CSP586
Project Instructions

General Instructions:

1. Identify the topic of your interest from the list below and team up with **ONE** of your classmates to form a team of **TWO**. The team will then select a team **LEADER** who will communicate with the TA all issues regarding the presentations.

2. **Team leader** is required to email the TA regarding the topic the team will work on, team members' names, and the date the team will present the presentations during class time, 5-10 minutes the length of your presentation.

3. **The deadline to notify the TA** through email about the topic that you selected, team members, and the date that you will present your project is set to be **2/15/19 by 9:00am**.

4. **There are THREE PHASES and deliverables for your project**

    1) **Phase 1: Presentation of your project feature list, requirements, use cases and use-case diagram. Select any Saturday from this list: 2/23, 3/2, 3/9. Though please note that everyone is interested to present the LAST day (3/9) and we need to balance the load on the different dates.**
    2) **Phase 2: Presentation of your project development including domain model, design model, Sequence diagrams, and design patterns utilized, 5-minutes video recording of your project implementation and run. Select any Saturday from this list: 3/16, 3/30, 4/6, 4/13, 4/20, 4/27, 5/4. Again, please note that everyone is interested to present the LAST day (5/4) and we need to balance the load on the different dates. In this phase it is expected that at least 25% of your requirements/use-cases are being implemented for the live-demo.**

3) **Phase 3:** Final delivery of your project is a SINGLE WinZIP file on Blackboard (5/4/19 by 11:59 pm)

5. Communicate with the TA regarding the topic you will work on, and the dates you will present the project during class time, 5-10 minutes the length of your presentation.

6. The deadline to notify the about the topic that you selected, and the two presentation dates that you will present re allowed to your project is set to be **2/15/19** by 9:00am.

7. Budget your presentation for 5 to 10 slides

Project Technical Requirements:

For your final project, you will model, design, and implement an application for searching and charting Chicago businesses and develop a Dashboard for data analytics and visualization for **ChicagoSocialHub** web-app utilizing **Yelp** and **Divvy** APIs.

It is a requirement to use the ChicagoSocialHub code templates provided by Dr. Bader in your implementation.

Here is the list of topics you need to choose one from:

1. Search for places on a street, and show divvy nearest dock stations for a selected place. And a Dashboard to show the different review counts and ratings using pie-chart, bar-char, stacked-chart, etc. for the search results for top rated or reviewed places based on a filter specified by the user.
2. Search for places in a zip-code, and show divvy nearest dock stations for a selected place. And a Dashboard to show the different review counts and ratings using pie-chart, bar-char, stacked-chart, etc. for the search results per zip code for the entire city of Chicago.
3. Search for places in a zip-code, and show divvy nearest dock stations for a selected place. And a Dashboard to show the HeatMaps for the different review counts and ratings for the search results.
4. Search for places in a zip-code, and show divvy nearest dock stations for a selected place. And a Dashboard to show the HeatMaps for the available docks for divvy dock stations in that zip-code.
5. Search for places in a zip-code, and show divvy nearest dock stations for a selected place. And a Dashboard to show the real-time line chart for the available docks for divvy dock-stations in that zip-code. And a pie chart to show number of dock stations that are more than 50% full or empty, and neither 90% full or empty.
6. Search for places in a zip-code, and show divvy nearest dock stations for a selected place. And a Dashboard to show the real-time alerts in a table to show those dock-stations that are more than 90% full or empty in that zip code. *And a pie chart to show number of dock stations that are* 90% full or empty*, and neither 90% full or empty.*

7. Search for places on a street and show divvy nearest dock stations for a selected place. And a Dashboard to show the daily and hourly average numbers of available docks for every dock station based on the user selection for the past week, month, and year; divvy data log stored on ElasticSearch server.
8. Search for places in a zip code and show divvy nearest dock stations for a selected place. And a Dashboard to show the daily and hourly average numbers of available docks for every dock station based on the user selection for the past week, month, and year; divvy data log stored on ElasticSearch server.

Please note that no more than 3 teams can work on the same project topic; topics will be allocated based on FCFS.

Your Web-App and Dashboard will be modeled and designed using UML, and implemented using code templates provided by Dr. Bader, and must be tested on Firefox and Chrome platforms.

It is a requirement to use platforms that are compliant with ECMAScript 2015 scripting 2015, (ES6):

- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Classes

List of browsers/platforms that support ES6 can be found under ***modern browsers*** link on this URL:

- https://developer.mozilla.org/en-US/docs/Web/JavaScript

You can use **D3** or **Google** charting libraries to model, design, and implement your object-oriented Dashboard library.

Project Deliverables:

You are required to submit a SINGLE WinZip file that has the following deliverables are:

1. Requirements and Design document has the following
    1) Brief project overview statement.
    2) Requirements/Features List (All Requirements and Features must be numbered)
    3) Use Cases and Use Case Diagram
    4) Activity Diagrams
    5) Sequence Diagrams
    6) Domain Model Class Diagram
    7) Design Model Class Diagram
    8) Documentation and class diagrams for Design Patterns used.
2. Source Code
    a. Packages used in your implementation
    b. Readme file how to run your application
    c. Total number codes written must be documented in the first line of your Readme file
3. Output report that has ALL captured screen-shots of your project run saved in OUTPUT.pdf
4. Video recording of 10 minutes as a demo for the run of your project using https://screencast-o-matic.com/

ONLY the Team leader will post the final project as a SINGLE WIN-ZIP that has the PDF file and source code along with the out report on Blackboard on 5/4/19 by 11:59pm.

Please post your project under the name "CSP586 Project - Lastname, FirstName". On the very first page write your names and email address.

Dr. Atef Bader

Appendix

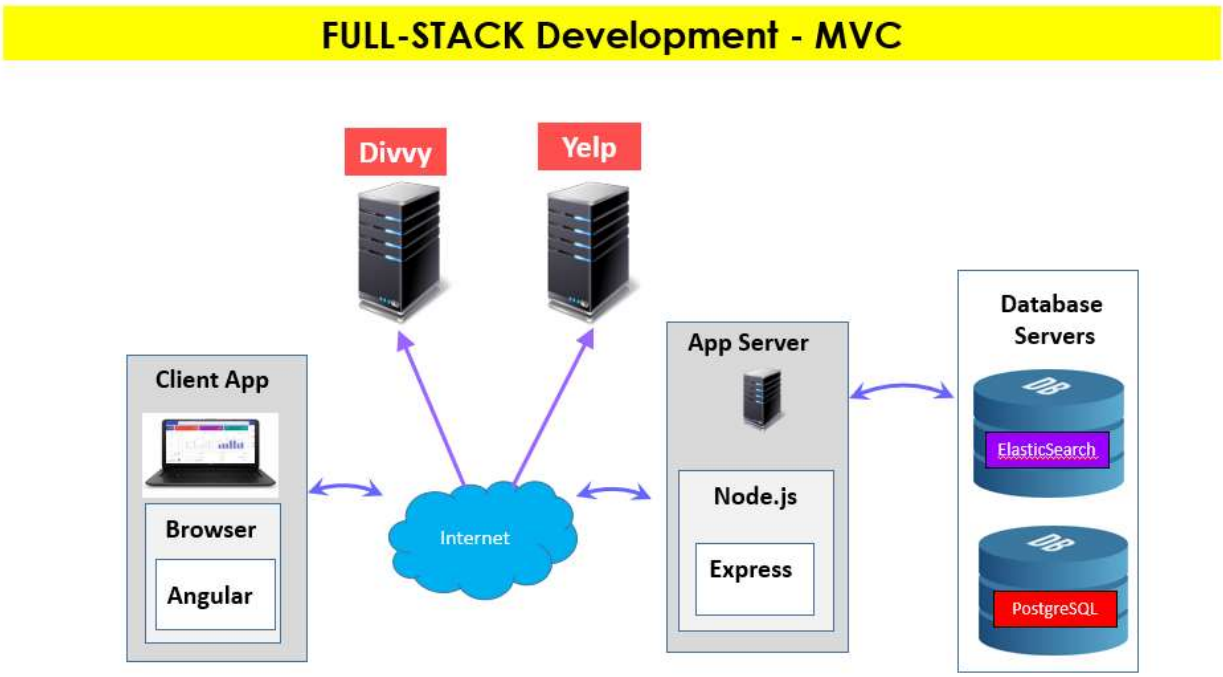
ChicagoSocialHub App

Technologies:
ChicagoSocialHub is a web-based real-time app that uses the following technologies:

1. Angular
2. Node.js/Express
3. PostgreSQL – to store Divvy station status
4. ElasticSearch – to store Yelp reviews for Chicago Businesses

Architecture:

ChicagoSocialHub utilizes the MVC architectural pattern:



Tools and Environment Setup & Installations

The following platforms/technologies and tools are needed in order to build and run the web-based application:

1. Javascript: platforms and browsers that are compliant with ECMAScript 2015 scripting 2015, (ES6): http://www.ecma-international.org/ecma-262/6.0/
2. Chrome and Firefox releases that are ES6 compliant. List of browsers/platforms that support ES6 can be found under *modern browsers* link (http://kangax.github.io/compat-table/es6/) on this URL: https://developer.mozilla.org/en-US/docs/Web/JavaScript
3. Java/JDK 11: https://www.oracle.com/technetwork/java/javase/downloads/jdk11-downloads-5066655.html
4. Python 3.7: https://www.python.org/downloads/
5. Angular 7: https://angular.io/
6. Anaconda: https://www.anaconda.com/distribution/#download-section
7. visual studio code: https://code.visualstudio.com/download
8. Node.js/express: https://nodejs.org/en/download/
9. Angular CLI: npm install -g @angular/cli
10. PostgreSQL: https://www.postgresql.org/download/
11. ElasticSearch: https://www.elastic.co/downloads/elasticsearch

How to Build and Run

1. Create your Yelp API Key and update the ipynb script with that key
2. Create your Google Map API Key and add your Key to the client/Angular frontend file, app.module.ts.
3. Run ChicagpSocialHub-Yelp.ipynb to create an index to Chicago Business on ElasticSearch
4. Execute the following commands from the command line window/terminal:
   4.1. Start ElasticSearch: server from the command prompt
   4.2. Start node.js server: node server
   4.3. Start Angular client: ng serve –open