<Chat Application >

Analysis and Design Document

Student:Ciontu Mihail & Pantea Paul

**Group: 30233**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <24/04/19> | <1.0> | <details> | <Ciontu Mihail & Pantea Paul> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

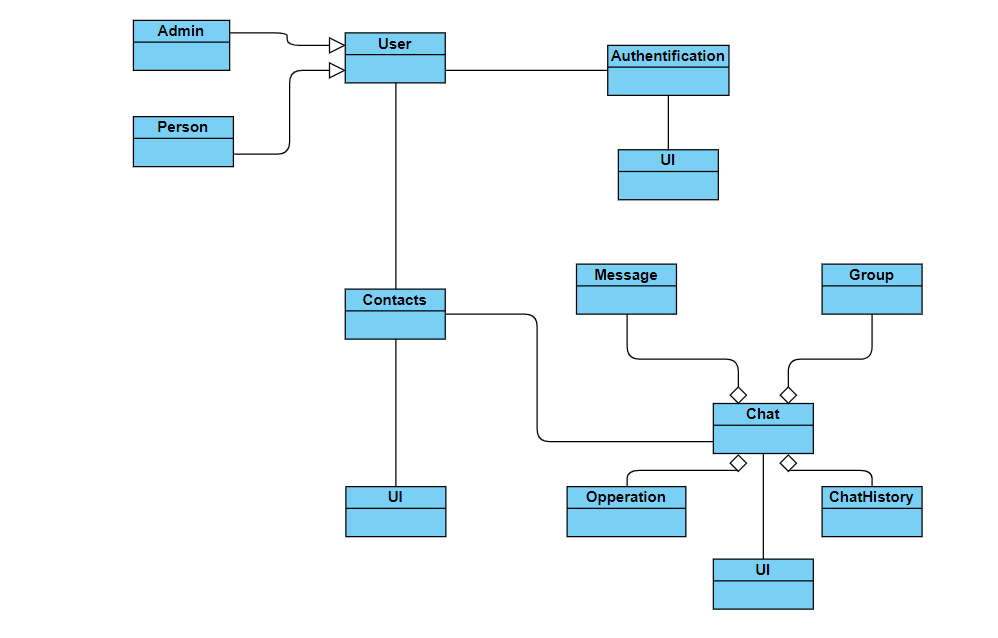
VI. Bibliography 5

# Project Specification

*[Present the project specification]*

# Elaboration – Iteration 1.1

# Domain Model



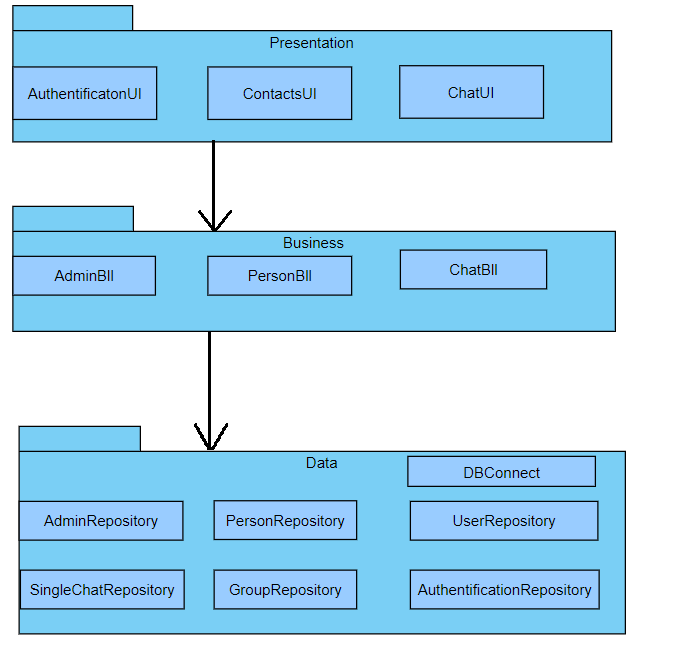
# Architectural Design

## Conceptual Architecture

The pattern that I used to implement the application is Layer Pattern Architecture, it is used to separate the classes into 3 main categories, presentation, business and database access.

Equipped with the knowledge of the layers to create, the relationships between them and the essence of the architecture, we are ready to implement it. As most of you probably expect, we will slice the system into layers by creating a separate package for each of them. When it comes to applying the dependency and separation rules, things are not so obvious. One could try putting each layer in a separate Maven module, but then capturing the weird relationship between domain and persistence would not be easy. I usually stick with packages and use common sense along with code reviews to make sure that none of the rules are broken.

## Package Design

**

## Component and Deployment Diagrams

# 

# 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