# State Transition Diagram – Report

**<<Team Number>>**

<<Team **Name**>>  
<<Team members>>

**<<Customer Organization>>**

<<Customer Name>>  
<<Customer Name>>  
<<Customer Name>>  
<<Customer Name>>

**CS 4310 Software Engineering I: Requirements Engineering**

Instructor: Dr. Oscar A Mondragon

TA: Leonel Lopes

# Change Summary

Table 2 presents the changes made between versions of this document.

Table 2: Change Summary Table

|  |  |  |
| --- | --- | --- |
| Date | Modifier (Team member name) | Description (Please provide sufficient details) |
|  |  |  |
|  |  |  |

# 

# 

# **Guidelines.**

<< Level 1. Please note that systems can show behaviors at different levels of abstraction. For example, at the system level (Human beings can be asleep, awake, or active), at the component level (the stomach subsystem/component may show some behaviors, and at the use case level, the vending candy machine shows behaviors to handle the vending transaction.

So to fill out the L1 section, your system may have STDs at the system, component, or UC level (while delivering a service). There is no predefined number of STDs L1 since it depends on the system of interest>>

<< Level 2. On one hand, the system’s behavior is not too complex and one diagram (level of abstraction) is enough; on the other hand, one or more states in your first diagram may have more behaviors inside them. So you need another diagram to show/decompose the sub-behaviors. This can be done by using super states. Inside a super state, you may show sub-states or you may have two families of sub-states and use orthogonal states.

So to fill out the L2 section, if one of the states at L1 may have sub-states (it is a super state), then draw the sub-states or their orthogonal states. You may consider in the example above that the sleep state at L1 may have the following sub-states: light sleep, deep slip, and REM sleep. The last three will be shown in an L2 diagram as a superstate Sleep with its sub-states. There is no predefined number of STDs L2 since it depends on the complexity of the behavior at the states in an STD L1 >>

# **1. State Transition Diagram L1**

<<Provide a brief description of a STD L1>>.

## Level 1 STD - <<System / componentName / UseCaseName >>

<<Provide a brief description of this STD>>

<<Insert the state transition diagram>>

## Level 1 STD - <<System / componentName / UseCaseName >>

<<Provide a brief description of this STD>>

<<Insert the state transition diagram

## Level 1 STD - <<System / componentName / UseCaseName >>

<<Provide a brief description of this STD>>

<<Insert the state transition diagram>>

# **2. State Transition Diagram Level 2:**

## Level 2 STD - <<System / componentName / UseCaseName >>

<<Provide a brief description of this STD L2>>

<<Insert the state transition diagram L2>>

## Level 2 STD - <<System / componentName / UseCaseName >>

<<Provide a brief description of this STD L2>>

<<Insert the state transition diagram L2>>