

# Microblogging Application

## Description:

A microblogging application provides users with a platform to share short-form content, typically limited to a certain character count per post. Here's a detailed description of the key features and components of a microblogging application:

- 1- **User registration:** Users can create accounts by providing necessary information such as username, email, and password.
- 2- **Profile Management:** Users can customize their profiles with personal information, profile pictures, cover photos, and bio descriptions. They can also manage privacy settings to control who can view their content and interact with them.
- 3- **Posting Content:** Users can create short posts, often referred to as "tweets," within the character limit defined by the platform.
- 4- **Interactions:** Users can engage with tweets by liking, retweeting, replying to, ,quoting them or bookmark . These interactions contribute to the engagement metrics of tweets and enable conversations between users. The application should support real-time updates to reflect interactions and new content.
- 5- **Timeline and Feed:** The application displays a timeline or feed where users can view tweets from other accounts . The feed is typically organized chronologically, with the most recent tweets appearing at the top. Users can scroll through their timeline to view updates from other accounts.

By incorporating these features and components, a microblogging application can provide users with a dynamic and engaging platform for sharing thoughts, ideas, and experiences in a concise and interactive form

## Functional Requirement

### Requirements Priorities:

Requirements can be ordered according to its priority by many different methods. one of the methods that we used called "Moscow scheme", which is applied as following:

- M: Must have requirement, which mean the basic requirement of the system that can't work without.
- S: Should have requirement, which mean the requirement can be added if it possible.
- C: Could have requirement, which mean it's not critical and can be neglected.

W: Will not have that requirement at this time, it can be made as development. We set the priority for each requirement in the previous tables.

**User Requirements Specification:** The system has 2 types of users' roles which is: (1Administrator, and 2-User), and every type of users has different basic functions.

Function Name:	Login
Description:	a function that uses ID and password which every type of users should use to login the website -open their account-.
Actors:	user -

Pre-Condition:	User's data is found on database.
Priority:	M

Function Name:	Sign-Up
Description:	A function allows user to register -have an account- on the website.
Actors:	user -.
Pre-Condition:	Valid email and password.
Priority:	M

Function Name:	Create a tweet
Description:	Users should be able to create and post messages, also known as "tweets," up to a certain character limit. The app should also allow users to attach media files, such as images or videos, to their tweets.
Actors:	user.

Pre-Condition:	User should have an account in the application.
Priority:	M

Function Name:	Retweet
Description:	User should be able to retweet (sharing an existing tweet).
Actors:	user.
Pre-Condition:	There should be a tweet.
Priority:	S

Function Name:	Quote tweet
Description:	User can post an old tweet with his comment.
Actors:	user.
Pre-Condition:	There should be a tweet.

Priority:	S
-----------	---

Function Name:	comment
Description:	User can give a comment on a tweet.
Actors:	user.
Pre-Condition:	There should be a tweet.
Priority:	S

Function Name:	reply
Description:	User can reply to a comment on a tweet.
Actors:	user.
Pre-Condition:	There should be a tweet.
Priority:	S

Function Name	Like a tweet.
---------------	---------------

Description	Users should be able to like other users' messages
Actors:	user.
Pre-Condition	There should be a tweet
Priority:	C

Function Name:	User profile.
Description:	The microblogging app should allow users to create and edit their profiles, including their profile picture, bio, and other personal information. The app should also display users' timelines, which show their own tweets and tweets from users they follow.
Actors:	user.

Pre-Condition:	User should have an account in the application.
Priority:	M

Function Name	News feeds.
Description	The news feed should display tweets from the users you follow in reverse chronological order, with the most recent tweets appearing at the top of the feed
Actors:	user.
Pre-Condition:	User should have an account in the application & user should follow another user.
Priority:	M

Function Name:	Edit profile
Description:	User can edit his profile



Actors:	user
Pre-Condition:	User should have an account in the application.
Priority:	M

Function Name:	change password
Description:	The user must able to change his password.
Actors:	user
Pre-Condition:	User should have an account in the application
Priority:	M

### System Requirements Specification:

Function's Name:	Assign to database function.
Description:	a function that assigned all data that must be assigned to the database.
Actors:	System.
Pre-Condition:	Valid data need to assign to the database.
Priority:	M

Function's Name:	Confirmation mail function.
Description:	a function that responsible to send a confirmation mail to the user after some processes is done.
Actors:	System.
Pre-Condition:	A process need confirmation mail like register and payment processes is done.
Priority:	W

Function's Name:	Show results function.
Description:	a function that responsible to view the search results depended on search process.
Actors:	System.

Pre-Condition:	A search process
Priority:	S

Function's Name:	Show comments function.
Description:	a function that responsible to send the users' comments to the admin.
Actors:	System.
Pre-Condition:	A comment had written by a user.
Priority:	C

Function's Name:	Error message's function.
Description:	a function that responsible to disappear an error message if a problem had found.
Actors:	System.
Pre-Condition:	A problem has happened during a process progressing
Priority:	S

# Non-Functional Requirement

## Categories of Non-Functional Requirements

The non-functional requirements can be divided in many ways, one of these is that we can divide it into three basic types every type can be divided to many secondary types:

1-Product requirements: Requirements determining that the product delivered will act in a specific manner. Like: security requirements, usability requirements, performance requirements, space requirements, etc.

2-Organisational requirements: Requirements arising from organizational policies. Like: development requirements, operational requirements, environmental requirements, etc.

3-External requirements: Requirements resulting from factors outside the program and its phase of growth. Like: ethical requirements, safety requirements, etc.

Function Name:	Check unique function.
Description:	a function that responsible to check that User's email and ID is not duplicated
Type:	Security requirement -under products requirements-.
Fit-Criteria:	There is no duplicated email or ID on the system.

Function Name:	Hide password function.
Description:	a function that responsible to hide the users' passwords.
Type:	Security requirement -under products requirements-.
Fit-Criteria:	Password is hashed when user write it and hashed on the database.

Function Name:	Hide sensitive information function.
Description:	a function that responsible to hide the users' sensitive information.
Type:	Security requirement -under products requirements-.
Fit-Criteria:	The sensitive information is not available to any one also the admin.

Function Name:	Connecting with administrator
Description:	The user can easily connect with administrator is any problem happened.
Type:	Efficiency requirement -under products requirements-.
Fit-Criteria:	The user had the reply in his comment for 24 hours.

Function Name:	Available servers and services
----------------	--------------------------------

Description:	
	The system servers and services should be available when it is requested by the user.
Type:	Reliability requirement -under products requirements-.
Fit-Criteria:	The servers and services are work 24/7 hours in the perfect performance.

Function Name:	Simple user interface
Description:	The user interface is simple and easy to use
Type:	Usability requirement -under products requirements-.
Fit-Criteria:	Users can use the website easily and can learn how to use the website at most on 30 minutes.
Function Name:	System maintenance
Description:	The system can be maintained easily because using object-oriented modules.



Type:	Performance requirement -under products requirements-.
Fit-Criteria:	If there are any founded problem in the system, it won't take more. than 24 hours to fix it.

Function Name:	Correct information
Description:	The system displays only correct information.
Type:	Reliability requirement -under products requirements -.
Fit-Criteria:	All product details on the system is true. Like: images, colors, and other details are like the real product.

Function Name:	Sizing
Description:	User can write only 280 characters in any tweet or comment.
Type:	Reliability requirement -under products requirements.
Fit-Criteria:	Users can use the website easily.

## scope architecture:

### By Technology Layers:

Frontend: Angular - This is the single-page application that users interact with for creating tweets, viewing feeds, etc.

Backend: Spring Boot - This layer handles server-side logic like user authentication, database interactions, and communication with other services.

Database: MySQL - This stores all your application data like users, tweets, followers, etc.

Containerization: Docker - This packages your application with all its dependencies into portable containers for easy deployment and scaling.

### By Functional Layers:

Presentation Layer: This is handled by Angular. It deals with how data is presented to the user and how the user interacts with the application.

Business Logic Layer: This is handled by Spring Boot. It implements the core functionalities of your Twitter application like user management, tweet processing, and interactions.

Data Access Layer: This interacts with the MySQL database. Spring Boot likely uses libraries like JPA or Spring Data to handle database operations

#### Overall Architecture:

Our project utilizes a well-established architecture pattern that combines popular technologies for frontend, backend, database, and containerization.

This layered approach promotes separation of concerns, making development, deployment, and maintenance easier.