DRAFT SYSTEM DESIGN

DOCUMENT

**MECHATRONICS 4TB6**

**GROUP 2**

NATHAN FUJIMOTO

PRAKHAR GARG

JOSH GILMOUR

AARON JASS

TYLER JASS

FAUZIA KHANUM

JACK LIU

GAGAN SINGH

Table of Contents

[Table of Tables 3](#_Toc499297378)

[Revision History 4](#_Toc499297379)

[Purpose 5](#_Toc499297380)

[1 Overall Process Workflow 6](#_Toc499297381)

[2 How Steps are Completed 8](#_Toc499297382)

[2.1 Tools and Versions 9](#_Toc499297383)

[2.2 Special Instructions 10](#_Toc499297384)

[2.3 Standards Followed 10](#_Toc499297385)

[2.4 Delegation of Steps 11](#_Toc499297386)

[3 How Version Control is Being Used 12](#_Toc499297387)

[4 Dealing with Changes to Artifacts 13](#_Toc499297388)

[4.1 Bug Tracking/Change Request Tool 13](#_Toc499297389)

[4.2 Documentation of Change Requests/Bugs 13](#_Toc499297390)

[4.3 Classification of Changes 13](#_Toc499297391)

[4.4 Disposition of Changes 13](#_Toc499297392)

# Table of Tables

[Table 1 - Overall Process Workflow 6](#_Toc499297373)

[Table 2 - Tools and Versions 9](#_Toc499297374)

[Table 3 - Delegation of Steps 11](#_Toc499297375)

[Table 4 - Git Organization 12](#_Toc499297376)

# Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Rev** | **Author(s)** | **Description of Change** | **Peer Reviewed** | **Date** |
| - | T.Jass A.Jass N.Fujimoto  J.Gilmour  P.Garg  G.Singh  F.Khanum | Original Document |  |  |

# Purpose

# Scope

## 

## 

# 1 Context Diagram of Boundaries

2 System Component Diagrams

3 Monitored and Controlled Variables

4 Constants

5 Behaviour Overview

6 Component Details

7 Normal Operation

8 Undesired Event Handling

References

## 2.1 Tools and Versions

**Table 2 - Tools and Versions**

|  |  |  |
| --- | --- | --- |
| **Tool** | **Version** | **Purpose/Function** |
| **Software Tools:** | | |
| Google Drive | N/A | Storage accessible by entire group for deliverable documents. |
| Google Docs | N/A | Real-time collaboration. |
| Draw.io | N/A | Online web tool for model mapping and flow chart development. |
| Git | 2.15 | Software Version control, deliverable document, and internal document storage. |
| Ubuntu MATE | 3.0 | OS being used on our systems controller (Raspberry Pi 3B). |
| ROS (Robotics Operating System) | Kinetic | Robotics operating system library being used to develop project’s autonomous navigation capabilities. |
| Android Studio | 3.0 | Environment for Android App Development. |
| Autodesk Inventor | 2017 | Model all physical Robotics systems. |
| **Mechanical Tools:** | | |
| 3D Printer | N/A | Prototype any mechanical parts for motor and braking systems. |
| Hand Drill | N/A | Used for drilling holes |
| Saw | N/A | Used for cutting material |
| CNC Mill | N/A | Produce finished mechanical parts for motor and braking systems. |
| **Electrical Tools:** | | |
| Soldering Iron | N/A | Soldering wires/ circuit boards |
| Multimeter | N/A | Measure voltage and current |
| Power Supply | N/A | Powering prototype circuits/Motors |
| Oscilloscope | N/A | Measuring electrical signals |

## 2.2 Special Instructions

All document resources will be backed up to the corresponding version control systems.

## 2.3 Standards Followed

* All code should be commented to allow for quick and painless troubleshooting, our motto is ***“Comment as if the next person who reads your code is a serial killer”***
* When coding, no function should have more lines than what can be seen on your computer screen at one time
* No software or hardware tested by the person who implemented it can deemed “Working” (basically do not test your own work, get someone else to test it)
* Metric Units will be used for all modeled components

## 

## 

## 2.4 Delegation of Steps

**Legend**

1 - Primary Responsibility

2 - Secondary Responsibility

3 - Tertiary Responsibility

*Highlight - Persons Responsible for section of project*

**Table 3 - Delegation of Steps**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Josh** | **Aaron** | **Jack** | **Nathan** | **Fauzia** | **Prakhar** | **Tyler** | **Gagan** |
| **Phone application development** |  | 1/2 | 1 |  | 3 | 2 | 3 |  |
| **Autonomous navigation software** | 1 |  | 2 | 2 |  |  |  |  |
| **Low-level software** | 2 | 3 |  | 1 | 1 |  | 1 |  |
| **System integration** | 3 |  | 3 |  | 2 | 1\* |  | 3 |
| **Wheels and brakes modifications** |  | 1/2 |  | 3 |  | 3 | 2 | 1 |
| **Handle Bar (brake interference)** |  |  |  |  |  | 1 |  | 2 |

## 

## 

# 3 How Version Control is Being Used

Version control will be used to manage all source code, and as well as store documents. The source code will be organized into different directories based on the application. Documents including will be both deliverables and internal documents used for team knowledge sharing.

Our Git is organized into folders that relate to different aspects of the project. These folders are described in the table below.

**Table 4 - Git Organization**

|  |  |
| --- | --- |
| **Folder** | **Purpose** |
| CADs | Store the CAD files created of the walker and other components |
| Low level | Store documentation and files related to the software built for the Arduino and Raspberry Pi |
| Phone application code | Store documentation and files related to the phone application that is used to control and setup the walker. |
| ROS | Store documentation and files related to the ROS software on the walker |

# 4 Dealing with Changes to Artifacts

## 4.1 Bug Tracking/Change Request Tool

If we find that we are running into a lot of bugs and are spending quite a bit of time resolving them we will look into using Bugzilla. It is free for any number of users, compatible with Linux, MacOS, and Windows, and contain the following:

* Time tracking
* Reporting
* Duplicate bug detection
* Customization: custom fields to suit application

## 4.2 Documentation of Change Requests/Bugs

All of documentation is first drafted in Google Docs. Google Docs allows real time collaboration, commenting, and revision control. We can use this to document and track changes during the creation and drafting process. Once complete, we can export to word and finalize the document.

## 4.3 Classification of Changes

Changes will have different statuses such as:

* In queue (Waiting to be dealt with)
* Working (Currently being investigated)
* Resolved (Issue/Bug successfully dealt with)
* Unresolved (Unable to solve this issue/bug)

They should also have different levels of urgency based on how core the issue/bug is to the particular systems functionality.

## 4.4 Disposition of Changes

Change statuses will be verified and changed to different status by the appropriate party. For example, anyone can raise an issue and add them to “In Queue”. Depending on the given issue, a particular team may be responsible to fix that issue. Only that team can change that issue’s status to “Working”. Since it would bad practice to have the same person who fixed the issue, also verify the issue, someone else would be responsible for moving the issue to “Resolved”. If the person fixing the issue cannot fix it, they can move it to “Unresolved” so someone else can try it.