

Project Final Report

CSE 6242 Data & Visual Analytics

Rahul Agrawal, Yi Ding, Holly Williams, and Anqi Zou

Introduction & Motivation

The explosive impact of social media and e-commerce has led to the propagation of sharing behaviors, especially financially, among individuals. Bundling and sharing living space are typical examples of such behaviors. A broad array of users including students, housemates, and even couples are seeking easier tools to track personal and shared expenses. Within the scope of this project, we will develop a prototype of such an application that delivers an optimized solution to the personal and shared expenses problems and helps users visualize patterns embedded in their spending behaviors.

Problem Definition

Through this project we will build the prototype of a personal financial management application that gives the users the ability to:

- Keep track of expenses, personal and shared
- Auto-categorize expenses to help organize data.
- Visualize data across different dimensions such as time, categories, and transactions shared with other users
- See trends and projections for expenses

Survey

Although there are a significant amount of applications specialized in personal finance management in the market today, we surveyed the most popular ones shown in Table 1. Many of these apps have some useful features, but none of them are complete enough for managing both shared and personal finance.

Table 1. Features Comparison of Various Existing Tools

Features \ Applications	<u><i>Mint</i></u>	<u><i>Buxfer</i></u>	<u><i>BillMonk</i></u>	<u><i>Splitwise</i></u>	<u><i>ClearCheckbook</i></u>
<i>Personal Finance Management</i>	Yes	Yes	No	No	Yes
<i>Shared Expenses/IOU</i>	No	Yes	Yes	Yes	No
<i>Visualization</i>	Basic	Basic	No	No	Basic

<i>Trend Projections</i>	No	Yes	No	No	No
<i>Auto-categorization of expenses</i>	Yes	No	No	Yes	No
<i>Reminders, Budgeting</i>	Yes	Yes	No	No	Yes

The comparison table above shows that none of these existing applications fully-support the features we proposed for managing expenses. Additionally, we also observed two areas for improvement. Firstly, visualization functionality on these apps is fairly basic and does not give much insight into user data. Almost all of them present a single table view of the expenses, without offering functionality to visualize the data across different dimensions such as time, expenses shared with a particular friend, and so on. Secondly, the systems aren't fully capable of intelligent categorization and prediction. The machine learning based features such as auto-categorization of expenses could be yet not implemented.

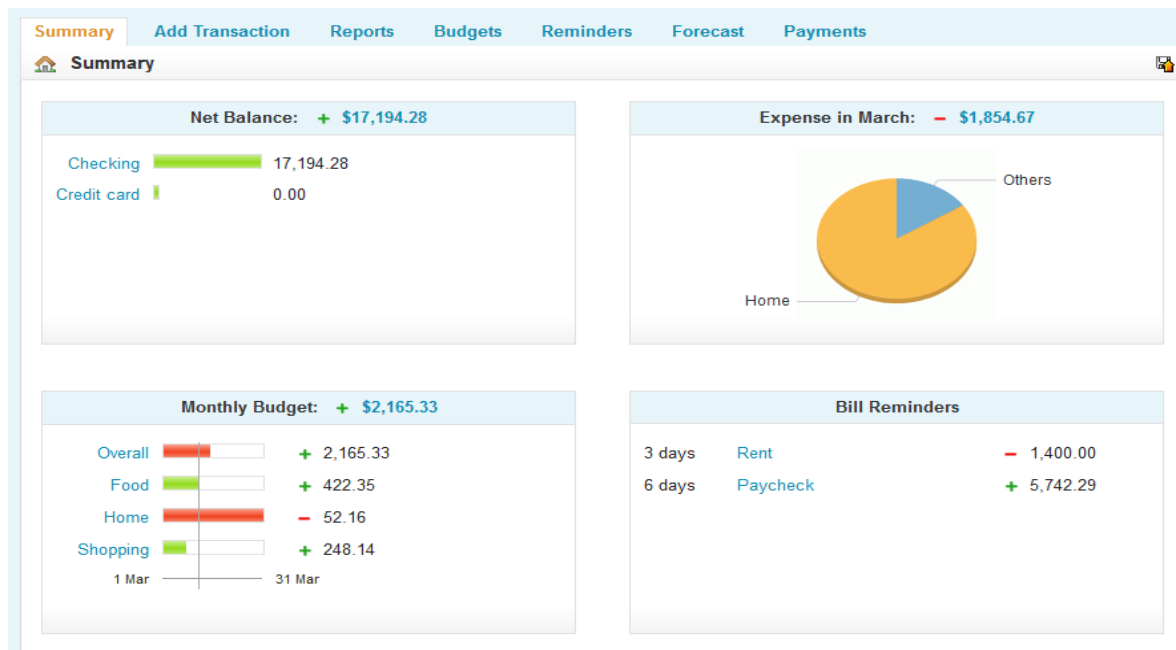


Figure 1 Buxfer Website Screenshot

Proposed Methods

From the previous survey we can conclude that, despite the many popular apps readily exist, none of them fully satisfy a user's need or visually well present the financial data. The ideal application will eliminate the unnecessary effort to switch among software applications and assist users by allowing them to better view and manage their expenses all in a one-stop-shop. We are here to present a prototype for such an application.

Data Storage. Due to the privacy of personal financial data, we will not be accessing user's bank account like other applications do. Instead, we simulate data (to be discussed later) in the following format:

User table:

User ID	Name
---------	------

Expense table:

Transaction ID	Description	Amount	Category	Date
----------------	-------------	--------	----------	------

Descriptions is in the format on a typical bank statement, such like: *walmart howell mill rd* and *bestbuy.com*. We will process the data in lower cases (easy in python and database).

ExpenseUser table:

Transaction ID	User ID	Paid	Expense Share
----------------	---------	------	---------------

Paid is the amount that the user paid on the transaction and *Expense Share* is the amount that the user should have paid for this transaction.

String Matching Based Transaction Categorization Algorithm. Our transaction categorization algorithm is based on string matching techniques. Firstly, we store the lists of top merchandiser names in each of the following categories: *Entertainment* (music, movies, games, books), *Food and Dining*, *Utilities*, *Shopping* (clothes, electronics), *Home*, *Transportation*, *Special* (tuition, tax) and *Miscellaneous*. For examples, *Entertainment* category contains stores such as Barnes & Noble and Regal Theater, while Lowe's and Home Depot are in the *Home* category. Then, we add keywords in each list: for instance, restaurant, bistro, dining, bistro, etc. for *Food and Dining*. Finally, given the descriptions, we compare the names along with keywords in the lists to the words in the descriptions. Edit Distance Difference is used for comparison. With categorized expenses, we can provide a visualization of the division of a user's expenses.

Outlier Detection. In order to detect outliers that need to be treated differently or identified to alert the user, we plan on integrating a simplistic outlier prediction algorithm based on the three-sigma rule of Normal Distribution. This rule states that for a normal distribution, nearly all values lie within 3 standard deviations of the mean. Specifically, about 68.27% of the values lie within 1 standard deviation of the mean; about 95.45% of the values lie within 2 standard

deviations of the mean; and nearly all (99.73%) of the values lie within 3 standard deviations of the mean. In mathematical notation, these facts can be shown as follows:

$$\begin{aligned}P(\mu - \sigma \leq X \leq \mu + \sigma) &\approx 0.6827 \\P(\mu - 2\sigma \leq X \leq \mu + 2\sigma) &\approx 0.9545 \\P(\mu - 3\sigma \leq X \leq \mu + 3\sigma) &\approx 0.9973\end{aligned}$$

Visualization & User Interface. This prototype means to visualize various forms of expenses. The following mockups display some features that could be implemented progressively.

Time-Based Views

The time-based views will show a visualization of a period of expenses using bar chart or line chart as shown in Figure 2.

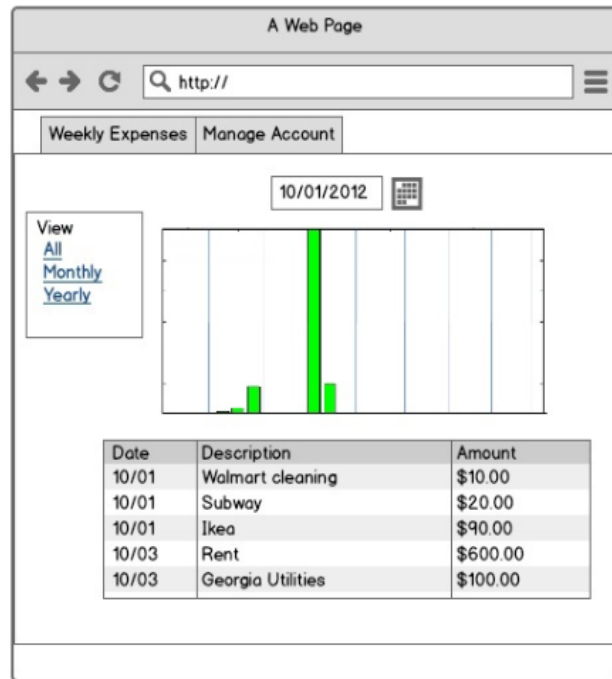


Figure 2. Time-based Views: Weekly View

Calendar View

Similar to weekly view, this feature provides a monthly view of expenses using a calendar as shown in Figure 3. A short description of expenses will be included in the day entry. Days will be colored coded indicating the amount of money that was spent that day. If no money was spent, the day will be left blank. Other days will be colored different shades of green from pale green to a darker color of green for smaller to larger expenses.

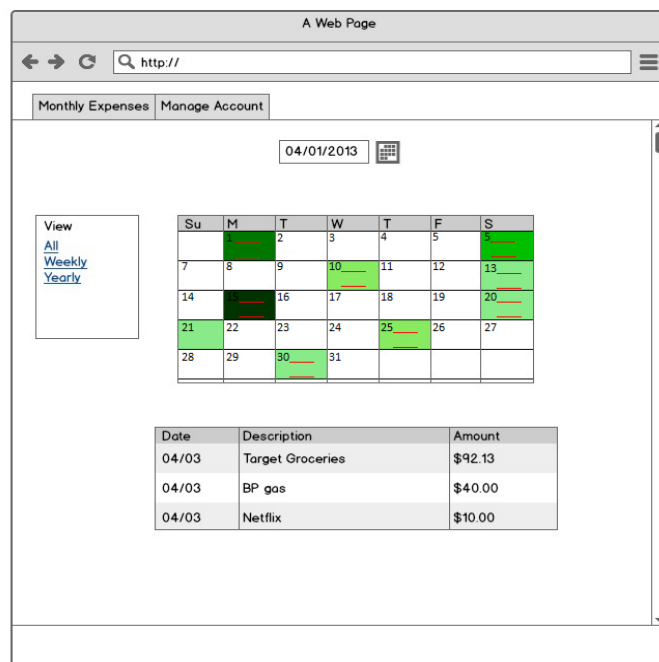
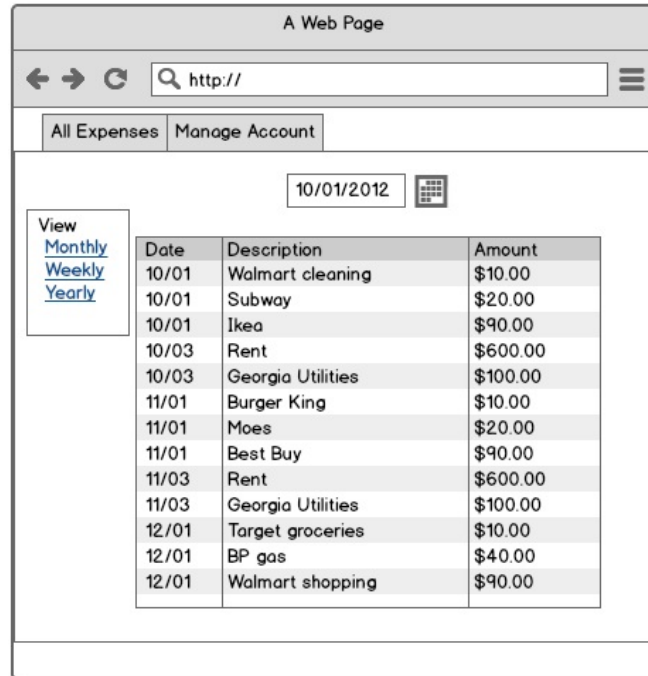


Figure 3. Calendar View: Monthly View

Filter View

This view will show all expenses and allow the user to sort and filter by date, description, category, and amount (Figure 4).



Date	Description	Amount
10/01	Walmart cleaning	\$10.00
10/01	Subway	\$20.00
10/01	Ikea	\$90.00
10/03	Rent	\$600.00
10/03	Georgia Utilities	\$100.00
11/01	Burger King	\$10.00
11/01	Moes	\$20.00
11/01	Best Buy	\$90.00
11/03	Rent	\$600.00
11/03	Georgia Utilities	\$100.00
12/01	Target groceries	\$10.00
12/01	BP gas	\$40.00
12/01	Walmart shopping	\$90.00

Figure 4. Filter View

Category-Based View

The category-based view will show the division of a user's expenses, where circles will be colored based on the categorization of the expense, and the size of the circle is based on the amount of the expense.

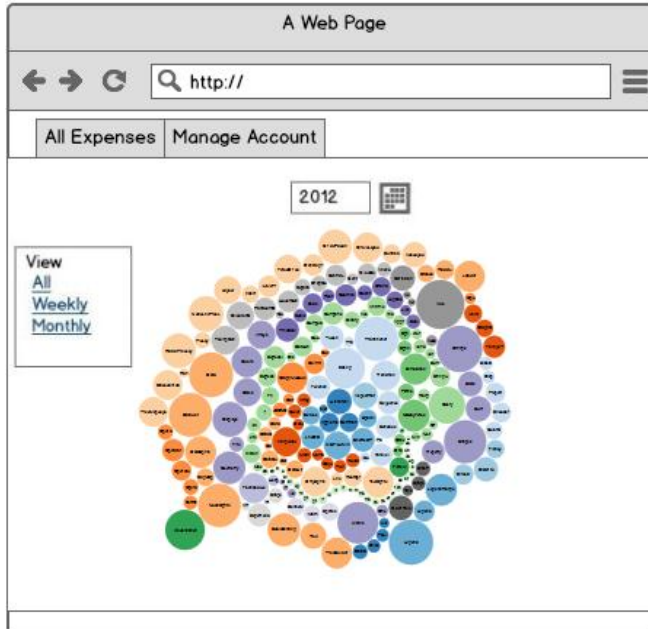
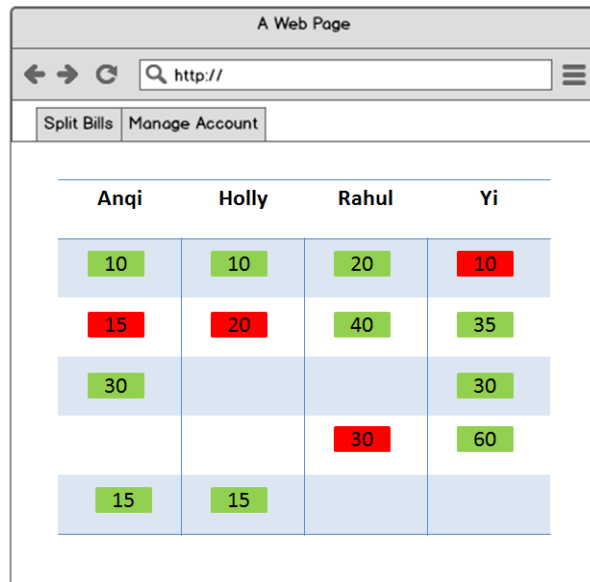


Figure 5. Category-Based View

Shared Expense View

This view shows split expenses between different users. Red blocks show expenses that are owed. Green blocks show expenses that have been paid. A user will be able to select other users to only see expenses shared with those chosen.



Anqi	Holly	Rahul	Yi
10	10	20	10
15	20	40	35
30			30
		30	60
15	15		

Figure 6. Shared Expense View

Tools. We plan on using a SQL-based database to store the data, use Python as the main backend implementation language, Twitter Bootstrap, jQuery, and JavaScript for frontend UI, and d3 as visualization tool.

Implementations & Evaluation

We use the software to categorize and visualize the expenses from simulated data (discussed later) in order to test the functionalities of this prototype. The actual implementations covers most of the algorithms proposed to categorize the expenses and provide different views mentioned above.

Data Generation. Due to the privacy sensitivity of bank account data, we decided to simulate expense data as input using Python. We simulated the expenses for several years within 8 different categories of expenses: entertainment, grocery, home, restaurants, shopping, special, transportation, and utilities. For each of these categories, well-known stores were used to create a directory. For each user, expenses were created for several years in each of these categories. For each category, a range is specified for the number of times per month a person will have expenses in that category. A number of expenses are randomly generated within that range. Another range is specified for the cost of expenses in a category as well, and new expenses have a cost that is within this range. For each user in the system and all the years and months, expenses are generated.

Split Expenses were also created. A decision is made to randomly split an expense with low probability since it is expected that most expenses will not be shared. After deciding to randomly split an expense, the category, date, and amount of the expense are decided. A number of users involved in the split are generated and the users are randomly selected from the current users in the system. A user involved in the split is assigned a fraction of the total expense. Some split expenses are evenly split while others are not. The assumption is that most expenses will be split evenly with a small probability being uneven. Also in split expenses, all users involved in the split may have paid their portion or there could be users that owe money.

Outlier Detection. Our implemented outlier prediction algorithm uses the 2 standard deviation range of the mean, assuming user expenses follow normal distribution on the long run. When the shared expense of a user lies in the range, we flag it as reasonable; otherwise questionable. We choose the 2-StdDev range because we want the percentage of expenses to which users should pay attention to be neither too small nor too large. 4.5% is large enough to include questionable expenses without alerting users on too many reasonable expenses. We believe that the outlier prediction method can serve as both an expense watcher and a fraud detector, in case that credit card information was stolen.

Table 2 Performance of Outlier Prediction

User	1	2	3	4
In Range Perc.	99.39%	95.29%	98.33%	95.31%

Table 2 shows the performance of this outlier prediction algorithm on the expense data. Based on over 1700 expenses of 4 users, the in range percentage of expenses is higher than 95% of all expenses.

Visualization & User Interface. We implemented most of the proposed forms. The forms and charts are implemented using d3, and the main framework is written in JavaScript and HTML.

Shared Expense View

The following screenshot shows the implemented Split Expenses where the red blocks are owed and the green ones are already paid.

Split Expenses					
January 2012					
Expense ID	Date	Rahul Agrawal	Yi Ding	Holly Williams	Anqi Zou
1	1/4/2012	23.75	16.81/23.75	23.75	30.87/23.75
877	1/15/2012	7.65/8.5	6.8	7.65	11.9/11.05
5	1/19/2012	16.63	14.25	14.25	49.87
1318	1/24/2012	86.4	-	105.6	-

February 2012					
Expense ID	Date	Rahul Agrawal	Yi Ding	Holly Williams	Anqi Zou
893	2/1/2012	20.75	20.75	20.75	20.75
1362	2/4/2012	11.25	11.25	11.25	11.25
31	2/6/2012	7.7	8.8	-	16.5
887	2/16/2012	14.7	18.9	21	29.4

Figure 7 Shared Expense View

The user can also view one individual's data in the format shown below.

User Expenses

EID	Date	Description	Category	Paid	Total Amount
1	1/3/2012	train	transportation	\$23.75	\$95.00
5	1/4/2012	Citgo	transportation	\$16.63	\$95.00
1319	1/1/2012	Sunoco	transportation	\$29.00	\$29.00
1320	1/0/2012	RaceTrac/Raceway	transportation	\$93.00	\$93.00
1321	1/5/2012	Kroger brand gasoline	transportation	\$79.00	\$79.00
1322	1/0/2012	ARCO	transportation	\$65.00	\$65.00
1323	1/1/2012	Southwest	transportation	\$40.00	\$40.00
1324	1/2/2012	AMC	entertainment	\$27.00	\$27.00
1325	1/3/2012	book off usa	entertainment	\$13.00	\$13.00
1326	1/2/2012	Lord & Taylor	shopping	\$28.00	\$28.00
1327	1/5/2012	Hollister	shopping	\$23.00	\$23.00
1328	1/1/2012	Boscov's	shopping	\$54.00	\$54.00
1329	1/3/2012	The Bon-Ton	shopping	\$172.00	\$172.00
1330	1/4/2012	verizon	utilities	\$298.00	\$298.00
1331	1/0/2012	Arctic Circle Restaurants	restaurants	\$18.00	\$18.00

Figure 8 Individual Expenses

Time-based Views

Figure 10 shows a screenshot for a weekly view where it displays daily expenses throughout the week. Figure 11 illustrates the monthly expenses within a year. These views are implemented in JavaScript and HTML with the charts generated by d3.

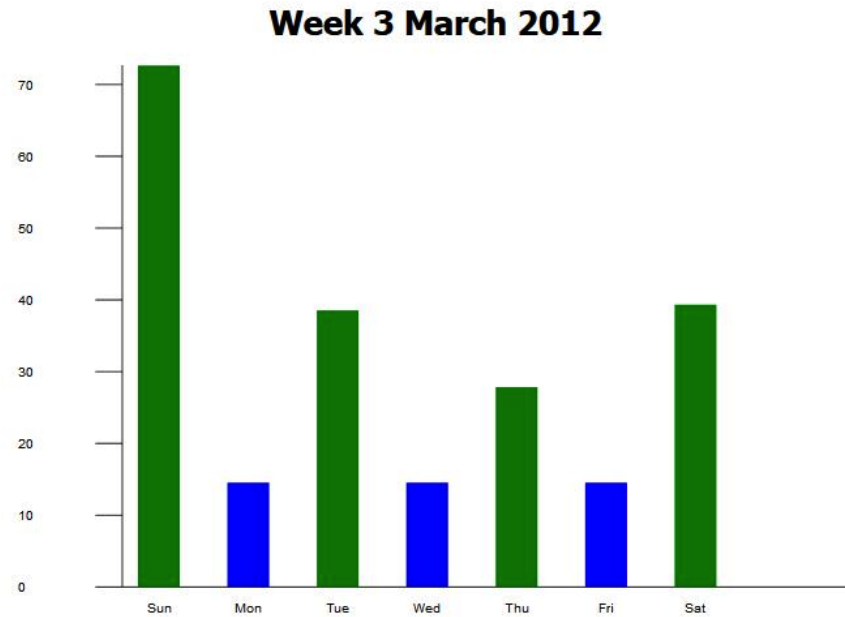


Figure 9 Time-based View: Weekly View

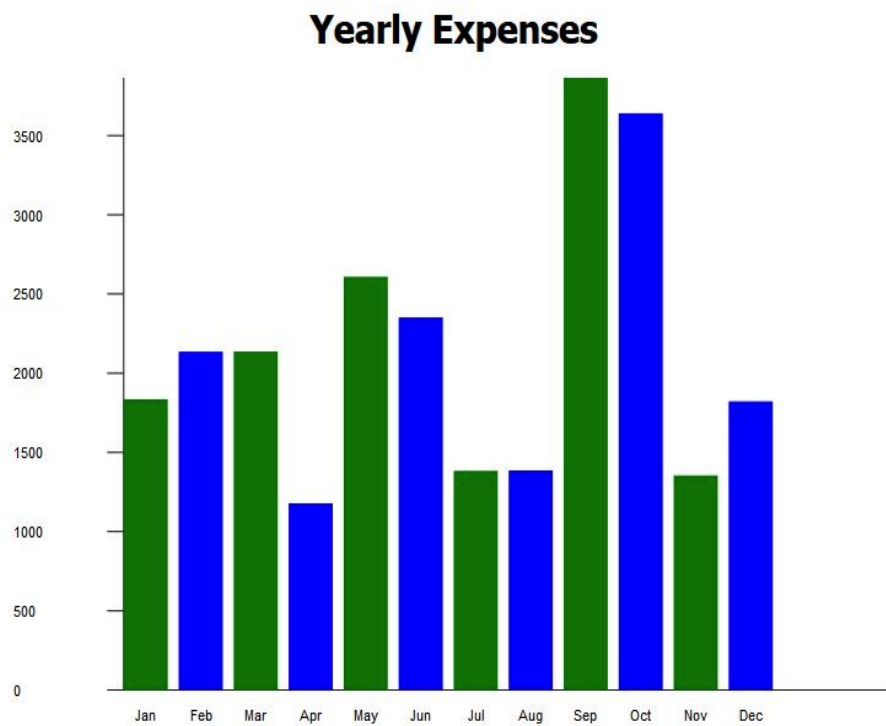


Figure 10 Time-based View: Yearly Expenses

Calendar View

Figure 13 shows a snapshot of the calendar-based visualization for expenses. This view enables the users with the following functionalities:

1. Visualizing the expenses per-day, including the total money spent on each day;
2. Differentiating shared and personal expenses by using different colors for each type;
3. Categorizing each of the expenses, using category icons for each of the expenses;
4. Clicking description pane to see the detailed information on each expense.

The calendar view is built using Twitter Bootstrap framework and uses popular JavaScript libraries including d3, jQuery, slimScroll. We also modified an existing calendar view code from github, to customize for our needs. The color and icon choices were based on intuition, but a more scientific approach could be taken to improve the visual appeal and ease of information. The view is designed for screen resolutions of 1600x900 and above. A higher resolution was needed to tackle the amount of information that is possible with the view. To restrict large number of expenses per day, we added scrolling feature within each cell of the calendar. The scroll bar becomes active if there are more than 4 expenses per day.



Figure 11 Calendar-based visualization of Expenses

Discussion& Future Work

This project enables the team members to experiment different tools used for data analysis and visualization during a software engineering process. The following works can be taken into account in future study and experiments:

- Collect real data. Data from actual bank statements can be used to generate realistic analysis and to improve the algorithms in order to accommodate unexpected situations.
- Train algorithms with real data. Real data can be used to better support the categorization algorithm.
- Implement and integrate all views. Though we have implemented most of the views proposed, the category-based view and filter view were not successfully carried out. The currently views can potentially be integrated into a single platform or taken to a web application.
- Get user feedback to improve functionalities.

Related Work

Applications

<http://www.apartmenttherapy.com/manage-personal-finances-with-these-online-tools-174639>

<https://www.trackeverycoin.com/>

<http://www.walletmap.com/>

mint.com

splitwise.com

buxfer.com

billmonk.com

<http://lifehacker.com/5584273/five-best-personal-money-management-sites>

expensure.com

<http://www.youneedabudget.com/features>

<https://www.clearcheckbook.com>

Papers

<http://www.cc.gatech.edu/~john.stasko/papers/infovis07-casual.pdf>

<http://dl.acm.org.prx.library.gatech.edu/citation.cfm?id=1753409&bnc=1>

<http://www.personalinformatics.org/chi2010/>