

Software Requirements Specification

for

Twitter Clone

Prepared by:

Nilusche Liyanaarachchi
<https://github.com/Nilusche>

FH Aachen University of Applied Sciences, Germany
Software Engineering

Aachen, 03 March 2022

Contents

1	Introduction	2
1.1	Purpose	2
1.2	Intended Audience	2
1.3	Product Scope	2
1.4	Definitions, Acronyms, and Abbreviations	2
2	Overall Description	3
2.1	User Classes and Characteristics	3
2.2	Operating Environment	3
2.3	Constraints	3
2.4	References	4
3	System Features and Requirements	5
3.1	Functional Requirements	5
3.1.1	Users should be able to post new tweets	5
3.1.2	The service should be able to create and display a user's timeline consisting of top tweets from all the people the user follows	6
3.1.3	A user should be able to follow other users	6
3.1.4	Users should be able to retweet tweets	7
3.1.5	Users should be able to mark tweets as favorites	7
3.1.6	Users should be able to edit their profile	8
3.1.7	Users should be able to reply to tweets	9
3.2	Non Functional Requirements	10
3.2.1	Performance Requirements	10
3.2.2	Database Requirements	10
3.2.3	Safety Requirements	10
3.2.4	Software Quality Attributes	10
3.3	Extended Requirements	10
3.4	External Interface Requirements	10
3.4.1	User Interfaces	10

3.4.2 Hardware Interfaces	11
Appendices	12
A Appendix: The Product As Is	13
A.1 Login	14
A.2 Register	14
A.3 Navigation	15
A.4 Tweet timeline	16
A.5 Trends and Search	18
A.6 Create new Tweet	19
A.7 Tweet detailed View	19
A.8 Follow Suggestions	20
A.9 Tweet Recommendations	20
A.10 Notifications	21
A.11 Profile Section	22
A.12 Messages	23

Chapter 1

Introduction

1.1 Purpose

Twitter is an online social networking service where users post and read messages called "tweets".

Registered users can post and read tweets, but those who are not registered can only read them.

Users can access Twitter through their website interface or as mobile app.

Twitter has grown exponentially fast in the past years and its purpose is to spread information fast. This Minimum Viable Product (MVP) exists for educational purposes - So that i can learn about the fundamentals of Software Design and to challenge my programmatic capabilities in the full-stack area.

1.2 Intended Audience

This document does not have a particular audience in mind. Ideally this document should address fellow software engineers that have experience in reading/writing Software Requirements Specifications (SRS). The MVP should interest Web-developers that operate on the entire stack.

1.3 Product Scope

The purpose of this clone is to provide the essential functionalities of twitter and above all to provide a similar user experience using this software.

1.4 Definitions, Acronyms, and Abbreviations

Term/ Acronym / Abbreviation	Expansion / Description
MVP	Minimum Viable Product
GUI	Graphical User Interface
NLP	Natural Language Processing
LDA	Latent Dirichlet Allocation
RAKE	Rapid automatic keyword extraction
TFIDF	Term Frequency Inverse Document Frequency

Chapter 2

Overall Description

2.1 User Classes and Characteristics

Different Users of the system should be able to have different permissions to access all functionality of the app, including:

- Deleting Users and Tweets

Standard users will have access to all functionality of the app except those listed above.

2.2 Operating Environment

Operating Environment for the MVP is as listed below.

- NoSql Database Firebase
- operating System: Windows.
- platform: Vue.js/Nuxt.js

2.3 Constraints

Since this MVP will only have a web-interface appropriate Web-Browsers like Firefox, Chrome, Edge should be used in their latest stable version.

2.4 References

1. IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.

Chapter 3

System Features and Requirements

3.1 Functional Requirements

3.1.1 Users should be able to post new tweets

Title: Create new tweet	Priority: high
As a user I want to create a new tweet so that it can get added to my timeline	

Actors: User with appropriate permissions

Preconditions:

1. User needs to be authorized

Trigger: Clicking the menu "Create new Tweet"

Procedure:

1. User fills in fields: "Tweet content" (required), Image field" (optional)
2. User clicks on Button "Create Playlist"
3. System generates new database record for tweet
4. System uploads Image to database if given
5. System adds Tweet to Timeline
6. System redirects user to homepage

Postconditions:

1. A new tweet record in database

Exceptions:

1. Database error
2. Required fields error

3.1.2 The service should be able to create and display a user's timeline consisting of top tweets from all the people the user follows

Title: Show Timeline	Priority: high
As a user I want to see my Timeline so that I can see my likes and retweets of those	

Actors: User with appropriate permissions

Preconditions:

1. User needs to be authorized

Trigger: Navigating to site "Home"

Procedure:

1. User clicks on Navigation "Home"
2. System fetches Tweets of User (Following Tweets too)

Postconditions:

1. Timeline with tweets

Exceptions:

1. Database error

3.1.3 A user should be able to follow other users

Title: Follow other Users	Priority: high
As a user I want to follow other users so that I can have their tweets on my timeline	

Actors: User with appropriate permissions

Preconditions:

1. User needs to be authorized

Trigger: Navigating to the Profile of another user and clicks on follow

Procedure:

1. System adds new follow relationship between two users
2. System updates User timeline

Postconditions:

1. Timeline with updated tweets

Exceptions:

1. Database error

3.1.4 Users should be able to retweet tweets

Title: Retweet tweets	Priority: medium
As a user I want to retweet a tweet so that it is distinguished from the other tweets in my timeline	

Actors: User with appropriate permissions

Preconditions:

1. User needs to be authorized

Trigger: Clicking the "retweet" Button

Procedure:

1. System adds new "retweet" relationship between user and tweet
2. System updates tweet visual

Postconditions:

1. Timeline with retweeted tweet

Exceptions:

1. Database error

3.1.5 Users should be able to mark tweets as favorites

Title: favor tweets	Priority: medium
As a user I want to favor a tweet so that it is distinguished from the other tweets in my timeline	

Actors: User with appropriate permissions

Preconditions:

1. User needs to be authorized

Trigger: Clicking the "favor" Button

Procedure:

1. System adds new "favor" relationship between user and tweet
2. System updates tweet visual

Postconditions:

1. Timeline with retweeted tweet

Exceptions:

1. Database error

3.1.6 Users should be able to edit their profile

Title: Edit Profile	Priority: medium
As a user I want to change my profile picture and twitter bio so that it is updated on my profile	

Actors: User with appropriate permissions

Preconditions:

1. User needs to be authorized

Trigger: Clicking the "Edit Profile" Button

Procedure:

1. User fills in fields: "Bio" (optional), Image field" (optional)
2. User clicks on Button "Update Profile"
3. System updates User Profile

Postconditions:

1. Profile with changed Information

Exceptions:

1. Database error

3.1.7 Users should be able to reply to tweets

Title: Reply to tweet	Priority: medium
As a user I want to reply to a tweet so that it is updated on my profile	

Actors: User with appropriate permissions

Preconditions:

1. User needs to be authorized

Trigger: Clicking the "Edit Profile" Button

Procedure:

1. User fills in fields: "Content" (required)
2. User clicks on Button "Tweet"
3. System adds Tweet to timeline
4. System adds Parent-Tweet to replyTo Property of Tweet

Postconditions:

1. Timeline with Tweet

Exceptions:

1. Database error

3.2 Non Functional Requirements

3.2.1 Performance Requirements

1. The service needs to be highly available
2. Acceptable latency to generate the timeline is 200ms

3.2.2 Database Requirements

The underlying database needs to be normalized to reduce the risk of redundancy to cause anomalies. Modification anomalies arise when data is added to, changed or deleted from a database table.

Normalization is the process of breaking down a table into smaller tables, so that each table deals with a single team.

As we are working with a document based database-information should not be redundant.

3.2.3 Safety Requirements

- Backups of the databases should be done daily and be kept for a week.

3.2.4 Software Quality Attributes

- Availability: The Tweets should be available to be seen at an instance when they have been created
- Correctness: The application should not allow anyone to manage tweets of other users other than ones of themselves
- Maintainability: The application should use continuous integration so that features and bug fixes can be deployed quickly without downtime
- Usability: The interface should be easy to learn without a tutorial and allow users to accomplish their goals without errors.

3.3 Extended Requirements

1. Searching for tweets
2. Trending Topics
3. Tagging other users
4. Tweet Notification
5. Tweet Recommendation system
6. Private Chat

3.4 External Interface Requirements

3.4.1 User Interfaces

- front-end software: Tailwindcss, Vue.js, Nuxt.js
- back-end software: Firebase, Redis

3.4.2 Hardware Interfaces

- Windows
- A browser which supports HTML and JavaScript ES6

Appendices

Appendix A

Appendix: The Product As Is

This chapter will run through the basic functionalities of the MVP.

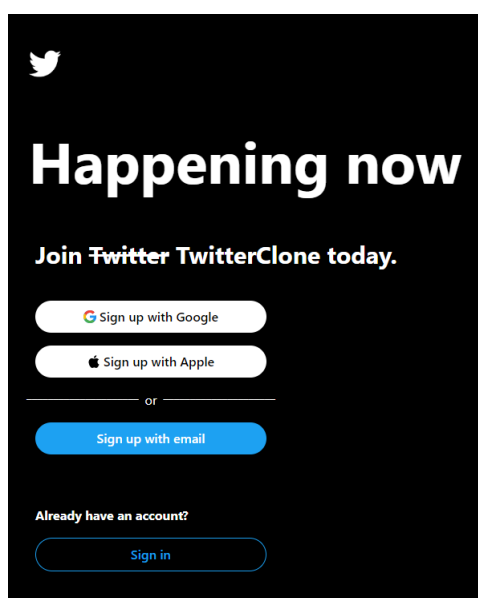


Figure A.1: Home

A.1 Login

Google and Apple Login is not implemented. You can use any generic email provided it is not already in use. If so you will be prompted to change the email.

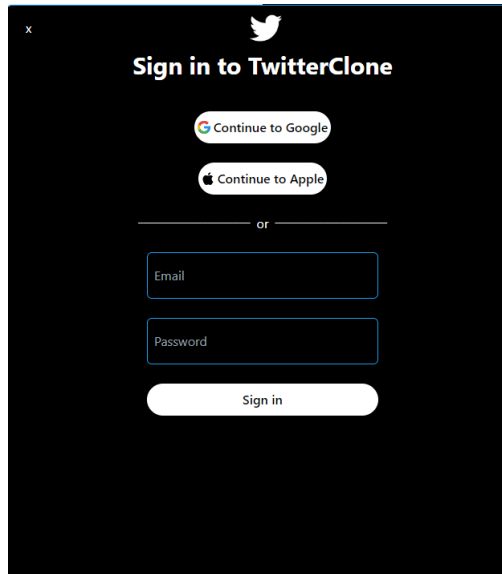
A login form titled "Sign in to TwitterClone" with a Twitter bird logo. It features two social login buttons: "Continue to Google" and "Continue to Apple". Below these, separated by an "or" divider, are input fields for "Email" and "Password". A "Sign in" button is at the bottom.

Figure A.2: Login Form

A.2 Register

A User account requires a name, unique email , unique tag and password. You will be prompted for incompatible Input

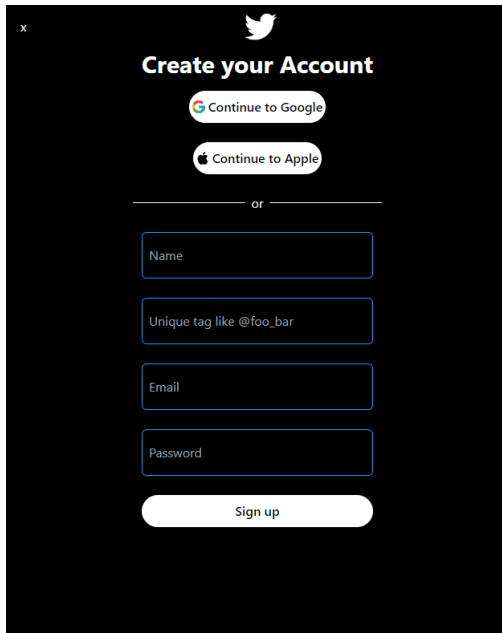
A registration form titled "Create your Account" with a Twitter bird logo. It features two social login buttons: "Continue to Google" and "Continue to Apple". Below these, separated by an "or" divider, are input fields for "Name", "Unique tag like @foo_bar", "Email", and "Password". A "Sign up" button is at the bottom.

Figure A.3: Register Form

A.3 Navigation

There are several different Pages available. The pages "Bookmarks" and "Lists" do not have any functionality behind them.

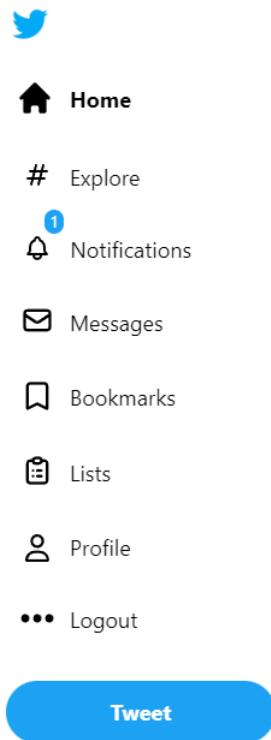


Figure A.4: Navigation

A.4 Tweet timeline

The Users Timeline consists of all the tweets the user has created or retweeted and all the tweet of the users the user is following.

As I imported a Dataset from Kaggle with hundreds of tweets about Crypto retrieving tweets in that volume results in a large amount of reading operations from the database.

I used a redis cloud instance to cache a users timeline daily so that the operations on Firebase's Firestore are limited significantly.

I store the Firebase documents as a Json string with a key that consists of the User's ID and date.



Figure A.5: Timeline

As you can see from the figure below the number of read operations were reduced significantly even though the app usage was pretty much the same.

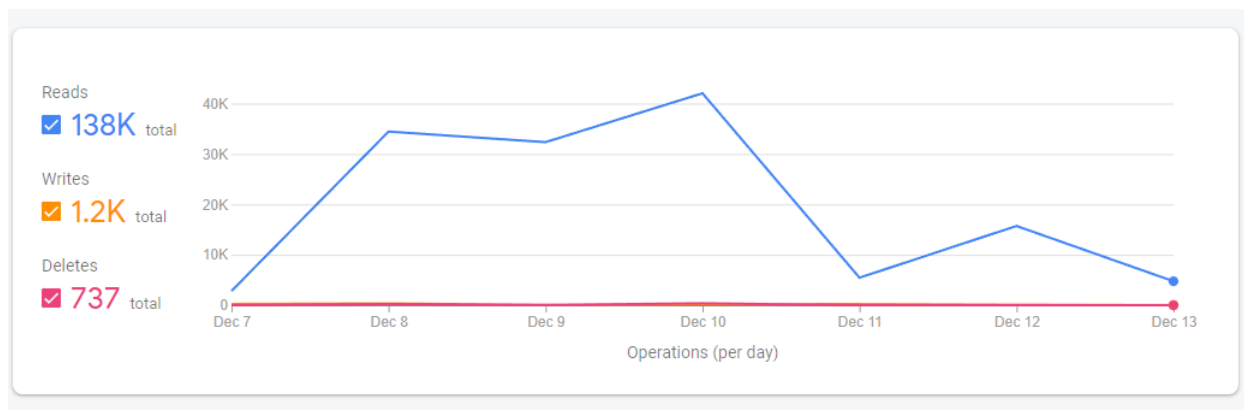


Figure A.6: Dec 11 is the day I started using a Redis cache

A.5 Trends and Search

You can search for any tweet or person using the Search input field on the Pages "Home" and "Explore". By clicking on any trending keyword in "Trends" the site is going to reroute the page to a search with the keyword as search keyword.

Trends are extracted by their relevance in all the tweets.

I use a basic method of NLP where i calculate the Term Frequency Inverse Document Frequency (TFIDF) To extract the most important hashtags and keywords.

Noteworthy algorithms i also implemented to compare those are the Rapid-Automatic-Keyword-Extraction Algorithm (RAKE) and Latend Dirichlet Allocation (LDA).

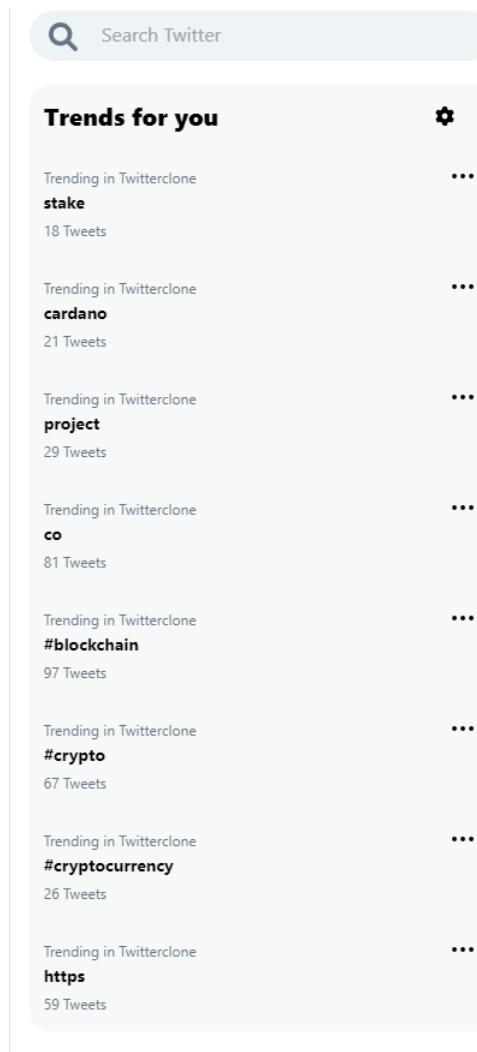


Figure A.7: Trends and Search

A.6 Create new Tweet

You can create new tweets with an optional image.
The upload of multiple images is under development.

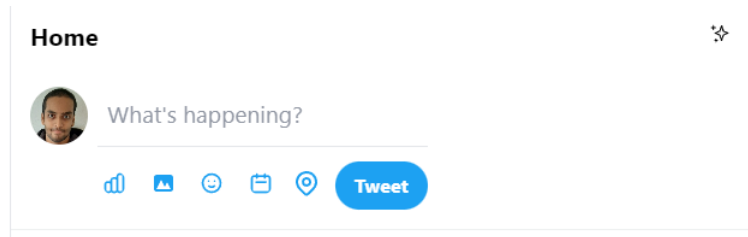


Figure A.8: Create

A.7 Tweet detailed View

The timelines consists of multiple tweets such as this one below:

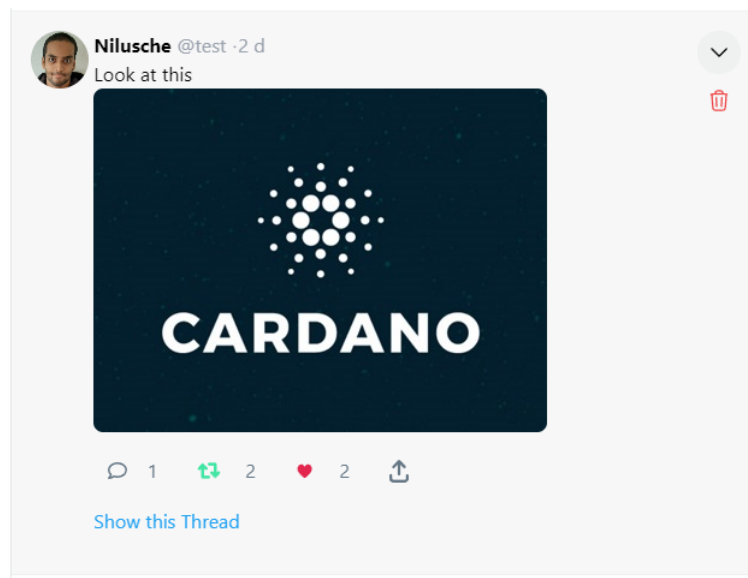


Figure A.9: Tweet

You can like, retweet and reply to a tweet.
Additionally if the tweet is yours than their is an additional button to delete the tweet

A.8 Follow Suggestions

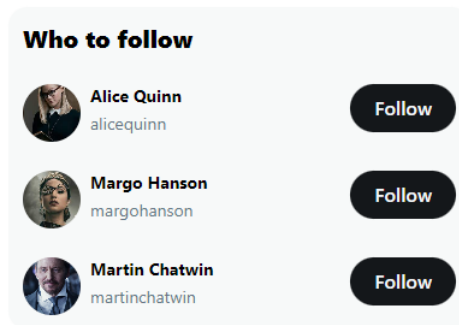


Figure A.10: Follow Suggestions

A.9 Tweet Recommendations

Like with the trends i use TFIDF to get recommendations based on the tweets that a user likes. I use Cosine Similarity to find those tweets that are the most similar to the tweets the user likes.



Figure A.11: Tweet Recommendations

A.10 Notifications

With Firebase's Snapshot Listeners I implemented Real time Notifications when a user likes, retweets or replies to a user's tweet.

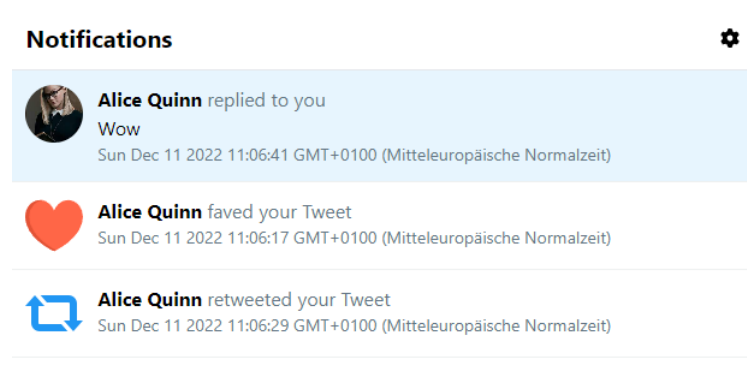


Figure A.12: Notifications Page

A.11 Profile Section

A user can change their bio and profilepicture in the Profile-Section.

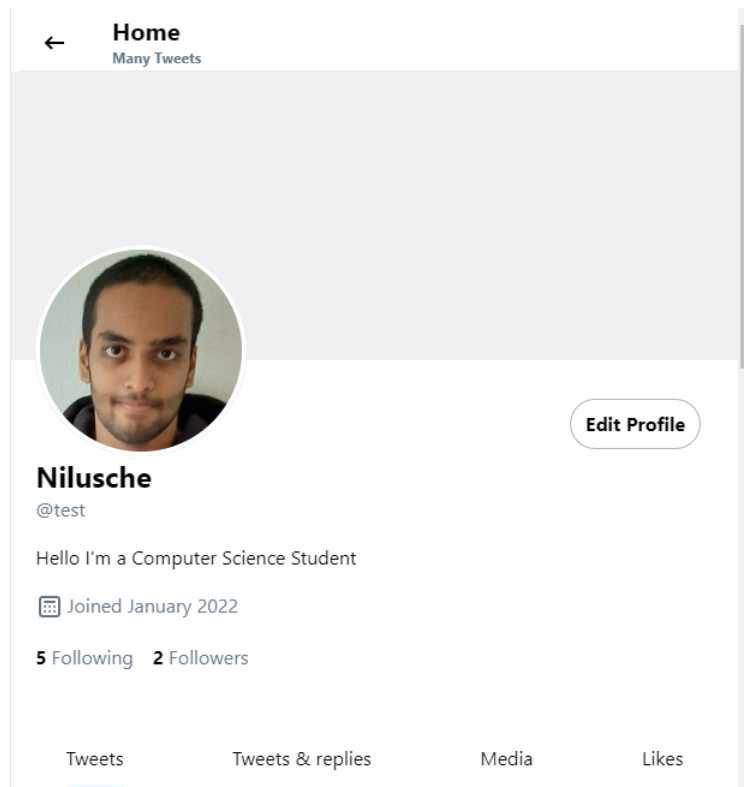


Figure A.13: Profile

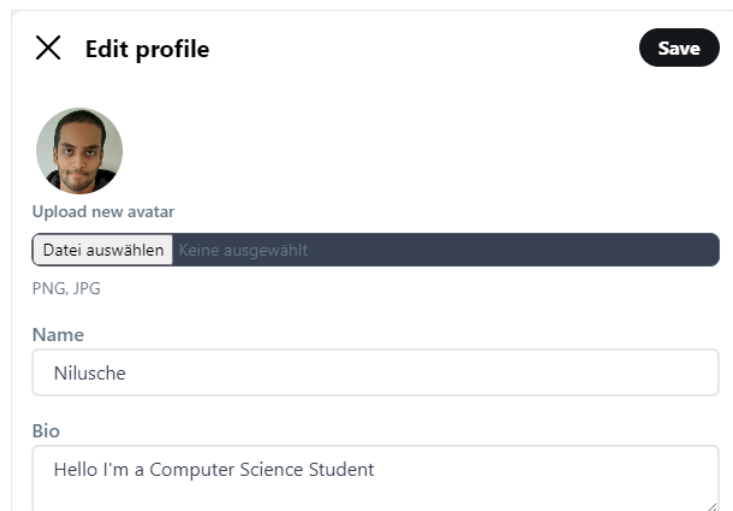


Figure A.14: Edit Profile

A.12 Messages

A user can send messages to other users in real time.
Once again I use Firebase's Snapshot Listeners for this real time Feature.

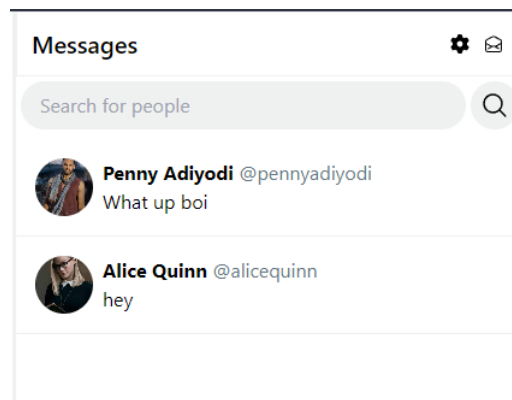


Figure A.15: Chats

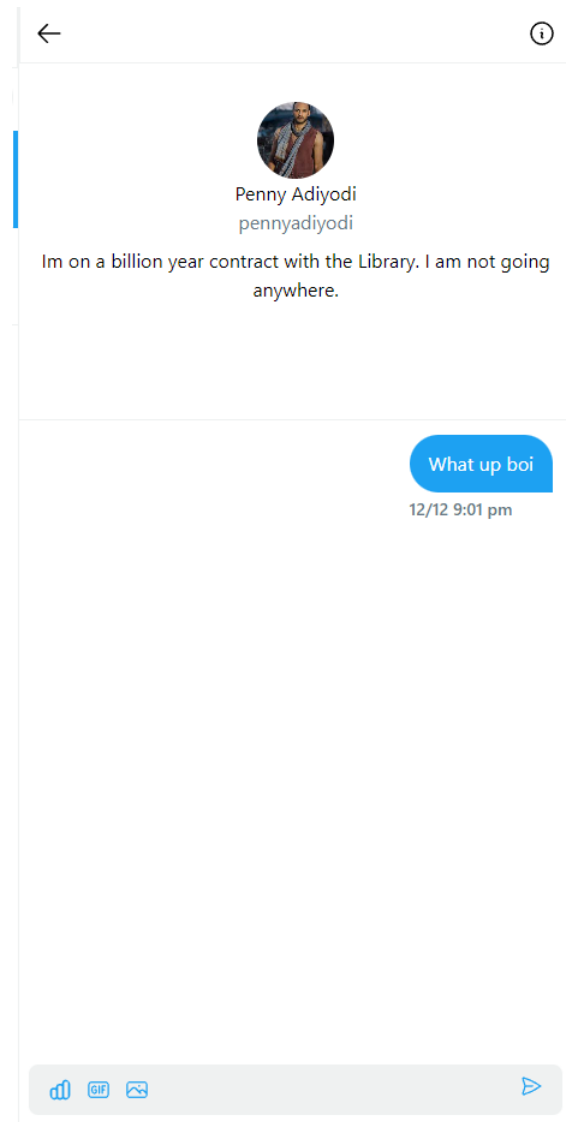


Figure A.16: Sending Messages