Bookkeeping Phase 2 Report

Jennifer Brady

Matthew Dobson

Andrew Eissen

Kevin Ramirez

Christian Rondon

Steven Wu

University of Maryland University College

CMSC 495

Dr. Robinson

November 26, 2018

Contents

Achieved Functionality Milestones

Overview

As of the submission of the project phase 1 report last week, over half a dozen pull requests containing several important updates have been submitted, reviewed, and committed to the main codebase. These submissions represented over two dozen commits in total, ranging from updates to the API endpoints to the development of test code handing the integration of the front and back-ends of the application. As it stands, the codebase has received a total of 123 commits and 31 pull requests in total, with nearly 500 lines added to the codebase this last week alone. As the project nears the critical development period, these numbers are expected to increase drastically.

Implemented Back-end Functionality

Over the span of the last week since the previous project phase 1 report, the back-end team has made significant progress towards the completion of a number of appointed tasks. Perhaps most importantly, the development of API endpoints needed by the front-end to integrate the interface with the database has begun in earnest. Thus far, endpoints related to user account login, password reset, and the addition of new customers to the database have been built, tested, and integrated with the front-end via test code, with additional updates planned for the end part of the week as developer schedules permit.

Additionally, the back-end team has begun work on some of the other PHP code related to the collation of user account info and the other associated ancillary classes handling the primary bookkeeping functionality. Furthermore, additional updates will be made to the server configuration, building off of changes made over the last few weeks.

Implemented Front-end Functionality

As discussed in some detail in the previous section, the development of several RESTful API endpoints by the back-end team has led to the concurrent development of JavaScript handling functions. Each of the ascribed endpoint actions and associated in-interface scenes allows the user to oversee the passage and retrieval of data mediated through the pressing of interface buttons.

Related to this, the main JS request handler, `inaccessible.sendRequest`, has seen significant testing in its main capacity of passing data in different forms to the back-end database via `POST` requests and in the handling of returned JSON data from the server by way of `GET` requests. In the course of this testing, several refinements have been made to the process by which data is passed to the server. In the original incarnation of the function, the request handler passed JSON string data to the server. However, this was replaced by another REST-compliant method, the use of query strings to denote via a simple URL fragment string the parameters and their values to be passed to the back-end.

Other ancillary tasks undertaken this week related to the development of the JavaScript codebase include the definition of several additional scenes related to the inclusion of customer entries as well as the introduction of a general-purpose popup modal framework. This modal framework will allow for the inclusion of mini-scenes within the body of the modal related to login, account creation, document entry, and other such tasks. Lastly, the JS-mediated animation related to the swiping of content to the right during page-clearing operations has been improved slightly to allow for a cleaner, less buggy transition.

On the CSS side of the front-end house, the CSS specialists have continued their development of the main interface design. Work has also begun on the initial definition of the main mobile design based on the breakpoint widths discussed in the preceding phase 1 report, with the engineers coming to the decision to split the CSS into separate compartmentalized and encapsulating files based on each's specific role in the styling of the overall application, its child scenes, and the various supported device widths. The team also provided the group with a series of five different mockups illustrating several potential design layouts and color palates from which to choose. Though discussion is still ongoing, several contenders have emerged and merited consideration.

In order to reduce the negative effects a multiplicity of CSS file imports would have on the bandwidth, the JavaScript engineer developed a PHP file that serves as the sole CSS-related import of the application. This file concatenates all the various CSS files into one import by means of an output buffer, minifying the collective styles through the removal of whitespace, newlines, and comments with regex before loading and applying them to the page. Additionally, to reduce the risk of render blocking and to thus allow imports to load without encountering interference from each other, the `@import` statements used to load external font dependencies were removed from the root `styles.css` file and loaded in the head of the sole HTML file.

Remaining Functionality Milestones

Overview

As discussed in some detail in the previous phase 1 report, though great progress has been made toward the completion of the project milestone objectives, work remains in a number of important areas in both the front-end and back-end sides

of house. Integration of the interface and database server remains foremost in the minds of both teams, with particular attention being given this week to the development of API endpoints and the simultaneous definition of handlers capable of displaying the resultant data to the user in some shape or form.

Remaining Back-end functionality

As far as remaining back-end functionality milestones are concerned, the back-end team will continue to focus on the completion of milestones related to the RESTful endpoints discussed in some detail in the preceding sections and in the previous project phase report from last week. Additionally, though much of the underlying code related to the main bookkeeping functionality has reached a high degree of workability and level of completion, refinements will continue to be considered and implemented on a rolling basis depending on the needs of the teams as a whole and whatever issues may happen to arise during the testing phases. As in previous weeks, server configuration remains an ongoing minor background task of the back-end team.

Remaining Front-end Functionality

On the front-end side of house, the front-end team is presently concerned with the development of a few major elements of note. As stated previously, the JavaScript engineer is presently most concerned with developing action-specific handlers capable of interfacing with the server through the use of `POST` and `GET` requests. While the request handler has been proven to work beautifully through the testing of its capabilities via sample code, the use of the returned or passed data in a useful fashion is the present focus of his work. The client will want to do more than just look at returned JSON in the browser console, so the development of interface scenes and

associated handlers that display this data in a user-friendly fashion is of foremost importance until the start of the unit testing period of the production schedule.

As far as the CSS engineering team is concerned, this week will see further development and refinement of the main interface design, with particular attention being paid to the configuration of the responsive mobile design. Additionally, with the completion of the main modal popup window framework, refinements will be made to the rough popup modal styles provided in the JavaScript modal framework update to ensure all elements share the same aesthetic design ethos.