Bookkeeping Phase 3 Report

Jennifer Brady

Matthew Dobson

Andrew Eissen

Kevin Ramirez

Christian Rondon

Steven Wu

University of Maryland University College

CMSC 495

Dr. Robinson

December 3, 2018

Contents

Achieved Functionality Milestones

Overview

Over the span of the last week, the bookkeeping project team has pushed to master a total of 9 pull requests representing some 14 commits and 2,451 new lines of code. These changes range from improvements to the overall design of the application to the development of new back-end API endpoints to the addition of new modal mini-scenes. Having tested a variety of these new functions, the team is confident going forward that all deadlines will be met and all milestones will be reached prior to the submission of the production build of the project. Under present circumstances, a total of 158 commits have been pushed to master and 40 pull requests successfully merged.

Implemented Back-end Functionality

Since the writing of the last project phase report, the back-end team has made significant progress in the implementation and integration of the back-end database with the front-end interface. To this end, the team has overseen the addition of several new API endpoints allowing for the retrieval of extant customer and vendor listings and the addition of new entries, a total of 4 new endpoints fittingly named `add_customer`, `add_vendor`, `get_customers`, and `get_vendors`. In conjunction with these updates, the team has also made some adjustments to the `PhpConnection` file in an effort to more readily facilitate the transmission of data to and from the front-end.

Additionally, some updates have been made to the back-end server configuration as in weeks past, as well as some minor updates to the actual bookkeeping functionality required for the manipulation and evaluation of user data in the interface. Some minor unit testing has been undertaken as well, though most is planned for the final week.

Implemented Front-end Functionality

As far as updates to the front-end side of house are concerned, many improvements have been made over the span of the last week towards the facilitation of user interaction with the database by means of modal scenes. Starting with the addition and refinement of the general-purpose modal framework, the JavaScript engineer has overseen the development of several new modal “mini-scenes” related to the changing of the user password, the addition of new customer and vendor entries, and the passage of new document data to the back-end server. Though work remains to be done on the document addition modal (pending the back-end team’s ongoing development of the requisite endpoint), these modal scenes and their associated action handlers and event listeners have been tested and proven capable of interacting with the server, handling returned success/failure JSON responses to `POST` requests as expected through the display of success or failure status messages in the modal.

As far as the CSS engineering team is concerned, several major updates have been made and committed to the codebase since the last phase report. Foremost among these is a major update to the mobile application design, allowing for the display of the main dashboard general ledger on mobile devices in both conventional portrait and landscape views. Additionally, with the concurrent addition of a “print ledger” button to the upper right-hand navlinks bar by the JavaScript engineer, a dedicated `@media print` view has been developed as well, allowing users to print out a clean and readable version of the currently viewed ledger for documentation purposes. With the commit of the main modal framework to the codebase, CSS related to the display of the modal on the page and the darkening of the background has been committed as well.

Remaining Functionality Milestones

Overview

As far as remaining milestones are concerned, the project group's focus remains fixed on the successful integration of the back-end database with the front-end interface, allowing of the transmission and request of user data and the display of that data on the page. As such, most remaining milestones are related to the development of the remaining API endpoints and the relevant page scenes used to display the data returned from the endpoints and associated `GET`/`POST` requests.

Remaining Back-end functionality

The back-end team is presently focused on the creation of the several remaining endpoints required for the implementation of the full-range of bookkeeping functionality. These endpoints are related to the addition of new documents of the five main types, the user logout handler and associated security-based functionality, and the joint removal of specific document transaction entries from the front-end ledger table and back-end database table. Additionally, as in weeks past, development continues on the server configuration, building off of work completed since the first two phase reports.

Remaining Front-end Functionality

Related to the remaining back-end milestones, the front-end JavaScript engineer will also be singularly focused on facilitating the integration of new front-end interface scenes with the back-end endpoints on a rolling basis as they are written. At present, the remaining scenes and mini-scenes requiring implementation are those related to the display of a documents listing in the dashboard table and an account creation modal or module. Refinement of existing scenes and handlers is needed for the account logout listener and the documents addition handler, with the latter placed on standby pending the completion of the appropriate endpoint by the back-end team. Other ancillary fixes and additions requiring some attention include the handling of automatic modal closing on clicks outside the modal window, the addition of a documents listing and general ledger toggling button to the left-hand sidebar, a general removal of some of the vestigial “magic numbers” and “magic words” in the body of the `app.js` file to dedicated enums for ease of readability, and the addition of a production build minified `app.min.js` file once the JS code is mostly completed.

Regarding remaining CSS functionality, the front-end team specialists are continuing their refinement of the mobile view in accordance with content portability and responsive design best practices as stated in previous reports. Additionally, work continues on the ongoing development of the main desktop views of the application. Though most of the major elements have seen significant styling, the team is continuing in its efforts to refine present styling while simultaneously handling the addition of new scenes and functionality to the interface on a rolling basis.