Gabriela Carrillo
Allison Kaufman
Jason Lu
Carole Sung

CS411 Final Project Deliverables

In this word doc, we have included all of our deliverables from the semester. There are a few changes included since our project changed over the course of the semester. The changes are discussed in blue text at the beginning of every deliverable in order to show the "story of our project" and how it changed over time. The original deliverable follows the blue text in some cases, but in others (for example deliverable 2: User Stories) we made changes over the course of the semester and so they are all displayed together.

And a link to our github repo where all of the code is located and some diagrams as well:

https://github.com/gpcarr/411-project-master

Tables of Contents

Deliverable 1:

Our first project idea was approved and we ended up implementing a very close version of it to the version discussed here in Deliverable 1. The only difference is that we did not include a favorited page, instead we redirected the user to the Yelp! page of their desired location.

Assignment 1: Pitch

The first project idea is an application that helps people to better plan their vacations and schedule around weather issues that may arise during travel. We intend to utilize the GoogleMaps API and the Weather Channel API in order to judge the weather conditions in the user's travel destination and offer suggestions accordingly for indoor and outdoor activities. For example, if it is going to rain all day in the area, the app might display options for local movie theatres, or if it is going to be sunny it might likewise offer up beaches nearby. We would also aim to allow the user to favorite certain destinations and save them for later in order to allow them to create a travel plan that is customized to their interests and the ideal activities for the weather. These customized plans will be stored in a database that will hold information collectively making up the user's profile. The app would also incorporate reviews, perhaps from the Yelp API, so that the user can easily decide which destinations are for them. In addition, the user will be able to log in through either Facebook or Twitter. Some of the states the application will have are a login page where the user can use third party authentication, an initial profile page where the user can enter their travel destination and activity interests, a display page for the application's suggested activities, and a favorited page that shows the user their saved destinations so far.

The second project idea is an application that makes searching for the perfect movie easier for the user. We would utilize the Facebook API to access the Facebook likes of the user and the Rotten Tomatoes API to search through movies using the likes as key words. The application would then return movie suggestions, along with a description and rating of the movies. For instance, if a user has liked a page about dogs on Facebook, the application might return a list of dog movies, preferably sorted by their rating. Also, we intend to allow the user to specify a genre of choice and to input additional likes, such as actors or directors, which they may not have liked previously on Facebook. These additional likes can be stored in a database that will be a part of the user's profile. We would also look to find an API that gives a list of the places to watch the suggested movies, such as Netflix or Amazon, to make it easier for the user to locate a movie they find interesting. The user could use Facebook or Twitter to log into the application. Some examples of states in the application are the initial login page, where the user can use third party authentication, a profile page where the user can input any additional likes or preferences they wish to be included in the movie search, a display page for the application's suggested movies with descriptions and ratings, and a favorited page that displays the user's saved movies.

CS 411 Team Assignment 2 – User Stories (Deliverable 2)

Application: Travel Itinerary Application

We started with just the first four user stories originally. We added the fifth one half way through the semester after thinking that the other user stories did not really do a good job of describing our project. Then, once we started implementing actual code we added the sixth and final user story. The sixth user story is the best description of what our project does and was what we were mainly trying to follow when actually implementing the project, trying to add in as many features, or at least plan for features, from all six user stories.

First User Story: Location Serialization and Notification

"For the home page, I want to be able to login and have my request successfully processed by the third-party Kerberos authentication, or try again with Captcha if denied. I want to then be able to enter a location of interest, the current weather forecast, any special accommodations my party and I may have, and other necessary criteria. The application will serialize my input / query into a log, which will probably read something like this:"

Name: John Smith
Address: 123 Easy Street, New Orleans, LA
Place of Interest: Paris, France
Current Weather of Paris: 45 degrees Fahrenheit
Weather forecast for [user-input] days: sunny, with a chance of snow near Marlon Rouge and Tour De Eiffel
Clothes to bring: rain-jacket, boots, and hoodie
Special notice: none

"Then, after adding in my information, I would like to be able to figure out, from the traveling information given, exactly how much time I have to prepare for the travel, and the application might display some information about any plane delays (probably via an Airbnb or American Airlines database), any civil disputes (in case of problems in the destination), and most importantly, lodging and fine dining, as Paris is considered the 'City of Lights'. Finally, I would like to be able to log out of the application, and then continue to booking my flight."

"In case the application fails to display timely information, I expect it to route to a website of interest, such as the Google Maps API or Weather Channel for weather preparation and other necessity for travel, as a tour for the Louvre would be good for the rainy day scenario."

Secondary User Story: Asking Google Maps the best place for fine dining

"I am traveling (from Houston) to Boston University for a college tour with my family. We tried the original google mapping apps, but the scaling of the map is far too small to read on my iPhone 7. So, we decided to fire up an iPad, and give the travel itinerary app a whirl. I tried to also ask Siri, but all she gave me was gibberish and nonsense. So, the app takes in my input, 'Fine Dining at Boston University', and it displays me the following, as a tabular template made from a database back-end source (maybe mongoDB?):"

Location	Peak Hours	Meal of The Day	Pricing
George Sherman Union	12pm – 5 pm	Basho Trio Special	$4.00-$25.00
Nudpob	12pm – 3 pm	Pad Thai	$6.00-$21.00
Rize Late Nite	9pm – 11 am	Miso Salmon	$4.00-$15.00
Beijing Cafe	12pm-4pm	Pork Bone Soup	$6.50 - $18.50
Raising Cane's	9pm-12am	Terrier Pack	$4.50 - $10.30

Disclaimer: the foods and pricing labeled here are somewhat fictitious, only for user story purposes

"In the event that the Travel Itinerary App does not have bring the best BU dining, I sure hope it has some hyperlinks to the Warren Towers, Bay State, and West Campus dining halls. I want to make orientation great for both my family and I! Feeling hungry for some good user inputs, strong recommendations, and the most optimal way to me to splurge my way into BU without putting all of my Convenience Points to zero."

Third User Story: Sleep and usage of Google Map Api for Affordable Lodging

"I'm tired. It's been a long day walking around in Los Angeles. I take the train back home to quiet Oakland. However, I first long to watch the Golden State Warriors game on ESPN tonight, but there is a traffic delay on the Golden Gate Bridge. I do expect the travel itinerary application to use the Google Map API to route me to the nearest motels and lodging, with the most reasonable prices. Don't want to fall asleep when the Warriors go lights out in Oracle. In case the application doesn't get me what I want, that's OK, I can have a hyperlink to Hotels.com, and it will choose for me the best lodging places. Good night."

Fourth User Story: Driving and Shortest Route to Destination from Source

"I am so excited to start my internship with Fidelity in Toronto! Yet, I am extremely confused at the Rainbow Bridge, where Customs are holding up everybody before they cross the border via Niagara Falls. I sure hope the app can link to Yahoo Weather and News or at least a working implementation of Google API, and tell me where is the nearest exit to Toronto. Don't want to arrive late to work on my first day. It is just as bad as arriving to work not dressed the first day. First impressions last the longest."

Fifth user story: Everything on one convenient, easy to understand screen

"For the home page, I want to be able to login using Google and if denied, the app should ask me to try again still using Google. The app will then ask me to enter my location of interest and what dates I plan on staying there. The application will serialize my input / query into a log, which will probably read something like this:

```
Name: John Smith
Place of Interest: Paris, France
Current Weather of [user-input]: 45 degrees Fahrenheit
Weather forecast for [user-input] days:
        Day one: High of 45 degrees Fahrenheit; low of 37
                - Sunny with 43% chance of rain
        Day two: High of 46 degrees Fahrenheit; low of 35
                - Clear skies with 11% chance of rain
Top rated restaurants: 58 Tour Eiffel Restaurant
Top rated tourist sites: Eiffel Tower, The Louvre
Special notice: none
```

I would like the app to display all of this information in a single screen so I can decide what to do while on my trip, what kind of weather to prepare for, and places I should visit while there. I also want to be able to favorite specific feedback from the app, such as a specific restaurant."

Sixth user story:

"I would like an app where I can input my desired location that I want to travel to and the app will tell me what the weather is like there. I would also like the app to display places I can visit while I am there based on the predicted weather. A login option should be available: either to create an account or to log in using something like Facebook or Twitter. I would like for there to be a homepage that gives options like 'Find Weather Predictions,' 'Find locations to visit,' and a login option or create an account option with a welcome message on the homepage."

Deliverable 3

CS 411 Team Assignment 3 – Analysis and Architecture Design

Our group made several transitions from our original architecture plan as we came to better understand our skillsets. We kept Python as our primary language, but decided ultimately to also use Django as our framework for the web application. We found that Django was very accessible, and, considering none of us had much previous application experience, this accessibility became a great asset. We also had originally intended to use MongoDB as our database system; however, because we decided to stick to Python and Django, we realized that using sqlite3, a built in database infrastructure in Python, would clearly be the most effective approach. In the end, Python, in addition to Django, seems to have been a strong choice, as it allowed us to explore the complexities of creating a web application while still having a somewhat more comfortable and accessible platform to fall back upon.

Diagram: 5th User Story: Original Analysis and Architecture Design

Happy Path: The user is able to successfully login using his/her Google account, then the user is able to input his/her location of interest and amount of days planning to spend there, then the app will successfully request data from the appropriate API (Depending on weather, restaurants, or tourist sites the app will request data from the Weather Channel API, Yelp's API, and Trip Advisor's API respectfully), then the API or database will send data back to the app which the app will then display in a user friendly way.

If logging in is unsuccessful then the app will redirect the user to login again, displaying a "login failed, please try again" message.

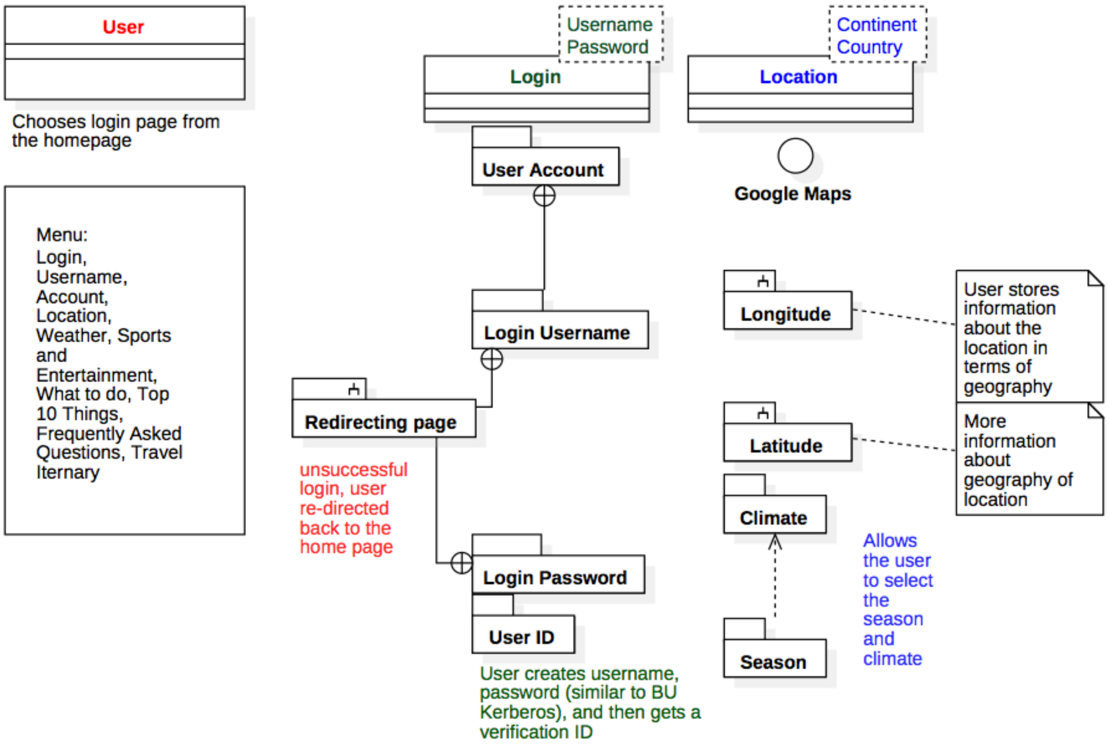
Exception 1: If the app fails to request information from an API or database, the app will display a "Data not found. Please enter a new location or try again" message.

Exception 2: If the information fails to reach the app from the API or database the same message will be displayed and the user will have to try again by entering a new location of interest or trying the same one again.

We believe that using Python as our primary front-end language and mongoDB as our primary backend framework for several reasons. The APIs that we are using to build our app support request formats in Python. Python also has a very extensive and helpful set of libraries that we think will make developing the app simpler. We chose mongoDB because we believe it is better for user data storage, and very compatible with database management.

We did not choose to go with Django as our framework for other reasons. We want to widen our skill set and we do not have experience with Mongo. If when we start actually coding we realize that we may need to use Django, we can make that change, but we believe that Django is not as good at supporting a larger user database. We decided not to go with PHP because we believe Python will be simpler to implement in and PHP is not as good as Python when it comes to object-oriented programming.

UML Diagram



Chooses login page from the homepage

Menu:
Login,
Username,
Account,
Location,
Weather, Sports and Entertainment,
What to do, Top 10 Things,
Frequently Asked Questions, Travel Iternary

unsuccessful login, user re-directed back to the home page

User creates username, password (similar to BU Kerberos), and then gets a verification ID

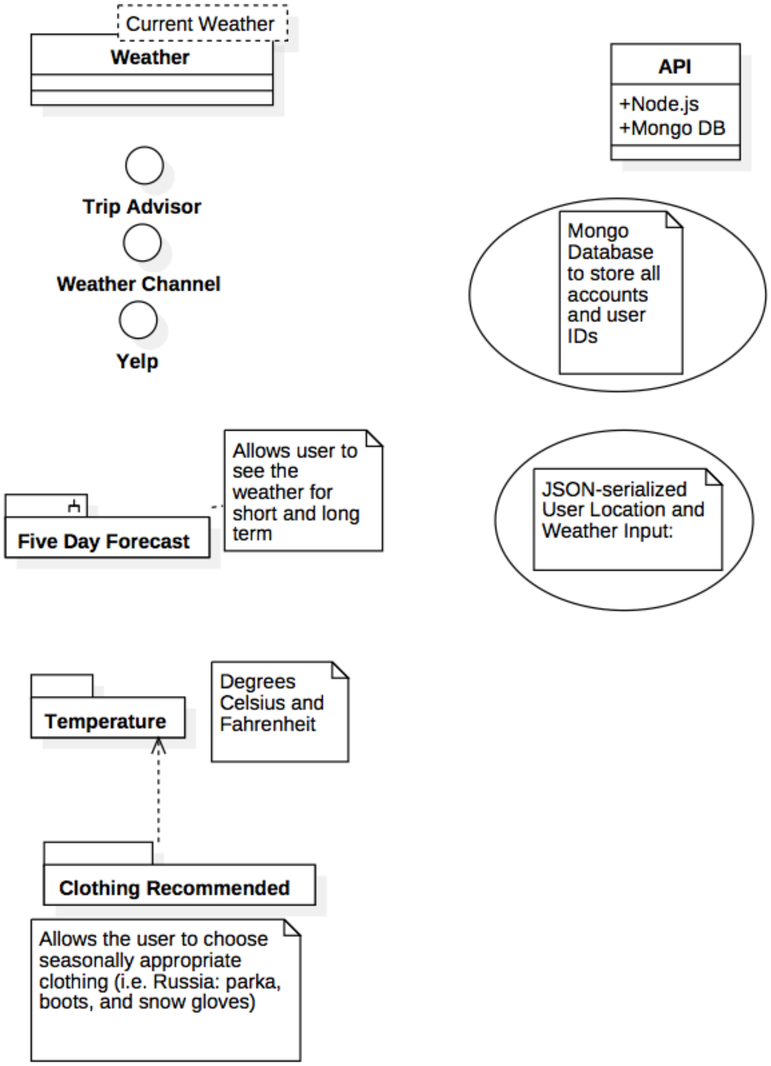
User stores information about the location in terms of geography

More information about geography of location

Allows the user to select the season and climate

General User Story Procedure: 1) User logs into the application, 2) User chooses a location based on latitude and longitude and county / continent, 3) Based on the information provided by the user, the application shows the long and short terms of weather forecast and what type of clothing / necessity the user will require, 4) The app then stores the information into the back-end (mongoDB and JSON)

UML Diagram (cont.)


Current Weather
Weather
Trip Advisor
Weather Channel
Yelp
Five Day Forecast
Allows user to see the weather for short and long term
Temperature
Degrees Celsius and Fahrenheit
Clothing Recommended
Allows the user to choose seasonally appropriate clothing (i.e. Russia: parka, boots, and snow gloves)
API
+Node.js
+Mongo DB
Mongo Database to store all accounts and user IDs
JSON-serialized User Location and Weather Input:

Deliverable 4: API Prototype

This deliverable was presented in class and the code for it can be found on our Github repo in the file called “views.py”

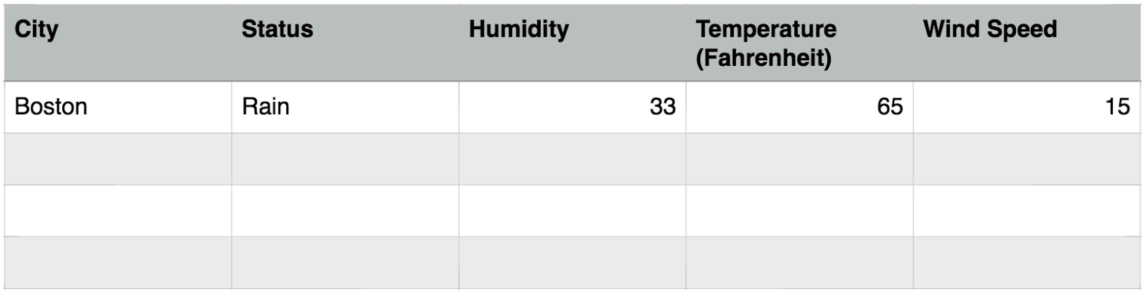
Link to Github repo:

https://github.com/gpcarr/411-project-master

Deliverable 5: Data

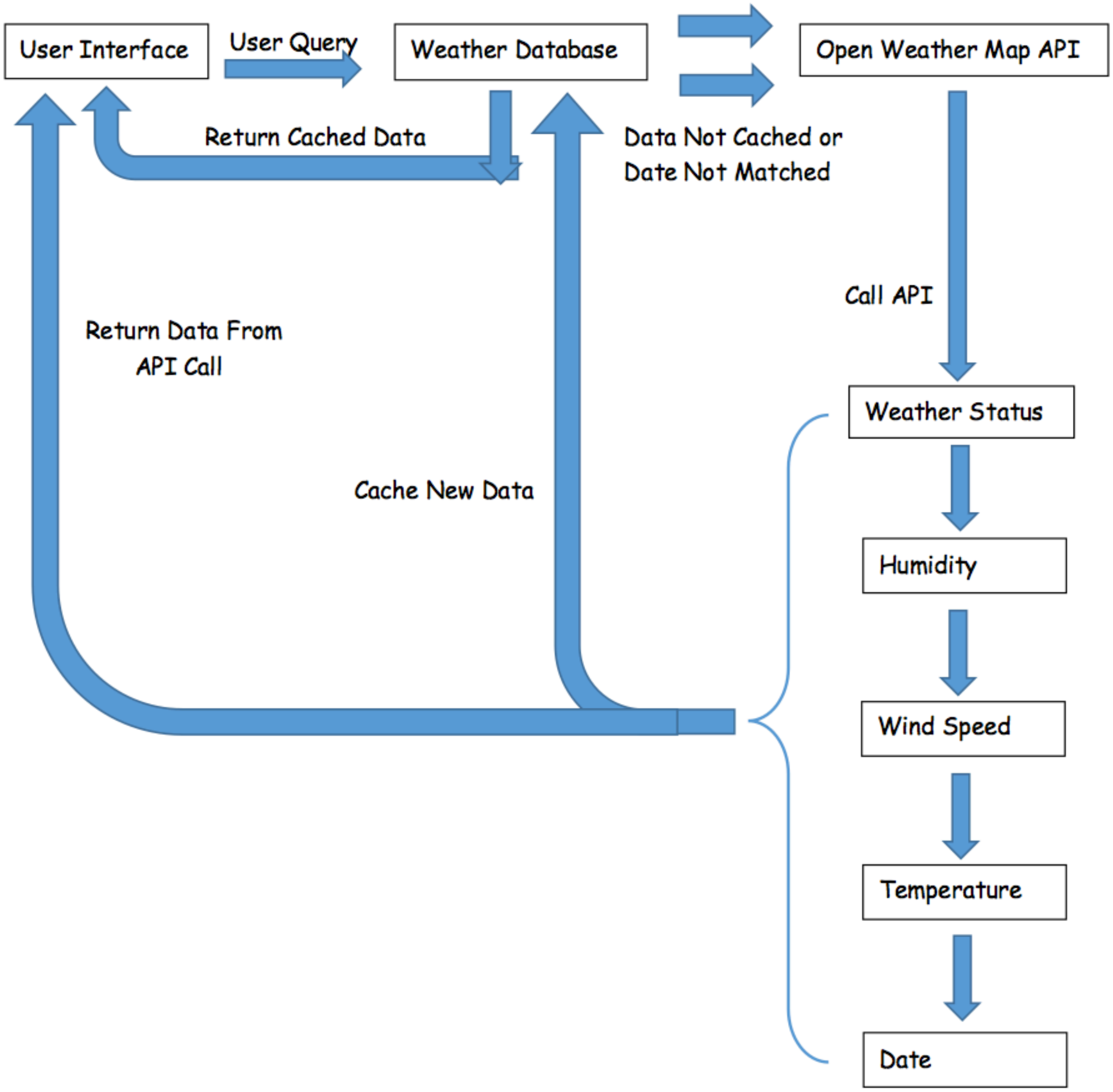
Diagrams are also found in our github repo:
https://github.com/gpcarr/411-project-master

Entity Relationship Model:



City	Status	Humidity	Temperature (Fahrenheit)	Wind Speed
Boston	Rain	33	65	15

Sequence Diagram:



User Interface

The UI files are the ones ending in .html and .css found in our Github repo. We used html and css with bootstrap for the front end.