# CS 428 Final Documentation

Team: Who's Out There

TA: Sadaf Tayefeh Hosseinlou


**Team Members**

Joshua Heng - heng3

Shuotian Chen - schen79

Samuel Xiaojie Zhang - zhang369

Xiaoying Feng - feng30

Weijian Zhou - zhou86

Yanyan Zhang - zhang 250

# Table of Contents

# Description of Project

There are many tools to aid in trip planning, covering transportation, lodging and other aspects of the trip. However, there is no convenient tool for travellers to find friends and acquaintances who are staying at the destination. We developed this application as a tool for travellers to not only identify friends staying at their destination, but also for them to communicate with these friends and set up meetings. To this end, we will leverage the data available on Facebook to identify the locations of the user's friends, while use Google Maps to present this data in a clear and concise manner. Because of the universality of Facebook, it's messaging service will also be our platform of choice for communicating with the user's friends.

# Development Process

Our development process is based on the Agile Model. The agile model is based on the following principles:

- Customer satisfaction through rapid delivery of software
- Flexibility to changing requirements
- Short development cycle together with frequent deliverables
- Collaboration between business people and developers
- Providing motivated developers with the necessary tools and support to allow them to perform
- Emphasis on face-to-face meetings
- Progress is measured by working software
- Comfortable pace for sustainable development
- Attention to good design and technical excellence
- Simplicity in design to minimize the amount of useless work
- Self-organizing teams
- Regular fine tuning on team dynamics to become more effective

While not all of these principles are relevant to our classroom context, we will adopt those principles which most reflect the agile development process. In our context, we will treat the professor and TAs as our customers.

The two week iterations for our project represent a short enough time span to enforce rapid development as well as frequent assessment of customer satisfaction, yet long enough for the application to have significant improvements over each previous iteration.

Weekly team meetings reinforce the emphasis on face-to-face meetings, and also encourages good design and technical excellence through code reviews. The team meetings also provide a way for the team agree on approaches to changing requirements.

User Stories and Use Cases provide a way to divide the application into modular components. This gives us greater flexibility in changing requirements because we can change a single user story or use case without having to significantly modify the rest of the project. It also ensures that the work we do directly contributes to the required functionality of the application. Finally, user stories and use cases also allow us to measure our progress in a concrete manner, which helps us to further improve team dynamics.

Other ways for us to ensure the quality of our software was through refactoring and testing. Although we did not pursue a strictly test driven development like in XP, we always put an emphasis on having a wide test coverage, especially in cases which involve user interaction. Towards this end, we used the QUnit testing framework for our Javascript functions, PHPUnit testing framework for our PHP functions, and Selenium testing framework for UI based testing. These frameworks allowed us to thoroughly test both our front end and back end functions.

Refactoring was performed when the code base became substantial enough and it was important to prevent build up of spaghetti code. Because development was performed without the aid of an IDE, refactoring code was also mainly done manually. This also helped us gain a deeper level of understanding of our code base and helped us write cleaner code.

We followed a pair based model for development. From our experiences with XP, we found that two people was the ideal number to work on a feature. Working in pairs allowed each individual to fill in the gaps in his/her partner's knowledge, while minimizing the overhead of working in groups. Frequently pairs also found themselves collaborating with other pairs when implementing closely related features, and it is in these scenarios where we adopted a very modular design strategy, where we agreed upon clear input and output formats to ensure that the components worked together.

In summary, our development process was heavily based on the Agile model, and the principles in the manifesto served as the guidelines for our process. The model provided us with a structured way to approach the development process so that our product would be well designed, technically solid and most important of all, fulfilled our customers' requirements.

# Requirements and Specifications

## User Stories

- Allow user to know where his Facebook friends are staying
- Allow user to visualize the locations of his friends on a map.
- Allow user to search which friends are living in the city or place that he is traveling to.
- Allow user to contact friends when he is traveling to their place of residency.
- Allow user to create a travel itinerary.
- Allow user to add meetings to their trip.
- Allow user to share status on Facebook
- Allow user to check the weather of a city he is travelling to.
- Allow user to see route direction from current location to destination

## Use Cases

- User - Look up city
- User - Filter out friends you want to meet
- User - Add meetings
- User - Look up weather
- User - Chat with friends
- User - Share itinerary with friends.
- User - Get directions to destination.
- Facebook system - Generate friend-location list
- Google maps system - Show red marker on Google maps.

# Architecture and Design

## Use Case Diagram

In the initial phase of the project, we had to define what the user was able to do with our app and the necessary external libraries and APIs we had to use to provide the functionality for the app.

Therefore, we came up with user stories to understand what a user would want from using our app. Building on top of these user stories, we came up with use cases to help us identify the functionality of the app and which are the necessary APIs we have to use and whether they were able to support or provide us with the functions we needed. Shown below is a use-case diagram which helps us visualize the interaction between the user and the main APIs we are using.

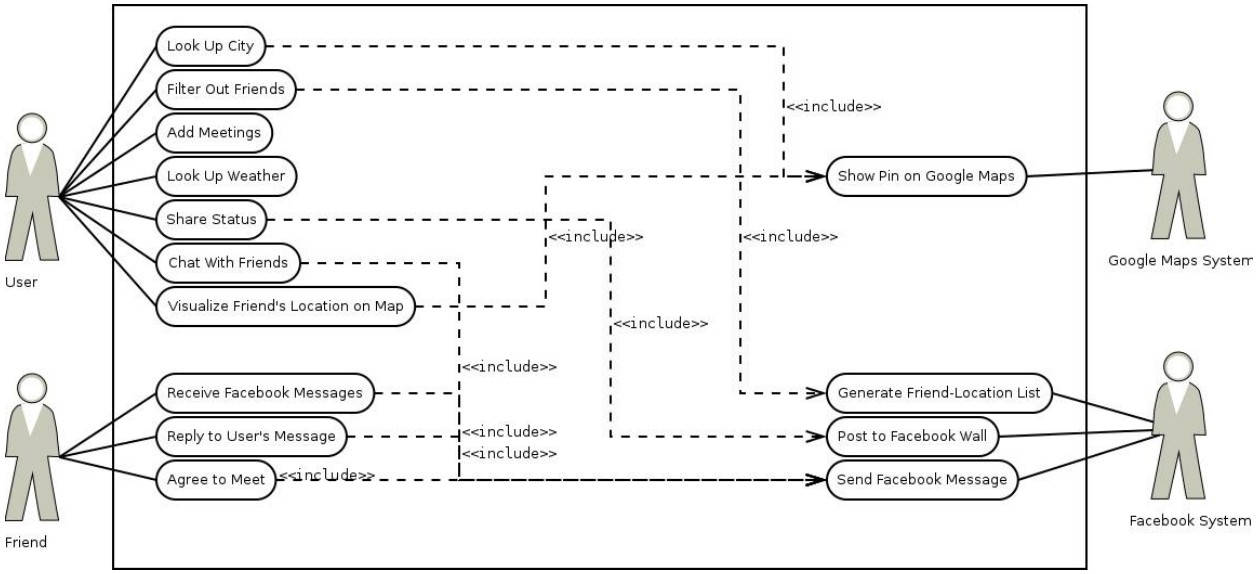


Fig. 1 - Use Case Diagram

## ER Diagram

Next we had to set up a database to store the user's information, details of an itinerary, details of a meeting.

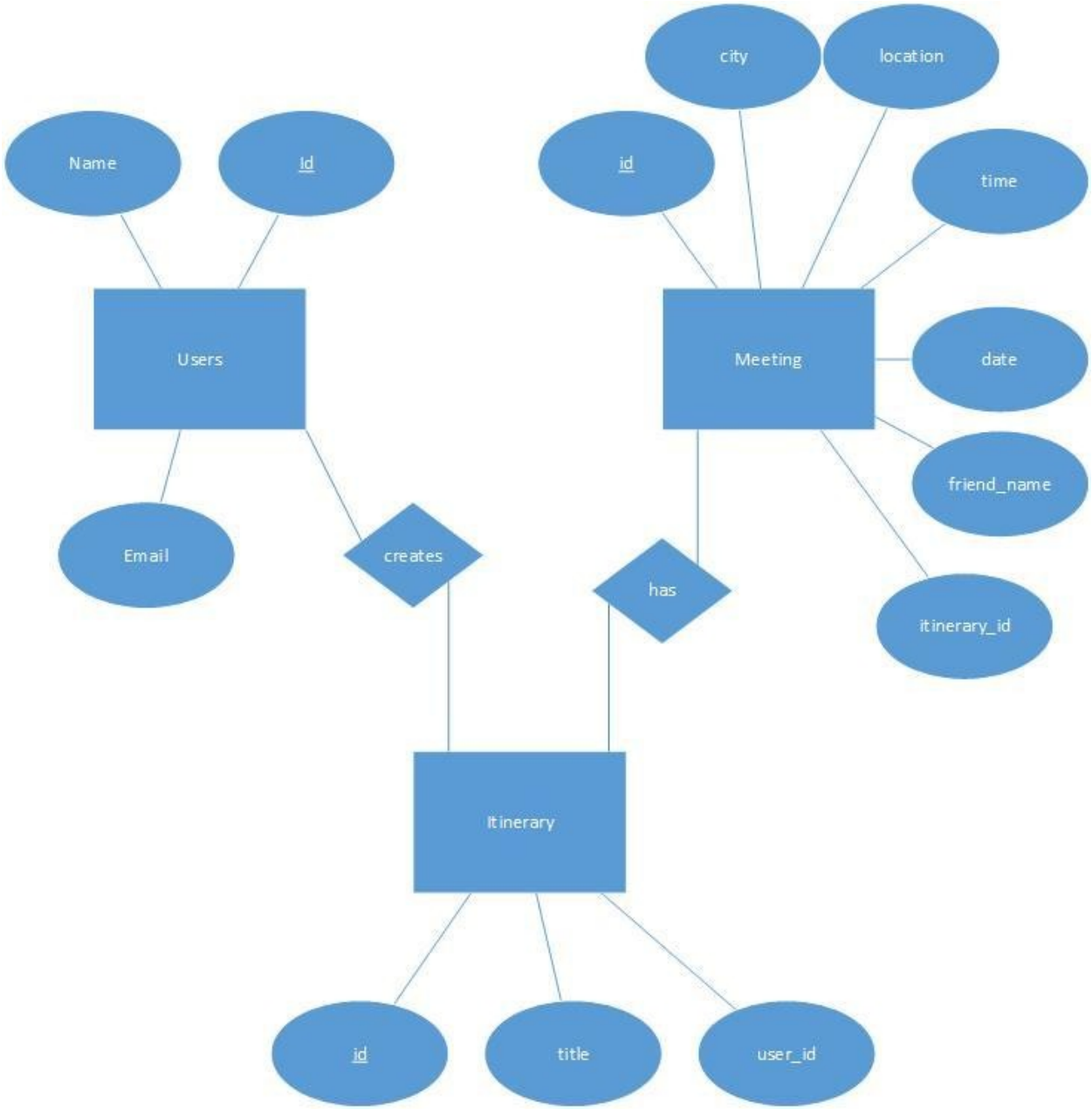


Fig. 2 - ER Diagram

`Users` table

Stores required information about a user when they first login with facebook.
Information stored:

- Facebook Id (used as a foreign key in itinerary table)
- Name
- Email

`Itinerary` table

Stores all information about an itinerary.
Information stored:

- Id
- title
- user_id

`Meetings` table

Stores all information about a meeting.
Information stored:

- Id
- city
- location
- time
- date
- friend_name
- itinerary_id

## Overall Sequence Diagram

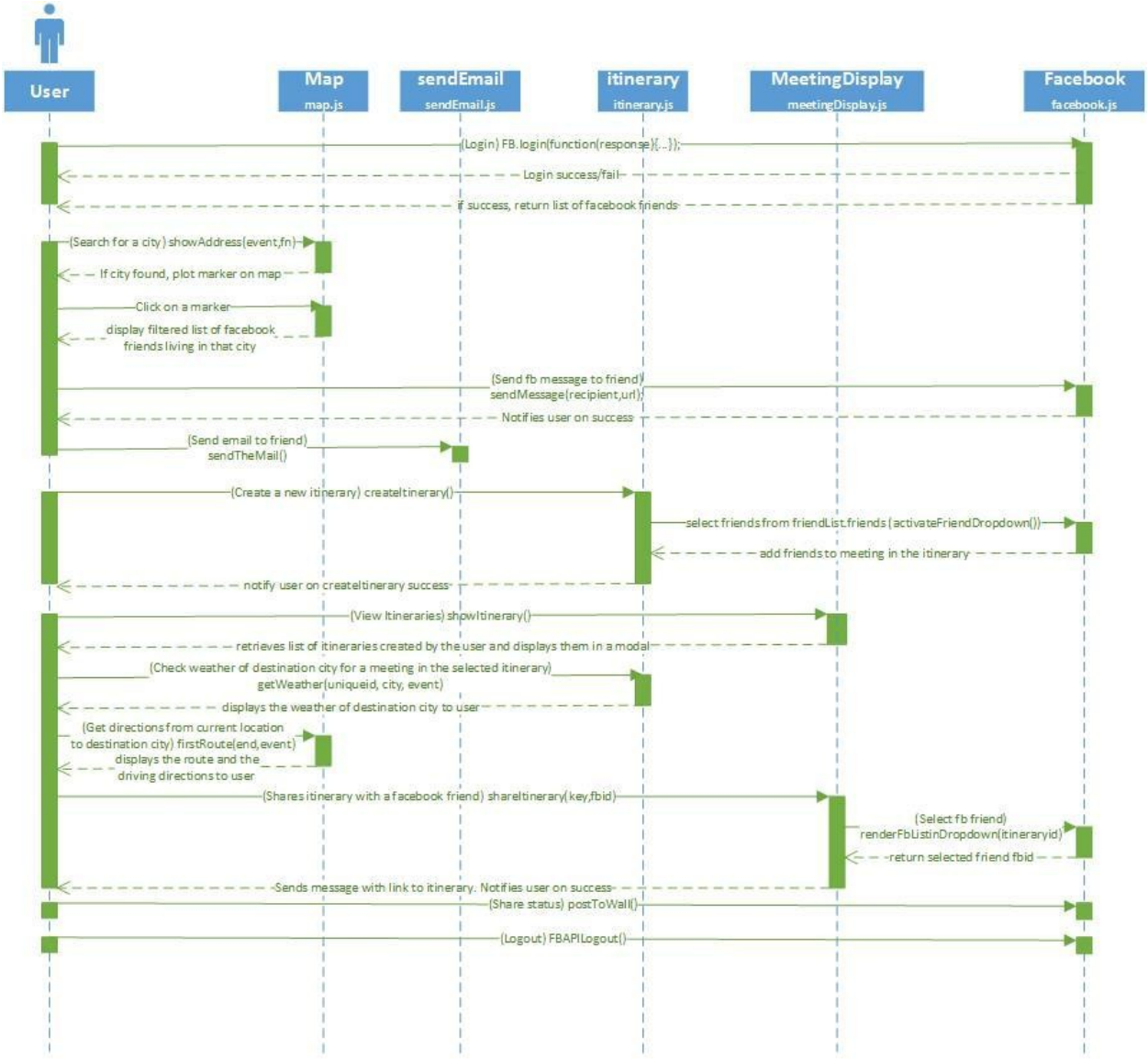


Fig. 3 - Sequence Diagram

In our project, we took an MVC approach in developing our web application. Specifically, we separated our front-end operations from the backend logic. By using an MVC framework, we separated our view(.html) files from the controllers(.js) files and the model (.php) files. This helps to minimize conflicts when we are making changes in one component as they are mostly isolated from each other. In our application, users will interact with the GUI first and these interactions will either makes a call to a function in the controller which either render or hide a view, send a post request to an API or interacting with the database via our backend PHP scripts to change/update our model. As seen in the overall sequence diagram above, we have different controllers for each unique interaction between the user and the app.

These are the operations users can perform:

- Render/hide view:
  - Clicking on a marker on the map to view a list of friends living in that city. (Here our list of friends were already pre-loaded when we logged in. All we are doing here is to filter friends living in the particular city).
  - Clicking “Create Itinerary” button to view the create itinerary modal.
- Making post requests to API
  - Check weather of a destination (Sends a post request to wunderground API)
  - Check direction from current location to destination (Sends post request to Google Maps API).
  - Sending Facebook message to a friend by using Facebook message API.
  - Sharing an itinerary with a friend via Facebook message API.
  - Sending email message to a friend by using Mandrill by mailchimp.
- Making post requests to PHP scripts on our backend:
  - Adding new user to database when user first uses the app.
  - View list of all itineraries by clicking on the “Itinerary” button on the navigation bar. It retrieves all the itineraries created by the user.
  - Creating a new itinerary.

## Class Diagrams

The two main usage of the app includes

- Locating friends living in a particular city, getting in touch with them.
- Creating an itinerary where we can have multiple meetings for an itinerary. For each meeting, we are able to get directions and weather of the destination city.

Locating friends in a particular city, getting in touch with them class diagram:

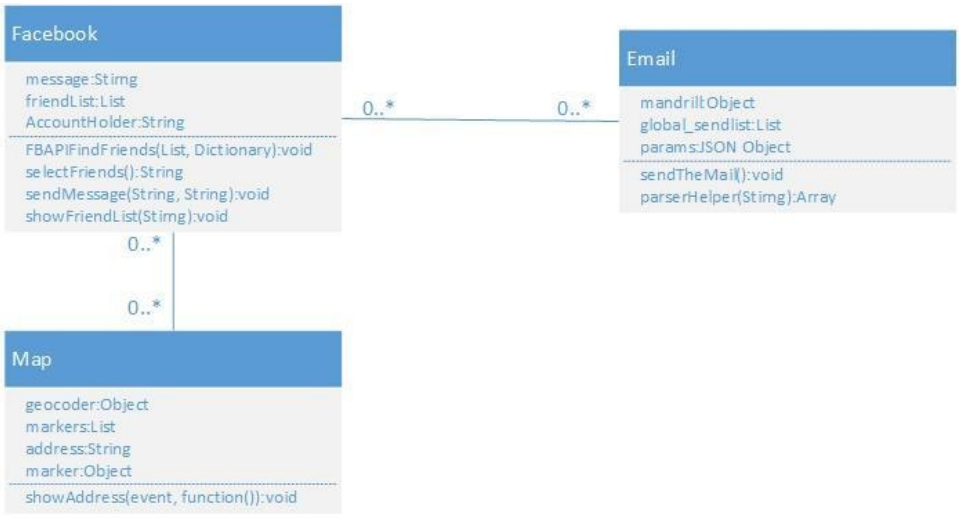


Fig. 4 - Class Diagram 1

Creating an itinerary, viewing itineraries, checking weather and getting directions to destination city class diagram:

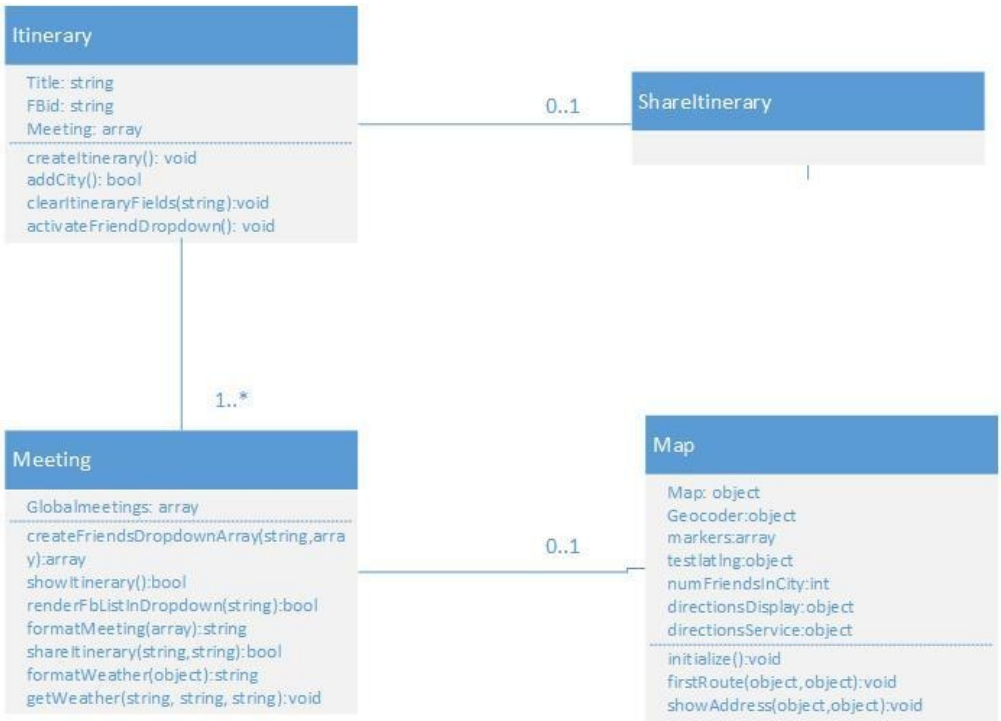


Fig. 5 - Class Diagram 2

## Main Classes

### Front-End Classes

Javascript does not have a strong class based system, and since most of our project is done with Javascript, we decided to create a separate Javascript file for each distinct component in our project and treat these as classes.

#### `facebook.js`

Handles all the Facebook functionality in the application, including Facebook API calls as well as processing the data that is returned from these API calls.

Functions:

- Facebook Login
- Facebook Logout
- Get User's Facebook ID
- Find User's friends and their locations
- Display a list of friends at a particular location
- Handling selection of friends
- Send Facebook Message
- Post status to Facebook Wall

#### `itinerary.js`

Handles all itinerary creation functions to allow the user to create new itineraries. Each itinerary is a list of cities that keeps track of the meetings that a user has in each city.

Functions:

- Create Itinerary
- Add City to Itinerary

#### `map.js`

Handles all mapping functionality including Google Maps API calls including displaying the map, geocoding cities into logitude and latitude representations, dropping pins and routing users.

Functions:

- Creating a new Google Maps display
- Dropping a pin on a specific address
- Routing the user from the current location to his/her destination

### `meetingdisplay.js`

Handles all displaying itinerary functions that allow the user to perform various functions such as viewing previously created itineraries, sharing the itineraries on Facebook and checking the weather at the different cities.

Functions:

- Show itinerary
- Display meetings
- Share itinerary
- View weather

## Back-End Classes

Since we only have a 3 types of interaction with our database, we created 3 different php endpoints on the server. In each endpoint, we make a connection to the server via require('connect.php'); . By extracting the connection code into a separate script, we can avoid repeating connection code in all 3 php endpoints. This also makes it easy to test each function. Each php endpoint then calls a particular helper function in functions.php . By putting all the different functions in a single class, in the future if we needed to use a function in a different php endpoint, this separation of functions from endpoints would make it possible.

### `connect.php`

Makes a new connection to sql database.

`functions.php`

Stores the different functions which are executed by the php endpoints

Functions:

- Adding a new user. Function called: **`addNewUser`**
- Adding a new itinerary. Function called: **`addItinerary`**
- Getting all itineraries created by a user. Function called: **`getItinerary`**

### `addNewItinerary.php`

PHP endpoint to add a new itinerary into the database. It calls **`addItinerary`** to execute the insertion of a new itinerary and meeting into the database.

### `addNewUserToDb.php`

PHP endpoint to add a new user into the database. It calls **`addNewUser`** to execute the insertion of a new user into the database.

### `getItinerary.php`

PHP endpoint to get all the itineraries from the database. It calls **`getItinerary`** to get all the itineraries and their meetings from the database.

# FUTURE PLANS

While the core functionality of our application is mostly completed, there are still many new feature that could potentially improve the user experience. Here's a short list of the possible features that could have been added if we had more time with the project:

- Info page to guide first time users
- Predefined template to share itinerary to Facebook wall (instead of asking the user to define their own status message)
- Uploading of photos to Facebook
- Complete trip routing (multiple cities)

## Personal Reflections - Shuotian

As a relatively newcomer to web development, this project was challenging, yet a valuable learning experience. Having a self-directed project meant greater flexibility, but it also placed a greater responsibility on us to not only come up with a meaningful project, but also to be motivated enough to see this project to the end and not take the easy path, such as changing requirements when we meet a particularly difficult problem. Our Agile Development Process provided the basic framework to coordinate team dynamics and produce deliverables. As compared to XP in CS427, I personally feel that Agile provides more flexibility in the process. While XP sought to control every aspect of the development process, such as enforcing pair programming and test driven development, Agile has fewer of such strict activities. Instead, Agile encourages good software development through emphasis on customer satisfaction, flexibility to changing requirements, as well as a rapid development process. How the developers implement these principles is almost left entirely to the discretion of the developers. Working in a team of 6 was not easy, and we could have easily have ended up in integration hell if we did not maintain a fairly modular system as well as enforce good version control principles using GitHub. Nevertheless, we were motivated to build a good app and for most part, I feel that we have succeeded and abided by the principles of Agile development.

## Personal Reflections - Yanyan

This project brought us many challenges as well as many valuable experiences. Even though this is a self-paced project, we did not change the requirements easier because we encountered difficulties during implementation. Yet we remained self-motivated and tried our best to accomplish the goals. Each team member was responsible for their own task. This guaranteed that our project is following the schedule. Apart from that, whenever there is any problem with the implementation, all team members would contribute to solving the problem. Within the development cycle of this project, I had great experience of teamwork. Hence, I regard this project a valuable experience.

## Personal Reflections - Xiaoying

This senior project is really interesting and challenge, because it requires strong teamwork during the processing. It also gives us a taste of a real life programing project. Working in a team of six is much difficult than working individually. We spend some effort on communicating with each other, but we definitely benefit from cooperating with other teammates. Moreover, we are involved in Agile process rather than XP taught in CS427 for this project. I really appreciate Agile development which is more

flexible and allow us to change the requirement anytime. Although the project is open ended, excellent users experience is the ultimate goal to satisfied. Hence, we are glad to modify our product to achieve it.

## Personal Reflections - Weijian

This semester we have implemented a very interesting project.  During which my pair and me followed the strict pair programming process. Most of the time we worked together and committed our job with each other's name. In addition, I believe we also obey other Agile process in developing this project. In contrast to the XP process we followed last semester, Agile process provides us a more flexible experience, which I consider is helpful for us to develop this project. When working this project, everyone has been assigned jobs and devote themselves. We did meet some challenges however this process made us capable of solving these kind of difficulties in the future.

## Personal Reflections - Xiaojie

As a student major in Computer Engineering, I found CS 429 quite different from what I did before for the project of my ECE courses. First, the project is very open ended. There is almost no requirement for the topic of the project.The Agile software engineering process make the whole developing process smooth. Although, we also make several changes from the original project proposal, modified the potential risk and challenges, with the help of Agile process, we are able to adapt the “runtime” changes into our project very well. Also, our team works pretty well during all the semester, all the members share knowledge with each other, and always willing to help.  The experience of this course is very helpful for my future career. I know a lot of companies use Agile process in their product developing. With our hard work in CS 429, I believe we can easily get used to the working environment in the real industrial world.

## Personal Reflections - Joshua

I have been involved in a few software development projects between class and outside of school and I have experienced different forms of development process that are similar to the Agile development process. From working on a 3 month long mobile app project to a semester long class project in 428, i can appreciate the flexibility of the Agile development process. Many times along the way in this project, we realized that there were limitations to certain APIs and the accommodation of certain browsers. Therefore, we had to change the way we implemented certain features or completely change them. This is way more different in XP since we had more requirements to follow for every iteration. Another thing in Agile that I feel is one of the key is user feedback. Even though we may have a working product, we do not know what the user likes or dislike about the app. The only way to find out is to do user testing and make changes to it to make the app more user friendly to the user. I feel that we have followed the principles of the Agile development process and I am sure the other team members have had some valuable experience through this project which they could apply in future endeavors.