Software Design Description
for the
Track and Control System

Version 1.0

Robert Moss, Aaron Periera, Matthew Shrago

February 18, 2014

Contents

1 Introduction

1.1 Purpose

Identify the purpose of this SDD and its intended audience. (e.g. This software design document describes the architecture and system design of XX. .).

1.2 Scope

Provide a description and scope of the software and explain the goals, objectives and benefits of your project. This will provide the basis for the brief description of your product.

1.3 Overview

Provide an overview of this document and its organization.

1.4 Reference Material

The list of references below are software documentation that we will be using:

1. OpenCV documentation: http://opencv.org/
   - FaceRecognizer API: http://docs.opencv.org/trunk/modules/contrib/doc/facerec/facerec_api.html
2. Windows API Index: http://msdn.microsoft.com/en-us/library/hh920508(v=vs.85).aspx
3. QT C++ documentation: http://qt-project.org/

1.5 Definitions, Acronyms, and Abbreviations

- Application Specific Definitions
  - TACS - Track and Control System
  - TM - Tracking Module
    - OT - Object Tracker
    - FRT - Facial Recognition Tracker
  - WCM - Windows Control Module
  - SM - Settings Module
- Industry Definitions
  - SRS - Software Requirements Specification
  - OpenCV - Open Computer Vision: An open source library for object tracking via the camera.
  - SQLite - A lightweight, low maintenance, self contained local database.
  - DB - Database
  - RGB - Red, Green, Blue color values.
  - HSV - Hue, Saturation, Value.
  - API - Application Programming Interface

- C++ - An object oriented programming language.
- GUI - Graphical User Interface
- QT - An API for building GUIs

2 System Overview

Give a general description of the functionality, context and design of your project. Provide any background information if necessary.

3 System Architecture

3.1 Architectural Design

Develop a modular program structure and explain the relationships between the modules to achieve the complete functionality of the system. This is a high level overview of how responsibilities of the system were partitioned and then assigned to subsystems. Identify each high level subsystem and the roles or responsibilities assigned to it. Describe how these subsystems collaborate with each other in order to achieve the desired functionality. Dont go into too much detail about the individual subsystems. The main purpose is to gain a general understanding of how and why the system was decomposed, and how the individual parts work together. Provide a diagram showing the major subsystems and data repositories and their interconnections. Describe the diagram if required.

3.2 Decomposition Description

Provide a decomposition...

3.3 Design Rationale

Discuss the rationale for selecting the architecture described in 3.1 including critical issues and trade/offs that were considered. You may discuss other architectures that were considered, provided that you explain why you didnt choose them.

4 Data Design

4.1 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.

4.2 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.

5 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.

6 Human Interface Design

6.1 Overview of User Interface

Describe the functionality of the system from the users 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.

6.2 Screen Images

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

6.3 Screen Objects and Actions

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

7 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.

A Appendix