Design Document Template

Contents

[Introduction 1](#_Toc305679743)

[User requirement Analysis 1](#_Toc305679744)

[Architecture Overview 1](#_Toc305679745)

[Software Detailed Design 1](#_Toc305679746)

[Quality Assurance 2](#_Toc305679747)

[References 2](#_Toc305679748)

# Introduction

# User requirement Analysis

This section should identify all the external activities that trigger the system to react. Generally, it is recommended to:

* Identify all the “User Activities” that can arise due to external events (such as a user starting a process, or a system initiating a cron-job, etc) that would trigger a response from the system
* Draw a high-level diagram for expected input and output

# Architecture Overview

This section should design a high-level architecture of the system. It is recommended to have the following sections:

* High-level Component Model

For each component describe the following items:

* + Component Diagram
	+ Component Description
	+ Component Decomposition Chart
	+ Class Diagrams (search “UML Class Diagram” for more information)
* Inter-Component Interactions

Describe how components interact. This section should provide component-level interactions between all the activities that were identified in the section “User Requirement Analysis”. The user activities can now be decomposed into multiple “component activities”. For example, an RMI call from the server can be decomposed into activities performed at the stub, and the activities performed at the skeleton.

# Software Detailed Design

This section will have a detailed design for each component, and finally have a detailed design for the complete system

* Detailed Design for Component-n

Each component will have one section devoted to the detailed description of the component

* + Introduction/Purpose of this Component/Entity
	+ Input for this Component/Entity
	+ Output for this Component/Entity
	+ Component/Entity Process to Convert Input to Output
	+ Design constraints and performance requirements of this Component/Entity
	+ Process (pseudo-code algorithm)
* Detailed Sequence Diagram

For each activity identified in the section “User Requirement Analysis”, you will draw a detailed sequence diagram (search “UML Sequence Diagrams” for more information). The sequence diagram will show how each user triggers an activity of a method in a class, which in turn calls another method (possibly in another class). The diagram may also describe if the method call is synchronous or asynchronous.

# Quality Assurance

* Test Plans and Procedures
	+ Module/unit test
	+ Component test
	+ Integration test
	+ System verification test
* Test Cases

# References

We acknowledge the following sources

* bainusa.com/Documents/Detailed%20Design%20Document.doc
* http://wwwis.win.tue.nl/2R690/projects/spingrid/ddd.pdf
* ranger.uta.edu/~huber/cse4317/Docs/DDSContentSample.doc
* www2.cs.uidaho.edu/~jeffery/courses/383/SSDD\_Template\_A1.doc