System and Software Architecture Description (SSAD)

We Are Trojans (WAT) Network

Team01

Team members	**Roles**
Eirik Skogstad	Project Manager, Life Cycle Planner
Min Li	Feasibility Analyst, Operational Concept Engineer
Pittawat Pamornchaisirikij	NDI/NCS Acquirer & Evaluator, Tester
Punyawee Pakdiying	System Architect, Feasibility Analyst
Saloni Priya	Requirements Engineer, UML Modeler
Ameer Elkordy	IIV&V, Quality Focal Point
Suleyman Erten	Operational Concept Engineer, Requirements Engineer
Kamonphop Srisopha	Prototyper, UML Modeler

10/19/2014

Version History

Date	Author	Version	Changes made	Rationale
10/13/14	PP, SP	0.5	• Create initial SSAD document for Fundamental Commitment Package	• Used in Fundamental Commitment Package
10/19/14	PP, SP	0.6	• Update system context, artifact and information, and use-case diagrams	• Further understandings regarding the project and documents are acquired
10/19/14	PP, SP	0.7	• Update wording to have consistency across documents	• There is inconsistency in terminologies used in each document
10/19/14	PP, SP	1.0	• Update use-case diagrams and its course of action	• Update the document according to the comment in the ARB session and a better understanding toward the project

Table of Contents

Table of Tables

Table of Figures

1. Introduction

1.1 Purpose of the SSAD

- The report demonstrates the whole picture of the project, which includes a synopsis of the key features and people who will be involved in the “WAT” Network.
- The report summarizes the architectures, both software and hardware, used in the project.
- The report presents essential details about the system to be developed, and avoids the generic introduction relating to our project.
- The SSAD presents the system structure independent of the implementation technology, and provides a clear picture of what needs to be done rather than how things need to be done.

1.2 Status of the SSAD

Currently, we have updated the SSAD report to include the System Context diagram, Use Case diagram, and the some essential process of the system in accordance with to our project “WAT” Network.

2. System Analysis

2.1 System Analysis Overview

The primary purpose of "We Are Trojans" Network is to provide a platform where students can interact with fellow Trojans. The system provides users with an online forum, where users can interact via posting on the forum. The forum allows the users to comment on threads, like posts, and dislike posts. To encourage more and more people to join the forum, the system uses a WAT Points. The WAT Points are awarded to a particular user when other users like his post on the forum. The points can be earned to gain recognition on the leaderboard as well as can be used to redeem USC items/ USC Bookstore gift cards via the website.

2.1.1 System Context

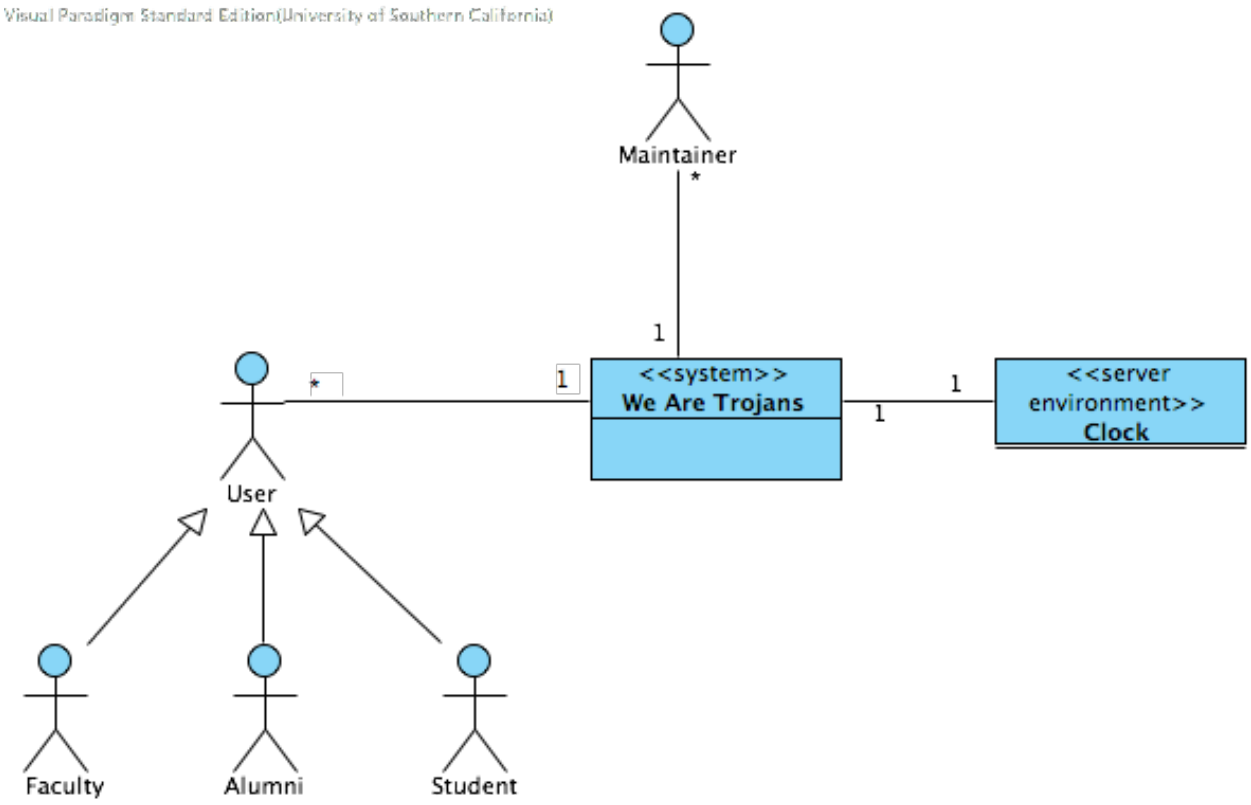

Figure 1: System Context Diagram

Table 1: Actors Summary

Actor	Description	Responsibilities
User (Student, Faculty, Alumni)	USC students, faculty, and alumni who participate in the Trojan network	• Start a thread and post on a thread. • Like, dislike a post/thread in the system to give credibility of both posts and threads • Redeem a gift card, items from points earned in the system • Update their own profiles reflecting their personal information
Maintainer	Selected personnel to maintain the system	• Review and delete rule-violating posts • Pin important posts • Create categories for the posts • Arrange posts to a categories • Manage users' accounts
Clock	System Clock	• Provide the system time

2.1.2 Artifacts & Information

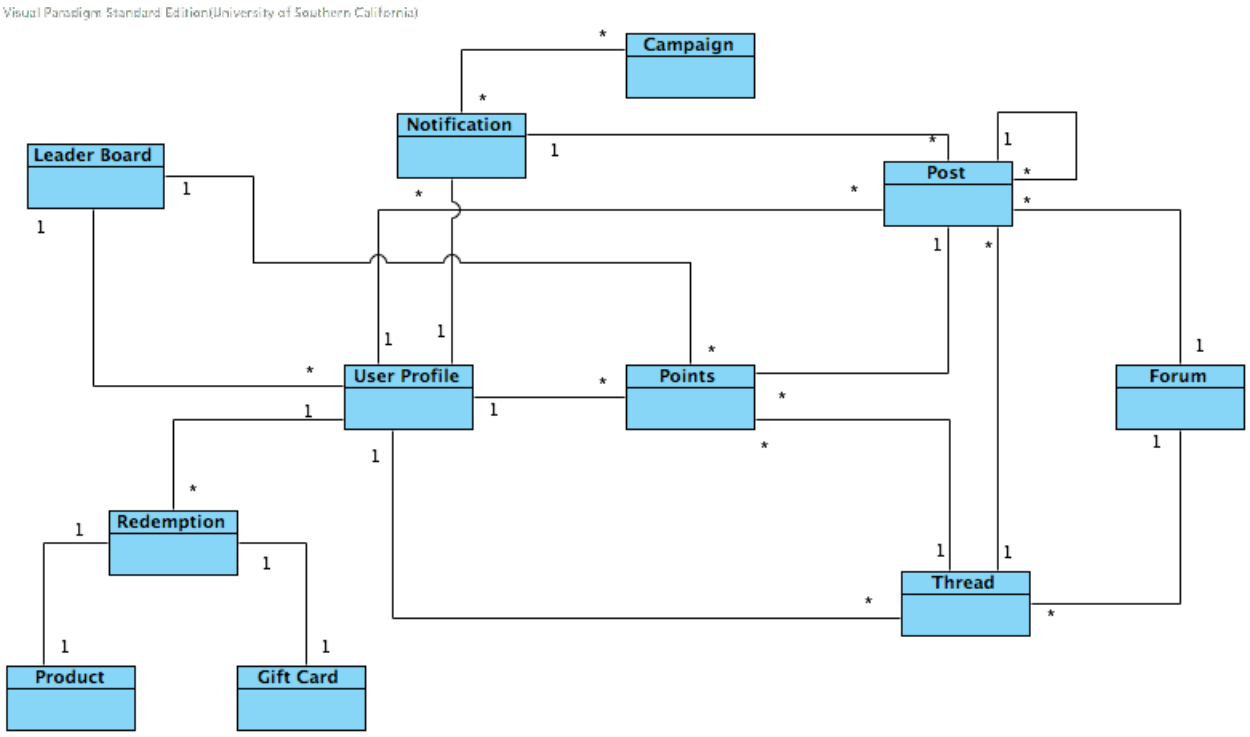

Figure 2: Artifacts and Information Diagram

Table 2: Artifacts and Information Summary

Artifact	Purpose
Leaderboard	Contain all information, personal profile, classes and points, about the user
Redemption	Contain all information regarding redemption for users
Product	Contain all information about items to be redeemed. This could include a list of available items and points for a particular item.
Gift Card	Contain all information about gift cards to be redeemed. This could include a list of available gift cards and points for a particular card.
User Profile	Contain all details about users. There is both prerequisite information set by a system and user-created fields for their special information.
Points	Contain all points in each system of a user.
Thread	Contain all thread posted by users. This includes a posting time, a title, and details of a particular thread.
Post	Contain all post created by users. This includes a posting time, a title, and details of a particular post.
Notification	This includes notification form threads, special events, and other possible notifications.

2.1.3 Behavior

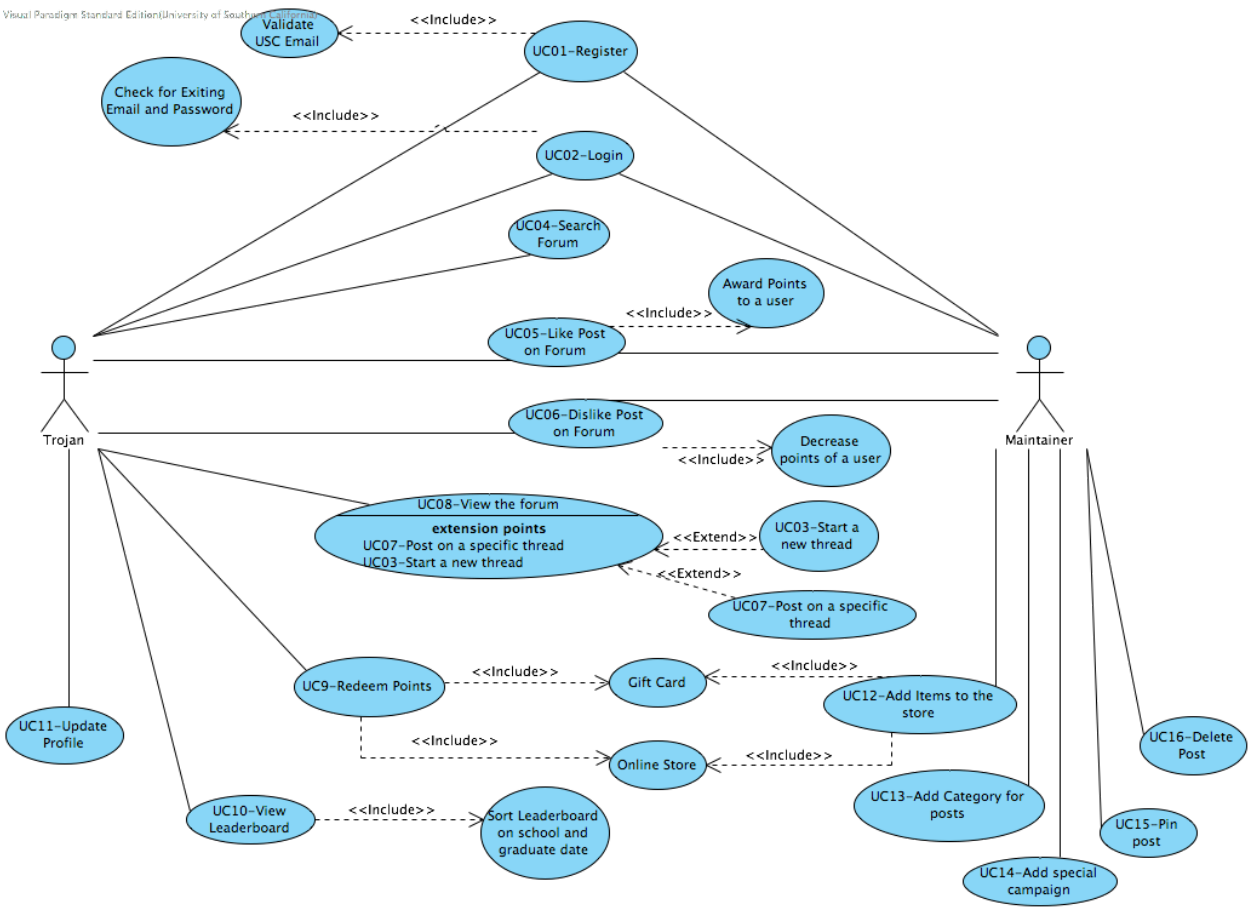

Figure 3: Use-Case Diagram for "We Are Trojans" Network System

2.1.3.1 Capability x

2.1.3.1.1 Process y

Table 3: Register Process

Actor	User
Identifier	UC01: Register new user
Purpose	By registering, user would have a valid user authentication to login to the system.
Development Risks	We will not be able to verify the user's email, which is essential to validate user's email.
Pre-conditions	User is not currently registered in the system.
Post-conditions	The user is registered, and can use the system with their USC email

<table>
<tr><td></td><td>and password.</td></tr>
<tr><td>Flow of events
• Typical course of action</td><td>
<table>
<tr><th>Seq#</th><th>Actor Input</th><th>System Response</th></tr>
<tr><td>1</td><td>User creates an account using USC-email</td><td></td></tr>
<tr><td>2</td><td></td><td>The system verifies the user’s email (whether it is a USC-email or not)</td></tr>
<tr><td>3</td><td></td><td>The system sends a verification email to the user’s email account</td></tr>
<tr><td>4</td><td>User verifies his/her email</td><td></td></tr>
</table>
</td></tr>
<tr><td>• Alternate course of action</td><td>
<table>
<tr><th>Seq#</th><th>Actor Input</th><th>System Response</th></tr>
<tr><td>1</td><td>User inputs non-USC email</td><td></td></tr>
<tr><td>2</td><td></td><td>The system shows red text on the email form say “only USC email”</td></tr>
</table>
<table>
<tr><th>Seq#</th><th>Actor Input</th><th>System Response</th></tr>
<tr><td>1</td><td>The user puts different passwords on the password form</td><td></td></tr>
<tr><td>2</td><td></td><td>The system shows red texts on the form say “Password mismatch”</td></tr>
</table>
</td></tr>
<tr><td>• Exception course of action</td><td>
<table>
<tr><th>Seq#</th><th>Actor Input</th><th>System Response</th></tr>
<tr><td>1</td><td>The user puts their information on the register form and click register</td><td></td></tr>
<tr><td>2</td><td></td><td>The system identify that the USC email is already registered.</td></tr>
<tr><td>3</td><td></td><td>The system show “the email is already registered email” on the email form</td></tr>
</table>
</td></tr>
</table>

Table 4: Login Process

Actor	User
Identifier	UC02: Login to the system
Purpose	In order to use the system, the user has to login to the system
Development Risks	

<table>
<tr><td>Pre-conditions</td><td>The user is registered in the WAT network, and has a valid email and password to log in to the system.</td></tr>
<tr><td>Post-conditions</td><td>The user is redirected to the WAT network homepage.</td></tr>
<tr><td>Flow of events
• Typical course of action</td><td>
<table>
<tr><th>Seq#</th><th>Actor Input</th><th>System Response</th></tr>
<tr><td>1</td><td>The user inputs his/her email and password to the login page.</td><td></td></tr>
<tr><td>2</td><td></td><td>The system verifies the email and password whether they are matched with the existing account.</td></tr>
<tr><td>3</td><td></td><td>The system redirects the user to the home page.</td></tr>
</table>
</td></tr>
<tr><td>• Exception course of action</td><td>
<table>
<tr><th>Seq#</th><th>Actor Input</th><th>System Response</th></tr>
<tr><td>1</td><td>The user puts incorrect email or password to the login page.</td><td></td></tr>
<tr><td>2</td><td></td><td>The system verifies the email and password whether they are matched with the existing account.</td></tr>
<tr><td>3</td><td></td><td>The system shows red texts indicating an incorrect the user’s email or password</td></tr>
</table>
</td></tr>
</table>

Table 5: Start a Thread Process

<table>
<tr><td>Actor</td><td>User</td></tr>
<tr><td>Identifier</td><td>UC03: User can start a thread</td></tr>
<tr><td>Purpose</td><td>The user starts a new thread on the system for other users to view, like, dislike, and post on this thread.</td></tr>
<tr><td>Development Risks</td><td>The NDI chosen to achieve this functionality is poorly matched.</td></tr>
<tr><td>Pre-conditions</td><td>The user is logged in the system and chooses to starts a new thread on the forum.</td></tr>
<tr><td>Post-conditions</td><td>The thread is posted on the forum for other users to view, like, dislike and post comments.</td></tr>
<tr><td>Flow of events
• Typical course of action</td><td>
<table>
<tr><th>Seq#</th><th>Actor Input</th><th>System Response</th></tr>
<tr><td>1</td><td>The user enters texts to be posted on the forum.</td><td></td></tr>
<tr><td>2</td><td>The user clicks the post</td><td></td></tr>
</table>
</td></tr>
</table>

		button on the forum.	
	3		The system checks the contents of the thread whether there are some words violating the rules of the forum or any system restricted statements, such as SQL injections or not.
	4		The system posts the thread on the forum.

- **Exception course of action**

Seq#	Actor Input	System Response
1	The user enters texts to be posted on the forum.	
2	The user clicks the post button on the forum.	
3		The system does not display the new thread on the forum.

Seq#	Actor Input	System Response
1	The user enters texts to be posted on the forum.	
2	The user clicks the post button on the forum.	
3		The system finds some words violating the rules of the forum
4		The system rejects the thread and displays errors

Seq#	Actor Input	System Response
1	The user enters texts to be posted on the forum.	
2	The user clicks the post button on the forum.	
3		The system finds some statements which can cause harm to the system
4		The system rejects the thread and displays errors

Table 6: Forum Search Process

Actor	User
Identifier	UC04: User can search the forum
Purpose	Users can search the forum for a particular thread they are interested in.

<table>
<tr><td>Development Risks</td><td>Clients and users might not appreciate the implementation.</td></tr>
<tr><td>Pre-conditions</td><td>• User is registered in the WAT system.
• User is logged in to the WAT system.</td></tr>
<tr><td>Post-conditions</td><td>The list of posts that is relevant to the search keywords is shown and sorted by relevance.</td></tr>
<tr><td>Flow of events</td><td></td></tr>
<tr><td>Typical course of action</td><td>
<table>
<tr><th>Seq#</th><th>Actor Input</th><th>System Response</th></tr>
<tr><td>1</td><td></td><td>The system prevents user to click search button</td></tr>
<tr><td>2</td><td>User puts their interested word in the search</td><td></td></tr>
<tr><td>3</td><td></td><td>When their are some text in the search form, the system allows the user to click search button</td></tr>
<tr><td>4</td><td>User clicks search button</td><td></td></tr>
<tr><td>5</td><td></td><td>System searches for the relevance posts and shows them to the user sorting by higher relevance to lower</td></tr>
</table>
</td></tr>
<tr><td>Exception course of action</td><td>
<table>
<tr><th>Seq#</th><th>Actor Input</th><th>System Response</th></tr>
<tr><td>1</td><td></td><td>The system prevents the user from clicking search button</td></tr>
<tr><td>2</td><td></td><td>User puts their interested key words in the search box</td></tr>
<tr><td>3</td><td></td><td>When their are some texts in the search form, the system allow user to click search button</td></tr>
<tr><td>4</td><td>User clicks the search button</td><td></td></tr>
<tr><td>5</td><td></td><td>System cannot find the relevance posts to the keywords. The system shows the “There is no relevance post” error</td></tr>
</table>
</td></tr>
</table>

Table 7: Process of Liking a Thread or Post

Actor	User
Identifier	UC05: User can like a thread or a post
Purpose	By liking a thread and comment, the user would give WAT points to the author of the threads and posts, and increase credibility of the user and the thread and comment. It is also an important part to create a competitive environment among peers. Posts with more likes

<table>
<tr><td></td><td>will be presented on top.</td></tr>
<tr><td>Development Risks</td><td>This functionality will be merged with the WAT point system. The development risk will mainly come from the WAT point system.</td></tr>
<tr><td>Pre-conditions</td><td>User enters the forum page and wants to give a like to the thread or a post he/she sees.</td></tr>
<tr><td>Post-conditions</td><td>1. After clicking a like button, the like button will be greyed out.
2. The number of likes in that thread/post goes up by one.
3. Semester points and total points of the owner of the thread or post calculated. If the post is less than one month old, the point will be in its pending period.</td></tr>
<tr><td>Flow of events
• Typical course of action</td><td>
<table>
<tr><th>Seq#</th><th>Actor Input</th><th>System Response</th></tr>
<tr><td>1</td><td>User clicks the like button</td><td></td></tr>
<tr><td>2</td><td></td><td>The system makes the like button greyed out.</td></tr>
<tr><td>3</td><td></td><td>The system makes the number of like in that thread/post goes up by one.</td></tr>
<tr><td>4</td><td></td><td>The system calculates points of the owner of the thread or post.</td></tr>
</table>
</td></tr>
</table>

Table 8: Process of Disliking a Thread or Post

<table>
<tr><td>Actor</td><td>User</td></tr>
<tr><td>Identifier</td><td>UC06: User can Dislike a thread or a post</td></tr>
<tr><td>Purpose</td><td>The dislike mechanism does opposite to the liking mechanism. Disliking a thread or a post means that the user does not see that thread or post useful. The thread or post with more dislike will sink down and eventually disappear.</td></tr>
<tr><td>Development Risks</td><td>This functionality will be merged with the WAT point system. The development risk will mainly come from the WAT point system.</td></tr>
<tr><td>Pre-conditions</td><td>User enters the forum page and wants to give a dislike to the thread or comment that are not useful to the community.</td></tr>
<tr><td>Post-conditions</td><td>1. After clicking dislike button, the dislike button will be greyed out.
2. The number of dislike in that thread/post goes up by one.</td></tr>
<tr><td>Flow of events
• Typical course of events</td><td>
<table>
<tr><th>Seq#</th><th>Actor Input</th><th>System Response</th></tr>
<tr><td>1</td><td>User clicks the dislike button</td><td></td></tr>
<tr><td>2</td><td></td><td>The system makes the dislike button greyed out</td></tr>
<tr><td>3</td><td></td><td>The system makes the number of dislike in that thread/</td></tr>
</table>
</td></tr>
</table>

			comment/post goes down by one.
	4		The system calculates points of the owner of the thread or comment.

2.1.4 Modes of Operation

The system will not have multiple modes. Therefore, no description could be stated in this section.

2.2 System Analysis Rationale

The major operational stakeholders of the system are the USC students, USC faculty and USC alumni. These are the users who will become the members of the system. The users will be authenticated by the system via USC email. The "WAT" Network profile would be created once the user is validated.

The points system is a critical feature of the system. It serves as the base for the development of other features of the system such as the leaderboard and the like/dislike functionality for a post. The users actions in our system are associated with earning points. The more the users participate with the system the more points they can earn and gain recognition on leaderboard or redeem items for store or redeem a gift card.

3. Technology-Independent Model

3.1 Design Overview

3.1.1 System Structure

<< This section should contain

- a UML hardware component class diagram
- a UML software component class diagram
- a UML deployment diagram
- If necessary, a class diagram for the system's supporting software infrastructure
- and descriptions of the hardware components, software components, and, if necessary, the supporting software infrastructure components of the technology/platform-independent system architecture

More information and example can be found in **ICM EPG> Task: Define Technology-Independent Architecture >>**

<<Hardware Component Class Diagram>>

Figure 4: Hardware Component Class Diagram

<<Software Component Class Diagram>>

Figure 5: Software Component Class Diagram

<<Deployment Diagram>>

Figure 6: Deployment Diagram

<<Optional: Supporting Software Infrastructure Diagram>>

Figure 7: Supporting Software Component Class Diagram

Table 9: Hardware Component Description

Hardware Component	Description

Table 10: Software Component Description

Software Component	Description

Table 11: Supporting Software Component Description

Support Software Component	Description

3.1.2 Design Classes

This section should contain:

- UML class diagrams showing all the boundary, entity, and control classes in the design of the system being developed
- and a description of each class in the diagram

More information and example can be found in **ICM EPG> Task: Define Technology-Independent Architecture >>**

3.1.2.1 <Classes n>

<<Design Classes Class Diagram>>

Figure 8: Design Class Diagram

Table 12: Design Class Description

Class	Type	Description

3.1.3 Process Realization

<< This section shows how the proposed architecture can be realized by constructing sequence diagrams. More information and example can be found in **ICM EPG> Task: Define Technology-Independent Architecture >>**

<<Process Realization Diagram>>

Figure 9: Process Realization Diagram

3.2 Design Rationale

<< This section should contain an explanation of how/why the architecture/design described in previous sections was chosen. More information and example can be found in **ICM EPG> Task: Define Technology-Independent Architecture >>**

4. Technology-Specific System Design

<< Once you know specific technology that you team is going to use, design the system and software architecture and document them in this section. >>

4.1 Design Overview

4.1.1 System Structure

<<Hardware Component Class Diagram>>

Figure 10: Hardware Component Class Diagram

<<Software Component Class Diagram>>

Figure 11: Software Component Class Diagram

<<Deployment Diagram>>

Figure 12: Deployment Diagram

<<Optional: Supporting Software Infrastructure Diagram>>

Figure 13: Supporting Software Component Class Diagram

Table 13: Hardware Component Description

Hardware Component	Description

Table 14: Software Component Description

Software Component	Description

Table 15: Supporting Software Component Description

Support Software Component	Description

4.1.2 Design Classes

4.1.2.1 <Classes n>

<<Design Classes Class Diagram>>

Figure 14: Design Class Diagram

Table 16: Design Class Description

Class	Type	Description

4.1.3 Process Realization

<<Process Realization Diagram>>

Figure 15: Process Realization Diagram

4.2 Design Rationale

5. Architectural Styles, Patterns and Frameworks

<< Describe any implementation architecture styles (e.g. the Prism style and 3-tier architecture), patterns (e.g. pipe-and-filter and client-server), or frameworks (e.g. Java and CORBA) used to describe the system architecture. >>

Table 17: Architectural Styles, Patterns, and Frameworks

Name	Description	Benefits, Costs, and Limitations