<Instant Messenger >

Analysis and Design Document

Student:Oltean Mihai

**Group:30238**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <07/04/2017> | <1.0> | The second presentation of the project documentaiton | Oltean Mihai |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

Create an application which allows users to communicate between them via messeges. It needs to use a server that handles the communications between users.

The user should do the following operations:

-login/logout

-register

-add people to contact list

-remove people from contact list

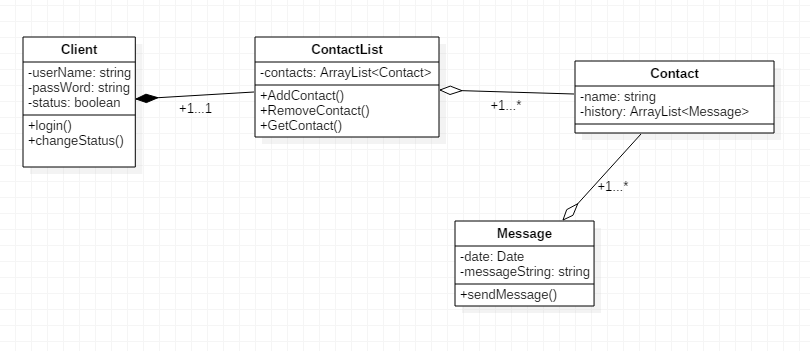
-send message to a contact.

The admin should do the following operations:

-do CRUD operations on users

# Elaboration – Iteration 1.1

# Domain Model

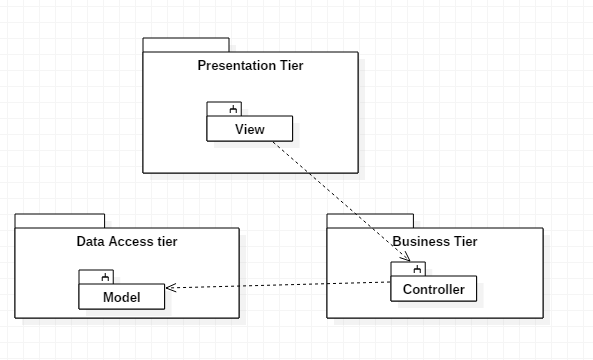


# Architectural Design

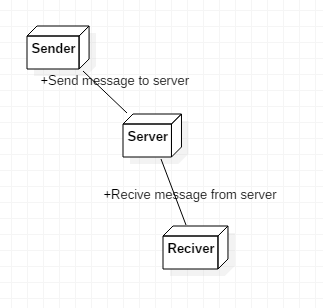
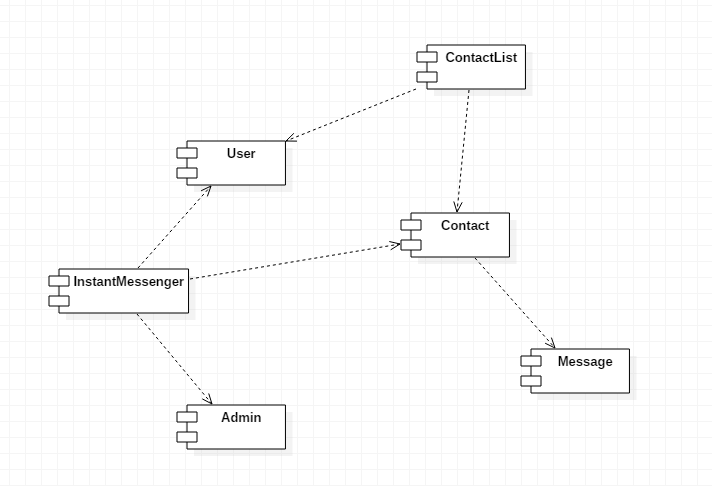
## Conceptual Architecture

The application has to implement the Three Tier and MVC Arhitectural Pattern. We use this because it maintain stability over the whole system, splitting the methods over multiple classes and use a logical order in term of usability

## Package Design



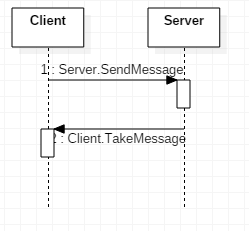
## Component and Deployment Diagrams



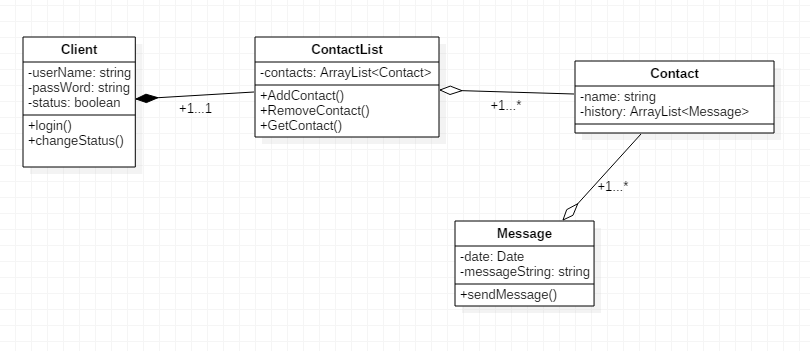
# Elaboration – Iteration 1.2

# Design Model

## Dynamic Behavior



## Class Design

**

# Data Model

The model used for this application will be the following:

Client – which represents the user that will connect to the server

Contact – will represent a persona of a client, through which he will communicate through

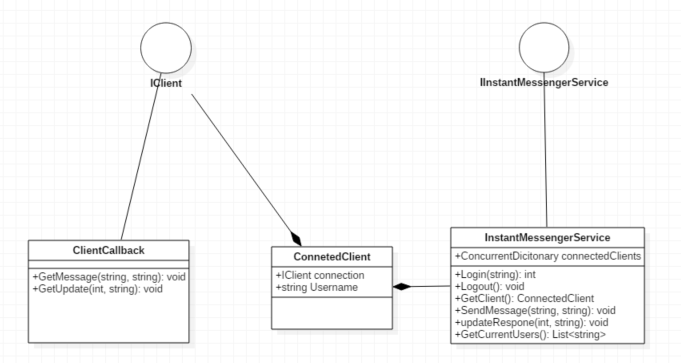
Message – The message will represent the channel which connect multiple users

# Elaboration – Iteration 2

# Architectural Design Refinement

I’ve been think to change the application to be a anonymous chatting messenger, where anyone can connect through an username, and can send messages to all the users connected to the messenger

# Design Model Refinement



I’ve reached to this UML diagram because I’ve made some changes using WCF and WPF

# Construction and Transition

# System Testing

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

# Future improvements

The application can be improved by transform it from an anonymous chat to an Whatsapp-like, application, for private instant messenger, where you can have contact lists and contacts.

Another improvement could be using a database to save the users and the chat history for every user.

# Bibliography