address) in a text file in case we need to contact you.**