Individual Project 02: Requirement and Architecture

A Web Service to Manage User’s Password

Hong Van

CS 504 – Software Engineering

Instructor: Shane Frigon

City University of Seattle

August 1, 2021

**Project Requirement:**

* Each student will conduct a requirement engineering process and map the resulting requirements into a systems architecture or high-level design. The student will trace the mapping of requirements to modules in the architecture.
* The student will choose an architectural style based on the appropriateness of the style to the requirements. Based on the style the student will complete the architecture/high level design to the level where it is ready for implementation.
* The student will create all design artifacts appropriate to the selected style.

**Introduction**

In a modern technological world, almost everyone has various high-tech devices that are used to connect and share information with each other. Passwords also become an essential part of this online world. People must create user accounts and passwords to be able to log in to websites/any devices they want to. Hence, this will lead to a consequence that users will have so many passwords and sometimes they do not know how to manage them effectively. Some people manage their passwords by using simple, easy-to-remember passwords, or same password for many different websites and this will put them in a high risk of being stolen. Hence, the better password management, the more protected their accounts will be from hackers and malicious software. In this project, the author is going to develop a full stack web service to allow users to generate, store, update, and delete their usernames and passwords.

**Requirement Engineering:**

Requirement engineering is a process of gathering and defining the needs and desires from stakeholders. It helps to identify the problem clearly and completely as well as to ensure that the solution is correct, reasonable, and effective. For this project, the author is going to analyze in detailed the requirement engineering process for the web service that allow users to manage their accounts and passwords effectively. The requirement engineering for this project will be divided into 2 parts which are user requirements and system requirements. However, in this paper, the author will just explain in detail about the system requirements, which include the description of the web service and its operation such as how the web service will be used and how to develop the core features showed on the web service.

**Architecture Style:**

The architecture style is the description of relationship types and elements along with a set of constraints to implementing a software system. There are many choices of architectural styles that help to design the solution quicker. In this project, the author will use REST API Style as it is designed to make optimal use of HTTP and will defines several constraints for API design. Moreover, the REST style will ensure that APIs use HTTP correctly. Most importantly, this style performs create, read, update, and delete operations “which can be easily mapped to HTTP methods: creation can be performed by a POST, reading is performed by GET, updating is performed by PUT and a deletion is performed by a DELETE” (API University, 2021) – see the figure 1 below.

Diagram

Description automatically generated

*Figure 1: REST architecture style constraints. Source: Yadav (2020).*

**Design Artifacts based on the Requirements and Architecture Style Selected**

The main core feature of this web service is the password manager feature. This feature will allow a user to create/add, store, update, and delete their accounts and passwords into the system database. To do this, the users need to list all the name of the applications, usernames, and password to input this information into the system – see figure 2: Example of application, username, and passwords.

|  |  |  |
| --- | --- | --- |
| **Name of Application** | **Username** | **Password** |
| Google | jane.doe@mail.com | 123549 |
| Facebook | jane.doe@mail.com | efgb1559 |
| Salesforce | jane.doe@mail.com | 58648dfd |
| Microsoft | jane.doe@mail.com | Hdinv569 |
| Netflix | jane.doe@mail.com | 3554968 |
| Twiter | jane.doe@mail.com | 5597463 |
| Amazon | jane.doe@mail.com | vvfdvsdf5 |
| Spotify | jane.doe@mail.com | vd8894 |
| Paypal | jane.doe@mail.com | hinslin897 |

*Figure 2: Example of application, username, and password*

The Figure 3 below shows a sample screenshot for a user interface (UI) for the list of available application they have added to the web service. The UI for my project will look similar like this. When the user clicks on each application, its password will show.

**Graphical user interface, application

Description automatically generated**

*Figure 3: User Interface Example. Source: https://www.lastpass.com*

*Backend and Frontend Services:*

* The frontend is client-side, and it is the part the user interacts with. HTML, CSS, and Javascript will be considered as the frontend services for this project.
* The backend is often used for data storage or communication. This service is to ensure that user’s information is all in one convenient location. In the other ways, it is sort of a repository of everything that makes the web presence smoothly. For this project, the server-side tech stacks may involve NodeJS and database.

*Database:*

The author will use SQL database for reporting and it will perform the following tasks:

* User profile
* Track user registration activity and status
* Analyze information about what actions are performed by users
* View a list of users whose Questions and Answers profiles must be updated to comply with the current administrator-defined settings
* Track Support Center operators’ activity
* Other potentially

**Reference**

Yadav, A. ( 2020, August 19). A Look at REST API Design Pattern.

https://dzone.com/articles/a-look-at-rest-api-design-patterns

Lastpass. (2021). How it works.

https://www.lastpass.com/how-lastpass-works

Elgabry, O. (2016, September 13). *Requirements Engineering – Elicitation & Analysis (Part 2)*.

https://medium.com/omarelgabrys-blog/requirements-engineering-elicitation analysis-part-2-a02db801f135

API University. (2021). REST

https://api-university.com/api-lifecycle/api-design/rest/