Report #1 - SSR Project Proposal

**Table of Contents**

1. Introduction…………………………………………………………………………………………………..

1.1. System Project Manager…………………………………………………………………...

1.2. Problem & Proposal Solution …………………………………………………………..

1.3. Definitions, Acronyms, and Abbreviations………………………………………….

2. Specific Requirements & Architecture.…………………………………………………………….

2.1. Functional Requirements………………………………………………………………….

2.2. Software Requirements…..………………………………………………………………..

3. Interfaces …………………….….……………..…………………….………………………………………..

3.1. User interface…………………………………………………………………………………..

3.2. Hardware interface…………………………………………………………………………..

3.3. Software interfaces…………………………………………………………………………..

3.4. Communication interfaces…………………………....................................................

4. Market Survey………………………………………………………………………………………………..

5. Appendices…………………………………………………………………………………………………….

5.1. Work plan…………………………………………………………………………………………

5.2. Risk Management……………………………………………………………………………..

5.3. UML Diagrams………………………………………………………………………………….

5.4. Database Collections…………………………………………………………………………

5.5. Use-Cases………………………………………………………………………………………....

5.6. User Requirements Document…………………………………………………………….

**1. Introduction**

This stage gives a scope description and overview of the projects first steps. Planning management aspects and analyse requirements of the product being developed. Offers a preliminary solution and investigates the software engineering characteristics of the project.

**1.1 System Project Manager**

|  |  |  |
| --- | --- | --- |
| Link | System | # |
| [Link1](https://github.com/dor132133/SSR-Smart-Shooting-Range) | Repository | 1 |
| [Link2](https://github.com/dor132133/SSR-Smart-Shooting-Range/wiki) | Diary | 2 |
| [Link3](https://github.com/dor132133/SSR-Smart-Shooting-Range/projects/1) | Project Manager | 3 |
| Tablet with Installed application | Distribution | 4 |

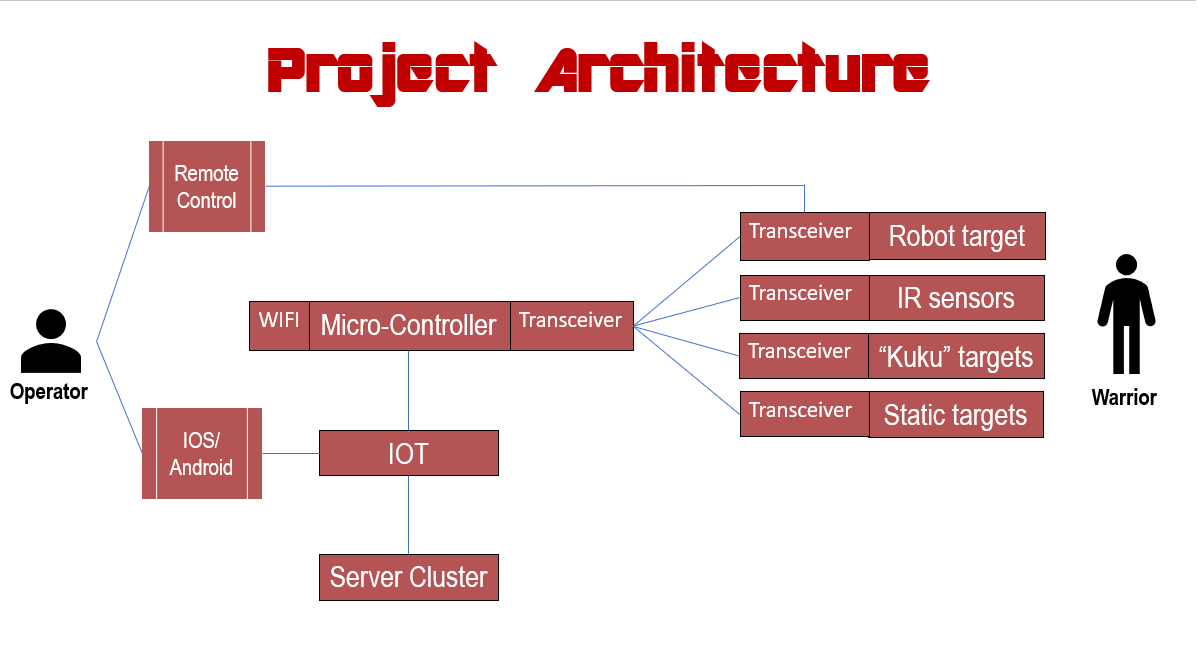
**1.2 Problem & Proposal Solution**

Today, shooting range clubs find it **difficult** to manage and document their activities on an ongoing basis. The ability to systematically follow different and varied firing sessions by many warriors, **hardly exists**. The training process of a warrior requires the ability to monitor his physical and professional achievements in order to make the training process more efficient and professional.

In addition, there is a desire and need to improve the shooting range itself, in order to give the warrior a realistic experience as much as possible. Therefore, a smart shooting range will answer this need by a command and control system of active and passive hardware components.

**The Solution**

A smart shooting range with robotic targets and motion sensors, which are autonomously controlled according to various predetermined plans. In addition, a command and control application will be added, with the ability of processing and managing data, calculation of speeds and times, statistical calculations and issuing reports.



**1.3. Definitions, acronyms, and abbreviations**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| SSR | Smart Shooting Rang |
| SR | Shooting Range |
| MC | Micro-Controller |
| Transceiver | A device comprising both a transmitter and a receiver that are combined and share common circuitry or a single housing. |
| Warrior | Someone who using the SR as a shooter, can be a costumer or guide. |
| Operator | Someone who run the SR scenario, using the management app. |
| Administrator | System administrator who is given specific permission for managing and controlling the system. |
| Web-Server | A server which provide RESTful API. |
| RESTful API | an application program interface (API) that uses HTTP requests to GET, PUT, POST and DELETE data. |
| Node-JS | Node.js is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code outside of a browser. |
| Node-JS server | handling multiple events asynchronously. |
| ESP | The chosen micro-controller having WIFI module. |
| Front-end | The mobile application. The part of the system that the operator can see and interact with. |
| Back-end | The server services – web-server, database, and other processing services. |
| SR components | All shooting range objects in the field: walls, robotics targets, motion sensors and recoil sensor |
| SR scenario | Session of all shooting events of a warrior during a runway, while recording all events. |
| Robotics targets | Motorized shooting targets with recoil sensor, micro-controller and transceiver module. |

**2. Specific Requirements**

This section contains all the functional and quality requirements of the system. It gives a detailed description of the system and all its features.

**2.1 Functional Requirements**

In many countries there is a need to train security personnel for many various purposes. Israel known as a country that engages heavily in defense and the training of combatants.

In Israel, there are quite a few shooting clubs, both private and public, which prepare the fighters to be ready when the time comes. The need to **streamline** the training process is requested, both in terms of time, both in terms of quality and in terms of money.

In order to meet these requirements, the proposed idea is to **convert** existing shooting ranges to "smarter" by installing a number of technological systems with good integration capability.

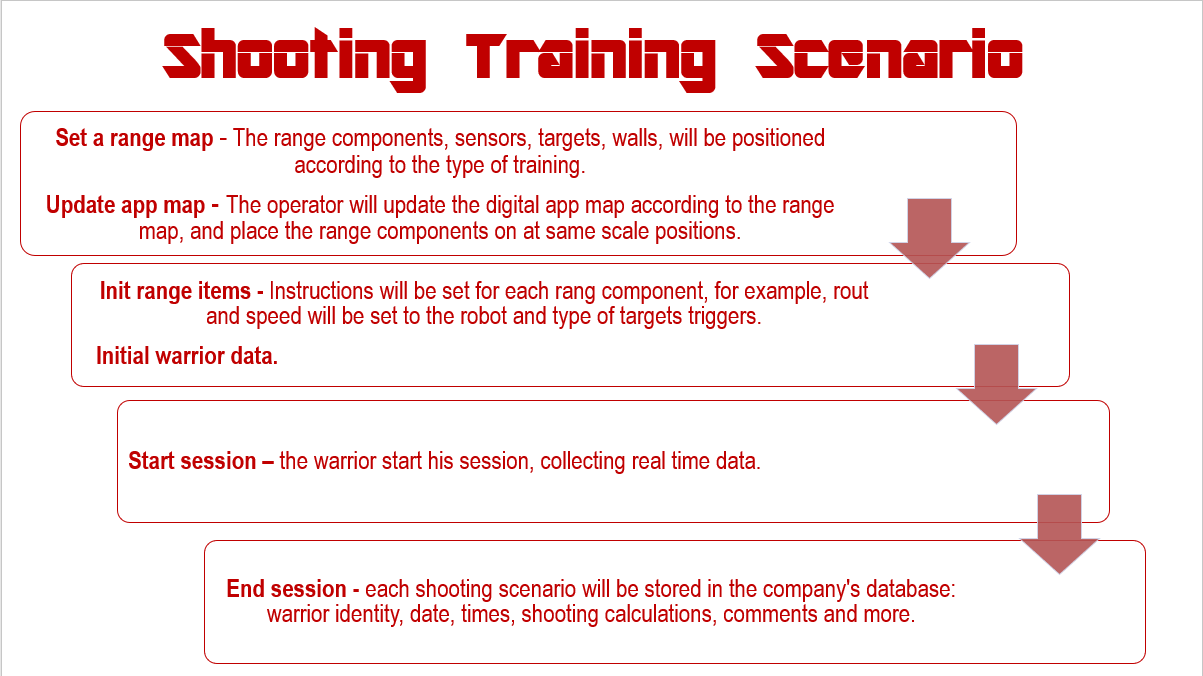
First, a **mobile management and control application** is required.

* SR operator and administrator will be the major application costumers.
* The application will include:
  + A graphic mapping of the shooting range and arrangement of the SR components on the map.
  + Create and view of reports and lists.
  + Management of warriors personal and professional information.
  + Displays statistical information of warriors and teams.
  + Launching of SR scenarios system and displays all information recorded during the session.

A **command and control hardware system**, each SR component installed by MC and transceiver in purpose to communicate with the main shooting range controller – the ESP.

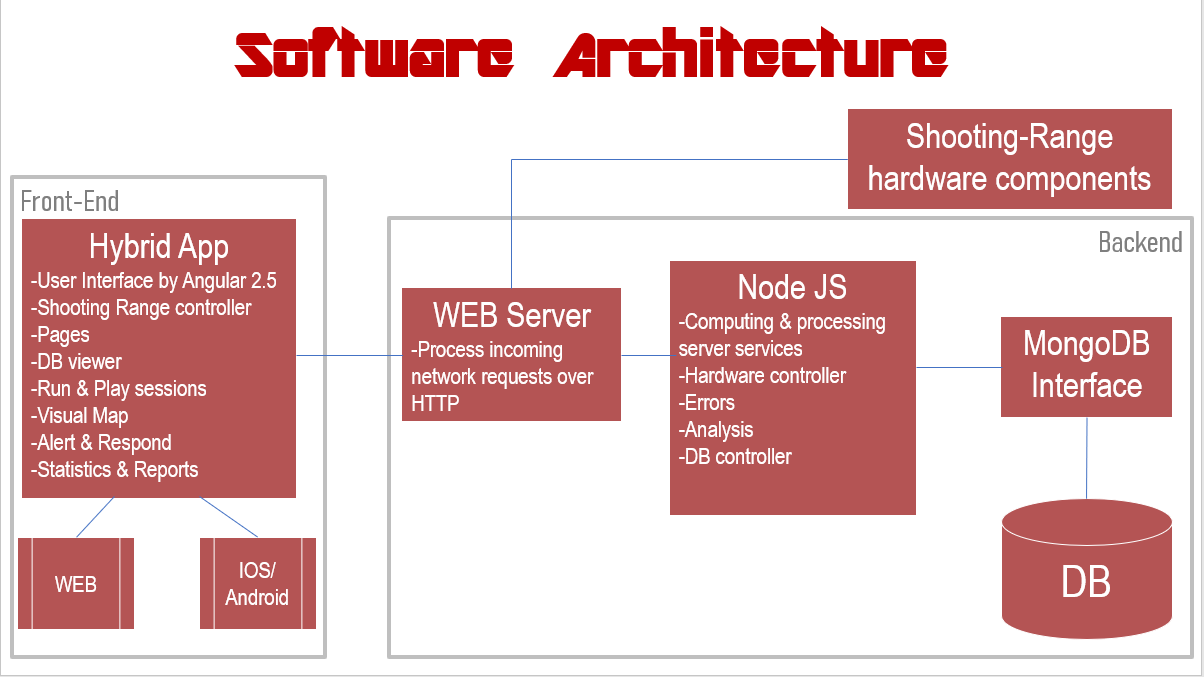
The ESP manage the SR scenario "in the field" listening and responding to any SR scenario events – target shooting hits, sensors crossing and targets movement. In addition, The ESP documents all SR scenario events according to their occurrence.

A **main server cluster** provides all processing, communications and data management services. Functions as an integrator between the frontend app and the ESP part. The communication with the main server performed by a web-server service through the global internet network.

**SR scenario**, at the beginning of each shooting session, the operator determines the map structure and push the start button. Since than all the scenario runs autonomously, act and response according to the warrior moves. At the end all sessions records are send back to the main server for processing and storing.

**2.2. Software Requirements**

Software development has some serious challenges. Changes in software development are unavoidable, changes occur rapidly and accommodating these changes to develop complete software. SSR project has a few software challenges that requires a good architecture design and well making decisions. There are 3 software parts in this project, frontend app that runs on mobile tablet, main server cluster that divided into several services and real time micro-controller side. All parts should integrate perfectly, mainly because each part run on a different machine, programmed on different work frames in a different languages. Another important challenge is the communication issue that should be accessible and scheduled between all parts. each part belongs to different network, for instance, the main server runs on some cloud/remote machines, while the ESP connecting to the local SR network using WIFI module.



The chosen work frames and the programme languages are:

* Angular2.5 writing in TypeScript for the mobile hybrid application.
* Node-JS writing in JavaScript for the main server services.
* C/C++ for the ESP real time chip.
* MongoDB as a database service.
* RESTful API as the main server gatekeeper and communicator.
* TCP/IP as a communication protocol.
* \*\* ESP using RF transceivers to communicates against all SR components

**3. Interfaces**

This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.

**3.1. User Interface**

First, user of the mobile application (operator) should see the home project page, see Figure 1 when he/she opens the application the app set communication with the backend server and check for "healthiness". If the connection fails, an error message will appear. From the home page the operator can choose one of the displayed. Warrior will open warriors age with an option to search wanted warrior, see Figure 3, add new warrior and edit an exist warrior profile. Every warrior has its own personal information about his training process.

Figure 1

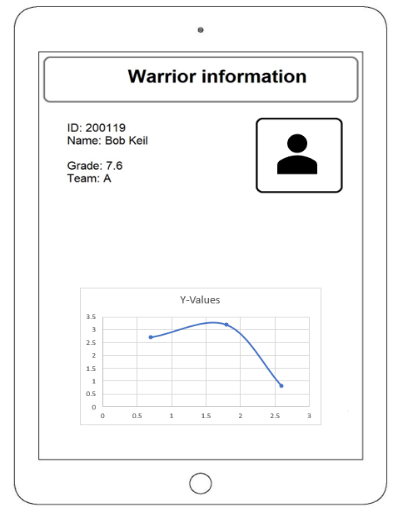
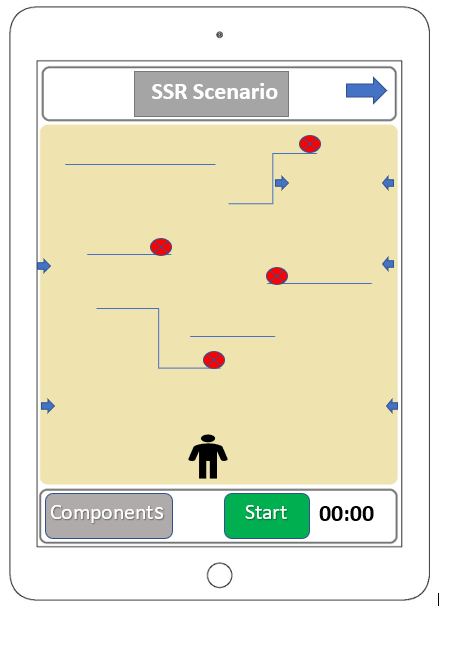


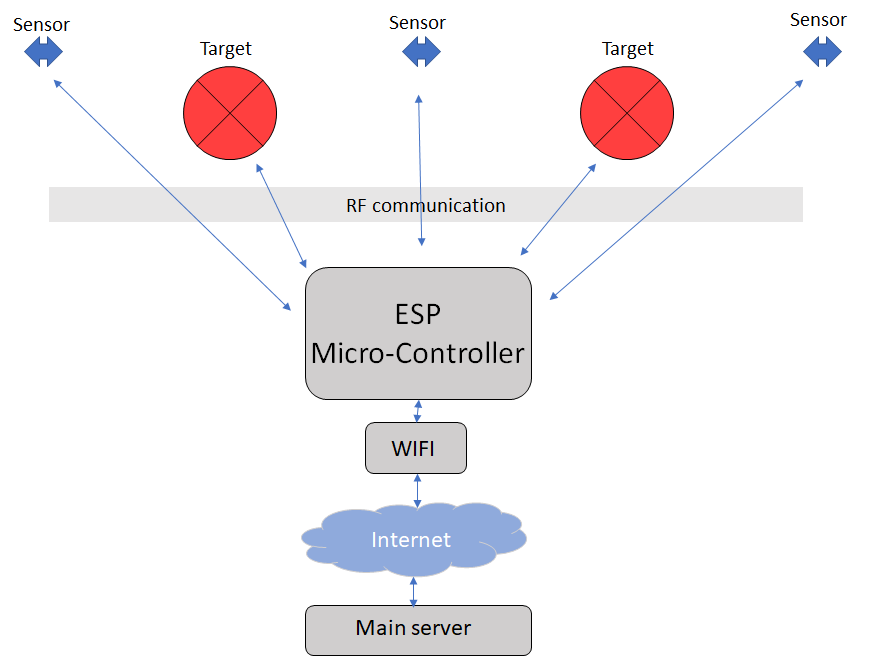
Figure 2

Figure 3

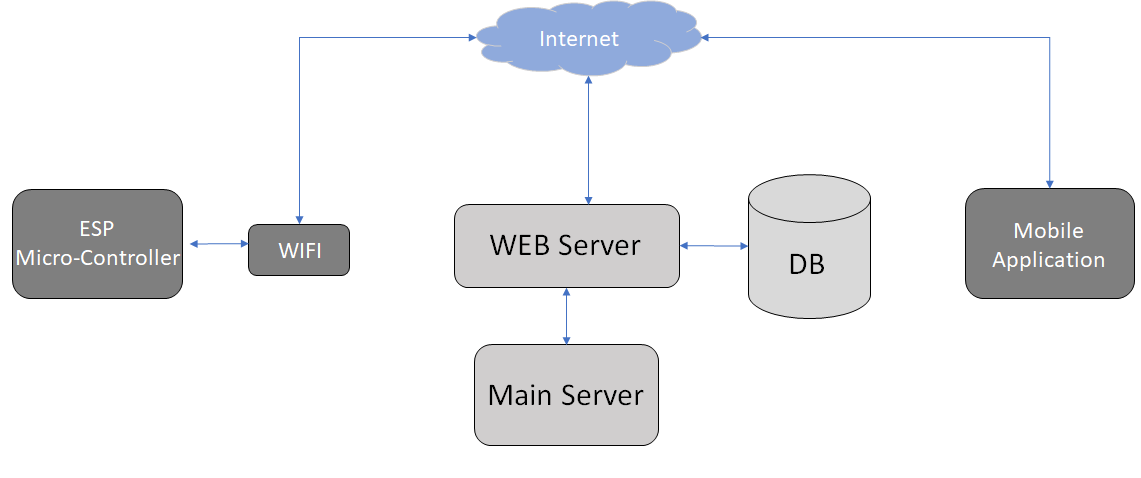
In the scenario page, the operator determines the map status, he set each SR components (targets, sensors, walls) at the current position, in addition he set instruction for each component, for instance, "target #1 activate by sensor #1". When everything ready the operator press "Start". When his press Start, all of the map information (instructions) are send to the backend service that deliver it to the MC in the field.

**3.2. Hardware Interface**

Micro Controller and hardware components should communicate autonomously right after the instructions set sent from the mobile app through the backend server. MC code should be able to polling each sensor and target in the arena, response accordingly and document everything that goes on.

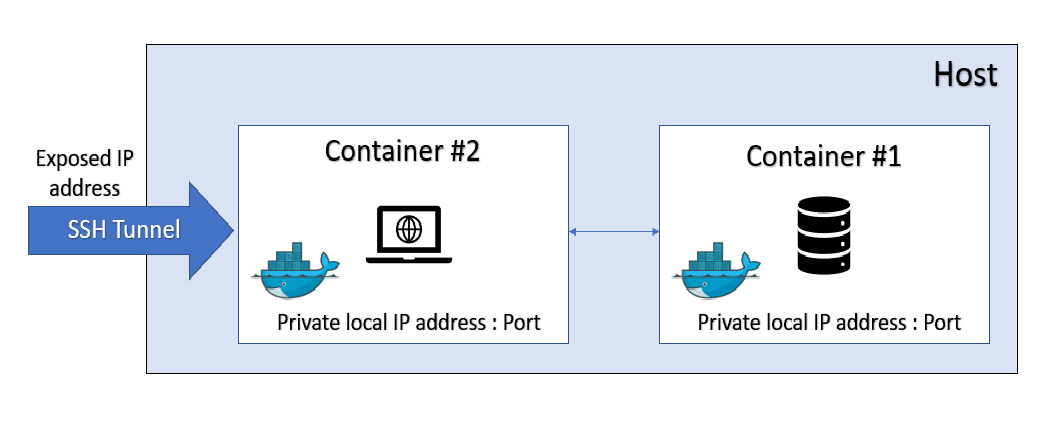


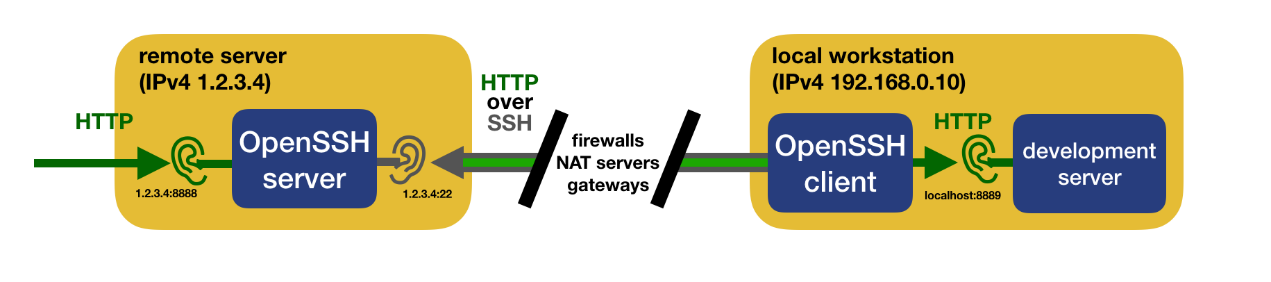
**3.3. Software Interface**

****The user application and the MC use the web server service to communicate, send and receive data. The web server has REST API that provide that functionalities. In addition the web server maintain data, using database, in order to get MC and the app information about any of their relevant data. The communication between the database and the web server consists of operation concerning both reading and modifying the data.

**3.4. communication Interface**

The communication between the different parts of the system is important since they depend on each other. The main communication leaning on a TCP/IP protocol through the global internet network, since the main services are not really physically in the field, they can deploy to any desire machines. In addition, the MC and the mobile application using WIFI chip to connect the internet. The web server (REST-API service), main server and database service are runs on Docker's containers in some host OS machine, and communicate each other in a local bridged network.



 Important to mention that the web service expose it's IP address through SSH tunneling protocol. The router that stands between our workstation and the internet makes it harder to expose a local socket to the outside. Most of the time, this is preferable for security. A typical home network is composed of a router and a number of devices connected to the router. The router is assigned a public IP address by the internet provider. Every device in the home network, including the router itself, is assigned a private IP address, usually in the form 192.168.x.x. The router acts as a gateway between the home devices and the internet. This gateway performs network address translation

(NAT) between the public and the private networks. While this allows a local device to reach a public IP address, it is usually not possible to initiate a communication from the outside.**Tunneling services** Some services offer you to set up a tunnel between their servers and your local workstation. These services can be very easy and convenient to use and provide advanced features such as monitoring and statistics, domain name and HTTPS support. However, these services are either charged or subject to limits and you may want to avoid a third-party accessing your data.

**4. Market Survey**

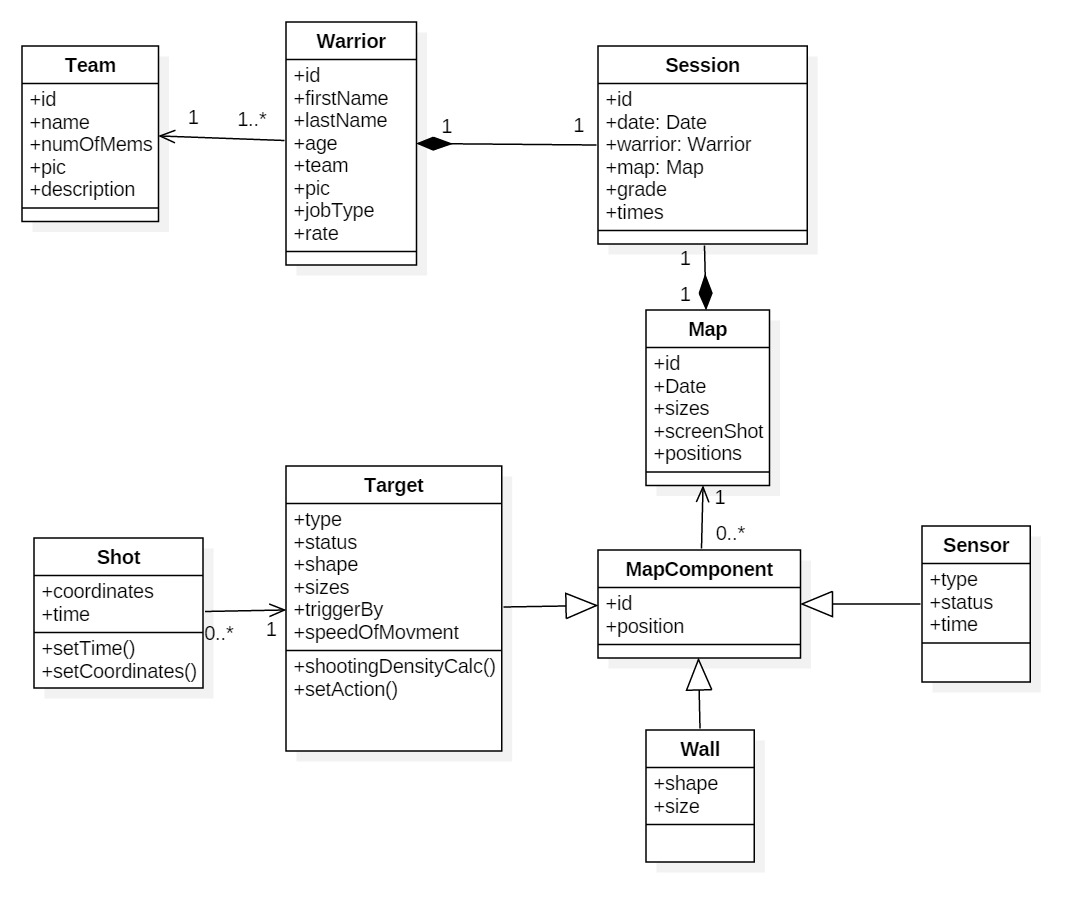
**5. Appendices**

**5.1. Work Plan**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Mission | Description | Date |
| 1 | Meeting with the client. | Introduce meeting with the client, talking about the problem. | 10.13.18 |
| 2 | Second Meeting with the client. | Solutions suggestions | 10.22.18 |
| 3 | Micro controllers, motors and sensors. | Selection of mechanical and electrical components. | 11.01.18 |
| 4 | Hardware Control. | Mechanical control of all shooting range hardware components: moving targets and sensors | 11.11.18 |
| 5 | First design of project environment. | includes hardware and software components, according to the needs of the system, environmental constraints and economic constraints. | 11.30.18 |
| 6 | Project architecture design | Primary project architecture design – according to client requirements | 12.06.18 |
| 7 | Software architecture design | Primary software architecture design – according to client requirements | 12.12.18 |
| 8 | Report #1 | Start working on project proposal | 12.13.18 |
| 9 | Setting up a web server | The main services integrator, RESTful - API | 12.20.18 |
| 10 | Database | Establishing a database service. | 12.25.18 |
| 11 | Web-server and DB integration | Create collection and insert documents to db using web server | 12.30.18 |
| 12 | First ESP and Main server communication | Configured the ESP to be able communicate external internet network | 01.07.19 |
| 13 | First mobile application components | Start writing mobile application | 01.16.19 |
| 14 | Pages Creations | Mobile app. | 01.20.19 |
|  |  |  |  |

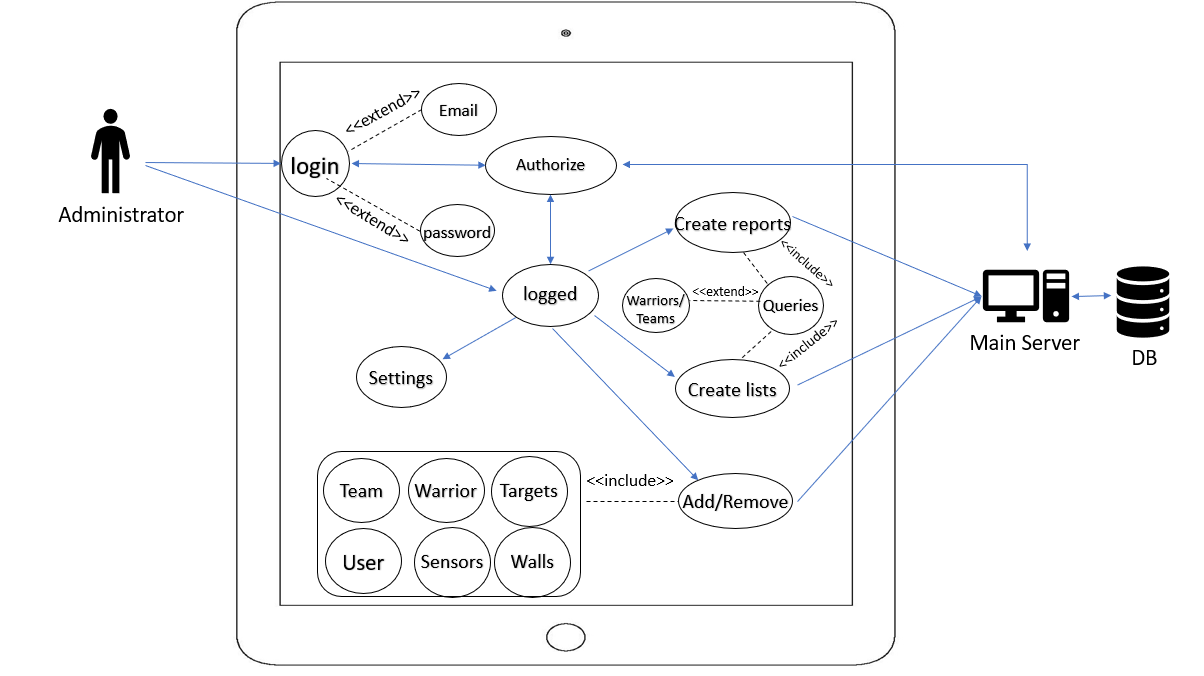
**5.2. Risk Management**

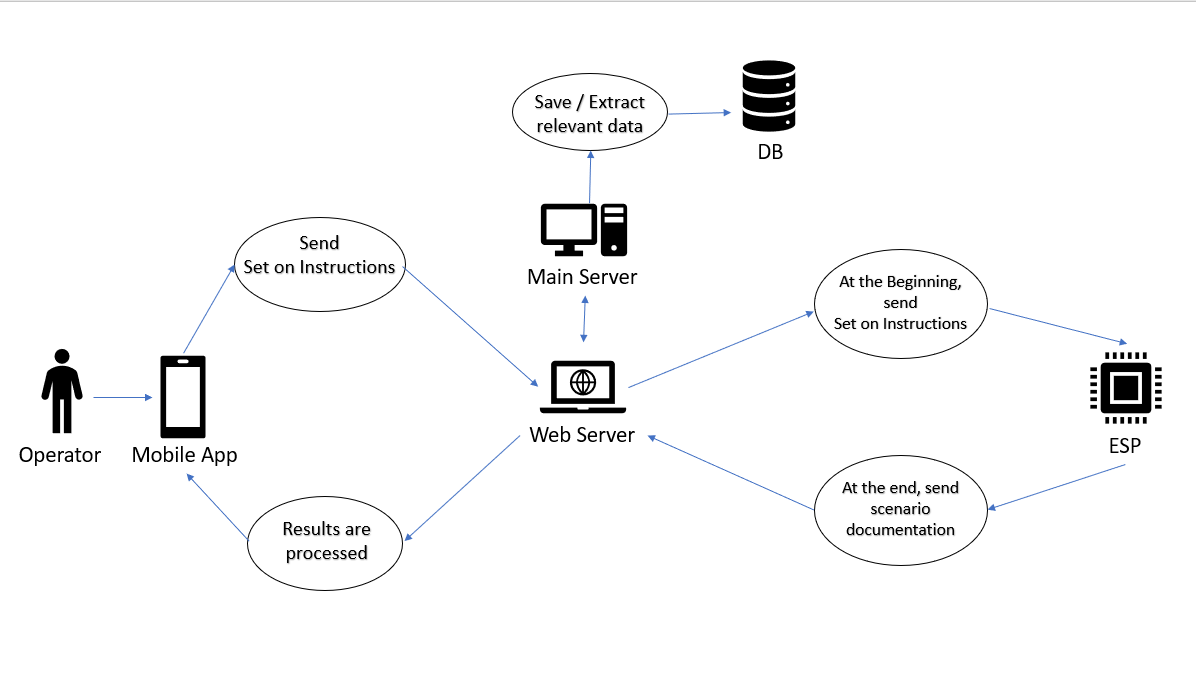
|  |  |  |  |
| --- | --- | --- | --- |
| Risk | Severity | Response |  |
| 1 | Wrong working plan design. | High | Clearly define goals, sub-goals and milestones placing. |
| 2 | Project scope extends beyond project limits. | High | Reach complete understanding between client, project programmer and project Supervisor regarding expectations, abilities and project requirements. |
| 3 | Working with new tools in an unfamiliar environment | Medium | Getting familiar with the software tools used for development by attending classes, hands down experience, and various tutorials. |
| 4 | Change requirements | Medium | The client changes its requirements in such a way that shifts the timetable of the project or defines the project as not meeting client's requirements. |
| 5 | Incorrect system size assessment | High | Provide the main requirements by adjusting them to constraints. |
| 6 | Communicating with the customer | Medium | The customer is busy and it can cause delay on development, if the product is as expected or not. the answers can be delayed, and it will influence the whole project. |
| 7 | Dependence on multiple hardware components | High | Interfacing between different hardware components and using many communication protocols increases the likelihood of communication falls. |
| 8 | Too many errors during testing | Medium | Finding too many bugs during testing can delay the termination of the current iteration in such a way that can delay the termination of the project in time. |
| 9 | Use new technology | High | As a result of the client requirements it been ask to create a realistic shooting range, which functions and responds to combat operations in the field while integrating a control system |
| 10 | Real Time response | Medium | Control of mechanical and electronic components should response in a short time. |

**5.3. UML Diagrams**

**5.4. Database Collections**

**5.5. Use-Cases**





**5.6. User Requirements Document**