Senior Design Documentation Library  Architectural Design Specification

Department of Computer Science and Engineering
UNIVERSITY OF TEXAS AT ARLINGTON

Team AVIAR

Child-Parent Pairing System

Team Members:
Bhavin Mistry
Brandon Gallagher
Brandon Salahat
Eugen Eremia
Tony Kieu

Last Updated: 8 February 2013 @ 10:07:00 PM
# Table of Contents

Document Revision History ................................................................. 1

1 Introduction.......................................................................................... 2
  1.1 Overview ......................................................................................... 2
  1.2 Product Overview ......................................................................... 2
  1.3 Project Scope ................................................................................ 2
  1.4 Definitions and Acronyms ............................................................... 3

2 Meta-Architecture .............................................................................. 4
  2.1 Architectural Vision ........................................................................ 4
  2.2 Guiding Principles and Rules ......................................................... 4
  2.3 Tradeoffs ........................................................................................ 5

3 Architecture Overview ....................................................................... 6
  3.1 I/O Layer ....................................................................................... 6
  3.2 Application Layer .......................................................................... 6
  3.3 Data Storage Layer ........................................................................ 7
  3.4 Communication Layer ................................................................. 7
  3.5 Child Unit Layer ........................................................................... 7

4 I/O Layer ............................................................................................ 8
  4.1 Description .................................................................................... 8
  4.2 Purpose .......................................................................................... 9
  4.3 Function ........................................................................................ 9
  4.4 Dependencies ............................................................................... 9
  4.5 Interfaces ..................................................................................... 9
  4.6 Processing ................................................................................... 9
  4.7 Data .............................................................................................. 9
  4.8 Subsystems .................................................................................. 10

5 Application Layer ............................................................................. 12
  5.1 Description .................................................................................. 12
  5.2 Purpose ....................................................................................... 12
  5.3 Function ...................................................................................... 12
  5.4 Dependencies ............................................................................. 13
  5.5 Interfaces ................................................................................... 13
  5.6 Processing .................................................................................. 13
  5.7 Data ............................................................................................. 13
  5.8 Subsystems .................................................................................. 14
# Table of Contents

6 Data Storage Layer ........................................................................................... 17  
6.1 Description ............................................................................................... 18  
6.2 Purpose ................................................................................................. 18  
6.3 Function ............................................................................................... 18  
6.4 Dependencies ......................................................................................... 18  
6.5 Interfaces ............................................................................................. 18  
6.6 Processing ............................................................................................ 18  
6.7 Data ...................................................................................................... 19  
6.8 Subsystems ........................................................................................... 19  

7 Communication Layer .................................................................................. 22  
7.1 Description ........................................................................................... 22  
7.2 Purpose ............................................................................................... 22  
7.3 Function .............................................................................................. 22  
7.4 Dependencies ....................................................................................... 23  
7.5 Interfaces ............................................................................................. 23  
7.6 Processing ............................................................................................ 23  
7.7 Data ...................................................................................................... 23  
7.8 Subsystems ........................................................................................... 23  

8 Child Unit Layer .......................................................................................... 26  
8.1 Description ........................................................................................... 26  
8.2 Purpose ............................................................................................... 26  
8.3 Function .............................................................................................. 26  
8.4 Dependencies ....................................................................................... 26  
8.5 Interfaces ............................................................................................. 27  
8.6 Processing ............................................................................................ 27  
8.7 Data ...................................................................................................... 27  
8.8 Subsystems ........................................................................................... 27  

9 Operating System Dependencies .................................................................. 30  

10 Inter-Subsystem Data Flow ......................................................................... 31  
10.1 Overview ............................................................................................ 31  
10.2 Producer-Consumer Relationship .......................................................... 31  

11 Requirements Mapping ............................................................................... 34  

12 Testing Considerations ............................................................................... 36  
12.1 Overview ............................................................................................ 36  
12.2 Testing Approach ................................................................................ 36  
12.3 I/O Layer ............................................................................................ 36  
12.4 Application Layer ............................................................................... 37  
12.5 Data Storage Layer ............................................................................. 38  
12.6 Communication Layer ........................................................................ 38  
12.7 Child Unit Layer ................................................................................ 39
List of Tables

Table 4-1: I/O Layer Interfaces .................................................................................................. 9
Table 4-2: Input Subsystem Interfaces ....................................................................................10
Table 4-3: Output Subsystem Interfaces ...................................................................................11
Table 5-1: Application Layer Interfaces .....................................................................................13
Table 5-2: Data Processing Subsystem .....................................................................................14
Table 5-3: Staff Management Subsystem Interfaces .................................................................15
Table 5-4: Unit Subsystem Interfaces .......................................................................................16
Table 6-1: Application Layer Interfaces .....................................................................................18
Table 6-2: Authentication Subsystem Interfaces .......................................................................19
Table 6-3: Database Subsystem Interfaces ....................................................................................20
Table 7-1: Communication Interfaces Subsystem .................................................................23
Table 7-2: USB Driver Subsystem Interfaces ..........................................................................24
Table 7-3: Encoding Subsystem Interfaces ...............................................................................24
Table 7-4: Transmission Subsystem Interfaces ........................................................................25
Table 8-1: Child Unit Layer Interfaces .......................................................................................27
Table 8-2: Radio Subsystem Interfaces .....................................................................................27
Table 8-3: Processing Subsystem Interfaces ............................................................................28
Table 8-4: Notification Subsystem Interfaces ..........................................................................29
Table 10-1: Data Elements .......................................................................................................32
Table 10-2: Producer-Consumer Relationship Matrix ............................................................33
Table 11-1: Requirements Mapping ..........................................................................................35
List of Figures

Figure 1-1: Project Scope Overview ................................................................. 3
Figure 3-1: Architecture Overview ............................................................... 6
Figure 4-1: I/O Layer .................................................................................. 8
Figure 5-1: Application Layer ................................................................... 12
Figure 6-1: Data Storage Layer ................................................................ 17
Figure 6-2: Database Entity Relationship Diagram .............................. 17
Figure 7-1: Communication Layer ............................................................ 22
Figure 8-1: Child Unit Layer .................................................................... 26
Figure 10-1: Data Flow between Layers and Subsystems .................... 31
## Document Revision History

<table>
<thead>
<tr>
<th>Revision #</th>
<th>Revision Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>01/20/2013</td>
<td>Initial Compilation</td>
</tr>
<tr>
<td>0.2</td>
<td>01/30/2013</td>
<td>Edit based on team MASS recommendations</td>
</tr>
<tr>
<td>0.3</td>
<td>01/31/2013</td>
<td>ADS Gate Review</td>
</tr>
<tr>
<td>1.0</td>
<td>02/08/2013</td>
<td>Final Revision</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 Overview
The following document specifies the architecture design for the Child-Parent Pairing System (CPPS). It lays out the conceptual models that will guide the construction of the CPPS system.

1.2 Product Overview
The CPPS system is intended for day cares and other facilities that transfer custody of children frequently throughout the day. It identifies and keeps record of the individuals dropping off and picking up the children. It will also notify children when their parents have arrived to pick them up.

This system will mitigate the human error in the transfer of custody process, and will help drastically reduce the number of children that are mishandled by a facility.

All components of this system will operate as independently as possible to ensure maximum reliability. This will be accomplished by passing data packets between subsystems. The system will also be designed in such a way that it can be expanded to handle more children, and customized for a facility should they request it.

The system will consist of a charging station, a radio transmitter, a user input panel (included a keypad and card reader), and a child wearable unit. The child unit will visually alert the child when it is time to be picked up. User input will be handled by a magnetic card reader, a keypad, and a USB camera.

1.3 Project Scope
The CPPS produced by Team AVIAR is intended for a small child care facility, and will be prototyped with a limited number of child units. This product mitigates human error during child transfer by requiring the parent to verify their identity by swiping their state ID and entering a given PIN number. The system then records this information into the child's record. When the child is being picked up, this information will be verified, and the child will be notified via a visual alert from the child unit. A process diagram is shown in the following figure.
Figure 1-1: Project Scope Overview

1.4 Definitions and Acronyms

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPPS</td>
<td>Child-Parent Pairing System</td>
</tr>
<tr>
<td>I</td>
<td>I/O Layer</td>
</tr>
<tr>
<td>A</td>
<td>Application Layer</td>
</tr>
<tr>
<td>D</td>
<td>Data Storage Layer</td>
</tr>
<tr>
<td>C</td>
<td>Communication Layer</td>
</tr>
<tr>
<td>U</td>
<td>Child Unit Layer</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>USB</td>
<td>Universal Serial Bus</td>
</tr>
</tbody>
</table>
2 Meta-Architecture

2.1 Architectural Vision

The CPPS system is composed of hardware and software components. On the hardware side, CPPS will have a radio transmitter and child unit receiver. On the software side, CPPS will have GUI software which interacts with user through an input panel (keypad & card reader). All the activities on the software are display on a standard computer monitor. The input panel will take all of the information required by the parent to verify their identity. The software will then record this data and pass a packet to the radio transmitter. This transmitter will call out the requested child unit, which will alert the child it is time to be picked up.

The construction of this system will be based around 5 architectural layers: The I/O Layer, the Application Layer, the Data Storage Layer, the Communication Layer, and the Child Unit Layer. Each layer will be designed to function independently from the other, and will communicate amongst them by passing data packets. This should ensure easier implementation and should help ensure the system is built in a reliable manner.

2.2 Guiding Principles and Rules

2.2.1 Reliability

This system is intended to be reliable for its intended use. It should be secure and work as intended.

2.2.2 Modifiability

This system is intended to be easily modified. The facility using it should be able to add as many child units as they need to it and the code should be written in a manner so that it may be changed and personalized for facilities in the future.

2.2.3 Scalability

This system should function properly regardless of how many child units are being used. It should be designed in such a way that it can easily be made to fit the needs of a child care facility regardless of size.
2.3 Tradeoffs

This project originally had a 6th layer, the Physical Layer, in order to be consistent with typical network layering. Members of Team AVIAR decided to remove this layer and combine it with the Communication Layer in order to simplify the system architecture in a way that it would be easier to expand later.

In the early stages of requirements planning this project was designed to work via a web system using routers and wireless receivers. This was ruled out as impractical because internet protocol receivers would be difficult to power in a manner that would be suitable for the wearable child units. Using a web interface also introduced more security complications that would increase the complexity of the system. It was opted to use an analog, closed radio system instead.
3 Architecture Overview

The Child-Parent Pairing System contains five layers: I/O, Application, Data Storage, Communication and Child Unit. Figure 3-1 represents the architecture overview with data flow.

3.1 I/O Layer
The I/O (Input/Output) Layer is responsible for handling staff/parent input and GUI output of the CPPS. The I/O Layer takes input from the staff or parent, performs minimum formatting on it and passes it on to the Application Layer. Also, I/O Layer takes messages from the Application Layer and passes it on to the GUI.

3.2 Application Layer
The Application Layer is the core framework that controls all the activities in the CPPS. The data for Application Layer come from I/O Layer and Data Storage Layer.
For system management, Application Layer will be responsible to store staff, user and device data to the Data Storage Layer. For pairing events, Application Layer will request resources from Data Storage Layer before talking to Communication Layer.

### 3.3 Data Storage Layer

The Data Storage Layer stores all relevant staff authentication data, parent’s data, children data, pairing data, and units’ data. It interacts with the Services Subsystem in the Application Layer. When any of the services request data in order to perform their actions, the Data Storage Layer replies with the inquired information. The same when services from the Application Layer need to store data, the Data Storage Layer will make sure that it is done.

### 3.4 Communication Layer

The Communication Layer of the CPPS will process the raw data packets received from the Application Layer for testing purpose or notification to a particular child unit. This layer will encode the data packets and then they will be transmitted securely to the Child Unit Layer to notify.

### 3.5 Child Unit Layer

The Child Unit Layer will be responsible for securely and accurately notifying arrival of the parent to the right child. This layer will receive encoded data packets from Communication Layer and then it will be processed to notify the right child unit.
4 I/O Layer

4.1 Description

The I/O (Input/Output) Layer is designated as an interface of all CPPS peripheral devices: Webcam (camera), Keyboard/Mouse, Card Reader, Keypad and Display (Monitor). This layer takes inputs from staff members or parents and displays system activities. The I/O Layer interacts with CPPS users through a GUI framework. The layer contains two different subsystems: Input Subsystem and Output Subsystem.
4.2 Purpose
The I/O Layer takes inputs from CPPS users and displays appropriate output back to the users.

4.3 Function
- Manage all CPPS peripheral devices status.
- Handle user data inputs and outputs uniformly for further processing.
- Handle all inbound and outbound data objects asynchronously.

4.4 Dependencies
The I/O Layer is dependent on all CPPS peripheral devices and the Application Layer.

4.5 Interfaces

<table>
<thead>
<tr>
<th>Type</th>
<th>Source</th>
<th>Destination</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>External</td>
<td>Webcam</td>
<td>Input Subsystem</td>
<td>Digital Image Format</td>
</tr>
<tr>
<td>External</td>
<td>Keyboard/Mouse</td>
<td>Input Subsystem</td>
<td>Text String/Events</td>
</tr>
<tr>
<td>External</td>
<td>Card Reader</td>
<td>Input Subsystem</td>
<td>Text String</td>
</tr>
<tr>
<td>External</td>
<td>Keypad</td>
<td>Input Subsystem</td>
<td>Text String</td>
</tr>
<tr>
<td>External</td>
<td>Application Layer</td>
<td>Output Subsystem</td>
<td>Output Objects</td>
</tr>
<tr>
<td>External</td>
<td>Input Subsystem</td>
<td>Application Layer</td>
<td>Input Objects</td>
</tr>
</tbody>
</table>

Table 4-1: I/O Layer Interfaces

4.6 Processing
All data entered by users at the I/O Layer will be encapsulated and sent to Application Layer for further use in down line systems – namely, the Data Storage Layer and the Communication Layer. The data from Application Layer will be formatted and displayed appropriately to the users via the I/O Layer’s Output Subsystem.

4.7 Data
- Digital Image Format
- Text Strings
- Action Events
- Output Objects
- Input Objects
4.8 Subsystems

4.8.1 Input Subsystem

4.8.1.1 Purpose
Gathers input from users and sends it to the Application Layer.

4.8.1.2 Function
Receives input data uniformly for further processing

4.8.1.3 Dependencies
CPPS Peripheral devices: Webcam, Keyboard/Mouse, Card Reader and Keypad

4.8.1.4 Interfaces

<table>
<thead>
<tr>
<th>Source</th>
<th>Data</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Webcam</td>
<td>Digital Image Format</td>
<td>Input Subsystem</td>
</tr>
<tr>
<td>Keyboard/Mouse</td>
<td>Text Strings/Events</td>
<td>Input Subsystem</td>
</tr>
<tr>
<td>Card Reader</td>
<td>Text Strings</td>
<td>Input Subsystem</td>
</tr>
<tr>
<td>Keypad</td>
<td>Text Strings</td>
<td>Input Subsystem</td>
</tr>
<tr>
<td>Input Subsystem</td>
<td>Input Objects</td>
<td>Application Layer</td>
</tr>
</tbody>
</table>

Table 4-2: Input Subsystem Interfaces

4.8.1.5 Processing
When there is inbound data from peripheral devices, the Input Subsystem encapsulates the data and sends it down to the Application Layer.

4.8.1.6 Data
- Digital Image Format
- Text String/Events
- Input Objects

4.8.2 Output Subsystem

4.8.2.1 Purpose
Display the necessary outputs from the Application Layer to the user.
4.8.2.2 Function
Receive output objects from the Application Layer and update the GUI accordingly.

4.8.2.3 Dependencies
Application Layer

4.8.2.4 Interfaces

<table>
<thead>
<tr>
<th>Source</th>
<th>Data</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Layer</td>
<td>Output Objects</td>
<td>Output Subsystem</td>
</tr>
</tbody>
</table>

Table 4-3: Output Subsystem Interfaces

4.8.2.5 Processing
When there is inbound data from Application Layer (in the form of output objects), the Output Subsystem formats it and displays it accordingly on the display.

4.8.2.6 Data
- Output Objects
5 Application Layer

5.1 Description
The Application Layer is designated as a common interface between the I/O, Data Storage and Communication Layers. This layer is responsible for routing data objects to the software and hardware infrastructures of the CPPS. The layer contains three subsystems: Data Processing Subsystem, Data Management Subsystem and Unit Subsystem.

5.2 Purpose
Route the data objects to the appropriate layers.

5.3 Function
- Parse the data from the I/O Layer in the form of input objects.
- Send the output data (results) to the I/O Layer in the form of output objects.
Send the pairing information to the Communication Layer in the form of Unit ID.
Request and handle data from the Data Storage Layer.

### 5.4 Dependencies
- I/O Layer
- Data Storage Layer

### 5.5 Interfaces

<table>
<thead>
<tr>
<th>Type</th>
<th>Source</th>
<th>Destination</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>External</td>
<td>I/O Layer</td>
<td>Data Processing Subsystem</td>
<td>Input Objects</td>
</tr>
<tr>
<td>External</td>
<td>Data Processing Subsystem</td>
<td>I/O Layer</td>
<td>Output Objects</td>
</tr>
<tr>
<td>External</td>
<td>Unit Subsystem</td>
<td>Communication Layer</td>
<td>Unit ID</td>
</tr>
<tr>
<td>External</td>
<td>Data Management Subsystem</td>
<td>Data Storage Layer</td>
<td>Requested Data</td>
</tr>
<tr>
<td>External</td>
<td>Data Storage Layer</td>
<td>Data Management Subsystem</td>
<td>Data Responses</td>
</tr>
<tr>
<td>Internal</td>
<td>Data Processing Subsystem</td>
<td>Data Management Subsystem</td>
<td>Data Objects</td>
</tr>
<tr>
<td>Internal</td>
<td>Data Management Subsystem</td>
<td>Data Processing Subsystem</td>
<td>Data Objects</td>
</tr>
<tr>
<td>Internal</td>
<td>Data Processing Subsystem</td>
<td>Unit Subsystem</td>
<td>Data Objects</td>
</tr>
</tbody>
</table>

**Table 5-1: Application Layer Interfaces**

### 5.6 Processing
For any incoming data from the I/O Layer, the Application Layer performs authentication and error checking processes with the Data Storage Layer. If the data does not meet the process requirements, the Application Layer informs the user that the request/response is held or omitted. If the data meets the requirements, it will be extracted for either user management or pairing purpose.

### 5.7 Data
- Input Objects
- Output Objects
5.8 Subsystems

5.8.1 Data Processing Subsystem

5.8.1.1 Purpose
Route data to the corresponding subsystems and layers

5.8.1.2 Function
- Route Data Objects to Data Management Subsystem
- Route Data Objects to Unit Subsystem
- Route Output Objects to I/O Layer

5.8.1.3 Dependencies
- I/O Layer
- Data Management Subsystem

5.8.1.4 Interfaces

<table>
<thead>
<tr>
<th>Source</th>
<th>Data</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/O Layer</td>
<td>Input Objects</td>
<td>Data Processing Subsystem</td>
</tr>
<tr>
<td>Data Processing Subsystem</td>
<td>Output Objects</td>
<td>I/O Layer</td>
</tr>
<tr>
<td>Data Processing Subsystem</td>
<td>Data Objects</td>
<td>Unit Subsystem</td>
</tr>
<tr>
<td>Data Processing Subsystem</td>
<td>Data Objects</td>
<td>Data Management Subsystem</td>
</tr>
<tr>
<td>Data Management Subsystem</td>
<td>Data Objects</td>
<td>Data Processing Subsystem</td>
</tr>
</tbody>
</table>

Table 5-2: Data Processing Subsystem

5.8.1.5 Processing
The Data Processing Subsystem takes input from the I/O Layer and decides which subsystems will handle the requests. It also passes them to the appropriate subsystems inside the Application Layer.
5.8.1.6 Data

- Input Objects
- Output Objects
- Data Objects

5.8.2 Data Management Subsystem

5.8.2.1 Purpose

The Data Management Subsystem is a set of procedures that enables storing, modifying and extracting data objects with the Data Storage Layer.

5.8.2.2 Function

The Data Management Subsystem provides catalog interoperability between the Data Processing Subsystem and the Data Storage Layer.

5.8.2.3 Dependencies

- Data Processing Subsystem
- Data Storage Layer

5.8.2.4 Interfaces

<table>
<thead>
<tr>
<th>Source</th>
<th>Data</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Processing Subsystem</td>
<td>Data Objects</td>
<td>Data Management Subsystem</td>
</tr>
<tr>
<td>Data Management Subsystem</td>
<td>Data Objects</td>
<td>Data Processing Subsystem</td>
</tr>
<tr>
<td>Data Management Subsystem</td>
<td>Requested Data</td>
<td>Data Storage Layer</td>
</tr>
<tr>
<td>Data Storage Layer</td>
<td>Data Responses</td>
<td>Data Management Subsystem</td>
</tr>
</tbody>
</table>

Table 5-3: Staff Management Subsystem Interfaces

5.8.2.5 Processing

The Data Management Subsystem maintains a dictionary of data collection information with keywords and attributes used by the Data Processing Subsystem and Data Storage Layer in achieving interoperability. The
dictionary also contains collection attribute and keyword mappings used to translate requests from the Data Processing Subsystem to the Data Storage Layer.

5.8.2.6 Data
- Data Objects
- Requested Data

5.8.3 Unit Subsystem

5.8.3.1 Purpose
Handle the pairing and testing events for the Child Unit Layer.

5.8.3.2 Function
Extract the Unit ID of the Data Objects and send it to the Communication Layer.

5.8.3.3 Dependencies
- Data Processing Subsystem

5.8.3.4 Interfaces

<table>
<thead>
<tr>
<th>Source</th>
<th>Data</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Processing Subsystem</td>
<td>Data Objects</td>
<td>Unit Subsystem</td>
</tr>
<tr>
<td>Unit Subsystem</td>
<td>Unit ID</td>
<td>Communication Layer</td>
</tr>
</tbody>
</table>

Table 5-4: Unit Subsystem Interfaces

5.8.3.5 Processing
For Pairing and Testing, this subsystem extracts the Unit ID and sends it to the Child Unit Layer.

5.8.3.6 Data
- Data Objects
- Unit ID
6 Data Storage Layer

Figure 6-1: Data Storage Layer

Figure 6-2: Database Entity Relationship Diagram
6.1 Description
The Data Storage Layer stores all relevant staff authentication data, parent’s data, children data, pairing data, and units’ data. It interacts with the Data Management Subsystem in the Application Layer. When data is requested from anywhere in the system, the Data Storage Layer replies with the appropriate information. Conversely, when data needs to be stored the Data Storage Layer will make sure that it is done as well. The Data Storage Layer consists of two subsystems: An Authentication Subsystem and the actual Database Subsystem.

6.2 Purpose
The Data Storage Layer’s purpose is to store persistent data about the objects in the Application Layer.

6.3 Function
The Data Storage Layer provides persistent data storage to the Child-Parent Pairing System by using a database abstraction pattern. The subsystems in the Application Layer request information from the Data Storage Layer, and in the same way the subsystems in the Application Layer send data for storage as well.

6.4 Dependencies
The Data Storage Layer is dependent on the Application Layer.

6.5 Interfaces

<table>
<thead>
<tr>
<th>Type</th>
<th>Source</th>
<th>Destination</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal</td>
<td>Authentication Subsystem</td>
<td>Database Subsystem</td>
<td>Query</td>
</tr>
<tr>
<td>Internal</td>
<td>Database Subsystem</td>
<td>Authentication Subsystem</td>
<td>Query Result</td>
</tr>
<tr>
<td>External</td>
<td>Application Layer</td>
<td>Authentication Subsystem</td>
<td>Requested Data</td>
</tr>
<tr>
<td>External</td>
<td>Authentication Subsystem</td>
<td>Application Layer</td>
<td>Data Responses</td>
</tr>
</tbody>
</table>

Table 6-1: Database Layer Interfaces

6.6 Processing
The Authentication Subsystem gets data from the Data Management Subsystem in the Application Layer, authenticates that the correct credentials have been provided along with the request, and queries the Database for the required data. The Database Subsystem then
returns a set of records, which in turn are passed back to the Application Layer through the Authentication Subsystem.

6.7 Data

- Requested Data
- Data Responses
- Query
- Query Result

6.8 Subsystems

6.8.1 Authentication Subsystem

6.8.1.1 Purpose

This subsystem acts as a checkpoint to ensure that all requests for data are authenticated properly. It also contains the appropriate processes and algorithms for salting and hashing of sensitive personal data.

6.8.1.2 Function

Receive requests for data from the Data Management Subsystem in the Application Layer, authenticate when necessary, and forward those requests as queries to the Database Subsystem. The Database responds and the returned data is passed back to the Data Management Subsystem.

6.8.1.3 Dependencies

Database Subsystem and Staff Management Subsystem in the Application Layer

6.8.1.4 Interfaces

<table>
<thead>
<tr>
<th>Source</th>
<th>Data</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Layer</td>
<td>Requested Data</td>
<td>Authentication Subsystem</td>
</tr>
<tr>
<td>Authentication Subsystem</td>
<td>Data Responses</td>
<td>Application Subsystem</td>
</tr>
<tr>
<td>Authentication Subsystem</td>
<td>Query</td>
<td>Database Subsystem</td>
</tr>
<tr>
<td>Database Subsystem</td>
<td>Query Result</td>
<td>Authentication Subsystem</td>
</tr>
</tbody>
</table>

Table 6-2: Authentication Subsystem Interfaces
6.8.1.5 Processing

The Authentication Subsystem receives data for storage and data requests from the Data Management Subsystem in the Application Layer. It then queries the Database appropriately. The Database returns the correct responses, which in turn are passed back to the Data Management Subsystem in the Application Layer.

6.8.1.6 Data

- Requested Data
- Data Responses
- Query
- Query Result

6.8.2 Database Subsystem

6.8.2.1 Purpose

This subsystem provides the actual database for storage of all records in the Child-Parent Pairing System.

6.8.2.2 Function

Receive queries from the Authentication Subsystem, and either store data or return records based on the query.

6.8.2.3 Dependencies

Authentication Subsystem

6.8.2.4 Interfaces

<table>
<thead>
<tr>
<th>Source</th>
<th>Data</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication Subsystem</td>
<td>Query</td>
<td>Database Subsystem</td>
</tr>
<tr>
<td>Database Subsystem</td>
<td>Query Result</td>
<td>Authentication Subsystem</td>
</tr>
</tbody>
</table>

Table 6-3: Database Subsystem Interfaces

6.8.2.5 Processing

The Database Subsystem performs standard database transactions provided by MySQL. The Authentication Subsystem sends either storage requests or record request – the database handles those appropriately based on the query, and returns records when requested.
6.8.2.6 Data

- Query
- Query Result
7 Communication Layer

7.1 Description

The Communication Layer of the CPPS system is responsible for taking in information from the Application Layer and converting into suitable and readable format for the Child Unit Layer. Communication Layer has 3 subsystems: USB Driver, Encoding, and Transmission Subsystem.

7.2 Purpose

The purpose of the Communication Layer is to encode the information received from the Application Layer and then transmit it to the Child Unit Layer.

7.3 Function

The main function of the communication is to send the information to the correct child unit based on the information received from the Application Layer. USB Driver Subsystem will be used in order to send the information from the GUI to the encoder which will encode the ID. Then Encoding Subsystem will receive digital data packets which will be converted into suitable format for the radio transmission. And then finally it will be broadcasted to the Child Unit Layer using a physical medium.
7.4 Dependencies

The Communication Layer requires that the Application Layer sends valid unit ID.

7.5 Interfaces

<table>
<thead>
<tr>
<th>Type</th>
<th>Source</th>
<th>Destination</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>External</td>
<td>Application Layer</td>
<td>USB System</td>
<td>Unit ID</td>
</tr>
<tr>
<td>External</td>
<td>Transmission Subsystem</td>
<td>Child Unit Layer</td>
<td>Encoded Data Packets</td>
</tr>
</tbody>
</table>

Table 7-1: Communication Interfaces Subsystem

7.6 Processing

The Communication Layer will process the information received from Application Layer through USB driver by encoding and transmitting it to the Child Unit Layer.

7.7 Data

Digital data, processed data

7.8 Subsystems

7.8.1 USB Driver Subsystem

7.8.1.1 Purpose

The purpose of the USB Driver Subsystem is to pass the data received from Application Layer to the encoder of the Communication Layer.

7.8.1.2 Function

Received information from Application Layer is encoded and then transmitted to the Child Unit Layer.

7.8.1.3 Dependencies

Receive correct unit ID from Application Layer.

7.8.1.4 Interfaces
7.8.1.5 Processing
Process the Unit ID

7.8.1.6 Data
Unit ID

7.8.2 Encoding Subsystem

7.8.2.1 Purpose
The purpose of the Encoding Subsystem is to receive the digital data packets and converts them to a format suitable for wireless radio transmission.

7.8.2.2 Function
Encoding Subsystem encodes the digital packets in suitable format for the child unit.

7.8.2.3 Dependencies
Digital data packets from the USB Driver Subsystem

7.8.2.4 Interfaces

<table>
<thead>
<tr>
<th>Source</th>
<th>Data</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>USB Driver Subsystem</td>
<td>Digital Data Packet</td>
<td>Encoding Subsystem</td>
</tr>
<tr>
<td>Encoding Subsystem</td>
<td>Analog Data Packet</td>
<td>Transmission Subsystem</td>
</tr>
</tbody>
</table>

Table 7-3: Encoding Subsystem Interfaces

7.8.2.5 Processing
Encode the digital data packets for transmission to the child unit.

7.8.2.6 Data
Digital data packets
7.8.3 Transmission Subsystem

7.8.3.1 Purpose
The purpose of the Transmission Subsystem is to receive the encoded packets and broadcasts them through the physical medium to the Child Unit Layer. The digital data packets need to be transmitted to the Child Unit Layer in order to notify the right child unit.

7.8.3.2 Function
The Transmission Subsystem will receive the encoded packets and then they will be broadcasted to the Child Unit Layer through the physical layer.

7.8.3.3 Dependencies
Processed data packets from Encoding Subsystem

7.8.3.4 Interfaces

<table>
<thead>
<tr>
<th>Source</th>
<th>Data</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encoding Subsystem</td>
<td>Analog Data Packet</td>
<td>Transmission Subsystem</td>
</tr>
<tr>
<td>Transmission Subsystem</td>
<td>Analog Data Packet</td>
<td>Child Unit Layer</td>
</tr>
</tbody>
</table>

Table 7-4: Transmission Subsystem Interfaces

7.8.3.5 Processing
Transmitting encoded analog data packets to the Child Unit Layer.

7.8.3.6 Data
Analog data packet
8 Child Unit Layer

8.1 Description

The Child Unit Layer of the CPPS will be responsible for notifying the child that parent has arrived to pick them up. Subsystems of this layer are: radio subsystem, processing subsystem, and notification subsystem.

8.2 Purpose

The purpose of the Child Unit Layer is to notify the child arrival of the parent by lighting up the LEDs on the unit.

8.3 Function

The function of the Child Unit Layer receives data packets from the Communication Layer and it will light up the LED lights through the Child Unit Layer.

8.4 Dependencies

The Child Unit Layer is dependent on the data packets received from the Communication Layer.
8.5 Interfaces

<table>
<thead>
<tr>
<th>Type</th>
<th>Source</th>
<th>Destination</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>External</td>
<td>Communication Layer</td>
<td>Radio Subsystem</td>
<td>Analog Data Packet</td>
</tr>
</tbody>
</table>

Table 8-1: Child Unit Layer Interfaces

8.6 Processing

The Child Unit Layer processes the analog data packets.

8.7 Data

The Child Unit Layer receives the encoded analog data packets which will be processed to determine if the unit should light up the LEDs or not.

8.8 Subsystems

8.8.1 Radio Subsystem

8.8.1.1 Purpose

The purpose of radio subsystem is to receive the radio waves from the Communication Layer and then it will be sent to the next subsystem to process.

8.8.1.2 Function

This Radio Subsystem will receive the data in packets to be processed for the child unit.

8.8.1.3 Dependencies

Analog data packets from the Communication Layer

8.8.1.4 Interfaces

<table>
<thead>
<tr>
<th>Source</th>
<th>Data</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication Layer</td>
<td>Analog Data Packet</td>
<td>Radio Subsystem</td>
</tr>
<tr>
<td>Radio Subsystem</td>
<td>Analog Data Packet</td>
<td>Processing Subsystem</td>
</tr>
</tbody>
</table>

Table 8-2: Radio Subsystem Interfaces
8.8.1.5 Processing
Analog data packets received from Communication Layer

8.8.1.6 Data
Analog data packets

8.8.2 Processing Subsystem

8.8.2.1 Purpose
The purpose of the processing subsystem is to determine if Notification Subsystem system needs to be triggered or not depending on the processed data that was sent from the radio subsystem.

8.8.2.2 Function
Analog data packets will be processed for verification of the purpose.

8.8.2.3 Dependencies
Analog data packets from the Radio Subsystem

8.8.2.4 Interfaces

<table>
<thead>
<tr>
<th>Source</th>
<th>Data</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radio Subsystem</td>
<td>Analog Data Packet</td>
<td>Processing Subsystem</td>
</tr>
<tr>
<td>Processing Subsystem</td>
<td>Digital Data Packet</td>
<td>Notification Subsystem</td>
</tr>
</tbody>
</table>

Table 8-3: Processing Subsystem Interfaces

8.8.2.5 Processing
Analog data packets from Radio Subsystem and converting into the digital data packet

8.8.2.6 Data
Analog data packet and digital data packet

8.8.3 Notification Subsystem

8.8.3.1 Purpose
The purpose of the notification subsystem is to illuminate the LEDs on child unit to notify the child.
8.8.3.2 Function
Based on the digital data packets, correct child unit’s LEDs will come on.

8.8.3.3 Dependencies
Digital data packets from the Processing Subsystem

8.8.3.4 Interfaces

<table>
<thead>
<tr>
<th>Source</th>
<th>Data</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processing Subsystem</td>
<td>Digital Data Packet</td>
<td>Notification Subsystem</td>
</tr>
</tbody>
</table>

Table 8-4: Notification Subsystem Interfaces

8.8.3.5 Processing
Digital data packets from radio subsystem to notify

8.8.3.6 Data
Digital data packet
The Child-Parent Pairing System will require a workstation at the check-in site in order to facilitate the check-in and retrieval operations. The workstation requires having Java 1.6 or greater. The Operating System can be either Windows or Ubuntu. This workstation will also require at least three unused USB ports in order to connect the magnetic card reader and 10-key keypad that will be used for authentication and the radio transmitter that will be used for sending radio signals to the hardware bracelets. The database, user interface, and processing functionality at the check-in station will all be provided as a part of the packaged, delivered software product. The database will be developed on the MySQL platform. The user interface will be developed as a pure Java solution, which will facilitate our cross-platform compatibility.

The radio transmitter will run on a PIC 18F452 chip. The hardware bracelets will run as embedded systems, which will be pre-installed by Team AVIAR at the time of product delivery or shipping. The bracelets will each contain a PIC microchip processor that will handle the incoming radio signals, and initiate the light sequence of the embedded LEDs.
10 Inter-Subsystem Data Flow

10.1 Overview

Relationship Mapping describes the data flow between layers and subsystems of the Child-Parent Pairing System.

10.2 Producer-Consumer Relationship

The data flows between different layers and subsystems are defined in Table 10.1. The producer-consumer relationship is described in Table 10.2. The producer is the source of data element; the consumer is the destination of the data element.
<table>
<thead>
<tr>
<th>Data Elements</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1</td>
<td>Digital Image Format from the Webcam</td>
</tr>
<tr>
<td>I2</td>
<td>Text String from the Keyboard and Action Events from the Mouse</td>
</tr>
<tr>
<td>I3</td>
<td>Text String from the Card Scanner</td>
</tr>
<tr>
<td>I4</td>
<td>Text String from the Keypad</td>
</tr>
<tr>
<td>I5</td>
<td>Input Objects that contain staff members or parents information</td>
</tr>
<tr>
<td>A1</td>
<td>Output Objects that contain status messages and system data</td>
</tr>
<tr>
<td>A2</td>
<td>Data Objects that contain staff members, parents and devices information</td>
</tr>
<tr>
<td>A3</td>
<td>Data Objects that contain staff members, parents and devices information</td>
</tr>
<tr>
<td>A4</td>
<td>Requested Data that used for storing, matching, updating or editing</td>
</tr>
<tr>
<td>A5</td>
<td>Data Objects that contain devices and pairing information</td>
</tr>
<tr>
<td>A6</td>
<td>Unit ID</td>
</tr>
<tr>
<td>D1</td>
<td>Responses data to the request</td>
</tr>
<tr>
<td>D2</td>
<td>Query to the Database Subsystem</td>
</tr>
<tr>
<td>D3</td>
<td>Result query from the Database Subsystem</td>
</tr>
<tr>
<td>C1</td>
<td>Digital Data Packets generated from the USB Driver Subsystem</td>
</tr>
<tr>
<td>C2</td>
<td>Analog Data Packets that will be encoded for transmission</td>
</tr>
<tr>
<td>C3</td>
<td>Analog Data Packets that will be sent for Child Unit Layer</td>
</tr>
<tr>
<td>U1</td>
<td>Analog Data Packets that received by Child Unit Layer</td>
</tr>
<tr>
<td>U2</td>
<td>Digital Data Packets used to display the notification on Child Unit</td>
</tr>
</tbody>
</table>

Table 10-1: Data Elements
<table>
<thead>
<tr>
<th>PRODUCER</th>
<th>Webcam</th>
<th>Keyboard/Mouse</th>
<th>Card Scanner</th>
<th>Keypad</th>
<th>Display</th>
<th>Input</th>
<th>Output</th>
<th>Data Processing</th>
<th>Data Management Unit</th>
<th>Authentication</th>
<th>Database</th>
<th>USB Driver</th>
<th>Encoding</th>
<th>Transmission</th>
<th>Radio</th>
<th>Processing</th>
<th>Notification</th>
<th>Child Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Webcam</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>I1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Keyboard/Mouse</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>I2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Card Scanner</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>I3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Keypad</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>I4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Display</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>I5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Processing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A1, A2, A5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authentication</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Database</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USB Driver</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Encoding</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transmission</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Radio</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Processing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notification</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Child Unit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 10-2: Producer-Consumer Relationship Matrix
## 11 Requirements Mapping

Team AVIAR uses the Requirements Mapping to verify that the architectural design satisfies all high level requirements that defined in the System Requirements Specifications.

<table>
<thead>
<tr>
<th>#</th>
<th>Requirement</th>
<th>I/O</th>
<th>Application</th>
<th>Data Storage</th>
<th>Communication</th>
<th>Child Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Child-Care Facility</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>3.2</td>
<td>Easy Pairing</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>3.3</td>
<td>Secure Pairing</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.4</td>
<td>Registration</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.5</td>
<td>Identification Pairing Registration</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.7</td>
<td>Wireless Wearable Unit</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>5.1</td>
<td>Open Space Operation</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.2</td>
<td>Walled Space Operation</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.3</td>
<td>Test Connection</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.4</td>
<td>Type of Power Source</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>5.5</td>
<td>Length of Operation Time</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>5.6</td>
<td>Durability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
### Table 11-1: Requirements Mapping

<table>
<thead>
<tr>
<th></th>
<th>Operation Frequency</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>8.1</td>
<td>Operating System</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
12 Testing Considerations

12.1 Overview
The system architecture will be examined and tested by Team AVIAR to ensure that the Child-Parent Pairing System meets the expectations set forth in the System Requirements Specification and Architectural Design Specification. Each layer and subsystem in our architecture has been designed to be modularly independent, which results in a relatively trivial facilitation of our testing approach. The layers and subsystems can – and will – be tested individually before integration or implementation.

12.2 Testing Approach
Team AVIAR will thoroughly test each subsystem to ensure that it fulfills its responsibilities and purposes before implementation carries through to the layer-level.

Team AVIAR will also thoroughly test each layer independently to ensure that it accomplishes its responsibilities and purposes before integration with other layers.

Unit tests, component tests, integration tests, and systems tests will each be conducted and examined to verify functionality against the system’s requirements. The Child-Parent Pairing System will be tested to performance specifications for quality assurance purposes, and to meet the expectations of the team and the product’s stakeholders.

Team AVIAR will validate that all inputs and outputs are accurate, and that data flows according to expectations within the Child-Parent Pairing System. This includes data flows amongst individual layers, and lower-level subsystems and components as well.

12.3 I/O Layer
The I/O Layer must receive data from users of the Child-Parent Pairing System, and transmit data to the Application Layer. User input will come from the magnetic card reader and the 10-key keypad.

12.3.1 Input Subsystem
Verify that the physical aspect of reading data from various hardware inputs – such as the magnetic card reader, the 10-key keypad, and other necessary hardware devices – results in accurate data being sent along to the Application Layer. This functionality should be provided out of the box because we aren’t developing hardware devices for user input, so they should be uPnP solutions.
12.3.2 Output Subsystem
Verify that data received from the Application Layer is correctly displayed to the user via the workstation’s monitor when appropriate. This functionality will rely on the standard video output system of the computer.

12.4 Application Layer
The Application Layer must handle the software aspect, including processing, of each system procedure. These system procedures include receiving user input from the I/O Layer, queries to the Data Storage Layer and their subsequent results, signaling to the Communication Layer, and sending output to the user via the I/O Layer’s Output Subsystem.

12.4.1 Data Processing Subsystem
Verify that data sent from the I/O Layer is processed correctly – more specifically, that decisions are made correctly. If the input is for a check-in operation, a data storage request including authentication details should be sent to the Data Management Subsystem. If the input is for a staff login, an authentication check should be sent to the Data Management Subsystem. If the input is for a child pick-up, an authentication check should be made and then a query request should be sent to the Data Management Subsystem.

If the Data received comes from the Data Management Subsystem, the decision made here should either be to test a unit, notify a unit, associate a particular unit with a child, or confirm authentication. These cases must all be tested to verify accuracy with the Data Management Subsystem’s decision-making capabilities. Stepping through the code in debug mode will be very beneficial in this testing.

12.4.2 Data Management Subsystem
Verify that all data requests sent by the Data Processing Subsystem are packaged into appropriate database access objects. Data access requests will be submitted with their authentication details, but authentication will not take place here. Data storage requests will be submitted with identity information where relevant, ensuring that secure pairing remains intact as the data requests are submitted to the Data Storage Layer.

Manual test hooks will be written here to view the queries that are generated from this subsystem. They will be provided via a log output system.

12.4.3 Unit Subsystem
Verify that the correct child device is signaled from the Data Processing Subsystem before being submitted to the Communication Layer. A log output will aid here to verify that the signal transmitted to the Communication Layer is as expected.
12.5 Data Storage Layer

The Data Storage Layer must securely store information about the children and their parents, the staff, and the hardware bracelets. The Application Layer will submit queries for storage into, and retrieval from the database.

12.5.1 Authentication Subsystem

Verify that parents’ and children’s sensitive personal data is accurately and securely stored. This includes only returning sensitive data results after the request has been successfully authenticated. It also includes monitoring our salting and hashing algorithms to ensure that sensitive data like PIN numbers are salted and hashed before storage will be granted. Verification will be to simply ensure that sensitive data is never actually inserted into the database, but rather that hash values should be submitted.

Since the request objects will already be in standardized transactional database query format when received from the Data Management Subsystem in the Application Layer, a test harness will be generated to manually view – in plain text – the information submitted to the Database Subsystem.

12.5.2 Database Subsystem

Verify that the query and result calls to and from the Authentication Subsystem result in the appropriate responses from the database. Verification of proper error handling will be necessary here as well, in the case that queries submitted result in failure at the database level. Multiple ancillary hooks will be necessary here, and these will be exclusively for testing purposes.

12.6 Communication Layer

The Communication Layer must receive information from the Application Layer and convert it to a suitable, readable, and useful format for the Child Unit Layer. The primary responsibility of this layer is to handle the actual transmission of these packets to the child device via radio transmission.

12.6.1 USB Driver Subsystem

Verify that the Application Layer can successfully send messages to the Communication Layer, and that these messages can be interpreted so that the radio transmitter can be controlled and utilized.

12.6.2 Encoding Subsystem

Verify that digital data packets received from the Application Layer are converted to a format suitable for wireless radio transmission. Manual viewing of the packets will be necessary in order to pass this validation. This will require an ancillary hook to be used exclusively for testing purposes.
12.6.3 Transmission Subsystem
Verify that encoded data packets are successfully broadcasted through the physical medium to the Child Unit Layer. Testing will require at least one hardware device to validate the test’s acceptance rate.

12.7 Child Unit Layer
The Child Unit Layer must receive a radio signal from the Communication Layer in order to provide a notification to the child, so he/she knows that they are being called by their parent. This layer is also responsible for the logic that will illuminate the embedded LEDs at the appropriate time.

12.7.1 Radio Subsystem
Verify that radio signals are received from the Communication Layer, and then passed to the Processing Subsystem. Low-level viewing of the signals on the board or in the chip will be required for validation.

12.7.2 Processing Subsystem
Verify that radio signals received from the Radio Subsystem are processed correctly in order to see if the Notification Subsystem needs to be triggered or not. Many ‘false’ cases will be required here to validate that only the correct signals are passed on to the Notification Subsystem. An ancillary view of the Processing Subsystem’s activity and real-time state will be utilized exclusively for testing purposes here.

12.7.3 Notification Subsystem
Verify that the LEDs light up when triggered by the Processing Subsystem. This will be a simple visual check for verification.