7/5/24

<BULLETIN> SYSTEM REQUIREMENTS DOCUMENT

DRAFT

TEAM “ALPHA”
SHAWN MURRAY, KATRICIA PENUS, CHRISTIAN REYES, JOEY BARBATO

Table of Contents

Introduction 1

Description Model 1

Class Diagram 1

Use Case Diagram 1

Use Case Scenarios 1

System Sequence Charts 1

INTRODUCTION (KP)

Describe the purpose of this requirements document and outline what it contains.

A system requirements document (SRD) provides a detailed overview of a software product's features, functionality and performance. Furthermore, it clarifies the project scope, is a guidance for development, can be used as a basis for testing, etc. Overall, a well-defined system's requirements is crucial for ensuring that the developed system meets the team's and stakeholder's expectations, functions correctly and aligns with the overall project goals. In this document, it further looks into the development for the app, "Bulletin" and provides detailed documentation and models. Included is the description model, class diagram, use case diagram, use case scenarios and the system sequence charts.

DESCRIPTION MODEL

Using text, describe the requirements for your system. Expand on the function section from your project plan. Include requirements for the following categories: Output, Input, Processes, Performance and Security.

Output Requirements:

- *User Interface (UI):*
    - *Profile Page: Display the user's information (based on privacy settings), profile picture, settings (public/private, online/offline status)*
    - *Community Browser: Present communities categorized by larger tags and specific tags based on the user's interest*
    - *Posts: Show posts with upvote/downvote counts, comments, and an option to bookmark*
    - *Notifications: Alert users about new posts, comments, or interactions*
- *Moderation Tools:*
    - *Admin Dashboard: Provide tools for site-wide admins to monitor and moderate communities and user activities*
    - *User Management: Display user reports and allow actions like warnings, suspensions, or bans*

Input Requirements:

- *User Input:*
    - *Account Creation: Accept user's information (email, password, address, etc.) for registration*
    - *Profile Settings: Allow users to input or update profile information and preferences*
    - *Post Creation: Input title, content, and tags*

    - *Comments: Input comments on posts with options for formatting and multimedia attachments*
- *Admin Input:*
    - *Moderation: Input from admins to manage user-reported content and enforce community guidelines*

Process Requirements:

- *Authentication and Authorization:*
    - *User Authentication: Verify user identity through email verification, phone verification, or CAPTCHA*

      Authorization Levels: Define roles (user, moderator, admin) with varying access rights

- *Community Management:*
    - *Community Creation: Enable users to create new communities with define settings and moderation rules*
    - *Moderation Workflow: Implement a workflow for user-reported content, including review, decision-making, and action*
- *Content Interaction:*
    - *Post Management: Facilitate creation, editing, deletion, and sorting of posts based on user interactions (upvotes, bookmarks, etc.)*
    - *Comment Management: Allow users to add, edit, or delete comments with moderation features (reporting, hiding)*

Performance Requirements:

- *Scalability:*
    - *User Growth: Handle a scalable number of users and communities without degradation in performance*
    - *Traffic Management: Manage peak loads and concurrent user interactions effectively.*
- *Response Time:*
    - *Page Loading: Ensure fast loading times for UI components like profiles, community listings and posts*
    - *Real-time Updates: Provide timely notifications and updates for user interactions (comments, votes)*

Security Requirements:

- *Data Protection:*
    - *User Data: Encrypt sensitive user information (passwords, personal information) during transmission and storage*
    - *Access Control: Implement role-based access control to ensure only authorized users have access to specific functionalities and information*
- *Content Security:*
    - *Moderation: Prevent and manage inappropriate content through automated filters and human moderation processes*
    - *Data Integrity: Ensure the integrity of user-generated content and prevent unauthorized modifications*

- *Compliance:*
    - *Privacy Regulations: Adhere to data protection laws regarding user consent, data storage, and processing*
    - *Community Guidelines: Enforce platform-specific rules and guidelines to maintain a safe and respectful environment for all users*

Class Diagram

Create a class diagram. The Class Diagram should contain all of the system objects, their attributes, and any known methods. This diagram may be included as a separate file – it does not need to be inserted into this Word document. ***will be in github***

Use Case Diagram

Create a Use Case Diagram for all of the "uses" of your system. This diagram may be included as a separate file – it does not need to be inserted into this Word document. ***will be in github***

Use Case Scenarios

Create a full description Use Case Scenario (detailed descriptions) for each use case of the system. This full scenario should include an enumerated list of steps involved in the activity as well as any exception conditions.

createPost

- Stakeholder: User who creates the post, Bulletin
- PreConditions: None
- PostConditions: None Required
- Flow of Activities:
    - Create Post Draft ⟶ Edit Post Draft ⟶ Save or Post Draft
- Exception Conditions
    - None

deletePost

- Stakeholder: User who creates the post, Bulletin
- PreConditions: createPost
- PostConditions: None
- Flow of Activities
    - Delete Post
- Exception Conditions
    - None

editPost

- Stakeholder: User who created the post, Bulletin
- PreConditions: createPost
- PostConditions: None
- Flow of Activities
    - Edit Post
- Exception Conditions
    - None

createCommunity

- Stakeholder: User who creates the community, Bulletin
- PreConditions: None
- PostConditions: None
- Flow of Activities
    - None
- Exception Conditions
    - None

deleteComunity

- Stakeholder: User who deleted the community, Users within the community, Bulletin
- PreConditions: createCommunity
- PostConditions: None
- Flow of Activities
    - Delete Community
- Exception Conditions
    - None

System Sequence Charts

For each Use Case Scenario, provide a sequence diagram. Use your class diagram, use case diagram and scenarios to create the corresponding System Sequence Diagram. ***will be in github***