CS490 Senior Design

Sprint 1 Planning Document

MyMapper

Team Members:

Chen Gong

Yunkai Sun

Yang Xu

Zhihao Hu

Ben Pastene

Mingsheng Xu

Contents

1 Sprint Overview

1.1 Overview

During the first sprint of the developing process, we firstly would like to get everyone's developing environment setup, including downloading OpenStreetMap data and set up Eclipse-JAVA developing environment for the frontend (web application) and backend (SPARK). Secondly, we are aiming at finishing some basic infrastructure building, setting up SPARK database and Tomcat web server. We are also targeting at answering some basic user stories by the end of this sprint. For example, developing a basic UI layout and coding for some basic functionalities on the Ingester.

1.2 SCRUM developing team and meeting schedule

Our SCRUM developing team members include Chen Gong, Yunkai Sun, Yang Xu, Zhihao Hu, Ben Pastene and Mingsheng Xu, with Yunkai being the team leader and Ben being the deputy. Our scrum team weekly meeting time would be every Thursday 1:30pm. However, once we divided tasks into small sub-developing groups, each group may have some more meeting time every week depending on people's available time.

2 Sprint Details

2.1 List of User Stories to be implemented

As a developer, I want to learn how to implement SPARK and set up database as well as how to get SPARK talking with web server.

As a developer, I want to learn how to interact with Tomcat and set up our web server.

As a developer, I will use github for source control, and rely on it to store finalized versions of all iterations of our design documents, in addition to source code. (Setup github for source control)

As a user (such as drivers, travelers etc), I want to visualize my query results in a web based map interface.

2.2 Task Description and Workload Estimation

- **Develop Basic Web UI:** Create basic web pages as .jsp files to take user input and display location info on an embedded map interface.
  Duration: 20 hours
  Team Members: Mingsheng Xu, Ben Pastene
- **Learn Spark:** Learn Spark by reading tutorials and going over Apache Spark examples.
  Duration: 20 hours
  Team Members: Yang Xu, Yunkai Sun, Zhihao Hu, Ben Pastene, Chen Gong
- **Set up Spark:** Get a spark cluster installed and running on at least one machine. Get it to respond to SQL queries. Solve any configuring problems that may arise.
  Duration: 25 hours
  Team Members: Yang Xu, Yunkai Sun, ZhiHao Hu
- **Set up Web Server:** Get an instance of Tomcat installed and running on a single machine. Have it host a single, simple landing page that can be loaded from any browser. Solve any configuring problems that may arise.
  Duration: 15 hours
  Team Members: Chen Gong, Ben Pastene
- **Download and format map data:** Download map data from OpenStreetMap and

format the data source to fit our database schema.

Duration: 5 hours

Team Members: Ben Pastene, Yang Xu, Yunkai Sun, Zhihao Hu

- **Build Ingestor:** Write a simple script to parse the openstreetmap .osm file for west lafayette to pull static points of interest (restaurants, libraries, etc.). Write functionality to insert the data into the spark database.

  Duration: 20 hours

  Team Member: Ben Pastene, Yang Xu, Yunkai Sun, ZhiHao Hu
- **Set up Git Repository:** Set up a GitHub repository, give everyone access, and add all of our documents and source files.

  Duration: 3 hours

  Team Member: Zhihao Hu, Yunkai Sun

3. Remaining Tasks

3.1 Remaining Functional Requirements

As a user (such as drivers, travelers etc), I want to:

- visualize my query results in a web based map interface.
- query for nearby POIs (Point Of Interests, including locations and buildings etc.).
- get the real time data of any moving objects in or around a specific area.
- use the query instance while I'm moving (the query moves follow the user and update results in real time).
- check how busy or congested an area is, such as checking how taxies are there in a giving area.
- view the closest object or POI of any type.
- create an account with the system so the application can deliver personalized results. (If time allows.)

As an administrator, I want to:

- generate moving objects data and insert them into the Spark database.
- add/delete/modify already existed data in the database.

- edit build-in queries in the system and provide more query options to the end user.
- have the ability to update the map data in the database when OpenStreetMap's repository is updated.

As a developer, I want to:

- distribute all map and object data into multiple machines in an efficient manner that allows for fast queries.
- generate the whole map in one user machine.
- update the locations of moving objects in real time, which requires a fresh batch of data inserted at regular, short intervals.

3.2 Remaining Non-functional Requirements

Performance Requirements

- As a user I want the query information to be real time.
- As a user I want the moving objects' location to be updated without too much lag.
- As a user I want to have an intuitive and clear UI.
- As a user I want the information displayed to be accurate.
- As a user I want the application to rank the results I find.
- As a user I want the application to return as many results that satisfy my requirements.
- As a user I want when the application crashes will have no impact on the other processes currently running on the machine.
- As a user I want the application to gracefully handle and process a heavy amount of data.
- As a user I don't want my phone or computer goes slow when i am using this application.
- As a user I want to see a pop up from the application if my internet connection is lost.

Platform Requirements

- As a user I want to access the web application from any major type of browsers.
- As a user I want to access the web application from any type of devices such as phones, tablets and computers, in which the screen size may vary.

- As a user I want to operate this application always in the same way, even when I run this application on different platforms.

Process Requirements

- As a developer, we will use github for source control, and rely on it to store finalized versions of all iterations of our design documents, in addition to source code.
- As a developer, Google docs and spreadsheets used by team members for collaboration will have final revisions duplicated to the github in PDF or other common format for recordkeeping purposes.

Other Requirements

- As a user, I want to see a user guide or tutorial when I use this application for the first time.(if time allowed).
- As an administrator, I want to validate and sanitize any user-inputted information for security concerns.
- As a developer, I want the system to support 100 concurrent users querying and using the application.