PROJECT_NAME

Software Requirements Specification
Version x.x

Team Members:
CS 471/472
Date
Table of Contents

Required – use word processing software to manage it.
Introduction (1)

This document represents the requirements for PROJECT_NAME. It will be used by the developers to design and implement their system and by the client to specify the system. This document may serve as the basis for a contract between these parties.

Purpose (1.1)

Describe the need for PROJECT_NAME in one paragraph. NOTE: all lengths given are guidelines – adjust if necessary or ask for advice.

Definitions (1.2)

Short list of important definitions, e.g. will, shall, item. Put a full list in the appendix (including these) if there are more than 5-10.

Document Conventions (1.3)

Any use of fonts/color meaning, etc.

References (1.4)

This includes things like templates, industry standards, books, etc.
General Requirements (2)

Overview (2.1)

*Describe what the system will do in 1-2 paragraphs.*

System Functions (2.2)

*List and describe each major function of the system using 1 paragraph. (e.g. customer can purchase items via a shopping cart)*

Users (2.3)

*Who are the users of the system? (e.g. customers, DB admin, product manager)*

Operating Environment (2.4)

*Where will this run – PCs, web browser, client-server, smart phone, etc?*
Requirements Specification (3)

Functional Requirements (3.1)

*List each functional requirement. Each one should be numbered for reference from other documents. Group as appropriate, e.g. all requirements related to the shopping cart or all requirements related to searching.*

Non-functional Requirements (3.2)

*List each non-functional requirement. Each one should be numbered for reference from other documents. Group as appropriate, e.g. requirements related to performance or requirements related to security.*
System Architecture (4)

Give a high-level overview of the anticipated system architecture, showing the distribution of functions across system modules. Architectural components that are reused should be highlighted. Clearly show what modules already exist and must be integrated properly.
User Stories (5)

These are mainly used with agile methods (which we will follow in CS 472) but it is a good idea to document them. This is simply a different way to look at and organize requirements. A sample user story would be “a customer adds several items to a shopping cart, enters their shipping address, provides payment information via a 3rd-party payment vendor and later receives their order from a shipping vendor.”
System Evolution

Describe the fundamental assumptions on which the system is based, and any anticipated changes due to hardware evolution, changing user needs, etc. This is extremely important for system designers as it may help them avoid design decisions that would constrain likely future changes to the system. For example, provide the current number of distinct items for sale and projections for the growth of that number.
Appendix

Revision History (A.1)

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Notes</th>
</tr>
</thead>
</table>

Glossary (A.2)