COMP4920 Project Plan

# Personal Budgeting App

# Group Details:

**Tutorial**: M12A

|  |  |
| --- | --- |
| **Name** | **zID** |
| Albert Kurniawan Kuswandi | z5083849 |
| Shan Wang | z5119666 |
| Jeremy Kin-Yu Ng | z5074986 |
| Brandon Duong | z5112732 |
| Huan Pua Pao | z5055177 |

### Project Description

As students who are soon going to join the workforce, we feel it is necessary that we have a method to track our spending and become more money-wise. This project is aimed at building a system where it is easy and intuitive to track the user’s spendings and help the user to understand where their money is going so they can make the necessary adjustments.

Key functionalities:

* Record and categorise transactions as they occur
* Visualise your activity to understand how you are spending your money
* Help you budget by setting spending limits and savings targets
* Get reminders for important dates to stay on top of your bills
* Receive basic financial and budgeting advice based on your activity
* Easy access via a web application

Possible extensions:

* Have a working mobile app for Android that matches the functionality of the web app
* Use Optical Character Recognition to automatically extract transaction details from common receipts (Supermarkets, Gas stations)
* Receive prompts to input possible transactions based on time spent in a location

### 

### User Stories

User stories are grouped into the following epics:

* MVP - for planning purposes, rather than grouping functionality
* Transaction Logging - systems for the recording and viewing of transactions
* Budgeting - organising transactions, setting schedules and reminders
* Financial Advice - receiving basic budgeting recommendations
* Stretch Goals - bonus tasks/expansions unlikely to be finished in given timeframe

*See PivotalTracker for information on individual stories (Short titles used to easily identify items on the board, see the descriptions for full details)*

### Product Backlog

The key items with highest priority in our MVP:

* Designing a robust database that will cover all our needs.
* Setting up proper framework that allows back end to talk with front end properly.
* Having an intuitive UI for users to interact with.
* Storing profiles that allow users to record,categorise and view transactions.

The remaining stories are sorted in a logical order relating to which functionalities may be prerequisites for others.

Possible extensions, such as mobile apps and OCR, are left to the end and tackled only if time allows.

*See PivotalTracker for further details*

### Estimation of Work

Once the initial stories were placed in the icebox, we conducted a round of “Planning Poker”, in which each team member suggested the number of points that should be assigned to each story. Fibonacci increments were used, with each point roughly equal to 1 “day” of work, with the maximum being a value of 8, which is a story expected to take up an entire sprint (and avoided - large stories should be split into smaller ones. The only 8 point stories are the “stretch goals” which would be more thoroughly researched and decomposed if time allows for their development).

Stories similar to tasks with which certain team members had experience were used as baselines for our estimations, but there were some stories with unknown difficulty.

Estimations should be reviewed as we get a better understanding of our team velocity, and the complexities of adding functionalities to the project.

### Release Plan

The target for completion of the MVP is for the sprint ending on Wednesday week 10.

Following this, there will be an iterative release at the end of each sprint, including new functionality developed during those weeks.

### Sprint Plan (Initial Sprint)

|  |  |
| --- | --- |
| **Story/Task** | **Developer(s)** |
| Database Design and Setup | Jeremy, Albert |
| Basic Framework Setup | Brandon |
| Backend Setup | Pua |
| UI Design | Shan |

Allocations for the initial sprint were made based off each team members’ past experiences, in order to get the groundwork completed as quickly as possible, such that functionality could be rapidly added over the following sprints.

Tasks for the first week are very technically focused, given that stories needed to be split into work manageable during a one week sprint. After these initial key steps, the remaining stories are more similar to the “traditional” agile user stories, adding functionality to the product that achieves clearly defined needs of the end user.

### Allocation of Roles

The product owner will be Albert, as he has the most experience with budgeting app prior to the project, hence he should be able to discern which features are essential to have a functioning app.

The role of scrum master will be rotated throughout the team for each sprint, to give all an opportunity to understand the role and help to keep the team on track. As the convener of the initial planning meeting, Jeremy will be the scrum master for the first sprint.

### Meeting Schedule

Each sprint iteration will be one week in length, with our enrolled meeting timeslot (Wednesday 11am) used to conduct a face to face meeting each week. This would include both the sprint review and retrospective for the most recent sprint, immediately followed by planning for the following sprint.

Given the busy schedules of group members, it is not expected that progress will be made on the project everyday. In addition to that, it was also difficult to choose specific days of the week for standup where every member would be available and have items to discuss. As such, we decided to have daily standups online, but with the recognition that each member would have a few days each week with no progress to report.

### Extreme Programming Practices

* Planning game - will be used to direct our weekly sprint planning meetings. The release planning phase allows us to revise the backlog and prioritise stories for the coming sprint, as well as modifying or adding new stories if required. The iteration planning phase facilitates discussion for how stories should be tackled, and then assigns tasks to individual developers.
* Small releases - whenever concrete progress is made, a test release should be created in order to check functionality incrementally, rather than only building at the end of a sprint where compatibility between new elements may create issues.
* Coding Standard - we will apply a common coding style across all code files to help with readability, and ensure collective code ownership. We have decided to apply the following python standard: <https://www.python.org/dev/peps/pep-0008/>
* Collective code ownership - every member of the team should understand all of the code in the project, and be able make changes to any section if required. We will ensure that this occurs by assigning varied stories to developers as the project progresses, so that each will have experience interacting with all of the different modules.