Software Requirements Specification

for

Facebook-*ish*

Version 1.0 approved

Prepared by Beau Schureck
Landon Gaillard
Heston Vaughan
Caleb Germany
Hunter Stanard

Group 1

September 6, 2023

1. Introduction

1.1 Purpose

This document aims to describe the requirements of Facebook-*ish* v1.0 as a social media service. This includes the user interface, web services, and data storage of the application.

1.2 Document Conventions

This document uses the Lato typeface with an 11pt font size.

This document has no special or specific typographical standards or conventions. All priorities are of equal importance.

1.3 Intended Audience and Reading

This document is intended for any individuals involved in this project and contains information about the high level requirements of the project as well as technical details as to how those requirements will be met. Developers, testers, and managers should find the whole document useful, especially sections 2.2-2.5, 3.1-3.x, and 4.1-4.4 for technical details. Marketing staff and documentation writers might find sections 1.1-1.5 and 2.1-2.2 helpful in describing the project as a whole.

1.4 Product Scope

Facebook-*ish* is a social media platform designed to allow users to network with each other via the internet. In expanding our company's reach in the social networking domain, having our own platform will allow us to monetize and promote user content ourselves.

1.5 References

N/A

2. Overall Description

2.1 Product Perspective

This product will be a stand-alone, self-contained social media platform that aims to provide features similar to Facebook, allowing users to connect and share content with friends online. The system structure will involve the front-end served to the user, managing user interactions, and interfacing with the database in the back end to store user data, posts, and other essential information.

2.2 Product Functions

Account Management Functions:
- Users can register an account.
- User functionality can only be accessed if a valid user is logged in.
- Users can edit their account information.

User Functions:
- Users can send and respond to friend requests.
- Users can only see their friend's posts in descending chronological order.
- Users can select a friend's timeline to view their posts.
- Users can like, comment, and share posts.
- Users can edit and remove posts.

Post Functions:
- Posts can receive and track likes from users, notifying the poster.
- Posts shared by friends of the poster will be displayed on reposter's timeline with a reference to the original
- Posts can be edited or removed by only the original poster

2.3 User Classes and Characteristics

Each user will have access to the same classification and overall functionality, although interactions are limited based on the interconnected classifications of the users.

Friend:
- If two users share the "Friend" classification, then their posts will show up on each user's timeline.
- These users can also view each other's profiles, like and share one another's posts, and leave comments.

Not Friend:
- If two users do not share the "Friend" classification, then their interaction is limited to view one another's profiles.

Note: If a user does not share the "Friend" classification with any other user, then nothing will show on their timeline. The functionality of this platform is dependent on connecting with other users.

2.4 Operating Environment

It will operate on any browser that supports html5 and Windows 10/11.

2.5 Design and Implementation Constraints

Technologies:

Python-Flask
- The basis of the web application
- Serves the front-end to the user.
- Manages user registration and login.
- Controls database management.
- Handles user interactions.

Flask-SQLAlchemy
- Object Relational Mapper to interact with the database through python code
- Abstracts operations to python objects rather than raw SQL.

SQLite3
- Database choice

JavaScript, HTML, CSS
- Enables ease of user interactions
- Tracks user clicks and serves appropriate data from the server

Hosting: The platform will be hosted on a home computer, limiting the user capacity

3. System Features

3.1 Logging In

3.1.1 Description and Priority

This is a high priority feature that involves allowing the system to identify what user is using the system by using a username and password.

3.1.2 Stimulus/Response Sequences

The user will be able to authenticate themselves by typing in a username and a password into two separate text boxes. The user will press enter and the system will authenticate whether or not that user exists.

3.1.3 Functional Requirements

REQ-1: There will be two text boxes. One labeled username and one password.
REQ-2: If a username password combination entered is incorrect then give an error and reset the login screen
REQ-3: If a username password combination entered is correct then navigate the user to the homepage

3.1.3 Output

When a username/password combination entered is correct, the user will be navigated to the homepage. Otherwise, the user will be shown an appropriate error message.

3.2 Posting and Editing Posts

3.2.1 Description and Priority

User can write a text-based post for their friends to see.
Priority: HIGH

3.2.2 Stimulus/Response Sequences

1. User clicks on “Post” button
2. System opens up a text box for the user to input text
3. User writes what they want, then clicks the “Post” button
4. System saves the content of the post for displaying when needed
5. If User is trying to edit, then user clicks on “Edit” button on a post
6. Sytems opens a text box filled with the content of the post
7. User clicks the “Finish Editing” button when finished

3.2.3 Functional Requirements

Post Button Display: System should have a “Post” button displayed clearly
Post Button Interaction: System should enable user to click on the button
User Input: System should allow User to input text into a text box
Finalize Post Button: System should display a “Post” button to submit post
Content Saving: System should save post to database
Post Display: System should display post to interface immediately
Edit Post Button Display: System should display an “Edit” button on posts
Edit Post Button Interaction: System should allow User to click on the button
Post Text Box: System should open text box filled with post content
Post Edit: System should allow User to edit content
Finish Edit Button: System should display a “Finish Edit” button
Finish Edit Interaction: System should allow user to click on the button
Post Update: On button click, the system should update the post in the database
Updated Display: The system should display the updated post

3.2.4 Output

New/Updated post displays on the screen

3.3 Friend Requests

3.1.1 Description and Priority

This is a high priority feature that involves allowing a user to be able to send and accept friend requests from others which will allow them to be friends with others and enable them to interact with them.

3.1.2 Stimulus/Response Sequences

The user should be able to search for other users and send friend requests to them from the search. Users should also have a list of incoming friend requests on their home page.

3.1.3 Functional Requirements

REQ-1: There will be a search bar that will allow the user to search for other users by username.
REQ-2: There will be a button by each user that says “Send Friend Request”. If the button is pressed, a friend request is sent to that user.
REQ-3: There will be a tab on the home page that has incoming friend request. Each friend request will have a check and x button next to the user.
REQ-4: Pushing the check button will make the user a friend. Pressing the x button will ignore the request.

3.1.3 Output

If the receiving user accepts friend request, a “friend” relationship is established between the users.

3.4 Post Interaction

3.4.1 Description and Priority

User can like, comment, and share their friend’s posts

3.4.2 Stimulus/Response Sequences

1. User presses “like” button on post
2. System saves liked status for post
3. User presses comment button
4. User comments in text box
5. System saves content of comment and displays new comment
6. User presses share button
7. A link to the post is copied to user’s clipboard

3.4.3 Functional Requirements

Post Attributes: System should have post attributes (# of likes, shares, comments)
Interation Buttons: System should have buttons for the corresponding attributes
Attribute Update: System should update “attribute count” when button is pressed
Comment Box: For comments, system should open text box for writing
Comment Display: When finished commenting, content should be saved and displayed on post
Sharing Link: For sharing, system should copy link to user’s clipboard on button press and update share count

3.4.4 Output

Like, share, and comment displays are updated on the post

4. Other Nonfunctional Requirements

4.1 Performance Requirements

Friend requests and posts should be in the database as soon as possible when sent from the user. It should seem instantaneous to the user.

4.2 Safety Requirements

Developers must be aware that users can post illegal posts to the platform. Although a moderation system was not requested by the customer, the developer must still be aware of the potential for these illegal posts.

4.3 Security Requirements

The platform must keep passwords and all other customer data secure from everyone else. This is private data and cannot be sold or given away.

4.4 Software Quality Attributes

The software must adaptable as we know that this only the first part of our project. We know that we will be given additional requirements for our software. Therefore, we must make sure that our software is adaptable to deal with the additional requirements.

5. Other Requirements

N/A

Appendix A: To Be Determined List

References

Appendix C: Glossary

CSS – Cascading Style Sheets

HTML – Hypertext Markup Language