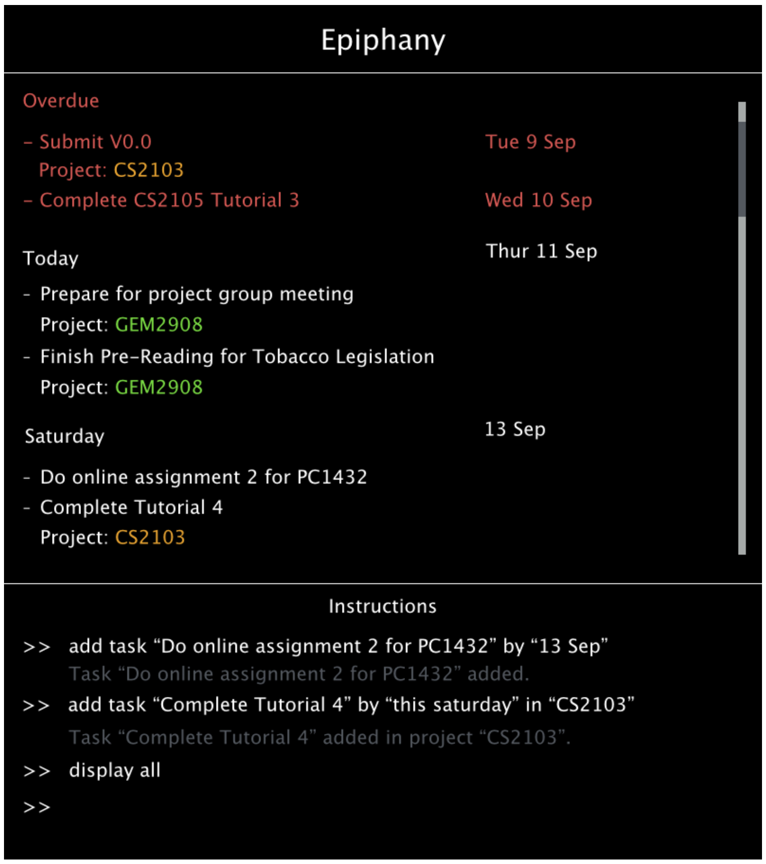
Epiphany

UI Sketch



Epiphany

Overdue

- Submit V0.0 — Tue 9 Sep
  Project: CS2103
- Complete CS2105 Tutorial 3 — Wed 10 Sep

Today — Thur 11 Sep

- Prepare for project group meeting
  Project: GEM2908
- Finish Pre-Reading for Tobacco Legislation
  Project: GEM2908

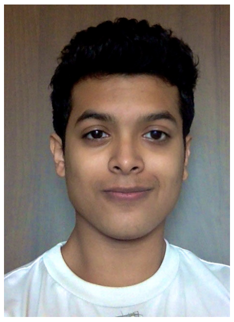
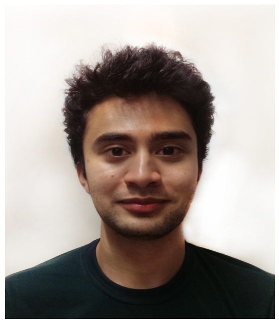
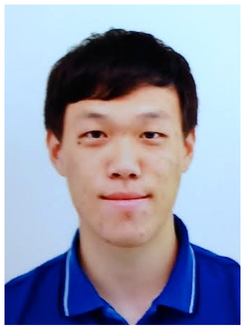
Saturday — 13 Sep

- Do online assignment 2 for PC1432
- Complete Tutorial 4
  Project: CS2103

Instructions

```
>>  add task “Do online assignment 2 for PC1432” by “13 Sep”
    Task “Do online assignment 2 for PC1432” added.
>>  add task “Complete Tutorial 4” by “this saturday” in “CS2103”
    Task “Complete Tutorial 4” added in project “CS2103”.
>>  display all
>>
```

Supervisor: Shawn Lee
Extra feature: GCal integration

			
Amit Lead Epiphaneer	Abdulla Epiphaneer Minion	Moazzam Epiphaneer Minion	Wei Yang Epiphaneer Minion

User guide

Overview

The aim of this project is to create a task manager/ scheduler to enhance the user experience. This would be achieved through ease of use and convenient access to tasks and schedules.

It is to be noted that the project especially aims to be keyboard friendly in the sense that all tasks; add, remove, edit or undo would be easily used through the keyboard and minimal if any interaction with the GUI would be required.

Graphically, the application is split into the output (upper) and input (lower) terminals. When the app is first launched, the default view in the output terminal are the tasks that are due today as well as any overdue tasks. The following subsections explains how each of the task would be performed.

Viewing tasks
There are several formats of viewing tasks in the application, all of which can be accessed by a simple two word command. The different formats are as follows

Input: display all
Description: displays all the tasks that the user has added which are sorted according to date. This takes the form of an infinitely scrollable list.

Input: display today
Description: displays all the tasks that the user has due today. This screen would also display all the overdue tasks. This would also be the default view in the GUI.

Input: display ongoing tasks
Description: displays all tasks that are currently ongoing.

Input: display #projectName
Description: displays all the tasks that the user has added under a specific project.
Example: display #CS2103

Input: display “date”
Description: displays all tasks due on a specific date.
Example: display “15 Sep”

Adding Tasks
Adding tasks can be done with a simple intuitive command which is close to natural English. Here the quotes act as delimiters and indicate where the content to be added begins and ends.

Input: add “taskDescription”
Description: adds a task without a deadline attached to it - a floating task
Example: add “Save all of mankind”

Input: add “taskDescription” by “dueDate”
Description: adds a task with a deadline attached to it.
Example: add “Prepare for project meeting” by “this Friday”

Input: add “taskDescription” #projectName
Description:adds a task without a deadline attached to it, to a specific project.
Example: add “Save all of mankind” #justEverydayThings

Input: add “taskDescription” by “dueDate” #projectName
Description: adds a task with a deadline attached to it.
Example: add “Prepare for project meeting” by “this Friday” #GEM2908

Editing Tasks

Input: delete taskId
Description: You would first have to display your tasks, which would give you unique ID’s (UID) for each task, you then can choose which of those tasks to delete.
Example: delete 123

Input: edit taskId to <follows add task syntax>
Description: Given the UID of a task, we can use this syntax to edit any task.
Example: edit 123 to “Prepare for project meeting” by “this Friday”

Input: add deadline taskId “dueDate”
Description: Given the UID of a task, we can use this syntax to edit any task.
Example: edit 123 to “Prepare for project meeting” by “this Friday”

Input: taskId is done
Description: Given the UID, you can mark a task as complete, which would place it in the archives and return to the default screen.
Example: 123 is done

Input: taskId is ongoing

Description: Given the UID, you can mark a task as complete, which would place it in the archives and return to the default screen.
Example: 123 is ongoing

Input: add “taskDescription” by “dueDate” #projectName
Description: adds a task with a deadline attached to it.
Example: add “Prepare for project meeting” by “this Friday” #GEM2908

Project Management

Input: create new project #projectName
Description: You can use this syntax to create a new project(list). You can later add tasks to this project using the syntax described in the adding tasks section.
Example: create new project #PC1432

Input: update #projectName to #newProjectName
Description: You can use this syntax to update an existing project.
Example: update project #PC1432 to #PC1431

Search

Input: search “keyword/phrase”
Description: searches all task names (both inside and outside of projects) for this keyword. Not case sensitive. Displays all of these in the output terminal afterwards.
Example: search “essay”

Sync

Input: sync
Description: Synchronizes all your tasks to google calendar. (This is our special feature)

Appendix A: User stories. As a user, ...

[Likely]

ID	**I can … (i.e. Functionality)**	**so that I … (i.e. Value)**
addTask	add a task	can record tasks with a deadline.
addFloating	add a task without a deadline	can record tasks that I want to do some day.
deleteTask	delete a task	no longer have to track it.
completeTask	mark a task as completed	can view it in archive later.
ongoingTask	mark a task as ongoing	so that I can plan my work.
modifyTask	change details of a task	can have flexibility.
displayTasks	view all my tasks in one place.	can plan my work
readInput	add tasks through command line style arguments	can easily create new tasks
remindMe	recieve a reminder	remember to do things.
editTask	modify a task	can edit the task
undo	undo the last action (this does not stack multiple times)	can easily disregard a wrong command.

[Unlikely]

ID	I can … (i.e. Functionality)	so that I … (i.e. Value)
addProject	Create a project which can then contain tasks within it.	can organize my tasks into an intuitive format and make filtering easier.
powerSearch	Search through all my tasks, including date and notes field.	can easily search for anything i want
syncToGCal	Export my tasks to google calender	can easily integrate with any other calendar app i may have.

addNotes	add notes to any task	can elaborate on tasks without making the title messy

Appendix B: Product survey

Product: iStudiez Pro **Documented by**: Amit Gamane <u>Strengths:</u> ● Good organization ● Easy to navigate UI ● Ability to add/edit tasks with deadline ● Shortcuts for deadline(due next class etc.) ● Ability to add teacher information ● Integrates with iCal and shows how you schedule looks like ● Variable Reminder settings ● Holidays can be added <u>Weaknesses:</u> ● Doesnt allow long term grade tracking, only task-based grading ● Backup works via email, could have included calendar syncing ● Displays all assignments due at once. I'd like to set an deadline to commit it to m schedule but may not want to see it on my list immediately.
Product: Todoist **Documented by:** Abdulla Contractor <u>Strengths:</u> ● Simple and clean design ● Sends reminders by email ● The app is available on a wide variety of platforms, allowing you to access it from almost anywhere. ● You have the choice to collaborate with others on a project. ● You can create subtasks and by doing so break large tasks into smaller managable ones. ● You can quickly write due dates using normal language, such as "monday at 2pm". <u>Weaknesses:</u> ● Does not sync to google calendar, it would be great if it could because the calendar on all my devices is integrated with google.
Product: http://todotxt.com/ **Documented by**: Tin Wei Yang

References:
http://computers.tutsplus.com/tutorials/how-to-manage-your-tasks-with-todotxt--cms-202
93
https://github.com/ginatrapani/todo.txt-android/releases/tag/release34

Strengths:

- Note taking at its purest, no reminders, checkboxes. Mimics the way how notes are taken down in real life - you write them down on a piece of paper
- Helps prioritize your to-do items & organize them into projects and contexts. This is done through the addition of a "+" tag followed by the project name at the end the task. e.g. Do the dishes. +cleaning. Context can be added with an "@" sign followed by the name of the context. Priorities are designated with an uppercase, A-Z e.g. (A). A has a higher priority compared to B and so on.
- Open source project. Works in a variety of apps like Todour
- Command line editing of your task file
- Clean and minimal interface
- Modular - ability to combine with apps and the command line interface makes it a powerful as what the user would like.

Weaknesses:

- A slight learning curve initially.
- Unable to set time reminders, notifications
- Information is displayed in text, may not appeal to people who are more visual.

Product: Google Calendar integration with S planner **Documented by:** Moazzam Ali Khan

Strengths:

- Great synchronization features.
- Ability to synchronize with phone as well.
- Allows both tasking and calendar mode.
- Lots of options for calendar view; Daily, 4 days, weekly, monthly, etc.
- Great search function.
- Ability to combine with other apps.
- Email and sms reminders for flagged events.
- Allows easy addition of events, click from email.
- Daily Agenda mode; enables a daily agenda to be received via email at a set tim
- Allows sharing of calendar with others.

Weaknesses:

- Design is complicated. Needs getting used to.
- Too many settings which take users time to understand.