

# **HandyEat**

## Group Project Report

E.C.W

SEIS 610 - 02 Software Engineering  
University of St. Thomas

## **Content**

<b>Background -Yu</b>	3
<b>Design &amp; Functionality – Yu, Will</b>	3
<b>Plan &amp; Execution Progress - Eric</b>	6
<b>Evaluation of Tools &amp; API - Will</b>	8
<b>Team Member’s tasks &amp; work log - Yu</b>	11
<b>Conclusion - Eric</b>	15

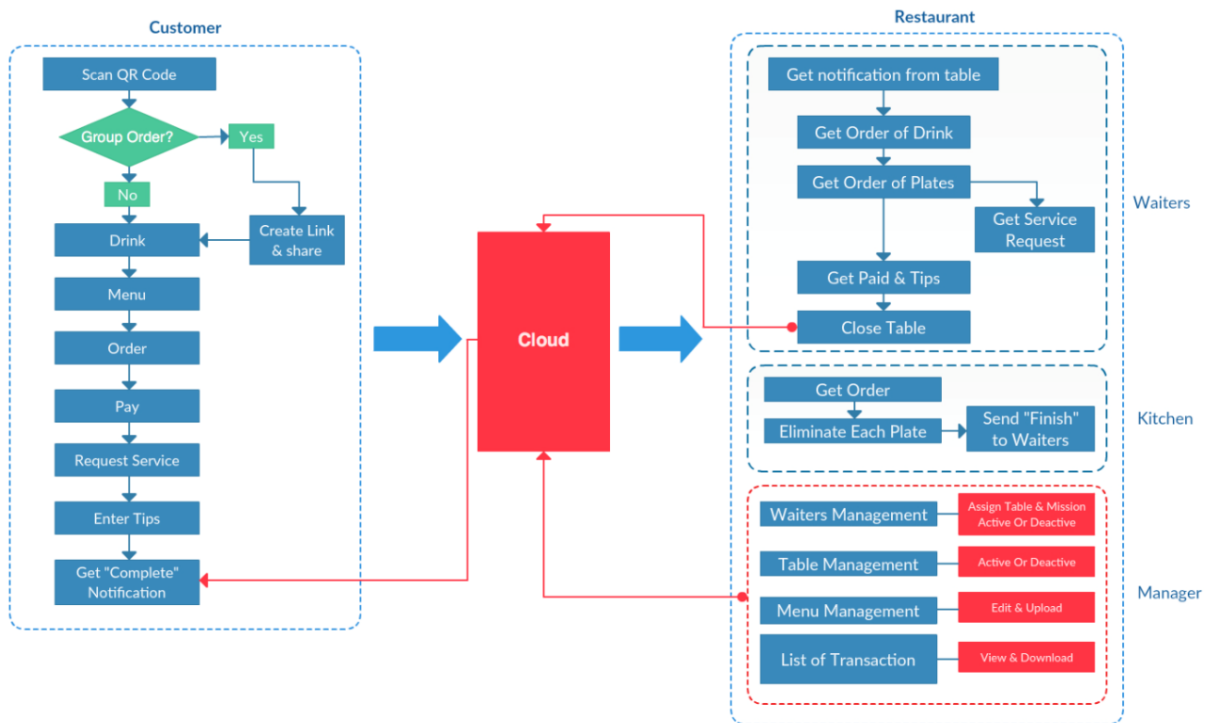
## Background - Yu

College students and office employees usually have limited time for lunch or dinner in weekdays due to their class schedule or company lunch time policy. In order to help college students and office employees to save time on dine-in service, we developed our HandyEat mobile App.

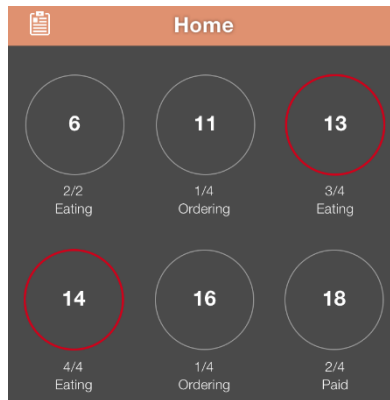
Customers are able to sit, order food and make payments themselves via our HandyEat App on their mobiles which allows customers' waiting time to be shortened. In the meantime, as both the needs of waiters and dine-in time will be decreased correspondingly, it also helps restaurants to reduce the cost of workforce by eliminating the number of waiters on duty and to improve table usage rate to improve restaurants' profits.

## Design & Functionality – Yu, Will

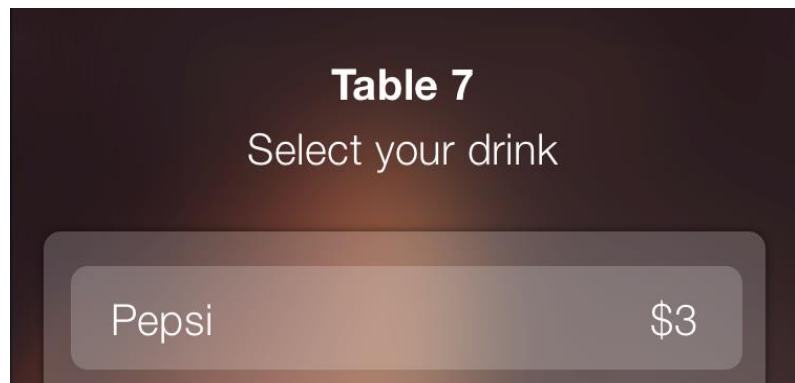
Flow Chart of HandyEat App consists of three major parts: customers-end, cloud (firebase) and restaurants-end. All data will be sent to and saved in cloud and notification will be sent to either customers-end or restaurants-end for actions needed. Below are details about how HandyEat App works from flowchart perspective.



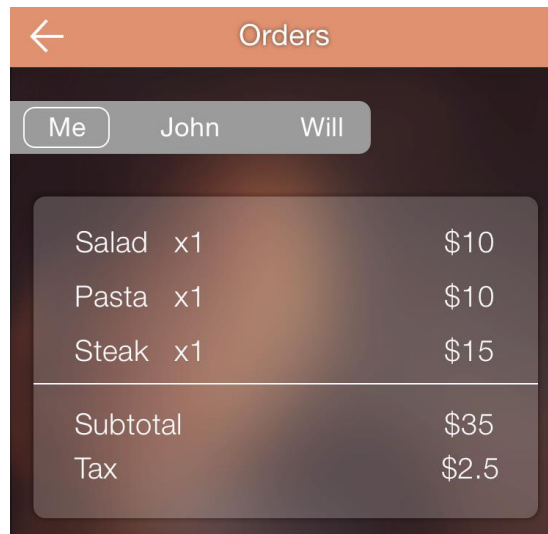
All restaurants registered in HandyEat App are assigned QR codes for all tables. Customers are able to book available tables in a specific restaurant to request dine-in service via scanning QR code. Once QR code has been scanned, data will be sent to cloud and restaurants will get updated.



Managers will activate tables and assign waiters correspondingly. There is also an option to create order link and share with others for group orders. Once everything is set, customers are able to place order.



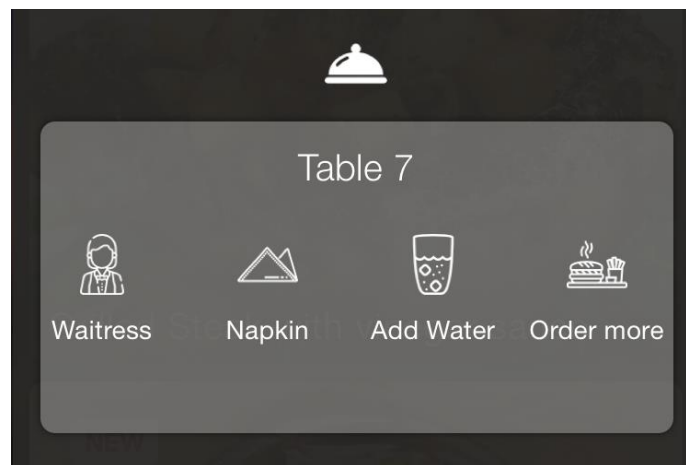
Customers order drinks first and data will be sent to cloud. Waiters will receive drink orders notification from cloud and then bring drinks and sets of tableware to customers. While waiters are preparing drinks and bringing to customers, dishes menu is displaying for customers to place food order. Customers who sitting on the same table will also allow to check friends' dishes from our app directly.



Orders		
Me	John	Will
Salad	x1	\$10
Pasta	x1	\$10
Steak	x1	\$15
Subtotal		\$35
Tax		\$2.5

Data will be sent to cloud once dishes have been placed and payments have been paid successfully. Waiters will get notification of payments successfully received and kitchen will receive food order information.

Once all dishes have been completed, Kitchen will send notification to waiters through cloud. Waiters will bring dishes to customers. During the dine-in time, customers can always request services via HandyEat App and corresponding data will be sent to cloud, and then to waiters.



Once customers finish eating, customers can grant tips and tips information will be sent to waiters via cloud. Customers will confirm if the dine-in service has been completed on their end. Once the “completion” notification has been generated to send to cloud, waiters will be notified to clean table. Waiters will send closing table notification to cloud once the clean-up is completed. Managers will then change table status from deactivated to active for further dine-in service.

In addition to waiters and tables management, managers are able to revise menu and view all transactions.

Records		
Time	Order Number	Total
11/3/2017	45243543	\$56.34
11/3/2017	34525242	\$12.00
11/3/2017	53652354	\$61.00

## Plan & Execution Progress - Eric

### Analysis & Define the APP

#### 1.1 Determining target customers

In the very beginning, we just want to focus on the college student who does not have enough dining time during the weekdays. After some research, we found that the office employees also have the same situation. We decided to focus on those two groups.

#### 1.2 Analyzing customers' needs

Based on the life experience of ourselves and our friends around us, we agreed that sometimes it takes too much time to eat in a restaurant. We all hope there is a way to save the dining time in the future.

#### 1.3 Analyzing markets scale

We will use the Twin Cities as our first marketing target as there are enough restaurants in this area and we are familiar with it. Based on the data we collected so far, restaurants area driving force in Minnesota's economy.

#### 1.4 Listing deliverables

In the early days of the project, we wanted to provide an APP that customers could spend less time in the restaurant. Later on, we found that in addition to spending less time on dine-in services by our customers, the turn-over rate in the restaurant could increase, achieving a win-win situation.

#### 1.5 Listing constraints

Setting constraints is not an easy task since the constraints directly affect the overall

project's outcome. Later, we decided to start from several general directions: schedule, budget, software restrictions.

#### 1.6 Determining project methodology

Initially, we chose phase development as our development methodology, and then we decided to use the Agile method as our project fits it better: regularly meetings, provide questions and solutions, and being more flexible.

#### 1.7 Listing tools

According to research, Android and iOS have a market share of 99% worldwide so that we will be developing on Android Studio and Xcode, and later complemented by some tools such as Sketch, ProcessOn, to help improve development.

### **Developing the APP**

#### 2.1 Designing logic & core function

The core function we designed is to interact with the needs of our customers, so at the client end, we designed: ordering, paying, and requesting service. At the restaurant end, we designed: waiters, kitchens, and managers parts.

#### 2.2 Designing process & user experience

At the beginning of the design process, only the features we want to achieve listed, and later found that we can integrate the information we already know into flowchart and DFD and use it to express our design easily.

#### 2.3 Designing logo & user interfaces

We use Sketch for Logo and interface design, interface design is based on our process design.

#### 2.4 Writing code

We did not write the program, but if the project is carried out, we will focus on the two main mobile platform: Android, and iOS.

#### 2.5 Testing and debugging

Based on the above, we did not write the program, so there is no way to test, but if the project is real, we will conduct black box and white box tests to ensure that our APP works well.

### **Delivery the APP**

We did not write the program, but we were confident we were able to deliver the perfect APP experience on time, based on the completion rate of our own set of tasks. We will also use Google Play and App Store as our main publishing platform.

### **Maintenance the APP**

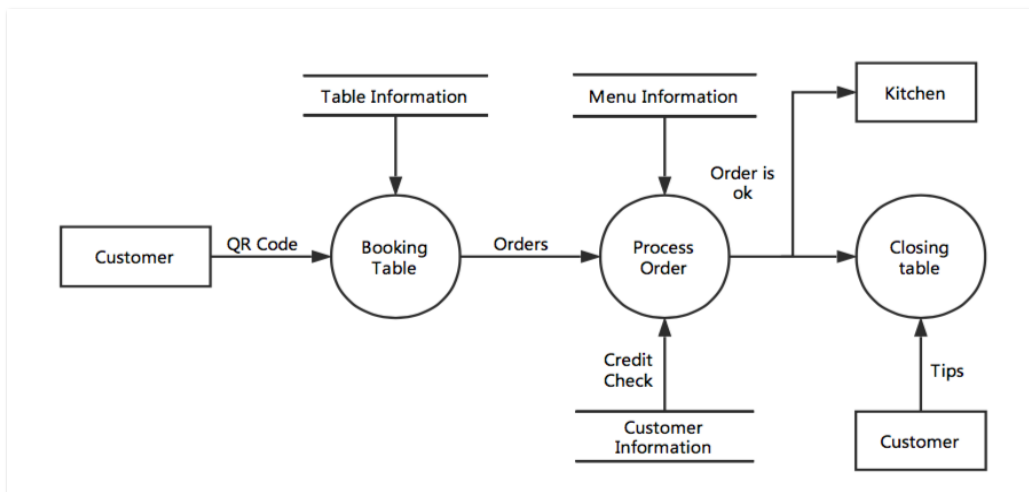
Maintaining an APP can provide a good user experience, we will use scrum method for maintenance. When we collect feedback from users, we will organize a meeting, and then release newer versions with the bug fixed.

## Evaluation of Tools & API – Will

### ProcessOn

When making software products, we often need to make visuals to go along with other materials we present. A simple flowchart can help people explain to team members or customers the software's structure. We planned to use ProcessOn as our flowchart maker, not only because the final chart is beautiful and clean, but also it is very easy to use and learn.

ProcessOn comes with tons of shapes and templates. They are sample versions of the visuals we are likely going to create. They feature drag-and-drop functionality that makes it easy to rearrange the shapes and connectors on the canvas. And because it's online, there's no need to worry about whether all the collaborators are running macOS or Windows. ProcessOn simply runs in the browser.



### GitHub

We have been using GitHub all the time during this project for documentation and communication, because of these three features – fork, pull request and merge.

GitHub is a Git repository hosting service, but it adds many of its own features. While Git is a command line tool, GitHub provides a Web-based graphical interface. It also provides access control and several collaboration features, such as a wikis and basic task management tools for every project.

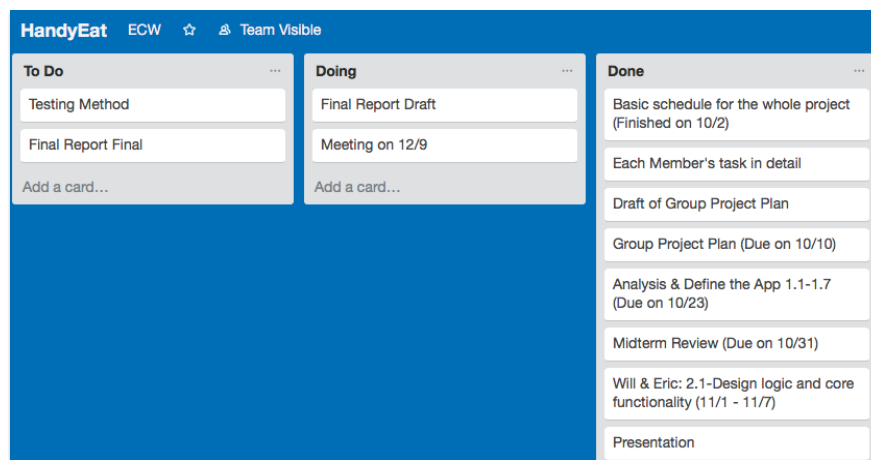


The flagship functionality of GitHub is “forking” – copying our project repository from one team leader’s account to another. This enables our team members to take a project that they don’t have write access to and modify it under your own account. If anyone makes changes and like to share, they can send a notification called a “pull request” to the team leader. Team leader can then, with a click of a button, merge the changes found in that team member’s repository with the original repository.

wsl100624 Final Report To-Do		Latest commit d0db216 3 days ago
Final Report	Update design and functionality	3 days ago
Meeting Log	First Meeting Log	20 days ago
Midterm Review	Merge branch 'master' of https://github.com/wsl100624/610Project-Hand...	27 days ago
Milestone1 - Analysis and Define	Step 1-3	16 days ago
Milestone2 - Developing	Add restaurant UI	3 days ago
README.md	Final Report To-Do	3 days ago

## Trello

Trello is our main project management tool. It is a collaboration software that helps members to discuss a project in real-time with very good drag-and-drop capabilities.. It is an easy-to-use web application that were designed to facilitate project delivery and make it more efficient.



While using Trello, we can set more realistic milestones and deadlines, tie up and automate interdependent assignments, and keep all of other team mates focused on each milestone of this project. It's a reasonably lightweight, flexible, and focused on collaboration tools.

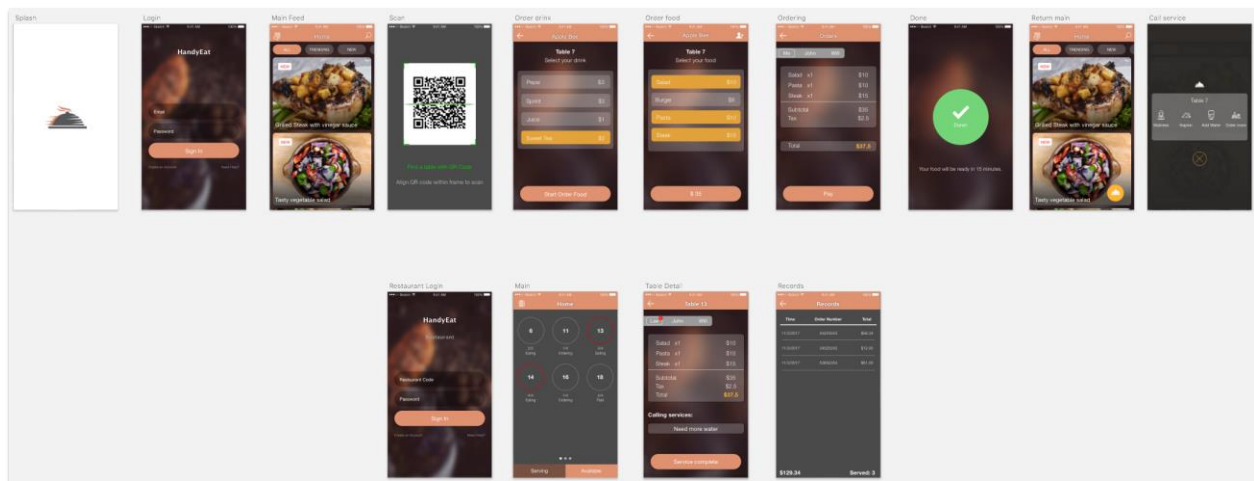
Every card on a Trello board has to be in a list, so before making any cards, we set up the workflow for our board as To Do, Doing and Done. The best thing about cards is that they allow us to talk about the task at hand. We can mention any teammates in a comment by typing “@” and their username. Then all pertinent communication stays with the task, and anybody can pick up the work from there.

## Sketch

Sketch is a design tool entirely vector-based and focused on user interface design. It's easy to pick up and costs a fraction of the price of Photoshop. And because it comes with 7 days free trial, I can finish our app's UI and logo design without any cost.

Because of its simplicity, anyone with little to no training can learn Sketch. On top of that, it's perfect for today's need for designing for multiple devices. As a result, we found Sketch far more approachable because it doesn't get in the way of producing simple designs.

Sketch comes preloaded with a number of user interface templates for iOS. The iOS and Material Design templates are particularly comprehensive. They'll be a great starting point for any designer, beginners and experts alike.



## Firestore

We planned to use Firestore as our backend server, which also acts as our datastore. The biggest features on Firestore is Realtime Database. It's a database that lets you store and sync data between your users in realtime.

We found this feature is very useful for our app because restaurant-end may change food's picture or price frequently. Based on the size of these images, we really need a database that can change and store data instantly. The connection between app and Firestore is WebSocket, instead of normal HTTP calls. According to research, WebSockets are much faster than HTTP. Developer doesn't have to make individual WebSocket calls, all of our data syncs automatically through a single WebSocket as fast as our network can carry it.

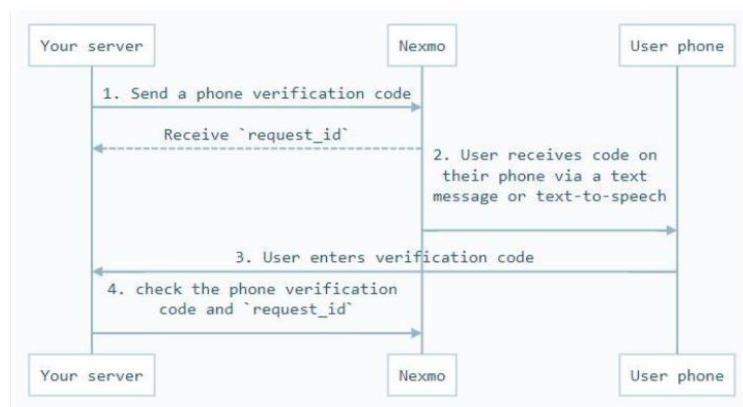
## Stripe

Stripe is a credit card processing system which enables you to receive payments from your customers, without the need to set up a Merchant account with your bank.

We decided to integrated Stripe in our app is because it's user-friendly API. The documentation is clear and concise. If we sign up an account, the documentation is customized to our profile. Because different account come with different tokens. So, we can literally copy and paste any code in directly to IDE and then see the result instantly. Additionally, Stripe supports several programming languages. These features will increase the efficiency of our app's development process.

## Nexmo

We plan to use Nexmo on a daily basis to send Promo and Transactional SMS to our Client's customers. When user create an account, or sending invitation to friends, we will also need Nexmo to generate verification code and send text messages. The real time APIs enable a seamless connectivity between our systems and Nexmo.



## Team members' tasks and work log – Yu

Generally, we are meeting 15 minutes in advance prior to class on every Tuesday to touch base with every team member to make sure everyone is good with their tasks. Below are 7 major meetings we have been held. It includes each members' task and work log. All three members (Eric Teng, Will Wang – team leader and Yu Chen) have attended all meetings.

Date	10/2 - 10/10	10/11 - 10/16	10/17 - 10/24
Yu	customers + markets analysis & possible deliverables	constraints & developing method & tools	Slack Time
Will			
Eric			
Total	22 Days - 3 Weeks		

## **Minutes for first meeting**

### **Call to Order**

First meeting of group project was held on 10/2/2017 at Anderson Student Center. It began at 6pm and finished at 8pm.

### **Agenda Items**

1. Making draft calendar plan for group project.
2. Assigning tasks to each group member with deadline dates.
3. Helping each other to understand actions need to be taken in the following weeks. Making sure everyone is on the same page.

### **Action Items**

1. Doing research and collecting data – getting more info for project – Yu
2. Creating project on GitHub - Will
3. Determining targeting users, analyzing users' needs, completing marketing scale analysis, and listing potential deliverables by 10/9/2017 - Yu
4. Completing group project plan by 5pm 10/9/2017. Reviewing & Revising group project plan by 3pm on 10/10/2017 - Team
5. Listing constraints, determining developing method and listing all potential tools by 10/16/2017 – Eric & Will
6. Holding meetings online as often as needed. Group members touch base on a regular basis for 15 minutes before each class.

## **Minutes for second meeting**

### **Call to Order**

Second meeting of group project was held on 10/28/2017 at O'Shaughnessy Science Hall. It began at 1:30 pm and finished at 4:00 pm.

### **Agenda Items**

1. Reviewing each member's task progress – ahead or behind? Still on the correct track/direction?
2. Discussing each member's task writing draft – completing all parts need to be writing up? Items need to be added, deleted or edited?
3. Gaining better understanding of our developing App via each members' completed task

### **Action Items**

1. Completing every tasks' midterm review – submitting via GitHub no later than 5pm CST on Monday, 10/30/2017 – Team
2. Completing functionality – API by 3pm on 11/7/2017 – Will & Eric
3. Holding another quick meeting on Monday night, 10/30/2017 to make a final review for midterm review report. Make sure all team members agree what looks like in the final publishing report. Exact meeting time – TBD.

Date	11/1 - 11/7	11/8 - 11/14	11/15 - 11/21
Yu		process - flowchart & DFD	
Will	functionality - API		logo & UI
Eric			testing
Total	22 Days - 3 Weeks		

### Minutes for third meeting

#### Call to Order

Third meeting of group project was held on 11/12/2017 in O'Shaughnessy – Frey Library Center. It began at 3:30 pm and finished at 5:00 pm.

#### Agenda Items

1. Reviewing each member's task progress – ahead or behind? Still on the correct track/direction? – Overall, a bit behind but it's still on control. Plan date got updated correspondingly.
2. Reviewing and discussing API and Functionality. Items need to be added, deleted or revised?
3. Gaining better understanding of our developing App via each members' completed task
4. Helping each other to understand actions need to be taken in the following week. Making sure everyone is on the same page.

#### Action Items

1. Completing Flow Chart, Logo and UI based on functionality & API by 3pm on 11/19/2017 – Yu & Will
2. Meeting at O'Shaughnessy – Frey Library Center at 3pm on 11/19/2017 for fourth meeting to review Flow Chart, Logo and UI
3. Meeting on 11/24 for first practice of presentation. Exact time and location – TBD
4. Meeting on 11/26 for final practice of presentation. Exact time and location – TBD

### Minutes for fourth meeting

#### Call to Order

Fourth meeting of group project was held on 11/19/2017 in O'Shaughnessy – Frey Library Center. It began at 3pm and finished at 4:30pm.

#### Agenda Items

1. Reviewing Flow Chart, Logo and UI. Discussing any parts need to be added, deleted or revised.
2. Revisiting presentation requirement.
3. Discussing PowerPoint idea & content.

#### Action Items

1. Completing DFD and UI – Yu & Will
2. Power Point formatting & Key points - Eric
3. PPT draft due on 11/22 – meeting online at 8pm: discussing PPT and assigning presentation tasks - Eric
4. Meeting on 11/24 for first practice of presentation. Location: 2800 Rustic Pl, Little Canada, MN, 55117. Time TBD.
5. Meeting on 11/26 for final practice of presentation. Exact time and location – TBD.

### **Minutes for fifth meeting**

#### Call to Order

Fifth meeting of group project was held on 11/24/2017 at 2800 Rustic Pl, Little Canada, MN, 55117. It began at 9:30am and finished at 11:30am.

#### Agenda Items

1. Reviewing DFD and UI
2. Reviewing PowerPoint draft (formatting & Key points)
3. Going through each member's presentation contents
4. Assigning timeline for each member's presentation to ensure the complete presentation will be within 12 mins

#### Action Items

1. Updating DFD and UI based on teammates' feedback – Yu & Will
2. Revising PPT based on DFD and UI changes - Eric
3. Meeting on 11/26 for final practice of presentation in O'shaughnessy – Frey Library Center at 3:30pm

### **Minutes for sixth meeting**

#### Call to Order

Sixth meeting of group project was held on 11/26/2017 in O'shaughnessy – Frey Library Center. It began at 3:30pm and finished at 5:00pm.

#### Agenda Items

1. Reviewing DFD and UI
2. Each team member practices their own presentation part in front of other teammates
3. Practicing presentation all together as a team

#### Action Items

1. Finalizing Presentation PowerPoint - Eric
2. Meeting on 11/28 at 4pm in classroom LL18 to do final presentation practice
3. Starting to write draft for final report - Team
4. Meeting on 12/9 to discuss final report draft. Time & Location – TBD

### **Minutes for seventh meeting**

#### Call to Order

Seventh meeting of group project was held on 12/09/2017 in O'shaughnessy – Frey Library Center. It began at 5pm and finished at 6:00pm.

#### Agenda Items

1. Reviewing Final Report Draft – each team member goes through their own parts with the rest team

#### Action Items

1. Revising each member writing parts based on teammates' feedback - Team
2. Meeting online at 8pm on 12/10/2017 to finalize final report

## **Conclusion - Eric**

In this team project, we learned some key points: time scheduling, development methods, and development tools. In the time scheduling section, after the actual execution of the tasks, we found that the longest time spent on the requirement analysis and definition was half of our total project hours. The time spent on functionality design is the second highest. The least time-consuming is about art design, such as the logo. At the selection of development methods, we chose phase development from the beginning, but after several meetings, we found that we needed agile methods that were more flexible and suitable for our application development. In the choice of tools, we looked at how to use GitHub to share documents during development. It provides each member access to all documents to make sure everyone is on the same page about our data and information. Greatly enhance the efficiency of our parallel operations.

Other things we learned in this project is how to work together to complete a project and use the personal strengths of our members to enhance the efficiency of the project. This experience is valuable. In the future, small tasks in life can also be treated as projects to take advantage of what we learn in the classroom and improve the quality and efficiency of life.