

Addis Ababa Institute of Technology

Center of Information Technology and Scientific Computing

Department of IT/SW Eng.

<Project Name>

Software Design Specification

Team Members

Advisors: Lisanu Tebikew

Fitsum Alemu

Date

Table of Contents

[List of Tables ii](#_Toc451061527)

[List of figures iii](#_Toc451061528)

[Definitions, Acronyms, Abbreviations iv](#_Toc451061529)

[1. Introduction 1](#_Toc451061530)

[1.1 Purpose 1](#_Toc451061531)

[1.2 General Overview 1](#_Toc451061532)

[1.3 Development Methods & Contingencies 1](#_Toc451061533)

[2. System design model 1](#_Toc451061534)

[2.1 Subsystem decomposition 1](#_Toc451061535)

[2.2 Hardware/software mapping 1](#_Toc451061536)

[2.3 Access control 1](#_Toc451061537)

[3. Object Model 2](#_Toc451061538)

[3.1 Class Diagram 2](#_Toc451061539)

[3.2 Sequence Diagram 2](#_Toc451061540)

[3.2 State chart Diagram 2](#_Toc451061541)

[4. Detailed Design 2](#_Toc451061542)

[References 3](#_Toc451061543)

# List of Tables

# List of figures

# Definitions, Acronyms, Abbreviations

# Introduction

## 1.1 Purpose

The purpose of System Design document is to translate the business requirements and business processes into a technical design that will be used to develop the application.

## 1.2 General Overview

Instructions: Briefly introduce the system context and the basic design approach or organization. Provide a brief overview of the system and software architectures and the design goals. Include the high-level context diagram(s) for the system and subsystems previously provided in the High-Level Technical Design Concept/Alternatives and/or Requirements Document, updated as necessary to reflect any changes that have been made based on more current information or understanding. If the high-level context diagram has been updated, identify the changes that were made and why.

## 1.3 Development Methods & Contingencies

Instructions: Briefly describe the method or approach used for the system and software design (e.g., structured, object-oriented, prototyping, J2EE, UML, XML, etc.). If one or more formal/ published methods were adopted or adapted, then include a reference to a more detailed description of these methods. If several methods were seriously considered, then each such method should be mentioned, along with a brief explanation of why all or part of it was used or not used. Describe any contingencies that might arise in the design of the system and software that may change the development direction. Possibilities include lack of interface agreements with outside agencies or unstable architectures at the time the SDD is prepared. Address any possible workarounds or alternative plans.

# System Architecture

## 2.1 Subsystem decomposition

UML Component Diagram (show at least three level of the system decomposition)

## 2.2 Hardware/software mapping

UML Deployment diagram

# 3. Object Model

## 3.1 Class Diagram

Provide a Unified Modeling Language (UML) based type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes.

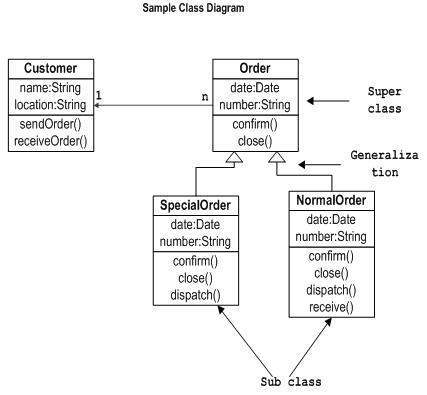
The following points should be remembered while drawing a class diagram:

* The name of the class diagram should be meaningful to describe the aspect of the system.
* Each element and their relationships should be identified in advance.
* Responsibility (attributes and methods) of each class should be clearly identified.
* For each class minimum number of properties should be specified. Because unnecessary properties will make the diagram complicated.
* Use notes whenever required to describe some aspect of the diagram. Because at the end of the drawing it should be understandable to the developer/coder.
* Finally, before making the final version, the diagram should be drawn on plain paper and rework as many times as possible to make it correct.

Now the following diagram is an example of an *Order System* of an application. So it describes a particular aspect of the entire application.

* First of all *Order* and *Customer* are identified as the two elements of the system and they have a *one to many* relationship because a customer can have multiple orders.
* We would keep *Order* class is an abstract class and it has two concrete classes (inheritance relationship) *SpecialOrder* and *NormalOrder*.
* The two inherited classes have all the properties as the *Order* class. In addition they have additional functions like *dispatch ()* and *receive ()*.

So the following class diagram has been drawn considering all the points mentioned above:



## 3.2 Sequence Diagram

Show how processes operate with one another and in what order. Depict the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

## 3.2 State chart Diagram (optional element)

UML State chart diagram (If you have an object that can be in many states only)

# 

# 4. Detailed Design

Here show the identified class in detail for example: -Assume a system has a Mobile Client Class then describe about the class in the following manner.

The classes represented here are the ones identified on your class diagram. But you must add the methods and classes identified in sequence and state chart diagram.

**Table: x Mobile Client class**

|  |
| --- |
| **Mobile Client** |
| +Name:String  + PhoneNumber:Integer  +Address:String  -Email:String  -ID:String |
| s+fillPersonal ()  +Beggar’sInfo ()  +SubmitInfo () |

**Table: x Attributes description for Mobile Client class**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Invariant |
| Name: | String | Public | Name <> NULL and must contain first, middle and last name and shouldn’t contain special characters and integers. |
| PhoneNumber | Integer | Public | PhoneNumber <> NULL must be 10 digits and must start by +251/09 |
| Address | String | Public | Address <>NULL and it must be between 12 to 20 characters |
| Email | String | Private | Email <> NULL   * Must contain @ * Must contain. (dot) * Position of @ >1 * Position of (dot)>position of @ + 2 * Position of (dot)+3<= total length of email address and the total character of the Email is at least 5 characters |

**Table: 7 Operation description for Mobile Client class**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Operation | Visibility | Return type | Argument | Pre-Condition | Post Condition |
| RegisterPersonalInfo | Public | void | . | The clients personal information shouldn’t exist | The clients personal information should exist |
| RegisterBeggar’sInfo | Public | void | . | The Beggar’s information shouldn’t exist | The Beggar’s information should exist |
| SubmitInfo | Public | void | - | Both the client’s and Beggar’s information shouldn’t exist | Both the client’s and Beggar’s information should exist |

# References

Bibliography

(list of book used for referance)

Web resource

(list of web pages you used as reference//address + access date )

<http://www.tutorialspoint.com/uml/index.htm> at May 5,2016(use this link for uml diagrams)