COMP-SCI 5551 (FS16) - Advanced Software Engineering
Project Team 1 - The Brokers
Dayu Wang (59); Chen Wang (58); Yunlong Liu (25); Joshua Neustrom (39)

Project Increment 1 - Report

(Due Sep 23rd, 2016)

- **Introduction**

The fall semester is well underway. Millions of college freshmen headed off to school last month. Managing money as a college student is critical. It's likely a priority for every student, unless you recently won the lottery. As we can remember, the first thing before we come to the college is shopping for the daily necessities, food, gathering textbooks, and packing up entire room. To most of these freshmen, it's the first time live to independently in their lives, and paying more attention to their personal finance. The average college student graduates with $24,000 in debt, according to the Project on Student Debt[1]. Most of the students will find on-campus or off-campus job to keep their financial independence, while some students still relying on their family. Most of the college students may roll their eyes at making a budget, they know the significances of managing money to college experience, but they don't know how to start and cultivating these good habits, they are paying their own way; receiving parents' help or get budget problems. Even there are several tips for these freshmen, create a budget, start from the bank statement; spend on the right things, take the time to impress the importance of using debt wisely; Look for student discounts, college students should become masters at exploring the ways how their educational status can see the money; avoid full-price textbook, as we know college textbook is very expensive, so it's a good idea to borrow from library or buy a used one; Set financial limits, it sounds like a most simple way to avoid over budget, but it's really hard to figure out how to keep the balance within certain limit.

- **Project Goal and Objectives** *(revised from the project proposal)*

1. Overall Goal

Base on the situation above, it's urgent to find a solution. And when we investigate and research all kinds of budget managing application, we have not found any application that targeted to college students, and that's the reason we have this idea, make an application for the college student, and help this group of people who need this most. Our application is called *Pocket Manager*, take care of the expense habit of college students, and help them to have the habit of recording every single expense, and review that, then have a clear spending plan in their mind. And at the very beginning, our application will provide these solutions for the users, we will have these system features: login system, budget creation, record a payment, optical input, ask for an item, get statistical result, and graphical representation.

Unlike other similar software which just help user understand the budget and check the bill seems mediocre.*Pocket Manger* try to help users to have a good expense habit, and make this process smooth which let user accept this easier.

2. Specific Objectives

Based on our summarization and research, there are four most common problems that may trouble college students (see Figure 1).

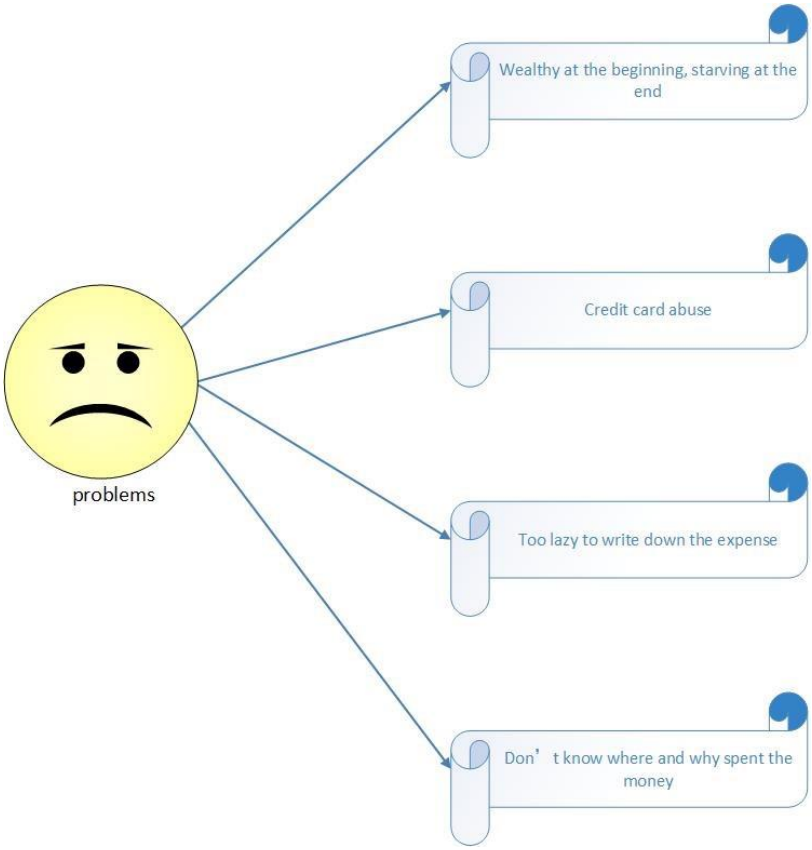

Figure 1. Expense problems that frequently occur amongst college students.

Problem 1 - So wealthy at the beginning of a month and so starving at the end of the month.

Many students may find that, every beginning of the month, when they get the salary from their on-campus or off-campus jobs, they are kind of wealthy, and spend half of the money in one week by eating in the restaurants every meal, shopping mindlessly, etc. and then find they become very pool, and live off balance. And starving at the end of the month, unfortunately, at the beginning of next month, repeat this situation, they may realize this, but don’t know how to stop this circulation.

Problem 2 - Credit card abuse always occurs.

It’s not really about the Credit Card fraud, it’s more likely a kind of over budget, serious overdraft, three-quarters of college students have at least one credit card and nearly half have four or more, according to a research by Take Charge America Institute[2]. It's really important for students to get their debts under control before they graduate. This situation not only occurs among college freshmen, based on this research, senior student have more risky business. It’s really important to think carefully every time before you slide the credit card (see Figure 2).

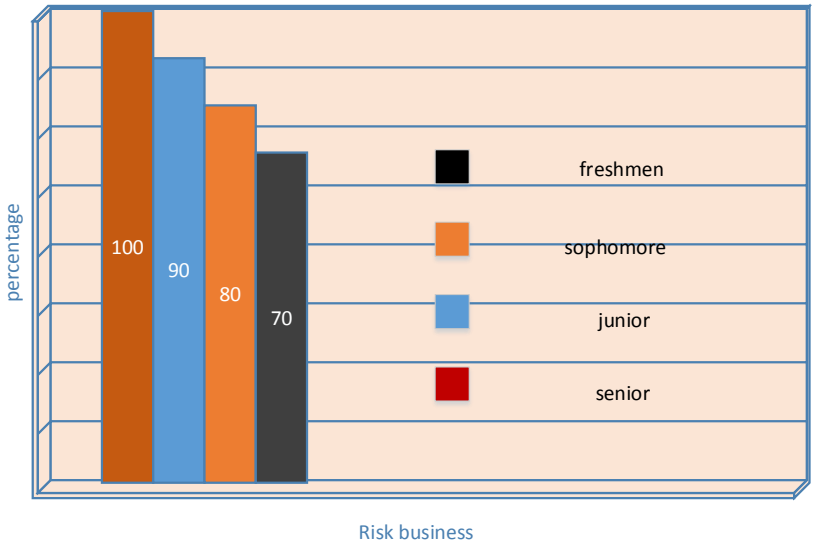

Figure 2. Risk business percentage relates to college students.

Problem 3 - Relying on credit card bills as expense record and so lazy to write down every expense.

Every time you use credit card, the bank will have a record, at the end of the month, the bank will send you a bank statement, every single expense will be on that statement, but please remember, not every expense was done with credit card, what about the payment with cash? You may won't remember how and when you spent that amount money, and another thing is, the bank bill will be sent to you at the end of the month, you will know nothing about this before that, so write down the detail of expense every will remind you what you have done this day, you will get a plan in your mind, how much money i have now, how much money I've spent. And realize that, what kind of things I shouldn't consider buying.

Problem 4 - "I kept being frugal, but why and where did I spent such a large amount of money?"

This problem seems very common, we make payment so many times a month, it's hard to have this habit, write down everything you've bought, write down every detail of the expense, so, that may cause us forget what we've bought, the students may don't care about certain single payment, but within a month, too many that kind of expense may be a large amount of money, but they may even don't realize, where, when or why they spent that number of money, that all because they don't have a review.

Since we have summarize these serious problems, let us talk about how solve these case, here are some solutions we can provide in our application.

Solution 1 - Create a budget.

Based on the monthly income of a user (college student), our system helps him/her to create a reasonable monthly budget. For example, the system tells the student how much you can spend on food, entertainment, exercises, and so on.

Solution 2 - Separate want from needs.

Several rules are generated based on normally how people understand what is a college student's need and what is a want. The rule differs from every major field for a student.

Solution 3 - Character/number recognition API applied.

Character/number recognition API will be applied in our system, which definitely facilitates the input of their expense by just taking a picture of their payment receipt.

Solution 4 - "Grading" of your last month performance in money management.

Statistical data will send to the user monthly to analysis his/her performance in expenses. Also, when accumulated expense in a certain aspect is approaching the budget, warning messages will be also delivered to the user.

3. Specific Features (see Figure 3)

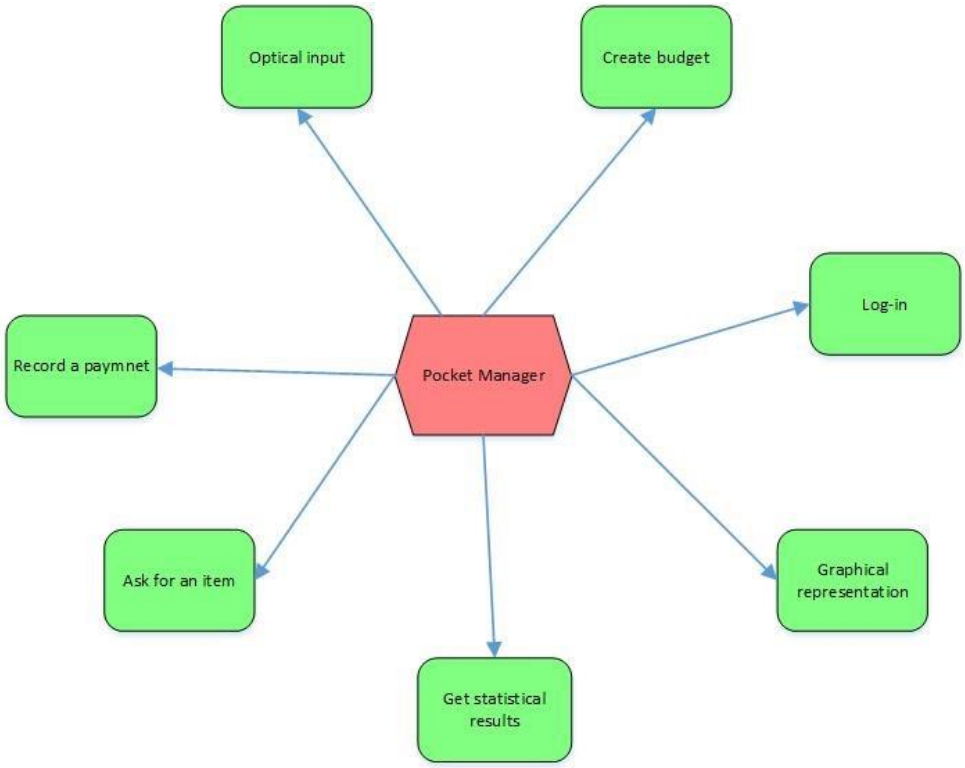

Figure 3. Features involved in the *Pocket Manager* system.

Feature 1 - Log-in system

This application will provide different functions and solutions for the users in different situations, so the log-in system is necessary, user may login with unique user's name or email address.

Feature 2 - Budget creation

This features existing for collecting all the user's income information, since our application focus on students, so we will probably have three suggestive monthly budgets which are depend on the income levels, because we want to design personal budget plan for every user, and base on our users are supposed to be college students. Three different budgets are created and may be the feasible levels of the budgets. This is called Budget Creation. But this part of income should be manageable income which is excluding rent, loan, etc. Because usually rent and loan are unchangeable or too stable to be managed.

Feature 3 - Optical input

This is a very characteristic feature; we will achieve the function with a few relative API, such as character or number recognition API. Users can simply scan or take picture of the bar codes or receipt number, it's very convenient for user record the payment, as we all know, and it's too hard to remember the prices of everything we've bought. And it's better than type the expense details manually.

Feature 4 - Record a payment

Though we will have the optional input feature, but it is still necessary to record some specific payment manually, as we know, sometimes we can't have the picture of the receipts, or even we don't have the receipts. And in this part we will classify the expenses into several categories, such as food, traffic, book, entertainment, etc. Users want to know more than a bank statement, they need to know how and why the spent the money.

Feature 5 - Ask for an item

This feature is more like a wish list, as the users may have items that they want to buy but may cause over budget. Based on the prices of these items and the monthly income, this feature will decide whether the item is necessary or not, this is called separate needs from wants, what is more, it will generate a time to suggest when the user can consider buying it.

Feature 6 - Get statistical results

This feature allows users check the statistical expense result of this monthly or previous month, absolutely, based on several categories, users always want more than a bank statement, and it is better to show the expenses in a form or separated pages than just a single list.

Feature 7 - Graphical Representation

This feature applies the graphical API, such as high charts, it will show the statistical result graphically, sector diagrams is a good choice to show the expense of every aspect, and also the percentage.

- **Project Plan**

The plan of developing the system of *Pocket Manager* requires us to estimate the difficulty of the implementation of the features mentioned in the project proposal prior to creating a realistic plan of development. Also, since we are just located at the very first stage of the system development at this moment, we further classified each feature as *compulsory* and *optional*, where compulsory feature means that the feature is fundamental to our entire system (e.g. the communication with Mongo DB in that a user's expense history is always securely stored and accessed) or will mark our system distinguishable from other similar systems (e.g. the *lazy recording* feature that allows the user to input an expense of him/her by just taking a picture of the payment receipt), and the optional feature means that the feature includes time-consuming or challenging techniques for us to accomplish under the several-week intense time pressure as a class project. For example, the *expense predicting* feature, which requires a complicated *machine learning* model in our system designation, is categorized as an optional feature to us, though it is a very good idea to remind the user of any bad expense habit detected by the machine learning component before the user actually suffers from such ill habit. Definitely, a world-class nice Android application requires a huge team of even 100 people to work together for from several months to even several years. For a small group like us, we would like our developed system to be rather a "completed" system, than a "cutting-edge" system but lack of integration. Therefore, if time allows, we would be happy to develop those optional features to make our final project more beautiful; but if we are falling behind with the original schedule, then we have to focus ourselves on the compulsory features to accomplish the project on time. The classification of our planned features in our final system is categorized in Table 4.

Table 4. Classification of each planned features to be developed in the *Pocket Manager* system based on the estimated difficulty of each feature. The estimated difficulty values vary from 1 (easy) to 5 (very hard) and were generated by our group discussion and/or experienced class teaching assistant.

System Feature	Estimated Difficulty	Compulsory/Optional	Note
Login System	2	Compulsory	Regular
Social Login System	3	Optional	Google, Facebook, etc.
Expense History	2	Compulsory	Communicate with Mongo DB
Expense Recording (by text)	1	Compulsory	User manually enters the amount.
Expense Recording (by receipt photo)	3	Compulsory	User takes a picture of the receipt.
Algorithm of Creating Budgets	3	Compulsory	Create 3 different budgets to choose.
Algorithm of Separating Wants from Needs	4	Compulsory	Avoid *credit card abuse*.
Statistics in previous expense behavior	1	Compulsory	Mathematical models
Expense Supervising Monitor	3	Compulsory	Give warnings when the user is approaching a certain budget.
Graphical Demonstration	3	Compulsory	Using Highcharts
Expense Prediction	5	Optional	Machine Learning Model

Another factor considered in generating our project plan is the *dependency* amongst all the desired features to be developed. Obviously, some features need to be implemented after some other features completely implemented. Therefore, when we are coming up to a project plan, a consideration of the sequence of features to be developed matters a lot in the working efficiency. This dependency is different from the streaming architecture or the workflow diagram of the system. Workflow is a kind of streaming diagram, but for implementation and unit testing, several components can be implemented and unit-tested in parallel. Also, when a feature is half accomplished implementation, another feature may be already good to start coding. Therefore, figuring out a "fine" dependency will help us increase our working efficiency and save us a lot of time (especially "waiting others" time).

Therefore, we would like out project plan to be finely structured before we put a realistic and efficient plan in a ZenHub. Figure 5 is a graphical representation of how the entire project can be divided into smaller pieces. In Figure 5, each horizontal level represents tasks that can be worked on in parallel. When the regular login system has been built up, then the next level can be started working on without the accomplishment of the social login system. Similarly, receipt scanning input can be somehow "independent" since the downstream components can be implemented when the text receipt input has been implemented. The wants separation feature is isolated from the rest part of the system, since it is targeted to a rectification of bad money habit amongst college students, and the budget creation feature embraces the related algorithm design. The advantage of this diagram is that if our plan changes in future, a new feature is added or a current feature is removed, then by appropriately placing the new feature into the diagram, we can visually and easily understand how our project plan can be changed.

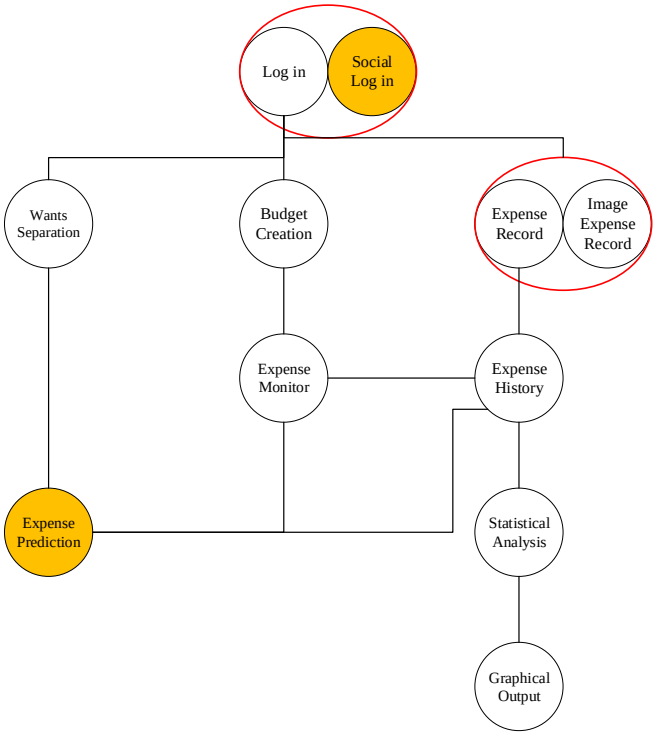

Figure 5. Analyzed dependencies amongst different features to be developed in the *Pocket Manager* system. The orange circles represent optional features and the others are compulsory features. The oval elements means that there are "sub-features" in this big feature and the sub-features can be accomplished by different people in different time.