

# Howudoin Project

## Phase I

### Back-End Requirements Document

---

#### 1. Introduction

This document outlines the requirements for the backend development of a messaging application, similar to WhatsApp, using Spring Boot. Name of the project is Howudoin. The backend will handle user authentication, friend management, group management and messaging functionalities.

##### 1.1 Purpose

The purpose of this document is to define the functionalities and non-functional requirements for the backend system, which will serve as the foundation for the mobile messaging application.

##### 1.2 Scope

This document focuses on the backend functionalities including user registration, friend requests, and real-time messaging. The backend will be developed using Spring Boot and will interact with a NoSQL database for data storage.

#### 2. Functional Requirements

##### 2.1 User Registration and Login ( 10 Points )

The system will allow users to register and log in using their email and password. JWT-based authentication will be used to secure the login process.

##### 2.2 Friend Request System ( 20 Points )

Users will be able to send friend requests to each other. Once a request is accepted, users will be able to send messages to their friends.

##### 2.3 Messaging System ( 20 Points )

Users who have accepted each other as friends will be able to send and receive messages.

##### 2.4 Database Requirements ( 20 Points )

The backend will use MongoDB as the database to store user, group, and message data. MongoDB's document-based storage will allow flexible storage of messages, user information, and group data. Collections will include Users, Groups, and Messages.

## 2.5 Group Creation and Group Messaging

( 20 Points )

Users will be able to create groups and send messages to all members of a group. Each group will have a unique group ID, and messages sent in the group will be stored in MongoDB, associated with the group ID. Users will be able to view the group members and the message history for the group.

## 3. Non-Functional Requirements

### 3.1 Security

( 10 Points )

The backend must implement proper authentication and authorization mechanisms using JWT tokens.

## 4. API Specifications

The backend will expose several REST APIs to support user registration, friend requests, and messaging.

### 4.1 Endpoints and Methods

#### Public Endpoints

- POST /register: Register a new user (with name, last name, email, password)
- POST /login: Authenticate and login a user (with email and password)

#### Secure Endpoints

- POST /friends/add: Send a friend request
- POST /friends/accept: Accept a friend request ( If there is a friend request)
- GET /friends: Retrieve friend list
- POST /messages/send: Send a message to a friend
- GET /messages: Retrieve conversation history
- POST /groups/create: Creates a new group with a given name and members.
- POST /groups/:groupId/add-member: Adds a new member to an existing group.
- POST /groups/:groupId/send: Sends a message to all members of the specified group.
- GET /groups/:groupId/messages: Retrieves the message history for the specified group.
- GET /groups/:groupId/members: Retrieves the list of members for the specified group.

Project Deadline for Phase I : Nov 21<sup>st</sup> 2024

Project Deadline for Phase II : Dec 27<sup>th</sup> 2024