**High Level Design (HLD)**

Rental Bike Share Prediction

**Document Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date Issued** | **Version** | **Description** | **Author** |
| 03/08/2022 | 1 | Initial HLD – V1.0 | A. Jana |
|  |  |  |  |
|  |  |  |  |

Contents

[Abstract 4](#_Toc110474109)

[1 Introduction 5](#_Toc110474110)

[1.1 Why this High-Level Design Document? 5](#_Toc110474111)

[1.2 Scope 5](#_Toc110474112)

[1.3 Definitions 5](#_Toc110474113)

[2 General Description 6](#_Toc110474114)

[2.1 Product Perspective 6](#_Toc110474115)

[2.2 Problem statement 6](#_Toc110474116)

[2.3 PROPOSED SOLUTION 6](#_Toc110474117)

[2.4 FURTHER IMPROVEMENTS 6](#_Toc110474118)

[3. Technical Requirements 6](#_Toc110474119)

[3.1 Software Requirements 7](#_Toc110474120)

[3.2 Tools used 7](#_Toc110474121)

[3.3 Hardware Requirements 8](#_Toc110474122)

[4. Design Details 8](#_Toc110474123)

[4.1 Process Flow 8](#_Toc110474124)

[4.2 Event log 8](#_Toc110474125)

[4.3 Error Handling 8](#_Toc110474126)

[5. Performance 9](#_Toc110474127)

[5.1 Reusability 9](#_Toc110474128)

[5.2 Application Compatibility 9](#_Toc110474129)

[5.3 Resource Utilization 9](#_Toc110474130)

[5.4 Deployment 9](#_Toc110474131)

[6. KPIs (Key Performance Indicators) 9](#_Toc110474132)

[7. Dashboards 9](#_Toc110474133)

[8. Conclusion 10](#_Toc110474134)

# Abstract

Bike sharing systems are new generation of traditional bike rentals where whole process from membership, rental and return has become automatic. Through these systems, user can easily rent a bike from a particular position and return at another position. Currently, there are about over 500 bike-sharing programs around the world which is composed of over 500 thousand bicycles. Today, there exists great interest in these systems due to their important role in traffic, environmental and health issues.  
  
Apart from interesting real-world applications of bike sharing systems, the characteristics of data being generated by these systems make them attractive for the research. Opposed to other transport services such as bus or subway, the duration of travel, departure and arrival position is explicitly recorded in these systems. This feature turns bike sharing system into a virtual sensor network that can be used for sensing mobility in the city. Hence, it is expected that most of important events in the city could be detected via monitoring these data.

The goal here is to build an end-to-end regression task. Here the user will provide the

data and the result will be given by the best performing hyper tuned Machine Learning

model. The user will also get privileges to choose the deployment options.

# 1 Introduction

## 1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

* Present all the design aspects and define them in detail
* Describe the user interface being implemented
* Describe the hardware and software interfaces
* Describe the performance requirements
* Include design features and the architecture of the project
* List and describe the non-functional attributes like:
* Security
* Reliability
* Maintainability
* Portability
* Reusability
* Application compatibility
* Resource utilization
* Serviceability

## 1.2 Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly technical terms which should be understandable to the administrators of the system.

## 1.3 Definitions

Term Description

|  |  |
| --- | --- |
| RBS | Rental bike share |
| Database | Collection of all the information monitored by this system |
| IDE | Integrated Development Environment |
| AWS | Amazon Web Services |

# 2 General Description

## 2.1 Product Perspective

RBS is a regression-based predication model of bike rental count hourly or daily based on the environmental and seasonal settings.

## 2.2 Problem statement

To create an AI solution using RBS to implement the following use case.

* Create a virtual sensor network that can be used for sensing mobility in the city. Hence, it is expected that most of important events in the city could be detected via monitoring these data.
* To detect high traffic load which leads to higher accident rate on a particular day or hour and send details to the concerned authorities (e.g., ambulance, police fire department etc.) to stay on high alert.

## 2.3 PROPOSED SOLUTION

The proposed solution is a regression-based predication model on RBS data which will count the number of bike rental daily or hourly basis. This will help us understand the general traffic load in a city and a user and make informed decisions on their travel. Since higher traffic leads to higher chance of accidents, RBS will share daily or hourly estimate of traffic with the authorities to take rapid action.

## 2.4 FURTHER IMPROVEMENTS

More data can be added to RBS system to have a better understanding of human health and carbon footprint of a city. Data like distance travelled by a user, age, income can be useful addition.

# 3. Technical Requirements

The following are the technical requirements.

* IDE: For performing modular coding, debugging, git integration.
* Cloud based NoSQL DB: Cassandra for storing and retrieving data.
* Flask: Web application framework to interact with UI and backend python code.
* UI (HTML, CSS): for UI development.

## 3.1 Software Requirements

* Python >3.6
* CSS
* HTML
* Java Script
* PyCharm IDE
* Web Browser: Google Chrome, Internet Explorer, Mozilla etc.
* Data Base: No SQL Cassandra DB

## 3.2 Tools used

Python programming language and frameworks such as NumPy, Pandas, flask are used to build the whole model in PyCharm IDE.

|  |  |  |
| --- | --- | --- |
| Logo, icon  Description automatically generatedLogo  Description automatically generated with medium confidence | Logo  Description automatically generated | A picture containing text, businesscard  Description automatically generated |
| Text  Description automatically generated with low confidence | Diagram  Description automatically generated | Logo  Description automatically generated with medium confidence |
| Logo  Description automatically generated with medium confidence | Icon  Description automatically generated with medium confidence | A picture containing text, tableware, dishware, plate  Description automatically generated |
| A picture containing text, clipart  Description automatically generated |  | Graphical user interface, text, chat or text message  Description automatically generated with medium confidence |

* PyCharm is used as IDE.
* Cassandra DB is used to retrieve, insert, delete, and update the database.
* For visualization of the plots, Plotly is used.
* Front end development is done using HTML/CSS or Tkinter.
* Python Flask is used for backend development.
* GitHub is used as version control system.
* AWS is used for deployment of the model.
* Tableau/Power BI is used for dashboard creation.

## Hardware Requirements

* Minimum i3 processor is required.
* As this is a web application user needs a system with the internet connection.
* Disk space should be minimum 1 GB is required.

# 4. Design Details

## 4.1 Process Flow

Proposed methodology

## 4.2 Event log

The system should log every event so that the user will know what process is running internally.

**Initial Step-By-Step Description:**

1. The System identifies at what step logging required
2. The System should be able to log each and every system flow.
3. Developer can choose logging method. You can choose database logging/ File logging as well.
4. System should not hang even after using so many loggings. Logging just because we can easily debug issues, so logging is mandatory to do.

## 4.3 Error Handling

Should errors be encountered, an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal and intended usage.

# 5. Performance

## 5.1 Reusability

The code written and the components used should have the ability to be reused with no problems.

## 5.2 Application Compatibility

As the all the components are interrelated and should interact each other and changes in the data should be reflected immediately and graph also should change dynamically with the data changes.

## 5.3 Resource Utilization

When any task is performed, it will likely use all the processing power available until that function is finished.

## 5.4 Deployment

Deployment be done in one of these platforms

# A picture containing logo Description automatically generated 6. KPIs (Key Performance Indicators)

* Asad
* Asdas

# 7. Dashboards

Dashboards will be implemented to display and indicate certain KPIs and relevant indicators for the unveiled problems that if not addressed in time could cause catastrophes of unimaginable impact.

 Logo

Description automatically generated 

As and when, the system starts to capture the historical/periodic data for a user, the dashboards will be included to display charts over time with progress on various indicators or factors.

# 8. Conclusion