Lab 1 – Power Play Product Description

Nicholas Benfield

Old Dominion University

CS 411

Janet Brunelle

March 12, 2017

Version 2
Table of Contents

1 Introduction.................................................................................................................. 3

2 Power Play Product Description ................................................................................. 4
   2.1 Key Product Features and Capabilities ................................................................. 4
   2.2 Major Components (Hardware/Software) ............................................................ 5

3 Identification of Case Study....................................................................................... 6

4 Power Play Product Prototype Description ............................................................. 7
   4.1 Prototype Architecture (Hardware/Software) ....................................................... 7
   4.2 Prototype Features and Capabilities .................................................................. 8

Table 1 Feature comparison between prototype and full product................................. 9

4.3 Prototype Development Challenges ...................................................................... 9

Glossary ......................................................................................................................... 10

References.................................................................................................................... 11

Figures and Tables

Figure 1 Major functional components diagram ......................................................... 5

Table 1 Feature comparison between prototype and full product .............................. 8
1 Introduction

When a natural disaster happens, electricity is a very common utility to lose service. Utility companies must deploy personnel to resolve and restore service. In 2003, Hurricane Isabel took out more than 700,000 residents within North Carolina lost power to their homes during the Hurricane (Ambrose, K). When utility companies deploy employees to resolve lost services, they are reacting to situations they are forced into. Decisions like these often lead to costly repairs, service down time and employee expenses.

American society relies on having electricity being produced for the country. Electric companies like Dominion Power, supply massive amounts of electricity to parts of the country. Dominion specifically distributes power to Virginia and North Carolina (Dominion). Dominion has a lot responsibility when it comes to delivering power to their customers. From the managers of the dispatchers all the way up to the CEO, every decision must be made are reactive to the situation. Any decision that has incorrect information could potentially be disastrous for the company if the decision is implemented. Research Analysts of Dominion look at all the data that has been gathered throughout the year and present the data an annual decision meeting.

With data that reflects various possible scenarios researches will be able examine data more efficiently. A cross platform game that has different types of scenarios that players solve would provide solutions to researchers for analysis. This allows researchers to sort through
solutions and provide the most efficient way to handle similar situations. A game must be user friendly and provide an environment to bring players back to solve scenarios.

2 Power Play Product Description

Power Play will be a free to play game that is accessible through a browser. The game will be available on any device that is capable to connect to the Internet. Google Maps will be used to provide real world locations that will provide the mapping of the area. The Spark Simulation engine will be receiving data from the game that will be used to generate real numbers for the player’s actions. As the player plays through the game, their decisions will be reported and recorded to be analyzed by research analysts.

Ultimately, the game will provide data for researchers to improve future decision making at Dominion Power for future incidents. This will allow for an informed decision from the analyzed data. Each scenario a player is given, will result in different outcomes between multiple players. As the player replays the same scenario, they can optimize the approach towards the given scenario.

This game will be provided to Dominion will allow them to use the various outcomes to train personnel on the best practices for scenarios. Each scenario will be unique and this uniqueness allows the personnel to have more than one approach to solving the issue.

2.1 Key Product Features and Capabilities

Players will be able to create an account for the game by using a web browser to sign up. When they sign up, they will be given a profile that will keep track of the outcomes from scenarios that they have completed. Achievements will be given to players who complete various tasks within scenarios. At the end of every scenario, an analysis of the decisions they
made throughout the game will be shown to them. The more scenarios a player completes, the higher the rank they become and will be compete with other players for higher ranks. The Spark Simulation will handle the simulation data that is generated when a turn is finished. Players will be engaged in a turn-based environment where they will make decisions and end their turn.

2.2 Major Components (Hardware/Software)

In Figure 1, the major components that are required for this product to function are shown. There are two databases that communicate with each other through information stored in each database. The web server will handle players access the website along with providing the player with information from the scenarios. Spark Simulation engine will handle the simulation of the scenarios and processing the user’s data when they finish a turn. The three databases will contain player information, gameplay information, and results from the scenarios they have completed. Each of these databases act as intermediators between the two servers. Players and research analysts can access the website and see reports from completed scenarios.

![Major Functional Components Diagram](image-url)

Figure 1 Major functional components diagram
This server will be running Ubuntu as the operating system for running the webserver and databases on. Python will be used to make connections to the Spark Simulation. These databases are using Flask as the central way of connecting to the databases. The databases will be using MongoDB to store the information from the web server. The front end of the game will be using PHP and AJAX to generate the various scenarios players will see through the web browser.

3 Identification of Case Study

The focus of this product is for companies such as Dominion to utilize the data that is gathered from players to make smarter decisions towards issues they may face. With the use of this game, players will learn the struggles that utilities companies face during certain scenarios. Utility companies benefit from completed scenarios because they can take the players outcome and apply it to similar scenarios. This game is designed with power companies in mind, but it is designed to be able to expand to other types of utility companies.

A grant between Dominion Power and Old Dominion University’s Systems Engineering Department is allowing the Spark Simulation Engine to be developed. The Spark Simulation is using real data from Dominion to create accurate scenarios.

With the information that is gathered, utility companies can compare how they previously handled specific scenarios in regions to the way other players have handled the situation. This reduces the turnaround time that they face when they must make decisions on building or maintaining facilities.

(This space is intentionally left blank)
4 Power Play Product Prototype Description

Power Play is a free to play turn-based game that will be accessed through a web browser. Players will be shown a user interface where they will be acting as CEO over a utility company. At the start of the game, the player will be given a scenario in which they are to complete. After the player, has completed a scenario they will be given feedback on how well they handled the scenario. The feedback will also be stored and be accessible by research analysts.

Dominion Power research analysts will be able to review the decisions players have made and analyze where the player went wrong. As more player’s results get generated, researchers can view common mistakes between different players.

4.1 Prototype Architecture (Hardware/Software)

The server that the prototype will be hosted on a Virtual Machine provided by the Old Dominion University Computer Science Department. The Virtual Machine will be using Ubuntu as it has a lower overhead on the hardware and provides stable releases. The databases will be using MongoDB to store data from the Spark Simulation Engine. We will have one database that will be connecting to the Spark Simulation Engine and provide the prototype with scenarios. The other two databases will be populated with information generated by players signing up and the decisions they make within the scenarios. The main programming language that will be used will be HTML. This will be what the web browser will receive to display to players. PHP and AJAX will be how the players can connect to the server without having to constantly refresh the page after each turn.
4.2 Prototype Features and Capabilities

In Table 1, the features are being compared to the real-world product and the prototype. The base functionality of the real-world product can be accomplished in the prototype. The Player Features and the Administrative Features are the focus that our prototype will achieve.

<table>
<thead>
<tr>
<th>Features</th>
<th>RWP</th>
<th>Prototype</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Player Features</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plant retirement</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Plant refurbishment</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Plant construction</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Viewable resources</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Viewable Environmental Impact</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Viewable Earnings</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Administrative Features</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create game scenarios</td>
<td>✓</td>
<td>✓ partially implemented</td>
</tr>
<tr>
<td>Create and remove player accounts</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Reset collected data</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Access player game results</td>
<td>✓</td>
<td>✓ partially implemented</td>
</tr>
</tbody>
</table>
### Table 1 Feature comparison between prototype and full product

<table>
<thead>
<tr>
<th>Feature</th>
<th>Prototype</th>
<th>Full Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>Player update notification</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td><strong>Analysis Features</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Viewable decision data</td>
<td>✔</td>
<td>✗ eliminated</td>
</tr>
<tr>
<td>Viewable event collection data</td>
<td>✔</td>
<td>✗ eliminated</td>
</tr>
<tr>
<td>Viewable data collection by type</td>
<td>✔</td>
<td>✗ eliminated</td>
</tr>
<tr>
<td>Downloadable data by region</td>
<td>✔</td>
<td>✗ eliminated</td>
</tr>
<tr>
<td>Downloadable data by date</td>
<td>✔</td>
<td>✗ eliminated</td>
</tr>
</tbody>
</table>

4.3 Prototype Development Challenges

A challenge that the prototype will face, will be connecting to the Simulation Spark Engine with the MongoDB databases. The simulation engine is being developed by another group of software developers. The design of the prototype must be adjustable to accommodate the engine and updates the engine may have.

The integration of the Google Maps API will be a challenge as the game will be creating elements on top of the maps as they are generated. Most of the development team has experience with web development and the ability to implement features fully is a heavy challenge on the prototype.
Glossary

**AJAX**: A client side application that allows for asynchronous connection to a webserver without having to reload the page.

**Flask**: A pithing microframework for Python.

**Google Maps API**: An API that allows programmers to access data from Google Maps.

**MongoDB**: A document oriented database. A NoSQL database that uses JSON-like documents to create the schema.

**PHP**: Hypertext Preprocess (PHP) is a server side scripting language that generates webpages dynamically.

**Python**: A high-level programming language.

**Spark Simulation Engine**: A simulation that generates real world calculations for scenarios.
References


