CS2103 Project Proposal

group: cs2103jan13-t13-4j

LIU BOHUA	A0091879J
LIN ZHIYUAN	A0091859M
PANG KANG WEI, JOSHUA	A0087809M
WONG JING PING	A0086581W
HAO WEI	U080159R

Context

In this project, our targeted users are working adults. With the busy schedule that working adults have nowadays, coupled with the profusion of financial transactions that their lives revolve upon, it is often difficult to keep track clearly of every debit or credit entry for one's finances. Moreover, with the abundance of details that users often have to swim through to find out what they find, users can be put off by a software that merely serves as a secondary memory storage for their assorted transaction. As such our group has decided to develop a simple software that is specially tailored to address the above pain points, and in the process, make expense tracking a meaningful and delightful experience.

Problem

1 Limited Time and Energy

In general, our programme is designed to provide an efficient personal finance tool for the working adults. Since the working adults may not have enough time and energy to track their daily financial activities, they need a simple and user-friendly tool that can help them record their daily expenses and incomes. In other words, our programme is very much like a housekeeping book to address the recording issues of their income and expense.

2 Potential to Forget Bills

In addition, the user may be occasionally prone to forgetting bill payments. Given the many sources of bills and IOUs that the average working adult has these days, keeping track of these mentally will no doubt result in some errors due to one's forgetfulness. In some cases, such as late credit card payments allow banks to impose additional payment fees, having a bill reminder will help the user not only pay one's bills on time, but also avoid such unnecessary charges.

All in all, we hope to target these 2 main issues faced by the average working adult in our software through an intuitive and user-friendly graphical interface, coupled with a bill reminder feature.

Project Scope

Our software will be targeted at working adults, with **1 user** in mind. We have categorised our features according to their priority, with the highest priority features in the Baseline Features, and the additional features under Good-to-have Features.

Baseline Features

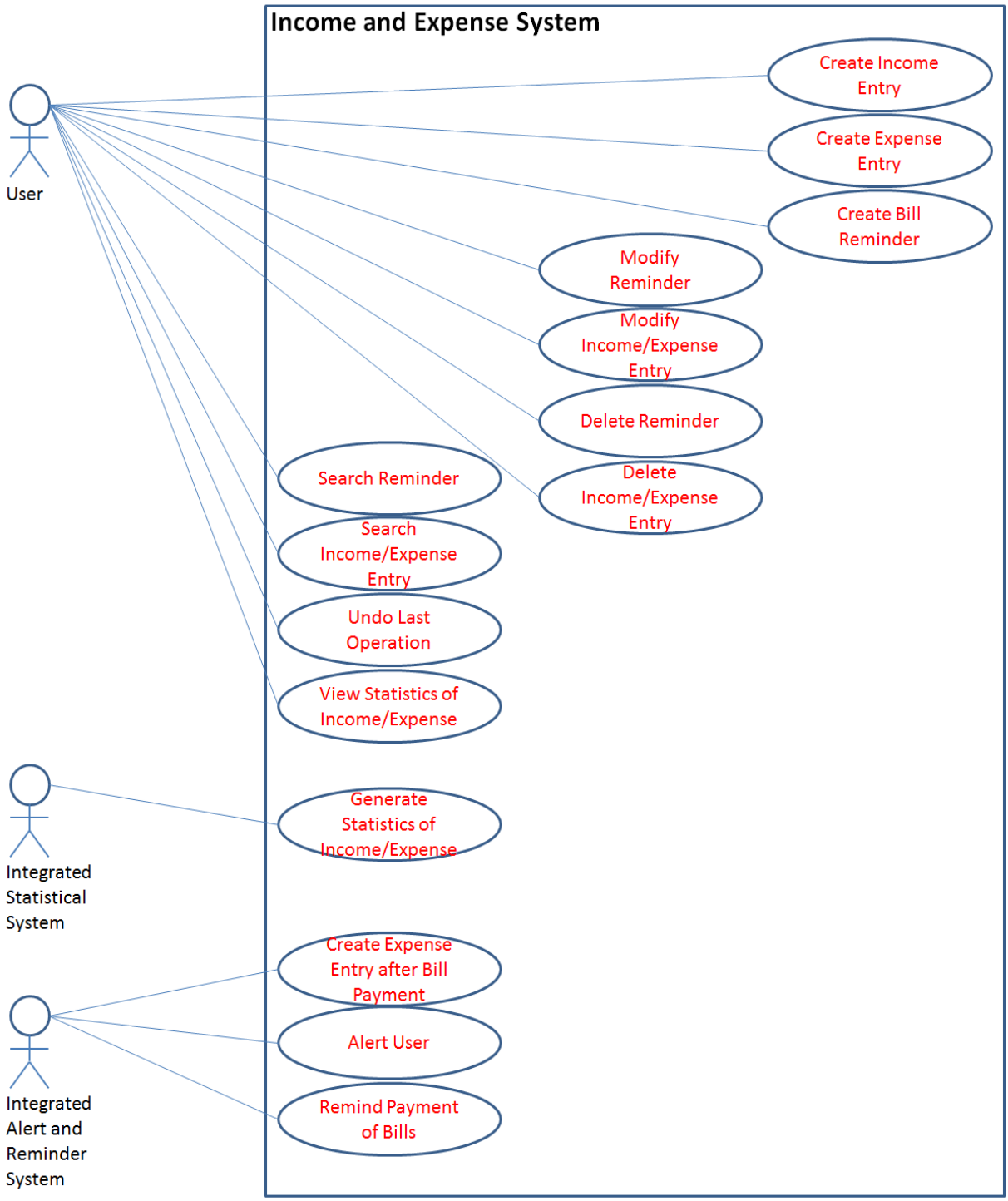
- Tabbed views
    - for separating the details of each transaction from the ‘big-picture’ and analyses
- CRUD operations for expense, income and reminder entries
- Display of expense, income by categories
- Simple search
- Undo

Good-to-have Features

- Presenting an overall view of one’s finances at a glance through
    - bar and time series charts (using 3rd party APIs)
    - a colour-coded, clutter-free GUI
- Alerts from reminders on the dashboard
- Creating an expense entry after paying a bill (automatically)

Recurring expenses and income entries

Use-Case Diagram



Use-Case Descriptions

User Functions

Create Income Entry: To create an income entry for the user and stores to income.xml (Required parameters: ID, date, description, category, amount)

Create Expense Entry: To create an expense entry for the user and stores to expense.xml (Required parameters: ID, date, description, category, amount)

Create Bill Reminder: To create reminder for bill payments and stores to reminders.xml (Required parameters: ID, year, month, day, description, amount, payee)

Modify Bill Reminder: To update a reminder entry (Possible required parameters: ID)

Modify Income/Expense Entry: To update an income entry or expense entry (Possible required parameters: ID)

Delete Income/Expense Entry: To delete an income entry or expense entry (Possible required parameters: ID)

Search Reminder: To search for a particular reminder entry (Possible required parameters: ID || year || month || day || description || category || amount)

Search Income/Expense Entry: To search for a particular income entry or expense entry (Possible required parameters: ID || year || month || day || description || category || amount)

Undo Last Operation: To undo the latest logged entry which is stored in log.xml

View Statistics of Income/Expense: To view statistical charts and tables of entered income and expenses from income.xml and expense.xml

Integrated Statistical System Functions

Generate Statistics of Income/Expense: To generate the statistical charts and tables of entered income and expenses from income.xml and expense.xml

Integrated Alert and Reminder System Functions

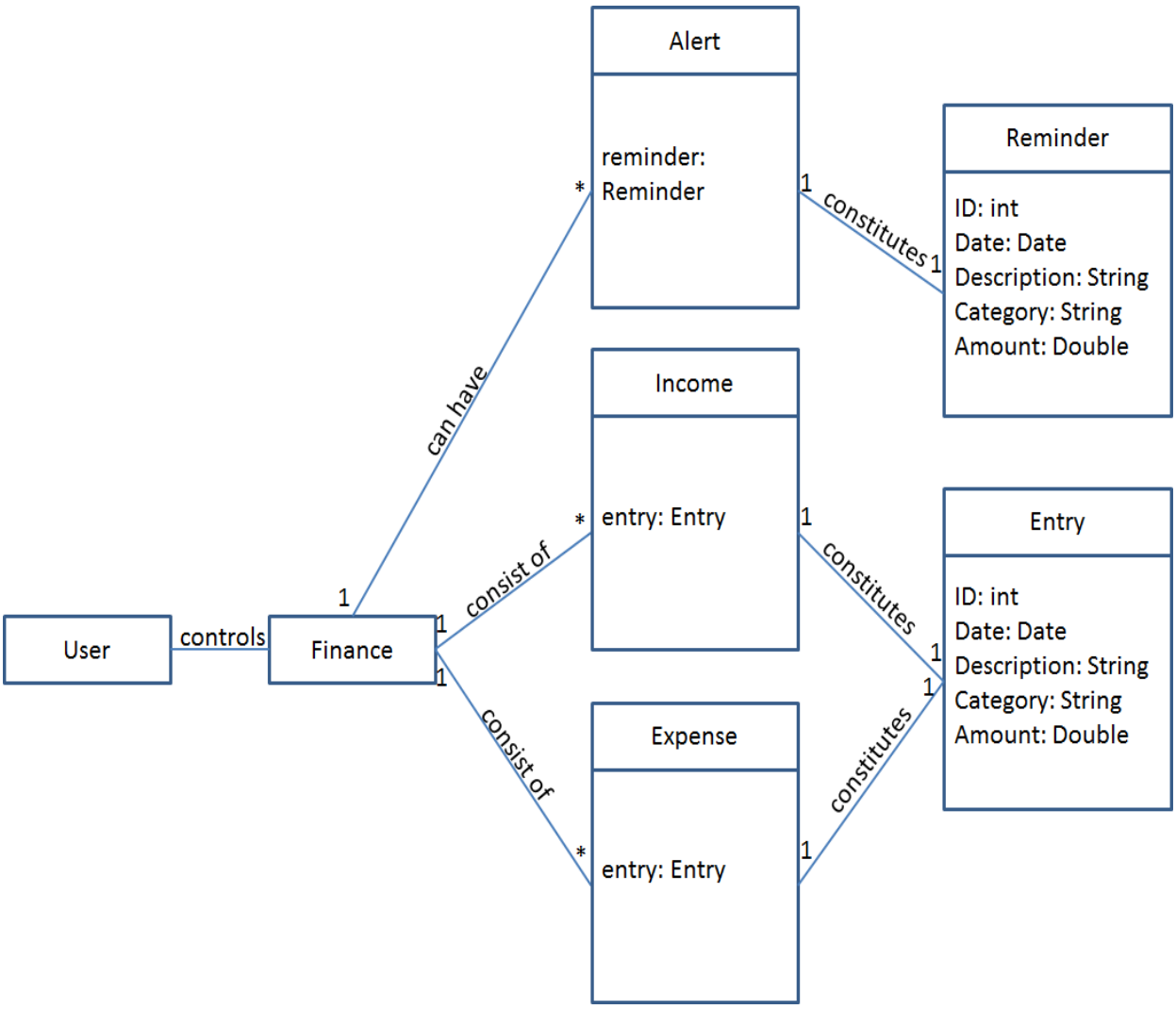
Create Expense Entry after Bill Payment: To automatically create an expense entry after confirming payment of a bill

Alert User: To alert the user in the following scenarios

1. Expenses are greater than income
2. Alert user when payment of a bill is reaching 2 days before due date, or even past its due date

Remind Payment of Bills: Remind user payment of bills from entry date to 3 days before due date

Domain Diagram



GUI

The following figure shows the initial display, also known as the dashboard.
Navigation between will be done through the 4 tabs (Dashboard, Expense, Income, Reminders), with the active view shown in bold.
Buttons for user functions are located at the bottom (Add, Update, Delete, Undo, Search), with usable functions shown in black and non-usable functions shown in grey.

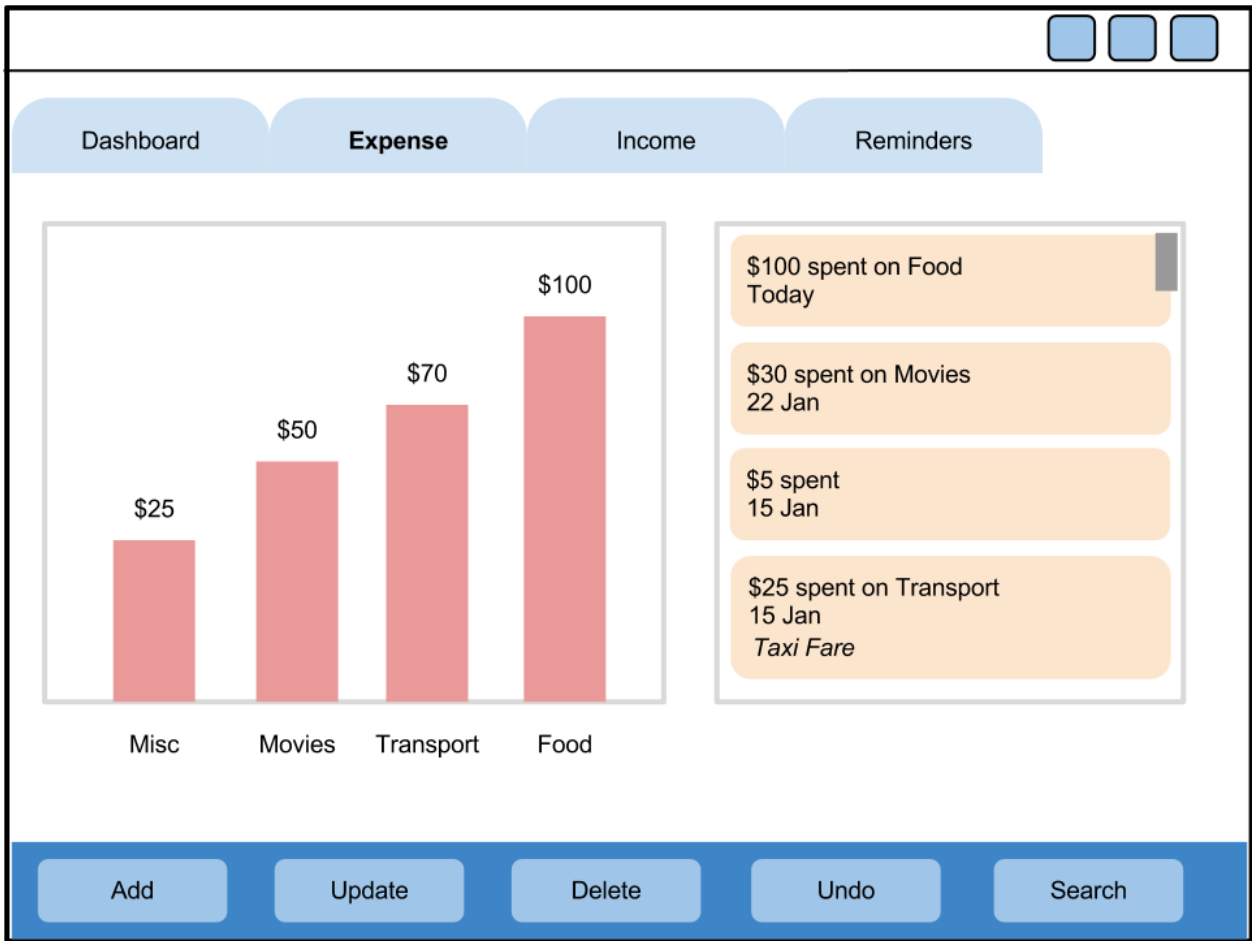
Dashboard View

On the dashboard, the user will get to see his total income and expenses, a time series chart of both his income and expenses, with buttons allowing the user to view a variable time horizon. In addition, there will be call-out with bill reminders to the user. Within the callout, there are 2 buttons, ‘Paid Up!’ and ‘Forget it...’. Upon clicking ‘Paid Up!’, an expense entry is automatically created based on the reminder’s information, and the charts automatically updates. On the other

hand, clicking on ‘Forget it...’ will ‘snooze’ the reminder for the rest of the day. The callout disappears after clicking either of the buttons.If the user selects the Expense tab, the view changes to the expense view.
Here the user can see a chart of his expenses by categories, as well as a list of his expenses on the right. The list of records will be housed in a frame that will be scrollable.

Expense View

For expenses made on the same day, ‘Today’ will be displayed instead on the date, as it is more intuitive. If the expense entry was made in the same year, then the year is omitted to reduce clutter. Description of the expense (if any) will be shown below the date in italics.



At the Expense tab, the user is able to add, update, or delete the expense entries.
To add, the user only needs to click ‘Add’, and a pop-up will appear, prompting the user for information.
To update, the user may click on the relevant entry in the frame, then click update. Alternatively, the user may double-click on the relevant entry directly to bring up the pop-up.
To delete, the user may click on the relevant entry in the frame, then click delete. This will bring up a confirmation pop-up.

Pop-up Window

When a pop-up window is activated (for add, update, search), the main window is inactive(not usable). Clicking anywhere on the main window will only direct the user to the pop-up window.

When the user selects ‘Add’, a new entry object with blank fields will be created, and the pop-up window will get the information from the user. When inputting, amount and date are compulsory fields, while description and category are not. If the expense is a recurring one, the user can enter the frequency of the expense (for example, 7 days) so as to reduce the effort in noting recurring expenses. When the optional fields are left unfilled, the expense is assumed to be once-off, description will be left blank, while the expense will automatically be categorised under ‘miscellaneous’.

If ‘Update’ was selected, the fields will be populated with the relevant information from the referenced entry. The user can proceed to edit the respective fields.

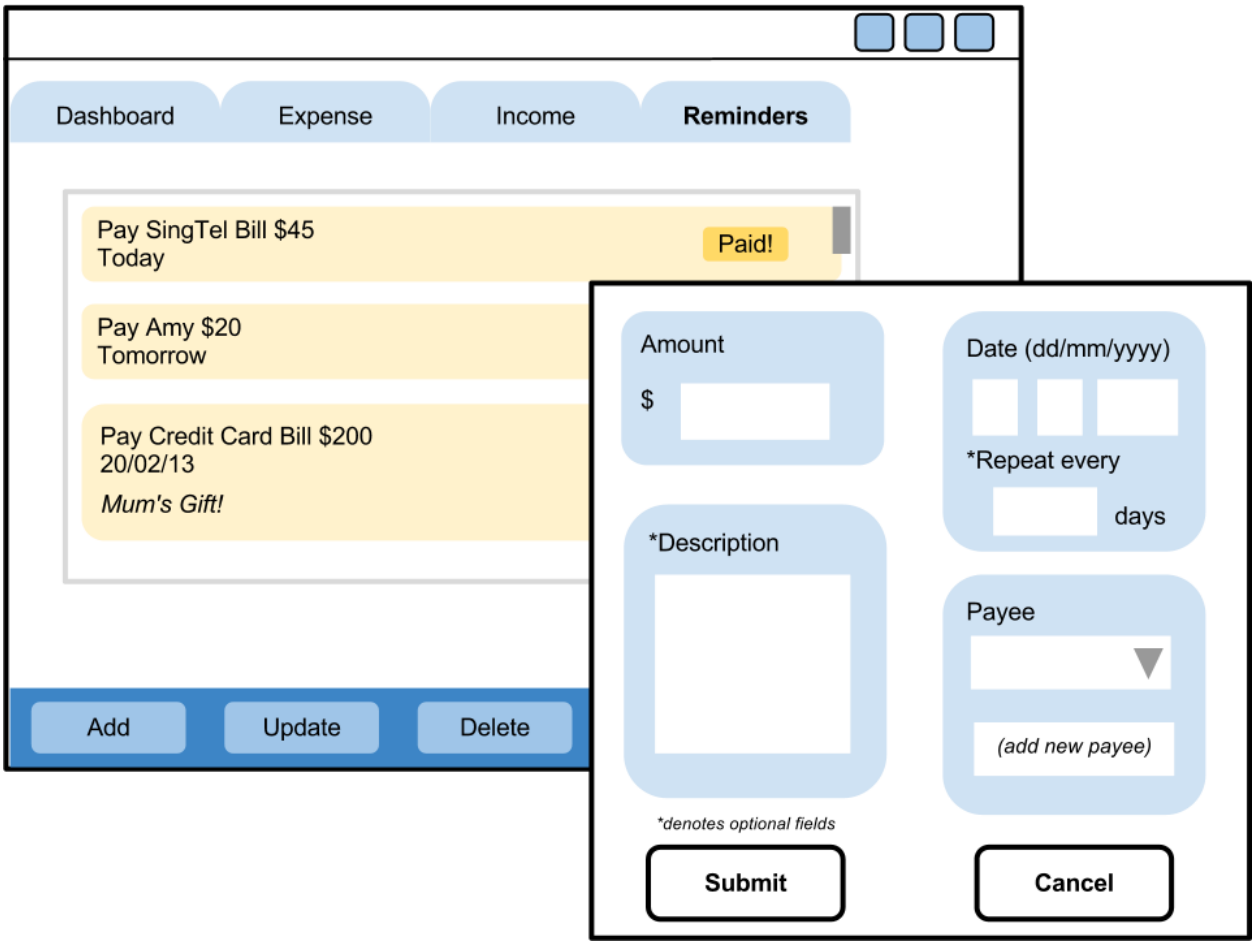
Upon clicking ‘Submit’, the entry is updated, and the pop up window closes.

If ‘Cancel’ was selected, the pop up window closes and any changes will not be registered.

If ‘Search was selected, the user can enter anything in any of the fields. Upon clicking ‘Submit’, a search will be conducted for any entries matching of *any* of the filled fields, and the pop up window closes. The results of the search will be displayed on the main window below the chart and the list of entries.

All of the following functions work similarly for the income tab. Some things to note though, is that the charts for income will be green, and that the income categories and expense categories are not shared (separate).

Reminders



When the Reminder tab is selected, a list of all reminders (snoozed or un-snoozed) will appear in the scrollable list. As with the expense and income entries, the reminders will be displayed in chronological order in a standard format, though this time, beginning with the earliest due and ending with the latest. At the side of each entry is a ‘Paid!’ button that allows the user to indicate that the bill has been paid. Upon clicking the button, the information from that entry is entered as a new expense, with the payee as the category, and then the reminder is deleted.

At the Reminder view, the user can add, update or delete reminders by selecting on the respective entry in the pane and then the button below. A pop up will then appear, in a similar fashion as with the expense and income views. However, instead of categories, reminders will have a different field called payee instead. As the idea of having a payee is for categorization, the layout of the pop up is intentionally kept consistent.

Project Logistics

Roles and Responsibilities

Team Member	Role
Liu Bohua	Timekeeper (ensure we meet our deadlines)
Lin Zhiyuan	Programme tester
Pang Kang Wei, Joshua	Programme code
Wong Jing Ping	GUI designer
Hao Wei	Documentations

Project Timeline

Week	Tasks
3	Discussed about the general ideas about the project and decided the features that would be included in our programme
4	Discussed our primary views with project tutor Composed primary parts of the proposal
5	Finish and submit the 1st proposal
6	Finalize program architecture Assign coding tasks to each individual for baseline features
Recess	Code individual parts
7	(Midterm) Code individual parts
8	Test components Build v0.1
9	Push to repo v0.1 Project Report v0.1

Tool Infrastructure

Development Tool	Eclipse Juno & Windows Builder
Third-party Library	jFreeChart, charts4j
Versioning Tool	TortoiseHG & Google repository
Team communication & documentation tool	Google groups & Google docs