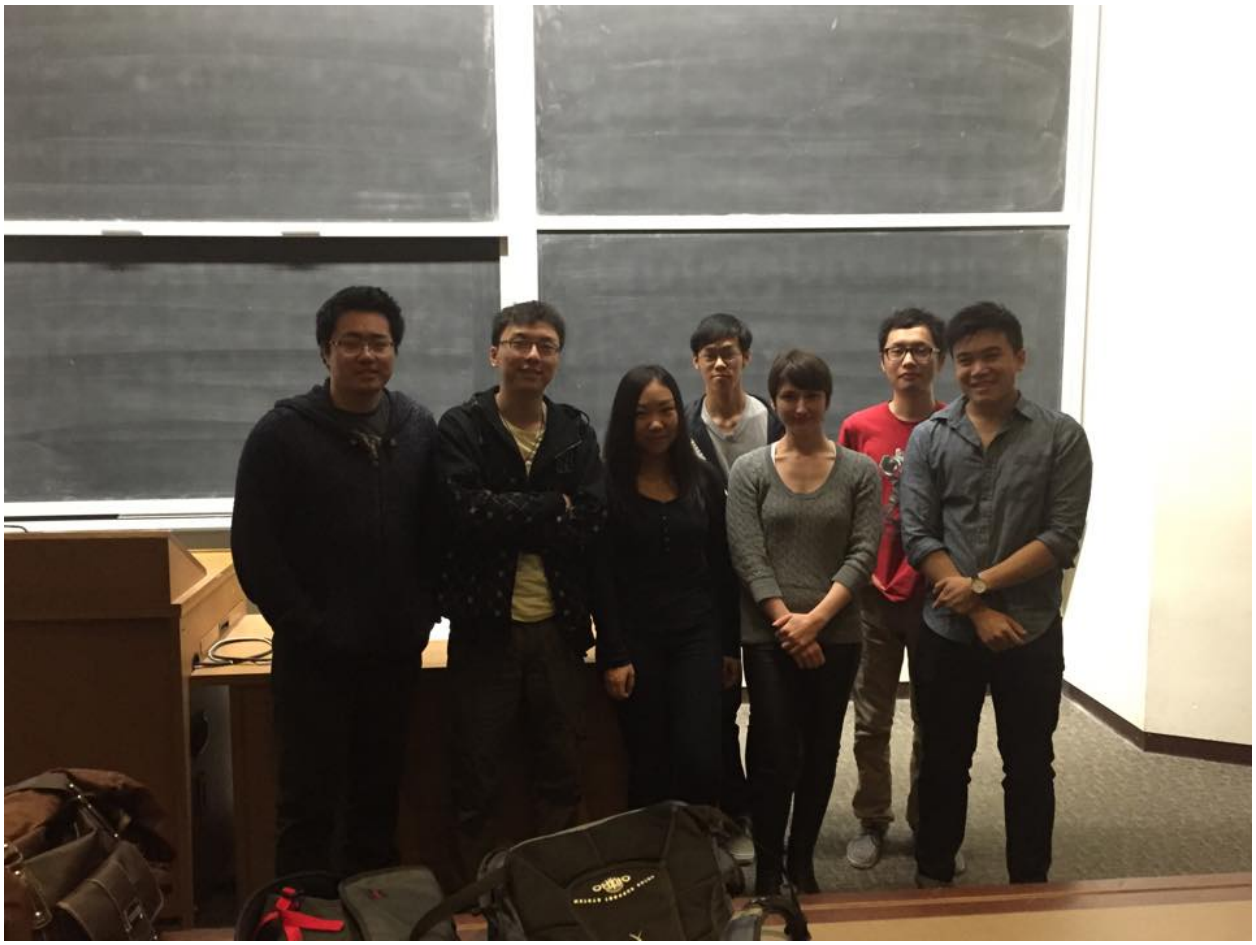
Introduction

Team introduction

Team members



Short Bios (sorted alphabetically by first name):

Alina Gvozdik	A third-year computer science student. Previous experience includes Android programming, as well as website design for a start up hearing company.
Kathy Zhang	A third year computer science student. She is experienced with Java, JavaScript, C, and HTML. Last summer, she worked as an intern in a bank to help design forms and websites. In her spare time, she likes baking, running and travelling.
Li Wan	Li Wan is a senior student at University of Toronto, with bachelor degrees in Computer Science(focus in scientific computing) and Mathematics in progress. He currently works in a campus team called Konectivity, where he works in a group to develop a mobile app on iOS, Andriod and BlackBerry using Javascript and HTML.
Owen Lu	Third year CS major (plus Math minor or major? not yet decided), have experiences in python, java, C, SQL, javascript, html5 and web programming. He's a professional modeller who often goes to hobby shops.
Timothy Chu	A third year computer science specialist student at the University of Toronto St George campus. In his spare time, he likes to hang out with my friends or play sports. He is a huge NBA fan and strives to play basketball as often as possible. Is experienced with Java, Python, HTML, javascript, SQL and other languages.
Lingfei Gao	He studied CSC309 last summer and complete a big web programming project with Timothy and Matthew. He found it interesting to work with a team which has more than 4 people, thus he is taking CSC301 this term. He is experienced with Java,C,HTML,JS and etc. He is pro at gaming!
Matthew Lei	"3.5" year computer science specialist / cognitive science major. Previous relevant projects include developing a website with a team, and researching for and creating front end designs for an android application. He is experienced with photoshop and basic android development.

Choosing the Product

The problem that our group has decided to work on is the "time management tool" for students. Through a group meeting, we unanimously agreed on the time management tool over the messaging tool. We all agreed that a messaging tool was not as essential as a time management tool because we are already accustomed to many applications such as Piazza or Facebook to communicate with others and broadcast information. However, as students, we often have many responsibilities outside of school, and it can become hectic to allocate the proper amounts of time for school projects, tests, jobs and social events. Therefore agreed time management tool would definitely be more useful, allowing students to organize their busy lives.

Personas

- Julia is a 4th year Psychology student. She commutes to school 5 days a week. She never carries her laptop with her, but she is inseparable from her tablet and phone. She does not believe in spending a lot of time on organizing her schedule, and likes to do things as they come. As such, she will not spend the time to input every class into the organizational tool manually. She likes her schedule to be as flexible as possible to be able to accommodate various social events. She would like to be informed of campus events such as guest lectures on topics such as psychology and astronomy. She is also currently studying for her GMATs, so she would like to know about any mock exams organized by the university.
- Jim is a teaching assistant for a computer science course, and a PhD student. He is very dedicated, so he would like to be as available to his students as possible, but he has a very busy schedule and his availability tends to change up from week to week. In addition to that, he lives very far from campus. As such, he would like to be able to schedule additional office hours dynamically either by sending out his suggested office hours and receiving number of students planning to attend, or by approving requests from his students. Jim also has a tendency to become very engrossed by what is doing at the moment, so he needs very loud reminders of upcoming appointments.
- Shadi is an undergrad student. She is currently doing her specialist in philosophy. Shadi lives very close to campus; she rents a house in the area with various other students. She is a vegan, and has more food allergies than she can count. She has a tendency to procrastinate until the last moment. This applies to a lot of things with her – from exercising to schoolwork. Shadi learns best when she studies with other people, but the class sizes are large, and she is too shy to approach people she doesn't know. She would like to be able to post study group requests that her classmates will be able to see and confirm. She would also like to have reminders for things like her yoga classes, paying her rent, and project and assignment due dates.

User Stories

High Priority (Essential features) 5-7 : -- we intend to implement in Phase 1
Medium Priority (Valuable, but product is viable without it) 3-4 :
Low Priority (Might be valuable, look into later) 1 - 2 :

	User Story	**Difficulty/Size**	**Priority**
1	As a student, Tim would like notifications for his work schedule and projects	medium-hard	7
2	Alina would like to sync/import calendars from other devices to a task manager so there is no pain in swapping to it	may be difficult if no APIs are provided	4
3	Owen, a student, would like to colour code his tasks so he may easily separate work from fun	easy	5
4	Ling Fei is a student that would like an application that helps him quickly decide when he should allocate a task/events so he can plan his schedule easily	medium	7
5	*Kathy is a professor that would like to remind students of her additional office hours and locations so she can better their learning in her currently empty office	medium-hard	4.
6	*Professor X would like to push notifications to his students to remind them of assignment due dates	“”	“”
7	Tanya would like a tool which summarizes her time spent so she can better understand how to allocate her time	medium-hard	4
8	Harry would like notifications whenever a scheduled task approaches so he can stay punctual	medium	6.5
9	Ron would like to have an aesthetically pleasing calendar so he can get rid of the currently crowded looking applications on his phone	hard, we need to define “aesthetically pleasing”	5
10	Hermione would like a schedule to keep track of her classes so she knows when and where her classes are	Easy-medium	7
11	*As a student, Shadi would like to be able to make study group requests to her fellow classmates.	medium-hard	2/3
12	*As a student, John would like to be able to accept study	“”	2/3

	group requests from his fellow classmates.		
13	*As a TA, Jim would like to be able to suggest additional office hours, and receive confirmations from students planning to attend.	“”	2/3
14	*As a TA, Jim would like to be able to receive requests for office hours and to respond to such requests.	“”	2/3
15	*As a department head, I would like to be able to send push notifications for upcoming events to every student in the program.	“”	3
16	*As a student, John would like to be able to unsubscribe from push notifications from certain parties.	medium-hard	3
17	*As an administrator, Bob would like to be able to send push notifications for important dates (such as university closure), or other alerts.	medium-hard	2
18	As a student, Max would like to be able to request an extension on assignments.	medium	2
19	As a professor, John would like to be able to review extension requests.	medium	2
20	As a professor, John would like to be able to disable extension requests on individual assignment/test dates.	medium	2
21	As a student, I want to share my schedule with my friends and team members.	easy	4
22	As a student, I would like to label my finished tasks.	easy	7
23	As a professor, I would like to see my students’ other courses’ deadlines.	medium	2
24	As a student, I want to save my schedule of the next week/month as a pdf file.	difficult	4
25	As a student, I want a calendar to be able to calculate how many days I have between two deadlines.	easy	3
26			

* Requires some kind of database with different groups (People, Admin, Prof, TA, Students, Department, etc...).

BRAINSTORM

Design (we want to be unique)

- Weekly planner?
- calendar is hard to look at on a small device (solution: print/pdf/email/web friendly)
- we can add “today’s quest” indicate thing should be done that day as a list
- reward yourself system? (Post on facebook option available.)
- Smart encouragement function when the user checks a task as finished or when the user will have a very busy day/week.
- location detection: remind to get coffee/food/life supplies
- remind the user to go to the gym and track the user’s workout history
- when there is a unique event going on near the user’s location and the user is currently free remind the user to join
- tell the user a joke/news when detecting the user might be very stressful studying
- smart weather report: remind all users to take umbrellas/go to school earlier/ wear more/less clothes
- set milestone (half semester/ half year)
- set count down
- text to speech/ siri
- search event

Minimum Viable Product

A summation from User Stories:

- Add/remove tasks
    - notify and help user when two tasks are assigned in same time slot
- Add/remove push notifications
- Log in
- Add schedule (timetable sort of thing)
- Notifications when event/task arises
- Quick-task adder (Suggest time slots when user specifies a time)
- Simple, clean design.

The MVP:

The timemanager will help a user handle their time. Users will be able to set up a schedules, events, and notifications for them. A schedule will help users keep track of recurring events by reminding them periodically of looming scheduled tasks or events. A notification will be sent by the application as reminders. They will be able to adjust how early and how often they are reminded. A special unique feature our product will have is a 'smart' quick task adder. When a user wishes to add a new task, the application quickly calculates a good possible time slot for the task. The application will not be a simple calendar application. Users will experience a sleek task manager that will actively help the user plan their schedule.

Release & Iteration Planning

For our first release, we plan to only incorporate the most essential user stories. As a group, we believe one of the most important functions our tool will need is notifications. This is the adresses the first user story, where the student would like notifications to help balance his school and job life. We believe this is one of the most important concerns because students often have many responsibilities outside of school and will require reminders to help them manage the multiple aspects of their lives. The third user story is also important for the same reason. We want students to be able to quickly distinguish what type of task or event is coming up allowing the student to easily plan accordingly. This also ties in with user story 10, which states that the student will want their school schedule on the time management tool. For our first release we plan to not only show the student's schedule, but have it added automatically. Their schedule will definitely use a large portion of their time and making it automatic will save tons of work for the student. Another user story we think will be crucial for our first release is importing the calendar from other devices or frameworks. We understand that many students will already be using other calendars, and having an easy way to sync or import their previous calendars will make their transition simpler. Lastly, perhaps one of the most important functions

will be the ability to suggest activities for a given period of free time and to easily add the tasks to the calendar for that specified period of time. This is described by user story 4. Since this is a time management tool, it should help students use their time more efficiently and it can do that by suggesting how to use unallocated free time. These core user stories will be our main focus for the first release of our product.

However, we also have many other non-essential services we want our product to provide. Two user stories we will leave for later are number 11 and 12. These two suggest that students may want to organize group study sessions with other students in the class. We are leaving this functionality until after the first release because it is not our main focus to connect students with one another in this project. We agree that it would be a great additional feature if students could create a group study session on their calendar and have other students see the event on their own calendars and join, but we need to focus on more essential time management features first, such as being able to add study sessions for the student themselves first. Another user story that we will definitely need to incorporate is the ability for professors to send notifications to students in their classes. This is a very important functionality that we will need for our final product. However, the reason it is not in our first release is because we are assuming that professors will not be very cooperative in using this tool, so even if we added this functionality, it is likely it will not become very useful since it is possible that no professors will use it. Lastly, one more feature that is important but not included in the first release is collecting statistics about how users are spending their time. This is user story 7. We believe this is very useful because it will allow students to reflect on how they are using their time and will be a true evaluator of how effectively they are managing themselves. Not to say that this is insignificant, but it is more vital to suggest how to use time more effectively than show how they are not using time effectively. Since this is only our very first release, we will not have enough time to implement both features, but this will be one of the key features to be developed in our second release.

Lastly, we have planned out our first iteration and will be laying the foundation for our product. Therefore, the only features and user stories we will be addressing will be notifications and adding or removing tasks and events. It is imperative that we start with the absolute most basic functionalities and build upon them. These two will allow for a very basic time management tool that we will further expand and develop in our subsequent iterations.

CRC

Class: User
Responsibilities: - getUserType - getFirstName - getLastName - getEmail - getDepartment - getCourses - getUserId
Collaborates - UserType, Department, Courses, Event

Class: UserType (Enum)
Responsibilities: - types (STUDENT, TA, PROFESSOR, etc.) - getUserType
Collaborates - User

Class: Course (Enum)
Responsibilities: - getCourse
Collaborates - User

Class: Department (Enum)
Responsibilities: - types (ENGINEERING, etc.) - getDepartment
Collaborates - User

Class: Course (Enum)
Responsibilities: - getCourse
Collaborates - User

Class: BaseEvent (Abstract)
Responsibilities: - set/get time/date/deadline of event - set/get type of event - notification date threshold - set/get title of the event - set/get invitee id - set/update event location
Collaborates - User

Class: EventAction (Interface)
Responsibilities: - Create event. - Update event. - Set event. - Delete event.
Collaborates - User

Class: Appointment, **Super Class:** BaseEvent
Responsibilities: - set time/date/deadline of appointment - notification date threshold - set title of the appointment - set/get invitee id
Collaborates - User

Class: Task, **Super Class:** BaseEvent
Responsibilities: - set time/date/deadline/title of task - notification date threshold - mark as completed - set/get invitee id
Collaborates - User

Class: Note, **Super Class:** BaseEvent
Responsibilities: - set time/date for note - set title of note - set/get invitee id
Collaborates - User

Class: EventInviteType (Enum)
Responsibilities: - types (INDIVIDUAL, COURSE, DEPARTMENT, etc.) - getInviteType
Collaborates - BaseEvent

Class: EventType (Enum)
Responsibilities: - types (NOTE, APPOINTMENT, etc.) - getType
Collaborates - BaseEvent, User

Class: HttpClient
Responsibilities: - Sends reminders of upcoming events/push notifications. - Interacts with Google cloud API (facilitates interaction between our server and Google server).
Collaborates

Class: MobileClient
Responsibilities: - Receives c2dm message notifications from the Google cloud. - Retrieves content from our Server.
Collaborates

Class: UserDao
Responsibilities: - Class that works with the database (user information). - Implements CRUD operations.
Collaborates - ServerActionHandlerFactory

Class: EventDao, **Implements:** EventAction
Responsibilities: - Class that works with the database (event information). - Implements CRUD operations (create, retrieve, update, delete).
Collaborates - ServerActionHandlerFactory

Class: ServerActionHandlerFactory
Responsibilities: - A factory class that creates a new instance of corresponding action hander (CreateTask, GetTask, etc).
Collaborates - ServerDao, EventDao.

Three User Stories:

1. *"As a student, Tim would like notifications for his work schedule and projects "*

2. *"Ling Fei is a student that would like an application that helps him quickly decide when he should allocate a task/events so he can plan his schedule easily"*

3. *"Hermione would like a schedule to keep track of her classes so she knows when and where her classes are"*

Scenario "Play out":

1. Tim's mobile application subscribes to c2dm cloud service to receive push notifications from the cloud. Server interacts with Google cloud by sending various messages via c2dm framework. Tim's mobile mobile application receives only those messages that he subscribed to.
2. User Class with userid number and Usertype = student number, collaborates with EventAction interface that creates/updates/deletes an event (which has basic event information like time, location etc) with BaseEvent class which collaborates with EventType and EventInviteType classes to set type of event and invitation type of event. Mobile application interacts with the server via the http client by making use of custom REST API implemented on the server. The server processes the incoming API calls, and obtains the corresponding action handler through the factory class. The server executes the action handler. Based on the selected handler, one of the following operations will be executed: create task, update task, retrieve task, delete task, create appointment, update appointment, retrieve appointment, delete appointment (same thing for Note). The execute() method creates the right instance of the BaseEvent and passes that event into the EventDAO object (which implements the EventAction interface). The EventDAO object will perform the correct action on the database. BaseEvent class interacts with EventType and EventInviteType classes to set type of event and invitation type of event.
3. When Hermione wants to create an Appointment event she has the option of filling out the address field for the appointment. In order to create the event, the mobile application interacts with the server via the http client by making use of custom REST API implemented on the server. Following a similar process as above, information is stored in the database.