<Project Name>

Analysis and Design Document

Student:

**Group:**

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]*

# Develop a Java-based client-server messaging application for an intranet. The server component stores user data, handles connections, and supports messaging and file sharing. The client component allows registration, login, and logging off, with a GUI. Security measures include encryption and authentication.

Features:

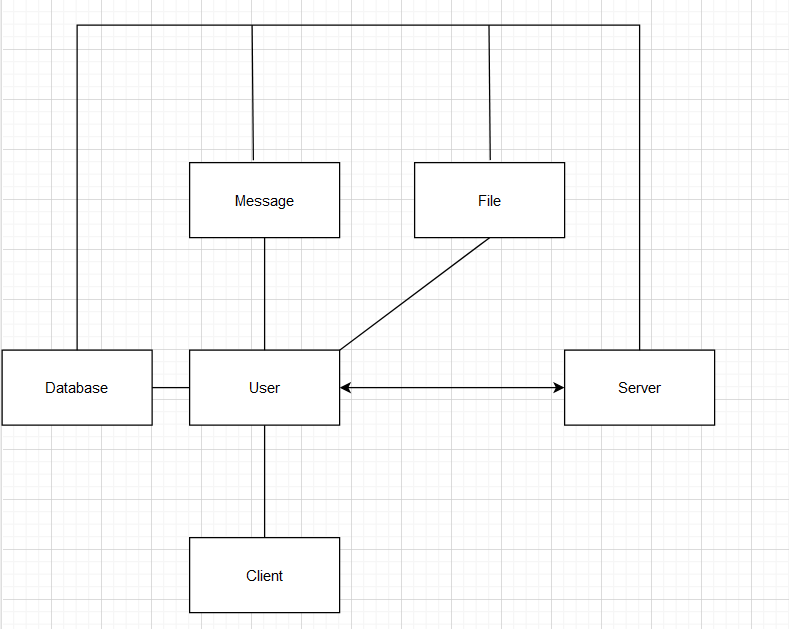
* Server Component:
  + Store user data in a secure database.
  + Handle incoming client connections and communication.
* Client Component:
  + Allow users to register and log in.
  + Display a list of online users.
  + Enable messaging and file sharing.
  + Support logging off from the application.

# Elaboration – Iteration 1.1

# Domain Model

*[Define the domain model and create the conceptual class diagrams]*

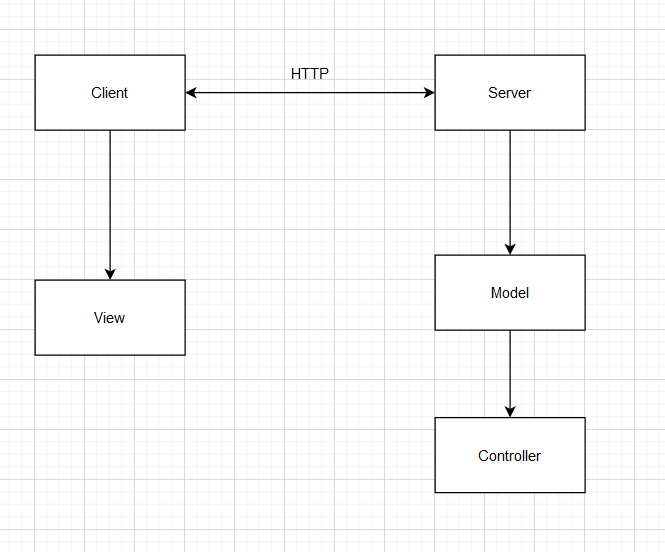
1. User: Represents a registered user of the messaging application. Contains attributes such as username, password, online status, and message history.
2. Server: Represents the server component of the application. Handles incoming client connections, stores user data in the database, and manages messaging and file sharing functionalities.
3. Client: Represents the client component of the application. Handles user registration, login, and messaging functionalities. Communicates with the server for sending/receiving messages and files.
4. Message: Represents a message sent between users. Contains attributes such as sender, receiver, message content, and timestamp.
5. File: Represents a file shared between users. Contains attributes such as file name, file size, sender, receiver, and timestamp.
6. Database: Represents the database used for storing user data, including user accounts and message history.



# Architectural Design

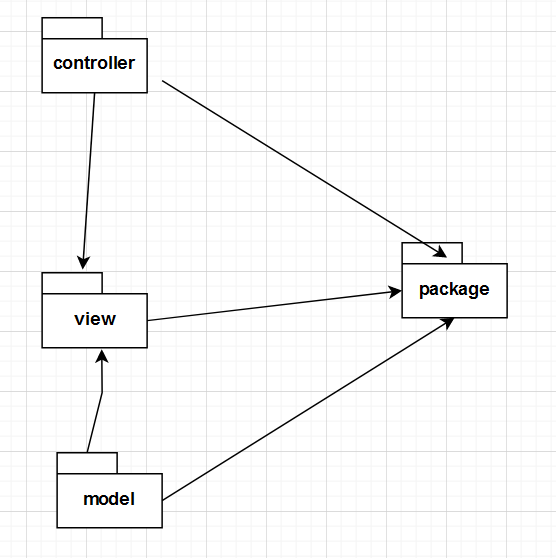
## Conceptual Architecture

*[Define the system’s conceptual architecture; use an architectural style and pattern - highlight its use and motivate your choice.]*



## Package Design

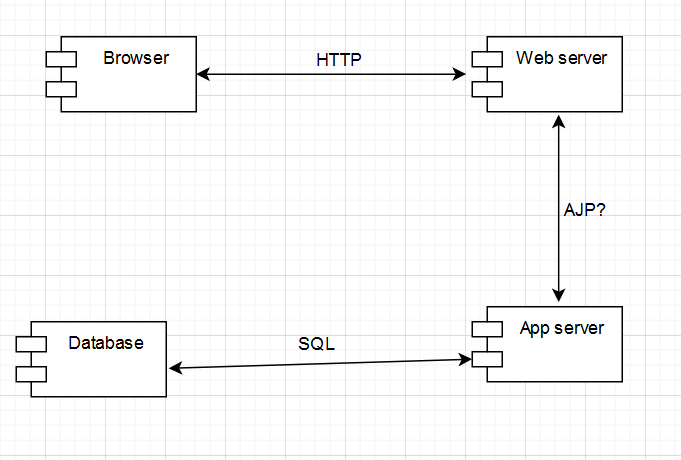
*[Create a package diagram]*



## Component and Deployment Diagrams

*[Create the component and deployment diagrams.]*

# 



# Elaboration – Iteration 1.2

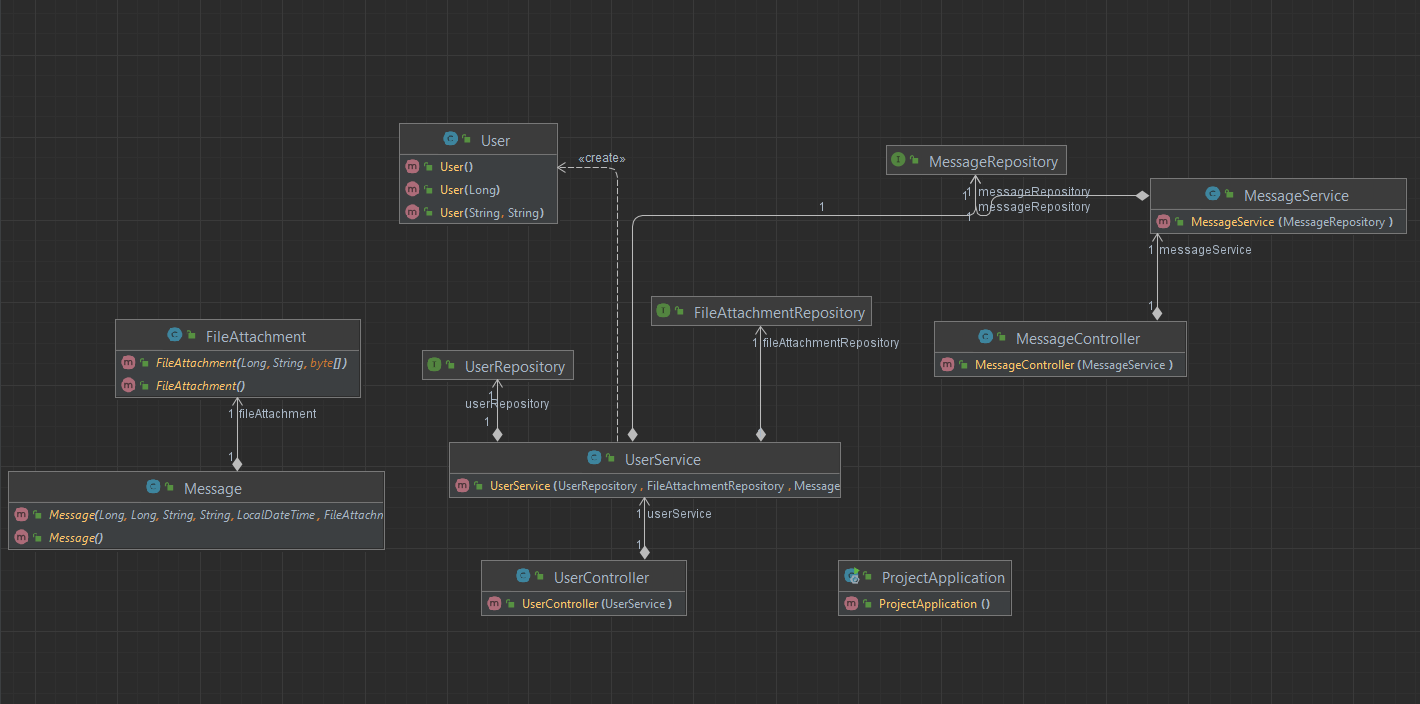
# Design Model

## Dynamic Behavior

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

## Class Design

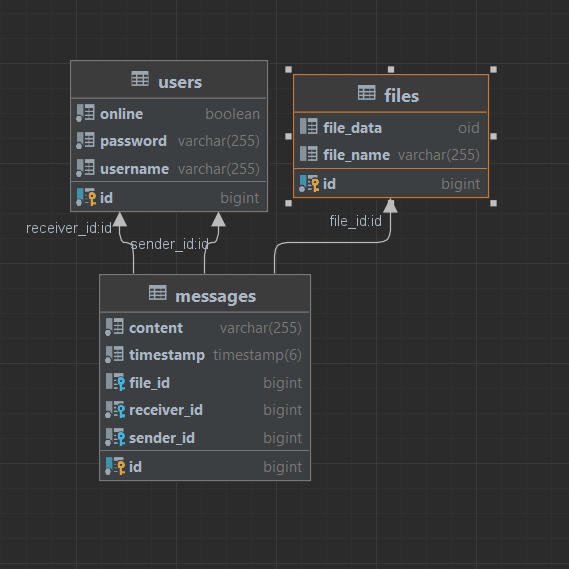
*[Create the UML class diagram; apply GoF patterns and motivate your choice]*



# 

# Data Model

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



# Unit Testing

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

The system has been tested by deploying both the backend and frontend and going through the different parts of the application step by step, such as logging in and sending messages. To simulate multiple users, multiple instances of the frontend can be created.

# 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