**SOUVENIR**

Design Document and Requirements Guidelines

By Reacter

December,2022

Xiao Zhang 2227300

Lei Zhou 2195778

Xiao Pei 1996017

PART A

**Introduction with Motivation**

Souvenir – a social media web app.

Do you have any memories that you want to record? Do you have any important moments to be remembered? Do you have any news or opinion that you want to share with friends? *Souvenir* can help you keep down all of them.

Souvenir is a social media web application, it’s similar to blog, but adding more user interaction features, and we focus on short posts, not long articles.

With *Souvenir* we can store our memories, share information and make connections. On a personal level, it allows us to communicate with our friends and families, learn new things, develop our interests, or just be entertained. On a professional level, we can broaden our knowledge in a specific field and build our professional network by connecting with other professionals in this field.

**Functional Requirements**

In our *Souvenir* web application all the posts are open to public, so everyone can browse through the posts but can’t manipulate them. We have two layers of login: administrator and user.

Visitors can browse all the posts or apply for membership. To successfully registered, an unique email address and unique username are required, because we can use both email and username to login. If the email or username is already used, user will get a reminder to choose another email address or username. The register date will be generated automatically. The photo is not optional, is photo is not provided we will give user a default photo.

After login Users can create a post. We focus on short posts, not long articles, the maximum length for each post is 280 characters. Every post should have a title. Users can also edit his own post except change its statue (active or banned). After login users can comment other user’s post. The maximum length of each comment is 140 characters.

Administrators account are set and allocated by App provider. Their information is pre-defined in the database. Visitor can not register as an administrator. Administrator can edit user’s information except the register date, because it’s generated automatically. They can also delete a post or change its statue. They don’t have authority to edit post title and content.

**Technical and soft requirements**

Technical Requirements

* Must be a transactional web application.
* Must have React front-end .
* Must include Form text input.
* Must perform validation of input .
* Must handle errors properly (fail gracefully) .
* The application should not crash.
* The user should be able to perform all interactions with the application by visiting a home page and then navigating appropriately within the website.
* The website must have an "About Us" that describes your team and the motivation for the website.
* Must get, display and manipulate data .
* 3rd party API or CRUD backend for core domain data .
* CRUD for user database
* Must use props & state appropriately.
* User must interact with data and site must respond to user actions appropriately.
* E.g., Add items to playlist or cart, sort items, filter items, tag items.
* Must have a Dark mode/Light mode.

Soft Requirements

* Must be appealing (i.e., a good aesthetic)
* Decent color scheme, intentional UI design philosophy, images
* Must be usable
* Well laid out, obvious operations, user-friendly interface
* Clear input directions, Understandable navigation
* Must be focused
* No extraneous, distracting content

Development tools

A picture containing logo

Description automatically generated

* MongoDB
* Express JS
* React JS
* Node JS
* IDE: Visual Studio Code
* System: Windows
* Frontend Framework: Bootstrap, Material-UI

PART B

**Interface Design**

**Datebase Design**

**Application Features**

**Coding Techniques/technologies**

PART C

**Timeline**

1. The timeline indicates the key milestones of the project.  Some key milestones to include in your timeline are:
2. December 2nd-submit design and requirement document
3. December 7th-Presentation on project status with preliminary demo.
4. December 9th-Final demo and project report
5. Appropriate milestones for testing and iteration
6. Note: Give yourself appropriate time to test your final submission after ALL final code changes.
7. The timeline may be given visually or in a list or table format.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Time line** | | | | |
| No. | Item | description | Deadline | Member |
| 1 | requirement document | Submit  design and requirement document | 2 Dec | Pei,Xiao |
| 2 | Presentation on project status | Presentation on project status with preliminary demo | 7 Dec | Zhang,Xiao |
| 3 | project report | Final demo and project report | 9 Dec | Zhou,Lei  Pei,Xiao  Zhang,Xiao  Wu,Wei |

**Tasking plan**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| * + 1. **Tasking Plan** | | | | | | |
| * + 1. No. | * + 1. Task | * + 1. Description | * + 1. details | * + 1. Member | * + 1. Deadline | * + 1. Priority |
| * + 1. 1 | registrations | * + 1. Have registration function |  | * + 1. Pei,Xiao | * + 1. 4 Dec | * + 1. must-haves |
| * + 1. 2 | Inventory management | * + 1. Have function of Managing inventory |  | * + 1. Zhang,Xiao | * + 1. 4 Dec | * + 1. Nice-to-Haves |
| * + 1. 3 | React front-end |  |  | * + 1. Zhou,Lei | * + 1. 4 Dec | * + 1. must-haves |
| * + 1. 4 | Form text input |  |  | * + 1. Wu,Wei | * + 1. 4 Dec | * + 1. must-haves |
| * + 1. 5 | validation of input | * + 1. Validate input |  | * + 1. Wu,Wei | * + 1. 4 Dec | * + 1. must-haves |
| * + 1. 6 | handle errors | * + 1. Error handling |  | * + 1. Wu,Wei | * + 1. 5 Dec | * + 1. must-haves |
| * + 1. 7 | CRUD for user database | * + 1. Have the function of get,create,update,delete user function |  | * + 1. Wu,Wei | * + 1. 4 Dec | * + 1. must-haves |
| * + 1. 8 | Add items to playlist | * + 1. Show playlist items |  | * + 1. Zhang,Xiao | * + 1. 4 Dec | * + 1. must-haves |
| * + 1. 9 | Dark mode/Light mode | * + 1. Have function of dark and light mode |  | * + 1. Zhou,Lei | * + 1. 4 Dec | * + 1. must-haves |
| * + 1. 10 | intentional UI design | * + 1. Optimize UI |  | * + 1. Pei,Xiao | * + 1. 4 Dec | * + 1. must-haves |
| * + 1. 11 | layout | * + 1. Optimize layout |  | * + 1. Pei,Xiao | * + 1. 4 Dec | * + 1. must-haves |

**Coordination plan**