# Table of Contents

1  Introduction ................................................................................................................. 8  
   1.1  Document Overview .............................................................................................. 8  
   1.2  Product Overview .................................................................................................. 8  
   1.3  Project Scope ......................................................................................................... 8  

2  Meta Architecture ...................................................................................................... 10  
   2.1  Architectural Vision .............................................................................................. 10  
   2.2  Guiding Principles ................................................................................................ 10  
   2.3  Tradeoffs ................................................................................................................ 11  
   2.4  Architecture Overview ......................................................................................... 11  

3  Input Layer .................................................................................................................. 14  
   3.1  Description ............................................................................................................. 14  
   3.2  Purpose .................................................................................................................... 14  
   3.3  Function .................................................................................................................. 14  
   3.4  Dependencies .......................................................................................................... 14  
   3.5  Processing ............................................................................................................... 14  
   3.6  Data ........................................................................................................................ 15  
   3.7  Subsystems ............................................................................................................ 15  

4  Data Controller Layer ................................................................................................. 18  
   4.1  Description ............................................................................................................. 18  
   4.2  Purpose .................................................................................................................... 18  
   4.3  Function .................................................................................................................. 18  
   4.4  Dependencies .......................................................................................................... 18  
   4.5  Processing ............................................................................................................... 19  
   4.6  Data ........................................................................................................................ 19  
   4.7  Subsystems ............................................................................................................ 19
9.3 Other requirements.................................................................................................................. 34

10 Operating System Dependencies .......................................................................................... 35
  10.1 Input Layer ...................................................................................................................... 35
  10.2 Data Controller Layer ..................................................................................................... 35
  10.3 Database Layer .............................................................................................................. 35
  10.4 Match Event Processor Layer ....................................................................................... 35
  10.5 Output Layer .................................................................................................................. 35

11 Testing Considerations ......................................................................................................... 36
  11.1 Overview ...................................................................................................................... 36
  11.2 Input Layer .................................................................................................................. 36
  11.3 Data Controller Layer .................................................................................................. 36
  11.4 Database Layer ............................................................................................................ 36
  11.5 Match Event Processor Layer ..................................................................................... 36
  11.6 Output Layer ................................................................................................................ 36
<table>
<thead>
<tr>
<th>Revision Number</th>
<th>Revision Date</th>
<th>Description</th>
<th>Rationale</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>8/8/13</td>
<td>ADS Draft</td>
<td>Compilation of team ADS sections</td>
</tr>
<tr>
<td>1.0</td>
<td>8/11/13</td>
<td>ADS for Gate Review</td>
<td>Updated with corrections, figures and tables</td>
</tr>
<tr>
<td>2.0</td>
<td>9/6/13</td>
<td>ADS Baseline</td>
<td>Updated with corrections from Gate Review</td>
</tr>
</tbody>
</table>
List of Tables

Table 3-1: Admin Input Interface ............................................................ 15
Table 3-2: Administrator Input Public Interfaces .................................. 15
Table 3-3: Equipment Input Interface ..................................................... 16
Table 3-4: Equipment Input Public Interfaces ....................................... 16
Table 3-5: Input Packager Interface ....................................................... 17
Table 4-1: Input Controller Interface .................................................... 19
Table 4-2: Database Controller Interface .............................................. 20
Table 4-3: Match Event Controller Interface ......................................... 20
Table 4-4: Game Output Packager Interface .......................................... 21
Table 5-1: Database Manager Interface ................................................ 23
Table 6-1: Match Event Processor Interface ......................................... 25
Table 7-1: Output Handler Interface ...................................................... 28
Table 7-2: Admin Display Interface ....................................................... 29
Table 7-3: Audience Display Interface ................................................ 29
Table 8-1: Data Flow Definition ............................................................ 31
Table 8-2: Producer-Consumer Matrix ................................................ 32
Table 9-1: Customer Requirements Mapping ....................................... 33
Table 9-2: Performance Requirements Mapping .................................... 34
Table 9-3: Other Requirements Mapping .............................................. 34
List of Figures

Figure 1-1: Conceptual Figure ........................................................................................................ 9
Figure 2-1: Architecture Layout Diagram ....................................................................................... 12
Figure 3-1: Input Layer .................................................................................................................. 14
Figure 4-1: Data Controller Layer .................................................................................................. 18
Figure 5-1: Database Layer ........................................................................................................... 22
Figure 6-1: Match Event Processor Layer ....................................................................................... 24
Figure 7-1: Presentation Layer ....................................................................................................... 27
Figure 8-1: Layer Diagram .............................................................................................................. 30
1 Introduction

1.1 Document Overview
The Architectural Design Specification (ADS) is intended to provide a high-level structure of how the Taekwondo Match Management System (TMMS) is designed. The ADS will define the guiding principles used in order to create the architectural design of the TMMS including the individual layers of that design. This document will also describe the functions of each layer, the subsystems within each layer, and the data flows and interfaces between the layers and subsystems. The document will close by defining the operating system dependencies and testing considerations for the TMMS.

1.2 Product Overview
The Taekwondo Match Management System (TMMS) will be used to manage a match between two players competing in a Taekwondo competition and display parts of this information to an audience. The TMMS will allow administrators to input competitor information and store it in an internal database as well as display this information back to the administrator and an audience. The TMMS will be started from a computer and will communicate with Daedo equipment that two competitors wear for protection. The Daedo equipment is outside of our system, but captures the force of hits and relays that information to the TMMS. The TMMS then uses this information to apply force deductions to a life bar which is displayed to the administrators and an audience. The TMMS also uses the force information to test the equipment using a test mode built into the TMMS which displays the amount of force to the administrators. The system will also allow administrators to apply penalties, set score modifier deductions, start medical timeouts for the system, and pull up match event information for an active competition.

1.3 Project Scope
The TMMS is a software management system that will receive data from Daedo equipment worn by two competitors engaging in a Taekwondo match. The system is a pure software system and has no external parts that make up the system, but must receive data from Daedo equipment in order to execute all of its functions. The main functions of the system are keeping life bars of two competitors based off of force deductions and ensuring that the system can keep track of both competitors in a reliable manner.
Figure 1-1: Conceptual Figure
2 Meta Architecture

2.1 Architectural Vision

The TMMS will consist of 5 layers: the Input layer, Data Controller layer, the Database layer, Match Event Processor layer, and the output layer. These layers have been designed to encompass our guiding principles and rules. The main considerations in the design of our architecture are real time performance and ease of use in order to accurately demonstrate force deductions and use the system with minimal training. Therefore the flow of data between layers and subsystems has also been carefully considered to ensure that the TMMS can maintain these main principles.

2.2 Guiding Principles

2.2.1 Real-time performance

The system has been designed in order to receive, interpret, and execute information from the Daedo equipment in real time. Therefore, the system must provide updates as quickly as possible so an audience and administrator can view the result of an event without a delayed response.

2.2.2 Ease of use

The system has been designed in order to allow users to operate the system with little or no training so that the system can be used by many different people without having to have an expert on site.

2.2.3 Reliability

The reliability of data is also a top priority and has been a guiding principle so that the system can accurately represent a Taekwondo competition. The reliable force data ensures that both competitors are competing in a fair match.

2.2.4 Modularity

The system has also been designed to be modular so that there are no overlapping functionality between layers and therefore each layer and subsystem can be easily changed.

2.2.5 Maintainability

The system has been designed to be maintainable in order to ensure the system can be easily fixed, altered, or upgraded if a problem or need for alteration arises.
2.2.6 Expandability

The TMMS has also considered expandability in the design so that the system can be expanded to interact with a tournament management system. A tournament will receive all the data from matches that are managed by TMMS and use it to store and display tournament information and statistics. The TMMS will then be a subset of the larger tournament management system.

2.3 Tradeoffs

The initial architectural design had a large processing layer which did the functions of both the Data Controller layer, parts of the Database layer, and the match event processor. The separation of this layer into an extra layer, the Data Controller layer, was decided upon because of the need to make sure that the data, after being processed, was being routed to the correct location. This layer was also considered more modular by the team, since it separated the processing, converting instructions into queries and calculating force data, from the packaging and routing of data. The tradeoff for modularity came in that the processing will be a little slower based off the extra packaging and routing that must be done with the data controller layer. Although this does cause slower processing of data, the team has decided that the amount of slowed processing shouldn’t affect the TMMS too much and therefore we gain more in this extra level of modularity.

The initial design also had the database management subsystem as part of the processing layer. This was moved into the Database layer because of the need for a clearer interface between the Database layer and the Data Controller layer. The initial design did the creation of queries inside the Processing layer and therefore there was no need for a Database layer. The original database just stored information and had no other defined functions. The current Database layer receives administrator instructions and match statistics that are then converted to queries that either request information or store information. The tradeoffs for this subsystem are in processing speed because of the added Database controller and Database manager subsystems there is an added level of processing that will therefore slow down the speed of the TMMS. Although the speed of processing will be slowed down the team has decided that it should not affect the real time performance and therefore we gain more in modularity and maintainability.

2.4 Architecture Overview

The architecture contains five layers each layer being independent of any other layer in their functionality. Each layer must contain a clear interface and must exchange data in a consistent manner between layers. The five layers are the Input layer, the Data Controller layer, the Database layer, the Match Event Processor layer, and the output layer. The architectural layers are displayed below.
2.4.1 Input layer

The Input layer will receive input from an administrator’s screen and the Daedo equipment. The administrator’s screen will enter data such as competitor information, setup of the system, and altering the system. The Daedo equipment will send raw data that represents the force of impact on each competitor’s equipment and the TMMS will receive this data. This layer will then determine the type of data it received and package the data for transfer to the Data Controller layer. The packaged data is data that is relevant to the next layer.
2.4.2 Data Controller layer

The Data Controller layer will be responsible for routing packaged data received from the administrator’s screen, from the Daedo equipment, and any received data from the database to its correct destination. The Data Controller layer is also responsible for holding data that has been received from the Input layer or the database, and packaging information received from the Database layer or the Match Event Processor layer. The Data Controller layer will then make sure that all the relevant information is packaged together to send to the output layer.

2.4.3 Database layer:

The Database layer will be responsible for receiving input from the database handler. The database manager inside the Database layer will then decide if this information needs to be stored or information needs to be retrieved based off of the data that has been sent to it from the database handler. The database manager will then create these queries, execute the queries, receive the result of the queries, and then package and send the data back to the Input layer.

2.4.4 Match Event Processor:

The match event processor will be responsible for changing the state of a match. Any raw force data that has been received from the Daedo equipment will pass through this layer in order to be converted to force data that can be read by the game output packager in the Data Controller layer. This layer will also add any score modifiers to that force data in order to correctly represent the state of each competitors life bar.

2.4.5 Presentation layer:

The Presentation layer will be responsible for correctly displaying the data that has come from the Data Controller layer. This data will be force deductions to the life bar, administration input data such as information about a competitor, and different modes of the TMMS such as test mode. The Presentation layer will also be responsible for making sure that the appropriate information is sent to the administrator screen and audience screen.
3 Input Layer

3.1 Description
The Input layer provides an interface for the administrator of a match and a competitor wearing the Daedo equipment. An administrator will be able to input commands to the system through a graphical user interface (GUI). The Daedo equipment will send signals to the Input layer.

3.2 Purpose
The Input layer will handle mouse click events and keyboard inputs from an administrator. These will be sent out to another layer in a format that can be used by the system. The input from the Daedo equipment will also be formatted in a way that the other layers can consume or process.

3.3 Function
The Input layer will contain the necessary functions to consume and produce administrator input, and input from the Daedo equipment. Once the inputs have been retrieved, the Input layer will package them in a consumable format and pass the data on to the Data Controller layer.

3.4 Dependencies
The Input layer is dependent on the Daedo equipment and the administrator’s computer and peripherals.

3.5 Processing
The Input layer will provide some light processing on the inputs from the administrator and the Daedo equipment. The input received from the administrator will be processed to send a
command to the Data Controller layer. The Daedo equipment data will be processed and sent to the Data Controller layer.

3.6 Data

The Input layer will package the data received from the administrator and the Daedo equipment into a consumable format for the Data Controller layer.

3.7 Subsystems

3.7.1 Admin Input

3.7.1.1 Purpose

This subsystem is responsible for handling any input given by the administrator.

3.7.1.2 Function

This subsystem will retrieve any input from the keyboard or mouse from the administrator’s computer and send the information to be formatted.

3.7.1.3 Interlayer Interfaces

Table 3-1: Admin Input Interface

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Information Required</th>
<th>Information Returned</th>
</tr>
</thead>
<tbody>
<tr>
<td>getAdminInput</td>
<td>Receives input from the administrator, formats, and sends the data to the Input Packager</td>
<td>Keyboard or mouse input</td>
<td>Formatted admin input</td>
</tr>
</tbody>
</table>

3.7.1.4 Public Interfaces

Table 3-2: Administrator Input Public Interfaces

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keyboard</td>
<td>Input Layer</td>
<td>Keyboard entries from administrator</td>
</tr>
<tr>
<td>Mouse</td>
<td>Input Layer</td>
<td>Mouse events from administrator</td>
</tr>
</tbody>
</table>
3.7.2 Equipment Input

3.7.2.1 Purpose

This subsystem is responsible for handling any raw input given by the Daedo defense equipment.

3.7.2.2 Function

This subsystem will retrieve any input from the Daedo defense equipment and send the information to be formatted.

3.7.2.3 Interlayer Interfaces

Table 3-3: Equipment Input Interface

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Information Required</th>
<th>Information Returned</th>
</tr>
</thead>
<tbody>
<tr>
<td>getEquipmentInput</td>
<td>Receives input from the equipment, formats the input, and sends the data to the Input Packager</td>
<td>Equipment input</td>
<td>Formatted equipment input</td>
</tr>
</tbody>
</table>

3.7.2.4 Public Interfaces

Table 3-4: Equipment Input Public Interfaces

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chest Equipment</td>
<td>Input Layer</td>
<td>Force data string</td>
</tr>
<tr>
<td>Head Equipment</td>
<td>Input Layer</td>
<td>Force data string</td>
</tr>
</tbody>
</table>

3.7.3 Input Packager

3.7.3.1 Purpose

This subsystem will retrieve data from the Admin Input and Equipment Input subsystems. It will format the data for consumption by the Data Controller layer.

3.7.3.2 Function

This subsystem will take the raw input from the Equipment Input subsystem and the Admin Input subsystem, and package it into a consumable data type.
3.7.3.3 Interlayer Interfaces

Table 3-5: Input Packager Interface

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Information Required</th>
<th>Information Returned</th>
</tr>
</thead>
<tbody>
<tr>
<td>setInputType</td>
<td>Receives data and determines type (admin or equipment)</td>
<td>Admin or equipment input</td>
<td>Formatted input data</td>
</tr>
</tbody>
</table>

3.7.3.4 Public Interfaces

None.
4 Data Controller Layer

![Diagram of Data Controller Layer](image)

Figure 4-1: Data Controller Layer

4.1 Description
The Data Controller layer will control where the data received from the Input layer will go.

4.2 Purpose
The purpose of this layer is to take the information received from the Input layer and send it to an appropriate layer to be consumed. The Data Controller layer will receive data from the Input, Database, and Match Processing layers. It will also send data to the Database, Match Processing, and Presentation layers.

4.3 Function
This layer will route all information to its destination. It is not meant to process, store, or display any information.

4.4 Dependencies
The Data Controller layer is dependent on the surrounding layers for data.
4.5 Processing

The Data Controller layer will have minimal processing. It will read the type of data that it has received from the Input layer and determine which layer it needs to send the data to.

4.6 Data

The data received from the Input layer will be requests for database access, match event data, and output data.

4.7 Subsystems

4.7.1 Input Controller

4.7.1.1 Purpose

This subsystem will retrieve input from the Input layer and direct it to the proper subsystem for consumption.

4.7.1.2 Function

The function of this subsystem is to recognize the type of data from the Input layer and route the data to the correct subsystem.

4.7.1.3 Interlayer Interfaces

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Information Required</th>
<th>Information Returned</th>
</tr>
</thead>
<tbody>
<tr>
<td>determineType</td>
<td>Receives data, determines data type, routes to proper subsystem</td>
<td>Data from Input Layer</td>
<td>Redirected data for input layer</td>
</tr>
</tbody>
</table>

4.7.1.4 Public Interfaces

None.

4.7.2 Database Controller

4.7.2.1 Purpose

This subsystem will retrieve data given to it by the Input Controller and send it to the Database layer. It will also retrieve information from the Database layer.
4.7.2.2  Function

The function of this subsystem is to communicate with the Database layer to send and receive data. This subsystem will also send data to the Match Event Controller for further processing.

4.7.2.3  Interlayer Interfaces

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Information Required</th>
<th>Information Returned</th>
</tr>
</thead>
<tbody>
<tr>
<td>getStorageData</td>
<td>Receives data, sets status of data, sends to Database Layer</td>
<td>Data from Input Controller or Match Event Controller</td>
<td>None</td>
</tr>
</tbody>
</table>

4.7.2.4  Public Interfaces

None.

4.7.3  Match Event Controller

4.7.3.1  Purpose

This subsystem will retrieve data and send it to the Match Event Processor layer.

4.7.3.2  Function

The function of this subsystem is to send and retrieve data from the Input Controller subsystem and the Match Event Processor layer. It also receives information from the Database Controller subsystem.

4.7.3.3  Interlayer Interfaces

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Information Required</th>
<th>Information Returned</th>
</tr>
</thead>
<tbody>
<tr>
<td>getMatchEventData</td>
<td>Receives data from Match Event Controller</td>
<td>Processed match event data</td>
<td>Match event data</td>
</tr>
</tbody>
</table>

4.7.3.4  Public Interfaces

None.
4.7.4 Game Output Packager

4.7.4.1 Purpose

This subsystem will receive data and format it for consumption by the Presentation layer.

4.7.4.2 Function

The function of this subsystem is to handle any data that needs to be output from the connecting layers.

4.7.4.3 Interlayer Interfaces

Table 4-4: Game Output Packager Interface

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Information Required</th>
<th>Information Returned</th>
</tr>
</thead>
<tbody>
<tr>
<td>setOutputType</td>
<td>Receives data from subsystems and determines output type</td>
<td>Controller data</td>
<td>Output data and type</td>
</tr>
</tbody>
</table>

4.7.4.4 Public Interfaces

None.
5 Database Layer

![Database Layer Diagram](image)

Figure 5-1: Database Layer

5.1 Description
The Database layer will provide access to a database which will store match, competitor, and country code information. It will also store match statistics, which includes information generated from a match.

5.2 Purpose
The purpose of this layer is to store information that will not be needed for quick access.

5.3 Function
The Database layer will receive all data from the Data Controller layer. It will store the information in the appropriate format and send the data back to the Data Controller layer when a request is received.

5.4 Dependencies
The Database layer is dependent on the Data Controller layer.
5.5 Processing

The Database layer will process requests from the Data Controller layer and create the necessary query for the requests.

5.6 Data

The data that is received will inform the Database layer what kind of information is needed, or what kind of information needs to be stored, along with the relevant data.

5.7 Subsystems

5.7.1 Database Manager

5.7.1.1 Purpose

This subsystem will provide access to the Database layer.

5.7.1.2 Function

The function of this subsystem is to receive requests from the Data Controller layer and create queries based on the data received.

5.7.1.3 Interlayer Interfaces

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Information Required</th>
<th>Information Returned</th>
</tr>
</thead>
<tbody>
<tr>
<td>queryDatabase</td>
<td>Receives data from Data Controller Layer and queries the database</td>
<td>Data from the Data Controller Layer</td>
<td>Query results</td>
</tr>
</tbody>
</table>

5.7.1.4 Public Interfaces

None.

5.7.2 Data Storage

5.7.2.1 Purpose

This subsystem will store all relevant system data.

5.7.2.2 Function

The function of this subsystem is to store data.
6 Match Event Processor Layer

Figure 6-1: Match Event Processor Layer

### 6.1 Description
The Match Event Processor layer will handle all of the computations for any match event.

### 6.2 Purpose
The purpose of this layer is to relieve any other layer from heavy processing that may occur from the data received from the Data Controller layer.

### 6.3 Function
This layer will process all match events, such as the force of a strike on the Daedo equipment, the life bar deduction data, and any score changing events.

### 6.4 Dependencies
The Match Event Processor layer is dependent on the Data Controller layer.
6.5 Processing
The Match Event Processor layer will process all of the match events that need to be calculated, including: force of hits, penalty deductions on the life bar, and any score modifiers.

6.6 Data
The data received from the Data Controller layer will be the type and value of a match event.

6.7 Subsystems

6.7.1 Match Event Processor

6.7.1.1 Purpose
The purpose of this subsystem is to send and receive data that is relevant to match events.

6.7.1.2 Function
The function of this subsystem is to receive data from the Data Controller layer and send the data to be processed, and then returned to the Data Controller layer.

6.7.1.3 Interlayer Interfaces

Table 6-1: Match Event Processor Interface

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Information Required</th>
<th>Information Returned</th>
</tr>
</thead>
<tbody>
<tr>
<td>calculateEvent</td>
<td>Receives data from the Data Controller Layer and calculates the match event data: force, penalty, score modification</td>
<td>Match event data</td>
<td>Processed match data</td>
</tr>
</tbody>
</table>

6.7.1.4 Public Interfaces
None.

6.7.2 Match Calculations

6.7.2.1 Purpose
The purpose of this subsystem is to perform calculations on match data.
6.7.2.2 Function

The function of this subsystem is to calculate match data based on force, penalty, or score modification.

6.7.3 Calibration

6.7.3.1 Purpose

The purpose of this subsystem is to perform calibrations on the Defense Equipment.

6.7.3.2 Function

The function of this subsystem is to allow the Defense Equipment to be calibrated to ensure that the competitors’ equipment is registering the proper force data.
7 Presentation Layer

7.1 Description
The Presentation layer will be responsible for displaying the GUI for the match administrator. It will also display the audience screen.

7.2 Purpose
The purpose of this layer is to display the GUI for the match administrator and to display the audience screen.

7.3 Function
This layer will receive information from the Data Controller layer, format the data, and output the properly formatted data to the administrator and audience screens.

7.4 Dependencies
The Presentation layer is dependent on the physical displays of the administrator and audience screen. It is also dependent on the Data Controller layer to receive the proper data to be output.

7.5 Processing
This layer will have no processing.
7.6 Data

The Presentation layer will receive data that will be used to update the administrator screen and
the audience screen.

7.7 Subsystems

7.7.1 Output Handler

7.7.1.1 Purpose

The output handler will determine which display needs to be updated.

7.7.1.2 Functions

This subsystem will receive data from the Data Controller layer and determine which display needs to be updated.

7.7.1.3 Interlayer Interfaces

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Information Required</th>
<th>Information Returned</th>
</tr>
</thead>
<tbody>
<tr>
<td>getOutput</td>
<td>Receives output data and determines output type and destination</td>
<td>Output data</td>
<td>Destination for display data</td>
</tr>
</tbody>
</table>

7.7.1.4 Public Interfaces

None.

7.7.2 Admin Display

7.7.2.1 Purpose

The admin display will display all the necessary GUI components and information for a match administrator.

7.7.2.2 Function

The function of this subsystem is to display the GUI and other match information.
7.7.2.3 Interlayer Interfaces

Table 7-2: Admin Display Interface

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Information Required</th>
<th>Information Returned</th>
</tr>
</thead>
<tbody>
<tr>
<td>displayAdmin</td>
<td>Displays admin data</td>
<td>Admin output data</td>
<td>Visual display on admin screen</td>
</tr>
</tbody>
</table>

7.7.2.4 Public Interfaces

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Presentation Layer</td>
<td>Admin display monitor</td>
<td>Admin interface</td>
</tr>
</tbody>
</table>

7.7.3 Audience Display

7.7.3.1 Purpose

The audience display will display the required match information.

7.7.3.2 Function

The function of this subsystem is to display any relevant match information for the audience.

7.7.3.3 Interlayer Interfaces

Table 7-3: Audience Display Interface

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Information Required</th>
<th>Information Returned</th>
</tr>
</thead>
<tbody>
<tr>
<td>displayAudience</td>
<td>Displays audience data</td>
<td>Audience output data</td>
<td>Visual display on audience screen</td>
</tr>
</tbody>
</table>

7.7.3.4 Public Interfaces

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Presentation Layer</td>
<td>Audience display monitor</td>
<td>Display match information to audience</td>
</tr>
</tbody>
</table>
8 Relationship Mapping

8.1 Overview
This section defines the relationships between the layers, data flows and subsystems.

8.2 Data Flow Definition
<table>
<thead>
<tr>
<th>Data Flow</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1</td>
<td>Keyboard and mouse events from Admin PC</td>
</tr>
<tr>
<td>I2</td>
<td>Raw input from Defense Equipment</td>
</tr>
<tr>
<td>I3</td>
<td>Formatted Admin input commands</td>
</tr>
<tr>
<td>I4</td>
<td>Formatted Equipment input signals</td>
</tr>
<tr>
<td>I5</td>
<td>Input data from Admin/Equipment that has been labeled for use in the Data Controller Layer</td>
</tr>
<tr>
<td>DC1</td>
<td>Static admin user input to be stored</td>
</tr>
<tr>
<td>DC2</td>
<td>Match event data (force information, competitor designation, equipment identifier, equipment location identifier, match alterations)</td>
</tr>
<tr>
<td>DC3</td>
<td>Admin output data (user interface alterations)</td>
</tr>
<tr>
<td>DC4</td>
<td>Match statistic data (country flags, completed match logs)</td>
</tr>
<tr>
<td>DC5</td>
<td>Formatted data from interlayer subsystems for the Database layer (match statistics, competitor information)</td>
</tr>
<tr>
<td>DC6</td>
<td>Processed match event data that needs to be stored (match log)</td>
</tr>
<tr>
<td>DC7</td>
<td>Match events to be output (life bar alterations, time alterations)</td>
</tr>
<tr>
<td>DC8</td>
<td>Formatted output data from interlayer subsystems (data that will be labeled for use by the Admin and/or Audience displays)</td>
</tr>
<tr>
<td>DC9</td>
<td>Match events to be processed</td>
</tr>
<tr>
<td>DB1</td>
<td>Queries to the database</td>
</tr>
<tr>
<td>DB2</td>
<td>Result of queries to the database</td>
</tr>
<tr>
<td>DB3</td>
<td>Data retrieved from database</td>
</tr>
<tr>
<td>M1</td>
<td>Match event data for calculations (force information, competitor designation, equipment identifier, equipment location identifier, match alterations)</td>
</tr>
<tr>
<td>M2</td>
<td>Calculated data for a given match event (force information, competitor designation, equipment identifier, equipment location identifier, match alterations)</td>
</tr>
<tr>
<td>M3</td>
<td>Return processed match event data (force information, competitor designation, equipment identifier, equipment location identifier, match alterations)</td>
</tr>
<tr>
<td>M4</td>
<td>Force data required for calibration</td>
</tr>
<tr>
<td>M5</td>
<td>Adjusted multiplier for force data</td>
</tr>
<tr>
<td>P1</td>
<td>Admin display data (Admin interface alterations)</td>
</tr>
<tr>
<td>P2</td>
<td>Audience display data (Life bar status, time status, round status)</td>
</tr>
<tr>
<td>P3</td>
<td>Output to GUI (Admin interface alterations)</td>
</tr>
<tr>
<td>P4</td>
<td>Output to audience screen (Life bar status, time status, round status)</td>
</tr>
</tbody>
</table>
### 8.3 Producer-Consumer Relationship Matrix

**Table 8-2: Producer-Consumer Matrix**

<table>
<thead>
<tr>
<th>PRODUCER</th>
<th>Admin Input</th>
<th>Equipment Input</th>
<th>Input Packager</th>
<th>Input Controller</th>
<th>Database Controller</th>
<th>Match Event Controller</th>
<th>Game Output Packager</th>
<th>Data Storage</th>
<th>Database Manager</th>
<th>Match Event Processor</th>
<th>Match Calculations</th>
<th>Calibration</th>
<th>Output Handler</th>
<th>Admin Display</th>
<th>Audience Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>Admin Input</td>
<td>X</td>
<td>I 3</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>Equipment Input</td>
<td>X I 4</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>Input Packager</td>
<td>X I 5</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>Input Controller</td>
<td>X D C1</td>
<td>D C2</td>
<td>D C3</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>Database Controller</td>
<td>X D C4</td>
<td>D C5</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>Match Event Controller</td>
<td>D C6</td>
<td>X D C7</td>
<td>D C9</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>Game Output Packager</td>
<td>X</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 Storage</td>
<td></td>
<td>X D B2</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>Database Manager</td>
<td></td>
<td>D B3</td>
<td>D B1</td>
<td>X</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>Match Event Processor</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>Match Calculations</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>Calibration</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>Output Handler</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>Admin Display</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>X</td>
<td></td>
</tr>
<tr>
<td>Audience Display</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>X</td>
<td></td>
</tr>
</tbody>
</table>
9 Requirements Mapping

The purpose of this requirements mapping is to ensure that all the requirements are met. These are high level requirements as per specified in the SRS document. They are numbered according to their sections. The mapping ensures that the layers that we have come up with are able to accommodate the requirements.

9.1 Customer requirements

Table 9-1: Customer Requirements Mapping

<table>
<thead>
<tr>
<th>Number</th>
<th>Requirement</th>
<th>Input</th>
<th>Data Controller</th>
<th>Database</th>
<th>Match Event Processor</th>
<th>Presentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Audience Display</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>3.2</td>
<td>Start screen</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>3.3</td>
<td>Control panel</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>3.4</td>
<td>Match Setup</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>3.5</td>
<td>Test Mode</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>3.6</td>
<td>Equipment hit detection</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.7</td>
<td>Health calculations</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.8</td>
<td>Penalties</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.9</td>
<td>Score modifiers</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.10</td>
<td>Equipment parts on / off</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.11</td>
<td>Match log</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>3.12</td>
<td>Administrator display</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
### 9.2 Performance requirements

**Table 9-2: Performance Requirements Mapping**

<table>
<thead>
<tr>
<th>Number</th>
<th>Requirement</th>
<th>Input</th>
<th>Data Controller</th>
<th>Database</th>
<th>Match Event Processor</th>
<th>Presentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Start Program</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>5.2</td>
<td>Life bar deduction response</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>5.3</td>
<td>Recording log</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>5.4</td>
<td>Response to button clicks</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>5.5</td>
<td>Processing force data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

### 9.3 Other requirements

**Table 9-3: Other Requirements Mapping**

<table>
<thead>
<tr>
<th>Number</th>
<th>Requirement</th>
<th>Input</th>
<th>Data Controller</th>
<th>Database</th>
<th>Match Event Processor</th>
<th>Presentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1</td>
<td>Store contestant information</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.2</td>
<td>Store match events</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.3</td>
<td>Store country information</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.4</td>
<td>Store match statistics</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.5</td>
<td>Calibration</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
10 Operating System Dependencies

10.1 Input Layer
The Input Layer will be dependent on C# System.IO.Ports library to interface with the USB receiver from the equipment. The Input Layer will be required to be user friendly depend on C# Windows Form Designer.

10.2 Data Controller Layer
The Data Controller Layer will be required to interact with every other layer in the system depend on C#.

10.3 Database Layer
The Database Layer will be required to interact with the database by sending SQL queries depend on C# MySql.Data.MySqlClient library and MySQL Connectors.

10.4 Match Event Processor Layer
The Match Event Processor Layer will be dependent on C# Math libraries to do force calculation and calibration.

10.5 Output Layer
The Output Layer will be dependent on Windows Form Designer for C# to display administrator screen and audience screen.
11 Testing Considerations

11.1 Overview
The system architecture will be tested to ensure it fulfills the requirements according to the System Requirements Specifications documents and the guiding principles defined in the architecture design. Team T.K Force will ensure the system has the real-time performance, ease-of-use, reliability, modularity, maintainability and the expandability. Each layer will be tested on its own to ensure its functionality before being tested in combination with the other layers.

11.2 Input Layer
The Input Layer shall be tested to ensure reliability, expandability and real-time performance. Different input from the administrator and equipment will be tested based on the correctness and react time. While taking the input from administrator, the system shall have a clear interface and guide for the user to enter the proper information. Equipment input shall be tested with different events happened during a regular match.

11.3 Data Controller Layer
The Data Controller Layer’s responsibility is to manage the configuration of the system; this layer shall be tested to ensure reliability and modularity. Different type of inputs and events will be tested over this layer. This layer shall be able to properly interact with other layers and retrieve and send data to other layers.

11.4 Database Layer
The Database Layer shall be tested to ensure reliability. The integrity of the system relies on data being sent to the system database and saved on to the local disk.

11.5 Match Event Processor Layer
The Match Event Processor shall be tested to ensure reliability, real-time performance, maintainability and expandability; these are the highest priorities for this system. The match calculations shall be tested to ensure reliability, it shall always return the correct numbers after the calculation done by certain formulas.

11.6 Output Layer
The Output Layer shall be tested to ensure reliability, ease-of-use, real-time performance and maintainability.