



Analysis document

Version 1.0

June 20th, 2021



Document history

Versions

Version	Date	Author	Changes	Status
0.1	03-03-2021	Kevin Heijboer	Document setup	Concept
0.2	07-03-2021	Kevin Heijboer	Functional requirements, non-functional requirements, use case diagram	Concept
1.0	06-20-2021	Kevin Heijboer	Minor changes	Concept



Table of Contents

Document history	2
Versions	2
1 Introduction	4
1.1 Purpose	4
1.2 Scope	4
1.3 Glossary	4
1.4 Overview	4
2 General capabilities	5
2.1 Quacking	5
2.2 Following	5
2.3 Profile	5
2.4 Personalized feed	5
2.5 Moderation	5
3 Requirements	6
3.1 Functional requirements	6
Use case diagram	6
3.2 Non-functional requirements	10



1 Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of Quacker. It will explain the purpose and features of the system, the different levels of permissions for the functionalities, what the system will do and the constraints under which it must operate. This document is intended for both the stakeholders and the developer of the system.

1.2 Scope

Quacker is a social media platform that can be used by anyone. The platform is designed to allow users to communicate with each other and share their thoughts in an instant. Users can communicate by posting small text messages called "quacks".

Users can also follow each other to stay up to date on each other's quacks. These quacks are then displayed in a personalized timeline on the homepage of the user. Users can customize their profile details such as a biography, location, etc.

Quacker is also moderated by moderators and administrators. These users have more power than normal users which gives them abilities such as banning misbehaving users.

Quacker is designed to be an enterprise-scale application. This means that it should be on enterprise level quality regarding non-functionals such as performance and security.

1.3 Glossary

Term	Definition
Quacker	Name of the application
Quacks	Quacker's version of tweets

1.4 Overview

The remainder of this document is divided into three chapters, General description, Specific requirements and Roles and permissions.

The *General description* section gives a clear and concise description of the application. It consists of a short view of the product and a short summary of the general capabilities of the product.

The *Requirements* section is subdivided in functional requirements, describing what the system should do, and non-functional requirements, describing the system quality attributes.

The *Roles and permissions* section gives an overview of the different roles and permissions a user has regarding the functionalities of the system



2 General capabilities

Quacker is similar to big social media platforms such as Twitter and Facebook. Its main features are posting quacks, following other users, a profile, a personalized feed, and a moderation platform.

2.1 Quacking

The quacking feature allows users to post their thoughts and activities in a 140-character quack. These quacks are stored in a database and can be found on the user's profile. Users can also @mention other users or #trends. Quacks can be searched for in the search bar.

2.2 Following

Users have the ability to follow each other. Following users will add newly posted quacks to the user's timeline. This way, users are always up to date on quacks of the users they follow.

2.3 Profile

Users can personalize their own profile by settings different types of information such as a profile picture, bio, location, etc. Other users can view these profiles publicly along with the profile owner's quacks. Profiles can also be searched for in the search bar.

2.4 Personalized feed

Users have their own personalized feed. This is a timeline of all quacks from users they follow and any quacks where the user is mentioned. The personalized feed will also contain any ongoing trends on Quacker.

2.5 Moderation

To keep Quacker civil, there is a platform for moderators and administrators to operate. This platform allows moderators and administrators to ban misbehaving users, promote and demote users and create new accounts.



3 Requirements

This chapter covers the functional requirements and non-functional requirements. Each requirement consists of a unique identifier, followed by description and a priority. The following priorities will be used, ordered from highest to lowest priority, as defined in

Must have: these requirements are essential for the product.

Should have: these requirements are not critical for the product to work, but are nearly as important as the Must haves, meaning they must be implemented if at all possible.

Could have: requirements which are not critical to the products success. If they can be implemented with little development costs, they can increase customer satisfaction.

3.1 Functional requirements

Use case diagram





ID	User story	Priority
Epic	Authentication / Authorization	
FR-1	As a user, I want to register an account so that I can login with an account	Must have
FR-2	As a user, I want to login to my account so that I can view my personalized feed	Must have
FR-3	As a user, I want to logout so that I can login as a different user	Must have
Epic	User profile	
FR-4	As a user, I want to set my username so that other users can recognize and tag me	Must have
FR-5	As a user, I want to set my profile picture so that other users can quickly recognize me	Must have
FR-6	As a user, I want to set my bio so that other users can get to know me	Must have
FR-7	As a user, I want to set my location so that other users know where I live	Should have
FR-8	As a user, I want to set a link to my website so that users have other ways to contact me	Should have
FR-9	As a user, I want to view my 10 most recent quacks so that I can remember what I quacked	Must have
FR-10	As a user, I want to like/heart a quack from a different user that I can show my appreciation of their quack	Could have
Epic	Search	
FR-11	As a user, I want to search for quacks so that I can find quacks that interest me	Should have
FR-12	As a user, I want to search for users so that I can follow them	Should have
Epic	Quacking	
FR-13	As a user, I want to quack so that other users know what I'm up to	Must have



FR-14	As a user, I want to view my latest quack on my start page so that I remember what I recently quacked	Must have
FR-15	As a user, I want to mention other users by adding @username in my quacks so that the other users know that I am referring to them	Should have

ID	User story	Priority
Epic	Timeline	
FR-16	As a user, I want to view quacks of users that I follow in a timeline so that I have a quick overview of their quacks	Must have
FR-17	As a user, I want to view my own quacks on the timeline so that I have a quick overview of my own quacks	Must have
FR-18	As a user, I want to view quacks that mention me so that I know who is quacking about me	Must have
Epic	Following	
FR-19	As a user, I want to view a list of users that I follow so that I can quickly see their profiles and latest quacks	Must have
FR-20	As a user, I want to view a list of my followers so that I know who follows me	Must have
FR-21	As a user, I want to follow other users so that I know what they're up to	Must have
FR-22	As a user, I want to unfollow other users so that I don't see their quacks anymore	Must have
Epic	Trends	
FR-23	As a user, I want to view quacks of current trends so that I am updated on the latest events and trends	Should have
FR-24	As a user, I want to tag a trend by adding #trend in my quack so that I can make a trend more popular	Must have
FR-25	As a user, I want to view all quacks regarding a trend when I click on the #trend tag so that I have a quick overview of all quacks regarding that trend	Should have



ID	User story	Priority
Epic	Moderation	
FR-26	As a moderator, I want to view a list of users and their roles so that I can perform actions on them	Should have
FR-27	As a moderator, I want to ban users from the platform so that they can't harm or insult other users	Should have
FR-28	As a moderator, I want to change the role of normal users to moderator so that more users can moderate the platform	Should have
Epic	Administration	
FR-29	As an administrator, I want to add new accounts so that I can help certain users with creating their account	Should have
FR-30	As an administrator, I want to ban users from the platform so that they can't harm or insult other users	Should have
FR-31	As an administrator, I want to change the role of other users so that I can moderate the moderators	Should have



3.2 Non-functional requirements

ID	Category	Description
NF-1	Availability	The system should be available 99% of the time
NF-2	Monitoring	The health, readiness and uptime of the system should be monitored and visualized
NF-3	Performance	The minimum response-time for an action must be 2 seconds
NF-4	Performance	The system should handle at least 1000 concurrent requests
NF-5	Scalability	The system must be divided in different microservices
NF-6	Scalability	The system must automatically scale up serves when needed
NF-7	Security	The system must minimize the risks of the OWASP top 10 (https://owasp.org/www-project-top-ten/).