Bookkeeping Project Plan

Jennifer Brady

Matthew Dobson

Andrew Eissen

Kevin Ramirez

Christian Rondon

Steven Wu

University of Maryland University College

CMSC 495

Dr. Robinson

November 1, 2018

Table of Contents

Software Description

Purpose

This document provides an overview of the front and back-end development for the CMSC 495 Bookkeeping Project. This project plan outline describes the intended functionality of the project as well as the organizational framework required for project implementation.

Scope

The end goal of the Bookkeeping Project is to provide a web interface for our client to perform simple bookkeeping for their small business. This tool will allow our clients to record invoices from and payments to vendors, and invoices to and payments from customers, and other administrative entries using full-accrual, double-entry bookkeeping. This data will be used to create a simple balance sheet for the client showing their assets, liabilities, and equity.

Our objective is to create a user-friendly experience with future upgrade potential as our clients expand their businesses. By utilizing our project, our clients will more easily be able to track expenditures and expected income. This utility will, hopefully, allow our clients an easier experience in tracking their finances and paying their taxes.

Implementation Requirements

System Specifications

- Development Languages and Technologies:
    - Back-end: PHP, SQL, MySQL
    - Front-end: JavaScript, HTML$_5$, CSS$_3$
    - Server: XAMPP Apache 5.6.14

- Supported Operating Platforms:
    - Chrome 70
    - Firefox 62

Requirements Description

A number of different technologies and programming languages are required for the successful implementation of the project specifications. As a database is required to handle the storage and retrieval of user-input data depending on the specific operations being performed, the project will be making use of the SQL and PHP languages for the manipulation, collation, evaluation, and eventual output of user data. On the front-end, HTML$_5$ and CSS$_3$ will be used in conjunction with JavaScript to handle the display of all graphical elements and ancillary user functionality required to facilitate user interaction and program manipulation.

In order to oversee the marriage of the back-end database to the front-end user interface and permit the easy configuration thereof, a private server capable of handling the full range of display and data-definition functionality is required for the duration of the project's development. To this end, the XAMPP Apache web server bundle (November 13, 2015 build 5.6.14) will be employed by the developers during the design and unit testing periods to assist in debugging tasks and test-running purposes. Once the application has been completed and a production build assembled, the application will be deployed to a free hosting site for external access by client users.

On the client side, the only requirement imposed on potential users will be the possession of an up-to-date supported modern browser with the “JavaScript enabled” permission set to true. At the present time, supported browsers include the latest versions of Google Chrome, Mozilla Firefox, Microsoft Edge, and Opera. Chromium-build alternative browsers like Blisk and Vivaldi will likely also enjoy support, though their custom functionality may interfere with the project's own functionality.

Functionality

Back-end Functionality

As far as database functionality is concerned, the back-end side of the program design will support a number of operations and applications that may be undertaken by the client as desired, with progress saved via a user account. These applications include the recording of invoices and receipts, the development of balance sheets collating the relevant data, and the rendering of transaction logs. All functionality is undertaken via the back-end codebase and displayed and rendered via the front-end user interface.

Front-end Functionality

As far as front-end functionality is concerned, the interactive web application interface will be designed to be as readily usable and understandable as possible from the outset, making use of an organizational layout that is logically developed in accordance with basic user expectations. This will be manifested in the program in the form of a navigational toolbar containing a menu of potential bookkeeping applications from which the user may choose. In addition to this options listing, the toolbar will also contain input textboxes related to user account login and session storage, allowing the user to log in and out of the application after saving any progress.

In terms of accessibility, the front-end interface will support the use of a multiplicity of devices, ranging from mobile devices to conventional laptops to external monitors of multiple widths. In accordance with the principles of responsive design and content portability, the look and feel of the application will not change substantially depending on device width, meaning that the user will be able to successfully make use of the program's functionality without having to rely on a specific device. This will ensure that the application is readily and reliably available at any point to the user.

Entity Relationship Diagram

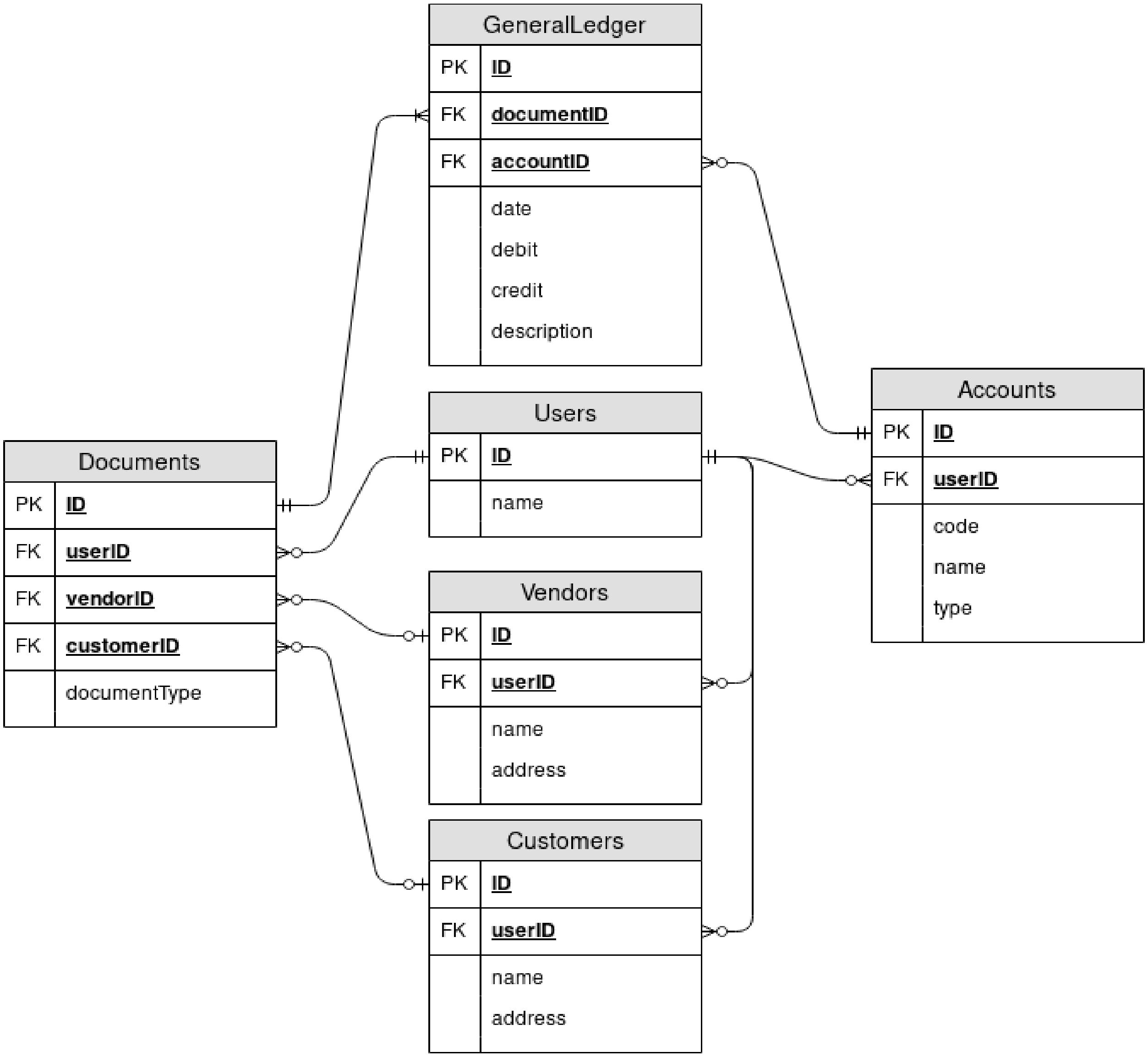

Figure 1. Entity Relationship Diagram. This figure illustrates the back-end structure.

Use Case Scenarios

Case 1 (Make an entry)

Primary actor: A user

Description: The user enters a new document name, the document type, a vendor or customer name if necessary, and the debit and credit lines of the entry. The bookkeeping system adds these values to the `Documents` and `GeneralLedger` database tables.

Case 2 (Pull a report)

Primary actor: A user

Description: The user specifies through filters what `Document`(s), `GeneralLedger` line(s), `Account`(s), or `Vendor`(s) he or she wants to view. The bookkeeping system translates these filters into a SQL query and displays the results to the user in an interface modal.

Case 3 (Run a balance sheet)

Primary actor: A user

Description: The user specifies a date on which a balance sheet is to be made. The bookkeeping system produces a balance sheet from the user's data in the `GeneralLedger` table.

Project Organization

Project Deliverables

- Project Plan
- Software Design Information
- Software Test Documentation

- Bookkeeping Database
- Customer Web Interface

Major Milestones

- Back-end milestones
  - Commit and retrieval of sample dataset via database
  - Manipulation, evaluation, and basic display of sample dataset
- Front-end milestones
  - Finalization of interface design
  - Development of mobile view
  - Development of desktop view
- Refinement milestones
  - Unit testing of application functionality
  - Optimization of functionality and running of browser speed tests
  - Documentation of codebase, other cleanup tasks
  - Final checks, submission and deployment of production build

Milestones Description

In order to facilitate the orderly execution of the project requirements, a set of waypoint milestones has been developed, the use of which is intended to aid the development group as a whole in proper planning and timely execution. These milestones, undertaken by either the front-end or the back-end team depending on the development stage, are related to the various pieces of functionality required by the program to function as intended and generally involve the application of different languages and technologies to the development phase in question.

The first milestone concerns the development of the primary back-end code related to the storage and retrieval of user data. The collation, manipulation, and evaluation of that data will be the primary focus of this milestone, allowing the team to gauge whether or not work may continue on elements related to the display of this data. This initial milestone will ensure that a sample data set can be saved to the database and retrieved successfully, and perhaps even displayed in a rough manner via some rough hardcoded HTML framework.

The second milestone will also be related to the back-end side of the development process, specifically to the rendering of the retrieved data in a graphical form conducive for interaction on the part of the clients and assorted users. This will likely make use of a yet-undefined API or library capable of displaying easily-readable and cleanly-notated diagrams, balance sheets, and transaction logs, among other such bookkeeping graphical elements.

Once it has been confirmed that the program is fully capable of accepting and rendering user-input data to the satisfaction of the group, work will begin in earnest on the front-end development of the user interface, the primary means by which the user/client making use of the web application will be able to interact with the data. The first front-end milestone in question will be completed once the complete design schema (both the mobile and desktop designs) has been finalized to the satisfaction of the group. A template mockup will be the end result of this milestone, a design that will be detailed enough to provide direction to the implementing front-end development team.

In accordance with front-end development best practices and the principles of the device portability and responsive design paradigms, the mobile view will be given initial priority and developed first, tested as needed on different device widths using emulation tools or actual physical mobile devices as appropriate. Given a more confined space in which to place user interaction elements and display modals, content will be prioritized and organized in such a way as to make the site's functionality still highly interactive despite the smaller device size, ensuring that the user is still able to make use of the features without having to engage the device's "desktop view” mode. The milestone will be reached upon the construction of a mobile view that works without error on all tested devices.

The next milestone related to the design will concern the creation of the desktop design view, with various screen widths and resolutions handled at specific breakpoints. In all likelihood, these will be the best-practice breakpoints defined by the Bootstrap CSS framework, breakpoints which are in large part based on the conventional laptop screen sizes and the scaling levels of conventional external monitor widths. The desktop view, like the mobile view, will adopt an organizational ethos that would naturally make sense to the user and ensure that functionality, diagrams, and the general display of data are all handled logically as the user would expect.

Once the back-end and front-end development has been completed to a reasonable degree of reliability, the next milestone will be set, one related to the unit testing and debugging of the entire application. The team will transition into a testing phase, in which the application will be run under extreme conditions with the end goal of breaking it to learn its specific weaknesses. The milestone will be reached upon the successful patching of all major bugs brought to light by these high-stress tests.

Once unit testing has been completed, the optimization of the codebase will be the concern of the next milestone. In order to ensure that the application loads and runs quickly so as to not leave the user/client waiting for data to be crunched and results to be displayed, optimizations will be undertaken where appropriate to streamline any potential communication weaknesses lying between the database and the display. Once this has been accomplished and the difference in speeds illustrated via a series of browser console speed tests, the milestone will have been successfully reached.

The final milestone will be the mass-documentation of the codebase in the event that some documentation has been inadvertently missed or forgotten in the course of implementation. In conjunction with this task, any last minute changes that must be made prior to the production build being committed, submitted, and deployed off-site will be handled and reviewed as needed. Once the project has been completed, the final milestone will have been reached and the process brought to an end.

Member Assignment

Table 1

An illustration of group members' primary duties and secondary roles

Group Member	Primary Role	Secondary Role
Jennifer Brady	Front-end developer, HTML focus	Documentation
Matthew Dobson	Back-end developer, PHP/SQL focus	Lead database design
Andrew Eissen	Front-end developer, JavaScript/CSS focus	GitHub maintainer, Documentation
Kevin Ramirez	Back-end developer, SQL unit testing	Database design
Christian Rondon	Front-end developer, HTML/CSS focus	Documentation
Steven Wu	Generalist developer, Main unit tester	Documentation, Debugging

Member Description

Project development milestones will primarily be completed by several teams, each with a specific focus that will last for the duration of the project. As in many dev shops, the design and development of the user interface and the means by which data is displayed will be handled by the front-end development team, while the development of the data storage and retrieval process via databases will be handled by the back-end development team. Members of both teams will pitch in to assist in ancillary development tasks as they arise, ranging from unit testing to documentation to bug fixing, aided in this task by a dedicated unit tester and development generalist.

The back-end team is composed of Matthew Dobson and Kevin Ramirez. Matthew Dobson, the project lead, serves as the main database schema developer and senior back-end developer, specializing in the use of PHP and SQL and the development of the database design. He is aided by Kevin Ramirez, a back-end developer overseeing the SQL unit testing process and tasked with providing secondary assistance in database design.

The front-end team is composed of Andrew Eissen, Christian Rondon, and Jennifer Brady. Andrew Eissen, in addition to serving as GitHub maintainer and documentation specialist, contributes to the team as a front-end developer with a specialized focus on JavaScript and CSS. Christian Rondon, a developer with experience in languages used by both development teams, will handle CSS and HTML duties alongside fellow HTML developer and documentation specialist Jennifer Brady.

Both teams are aided as needed by dedicated unit tester and generalist developer Steven Wu, a developer with experience in languages used by both teams.

His focus will include filling the gaps between groups and handling a host of ancillary tasks related to documentation, debugging, and pull request reviewing.

Project Management

Version control will be handled through a dedicated GitHub organization and master project repository. In order to assist the development team in cleanly and clearly documenting changes to the repository contents, all members are required to submit pull requests containing their proposed changes and associated commentary. These pull requests must be reviewed by at least one other group member before the proposed changes can be committed to the main master branch, ensuring that team members are both kept aware of ongoing code developments and closely involved in the process. The ability to either approve or request changes to proposed updates encourages group communication and conversation while neatly documenting all resultant discussion for future reference. The project repository may be found at this location.

As far as more immediate conversation is concerned, communication between members of the project team is undertaken via a mixture of Slack, Google Hangouts, and email. Slack was specifically chosen as the primary means of discussion due to its inbuilt ability to divide specific conversation topics into separate workspace channels, allowing for the easy division of content into discussion related to development, issues, pull request information, and general talk. Additionally, the use of Slack apps to integrate both Google Calendars and GitHub enables easy monitoring of off-Slack team activity, keeping members in the loop regarding ongoing code developments and design evolution.

Project Schedule

Table 2

An illustration of all important course-defined dates

Task	Duration (days)	Start	End
Project Plan	7	October 29, 2018	November 4, 2018
A. Software Scope	5	October 29, 2018	November 2, 2018
B. Software Languages	5	October 29, 2018	November 2, 2018
C. Update ER Diagram & Attribute Table	1	November 2, 2018	November 2, 2018
D. Assign Roles	1	October 30, 2018	October 30, 2018
E. Use Case Scenarios	2	November 2, 2018	November 4, 2018
F. Functional Requirements	2	November 2, 2018	November 4, 2018
User's Guide and Test Plan	7	November 5, 2018	November 11, 2018
A. Develop Test Cases	5	November 5, 2018	November 9, 2018
B. Write Test Plan Document	1	November 10, 2018	November 10, 2018
C. Review	1	November 11, 2018	November 11, 2018
Project Design	7	November 12, 2018	November 18, 2018
Phase 1 Development	7	November 19, 2018	November 25, 2018
Phase 2 Development	7	November 26, 2018	December 2, 2018
Phase 3 Development	7	December 3, 2018	December 9, 2018
Final Review and Testing	7	December 10, 2018	December 16, 2018