Greenworks Eco Map

Technical Design Document

Jason Castillo Reyes

Matthew Drennan

Beethoven Marhone

Ethan Nephew

James Rolon

Samuel Soares Walton

Table of Contents

1. Abstract 3
   1. Objective 3
   2. Motivation 3
2. Code Policy 4
3. Project Stakeholders 5
   1. Team Members 5
   2. External Parties 5
4. Requirements 6
   1. Legend 6
   2. Admin Requirements 6
   3. Developer Requirements 6
   4. User Requirements 7
5. Project Schedule 8
   1. Project Schedule 8
   2. Phases of Development 8
   3. Phase Objectives 8
   4. Development Technologies 9
6. Risk and Mitigation 10

Section 1: Abstract

Subsection a: Objectives

The objective of this project is to develop a map-based web application that will be accessible through web browsers. The map will display points that are of interest. Points of interest will be related to green technology and events.

Subsection b: Motivation

The motivation for this project is to increase the visibility of green technologies in Orlando.

Section 2: Code Policy

1. Do not push a bunch of files at once.
2. If you discover a bug, document it, and write a unit test for that bug.
3. Readable code is better than clever code.
4. Always test your changes before pushing to GitHub.
   1. Backend changes: verify that all unit tests pass.
   2. Frontend changes: double check in the browser.
   3. Frontend changes: verify mobile compatibility.

Section 3: Project Stakeholders

Subsection a: Team Members

Jason Castillo Reyes

Matthew Drennan – CO-LEAD

Beethoven Marhone

Ethan Nephew – CO-LEAD

James Rolon

Samuel Soares Walton

Subsection b: External Parties

Dr. Macon

City of Orlando IT

Section 4: Requirements

Subsection a: Legend

|  |  |
| --- | --- |
| Term | Definition |
| Actor | Who the requirement pertains to |
| Adm | Client-side actor who will maintain the web application database |
| Dev | Server-side actor who will develop the web application |
| User | Client-side actor who will access the web application |
| A#.# | Requirement ID pertaining to Actors of type “Adm” |
| D#.# | Requirement ID pertaining to Actors of type “Dev” |
| U#.# | Requirement ID pertaining to Actors of type “User” |

Subsection b: Admin Requirements

|  |  |  |
| --- | --- | --- |
| Requirement ID | Requirement | Actor |
| **A1.0** | **Access to web application** | **Adm** |
| A1.1 | Secure web-based portal | Adm |
| **A2.0** | **Data Validity** | **Adm** |
| A2.1 | Ability to view and edit database data | Adm |
| **A3.0** | **Interactivity** | **Adm** |
| A3.1 | Login and logout | Adm |
| A3.2 | Controls for creating, updating, inserting, and deleting data | Adm |
| **A4.0** | **Map** | **Adm** |

Subsection c: Developer Requirements

|  |  |  |
| --- | --- | --- |
| Requirement ID | Requirement | Actor |
| **D1.0** | **Access to web application** | **Dev** |
| D1.1 | Develop on Java platform | Dev |
| D1.2 | Apache Tomcat server for development | Dev |
| D1.3 | Host on City of Orlando server for production | Dev |
| **D2.0** | **Data Validity** | **Dev** |
| D2.1 | Admin panel | Dev |
| D2.2 | Build back-end that processes and saves data into a database | Dev |
| D2.4 | Reject improperly formatted data | Dev |
| D2.5 | Feature for bulk data insertion/load | Dev |
| **D3.0** | **Interactivity** | **Dev** |
| D3.1 | Implement back-end calls with Jakarta Server Pages | Dev |
| D3.2 | Implement display with HTML, CSS, and JavaScript | Dev |
| **D4.0** | **Map** | **Dev** |
| D4.1 | Display points of interest on map | Dev |
| D4.2 | Implement point filter based on point type | Dev |
| D4.3 | Implement point filter based on event day | Dev |

Subsection d: User Requirements

|  |  |  |
| --- | --- | --- |
| Requirement ID | Requirement | Actor |
| **U1.0** | **Access to web application** | **User** |
| **U2.0** | **Data Validity** | **User** |
| **U3.0** | **Interactivity** | **User** |
| U3.1 | Control the type of data points that will be displayed | User |
| U3.2 | Control which event day will be displayed | User |
| **U4.0** | **Map** | **User** |
| U4.1 | View location points displayed on the map | User |
| U4.2 | View event points displayed on the map | User |

Section 5: Project Schedule

Subsection a: Key Dates

1. Planning Documentation – Jan 31, 2022
2. Prototype – Feb 17, 2022
3. Test Plan – Mar 4, 2022
4. Beta Versions – Mar 18, 2022
5. Beta Test – Mar 28, 2022
6. Developer Documentation – Apr 19, 2022
7. Final Release – Apr 28, 2022

Subsection b: Phases of Development

1. Planning Phase
   1. Create Plan Documents
2. Prototype Phase
   1. Feature Prototype
   2. Refinement & Refactoring
   3. Unit Testing
   4. Documentation
3. Testing Phase
   1. Pre-Beta Integration Testing
   2. Pre-Beta Performance Testing
4. Beta Phase
   1. Demonstrate Wide Application Performance
   2. Verify Requirements
5. Developer Documentation
   1. Finalized User Manual
6. Project Finalization
   1. Submit Final Project

Subsection c: Phase Objectives

1. Planning Phase Objective

Obtain a teamwide understanding of the project requirements.

1. Prototype Phase Objectives

Develop an operational web application that fulfills the basic user and admin requirements. Refine features, refactor the code base, implement unit testing, and complete Javadoc.

1. Testing Phase Objectives

Verify that the user and admin requirements have been fulfilled. Verify that the project satisfies performance metrics. Code base finalization should be achieved by the end of this phase.

1. Beta Phase Objectives

All components will be functional. The product has reached a presentable state. All requirements have been met satisfactory standards.

1. Developer Documentation Objectives

Completion of a rigorous user manual for the product.

1. Project Finalization Objectives

The project is completed and shippable. The project will be hosted on a website.

Subsection d: Development Technologies

Software that will be used to complete this project:

* Java SDK
* Visual Studio Code
* Eclipse IDE
* MySQL Workbench
* PhpMyAdmin
* Microsoft 365

Section 6: Risk and Mitigation

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **Risk** | **Impact** | **Analysis & Mitigation** |
| Planning | Scope creep | High | Refer to planning documents and supported documents to enforce project requirements. |
| Lack of clarity | High | Present the point with one of the co-leads for clarity. |
| Prototype | Undeveloped feature | Medium | Consult Dr. Macon and find a solution. |
| Improper use of push | Low | We will be implementing a code policy. |
| Workflow problems | Medium | Bring up the problem in the group chat. One of the co-leads will assist with troubleshooting and resolution. |
| Inability to effectively unit test | Medium-High | Refactor the code base and extract logic out of super methods. |
| Code documentation is insufficient | Low-Medium | Identify areas where revision is necessary and form a strategy for addressing these areas. |
| Testing | Integration tests fail | High | Review the program logs and verify that unit tests are passing. Isolate the problem origin so a targeted approach can be attempted. |
| Performance tests fail | High | If our web application doesn't meet performance metrics, then we will need to deconstruct the components of our system and find ways to increase efficiency. |
| Bug Detection | Low-High | Bugs will likely be detected during this phase. For each discovered bug we will document it and write a unit test for that specific bug. |
| Beta | Missing requirement | High | It will be mission critical to implement all missing components at this stage. |
| Application is not operational | High | Consult Dr. Macon and find a solution. At this point, this will likely mean that we have failed our project goals. |
| Developer Documentation | Overly complicated instructions | Medium | If our installation method is too complicated, we must implement measures to simplify the process. |
| Project Finalization | Server deployment problems | High | Contact our IT sources. |
| Bug discovered after deployment | Low-High | Replicate the bug and discuss with team. |