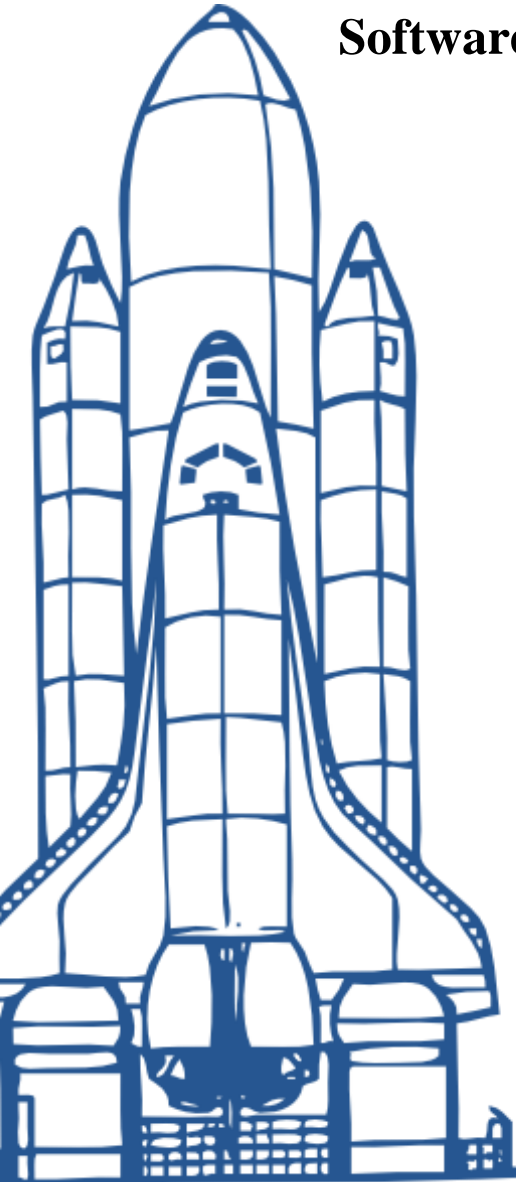




**Middle East Technical University**  
**Department of Computer Engineering**

# **Target Tracking by Using Seismic Sensors**

## **Software Requirements Specification Report**



**Team spaceShuttle**

Mustafa MIZRAK

Hüseyin ÜNAL

Anıl ERKOÇ

Ali Fatih GÜNDÜZ

2011

---

# TABLE OF CONTENTS

---

1	INTRODUCTION .....	3
1.1	PROBLEM DEFINITION .....	3
1.2	PURPOSE .....	4
1.3	SCOPE .....	4
1.4	USER AND LITERATURE SURVEY.....	5
1.5	DEFINITIONS and ABBREVIATIONS .....	6
1.6	REFERENCES.....	7
2	OVERALL DESCRIPTION .....	8
2.1	PRODUCT PERSPECTIVE .....	8
2.2	PRODUCT FUNCTIONS.....	9
2.3	CONSTRAINTS, ASSUMPTIONS and DEPENDENCIES .....	13
3	SPECIFIC REQUIREMENTS .....	13
3.1	INTERFACE REQUIREMENTS .....	13
3.2	FUNCTIONAL REQUIREMENTS .....	14
3.2.1.1	Add Sensor.....	14
3.2.2.1	Determining Track Positions on the Map .....	16
3.2.2.2	Send Track to the Map.....	16
3.2.2.3	Save to Database .....	16
3.2.2.4	Create Log File .....	17
3.3	NON-FUNCTIONAL REQUIREMENTS.....	17
4	DATA MODEL AND DESCRIPTION .....	18
4.1	DATA DESCRIPTION .....	18
4.2	COMPLETE DATA MODEL AND RELATIONSHIPS .....	19
4.3	DATA DICTIONARY .....	20
5	BEHAVIORAL MODEL AND DESCRIPTION .....	23
5.1	DESCRIPTION FOR SOFTWARE BEHAVIOUR.....	23
5.2	STATE TRANSITION DIAGRAMS .....	25
6	PLANNING.....	27
6.1	TEAM STRUCTURE .....	27
6.2	WORK FLOW .....	28
6.3	PROCESS MODEL.....	29
7	CONCLUSION.....	29

---

# 1 INTRODUCTION

---

This Software Requirements Specification (SRS) document provides a complete description of all requirements, interfaces, design issues, and component specifications of "Target Tracking by Using Seismic Sensors" project, senior design project of *spaceShuttle* team at Middle East Technical University Department of Computer Engineering and sponsored by ASELSAN Inc.

## 1.1 PROBLEM DEFINITION

---

Seismic sensors detect vibrations occurred on the ground by determining pressure changing. They are very sensitive and can detect vibrations caused by the movement of animals or human beings. People or objects moving over ground generate a succession of impacts and these soil disturbances propagate away from the source as seismic waves. Hence, they can be used to monitor protected areas to restrict entry of unwanted persons or animals.

The problem is real time object tracking by analyzing the waves of seismic sensors the object generates while it moves over the ground. In this project, *spaceShuttle* team will develop a target tracking algorithm such that it monitors route of moving objects/targets within an area using large-scale-deployed and spanned-hundreds-of-kilometers seismic sensors provided by Aselsan Inc. Moreover, team will implement target tracking simulator based on a Geographic Information System (GIS) to monitor deployed seismic sensors and route of the moving objects (real object and from others ).

## 1.2 PURPOSE

---

This document is intended to specify the requirements for Target Tracking by Using Seismic Sensors project that implements the track of a moving target within an area where seismic sensors are previously deployed. Dependencies, functional and non-functional requirements, interfaces, design constraints, data model and descriptions, usage of the program and also plan of the project consisting process timeline, team structure and division of work according to members are addressed.

The main purpose is to establish a core agreement between the developers in the *spaceShuttle* group and the project supplier on what this project is to do. It is prepared to cover all requirements before detailed design of the project in order to not redesign, recode, and retest the project. Furthermore, this document aims to eliminate misunderstandings and inconsistencies at the first stage/phase of the development process of the project. Also, all requirements and analysis specified in this document will be useful for the potential developers who want to work on Target Tracking, and they can modify these requirements and analysis to their needs.

## 1.3 SCOPE

---

Target Tracking by Using Seismic Sensors project can be divided into three parts. The first is developing a target tracking algorithm. Collected data/signals/alarms from seismic sensor previously deployed will be stored in a database and they will be interpreted and classified with *this algorithm* in order to detect the route of the moving objects/targets. The second part is about the simulation based on Geographic Information System (GIS). All sensors will be displayed on fully 3D globe map. Furthermore, all determined track of moving objects that are determined by team algorithm using real seismic sensors, and route of drawn with joystick, mouse and keyboard devices will be displayed on the map. The last part is about the command control. All information including the track of the objects and all sensor information (locations and IDs) will be managed with command control.

## 1.4 USER AND LITERATURE SURVEY

---

In order to stop terrorist raids from cross boundary, Turkish Ministry of Defense launched electronical eye project. With the directive of current Minister of the Turkish Ministry of Defense, Ismet Yilmaz, Turkish Defense Industry focused on this project. Aselsan is one of those companies interested in the *Electronical Eye Project*.

This project aims to remove boundary guard stations due to terrorist attacks. Instead of those stations, special equipment will be deployed on boundaries. The system will be camouflaged like stones and they will be placed at strategic points.

The equipment consists of seismic sensors, thermal camera and gun. In case of an intrusion, seismic sensors will alert the headquarter. Officers in the headquarter will hold down the intrusion by watching the images coming from the camera. After confirmation of the intrusion by the officers, the headquarter will be able to attack those targets by remotely controlling the gun of the system.

So before the terrorist approach to the strategic points, headquarter will be aware of them and even will be able to engage with them without losing any personnel. Moreover, the cost of border security will be lower since it will require less personnel and guard station.

Our project is a part of this system. We will be dealing with seismic sensors and tracking the intruders. A buried seismic sensor detects seismic activity at the detector location and generates an electrical signal that is analyzed by the processor to determine if a specified type of intrusion has occurred. A single seismic detector can be deployed to monitor a small area or trail, or detectors can be combined in a string to monitor large open areas or perimeters. Seismic activity information sent to the processor is distinguished between human and vehicular traffic by processing.

This system with some modification can be used at risky and sensitive areas such as airports, military establishments, international borders, power stations, governmental establishments, nuclear power plants, oil & gas refineries, research establishments, embassies, residential establishments of individuals at risk from kidnap or assassination,

banks, industrial plants, etc. This system is also able to be used against immigration, smuggling and to provide security of petroleum pipe lines as well.

Similar intruder detection technologies are currently used by US Border Patrol, US Military, British Petroleum, NATO, Australian Attorney General's Office, Australian Department of Defense, Chubb Electronic Security, Duke Energy, Egypt LNG etc.

## 1.5 DEFINITIONS and ABBREVIATIONS

---

**SRS:** Software Requirements Specification

**GIS:** Geographic Information System

**KF:** Kalman Filter, it is to use measurements observed over time, containing noise and other inaccuracies, and produce values that tend to be closer to the true values of the measurements and their associated calculated values.

**IKF:** Intelligence Kalman Filter, is derived for tracking the manoeuvring target model where the time varying variance of process noise is computed in an intelligent manner using a fuzzy system.

**SS:** Seismic Sensor

**UML:** Unified Modeling Language, a standardized general-purpose modeling language in the field of software engineering

**DB:** Database (PostgreSQL Database)

**GUI:** Graphical User Interface, a type of user interface that allows users to interact with programs with images rather than text commands

## 1.6 REFERENCES

---

- [1] IEEE Std 830-1998 (Revision of IEEE Std 830-1993): IEEE Recommended Practice for Software Requirements Specifications
- [2] Schach, Stephen R. Object-Oriented and Classical Software Engineering, Fifth Edition, McGraw-Hill, 2002
- [3] B.J. Lee, J.B. Park, Y.H. Joo and S.H. Jin, Intelligent Kalman filter for tracking a manoeuvring target, IEE Proceedings, 2004
- [4] M.I. Ribeiro, Kalman and Extended Kalman Filters: Concept, Derivation and Properties, Institute for Systems and Robotics Instituto Superior Tecnico, Portugal
- [5] J. P. Lielferty, Improved Tracking of Maneuvering Targets: The Use of Turn-Rate Distributions for Acceleration Modeling, IEEE Transactions on Aerospace And Electronic Systems Vol. 32, NO.4 October 1996
- [6] D. Li, K. D. Wong, Y. H. Hu and A. M. Sayeed, Detection, Classification and Tracking of Targets in Distributed Sensor Networks, Department of Electrical and Computer Engineering, University of Wisconsin-Madison, USA

---

## 2 OVERALL DESCRIPTION

---

### 2.1 PRODUCT PERSPECTIVE

---

In this project, there are two main tasks to be accomplished. The first task is target tracking by using seismic sensors. The other part is showing targets which are tracked by seismic sensors in part I on real map.

#### a. Target Tracking:

To detect targets path, signal strength, state of each sensor, sensor ID, and position of each sensor should be stored in database. These values should be measured periodically. Weather conditions, sensor deployment, seismic sensor signal strength and ground conditions can effect these measurements. This database is used for detecting possible and real positions of target. Kalman Filter can be used for this purpose.

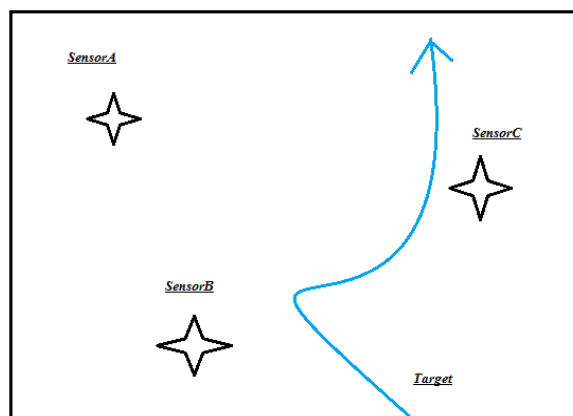


Figure: Sensors and Target

**b. Showing targets on a map:**

Detected positions of target are shown in this part. In this part target and sensors should be shown in real map. It can be thinkable; this part is simulation part of our project. User can visualize real position of target. GPS signal strength and seismic sensor communication affect this part. And it is also possible simulating target tracking application in this part. User can enter positions o sensors and user also can enter starting point of target. After that, this part can handle target tracking simulation.



Figure: Showing and simulating sensors

## 2.2 PRODUCT FUNCTIONS

---

Our system works with human actors. Human actors are user who has technical information about our application.

## 2.2.1 ADMIN LOGIN

---

<b>Use Case</b>	Login
<b>Primary Actor</b>	Admin
<b>Goal in context</b>	Accessing to system
<b>Scenario</b>	<ol style="list-style-type: none"><li>1. The system prompts the admin for their username and password.</li><li>2. The admin enters username and password</li><li>3. The system gets password registered to username</li><li>4. The system verifies the password</li><li>5. Admin enters the system to arrange users and other options.</li></ol>

## 2.2.2 REMOVE USER

---

<b>User Case</b>	Remove User
<b>Primary Actor</b>	Admin
<b>Goal in context</b>	Removing user
<b>Scenario</b>	<ol style="list-style-type: none"><li>1. Admin chooses a user from user list</li><li>2. Admin clicks Remove button</li><li>3. User is removed from user list</li></ol>

### 2.2.3 ADD USER

---

<b>User Case:</b>	Add User
<b>Primary Actor:</b>	Admin
<b>Goal in context:</b>	Adding user to database
<b>Scenario:</b>	<ol style="list-style-type: none"><li>1. Admin click add button in user management window.</li><li>2. Admin enters the user details.</li><li>3. User is added to database.</li></ol>

### 2.2.4 USER LOGIN

---

<b>User Case</b>	Login
<b>Primary Actor</b>	User
<b>Goal in context</b>	Accessing to system.
<b>Scenario</b>	<ol style="list-style-type: none"><li>1. The system prompts the admin for their username and password.</li><li>2. The admin enters username and password</li><li>3. The system gets password registered to username</li><li>4. The system verifies the password</li><li>5. Admin enters the system to arrange users and other options.</li></ol>

## 2.2.5 SHOW TARGET PATH

---

<b>User Case</b>	Show target path
<b>Primary Actor</b>	Admin or User
<b>Goal in context</b>	Visualizing path of target
<b>Scenario</b>	<ol style="list-style-type: none"><li>1. User or admin enter to system.</li><li>2. User or admin click the show on map button.</li><li>3. Target path is visualized on map.</li></ol>

## 2.2.6 SIMULATE SEISMIC SENSORS

---

<b>User Case</b>	Simulate Seismic Sensors
<b>Primary Actor</b>	Admin or User
<b>Goal in context</b>	Simulating seismic sensors.
<b>Scenario</b>	<ol style="list-style-type: none"><li>1. User or admin enter to system.</li><li>2. User or admin click the “simulate sensor” button</li><li>3. User enters sensor count.</li><li>4. User enters positions of each sensor.</li><li>5. User enters starting position of target.</li><li>6. User clicks the “simulate now” button.</li><li>7. Path is simulated on screen.</li></ol>

## 2.3 CONSTRAINTS, ASSUMPTIONS and DEPENDENCIES

---

1. Sensors are working fine.
2. Dataset is sufficient enough and dataset is handling many conditions enough.
3. There is no connection problem between sensors.
4. All libraries are on the system.
5. Seismic sensor signals gain distance is our constraint.

---

# 3 SPECIFIC REQUIREMENTS

---

## 3.1 INTERFACE REQUIREMENTS

---

There will be a simulation interface environment. The simulation environment will display a screen in which a real 3D world map is represented to users. The users will place the sensors on the map by simply clicking ADD SENSOR button. The positions of the sensors are needed by the system and should be provided by a pop-up window to the system to work properly. The sensors added to the system will be displayed on the map. Those sensors will start to send data to the system after they are added and as long as they are alive, not prevented by the user, they will continue to send signals and system will start to gather those signals to process.

The system will provide two options to users to work. First, the users will be able to use real sensors which are planted in the ground and sends real signals via zigBee. Those signals will be obtained by the sensors as objects move on the selected zone and they will be interpreted by the system. According to this interpretation, the system will draw the predicted track of the moving target on the map. The other option provided to the users is to work with imaginary data. Users will be able to create tracks on the map by using joysticks, mouse or keyboard.

The original track drawn by the user will be shown on the map. . After taking the track data from the user, the system will interpret the data as it was obtained by the real hardware sensors and predicted track will be created and compared with the original one on the screen. As stated above, the users will be able to add sensors on the map draw tracks. Further they will be able to delete those items from the screen. Users may select to disable the sensors as they wish, too. The detailed information such as position of the sensor and signal strength obtained from the sensors will be displayed in a second menu so that users will be able to control those data.

The system will give its output in pdf format. The users will be able to enter the time stamp they wish and the system will give the interpretation of the data in that time stamp and if exists any, track will be drawn on the real world map.

## 3.2 FUNCTIONAL REQUIREMENTS

---

There will be two kinds of functional requirements of the system.

### 3.2.1 SIMULATION ENVIRONMENT PROGRAMMING

---

#### 3.2.1.1 Add Sensor

---

The users will be able to add sensors and place them on the map on as they choose. The number of sensors is up to users.

#### 3.2.1.2 Give Imaginary Track Information

---

The users will be able to draw imaginary tracks on the map to give as input. The system will interpret this data as original data and make the computations on them. This track information will be given by keyboard, mouse or joystick.

### 3.2.1.3 Show Sensors

---

The system may hide or show the sensors on the map according to choice of the user.

### 3.2.1.4 Delete Sensor

---

The users will be able to delete the sensors they placed on the map before.

### 3.2.1.5 Disable Sensor

---

The users will be able to choose to ignore the data coming from special sensors if they wish. This will be used mainly if some sensors are thought to be sending wrong data. In this case they will be omitted and the system will make necessary calculations without regarding them.

### 3.2.1.6 Display Sensor Information

---

The data coming from sensors will be displayed by the system when the users want to see and check them.

### 3.2.1.7 Draw Track on the Map

---

The system will draw the track on the map as lines. Both the predicted track of the target and the track taken as input from the user will be able to drawn.

### 3.2.1.8 Report as PDF

---

When the user wants, the system will give the obtained information as output in a PDF file. The predicted tracks will also be able to draw in PDF files too.

## 3.2.2 TRACK MANAGEMENT

---

This will be the most important part of the system. There are four functional requirements of this part.

### 3.2.2.1 Determining Track Positions on the Map

---

When the system is launched by the users, the system will calculate the data obtained from the sensors or given as an imaginary track by the user, and then it will predict the track of the intruder.

### 3.2.2.2 Send Track to the Map

---

When it is demanded from the simulation component, the system will send the positions data of the track to the map which was predicted formerly according to given data.

### 3.2.2.3 Save to Database

---

The system will save the track taken from the users as input in its database. Moreover, the system will save the data coming from the sensors in database in order to calculate next positions of the created tracks.

#### 3.2.2.4 Create Log File

---

The system will save every action it did in the database in log file as well.

### 3.3 NON-FUNCTIONAL REQUIREMENTS

---

#### 3.3.1 RELIABILITY

---

The system is to draw the most accurate tracks on the screen. Since this is a product which is used for defensive purposes, the error should be at the minimum. The system is to predict the track of the intruder correctly.

As mentioned in the market research part, main customer of the product will be defense industry and it will be used to detect intrusions to borders, reliability is crucial. False alarms may be inevitable but should be kept at minimum.

#### 3.3.2 PERFORMANCE

---

This is a real time project and performance is important as much as correctness. The intrusions must be detected and the headquarter should be alarmed as soon as possible.

The most of the performance requirement will be needed by the simulation part of the system.

### 3.3.3 DESIGN

---

We will use Java in the project to create simulation part. Moreover we will be using C++ to get data from gateway in order to use in communication component. In the simulation part, NASA World Wind open source API will be used to show sensors and tracks on it as map. GIS will be used to hold the positions of the sensors and tracks.

### 3.3.4 PORTABILITY

---

In this project all parts except gateway software will be written in Java. Therefore the end product will work at many platforms since Java provides high portability. This system will be able to work on Windows, MacOS and Linux as well.

---

## 4 DATA MODEL AND DESCRIPTION

---

This section will give information data objects, class attributes, and data relations.

### 4.1 DATA DESCRIPTION

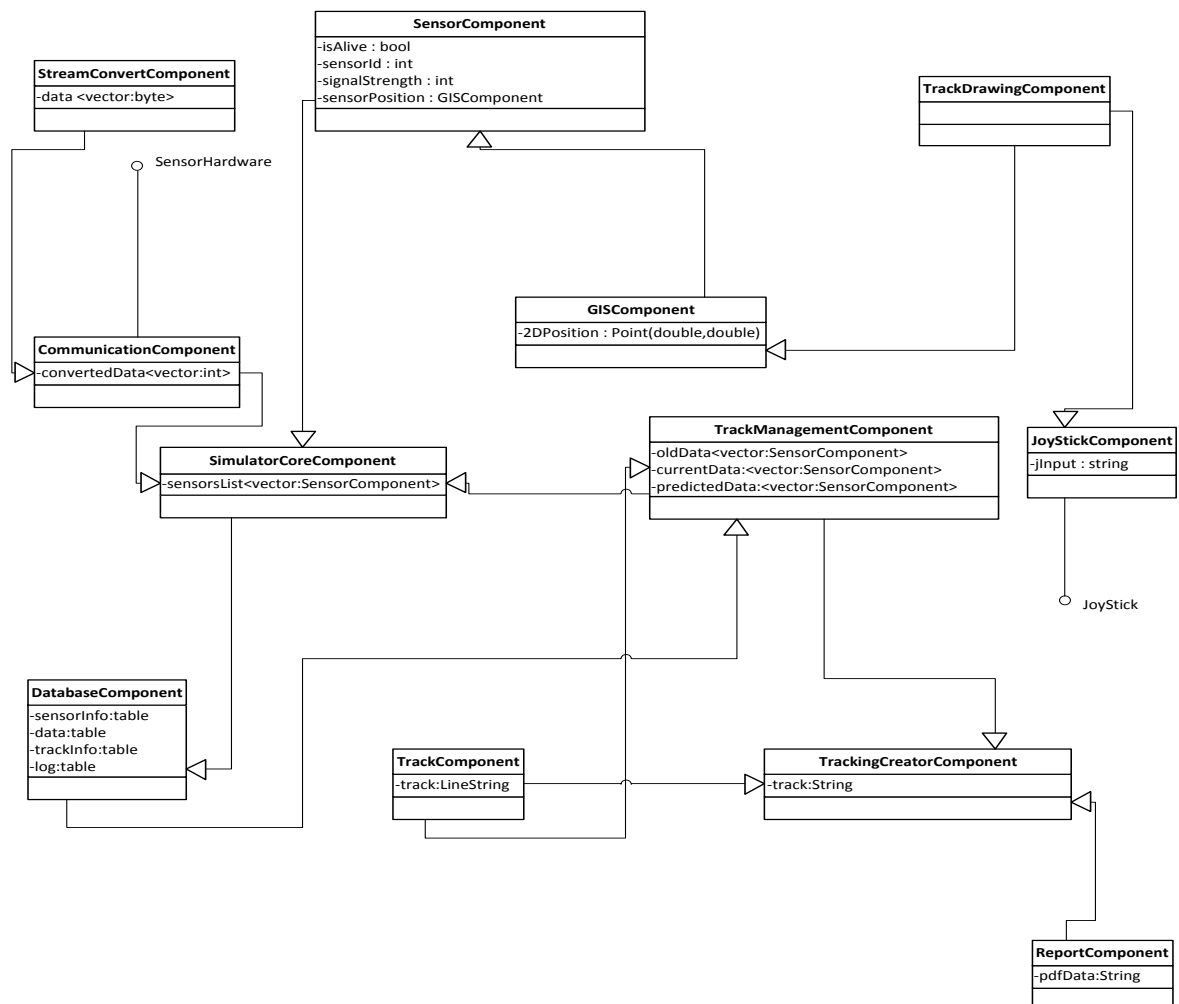
---

Data is collected over seismic sensors. Then these data are stored at database with record times. After storing data, Track management component performs main task and predicted data is calculated by Kalman filter in this class. Simulator core component gets predicted and real data from sensor component and track management component then simulate target tracking. User also can print out data in pdf format. Data is acquired from tracking creator component.

## 4.2 COMPLETE DATA MODEL AND RELATIONSHIPS

---

Relationships among data objects are described using an ERD- like form. During the software design process the relationships will be described in details.





**predictedData:** This attribute holds SensorComponent objects. These objects are hold as a Java vector.

This attributes are used in Kalman Filter. Main calculations in Kalman filter is done by these vectors.

#### 4.3.4 JoyStickComponent Class

---

**jInput:** Type of this attribute is string. To get input from joy stick, this attribute is used.

#### 4.3.5 ReportComponentClass

---

**pdfData:** This attribute is in string type. After final data is composed, final result is settled to pdfData string.

#### 4.3.6 DataBaseComponent Class

---

**sensorInfo:** This is in table type. This means; there is data table in database which holds sensor is alive or not, sensorId, signal strength of sensor. So all information about sensors is hold in this table.

**dataTable:** This is in table type. There is a table which holds sensor identification numbers and corresponding to time at which sensor is get signal.

**trackInfo:** This is in table type. Old sensor data, current sensor data and predicted sensor data are stored in this data table.

**log:** This is in data table type. User logs are stored in this data table.

### 4.3.7 SimulatorCoreComponent Class

---

**sensorList:** this holds vector of sensorComponent so it is in type <vector: sensorComponent>.

### 4.3.8 StreamConvertComponent Class

---

**convertedData:** This is in <vector: int> type. After manipulating data and converting data from byte vector to integer vector, this vector holds this converted vector.

### 4.3.9 CommunicationComponent Class

---

**Track:** Type of this attribute is lineString.

### 4.3.10 TrackComponent Class

---

**Track:** Type of this attribute is string.

---

## 5 BEHAVIORAL MODEL AND DESCRIPTION

---

In this section, we will describe the major events that occur in our system and describe how the system behaves on an event represented by *state machines*. State machines take each step based on when events occur in the environment of the behavior being performed.

### 5.1 DESCRIPTION FOR SOFTWARE BEHAVIOUR

---

#### 5.1.1 SIMULATION

---

The project runs over the simulation component. Simulation component shows the all required information containing sensors information, route of the moving object, and visualization. All required signal are stored in database and simulation component interpret those information with other components (Track Management component, Communication component) it can save results into database (updating the information), and report the results.

The first state in the project is the start state in simulation component and it invokes mode selection. There are seven modes namely Add Sensor, Show Sensors, Delete Sensors, Disable Sensors, Display Sensors, Display Sensor Information, Draw Track and Report Result as PDF. If Add Sensor selected, then the sensor location information, is sensor ID stored in the database, also after addition to database, the sensor will be displayed on real 3D map with real positions. If Delete Sensor is selected, the required sensor ID will be gathered or the list of the all sensors will be displayed and the user will select the desired sensor, and this sensor will be delayed from sensor table from database. Moreover, the sensor will not be displayed on the real map. If the user selects Disable Sensor, the *isLive* attribute of the sensor will be changed in False (updating the database), but the sensor will be still on the map with different color, no deletion will be done.

The Simulation will provide user to display all sensors, and routes of moving objects. If user selects Show Sensors, the simulation will display all sensors on the real map with real positions. If Display Sensor Information is selected, list of the sensors will displayed, and the system will provide the desired sensor, or it will accept the sensor ID, then the all information of that sensor from database will be displayed on the screen. If the user want to report the required information, the he/she need to select Report Results as PDF selection.

If the user wants to detect route of moving objects, he/she needs to select Draw Track. This mode will communicate with Target Management Component, it calculates the route of the object with an algorithm, then result will be displayed on the map.

## 5.1.2 DATABASE

---

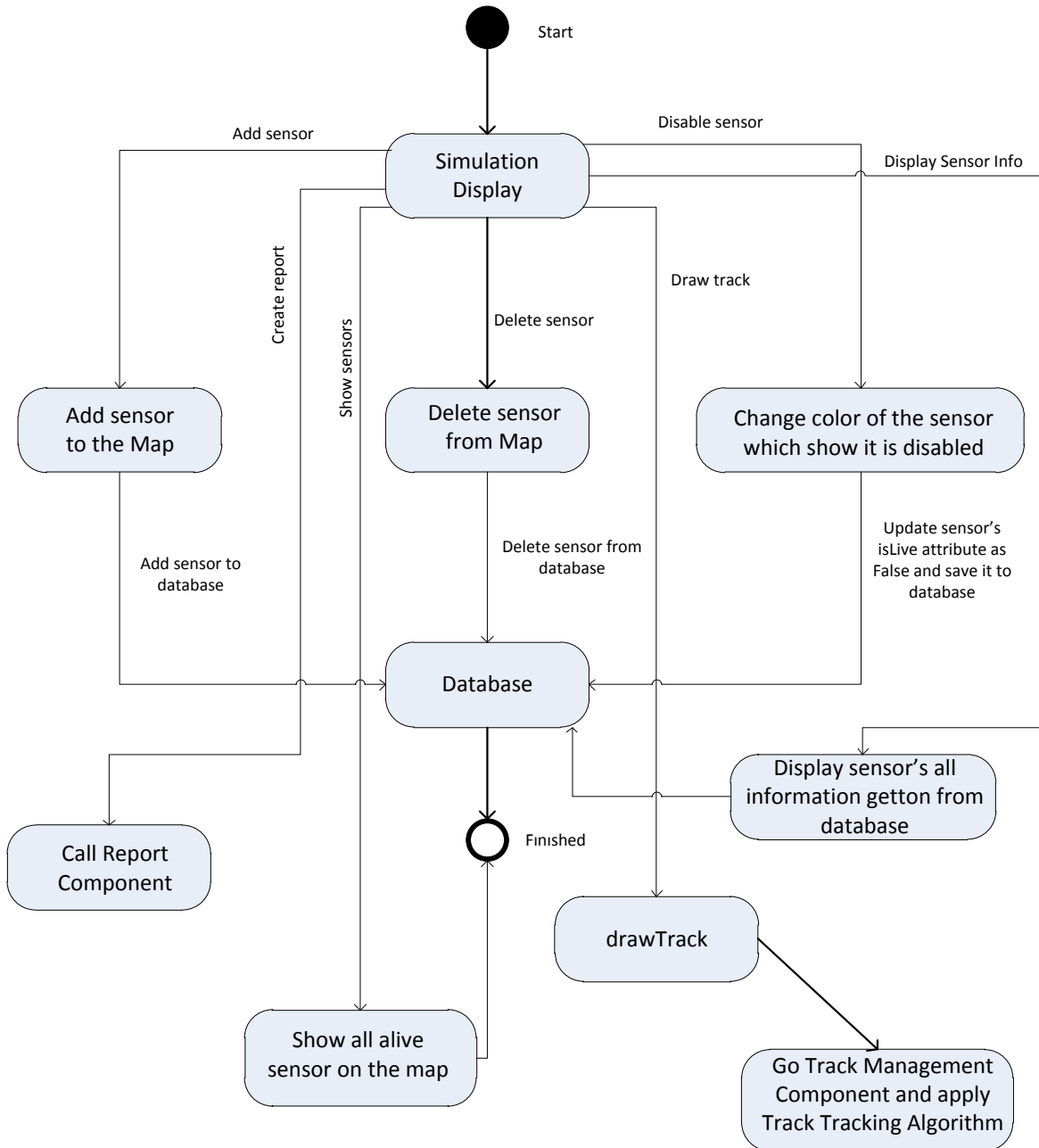
The users will enter to system as admin or simple user. They will access to the database with different rights. When the system is launched, the users will be prompted to the login at their status.

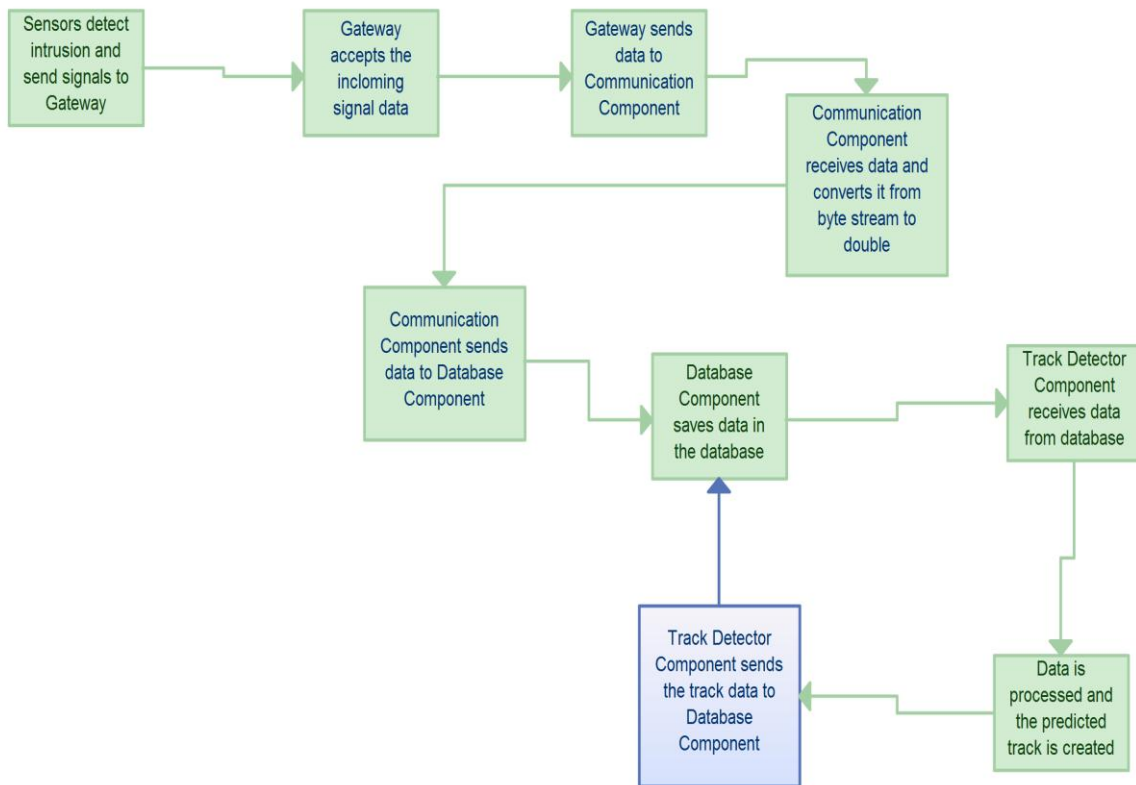
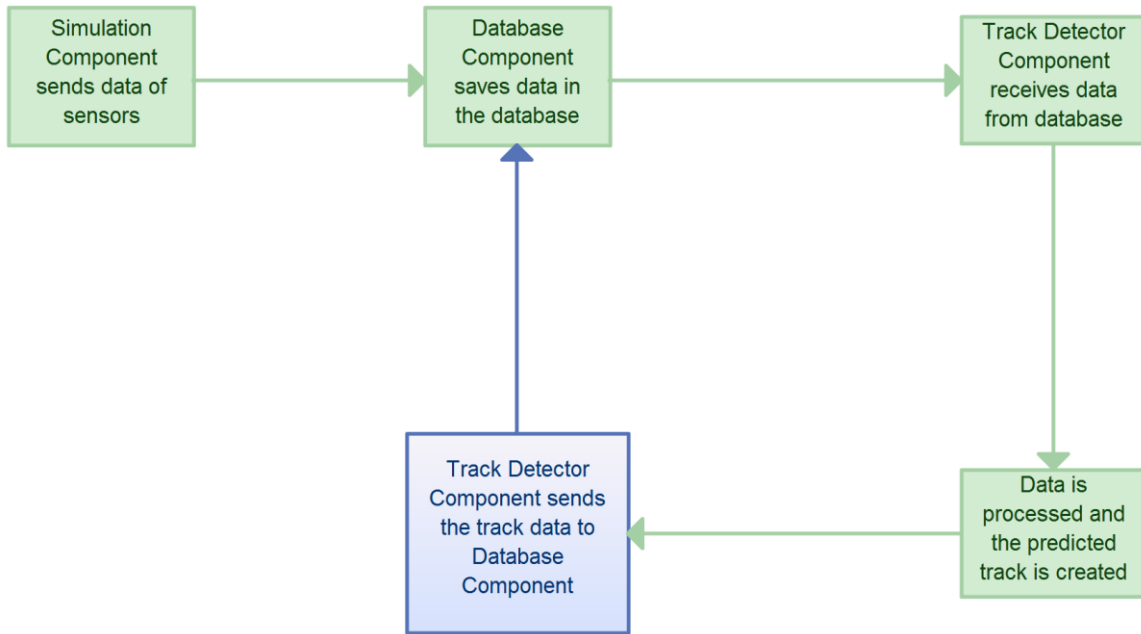
When the simulation component send the sensor related and track related data, the database component of the system will store them. The database will be holding Sensors table in which all the sensors, real hardware or imaginary sensors which are drawn on simulation screen, will be stored. The system will reach positions of the sensors and being active or disabled state of them from Sensors table. The database component will also hold the Data table, which will be used when the Track Detection component need to access former and current signal and activeness values of the sensors. Those data will be supplied to Track Detection component via database component. Track Detection Component will calculate the predicted track according to the supplied data to it.

Finally, the database will be holding the tracks in Tracks table in which origins and destination will be hold in successive steps to make logic about them. The input tracks will be hold as well. When the simulation component needs to compare the predicted track with input track, this data will be supplied to it by database component.

## 5.2 STATE TRANSITION DIAGRAMS

SIMULATION TRANSITION DIAGRAM





---

## 6 PLANNING

---

### 6.1 TEAM STRUCTURE

---

**Mustafa Mızrak - Researcher, Software Developer**

**Ali Fatih Gündüz - Researcher, Software Developer**

**Hüseyin Ünal - Researcher, Software Developer**

**Anıl Erkoç – Researcher, Software Developer**

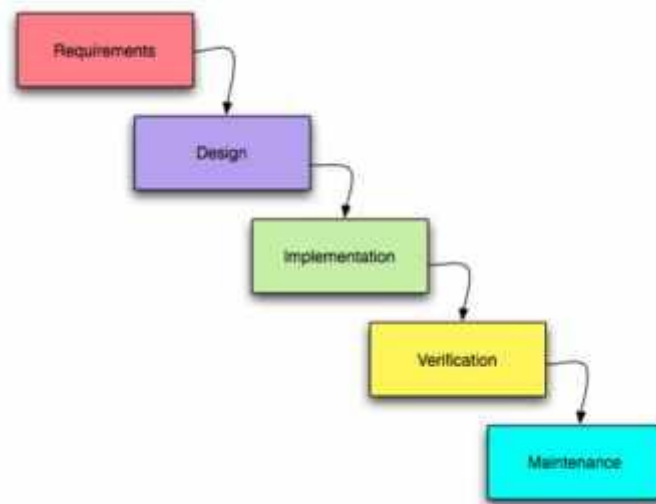
*spaceShuttle* team do not have a leader but our team has a coordinator, Mustafa, to synchronize and coordinate the work and make contact between the sponsor company and us. We have a collaborative decision mechanism. Communication between members is really important. We have meetings in strict hours and every time we are in touch. Since we make decisions together after discussing the issue, we do not need and a hierarchy in the team. For each part of the project, we assign everyone their roles in the relevant part. So, everyone knows his/her own role. If needed, we may exchange the roles in the team.



## 6.3 PROCESS MODEL

---

Because we first completed requirements specification and then plan to proceed to design and this design should be a plan for implementing the requirements given, we have chosen the Waterfall Process Model for our project.



---

## 7 CONCLUSION

---

This report is prepared to show Target Tracking by Using Seismic Sensor project's requirement details in terms of several aspects. Firstly, a brief description of target tracking and seismic sensors are introduced and the problem is defined. Then a marketing and technology research is carried out and the results are established. And at the body part, the requirement details of the project are described and Behavioral Model and Description is introduced. As a last of work, project's planning is presented. Thus, this report focused on the aspects which are seemed to be important.