

Homework 5

Group 10

Class: CS-361

I. User Stories: Sorted by high, medium, and low priority.

- A. **(High)** Adding Notes - As a user, I want to be able to add notes to an expense or spending of my choosing for future references if needed.
- B. **(High)** Drop down list – As a user, I want a drop down list of banks. This pre populated list of names will make it easier to just choose or select the bank name when inputting my financial information in the application.
- C. **(High)** Track Spending- As a user, I want to be able to see my spending habits each month. If I can view what I have spent I will be able to make adjustments based on this and the recommendations from the application.
- D. **(High)** Large Purchases- As a user I want to see what I can afford when it comes to large purchases such as: car, home, or some other expensive item. I want to know the options I have to buy said item and if I need to take out a loan. The system should tell me if making this large purchase will over extend my budget too much and put me into too much debt.
- E. **(Medium)** Outstanding Credit Card Debt - As a user I want to be able to see how much credit card debt, if I have any, and how long it will take to pay off based on a few parameters: My income, interest rate on card, and my spending each month. I should be able to see if I entered a certain amount how long it will take to pay off or if it is better to move all my debts to a single card and go from there.
- F. **(Medium)** Savings Goal feature - As a user, I want to be able to have a feature in which I can easily set an amount to save for a larger expense like vacation or holiday within a time frame. If I wanted to save X amount of dollars in Y months I can do this with convenience - maybe even an entire list.

- G. (Medium)** Monthly Forecast - As a user, I want to have a function which displays my potential monthly spending. If I can reduce spending in one category and it updates immediately, I can see the benefits and have a better idea of the state of my finances. On the other hand, if I increase spending, I want to see how much it affects my budgets overall.
- H. (Medium)** Loans- As a user, I want to be able to compare loans so that I may be able to choose the best option for my situation. I want to see a few options listed based on recommendations for me. I would like to see the interest rates, and a time frame of how long it would take to pay off based on my spending and income.
- I. (Medium)** Update my information- As a user, I want to be able to update my information such as: income, bills, credit cards, and general information so that I can continue to make changes to spending. I might have changed jobs, moved, or added more bills and I should be able to make adjustments to the system so it reflects that.
- J. (Medium)** Lost login- As a user, I want to be able to recover my login information in the event I forget my password. This is standard on most if not all apps and should be used here. Either by email, phone, or questions the app can ask.
- K. (Low)** Credit Cards- As a user, I want to be able to apply for a credit card based on the recommendations from the system. I want to have access to a few options listed and then have the link to the card site or apply through the app.
- L. (Low)** History Chart feature - As a user, I want see a histogram chart of my expenses. I want to see the chart display month by month expense amount, and as well as expense categories (i.e. electricity, mortgage, credit card, rent, water, phone, etc).
- M. (Low)** Bill Pay- As a user, I want to be able to pay my bills through the system. I could link my card or bank account to it so that it will automatically do it or just be able to manually pay my bill through the app.

- N. **(Low)** Automatic Save feature - As a user, I want to be able to have the option to deduct a certain amount of money from my income to be saved every pay day. This way I can save money without the need do it myself, thus making it more likely I will actually save. Here we can use a percentage or static amount.
- O. **(Low)** Financial Mock-Up - As a user I would like a feature that mocks up what-if scenarios to be able to set goals. For instance, if I were to make \$X amount of dollars, what kind of budget would I be working with and what I would be able to afford with that amount.
- P. **(Low)** Price Search - As a user, a feature that assists me in finding lower prices on potential purchases would be useful. I would be able to save money and apply that other expenses or savings. If I were to purchase say a TV on black friday, how much would I save than if I were to buy it this weekend?

II. Each story: Time-frame and task list

Adding Notes:

- High priority, due next week
- Set up a table in the database to store the notes
- Create a hyperlink with the wording “Add Notes”, so the user can click on
- Create a text box or an input form for users to enter text
- Create a submit or save button below the text box
- Effort estimates = ½ day

Drop Down List:

- High priority, due next week
- Set up a table in the database with the pre populated values of banks name
- Set up a connection to the database to read the data
- Create a drop down list in the browser
- Effort estimates = ½ day

Track Spending:

- High priority, due next week
- Set up a table in the database to house the spending expenses
- Set up a connection to the database to read the data
- Create a Track Spending button or feature in the app/browser for user to click on

- Create the spending categories in the app/browser
- Grab the spending data and display it in the browser according to its category
- Effort estimates = 1 day

Large Purchases:

- High priority, due next week.
- Create a Large Purchases feature button or feature in the app/browser for user to click on.
- Create large purchases categories list; car, home, boat, etc.
- Display the dollar value available to make the purchase for each categories for the user to see when clicked on.
- Create the word “Debt” and display the dollar value within each categories, to let the user see if they can afford it, or how much debt will the purchase create.
- Effort estimates = 2 days.

Credit Card Debt:

- High priority, due in two weeks.
- Create a database entry where user can enter desired debt reduction.
- Create the algorithm necessary for the debt and payoff calculations
- XML request to bank to check/get funds for payment
- Effort estimates = 3 days

Savings Goal feature:

- Button has to be created to access feature.
- Set up connection to database with entry.
- New entry for each purchase that they want to save for
- Function to calculate amount wanted to be saved
- Effort estimates = 2 days

Monthly Forecast:

- Function that takes the averages of the users spending then displays them back
- Call to the database
- Effort estimates = 2 days

Loans:

- Call to websites to get loan information.
- Page to display options shown
- Buttons for updating, options, and submitting
- Effort estimates = 7 days

Update my information:

- Update button on info page
- Function that calls the database to update the info. Using SQL in back end.
- Effort estimates = 1 days

Lost login:

- Button on login page to access recovery page.
- Function that stores email/phone for recovery of password.
- Function that gives the user questions to answer that they previously answered
- Effort estimates = 5 days

Credit Cards:

- Set up Questionnaire with radio buttons.
- Function to process info and sent to server.
- Page to display options
- Effort estimates = 10+ days

History Chart feature:

- Function to collect information from database.
- A page to display graphical charts for previous month
- Option to search by month
- Option to display different type of graphs
- Effort estimates = 2 days

Bill Pay:

- Set up functions that do XML requests for bill information.
- Function that gets user bank info to process.
- Page to display confirmation
- Effort estimates = 10+ days

Automatic Save feature:

- Create an input box in which allows the user to set amount they want to save, a number or percentage of income
- Display information as well as keep track of how much was saved - function within database that calculates in a selected time frame.
- Function that calls database to update information based on user preference/input
- Page to display current amount that is being saved.
- Effort estimates = 5+ days

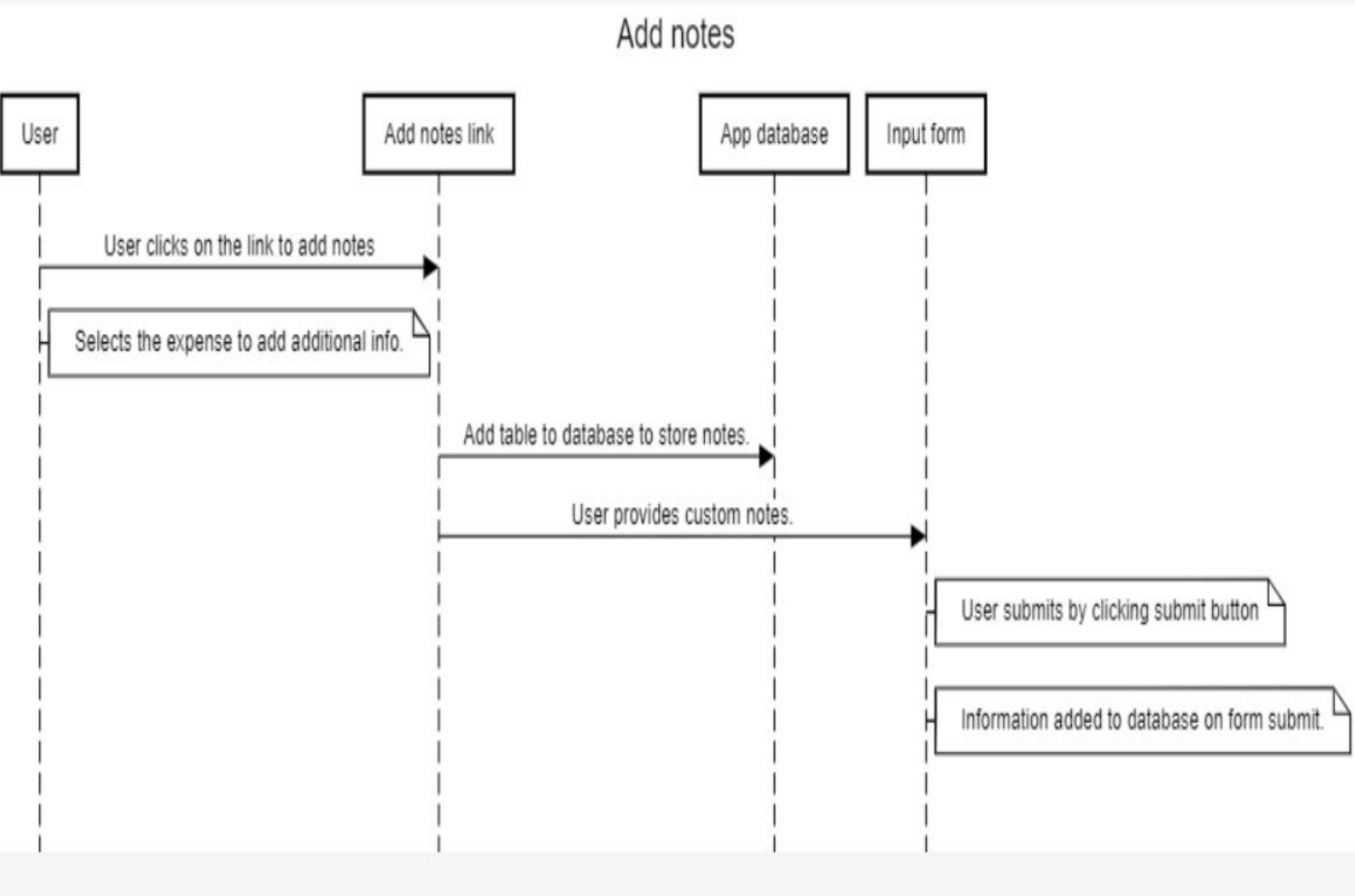
Financial Mock-Up:

- A button that goes to a screen with multiple input boxes.
- Algorithm that calculates the input boxes to display a page in which all categories like income, expenses, etc are shown.
- Button to reset all inputs.
- Export or save feature.
- Effort estimates = 3+ days

Price Search:

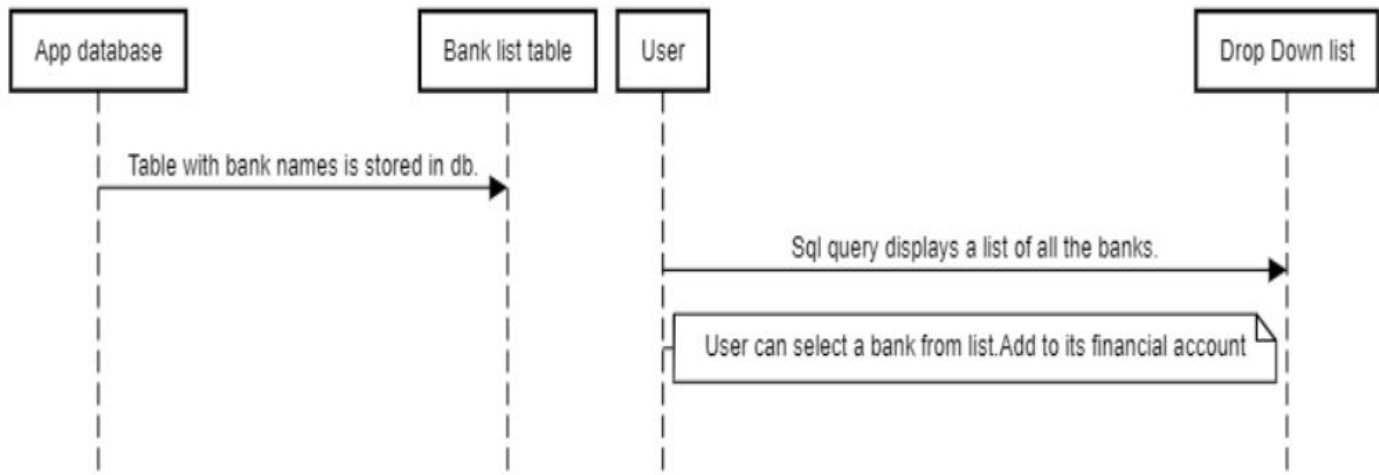
- Button to access features within this function.
- Search box that uses a search engine to look for item.
- Scanner button to access camera to scan - QR or barcode - to search the web for identical items.
- A save/export button to keep track of items.
- A screen display to show prices differences/savings amount.
- Effort estimates = 10+ days - would have to incorporate search engines and prices on external pages.

III. UML Sequence for stories due next week



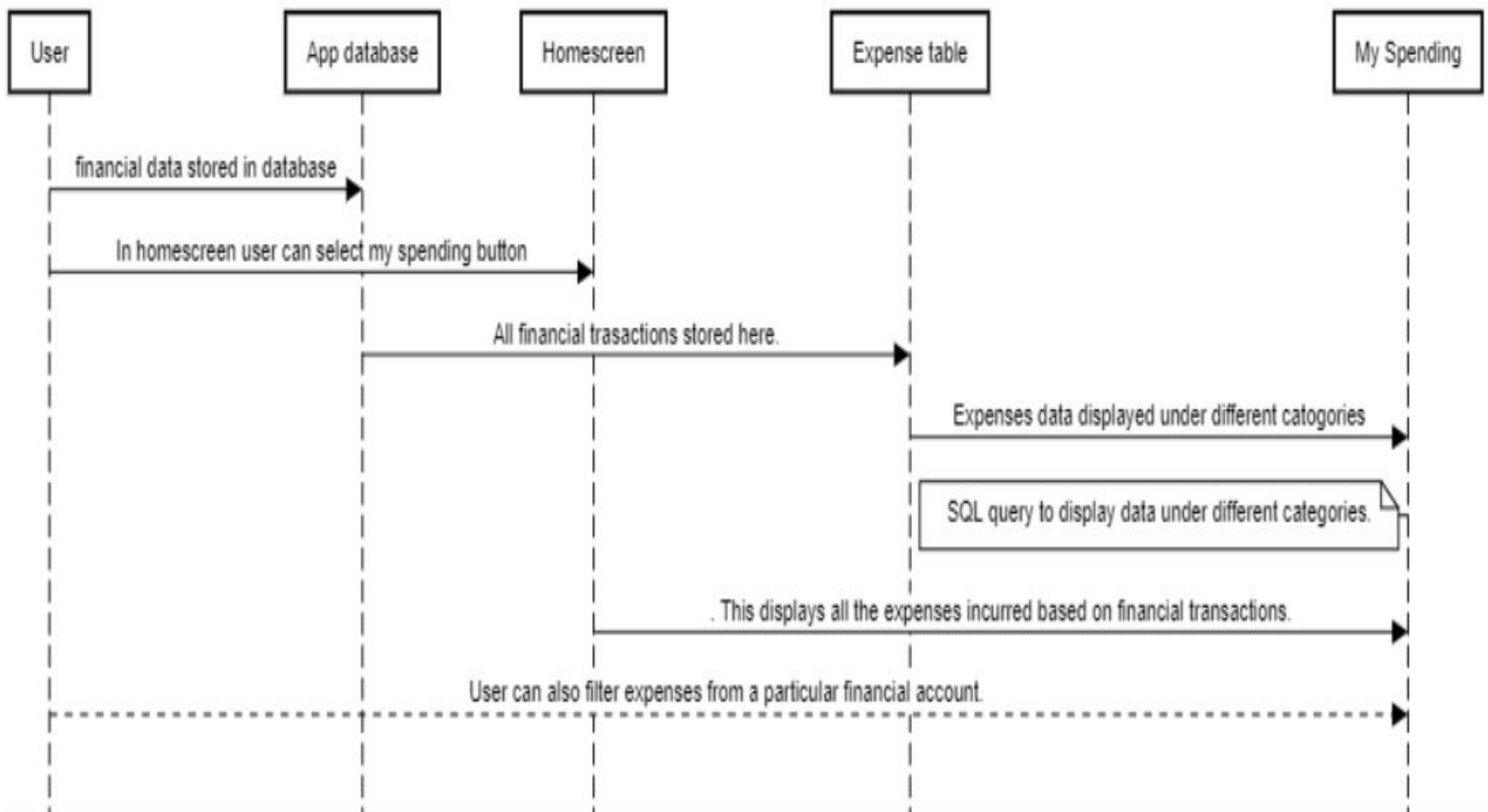
UML diagram 1: Add notes sequence chart

Drop Down List

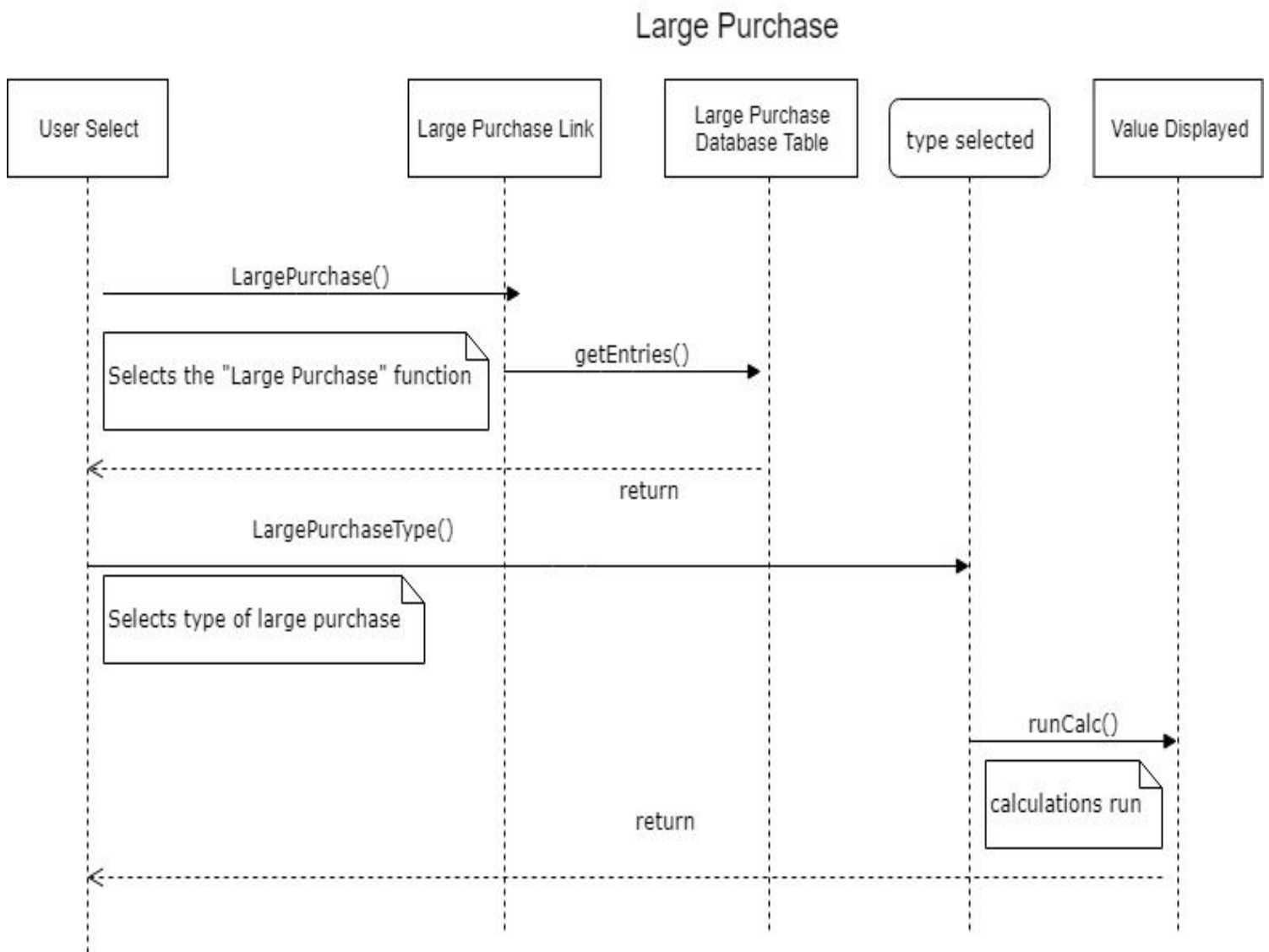


UML diagram 2: Add Drop down list.

Spending tracker



UML diagram 3: Spending Tracker



UML diagram 4: Large Purchase

IV. Plan for Implementing these stories

We have four high-priority user-stories due at the end of next week. These user-stories vary in terms of tasks to be accomplished and estimated effort.

Our team also has five very unique individuals, with very specific skills and technical fortes. As such, we are going to be grouping tasks by type, and then we will creating working (mostly in overlapping pairs of team members) to accomplish each grouping. We have made initial assignments for each effort (noted below).

Database Setup and Testing:

- Create a table with appropriate attributes for user-entered notes (for “Adding Notes” user story)
- Create a table with appropriate (defined in HW4) attributes for individual financial institutions (for “Drop Down List” user story)
- Create a table with appropriate attributes and entries to store user expenses and income entries (for “Track Spending” user story)
- Create “large purchases” categories list include; car, home, boat, etc. and add to “large purchases” table (for “Large Purchases” user story)
- Create a “cash_flow” table with default spending and income categories

Front-End Development:

- Create a hyperlink with the wording “Add Notes” (for “Adding Notes” user story)
- Create a text box or an input form for users to enter text (for “Adding Notes” user story)
- Create a submit or save button below the text box (for “Adding Notes” user story)
- Create a drop down list in the browser (for “Drop Down List” user story)
- Create a Track Spending button or feature in the app/browser for user to click on (for “Track Spending” user story)
- Create a Large Purchases feature button or feature in the app/browser for user to click on. (for “Large Purchases” user story)
- Create the word “Debt” and display the dollar value within each categories, to let the user see if they can afford it, or how much debt will the purchase create. (for “Large Purchases” user story)

Back-End Development:

- Set up a connection to the database to read and modify database entries (for “Drop Down List” user story)
- Retrieve spending data from user-expenses-tracking database and return it for display in the browser according to its category. (for “Track Spending” user story)
- Retrieve the dollar value available (from user-financial information) to make the purchase for each category for display in the browser according to its category. (for “Large Purchases” user story)

We will assign two team members to develop and verify the database tables. This will likely require the smallest effort of each of these three groups of tasks, so these two team members will be re-assigned to help with other tasks early on in the week (once this is complete).

- Database Setup and Testing will be done by Mike and Forrest. This is a critical path, which needs to be completed early, to allow for the other tasks to continue.
- Front-End Development tasks are the largest group of tasks, and will be done by Thach and Khoa, with help from Mike once the database is setup and tested fully.
- Back-End Development tasks will be started by Aseem, who will then be joined by Forrest once the database tables are established and tested.

V. Customer Interaction

We were able to meet with our customer on Wednesday evening, and received some excellent feedback - including help with prioritization of our users stories.

VI. Team Member Contributions:

- Aseem Prashar: UML Sequences
- Forrest Allen: Stories, Time-Frame & Task list
- Thach Vo: Stories, Time-Frame & Task list
- Khoa Phan: Stories, Time-Frame & Task list
- Michael Volz: UML Sequence, Plan for Implementation