Design Doc Template

*Author(s): xyz*

*Date: 22/05/2019*

Revision: 0

Document Status: Draft [Draft, Completed, Submitted, Reviewed, Final]

Project Status: In-Progress [In Review, Approved, In-Progress, Completed]

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Revision | Description | Author |
| 22/05/2019 | 0 | Initial draft of the design doc template | xyz |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

TOC \o "1-3" \h \z \u [Introduction4](#_Toc9445198)

[Summary4](#_Toc9445199)

[Background4](#_Toc9445200)

[Definitions, Acronyms, and Abbreviations4](#_Toc9445201)

[Design Overview4](#_Toc9445202)

[Requirements4](#_Toc9445203)

[Documentation4](#_Toc9445204)

[Minimum Viable Product5](#_Toc9445205)

[Stretch goals5](#_Toc9445206)

[Future work5](#_Toc9445207)

[Architectural Diagrams5](#_Toc9445208)

[System Diagrams5](#_Toc9445209)

[Application Programming Interface5](#_Toc9445210)

[Recommendations5](#_Toc9445211)

[User Interface6](#_Toc9445212)

[Data Models and Storage6](#_Toc9445213)

[Service Operability6](#_Toc9445214)

[Key Performance Indicators6](#_Toc9445215)

[Service Level Objectives6](#_Toc9445216)

[Project Overview7](#_Toc9445217)

[Communication and Tracking7](#_Toc9445218)

[Risks7](#_Toc9445219)

[Milestones7](#_Toc9445220)

[Project Phases7](#_Toc9445221)

[Cost7](#_Toc9445222)

[Frequently Asked Question7](#_Toc9445223)

[References7](#_Toc9445224)

[Addendum8](#_Toc9445225)

# Introduction

## Summary

Our problem statement is “automated system to clean railway stations”.

## Background

The main operation is to clean the railway stations. It has been always a challenge to clean public areas and lack of resources. So we came up with the automated system which can operate itself and cleans the dust differentiating the useful objects.

Our solution/tool acquire properties like less power consumption, less manpower, faster cleaning, easily movable, less cost and is of small size.

## Definitions, Acronyms, and Abbreviations

Robot: This consists of a sensor which helps in detecting the garbage.

Sensor: An ultrasonic sensor helps us to detect the garbage and obstacle.

Suction machine: This has similar functionality of vacuum cleaner.

# Design Overview

## Requirements

This requires a vehicle which to keep hold of the suction machine.

### Documentation

If the project requires any wiki pages, code comments, presentations, etc. that information should be included here

## Minimum Viable Product

A detailed description of the deliverable for this project, this is the minimal functionality required for the project to be considered successful and should not include stretch goals or future work.

## Stretch goals

Stretch goals include functionality beyond the scope of the minimum viable product that should be include in the project should time and budget permit. Unlike future work, stretch goals would be smaller tasks for features in support of the minimum viable product.

## Future work

This may include ongoing support, expansion of the original scope, work that requires transitions in project ownership, or details of projects designed to be broken up into multiple phases.

# Architectural Diagrams

Sensors

Suction machine which is upon a vehicle

Detects the obstacle

Cleans the surroundings

# System Diagrams

Obstacle

Vehicle

Robot with sensors

Automated

Cleaning system

Suction machine

# Application Programming Interface

The main interface in our product is sensor. The sensor is used to detect the garbage.

## Recommendations

Using a versioned endpoint simplifies the process of making future backwards incompatible API changes;

# User Interface



Forward-B

Right-R

Left-L

Stop-S

# Data Models and Storage

For projects requiring messages queue such as Kafka, MySQL, etc.

Kafka

* How many partitions are needed for this topic?
* How many days of retention will be needed?
* What will the partitioning key become?
* How much data will be written to the topic during peak hours?
* What type of Kafka cluster will be needed? (E.g. aggregate, queuing, tracking, metrics, logging)

MySQL

* What does the table schema look like and how are they all tied together (provide a UML)?
* What sort of updates will be made to the tables?
* How will users make queries to the tables? (e.g. Complex joins, pre-filtering, single record gets)
* What the strategy for indexing?

# Service Operability

## Key Performance Indicators

Key performance indicators (KPI), describe how a service should be monitored and how its performance can be gauged. This would typically include an overview of the types of metrics an application will need to emit, call time, error rate, etc.

## Service Level Objectives

Service level objectives (SLOs), set targets for various KPI through alerts via email or SMS, these targets may provide early indicators of approaching a capacity limit, changes in load patterns through various phases of an application, changes in duration of offline processing, etc.

# Project Overview

## Communication and Tracking

## Risk

There is a risk of getting stucked in the public area during cleaning.

## Milestones

First week consists of survey and basic component gathering. In second week all the components will be designed and connected. During third week product testing is done. Other 2 weeks will be for reviews, feedbacks, surveys. Last week we will deploy the product.

## Project Phases

For projects that are better tracked and reported on in multiple phases because of extended timelines, external dependencies, etc.

## Cost

We can develop the system in the reasonable time as all the resources are available in the market but there may be a delay in approval from government.

# Frequently Asked Question

# References

Links to any supporting documentation, other projects, or reference material

# Addendum

Additional diagrams or details that do not particularly belong in the body of the design doc. This could also be a place to describe additional examples that would otherwise bloat the introduction section. More specifics on APIs could also be placed here for engineers to reference.