

---

**Software Design Document**

**for**

**Hump Yard Scheduling**

**Version 1.0 approved**

**Prepared by Adam Halley, Anna Rue, Thomas Anthone, Chris Recinos**

**University of Nebraska Omaha**

<b>Table of Contents.....</b>	<b>ii</b>
<b>Revision History.....</b>	<b>ii</b>
<b>1. Introduction.....</b>	<b>1A</b>
<b>2. System Overview.....</b>	<b>1</b>
<b>3. Design Considerations.....</b>	<b>1</b>
3.1 Assumptions and Dependencies.....	1
3.2 General Constraints.....	1
3.3 Goals and Guidelines.....	1
3.4 Development Methods.....	2
<b>4. Architectural Strategies.....</b>	<b>2</b>
<b>5. Policies and Tactics.....</b>	<b>2</b>
<b>6. Detailed System Design.....</b>	<b>2</b>
<b>Appendix C: To Be Determined List.....</b>	<b>3</b>
<b>Table of Contents</b>	
.....	3

## Revision History

Name	Date	Reason For Changes	Version

## 1. Introduction

The design document will detail the implementation of the requirements as defined in the Software Requirements Specification for Hump Yard Scheduling Version 1.0. This document covers an overview of the project, design considerations, architectural strategies, system structure, policies and tactics, and a detailed system design description. The intended audience of this document is the system architect, lead developer, and development teams.

## 2. System Overview

The yard scheduling application will be web accessed user interface. The application's purpose will be to receive user uploaded inbound car schedule in a standard csv format. The application will have live access to the car inventory and capacities of the tracks. The application will use implemented logic to determine each cars' hump track destination, update the car inventory, and update track capacity. Users will be able to view cars on each track.

## 3. Design Considerations

### 3.1 Assumptions and Dependencies

*The application shall make the following assumptions, the application will assume the accuracy of uploaded documents by users, accuracy of car inventory databases, accuracy of track capacities, accuracy of cars physical presence, and track and car transport worthiness.*

*The application shall depend on stable networks, operational systems, end-user timely document uploads, and external physical awareness of safe operation verification.*

### 3.2 General Constraints

*The application constraints of Version 1.0 include:*

*Network constraints aforementioned in the SRS,*

*Hardware/Infrastructure*

*Federal Regulation*

*End-User*

*Interoperability*

### 3.3 Goals and Guidelines

*Our goal is to create a manageable and expandable application that will marketed to multiple railroads it teams. Other than database restructuring and manipulation we hope to achieve a consistent approach to scheduling that will assist another automated yard AI. We will use an agile approach to development were we will sacrifice documentation to focus on clean working code.*

### 3.4 Development Methods

*We used an agile software development approach to the Yard Scheduler app. We remained highly focused using tools like git, github, and jetbrains You track for version control and performance tracking. We focused primarily on clean effective code and short documentation. The result allowed us to develop a rather large application in a very short period of about two and a half months.*

#### **4. Architectural Strategies**

- Use of Python in cloud environment has gained popularity at most class 1 railroads in the united states. So we decided to use the Django framework.*
- Set up a rest framework API for future front end expansion and upgrades.*
- MYSQL is used for easy integration to multiple standardized databases and servers*
- Django admin is used for ease of control and standardization of security*
- NGINX server is used as a minimal cost solution to server deployment*
- Data will be distributable to for cross functional use in the company*

#### **5. Policies and Tactics**

- Primary IDE's were used include VSCode, VSCodium, VIM, and Pycharm*
- Gnome terminal was used to make edits and run the Django server*
- Flake 8 was used for code format and restrictions*
- Selenium IDE was used for web frame testing.*
- Pytest/unittests were used for all applicable software tests.*

#### **6. Detailed System Design**





