MICHIGAN STATE UNIVERSITY

Project Plan

Real Time Ad Campaign Management

The Capstone Experience

Team Urban Science

Zach Heick
Anthony Orr
Yoseph Radding
Hang Zhang

Department of Computer Science and Engineering
Michigan State University

Spring 2017

From Students…
…to Professionals

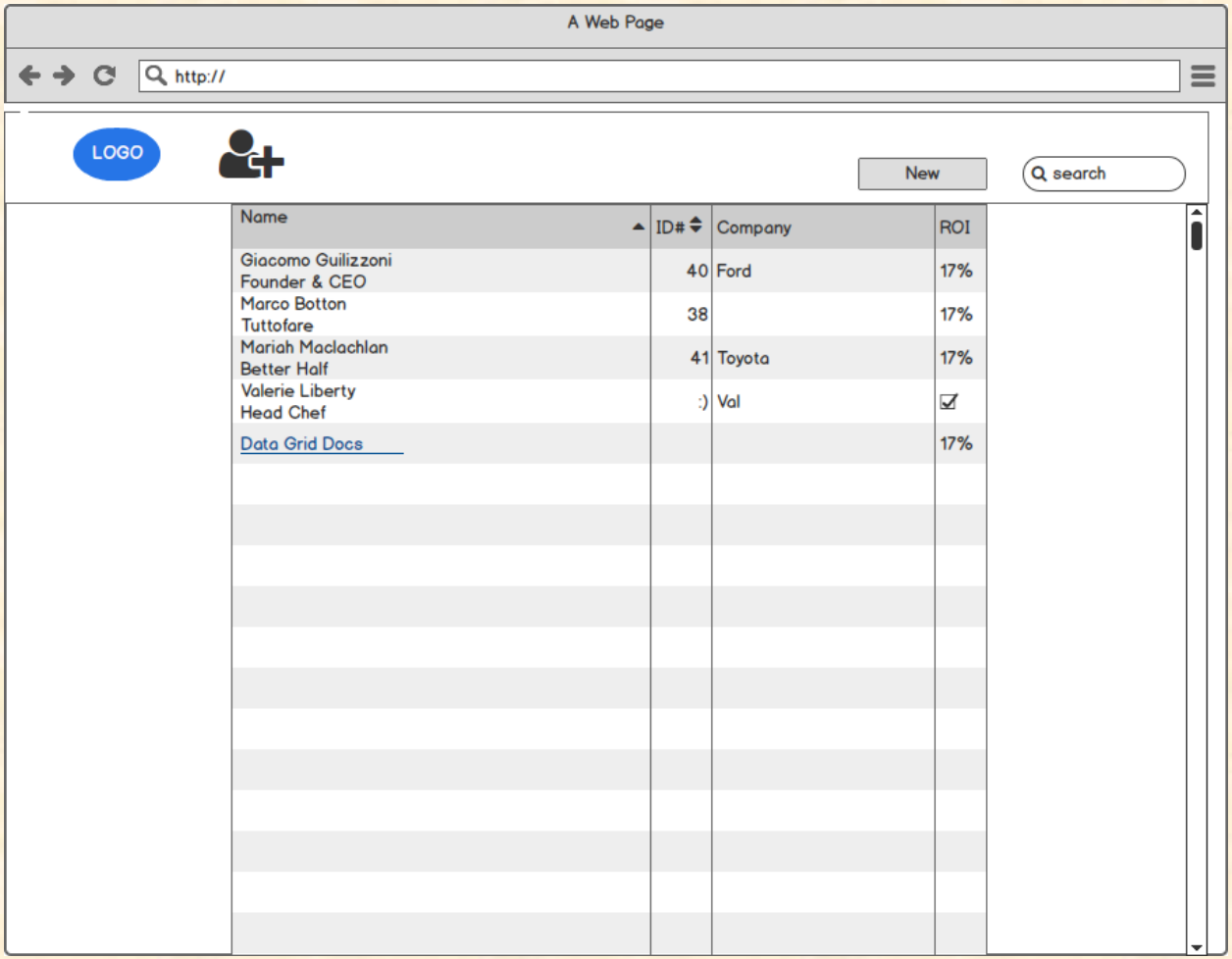
Functional Specifications

- The goal of this project is to create a real time analytic engine that processes changing data during an inflight ad campaign and recommends changes in order to save money
- The three main parts to be implemented are a control panel, analytic engine, and mock exposure data API

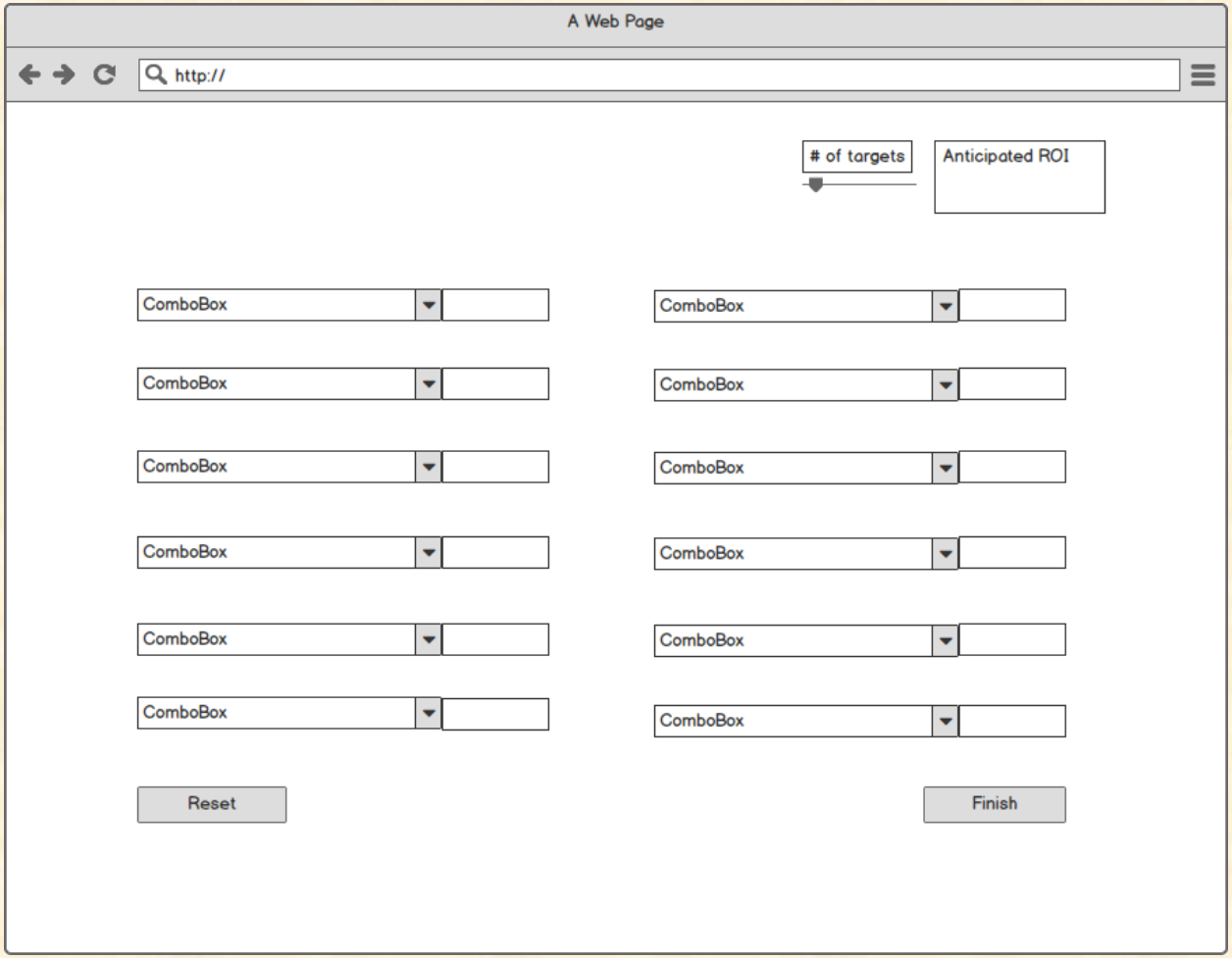
Design Specifications

- Control Panel
  - Campaign managers can use the control panel to add or manage campaigns
  - Consists of the homepage, create campaign, and manage campaign
- Analytic Engine
  - The engine will sort the data and update any changes to the current database
  - Uses a machine learning model to process key data points in order to provide a recommendation to the ad campaign managers about who to add or drop from the current campaign list
- Data Mock API
  - Exposure data is mocked from other data points

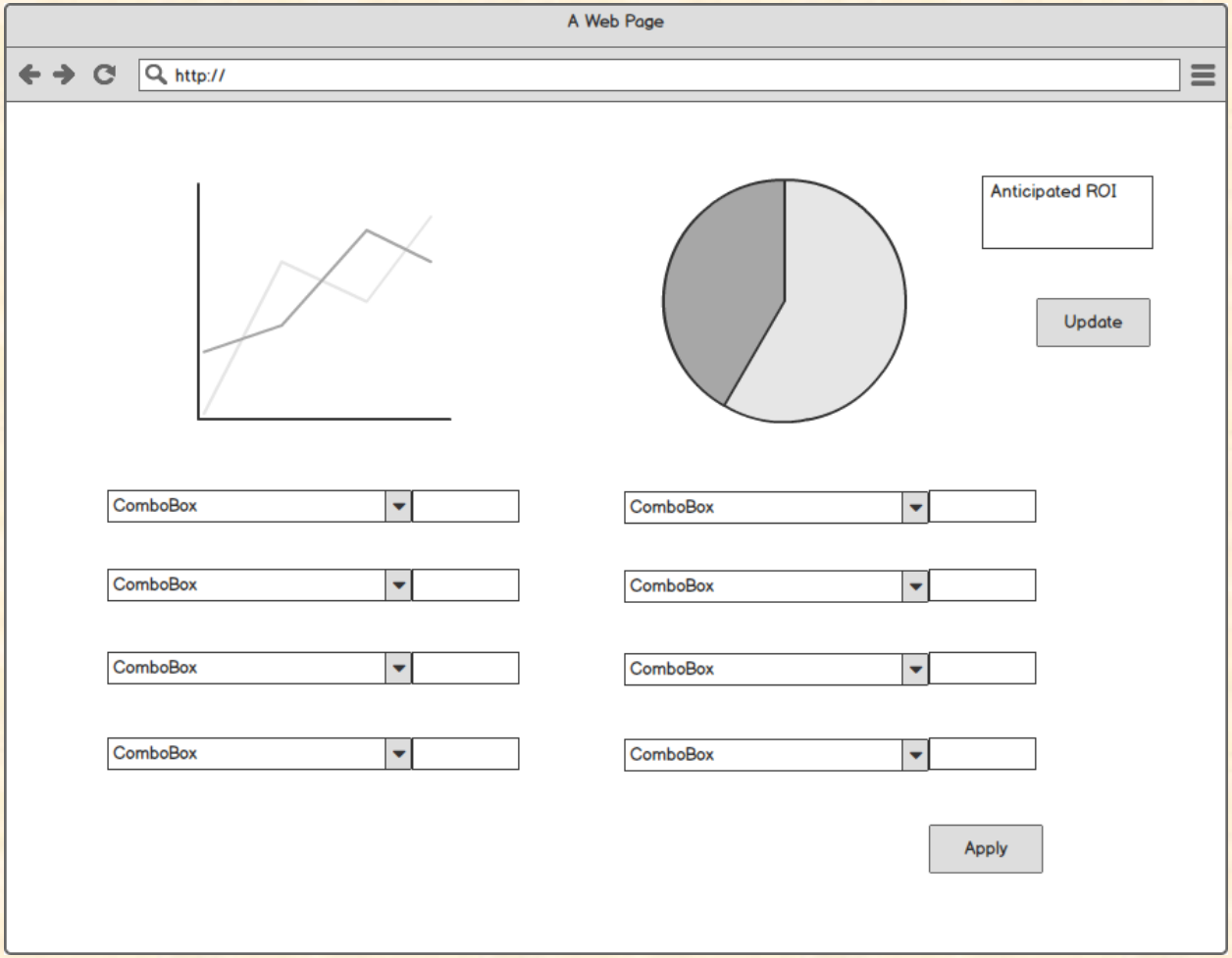
Screen Mockup: Homepage



Screen Mockup: Create Campaign



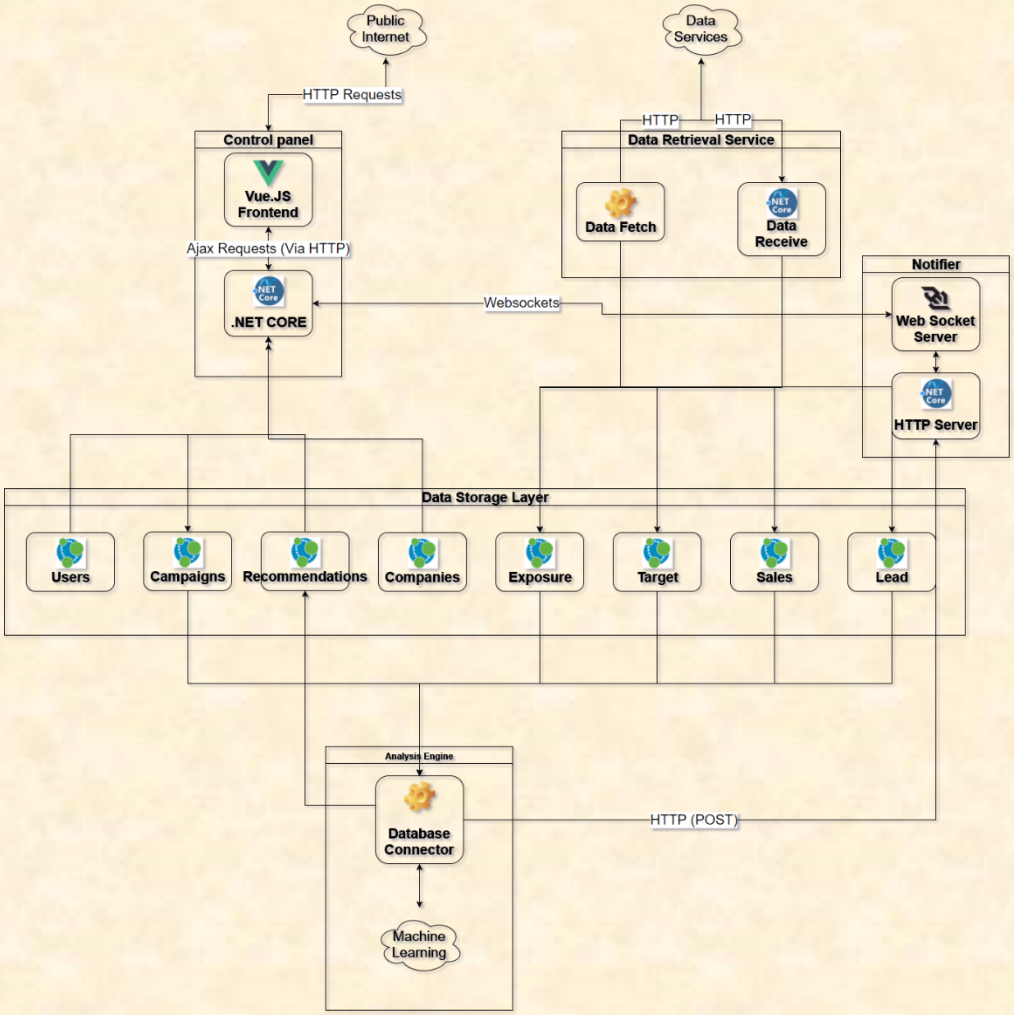
Screen Mockup: Manage Campaign



Technical Specifications

- Control Panel
  - Frontend will primarily present the data to the user and to update the data
  - Backend is the primary system, designed to retrieve data from our neo4j database and to provide that data to the frontend
- Analytic Engine
  - JSON formatted data from an outside source will be loaded into Neo4j and we can query and search for different relationships between properties of nodes and use this data for our recommendations
- Mock Exposure Data API
  - Customer ID's will be given to those who received an advertisement, selected percentage of these ID's will be selected as exposure data

System Architecture



System Components

- Hardware Platforms
  - Server
- Software Platforms/Technologies
  - Neo4j, Docker, .NET Core 1.0, Vue.js, HTML/CSS
- Development Technologies
  - Gitlab, Visual Studios, Ubuntu, Docker

Testing

- Unit Testing
  - For services written in C#, the unit tests will be written in XUnitTests, a library for unit tests from Microsoft
  - For Javascript, the testing will be done with Karma
- Integration Testing
  - Will comprise of testing the individual Docker subsystems using similar testing procedures listed above
- User Testing
  - Involves us and the clients going through and using the application like a real user
  - Additionally, we will use logging in order to make sure that the application is performing properly

Risks

- Mocking the Exposure Data
  - We do not have access to the real exposure data, so we need to generate realistic mock data for the analytic engine to process
  - To solve this, we will analyze the leads and sales data and create our own accurate mocked exposure data
- Implementing Machine Learning Algorithms in Neo4j
  - Neo4j uses a powerful query language that can handle very complicated flow control, but the efficiency is unknown
  - Read up on documentation and look at complex examples
- Dynamic Drop-down Menus
  - It will be difficult to populate these inputs while making recommendations for the campaign at the same time
  - A front-end framework, Vue.js, will help handle this challenge

Questions?