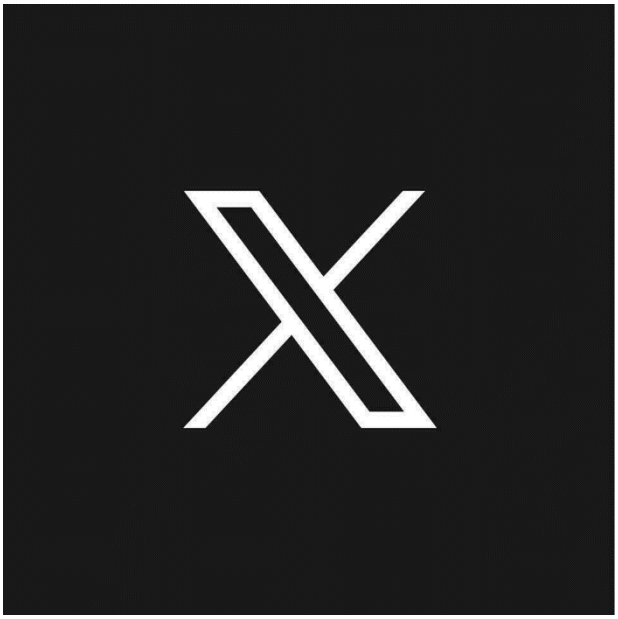
X CLONE

Requirements Specification



COS 301 Mini Project

Group MP9

Shiza Butt	u21451631
Ruan Carlinsky	u20416823
Mihail Dicoski	u
Brian Gates	u22516728
Jiahao Hu	u
Mekhail Muller	u21632792
Luke Nobrega	u22517244
Rebecca Oosthuizen	u20512008
Dawie Reyneke	u
Carter Shin	u21470864
Xadrian van Heerden	u

Table of Contents

NOTE:

Green text provides an explanation for what content should be placed in that section and will be removed once the actual content is filled in.

1. Introduction

1.1 Purpose of this Document

This software requirement specification acts as a detailed overview of the functionalities, features, requirements, and constraints of the project. Its purpose is to provide a communication medium between the development team and the project owner.

1.2 Product Scope

The product, *X Clone*, is designed to be a simplified replica of the social media platform X (previously known as Twitter). The system will replicate a selection of functionalities and features of the reference platform, as detailed below.

1.2.1 What the System Will Implement

- The system allows for user account registration and authentication through the use of usernames, passwords, and verification codes.
- Account authorization is implemented with different user roles.
- The system includes account and settings management, whereby users can adjust their profile information and visibility, deactivate, or delete their profile, and adjust application settings.
- Content posting and social interaction are included in the form of tweets, retweets, comments, followers, and collaborative posts.

1.2.2 What the System Will Not Implement

- The system does not provide an implementation of direct messaging, whereby users can have private, one-on-one or group conversations which are separate from the main tweeting platform.
- The system will not implement X's communities feature, which are exclusive groups where only specific users who are invited to the community may converse.
- The system will not implement an algorithm to determine what content should be displayed on a user's timeline and will instead display posts in reverse chronological order.
- The system will not implement a bookmarks feature, whereby users can save tweets for later viewing.

1.2.3 Objectives

The primary goal of this project is to create a simple, deployed clone of the social network X.

The objectives of *X Clone* are as follows:

- Facilitate online public discourse in a concise and interactive manner.
- Facilitate networking and collaboration between individuals with similar interests or professions.
- Facilitate information sharing and public visibility, particularly for businesses or organizations wanting to share news, updates, and other content with their audience.

1.3 SRS Overview

This document is laid out according to the IEEE SRS 830-1998 standard with structural alterations made with considerations to the project at hand. The biggest difference is that definitions and references appear in Section 4 as Appendices, instead of in this introductory section.

The following information is in this document:

- Section 2: General description of product, providing a background for its specific requirements.
- Section 3: Specific requirements of the product with functional requirements organized by user class.
- Section 4: Appendix, including a glossary and references.

2. General Description of Product

This section provides a background for the project by describing the factors that affect X Clone and its specific requirements, which are detailed in Section 3 of this document.

2.1 Product Perspective

This subsection of the SRS should put the product into perspective with other related products. A block diagram showing the major components of the larger system, interconnections, and external interfaces can be helpful.

This subsection should also describe how the software operates inside various constraints. For example, these constraints could include sections 2.1.1 to 2.1.8.

2.1.1 System Interfaces

List each system interface and identify the functionality of the software to accomplish the system requirement and the interface description to match the system.

2.1.2 User Interfaces

a) The logical characteristics of each interface between the software product and its users. This includes those configuration characteristics (e.g., required screen formats, page or window layouts, content of any reports or menus, or availability of programmable function keys) necessary to accomplish the software requirements.

b) All the aspects of optimizing the interface with the person who must use the system. This may simply comprise a list of do's and don'ts on how the system will appear to the user. One example may be a requirement for the option of long or short error messages. Like all others, these requirements should be verifiable, e.g., "a clerk typist grade 4 can do function X in Z min after 1 h of training" rather than "a typist can do function X." (This may also be specified in the Software System Attributes under a section titled Ease of Use.)

2.1.3 Hardware Interfaces

This should specify the logical characteristics of each interface between the software product and the hardware components of the system. This includes configuration characteristics (number of ports, instruction sets, etc.). It also covers such matters as what devices are to be supported, how they are to be supported, and protocols. For example, terminal support may specify full-screen support as opposed to line-by-line support.

2.1.4 Software Interfaces

This should specify the use of other required software products (e.g., a data management system, an operating system, or a mathematical package), and interfaces with other application systems (e.g., the linkage between an accounts receivable system and a general ledger system). For each required software product, the following should be provided:

— Name;

— Mnemonic;

— Specification number;

— Version number;

— Source.

For each interface, the following should be provided:

— Discussion of the purpose of the interfacing software as related to this software product.

— Definition of the interface in terms of message content and format. It is not necessary to detail any well-documented interface, but a reference to the document defining the interface is required.

2.1.5 Communications Interfaces

This should specify the various interfaces to communications such as local network protocols, etc.

2.1.6 Memory Constraints

This should specify any applicable characteristics and limits on primary and secondary memory.

2.1.7 Operations

This should specify the normal and special operations required by the user such as

a) The various modes of operations in the user organization (e.g., user-initiated operations);

b) Periods of interactive operations and periods of unattended operations;

c) Data processing support functions;

d) Backup and recovery operations.

NOTE—This is sometimes specified as part of the User Interfaces section

2.1.8 Site Adaptation Requirements

a) Define the requirements for any data or initialization sequences that are specific to a given site, mission, or operational mode (e.g., grid values, safety limits, etc.);

b) Specify the site or mission-related features that should be modified to adapt the software to a particular installation.

2.2 Product Functions

This subsection of the SRS should provide a summary of the major functions that the software will perform. For example, an SRS for an accounting program may use this part to address customer account maintenance, customer statement, and invoice preparation without mentioning the vast amount of detail that each of those functions requires.

Sometimes the function summary that is necessary for this part can be taken directly from the section of the higher-level specification (if one exists) that allocates particular functions to the software product. Note that for the sake of clarity:

a) The functions should be organized in a way that makes the list of functions understandable to the customer or to anyone else reading the document for the first time.

b) Textual or graphical methods can be used to show the different functions and their relationships. Such a diagram is not intended to show a design of a product, but simply shows the logical relationships among variables.

2.3 User Characteristics

This subsection of the SRS should describe those general characteristics of the intended users of the product including educational level, experience, and technical expertise. It should not be used to state specific requirements, but rather should provide the reasons why certain specific requirements are later specified in Section 3 of the SRS.

2.4 Constraints

This subsection of the SRS should provide a general description of any other items that will limit the developer's options. These include

a) Regulatory policies;
b) Hardware limitations (e.g., signal timing requirements);
c) Interfaces to other applications;
d) Parallel operation;
e) Audit functions;
f) Control functions;

g) Higher-order language requirements;
h) Signal handshake protocols (e.g., XON-XOFF, ACK-NACK);
i) Reliability requirements;
j) Criticality of the application;
k) Safety and security considerations.

There are a number of constraints by which the X Clone system must abide. The project's constraints, as detailed below, have been set by both the project owner, and the development team.

C1. X Clone must be implemented using Supabase.

C2. X Clone and its contents need to be appropriate for a professional environment.

C3. The frontend frameworks that may be used in the development of X Clone are limited to Angular, React, Vue, Svelte, or Flutter.

C4. The backend frameworks that may be used in the development of X Clone are limited to Vanilla JavaScript/Typescript/Python using the Supabase libraries with the option of additionally using NestJS/Express/Fastify/Python FAST API.

C5. The structure of X Clone must adhere to a client-server architecture.

C6. X Clone's design must be kept minimalistic and must exactly replicate that of the reference platform X.

C7. The development of X Clone must make use of an existing component library.

C8. X Clone's deployment must take place through a CI/CD process.

2.5 Assumptions and Dependencies

This subsection of the SRS should list each of the factors that affect the requirements stated in the SRS. These factors are not design constraints on the software but are, rather, any changes to them that can affect the requirements in the SRS. For example, an assumption may be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the SRS would then have to change accordingly.

3. Specific Requirements

This section of the SRS should contain all of the software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements. Throughout this section, every stated requirement should be externally perceivable by users, operators, or other external systems. These requirements should include at a minimum a description of every input (stimulus) into the system, every output (response) from the system, and all functions performed by the system in response to an input or in support of an output. As this is often the largest and most important part of the SRS, the following principles apply:

a) Specific requirements should be stated in conformance with all the characteristics described in 4.3.
b) Specific requirements should be cross-referenced to earlier documents that relate.
c) All requirements should be uniquely identifiable.
d) Careful attention should be given to organizing the requirements to maximize readability.

3.1 External Interface Requirements

This should be a detailed description of all inputs into and outputs from the software system. It should complement the interface descriptions in 5.2 and should not repeat information there. It should include both content and format as follows:

a) Name of item;
b) Description of purpose;
c) Source of input or destination of output;
d) Valid range, accuracy, and/or tolerance;
e) Units of measure;
f) Timing;
g) Relationships to other inputs/outputs;
h) Screen formats/organization;
i) Window formats/organization;
j) Data formats;
k) Command forma

3.2 Functional Requirements

The following functional requirements are organized by user class.

3.2.1 General Users

The *X Clone* system shall:

R1 Provide a secure user registration and authentication process to users.
- **R1.1** Allow users to create an account.
  - **R1.1.1** Require users to provide a unique username, password, and email address.
- **R1.2** Require users to verify their email address before they have full access to their account.
  - **R1.2.1** Send a verification code to the email address provided by the user.
  - **R1.2.2** Require the user to input this code.
  - **R1.2.3** Verify that the user's input matches with the emailed code.
  - **R1.2.4** Subsequently, give the user full access to their created account.

R2 Allow users to create and post content.
- **R2.1** Allow users to compose tweets with text that is limited to 280 characters.
- **R2.2** Allow users to attach images and videos to their tweets.
- **R2.3** Allow users to post their tweets to the platform.
- **R2.4** Allow users to edit or delete their own posted tweets.

R3 Allow users to view posted content.

R3.1 Provide a homepage which displays posts in reverse chronological order.
- **R3.1.1** Provide a "Following" timeline which exclusively displays posts from accounts that the user follows.
- **R3.1.2** Provide a "For you" timeline which displays recent tweets from all users on the platform.
- **R3.1.3** Enable a user to refresh their current timeline in order to display new tweets that have been posted from when the timeline was last loaded.

R4 Allow users to interact with content.
- **R4.1** Allow users to like tweets.
- **R4.2** Allow users to retweet tweets.
  - **R4.2.1** Post the retweeted tweet to the user's own profile and followers' timelines.
- **R4.3** Allow users to comment on tweets and can delete or edit their own comments.
  - **R4.3.1** Allow users to edit or delete their own comments.
- **R4.4** Track and display the timestamp, likes, comments, and retweets on each tweet.

R5 Allow users to view their own and other's profiles.
- **R5.1** Display a user's tweets, retweets, followers, and following list on their profile.

R6 Allow users to manage their profiles.
- **R6.1** Allow users to edit their profile picture, bio, display name, email address, and password.
- **R6.2** Allow users to manage the visibility of their profile.
  - **R6.2.1** Allow users to set their profile to public or private.
  - **R6.2.2** Adjust the profile's visibility based on the user's selection.
- **R6.3** Allow users to deactivate or delete their accounts.
  - **R6.3.1** Provide users with the option deactivate or delete an account.
  - **R6.3.2** Confirm the user's choice by requiring them to input their password.
  - **R6.3.3** Remove relevant data or adjust account visibility based on the user's choice.

R7 Allow users to edit their application settings.
- **R7.1** Include a settings page where users can edit visual and accessibility preferences.
- **R7.2** Implement users' preferences to their application.

R8 Allow users to receive and view notifications.
- **R8.1** Provide notification services to users for likes, comments, and new followers on their profile.
- **R8.2** Include a notifications page where users can view all their notifications.
- **R8.3** Implement toast notifications to display system notifications.

R9 Allow users to search for specific content and accounts.
- **R9.1** Include a search page with a search bar and filters.
- **R9.2** Display users and posts which match the search criteria.

R10 Provide a help page with contact information for where users can request assistance.

3.2.2 Premium Users

Requirements **R1** and **R3** - **R10** are applicable to premium users. Additions for this group are listed below.

The *X Clone* system shall:

R11 Allow users to create and post content.
- **R11.1** Allow users to compose tweets with text that is limited to 1000 characters.
- **R11.2** Allow users to attach images and videos to their tweets.
- **R11.3** Allow users to post their tweets to the platform.

R11.4 Allow users to edit or delete their own posted tweets.

R11.5 Allow users to create collaborative posts.

- **R11.5.1** Allow users to create a tweet and invite other users to collaborate on the tweet before it is posted.
- **R11.5.2** Send the invite to the other users and allow them to accept or reject the request.
- **R11.5.3** Post the tweet to the profiles and follower timelines of the collaborators who accepted the request.

3.2.3 Administrative Users

Requirements for admin.

3.3 Performance Requirements

This subsection should specify both the static and the dynamic numerical requirements placed on the software or on human interaction with the software as a whole. Static numerical requirements may include the following:

a) The number of terminals to be supported;

b) The number of simultaneous users to be supported;

c) Amount and type of information to be handled.

Static numerical requirements are sometimes identified under a separate section entitled Capacity.

Dynamic numerical requirements may include, for example, the numbers of transactions and tasks and the amount of data to be processed within certain time periods for both normal and peak workload conditions.

All of these requirements should be stated in measurable terms.

For example,

95% of the transactions shall be processed in less than 1 s.

rather than,

An operator shall not have to wait for the transaction to complete.

NOTE—Numerical limits applied to one specific function are normally specified as part of the processing subparagraph description of that function

3.4 Design Constraints

This should specify design constraints that can be imposed by other standards, hardware limitations, etc

3.5 Software System Attributes

There are a number of attributes of software that can serve as requirements. It is important that required attributes be specified so that their achievement can be objectively verified. The following are examples:

5.3.6.1 Reliability

This should specify the factors required to establish the required reliability of the software system at time of delivery.

5.3.6.2 Availability

This should specify the factors required to guarantee a defined availability level for the entire system such as checkpoint, recovery, and restart.

5.3.6.3 Security

This should specify the factors that protect the software from accidental or malicious access, use, modification, destruction, or disclosure. Specific requirements in this area could include the need to

a) Utilize certain cryptographical techniques;

b) Keep specific log or history data sets;

c) Assign certain functions to different modules;

d) Restrict communications between some areas of the program;

e) Check data integrity for critical variables.

5.3.6.4 Maintainability

This should specify attributes of software that relate to the ease of maintenance of the software itself. There may be some requirement for certain modularity, interfaces, complexity, etc. Requirements should not be placed here just because they are thought to be good design practices.

5.3.6.5 Portability

This should specify attributes of software that relate to the ease of porting the software to other host machines and/or operating systems. This may include the following:

a) Percentage of components with host-dependent code;

b) Percentage of code that is host dependent;

c) Use of a proven portable language;

d) Use of a particular compiler or language subset;

e) Use of a particular operating system.

3.6 Other Requirements

4. Appendix

4.1 Acronyms, Abbreviations, and Definitions

- Tweet
- Bio

4.2 References

IEEE Std 830-1998(R2009)