Password Generator

Analysis and Design Document

Student:Takacs Diana-Ingrid

**Group:30233**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mmm/yy> | <x.x> | <details> | <name> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

I. Project Specification 4

II. Elaboration – Iteration 1.1 4

1. Domain Model 4

2. Architectural Design 4

2.1 Conceptual Architecture 4

2.2 Package Design 4

2.3 Component and Deployment Diagrams 4

III. Elaboration – Iteration 1.2 4

1. Design Model 4

1.1 Dynamic Behavior 4

1.2 Class Design 4

2. Data Model 4

3. Unit Testing 4

IV. Elaboration – Iteration 2 4

1. Architectural Design Refinement 4

2. Design Model Refinement 4

V. Construction and Transition 5

1. System Testing 5

2. Future improvements 5

VI. Bibliography 5

# Project Specification

*[Present the project specification]*

# Elaboration – Iteration 1.1

# Domain Model

A domain model is a visual representation of real situation objects in a domain. A domain is an area of concern. Its used to refer to the area you are dealing with. The model is a diagram, for domain models the class diagram UML is mostly used. The class diagram is only used for the notation. The term domain model does not mean a set of diagrams describing software classes.

So for example imagine a store. For that store you want to build a brand new Point Of Sale system (lets call it POS system). A POS system is a computerized application used to record sale and handle payments. So you focus on the domain of the POS system. Now you will conceptualize the objects that will be used for this system. So you will get objects like: Sale, Payment, Register, Item etc. In a domain model you model these objects and draw associations between them so that you have an high level idea how this system will work.

# Architectural Design

## Conceptual Architecture

Conceptual architecture is a structural design that contains no implementation details. For example, a diagram that shows entities and relationships between them might be used to plan the structure of user interfaces, [software components](https://simplicable.com/new/software-components) or a data model.

Logical architecture gives as much detail as possible without constraining the architecture to a particular technology or environment. For example, a data model that includes key constraints but that doesn't include data types because these are often specific to a database.

## Package Design

Package diagrams are used to structure high level system elements. Packages are used for organizing large system which contains diagrams, documents and other key deliverables.

* Package Diagram can be used to simplify complex class diagrams, it can group classes into packages.
* A package is a collection of logically related UML elements.
* Packages are depicted as file folders and can be used on any of the UML diagrams.

## Component and Deployment Diagrams

Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed.

Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

# Elaboration – Iteration 1.2

# Design Model

## Dynamic Behavior

*[Create the interaction diagrams (1 sequence, 1 communication diagrams) for 2 relevant scenarios]*

## Class Design

*[Create the UML class diagram; apply GoF patterns and motivate your choice]*

# Data Model

*[Create the data model for the system.]*

# Unit Testing

*[Present the used testing methods and the associated test case scenarios.]*

# Elaboration – Iteration 2

# Architectural Design Refinement

*[Refine the architectural design: conceptual architecture, package design (consider package design principles), component and deployment diagrams. Motivate the changes that have been made.]*

# Design Model Refinement

## *[Refine the UML class diagram by applying class design principles and GRASP; motivate your choices. Deliver the updated class diagrams.]*

# Construction and Transition

# System Testing

*[Describe how you applied integration testing and present the associated test case scenarios.]*

# Future improvements

*[Present future improvements for the system]*

# Bibliography