Oakland Inconvenience Tracker

Sprint Review Report

Sprint 1

1.0

10/20/18

Dylan Sporrer

David Rowan

Brandon Donahue

Adam Farabaugh

Software Engineers

Prepared for

CS 1530

Fall Term 2018

University of Pittsburgh

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 10/20/18 | Initial Creation | Dylan Sporrer |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  | Dylan Sporrer | Software Eng. |  |
|  | David Rowan | Software Eng. |  |
|  | Brandon Donahue | Software Eng. |  |
|  | Adam Farabaugh | Software Eng. |  |

**Table of Contents**

Revision History ii

Document Approval ii

1. Introduction 1

2. Specific Goals 1

3. AnalTICS 2

4. CONCLUSION 3

# 1. Introduction

The primary goal of this sprint was to establish the skeleton of the overall application, specifically the platforms and external tools which will be utilized to carry out the primary functionality. To this end, the features worked on during this sprint were: building the server and database instances needed to allow web functionality and listing storage respectively, generating appropriate schema for this database, creating a class capable of interfacing with the Google Maps API, and beginning to construct the layout of the applications primary web pages. All told, these features were divided among four stories totaling 6 points.

Of the planned stories, the creation of the server, building of the database schema, and the structuring of the web pages were complete, with the Maps API interface remaining partially incomplete. The completed stories account for 4 or the 6 planned points.

While the sprint did not meet all of its goals, it has established a firm foundation on which the interactive and user-benefitting features of the inconvenience tracking system can be built. From here, the next sprint will focus on the creation of these features as isolated units before moving on to linking said features into a cohesive whole.

# 2. Specific Goals

***2.1* Setup AWS Server and RDS Instance (SID: 01)**

***2.1.1 Story Description:***

This story covers the creation of a server and database using Amazon Web Services toolset

***2.1.2 Story Acceptance Criterion:***

This story is considered complete when the AWS database can be connected to remotely by all members of the design team and example code can be used to build an application of the server

***2.1.3 Story Dependencies:***

This story has no dependencies

***2.1.4 Story Challenges:***

An initial inability to build the application server environment due to AWS account permissions delayed the creation of the SQL database instance

***2.1.5 Story Assigned to:*** Dylan Sporrer

***2.1.6 Story Points:*** 1

***2.1.7 Status:*** Complete

***2.2* Design and Creation of Database Schema (SID: 02)**

***2.2.1 Story Description:***

This story covers the design of a set of relational schema relevant to the primary function of the system and the instantiation of these schema on the AWS-hosted database

***2.2.2 Story Acceptance Criterion:***

This story is considered complete when a set of efficient relational schema which can be used to store the listing data, user profile information, and tag-set information has been designed and can be instantiated on the AWS database

***2.2.3 Story Dependencies:***

This story is dependent on SID 01

***2.2.4 Story Challenges:***

***2.2.5 Story Assigned to:*** David Rowan

***2.2.6 Story Points:***1

***2.2.7 Status:*** Complete

***2.3* Initial Web Page Construction**

***2.3.1 Story Description:***

This story covers the construction of a framework for the landing page which will be the primary UI element in the system

***2.3.2 Story Acceptance Criterion:***

This story is considered complete when a webpage with a structure suitable for inserting the basic interactive elements of the system exists

***2.3.3 Story Dependencies:***

This story has no dependencies

***2.3.4 Story Challenges:***

***2.3.5 Story Assigned to:*** Brandon Donahue

***2.3.6 Story Points:***2

***2.3.7 Status:*** Complete

***2.4* Maps Interface Class**

***2.4.1 Story Description:***

This story covers the creation of a class to interface with the existing JavaScript Google Maps API

***2.4.2 Story Acceptance Criterion:***

This story is considered complete when data can be passed into the class and returned as JSON that can be easily passed into some JavaScript that can use that information to query the Maps Api.

***2.4.3 Story Dependencies:***

This story has no dependencies

***2.4.4 Story Challenges:***

This story had a few challenges associated with it. First was finding the best approach to hitting the Google Maps Api. I did some research on the Maps Static Api and also the Maps Javascript Api. After doing this research I found that the Maps Javascript Api was more flexible and easier for the requirements of our application. Then, the next challenge was learning the basics of .Net framework in order to write a class that would pass the arguments to our script that would call the api and manipulate the DOM.

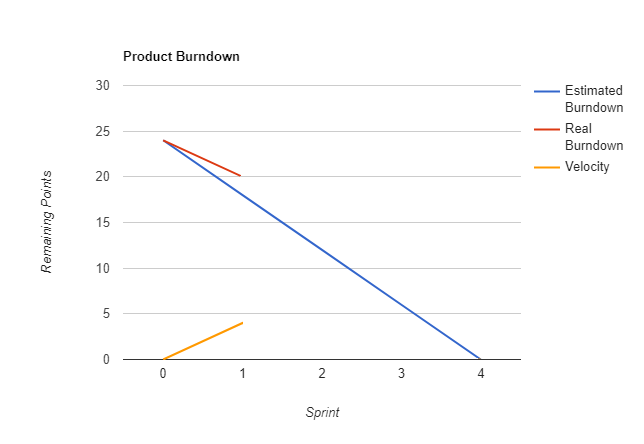
***2.4.5 Story Assigned to:*** Adam Farabaugh

***2.4.6 Story Points:***2

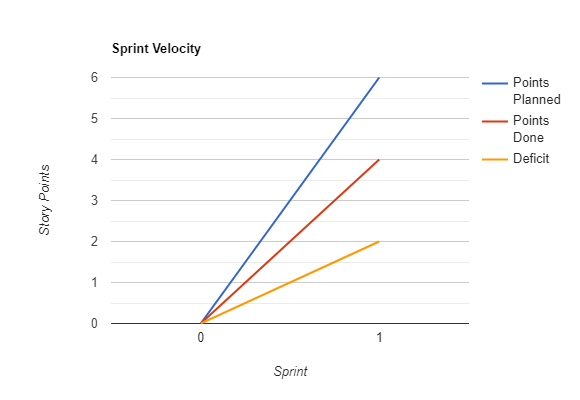
***2.4.7 Status:*** Complete

# 3. Analytics

## 3.1 Sprint/Product Burndown Chart (sample chart shown below)



## 3.2 Sprint Velocity (sample chart shown below)



# 4. Conclusion