Project Summary Document

Joseph Maxwell

Marcus Benesch

1. Introduction - overview of document
2. Project organization - discuss who eventually ended up doing what and explain if roles had changed from what was planned at the start of the project
   * Because there were only two members, tasks were divided into two buckets.
   * Front-end defined any task that was visible to the end-user.  This involved the html for the webpage and the ability to pull and manipulate data from the database.  Originally the goal of the front-end was grander than was possible
   * Back-end defined any task that was not visible to the end-user.  The database, webhosting, and maintenance of the first two.
3. Risk analysis - discuss the identified risks and which ones turned out to not be a problem and which one did, as well as other unanticipated problems that cropped up. For the problems that you encountered, describe the nature of the problem and what you had to do to deal with it. You can also talk about how you would write out the risk management plan differently if you had to do it over again.
   * The platform I used had compatibility issues
4. Hardware and software resource requirements - discuss libraries and tools you had to install to get the app working. Provide any special instructions for getting the application to build and run.
   * Tools used:
     1. JAVA: For the database updater program.
        1. Built BillTrackerDB, this allows for automatic pulling and pushing of data to the database. With this I used task scheduler to repeat the program daily at 7:53PM daily.
     2. Internet Information Services: Webhosting
        1. Within IIS I modified the web.config file to allow for virtual folders that vuejs uses to dynamically modify webpages.  In addition, I added a modifier to CORS protection that should have allowed for CORS to occur.
     3. Microsoft JDBC Driver for SQL Server: Connecting to the database using JAVA
     4. SQL Server Management Studio: Access to and management of the database
     5. Microsoft SQL: Engine to run and host the database
5. Work breakdown structure (WBS) - give a high-level view of the work breakdown, referring the details to your tracking system
   * There was mostly just division of tasks and each person was expected to do their job for each milestone.  Tracking was done weekly through the meeting on updates to the group project and automatically through github’s traffic tracking.  Each push and pull were recorded.  Whenever a team member pushed to the github, they could add a name and description of what changes were done.
6. Project schedule - summarize what functionality was available at each Code Milestone
   * Milestone 1:  There was no functionality at this point, the web host was functional and had a static webpage.
   * Milestone 2:  The database design was finalized, and the front-end webpage was planned.
   * Milestone 3: The database now was fully automated where bills were automatically pulled and submitted.  The front-end now displayed partial information from the database.
   * Milestone 4:  The front-end was integrated with the back-end and search functionality is implemented.
7. Monitoring and reporting mechanisms - discuss how, in practice, your team monitored project progress (i.e., which ones turned out to be most useful: weekly meetings, logs, tracking system, version control, others)
   * Monitoring and reporting were done weekly with prodding by each member in the days approaching the submission dates.
8. [Optional] Miscellaneous - anything else not covered by the above
   * Unlike many sane people, we built this program effectively from scratch. This cause all sorts of headaches that could easily have been avoided by using a template for a webservice that was freely available online.