**High School Book Tracking Application - Requirement Specification Document**

**To develop the software requirements specification, you must understand the objects the system will manipulate (information domain), the services (functions) the system will perform, the constraints on the project (time, money and technical) and the performance expected (timing). As you develop the software requirements specification, you may need to re-interview the client. Do not hesitate to do so. Please E-mail with questions or come to my office hours if you need to.**

**The software requirements specification focuses on what the system will do, not how the system will be implemented. It is produced as the culmination of the software requirements engineering phase in the process model. You must analyze the information domain, the function, performance, behavior and interface requirements of the system.**

**Your document should follow the template below. It is a modified version of the Pressman's Adaptable Process Model template for a software requirements document.**

**1.0 Introduction**

**This section provides an overview of the entire requirement document. This document describes all data, functional and behavioral requirements for software.**

**1.1 Goals and objectives**

**Overall goals and software objectives are described.**

**1.2 Statement of scope**

**A description of the software is presented. Major inputs, processing functionality and outputs are described without regards to implementation detail. Rank the major processing functionality from the developer's point of view. Use a simple ranking system such as: essential, desirable and future requirements. This should represent what you think your team can accomplish in the time frame of a semester. The essential requirements, you are sure you can complete. The desirable requirements you hope to complete, but are not sure about. The future requirements, you have strong doubts about. Strive to balance the desires of your client with the reality of the time it takes to develop a SW product.**

**1.3 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'.**

**1.4 Major constraints**

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

**2.0 Usage scenario**

**This section provides a usage scenario for the software. It organizes information collected during requirements elicitation into use-cases.**

**2.1 User profiles**

**The profiles of all user categories are described here.**

**2.2 Use-cases**

**All use-cases for the software are presented.**

**2.2.1 Use-Case Diagram**

**A UML Use-Case diagram is presented.**

**2.2.2 Use-Case Descriptions**

**Each specific usage scenario is described. Follow the template found on page 190 of the text.**

**2.3 Special usage considerations**

**Special requirements associated with the use of the software are presented.**

**2.4 Activity Diagrams**

**The graphical representation of the workflow of all use-cases may be presented. (This section is optional.)**

**3.0 Data Model and Description**

**This section describes the information domain for the software.**

**3.1 Data objects**

**Data objects and their major attributes are described.**

**3.2 Relationships**

**Relationships among data objects are described.**

**3.3 Complete data model**

**Use UML class diagrams to develop an ERD-like model of the data objects and relationships.**

**4.0 Functional Model and Description**

**This section describes the static structure of the software.**

**4.1 Class diagrams**

**The class hierarchy (OO) is presented.**

**4.2 Software Interface Description**

**The software interface(s)to the outside world is(are) described.**

**4.2.1 External machine interfaces**

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

**4.2.2 External system interfaces**

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

**4.2.3 Human interface**

**An overview of any human interfaces to be designed for the software is presented.**

**5.0 Behavioral Model and Description**

**A description of the behavior of the software is presented.**

**5.1 Description for software behavior**

**A detailed description of major events and states is presented in this section.**

**5.1.1 Events**

**A listing of events (control, items) that will cause behavioral change within the system is presented.**

**5.1.2 States**

**A listing of states (modes of behavior) that will result as a consequence of events is presented.**

**5.2 Statechart Diagram**

**Depict the overall behavior of the system.**

**6.0 Restrictions, Limitations, and Constraints**

**Special issues which impact the specification, design, or implementation of the software are noted here.**

**7.0 Validation Criteria**

**The approach to software validation is described.**

**7.1 Classes of tests**

**The types of tests to be conducted are specified, including as much detail as is possible at this stage. Emphasis here is on black- box testing.**

**7.2 Expected software response**

**The expected results from testing are specified.**

**7.3 Performance bounds**

**Special performance requirements are specified.**