

McMASTER UNIVERSITY

SMARTSERVE

SOFTWARE & MECHATRONICS CAPSTONE

---

# High Level System Design

---

*Authors:*

Christopher McDonald  
Harit Patel  
Janak Patel  
Jared Rayner  
Nisarg Patel  
Sam Hamel  
Sharon Platkin

*Professor:*

Dr. Alan Wassyng

*Teaching Assistants:*

Bennett Mackenzie  
Nicholas Annable  
Stephen Wynn-Williams  
Viktor Smirnov



Last compiled on December 5, 2017

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Project Overview . . . . .	3
1.2	Document Overview . . . . .	3
1.3	Naming Conventions and Terminology . . . . .	4
1.4	Project Scope . . . . .	5
<b>2</b>	<b>System Description</b>	<b>5</b>
2.1	System Architecture . . . . .	5
2.2	Subsystems . . . . .	5
2.3	Use Cases . . . . .	6
2.3.1	Use Case Diagram . . . . .	6
2.4	Sequence Diagram . . . . .	6
2.5	Behaviour Description . . . . .	8
2.5.1	Normal Operation . . . . .	8
2.5.2	Error Handling . . . . .	8
<b>3</b>	<b>Subsystems Overview</b>	<b>8</b>
3.1	SmartServe . . . . .	8
3.2	Computer Vision . . . . .	9
3.3	Shot Recommendation . . . . .	9
3.4	Shooting Model . . . . .	9
3.5	Shot Optimizer . . . . .	9
3.6	Data Storage . . . . .	9
3.7	User Interface . . . . .	9
3.8	Shooting Mechanism . . . . .	9
<b>4</b>	<b>Class Responsibility Collaboration (CRC) Cards</b>	<b>10</b>
4.1	SmartServe CRC Cards . . . . .	10
4.2	Computer Vision CRC Cards . . . . .	10
4.3	Shot Recommendation CRC Cards . . . . .	10
4.4	Shooting Model CRC Cards . . . . .	10
4.5	Shot Optimizer CRC Cards . . . . .	10
4.6	Data Storage CRC Cards . . . . .	10
4.7	User Interface CRC Cards . . . . .	10
4.8	Shooting Mechanism CRC Cards . . . . .	10

## List of Figures

1	Revision History . . . . .	2
---	----------------------------	---

2	Top View of the Tennis Table . . . . .	5
3	Subsystem Breakdown . . . . .	6
4	Use Case Diagram . . . . .	7

Date	Revision	Comments	Author(s)
Dec 1, 2017	1.0	Main content done for all sections	Christopher McDonald

Figure 1: Revision History

# 1 Introduction

## 1.1 Project Overview

SmartServe is an autonomous table tennis training system for table tennis players with various skill levels. SmartServe aids in diagnosing and improving a player's performance over time. The system trains table tennis players by shooting table tennis balls towards the player and detects successful returns from the player. The system can further adapt to the player's weaknesses and help them overcome it through further training. Importantly, SmartServe alleviates the problems of finding and working with a coach for players, as well as coaches trying to train multiple players simultaneously. The system will be deemed a success if the table tennis players and coaches, can enjoy and see a value added by using SmartServe.

The project started at the beginning of the Fall 2017 academic term and will conclude at the end of the Winter 2018 term. In addition, the core project team consists of final year Software and Mechatronics Engineering students who are enrolled in the MECHTRON 4TB6/SFWRENG 4G06 capstone project course.

## 1.2 Document Overview

This document will cover the entire system from a high level point of view. This means the system will be decomposed into subsystems which each have their own responsibilities and design. The subsystems will have their intended input, expected output and description of how the module will be used. In further documentation, each will be designed in a way which is abstracted from this document's perspective. The expected use cases will also be detailed to understand the expectations of the user and how each subsystem interacts with another. A more detailed view of this including timing as a factor will be detailed in the Sequence Diagram section.

For each subsystem, its purpose will be defined with respect to the overall system. After doing so, detailed input and output parameters will be defined. The means of communication will also need to be described as these could be simple method calls to requests over some transport protocol like HTTP or TCP. The general architecture will be decided for each subsystem will be given as each needs to satisfy unique requirements and thus needs to be decided at a system level.

To finish the document, each subsystem will be responsible for implementing some set of requirements. These requirements are outlined in the requirements document found here. In the ideal scenario, one subsystem will be the only one that is responsible for any one requirement but may not be the case for all the requirements.

### 1.3 Naming Conventions and Terminology

The following terms and definitions will be used throughout this document:

- **System:** encompasses both the hardware and software parts of SmartServe
- **Shooting Mechanism:** refers to the part of the system that shoots the table tennis balls towards the user side (player)
- **Team:** all team members of the core capstone project, as noted in the list of Authors
- **User Side:** the side of the table where the user (player) is standing
- **System Side:** the side of the table where the electromechanical system is placed; it is the opposite side of the User Side
- **ACID:** a database transaction which is atomic, consistent, isolated and durable
- **FPS:** frames per second
- **GUI:** graphical user interface
- **Pitch:** rotation along the y-axis; this rotation angle primarily dictates the range of the ball from the net to the edge of the table on the user side
- **Yaw:** rotation along the z-axis; this rotation angle primarily dictates the panning functionality of the shooting mechanism from the right side to the left side of the table
- **Roll:** rotation along the x-axis
- **HTTP:** hypertext transfer protocol
- **TCP:** transmission control protocol

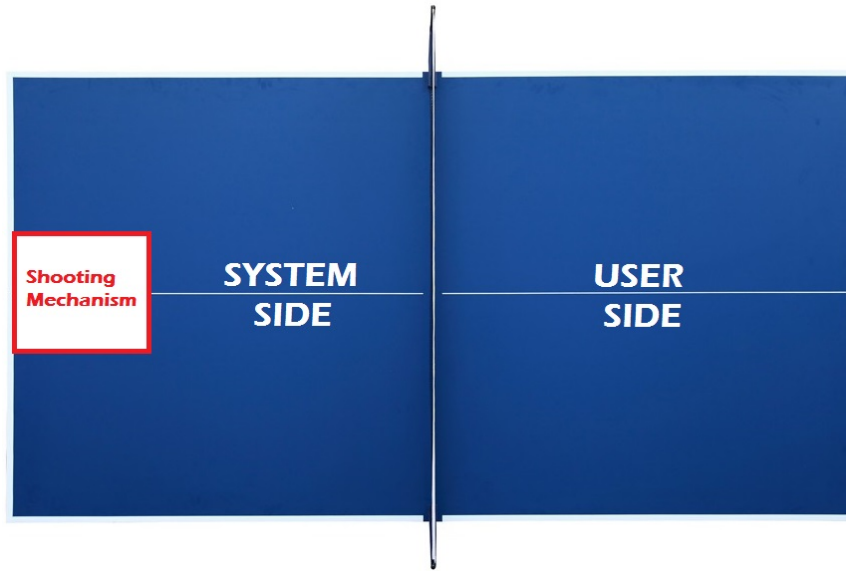


Figure 2: Top View of the Tennis Table

## 1.4 Project Scope

# 2 System Description

## 2.1 System Architecture

The system will follow a service-oriented / process-based architecture. This means that a central subsystem will interface with several services that serve a single purpose. Some subsystems will be simply told what needs to be done and others will be asked for some return value. This is done to implement separation of concerns where one subsystem doesn't know everything about the system and only what is necessary to satisfy their requirements. It also allows for easy increments of versioning, where one subsystem can be used as long as it is functional and easily swapped out for a newer version with extra features or increased performance. Moreover, this system has heavy timing constraints and an unpredictable environment so some actions must be taken in absence of a service's response. For example, the computer vision may take longer to track a ball depending on its trajectory where the system must shoot another ball in order to keep the user engaged.

## 2.2 Subsystems

The system will be broken down into the subsystems with the following purposes: general managing of subsystems, computer vision to detect returns, shot recommendation, model the shot's trajectory, optimize the distance travelled by the shooting mechanism, store the

data, take input and output from the user and shooting the ball toward the user. The diagram for this breakdown can be found in Figure 3.

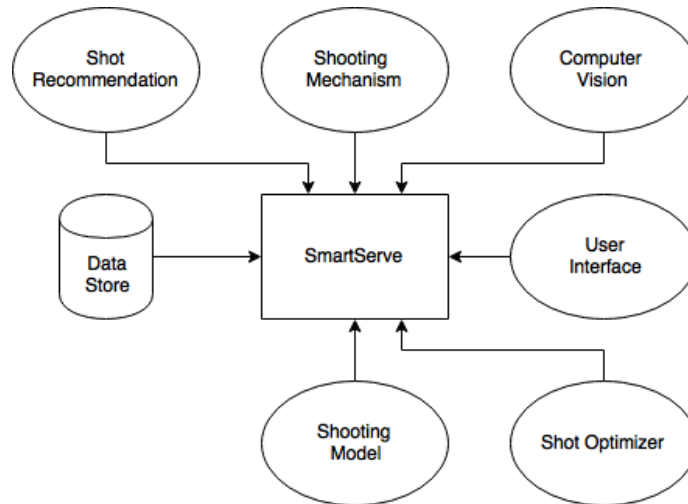


Figure 3: Subsystem Breakdown

## 2.3 Use Cases

The diagram including all use cases can be found in Figure 4. The user-instantiated ones will be described in detail below.

**Start Training**

**Stop Training**

**View Results**

### 2.3.1 Use Case Diagram

use boxes for all subsystems and list use cases

## 2.4 Sequence Diagram

basically the above with timing and loop

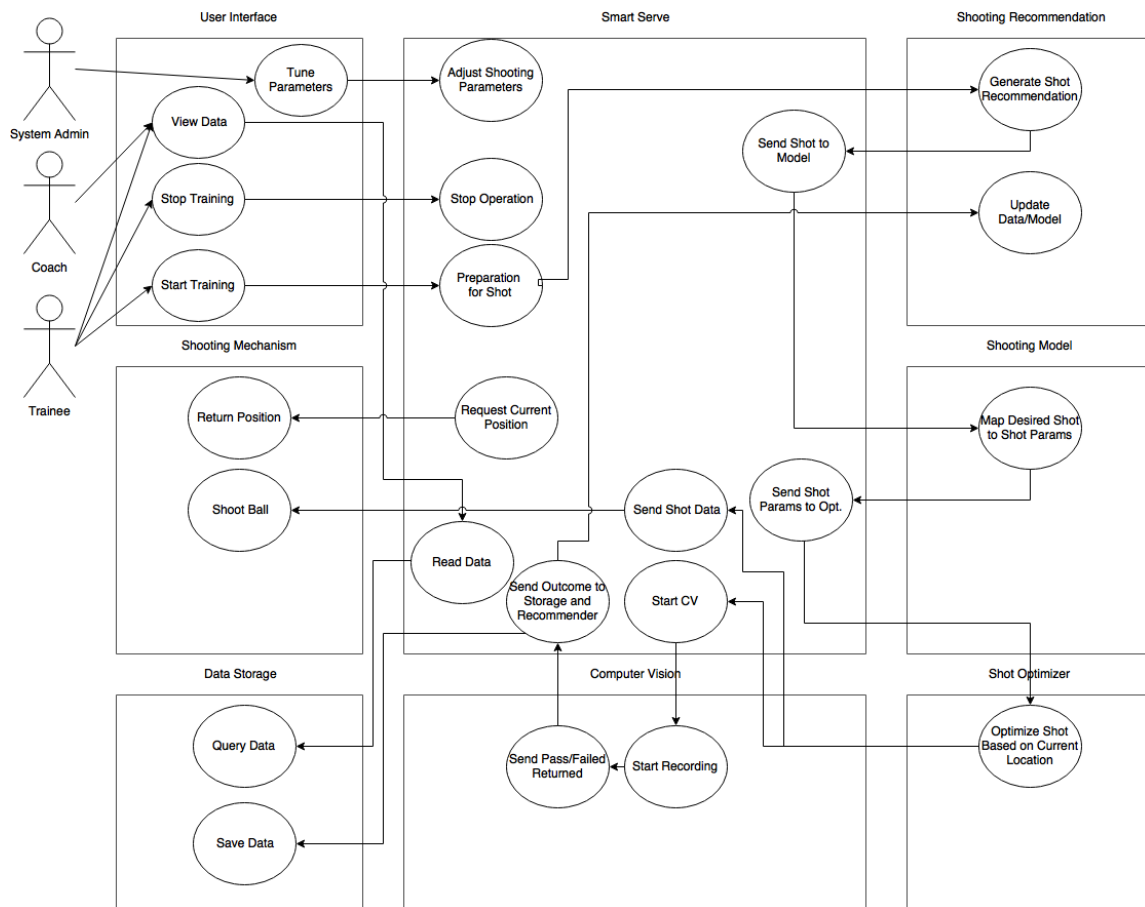


Figure 4: Use Case Diagram



## **2.5 Behaviour Description**

### **2.5.1 Normal Operation**

### **2.5.2 Error Handling**

## **3 Subsystems Overview**

### **3.1 SmartServe**

#### **Purpose**

This will serve the purpose of managing the state of the system and general input and output of all other subsystems.

Architecture

Interface

### 3.2 Computer Vision

Purpose

Architecture

Interface

### 3.3 Shot Recommendation

Purpose

Architecture

Interface

### 3.4 Shooting Model

Purpose

Architecture

Interface

### 3.5 Shot Optimizer

Purpose

Architecture

Interface

### 3.6 Data Storage

Purpose

Architecture

Interface

### 3.7 User Interface

Purpose

Architecture

Interface

### 3.8 Shooting Mechanism

Purpose

Architecture

Interface

## **4 Class Responsibility Collaboration (CRC) Cards**

who handles what requirements and who do they work with to do that

### **4.1 SmartServe CRC Cards**

### **4.2 Computer Vision CRC Cards CRC Cards**

### **4.3 Shot Recommendation CRC Cards**

### **4.4 Shooting Model CRC Cards**

### **4.5 Shot Optimizer CRC Cards**

### **4.6 Data Storage CRC Cards**

### **4.7 User Interface CRC Cards**

### **4.8 Shooting Mechanism CRC Cards**