SOFTWARE DESIGN DOCUMENT TEMPLATE (*modify as you deem appropriate*)

1.0 Introduction

The purpose of this software design document is to give a low-level description of the Person Identification System and to outline the design of the project. The following topics will be covered:

* Fill in
* Fill in
* Fill in
* Fill in
* Fill in
* Fill in

After reading this document, you should have a good understanding of how the Person Identification system functions.

1.1 Goals and objectives

The purpose of the Person Identification system is to be able to label unknown people with a unique “code”, and then be able to re-identify them from a non-overlapping camera that’s connected through a shared database. We also aim to provide destination prediction system that will function with multiple tracked people being tracked by the system at the same time. So, the primary objective is to create a functioning tracking and identification system.

This product must be able to accurately label and track unknown individuals quickly and efficiently. The system must be easy to setup. The user interface must be easy to understand and provide efficient navigation to each camera view. Other than these general design requirements, the application must also do the following functionalities:

* Fill in
* Fill in
* Fill in
* Fill in
* Fill in
* Fill in

1.2 Statement of scope

The Person Identification System runs using two main systems: the user interface that both receives user input and outputs the camera feeds, and a server-side database that communicates information throughout the camera network and updates the textual log. We will then break our features into three main groups: essential, which are mandatory to the basic functions of the system, desirable, which are additional features which we hope to complete, but are not sure about, and future requirements, which we have strong doubts about.

Essential Features

1. Feature
2. Feature
3. Feature
4. Feature

Desirable Features

1. Feature
2. Feature
3. Feature
4. Feature

Future Requirements

1. Feature
2. Feature
3. Feature
4. Feature

1.3 Software context

The Person Identification System will be a software package that the customer can purchase through major retail sources. Due to our high-skill based team, ample funding will be necessary to the design process of this advanced system. We are, however, prepared for the case where our funds are insufficient and if such a case were to arise, we would utilize an additional monthly membership fee on top of the software’s initial purchase price. We have also envisioned a micro-transaction system where the customer can pay for additional features such as increased camera count or larger textual log display.

1.4 Major constraints

The primary constraint for the Person Identification System is the allotted time for completion. There is a total of about three months from start to finish in which this time is devoted to development, testing, and documentation of this project. Also, the entire team has almost no experience with working on such a project, so a lot of time will be spent learning how to appropriately work together and allocate our resources effectively. Finally, every team member has a significant work load alongside this project, so the combination of little experience and a tight work schedule could result in less features on the initial release. This should have no impact on our core design features.

2.0 Data design

*A description of all data structures including internal, global, and temporary data structures.*

2.1 Internal software data structure

*Data structures that are passed among components of the software are described.*

2.2 Global data structure

*Data structured that are available to major portions of the architecture are described.*

2.3 Temporary data structure

*Files created for interim use are described.*

2.4 Database description

*Database(s) created as part of the application is (are) described. (Provide enough detail to create the database.)*

3.0 Architectural and component-level design

*A description of the program architecture is presented.*

3.1 System Structure

*A detailed description of the system structure chosen for the application is presented.*

3.1.1 Architecture diagram

*A pictorial representation, using a UML component diagram, of the architecture is presented.*

3.2 Description for Component n *(Your project may have multiple components)*

*A detailed description of each software component contained within the architecture is presented. Section 3.2 is repeated for each of n components.*

3.2.1 Processing narrative for component n

*A processing narrative for component n is presented. It should describe the responsibilities of the component.*

3.2.2 Component n interface description.

*A detailed description of the input and output interfaces for the component is presented.*

3.2.3 Component n processing detail

*A detailed algorithmic description for each component is presented.*

3.2.3.1 Design Class hierarchy for component n

3.2.3.2 Restrictions/limitations for component n

3.2.3.3 Performance issues for component n

3.2.3.4 Design constraints for component n

3.2.3.5 Processing detail for each operation of component n

3.2.3.5.1 Processing narrative for each operation

3.2.3.5.2 Algorithmic model for each operation

3.3 Dynamic Behavior for Component n

*A description of the interaction of the classes is presented.*

3.3.1 Interaction Diagrams

*A sequence diagram, for each use case the component realizes, is presented.*

4.0 User interface design

*A description of the user interface design of the software is presented.*

4.1 Description of the user interface

*A detailed description of user interface including screen images or prototype is presented.*

4.1.1 Screen images

*Representation of the interface from user's point of view.*

4.1.2 Objects and actions

*All screen objects and actions are identified.*

4.2 Interface design rules

*Conventions and standards used for designing/implementing the user interface are stated.*

4.3 Components available

*GUI components available for implementation are noted.*

4.4 UIDS description

*The user interface development system is described. (You might not have one for your project.)*

5.0 Restrictions, limitations, and constraints

The system requires a strong connection to the internet to work as the program retrieve data and video stream over the internet. A good connection is required to retrieve clear, real-time video feed from the cameras.

The system uses a network of ELP Mini USB cameras to record video feeds. Other cameras can be utilized on the program, however during the development process the ELP cameras were used primarily for testing and are currently the recommended device for the program.

The system is based on a network of cameras that connect to the main server. Each of these cameras must have a computer to run them, as well as an additional computer to run the server application. The team is currently achieving this with a network of laptops that run the cameras, then these laptops are connected to one additional computer that runs the server application

The system that is running the server application needs to have a 2.4 GHz Intel core i3 processor or higher to properly run the algorithm. If the system is running on a processor with less power than this the performance will be decreased, and the system will not run at the optimum performance levels.

The implementation of cameras by the user could potentially increase or decrease the systems effectiveness. For example, if to few cameras are used in a building or the placement of cameras results in ineffective recording of people’s locations on a site, the overall effectiveness of the system could be reduced.

6.0 Testing Issues

*Test strategy and preliminary test case specification are presented in this section.*

6.1 Performance bounds

*Special performance requirements are specified (if any).*

6.2 Identification of critical components

*Those components that are critical and demand particular attention during testing are identified.*