UNSW SYDNEY

School of Computer Science and Engineering

COMP4920 - Management and Ethics 2019, T3

Project Plan: Hangouts - An Event Planner System

Team Name: Pls Let Us Graduate!

Name	zID	Role(s)
Rodney	z5078786	Scrum Master/Developer
Anna	z5115212	Developer
Cindy	z5113176	Developer
Quoc-An	z5114240	Developer
Harvey	z5147986	Developer

Table Of Contents

Introduction

Existing Systems and their Drawbacks

There currently exists a few web applications that allow for scheduling events, or have introduced similar functionalities:
Facebook: Facebook enables users to create event pages, both public and private. These events allow for users to invite their friends, or be open for all users to join. They can be used to post event information and the discussion board is free for all invitees to comment or post in. Facebook also allows for events to be scheduled casually through Messenger, where people can select whether they'll be attending or not.
Meetup: Meetup is a web application that allows for online groups to schedule meet-ups together. Users can also choose to join events of their interests that are or will be taking place in the near future.
Timeweave: Timeweave is an app for University students to share their university timetables with friends and schedule meet-ups accordingly.

Although these applications serve well in scheduling events, they do not exclusively enable individuals to clearly identify their availability outside of their university timetables or one time events within a closed community among friends. For events that require knowing all invitees' availabilities, both Facebook and Meetup's event creation process involves a lot of back-and-forth communication between hosts and other users in order to pinpoint a suitable day and time where most/all members are available. Timeweave has a small demographic within which it operates, ie. University students, as it is only concerned with the sharing of university timetables. Therefore, it doesn't allow for the planning of events outside of the education domain. As such, the lack of attention in helping individuals to determine not only their own, but others' availabilities in order to meet up is the main problem domain of existing systems.

Our Solution

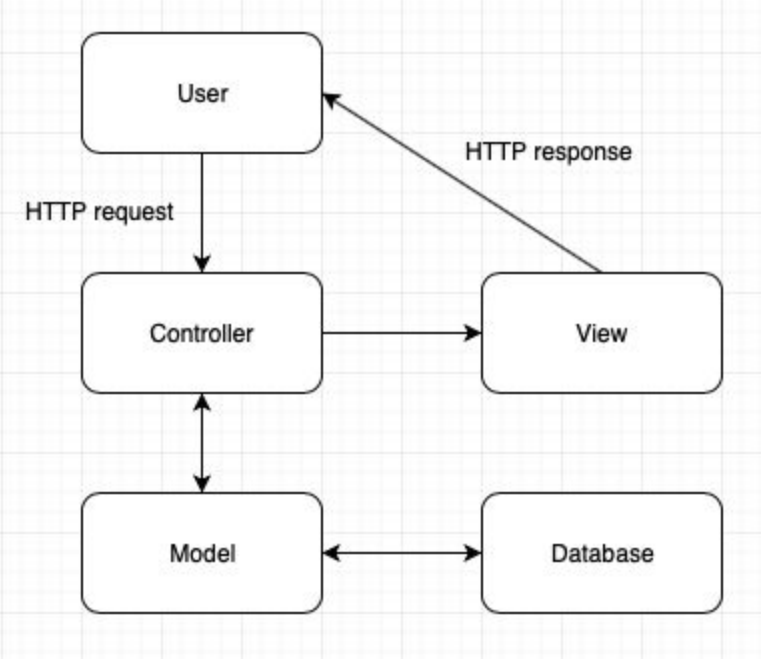
Targeting mainly students and young adults with busy and dynamic lifestyles, we endeavor to create an application with the sole purpose of helping individuals to schedule meetups. The application, 'Hangouts' will allow users to create and join events, invite their friends and allows event scheduling between users by requesting for availabilities for a particular time period, ensuring other users can see the availability of everyone's timetable. By creating a timetable for each individual based on their availabilities, Hangouts will allow users to pinpoint common available times between

each timetable and create a private event for the group accordingly. The host can also create an event on a specific date and enable users within the group to confirm their availability.

Our Differences

Hangouts will be more personalised compared to other social networking or meet up applications. Its focus is to provide a seamless experience for users within a private community to schedule meet-ups and events through direct and easy interactions between users for determining common availability. This convenience encourages more meet-ups with a structured format, ensuring that every user can vote for a particular meet-up date for the host to decide on the most convenient date. This kind of event creation system also streamlines the process of finding common availabilities between users.

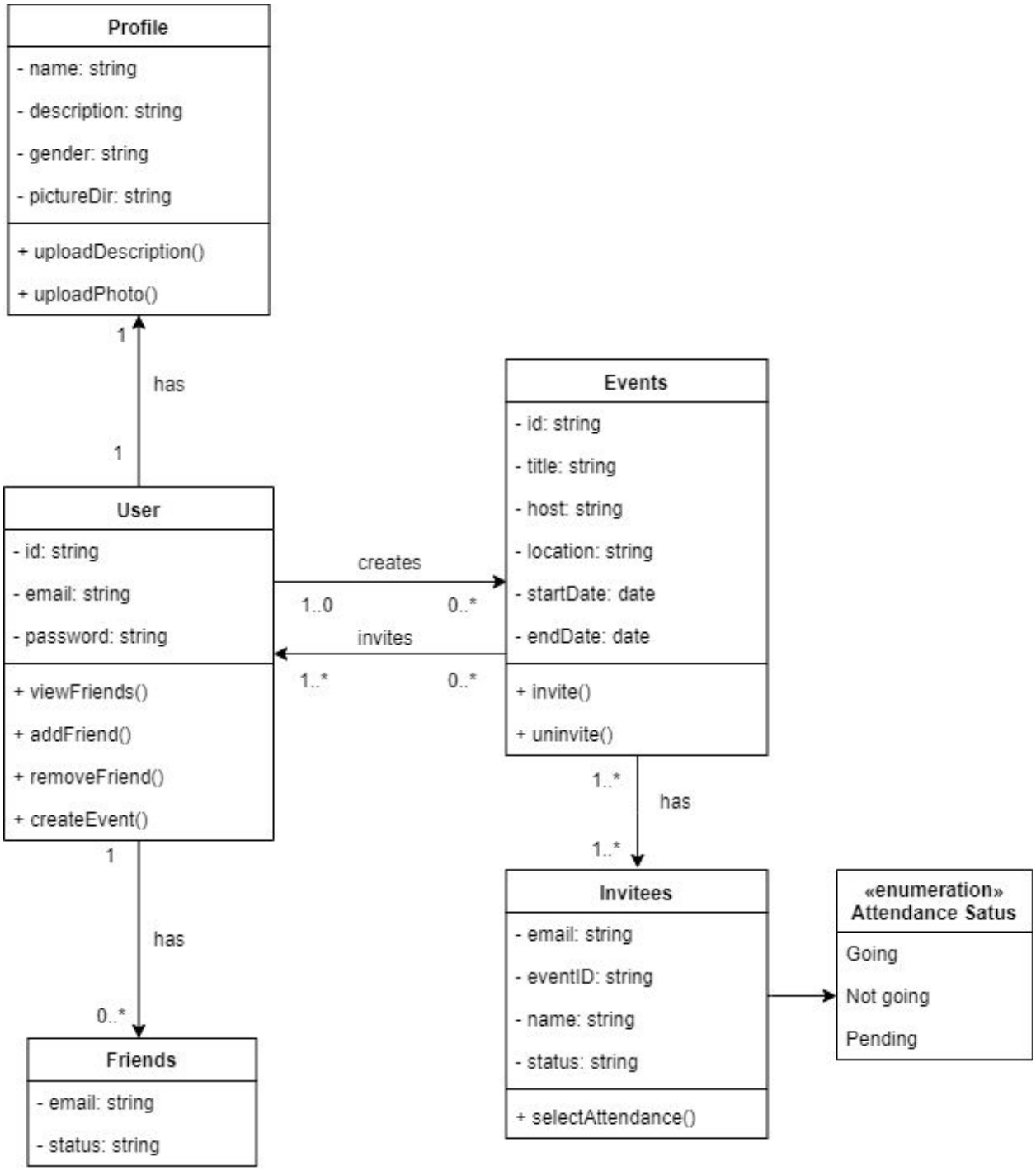
System Architecture



Model-view-controller (MVC) is a modern architectural pattern that divides the project into three interconnected parts. MVC allows for efficient code reuse and parallel development. This design pattern can possibly benefit the team.

Class Diagram

Below is a simple class diagram of our current vision for Hangouts.



User Stories

1 Story Point (SP) = 2 hours

EPIC01 - Account Component

US101

“As a user, I want to be able to log in and out of my account securely to protect it from being accessed by outsiders”

Description:

- The authentication system will involve verifying a user’s email with their password
- Once a user logs out, they will be unable to return back to the previous page prior to logging out

Size: 2 SP

Priority: High

US102

“As a user, I want to be able to write and edit my own profile as a way of introducing myself to others that view my profile so that they can identify me”

Description:

- Each user will have their own profile description that they can write or customise for others to see
- There will be an option to edit the profile as the user sees fit
- Any unsaved changes will be lost if the user were to forget to save before exiting the page

Size: 1 SP

Priority: Low

US103

“As a user, I want to be able to add and delete friends so that my friends list can always be updated”

Description:

- Each user will have a friends list that they can view
- By browsing other people’s profiles, users can request to add them as friends

- Through their friends list, users will also be able to delete friends from their list as they see fit

Size: 3 SP
Priority: High

EPIC02 - Hosting Component

US201

"As a user, I want to be able to create and edit my own events for my friends to see and join"

Description:

- An event can be created as long as it has a specified title and date
- There are two types of events:
    - Scheduled: Planned events with already planned date and time
    - Unscheduled: Events to be scheduled depending on the timetable of all those involved
- For unscheduled events, users have the option to be able to share their timetables with their friends, and vice versa, so that they can plan an event time and date if they need to do so
- All events are private and for users' friends to see only
- Users will be able to go back and edit the details of their event once it is created

Size: 3 SP
Priority: High

US202

"As a user, I want to be able to invite friends to my events so that they can be aware of it and join it if they're interested"

Description:

- There will be an option for users to invite friends from their friends list once they create an event
- If the user's friends agree to attend, they can select "Attending" and it'll show up in the attendance list
- Friends can also be uninvited, or can change their attendance status

Size: 1 SP
Priority: High

US203

“As a user, I want to be able to delete events in case of unforeseen circumstances”
Description:

- Once a user chooses to delete an event, it cannot be restored
- This event will be completely removed from the database
- No one else will be able to view this event once it has been deleted

Size: 2 SP
Priority: Medium

US204

“As a user, I want to be able to add and edit posts for my own event so that I can keep all attendees/potential attendees updated”
Description:

- Users can create posts and post them onto the event page for invitees to see
- If the event is public, all users can see the post even if they’re not specifically invited
- These posts can be edited as the user sees fit
- If any unsaved changes are made to a post, they will be lost if the user exits or refreshes the page

Size: 2 SP
Priority: Medium

US205

“As a user, I want to be able to remove invited friends from my events so that I can change who I want to come”
Description:

- Once friends are invited, host is able to remove the unwanted friend immediately
- The friend removed the event page will no longer be able to attend and will not see the event anymore
- The removed user will also not receive any updates about any changes relating to that event

Size: 1 SP
Priority: Low

EPIC03 - Joining Component

US301

“As a user, I want to be able to join events I’m invited to so that I can let others know that I’ll be attending”

Description:

- Once the user has been joined, the user has the option of going, not going or tentatively going which will be displayed on the event page for all attendees to be aware of
- Users have the option to change their attendance status at any time prior to the event
- A list of users can be viewed under each attendance option as well with the total number of users for those options

Size: 2 SP

Priority: Medium

US302

“As a user, I want to be able to invite friends to events that I’m attending if the event host allows so that I can share my interests with them”

Description:

- Users can navigate through their friend list to invite to events
- The host can give invite privileges
- The host can also reject or remove invite privileges
- Once invited, the invited person should be able to see the event page and its details

Size: 1 SP

Priority: Low

US303

“As a user, I want to be able to input my availabilities when I get invited to an event if the host allows for it so that it’s clear to everyone when I’m free to join”

Description:

- Users will be able to import their availabilities into a calendar that all invitees of the event can view
- This feature is only allowed if the host enables it

Size: 3 SP

Priority: Medium

EPIC04 - Social Component

US401

“As a user, I want to be able to search for friends so I can add them and view their profiles”

Description:

- A search bar is implemented where users can type in the name (first & last) to search for a desired user.
- The user is then returned with a list of possible candidates, with the option to add them beside their name.
- Clicking on any of the returned users link to their profile.

Size: 2 SP
Priority: Medium

US402

“As a user, I want to be able to create and join specific groups so that I can be a part of group events with them”

Description:

- Users will be able to create or join groups with their friends
- Within their groups, they can create or schedule group outings or events for the members to attend

Size: 1 SP
Priority: Low

Optional User Stories

The following user story overviews are optional features that we can implement if we find that we have underestimated the time needed for our project:

- Merging timetables: *“As a user, I want to be able to share my weekly schedule together with my friends so we can pinpoint the best day to meet up”*
    - Timetables can be shared between friends to find the most optimal time during which all or most are free to meetup
    - There will be a function to merge these timetables together to pinpoint a certain date or time, which will be outputted to the user and their friends to plan an event together
    - **Size:** 3 SP

- Chat system: *"As a user, I want to be able to have more social interactions with my friends for convenience and to improve my user experience"*
    - Implementation of basic chatting system that'll enable users to communication with their friends
    - **Size:** 4 SP
- Public events for all to join: *"As a user, I want to be made aware of large-scale public events that I may be interested in so that I can attend them"*
    - Allow for the creation of public events for anyone interested to join (such as large-scale demonstrations, festivals etc.)
    - **Size:** 2 SP
- Show all past events: *"As a user, I want to be able to look back on my old events so that I can look back at them for future event references or to reminisce about old memories"*
    - Our current project involves removing all events that have passed
    - This optional feature will allow for users to view past events that they've been apart of
    - **Size:** 1 SP

Product Backlog

Priority	User Story	Difficulty	Deliverables
High: Minimum Viable Product (MVP)	US101	2	Account and Authentication System
	US201	3	Create and edit events
	US103	3	Add and delete friends
	US202	1	Invite friends to events
Medium	US203	2	Delete posted events
	US204	2	Add and edit posts to event page
	US301	1	Support event RSVP
	US303	3	Input of availabilities
	US401	1	Search for friends
Low	US102	1	Change and edit profile
	US205	1	Uninvite friends

	US302	1	Allow mutual friend invites
	US402	1	Creation of groups

Project Methodology

For this project, we will adopt an agile SCRUM methodology with week long sprint cycles and regular stand-up and sprint meetings.

We will use the following to implement our project:
Main languages:

- Front-end: HTML, CSS, JavaScript
- Back-end: Python3

Framework: Flask
Database: MongoDB

Team Roles

SCRUM Master: Rodney
Product Owner: Shared between all members of the team, with weekly rotations
Front-End Developer: Cindy, Anna
Back-End Developer: Harvey, Quoc-An

Project Management

We will employ **GitHub** to store our project and share our progress with the rest of the team. We will also use **Pivotal Tracker** to track our project's progress through the use of the Pivotal Tracker's product backlog and sprint schedule. Each member will also keep records of our daily standups with their progress and planned responsibilities for the project in their individual **diaries**.

Sprint Schedule | Sunday

Our sprints will last a duration of 1 week, with each sprint being due on the Sunday of that week. As such, our schedule is as follows:

Dates - Day	EP-US	Deliverables
20/10/19 - Sunday	EP01: US101, US102	Authentication system,

(week 5)		Change and edit profiles
27/10/19 - Sunday (week 6)	EP01: US103 EP02: US201, US204	Addition and deletion of friends, creation of events
03/11/19 - Sunday (week 7)	EP02: US202, US205 EP03: US301	Invite and uninvite friends, Event RSVP
10/11/19 - Sunday (week 8)	EP02: US203 EP03: US303	Deletion of events, Input of availabilities
17/11/19 - Sunday (week 9)	EP03: US302 EP04: US401 EP04: US402	Search friends, Mutual friends invite, Groups
24/11/19 - Sunday (week 10)	Optional Epics	*possibly extra features if all core deliverables are completed on time

Meeting Schedule

There will be three face-to-face stand-up meetings, one sprint retrospective and one weekly online meeting per week:

1. (Stand-up) Tuesday → 1:00 p.m. to 2:00 p.m. @ Science and Engineering Building
2. (Stand-up) Thursday → 1:00 p.m. to 2:00 p.m. @ Science and Engineering Building
3. (Sprint retrospective) Friday → 1:00 p.m. to 2:00 p.m. @ Blockhouse G16
4. (Online meeting) Sunday → 7:30 p.m. to 8:30 p.m. @ Online via Discord voice chat

Extreme Programming Practices

A number of XP practices will be employed throughout the course of the project development, including the following:

- **Pair Programming**, where we'll be able to receive and share constant feedback on our code and project progress
- Daily standup meetings, whether they be face-to-face or online will be conducted to ensure all members are aware and held accountable for their responsibilities
- **Small incremental releases** as per the sprint schedule

- **Code refactoring** to ensure code coherency and that there is no redundant or tightly coupled code/functions
- **Simple design** to relieve pressure given the tight time-frame for the project development