Software Design Document (SDD) Template

Software design is a process by which the software requirements are translated into a representation of software components, interfaces, and data necessary for the implementation phase. The SDD shows how the software system will be structured to satisfy the requirements. It is the primary reference for code development and, therefore, it must contain all the information required by a programmer to write code. The SDD is performed in two stages. The first is a preliminary design in which the overall system architecture and data architecture is defined. In the second stage, i.e. the detailed design stage, more detailed data structures are defined and algorithms are developed for the defined architecture.

This template is an annotated outline for a software design document adapted from the IEEE Recommended Practice for Software Design Descriptions. The IEEE Recommended Practice for Software Design Descriptions have been reduced in order to simplify this assignment while still retaining the main components and providing a general idea of a project definition report. For your own information, please refer to [IEEE Std 1016­1998](http://www.cs.concordia.ca/~ormandj/comp354/2003/Project/ieee-SDD.pdf)1 for the full IEEE Recommended Practice for Software Design Descriptions.



1 <http://www.cs.concordia.ca/~ormandj/comp354/2003/Project/ieee>­SDD.pdf

# (Team Name)

**(Project Title)**

# Software Design Document

Name (s): Lab Section: Workstation:

Date: (mm/dd/yyyy)

**TABLE OF CONTENTS**

1. [**INTRODUCTION 2**](#_heading=h.gjdgxs)
   1. [Purpose 2](#_heading=h.30j0zll)
   2. [Scope 2](#_heading=h.3znysh7)
   3. [Overview 2](#_heading=h.2et92p0)
   4. [Reference Material 2](#_heading=h.tyjcwt)
   5. [Definitions and Acronyms 2](#_heading=h.4i7ojhp)
2. [**SYSTEM OVERVIEW 2**](#_heading=h.1t3h5sf)
3. [**SYSTEM ARCHITECTURE 2**](#_heading=h.4d34og8)
   1. [Architectural Design 2](#_heading=h.2s8eyo1)
   2. [Decomposition Description 3](#_heading=h.17dp8vu)
   3. [Design Rationale 3](#_heading=h.2xcytpi)
4. [**DATA DESIGN 3**](#_heading=h.3rdcrjn)
   1. [Data Description 3](#_heading=h.26in1rg)
   2. [Data Dictionary 3](#_heading=h.lnxbz9)
5. [**COMPONENT DESIGN 3**](#_heading=h.35nkun2)
6. [**HUMAN INTERFACE DESIGN 4**](#_heading=h.1ksv4uv)
   1. [Overview of User Interface 4](#_heading=h.44sinio)
   2. [Screen Images 4](#_heading=h.2jxsxqh)
   3. [Screen Objects and Actions 4](#_heading=h.z337ya)
7. [**REQUIREMENTS MATRIX 4**](#_heading=h.3j2qqm3)
8. [**APPENDICES 4**](#_heading=h.1y810tw)

### INTRODUCTION

## Purpose

It is a matter of time before autonomous vehicles enter the routine life of human beings.

Today there are a number of issues that will be solved by an autonomous vehicle.

Autonomous cars will improve the quality of life and in particular problems

The aim of the project is to deal with and explore the technology of autonomous vehicles, to impart autonomous capabilities on a small vehicle.

## Scope

Our project implements autonomous capabilities over the Jetson nano – race car.

we will deal with real time image recognition algorithms , using TensorFlow library ( we will mainly focus on recognizing objects , obstacles ,routes and etc. ) .

By using the algorithms mentioned above

By using the algorithms mentioned above we will be able to take over the car movement and control , which will eventually give the jetson car autonomous capabilities.

## Overview

This document will provide an explanation on the system we intend to use it.

## Reference Material



### SYSTEM ARCHITECTURE

## Architectural Design

• Real time image recognition algorithms , using TensorFlow library (by using the built in camera on the jetson)

• Full control over the jetson car movement , using the image recognition program.

•Develop our own android app , and replace the current Linux operating system.

•Work with AirSim simulation program , to test our code with various scenarios.

### DATA DESIGN

## Data Description

Explain how the information domain of your system is transformed into data structures. Describe how the major data or system entities are stored, processed and organized. List any databases or data storage items.

## Data Dictionary

Alphabetically list the system entities or major data along with their types and descriptions. If you provided a functional description in Section 3.2, list all the functions and function parameters. If you provided an OO description, list the objects and its attributes, methods and method parameters.

### COMPONENT DESIGN

In this section, we take a closer look at what each component does in a more systematic way. If you gave a functional description in section 3.2, provide a summary of your algorithm for each function listed in 3.2 in procedural description language (PDL) or pseudocode. If you gave an OO description, summarize each object member function for all the objects listed in 3.2 in PDL or pseudocode. Describe any local data when necessary.

### HUMAN INTERFACE DESIGN

## Overview of User Interface

Describe the functionality of the system from the user’s perspective. Explain how the user will be able to use your system to complete all the expected features and the feedback information that will be displayed for the user.

## Screen Images

Display screenshots showing the interface from the user’s perspective. These can be hand­ drawn or you can use an automated drawing tool. Just make them as accurate as possible. (Graph paper works well.)

## Screen Objects and Actions

A discussion of screen objects and actions associated with those objects.

### REQUIREMENTS MATRIX

Provide a cross­reference that traces components and data structures to the requirements in your SRS document.

Use a tabular format to show which system components satisfy each of the functional requirements from the SRS. Refer to the functional requirements by the numbers/codes that you gave them in the SRS.

### APPENDICES

*This section is optional.*

Appendices may be included, either directly or by reference, to provide supporting details that could aid in the understanding of the Software Design Document.