

Table of contents

1. Introduction
 - a. Purpose
 - b. Client
 - c. Scope of Project
2. Overall Description
3. Functional Requirements
 - a. User Management
 - b. Content Creation and Management
 - c. Interaction and Engagement
 - d. Notifications
 - e. Messaging
 - f. Community Features
 - g. Content Moderation
 - h. Search and Discovery
4. NonFunctional Requirements
 - a. User Interfaces
 - b. Hardware Interfaces
5. External Interface Requirements
6. Risks and Assumptions

1. Introduction

a. Purpose

The main purpose of this document is to present a detailed description of the Horizon social media app. It will explain the features, interfaces, what the system will do, and the constraints in which the software must operate. This document is intended for the developers of the system and the client and serves as a guide for future enhancements and modifications.

b. Client

The client is Edwin Rodríguez, a professor from Cal Poly Pomona who envisions a new social media platform that allows for meaningful engagement.

c. Company

Horizon is a newly formed company that specializes in web applications that are scalable, secure, and user-friendly.

d. Scope of Project

This app will be a social media app for the client. The app will feature user accounts, content creation, voting system, communities, and messaging between users. Horizon hopes to combine the features of popular social media platforms such as Reddit and Instagram to create a fresh and new idea.

2. Overall Description

Horizon-social media is a social media platform where users can create accounts, join communities, interact through posts, comments, voting, and messaging. The app is designed for users to create content and engage with others. Communities will act as a backbone for the platform and allow users to do a variety of functions.

3. Functional Requirements

a. User Management

i. Account Creation

1. Personal Info

- a. First and last name
- b. Email Address: Valid email that's verified for account recovery and communication
- c. Phone Number: Additional verification

2. Password

- a. Strong and secure password that consists of uppercase, lowercase, numbers, and special characters

3. Security Questions & Answers

- a. Optional to help with account recovery

4. Verification

- a. Email Verification: Clicking on a link sent to email will confirm its validity
- b. Phone Verification: Entering a code sent to your phone will verify your phone

5. Terms of Service and Privacy Policy:

- a. Users need to read and accept terms and policies of the service

6. Captcha Verification:

- a. To check for bots, users will need to complete a CAPTCHA

ii. User Profiles

- 1. User profile is visible as a public

iii. Following System

- 1. Follow: Users can follow communities or people to see their post in their feed
- 2. Unfollow: Users can unfollow communities or people to stop seeing their posts in their feed

b. Content Creation and Management

i. Post Creation

- 1. User can post images or videos
- 2. Description: Can add text that will be shown right below the post
- 3. Sounds: given a list of sounds or can search for sounds that will play when a user is looking at the post

4. Tags: Add tags to post that will help it be found by other users
 5. Make private or public: User can choose if they want their post to be seen by followers or everyone
 6. Hide/Show Comments: User can choose if they want people to be able to comment on their post
 7. Hide/Show Votes: Users can choose if the amount of upvotes or downvotes can be seen
- ii. Content Editing and Deletion
 1. Edit: Users can edit anything they posted by changing any of the features listed in Post Creation (b. i.)
 2. Delete: Users can delete a post
 3. Delete posts on the community
 - iii. Filter Content
 1. User can find the post by keywords and hashtags
 - iv. Analytics and Reporting
 1. Performance Metrics: track engagement, reach, and growth
 2. Insight: Analyze data to understand community data, preferences, and trends
 3. Reporting: Provides reports on community performance
- c. Interaction and Engagement**
- i. Voting System
 1. On a post view, there will be an upvote button and a downvote button. A user can click on the buttons to like or dislike a post. A total score will be shown at the top of the post. Upvotes will be added as positive and downvotes will be subtracted from the total score.
 2. The post model should have an upvote and downvote attribute in the database.
 3. When the user upvotes, the upvote button should darken
 - ii. Commenting
 - iii. Discussions
- d. Notifications**
- i. Activity Notifications
 - ii. Customizable Alerts
- e. Messaging**
- i. Direct messaging
 - a. Friends list view that allows the user to select among friends to send a direct message. When a friend is selected, direct to a messaging view that allows the users to send messages to one another.
 - b. In the messaging view, have a name displayed on the top, then the messages and/or images sent on the center of the view with user messages on the right hand side and other user(s) on the left hand side

- c. Be able to also send a message to an unfriended user as well. When receiving messages from an unknown user, display a warning that the user does not know the messenger.
- ii. Group Chats
 - a. The friends list view will also have a button to send a new message that can choose one or more users to add to a chat.

f. Community Features

- i. Follow Communities
 - 1. Notify users of posts within community
 - 2. Display community under the users following list
- ii. Moderation tools
 - 1. Logs:
 - a. List of what members do within community
 - b. List of what moderators do within community
 - 2. Reports:
 - a. List of complaints members of the community create
 - b. Report are by default unresolved
 - c. Can mark reports as resolved
- iii. Community Management
 - 1. Promotion: Owner or any mods can promote other users to help manage a community
 - 2. Hierarchy: Owner can define and create the ability of roles which can be assign to members of community
 - 3. Role Permissions:
 - a. Read/View
 - i. View other content
 - b. Write/Edit/Delete
 - i. Can post content listed in b. i.
 - ii. Modify your own post
 - iii. Save changes to draft
 - iv. Can delete your own post
 - c. Comment
 - i. Add comments to other posts
 - d. Vote
 - i. Can upvote or downvote other posts
 - e. Moderator
 - i. Delete any post on the community
 - ii. Ban users in community
 - iii. View logs
 - f. Access Control:
 - i. Modify user roles and permissions
 - ii. Manage access levels for different users or role
 - iii. Make edits to community name, description, profile pic, and banner

- iv. Add restrictions and guidelines
- 4. Default Roles
 - a. Administrator
 - i. Default
 - ii. Moderator
 - iii. Access Control
 - b. Moderator
 - i. Default
 - ii. Moderator
 - c. Default
 - i. Read/View
 - ii. Write/Edit/Delete
 - iii. Comment
 - iv. Vote

g. Content Moderation

- i. Reporting
 - 1. When seeing content that is offensive, a user should be able to click
- ii. Blocking
 - 1. Accounts should also have a feature that allows them from seeing another user. The button can be found on the other persons profile
- iii. Content Removal
 - 1. When a user has been reported and the report proves justified, the system should be able to effectively

h. Search and Discovery

- i. Search Functionality
 - 1. Searching Accounts
 - a.
 - 2. Searching for Communities
 - a. a
- ii. Recommendations
 - 1. Recommended Friends
 - a. If there
 - 2. Recommended Communities
 - a. If

4. Nonfunctional Requirements

- a. Accessible Rich Internet Application (ARIA)
- b. Consistent user interface, fonts, colors, format across all views
- c. The app should have reasonable performance and loading times
- d. Security Requirements that allow for safe keeping of user details

5. External Interface Requirements

- a. User Interfaces
 - i. The app has a UI that works on desktop platforms

- b. Hardware Interfaces
 - i. The system should be deployable on cloud infrastructure like AWS

6. Risks and Assumptions

- a. Risk of security breaches should be mitigated by using proper security by providing correct User access and levels
