Project Postmortem

Project objective

Multi-function web application, allowing valid user to manipulate data in the web environment.

Achieved objective.

Prototype was design with simple UI, allow user to learn and master it quickly.

UI composed of multiple icon with associated description, web application will directly user the appropriate page base on the icon.

User can able to view data and manipulate with some data (temporarily)

Failure.

The web application is unable to deserialize data because of the lack of skill of the developer.

Web application is missing web-service and not able to handle deserializing.

Schedule.

Project is delivered on schedule with missing functionality.

Performance.

The application is working as excepted. Except, it wont be table to write new data.

What went well in the project.

I don’t have much experience with front-end language. In this project, I was accumulated many knowledges and mostly through self-research while working on the project. One thing that catch my attention is that front-end language does live up to it name, it introduces flexible and convenient way to work with user interface compare to back-end language like java. For instance, animation is much easier and simpler to deploy in font-end. Thus, I really enjoy the designing part and most things in designing process went very smoothly.

What did not go well

Part of this is because I don’t have much experience with front-end. At the beginning stage of the project, I thought manipulating data in Web app environment can be achieve in the similar manner like IO in back-end language, but it actually not a case in front-end where serializing data from file can be done easily and it is similar with back-end. However, deserializing data is more complicate and it seems to require web app to interact with web server and this is where another language (framework) come up such as php.

Most designing stuffs went very smoothly until I reached the data part. I intended to have all the data store in json format then read and write it locally. I managed to have the web application successfully read data from Json but I was not able implementing the writing process without the web-server, thus my project is missing this features and I can only demonstrate in the hypothesis scenario. Detail will be in the second part.

Lesson learn and can applied to future project.

In this project, I was able to refresh my limited knowledge with front-end language such as html, CSS, and a little bit of JavaScript. After this project, I would like to experiment an application where front-end and back-end are combined; front-end will be used to handle user interface and back-end will be used to handle functionality and logic. Learning webserver and other frameworks is also become apart of my to do list.

Web app brief explanation

Data: (Json)

Cust.json (data such as existing services)

Donor.json (donation data)

User.json (login data including user name, password and user type)

Userdetail.json (including detail information of each user)

The web application will start with the login screen, user must enter valid username and password to process further. Login will be using data from user.json to identify user. If user enter the valid user name and password then user name will be added to the ending of the href link so the next page can be able to identify the user.

Added to the ending of the href link is demonstrated as follow.

[http://localhost:8080/HTML\_ser416ComPro/emp.html?**data=admin**](http://localhost:8080/HTML_ser416ComPro/emp.html?data=admin)

If user is valid the next page that pop up will have user first name display next the word welcome.

Login Page

A screenshot of a computer

Description automatically generated

Name displays next to the word welcome

A close up of a logo

Description automatically generated

Login display a notification if user input wrong username and password

A close up of a logo

Description automatically generated

Following are brief explanation of each user

Admin – including paid-employee

Admin will have these following options appear on the web page

A close up of a logo

Description automatically generated

* Check available schedule
  + Available schedule indicates the existing services that were created by customer and it have not been taken by any volunteers.

A screenshot of a computer

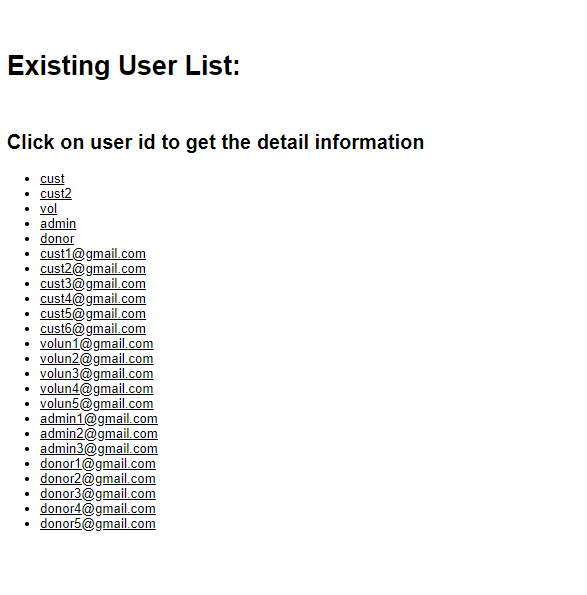
Description automatically generated

* Check occupied schedule
  + Indicates service that were already take by volunteers

A screenshot of a social media post

Description automatically generated

* View user list
  + View and manage all existing user in the database
    - View working
    - Manage is not working due the missing of web server
    - Manage will be demonstrate in the hypothesis scenario



If click on one of the items on the list, the appropriate dialog will pop and show detail information of the chosen user.

Image show info of [cust1@gmail.com](mailto:cust1@gmail.com)

A screenshot of a cell phone

Description automatically generated

If webserver is implemented, admin can able to update another user info by change value in text box then select update button.

Admin also has a right to remove any users, by selecting remove user.

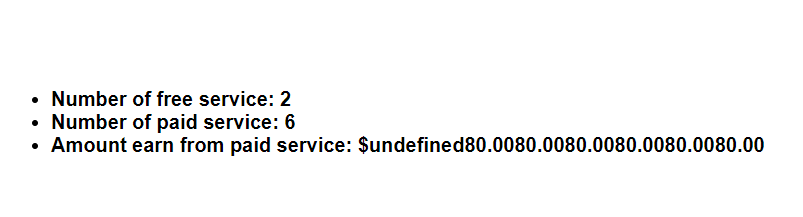
* View and Edit Personal Infor
  + View and manage the personal data of this user
    - Such as name, address, phone number
    - Manage is not working.

Admin can change their info by change value in text box then select submit button

A screenshot of a cell phone

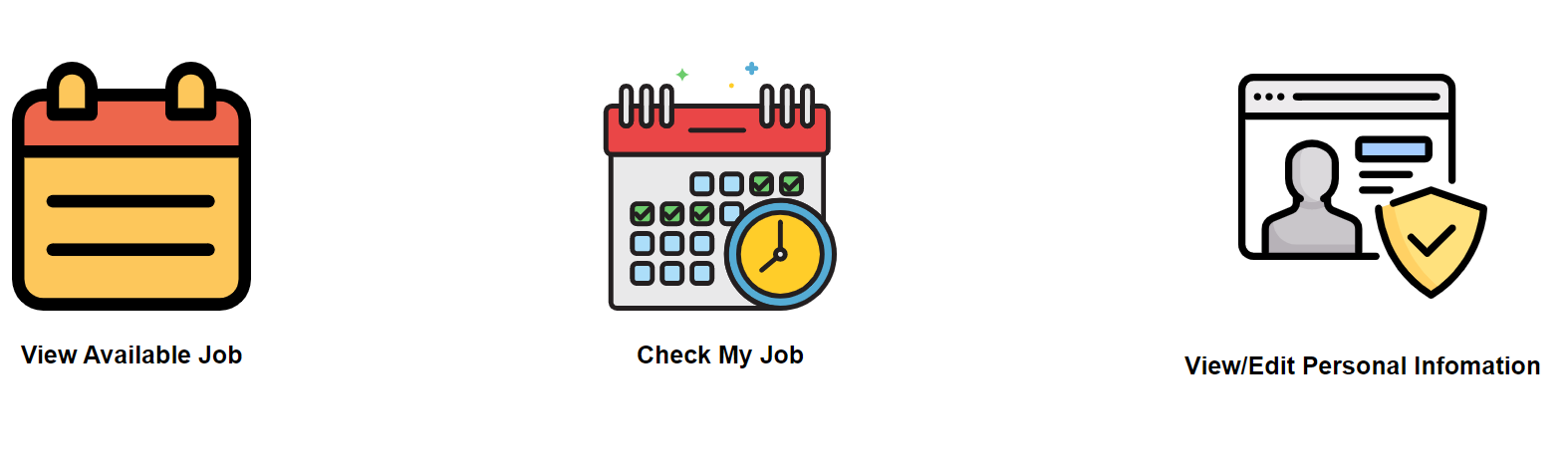
Description automatically generated

* View Report
  + Following item will be generated from json if this option is initiated
    - Number of existing free service
    - Number of existing paid service
    - The amount earned from paid service
    - Notice, this can be further expanded.



Volunteer user

Volunteer has three available option

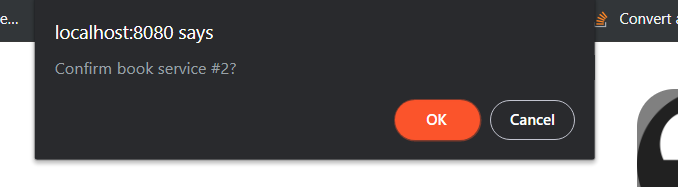


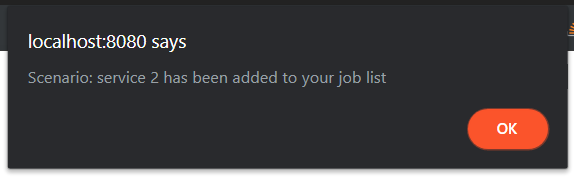
* View available job.
  + User can view all available job
  + User can apply for job in this option

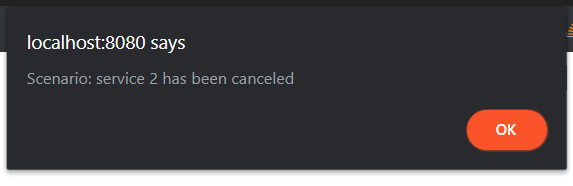
A screenshot of a cell phone

Description automatically generated

To apply for the job, user select on item in ID column (red color), then an appropriate confirmation pops up and ask for user confirmation.







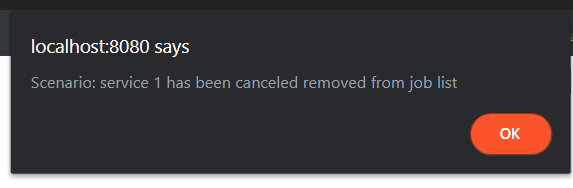
* Check my job.
  + User can view all job that is applied by this user
  + User can also cancel the job

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated



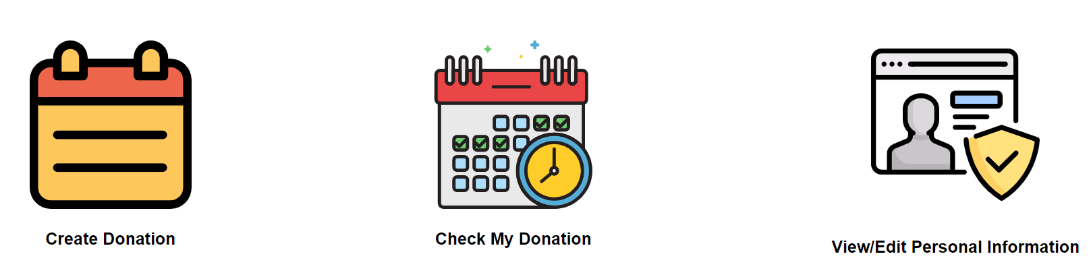
* View and Edit Personal Infor
  + View and manage the personal data of this user
    - Such as name, address, phone number
    - Manage is not working.

A screenshot of a cell phone

Description automatically generated

Donor user

This user also has three option



* Create Donation
  + Allow user to specify amount of the donation then submit it to the server

A screenshot of a cell phone

Description automatically generated

* Check Donation
  + Allow user to all donation have been donated by this user

A screenshot of a cell phone

Description automatically generated

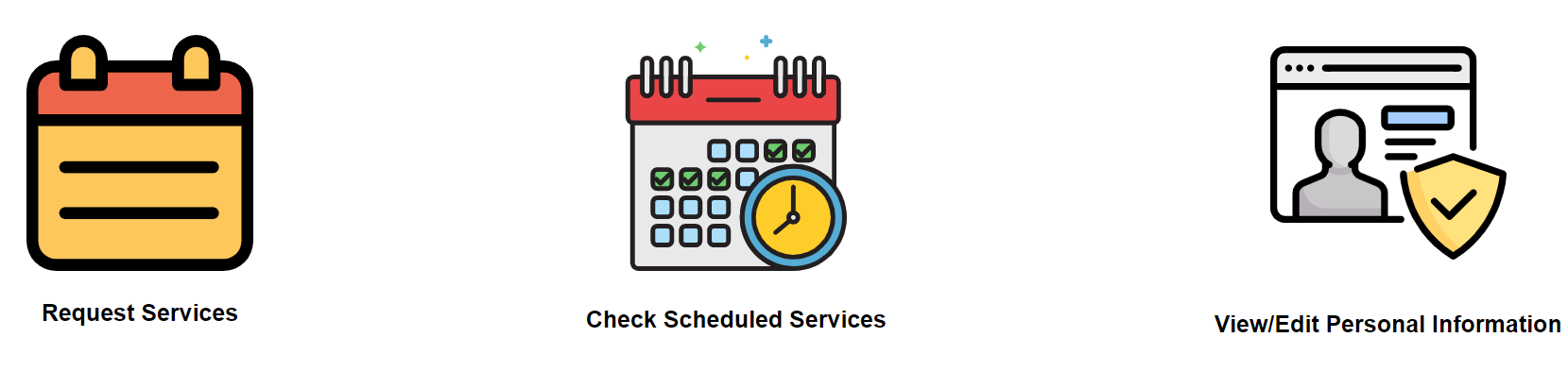
* Check personal info

A screenshot of a social media post

Description automatically generated

Customer user

This user also has three option



* Request services
  + Create service and post it on service list

A screenshot of a social media post

Description automatically generated

* Check Scheduled
  + Check all service that was created by this user
  + User is allowed to cancel scheduled service
    - If services are already taking by a volunteer, this user might become a subject of penalty. What penalty? To be determine later.

A screenshot of a cell phone

Description automatically generated

Cancelation service #1 which has been taken by a volunteer

A screenshot of a cell phone

Description automatically generated

Cancelation of service 2 which status is still vacant

A screenshot of a cell phone

Description automatically generated

* Check personal info

A screenshot of a social media post

Description automatically generated