Middle East Technical University
Computer Engineering

CENG 491 – Computer Engineering Design – I
Fall – 2011

Software Requirements Specification

BEE - TECH

1. Fatih SEMİZ_1752476
2. Güner ORHAN_1631050
3. Çağlar SEYLAN_1631126
4. Tuğba DEMİR_1630722

CONTENTS

1. Introduction

1.1. Problem Definition

Nowadays, the protection of the secret information is vital. The necessity of the intruder detection system is to abandon illegal entrance of the intruder into the specified area which is forbidden due to the confidential operations on military service. Our system will cope with the problem of intrusion and the trajectory of the motion of the intruder when it enters the illegal area.

1.2. Purpose

The purpose of the document is to establish a communication backbone between the sponsor, namely ASELSAN, and the developers, namely us. With SRS, possible misunderstandings about the product will be minimized and the project will be well defined. As the scope of the product, the product will be an intruder detection system that is well suited for military purposes.

1.3. Scope

The name of the project is “Intruder Detection System”. In this project we are going to detect intruders with the help of seismic sensors. With this project we are going distinguish a wind or natural effect from human activity.

The task of the hardware is taking signals from outside and saving them. Task of the software is processing signals and decide whether it is an alarm or not. Taking reports of alarming actions into a Doc or Pdf file will also be available.

1.4. User and Literature Survey

1.4.1. Literature Research

There are lots of ongoing projects on intruder detection systems with seismic sensors. The main purpose of these systems is to protect the border line from unauthorized entrance. There are many intrusion detection algorithms to achieve this goal. The main features that these algorithms differ from each other are;

- Detection range
- False alarm rate
- Power consumption
- Noise reduction algorithms
- Classifying the intruder

The main disadvantage of these seismic sensors is that they can detect earth movements or wind as an intruder. Therefore, algorithms must be efficient. Some of the algorithms that companies use are;

- Looking for the regular cadence of a typical human gait [2].
- Measuring the statistical distribution and detecting the extreme deviations from mean (Kurtosis) [2]

According these algorithms, noise and the detection range are inversely proportional. In the real world, it is assumed that the noise level is approximately medium, which means that intrusion detection range is approximately 20 meters. Applying Kurtosis in a high noise area with a specific threshold value can result to the misdetection of the intruder, because the noise level of the area is not always the same [4].

Another approach can be pursued. This approach is using strings of geophones for intrusion detection, and summation of the seismic signals. This contributes to the detection range. Even though all of the geophones have separate noises, only one geophone which is the closest to the walking person provides the main intruder's signal [4].

The most leading company in this area is Creare Inc. They have a very strong noise reduction algorithm to decrease the false alarm rate. Moreover, they can detect the intruder with approximately 60% correctness over 350 feet (105 m) distance [2]. With the adaptive noise cancellation (ANC) algorithm, they can detect the footsteps over 200 feet. They use off-the-shelf geophones (single-axis, 4.5 Hz geophones) [2]. The geophone is a machine that converts the ground movements to voltage [1].

Our intruder detection system will have more detection range than ASELSAN's current detection system algorithm. We can achieve this enhancement by creating powerful noise reduction algorithm, so that, not only the detection range will be enhanced but also false alarm rate will decrease. Other important enhancement that we will handle is reducing the power consumption. This enhancement is very crucial due to the difficulties on changing the batteries of the system. Another work is to create a simulation world to control our algorithms. Since finding an appropriate place to control our system is hard, creating a simulation is an inevitable part of our project. The last enhancement will be the classification of the detected object. The importance of the classification is that no one wants to give an alarm when a bird or any other animal enters the forbidden area.

This system is for border security, so the military service will use the system.

At the end of the year, we will demonstrate fully developed intruder detection system with noise reduction algorithm on both simulation world and video which we would have recorded while testing our system under different noisy conditions.

1.5. Definition and Abbreviations

SRS: Software Requirement Specifications

GUI: Graphical User Interface

1.6. References

[1] *Wikipedia, The Free Encyclopedia*. (2009, December). Retrieved October 19, 2011, from Wikipedia: http://en.wikipedia.org/wiki/Geophone

[2] Audette, W. E., Kynor, D. B., Wilbur, J. C., Gagne, J. R., & Peck, L. (2009). *Improved Intruder Detection Using Seismic Sensors and Adaptive Noise Cancellation.*

[3] Clarke, B. (2007, July 23). Retrieved October 20, 2011, from PSR-1 Seismic Intrusion Detector: http://www.prc68.com/I/PSR1.shtml

[4] Pakhomov, A., Sicignano, A., Sandy, M., & Goldburt, T. (2003). *Seismic Footstep Signal Characterization.* NY.

[5] http://www.forumalev.net/harita-kadastro/198490-google-harita.html

[6] http://web.ics.purdue.edu/~braile/edumod/as1mag/as1mag3_files/image011.jpg

1.7. Overview

For now, we plan that the SRS will consist of two parts except the introduction part. In the first part, product perspective and product function of the system, user characteristics, constraints, assumptions and dependencies and apportioning of requirements will be specified. In the second part, external interfaces, functional and performance requirements of the system will be specified.

2. Overall Description

In this section we aim to provide some information about general factors that affect the product and its requirements. We will firstly start with product perspective where we will describe any dependencies of the product. We will also describe the overall structure of the software in this part. Then we will provide an outline for functional requirements. Some use case diagrams will be provided to make this part more understandable. Finally we will mention some constraints and assumptions about the whole system.

2.1. Product Perspective

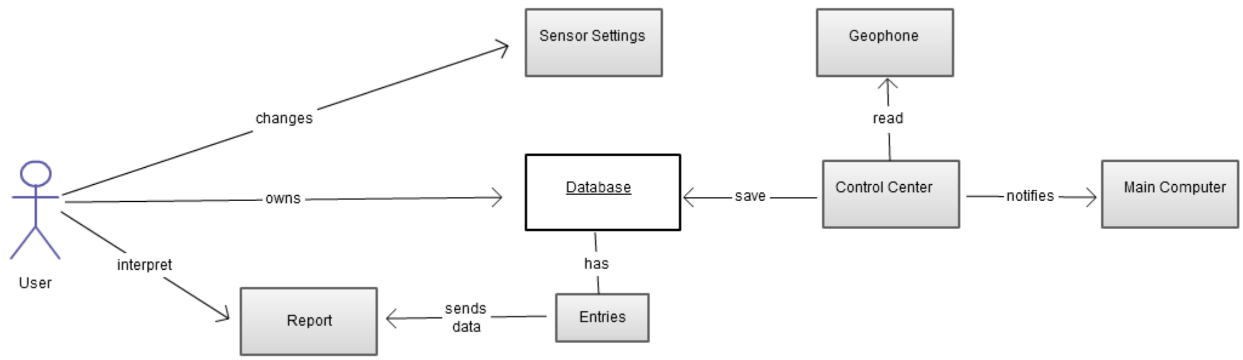


Diagram 1: Block diagram

The product can be used by everyone who wants preserve an area from the intruders. Actually this product provides an intruder detection system by producing alarms. It uses seismic sensors for providing signals. The sensors used will be in the ground. For this reason it is an outdoor application. After calculating the signals and deciding whether there is an intruder attempt or not the system will notify the control center to start the alarm system. The system will have a database to save the actions. After each operation as input the changes shall be written corresponding fields in the database. With this way the system shall provide statistical information. Furthermore system will have an interface where the user can observe the current state with the help of a map. In this map if there is an alarm, the sensor that detected the intrusion action will be visible from interface. Also statistical information will be visible from interface.

A block diagram that shows the general structure of the product is provided at Diagram -1. The functions showed in this diagram will be analyzed briefly at product functions part. Some use case diagrams about some of the major functions will also be provided.

The system will be an independent and totally self-contained system. Users reach to the functions via user interface which shall be connected to the database. The system shall deal with the database issues, like searching and retrieving data.

The system will be able to distinguish the signals coming from the natural environment (like wind, trees etc.) and the real intruders. To achieve this some specific algorithms will be used to determine a detection range. For instance signals exceeding some range will be taken as an intruder. The specific algorithms will be developed in time.

2.2. Product Functions

The system will take inputs from seismic sensors and the user. The system shall make some actions according to these inputs. We can categorize the functions into three parts:

2.2.1. Sending Alarm to the Control Center

This is one of the most critical parts of the system. This function is designed to send an alarm notification to control center in case of detection. The signals taken from the sensors will be tested for if it shows an intrusion action or not. If the signal is decided to be an alarm then a notification will be sent to the control center where the alarm system will be activated.

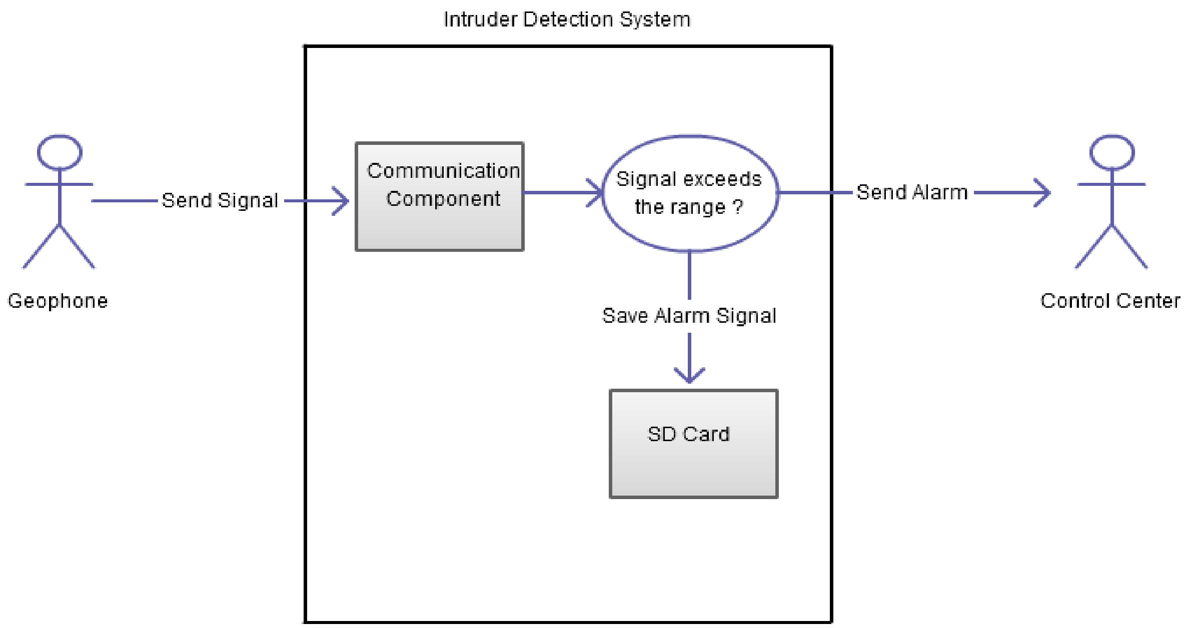


Diagram 2: Use case of sending alarm to the control center

In this use case diagram there are two actors and the system. One of the actors is the seismic sensor. This actor will take the role of sending inputs. The inputs will be signal values. Sensor actor will continuously send inputs.

The second actor will be the control center. The role of this actor is to take alarm notifications.

In between these two of them the intrusion detection system takes place. The role of the system here is to decide if the signal is an alarm and if it is notify the control center actor. Furthermore if the signal is decided to be detection then it is written to database.

2.2.2. Taking a Report Output from the Database

When there is a signal which is decided to be an alarm signal then the signal information will be saved to database in a format. After a while user can want some reports about a time period. This report shall contain time information as well as detection information. It will list the signals that are decided to be alarm signals. When this data is requested from the user the data will be written to a doc or pdf file according to user's request. This reporting process will be done by this functionality. To clarify this functionality better a use case diagram is provided below:

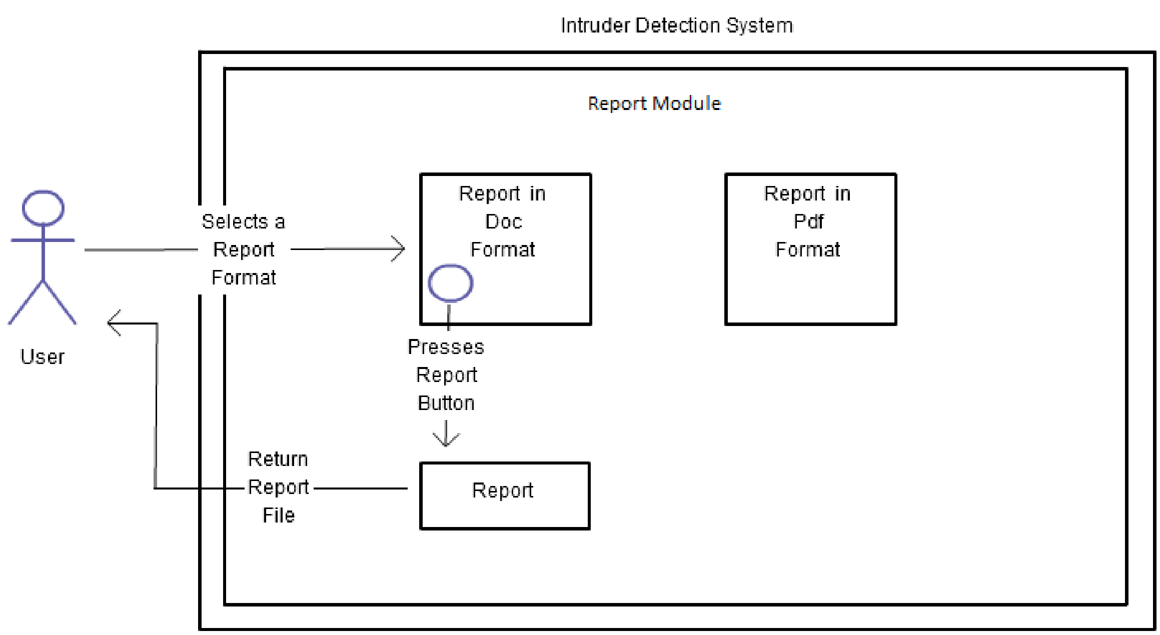


Diagram 3: Use case of taking a report output from the database

In this use case diagram there is only one actor which is the user and the system. The diagram shows the possibilities of the report to be written. (.doc or .pdf). Then the actor choses one of them and the report file will be provided to user.

2.2.3. Changing the Settings of Sensor

Sometimes users may want system to use less power than normal. For this kind of actions the system shall have a capability to change these settings. The sensors have sleep and normal modes and this capability provides control to user to change between these modes.

To illustrate this capability better the following use case diagram is provided.

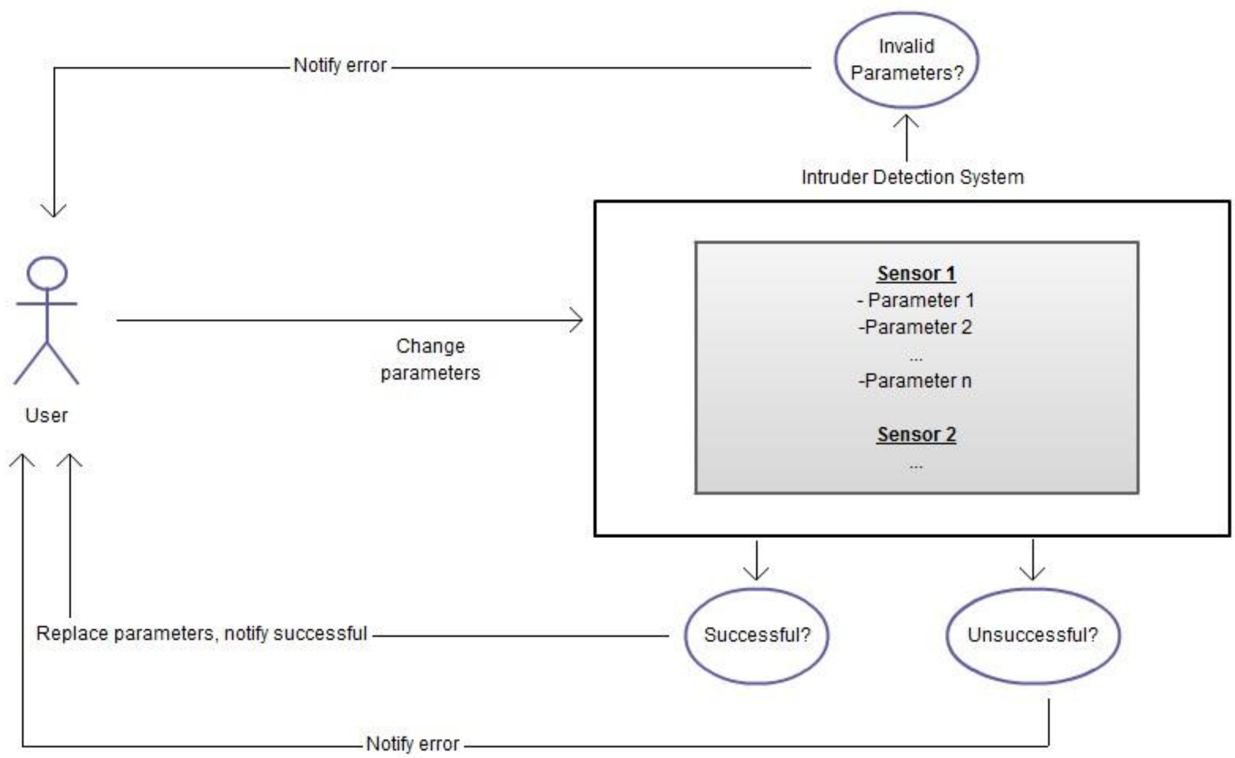


Diagram 4: Use case diagram of changing the settings of sensor

2.3. User Characteristic

This type of systems should be used by any type of users. Because of this system shall include a clear documentation, which should be understandable by elementary computer users.

2.4. Constraints of the System

2.4.1. Power Constraints

Developers of the system shall consider power consumption of the processes. And with the help of this they should provide some different modes that provides user to choose between more power consumptive ones and less power consumptive applications.

2.4.2. Safety Constraints

Developers shall consider safety of the information that this system will process.

3. Specific Requirements

3.1. Interface Requirements

We will have two types of users:

- Testers: The person who checks data send by sensors. Actions of these type of users are limited
- Admin: The person who will also be able to perform some additional operations such as registering new sensors and users.

3.1.1. User Interfaces

In this part, the graphical user interface of the application is explained.

3.1.1.1. Tester Interface

Login Page

To be able to access the system, user must enter his userName and password. If user did not register the system before, user will use **Register button** in this page. After Register button is pressed, a **form** will be opened in right side of the page. Form asks user to fill his name, surname, eMail address, username choice, password choice and retyping the password choice. After filling the form appropriately, user will use enter button to approve his register. After admin approval, he can access to system freely.

Operation Page

After successful login, users will be directed to this page. In this page there will be a map on which the places of geophones are shown. Looking at the map, user can have an idea about the states of geophones. In Figure 1, for illustration, arbitrary places are signed on map [5] as if there were geophones.

: Last data in less than 15 minutes

: Last data in less than 24 hours but more than 15 minutes

: No data in more than 24 hours

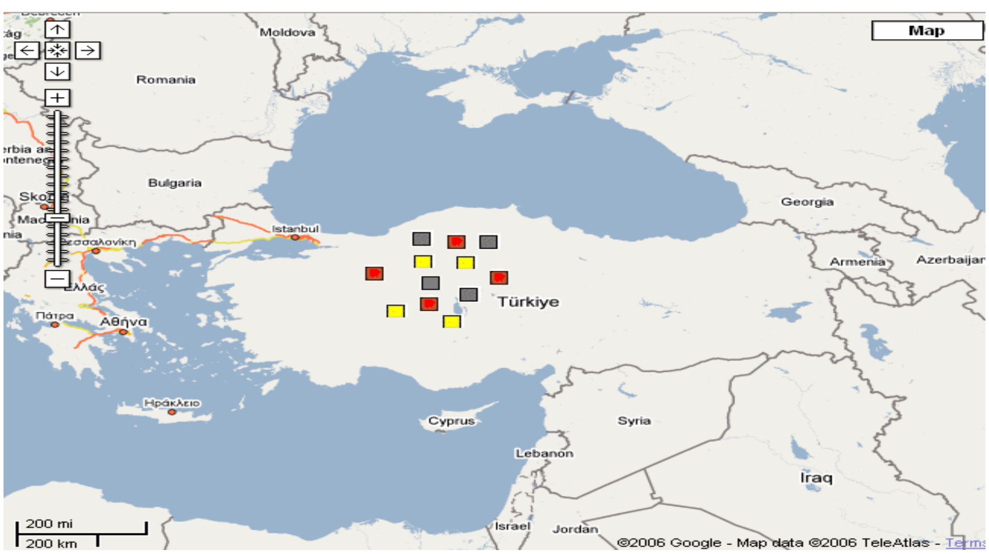


Figure 1: Geophone signed map of Turkey

After clicking on one of the colored squares, new page will be opened in front of user.

Geo_Specific Page

User will be able to get some specific information about sensor which is clicked lastly such as place information of sensor and last time data is get from that. This will be shown to user in table form like Table 1 below.

ID	Geo_1
Latitude	47.85
Longitude	68499
Elevation	538
Last Data Time	03/11/2011 – 7:40

Table 1: Sensor Properties

In this page, there will be also two buttons:

Time_Scale Button

After clicking this button, user can see the seismograms recorded on that day. Using the back and forward buttons below the chart, user can see the seismograms belonging to other days as well. Figure 2(taken from [6]) shows a 24 hour screen display of seismic data.

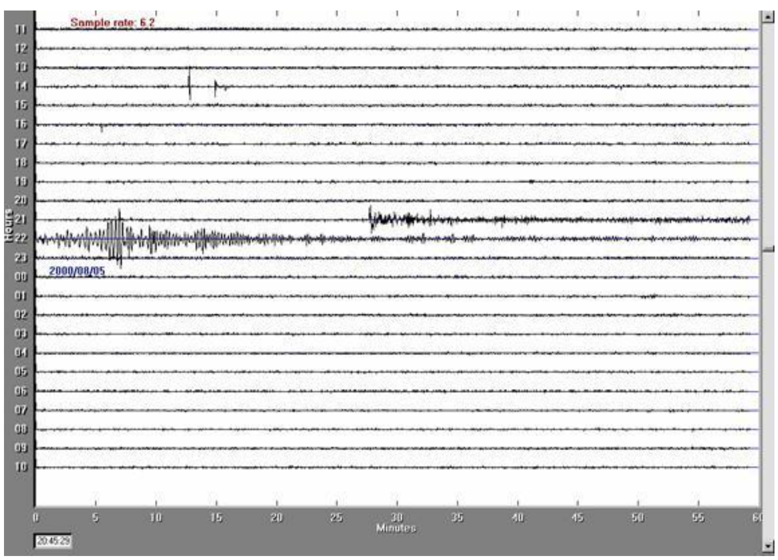
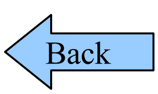
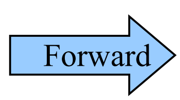


Figure 2: Seismograms





Report Button

Pressing this button, user will be able to print the records of a geophone in doc or pdf format.

3.1.1.2. Admin Interface

Admin interface is expanded version of testers. Admin also will have a login page like testers. There will be no Register button in this case. After successful login, admin will be directed to a page which has all features of Operation Page of other users. In addition, on the right side of that page, admin can see the notes about requests for membership like "2 New Requests for Approval". Clicking on the text, admin can see the information about owner of requests, and then he can approve or disapprove. In that page there will be also a button, **Register_New_Geophone**. Using this button, admin will be able to enter new geophone entries like coordinates and id. After this operation, new geophone will be saved to the database and will be put on maps. Admin can also determine who can see the geophones. For example, while geo_1 can be seen by tester1, it may not be seen by tester2.

3.2. Functional Requirements

In this section, we will explain seven fundamental function of the system, namely, sending alarm to the control center, reporting alarms recorded in the database, getting alarm records from a sensor, changing mode of a sensor, registering a new sensor to the control center, deleting a sensor from the control center, changing parameters of sensors. For each functionality, we will first give a description, then we will give exact flow of events, after that, we will give a use case diagram and finally we will list functional requirements.

3.2.1. Sending Alarm to the Control Center

3.2.1.1. Description

This is the most important function of the system. With this function, an alarm will be sent to the control center from a sensor so that users will be aware of an intruder. When the control center gets the alarm, a notification appears in the main GUI. Moreover, a timestamp is attached to the timeline, sensor ID and the time when the alarm is get is recorded in the time table. The point indicating the sensor on the map gets a red color.

3.2.1.2. Flow of Events

3.2.1.2.1. Normal Flow of Events

1. Geophone sends the signals to the processor.
2. The processor processes the signal and determines the cause is an intruder.
3. The processor saves the alarm signal to its SD card.
4. The control center is connected and the processor triggers the communication board to send an alarm signal.
5. The communication board sends the alarm signal to the control center.
6. The alarm is attached to the timeline, indicated on the alarm table, indicated on the map along with the sensor ID from which it comes.
7. The alarm is recorded on the database.

3.2.1.2.2. Alternative Flow of Events

Alternative Flow – 1

2. The processor processes the signal and determines the cause is not an intruder.
3. The processor ignores the signal.

Alternative Flow – 2

4. The control center is not connected.
5. The processor does not trigger the communication board.

3.2.1.3. Diagram

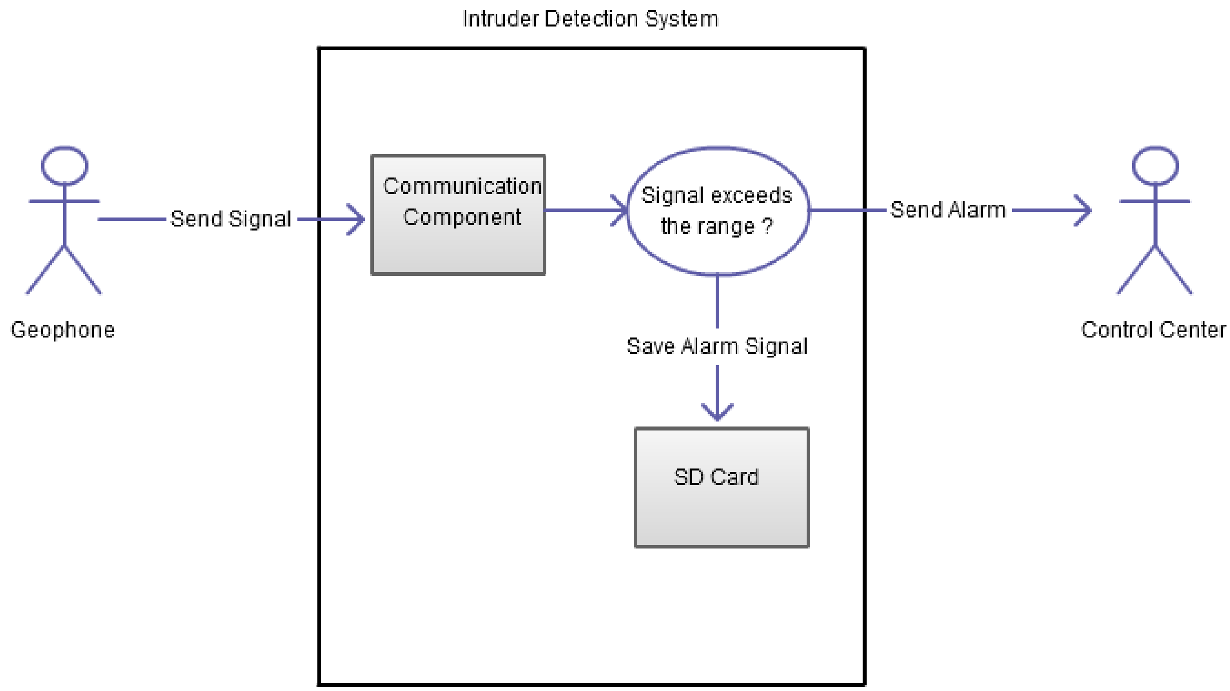


Diagram 5: Use case diagram of sending alarm to the control center

3.2.1.4. Functional Requirements

1. The system shall check whether the cause of the signal is an intruder or not.

2. The system shall check whether the control center is connected or not.

3.2.2. Reporting Alarms Recorded in the Database

3.2.2.1. Description

With this functionality, users will be able to get the saved alarms in the database in a report format. The format will be either .doc or .pdf. When the user wants to get a document of the alarms in the database, he/she clicks on "Report" button and indicates his/her preference as .doc or as .pdf format. The document is created by the system.

3.2.2.2. Flow of Events

3.2.2.2.1. Normal Flow of Events

1. User clicks on “Report” button in the GUI.
2. The user is redirected to a page where each of .doc and .pdf formats has an associated button.
3. The user makes the choice and clicks on “Report” button.
4. The GUI triggers the analysis component and it sends a query to the database to get saved alarms.
5. Analysis component sends the alarms to the reporting component.
6. Reporting component writes the alarms to .pdf or .doc files depending on the request of user.

3.2.2.2.2. Alternative Flow of Events

Alternative Flow – 1

3. The user clicks on “Report” button without making any choice.
4. The user gets a notification indicating he/she has to make a choice before clicking on “Report” button.

Alternative Flow – 2

6. Alarms cannot be written to the files.
7. The user gets a notification message that the files cannot be written with indicating the reason.

3.2.2.3. Diagram

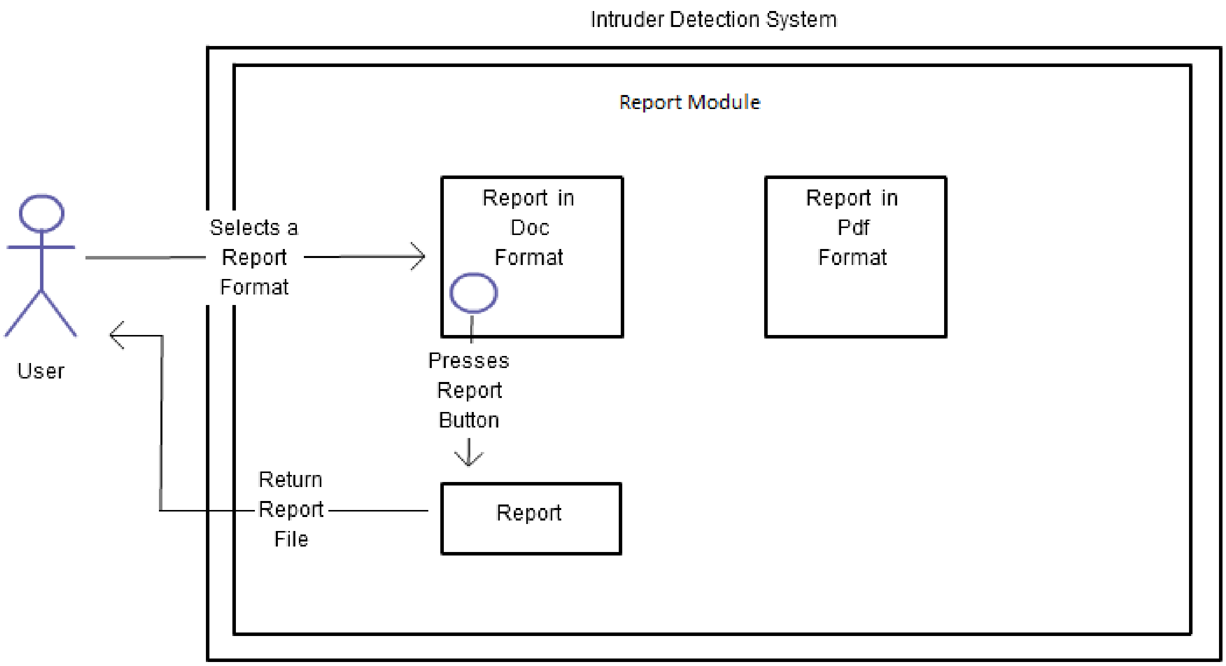


Diagram 6: Use case diagram of reporting alarms recorded in the database

3.2.2.4. Functional Requirements

1. The system shall check whether the user made a choice before clicking on "Report" button.
2. The system shall check whether the alarms are written to the files successfully.

3.2.3. Getting Alarm Records from a Sensor

3.2.3.1. Description

With this functionality, users will be able to get the alarm records those are saved in SD cards of sensors and save them to the database of the control center. When a sensor detects an alarm, the first action it takes is to save the alarm to its SD card. If there is a communication at that time between the control center and the sensor, the sensor also sends the alarm to the control center. However, there is the possibility of no communication between the sensor and the control center at that time. This function comes to action at this point. When the user wants to get the alarms recorded in the sensor at the time when there is no communication, he/she uses this functionality. Note that if the alarms already sent are not recorded in the database again.

3.2.3.2. Flow of Events

3.2.3.2.1. Normal Flow of Events

1. User clicks on "Request Alarms" button in the GUI.
2. The user is redirected to a page where sensors available are listed.
3. The user chooses the sensors from which he/she wants alarms and clicks on "Request" button.
4. Signal that requests alarms is sent from the control center to the sensors.
5. The sensors send all the alarm records in their SD cards to the control center.
6. The records are saved into the database.

3.2.3.2.2. Alternative Flow of Events

Alternative Flow – 1

3. The user clicks on "Request" button without making a choice.
4. The user gets a notification indicating he/she has to make a choice before clicking on "Request" button.

3.2.3.3. Diagram

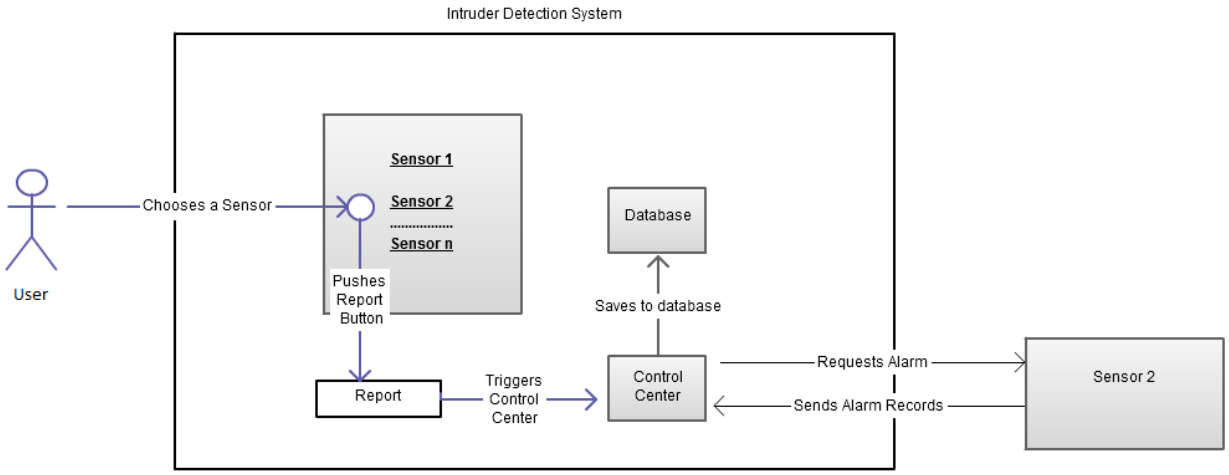


Diagram 7: Use case diagram of getting alarm records from a sensor

3.2.3.4. Functional Requirements

1. The system shall check whether the user made a choice before clicking on “Request” button.

3.2.4. Changing Mode of a Sensor

3.2.4.1. Description

With this functionality, the users will be able to change the mode (sleep mode or normal mode) of a sensor from the control center. In normal mode, signal getting frequency of a sensor is high. However, in sleep mode, signal getting frequency of a sensor is low so that is consumes low energy. Note that when the user goes to the mode changing page, previous modes of the sensors will be chosen.

3.2.4.2. Flow of Events

3.2.4.2.1. Normal Flow of Events

1. User clicks on “Change Mode” button on the GUI.
2. The user is redirected to a page where each available sensor is listed along with the modes they can work in.
3. The user selects one of the modes for each sensor.
4. The user clicks “Change Mode” button.
5. Signals to change the mode of sensors are sent to the sensors.
6. The sensors change their modes successfully and inform the control center.
7. The user informed that the sensor changed its mode successfully.

3.2.4.2.2. Alternative Flow of Events

Alternative Flow – 1

6. At least of the sensors cannot change their mode.
7. The sensors those cannot change their modes informs the control center.
8. The user gets a notification message indicating those sensors with reasons for each of them.

3.2.4.3. Diagram

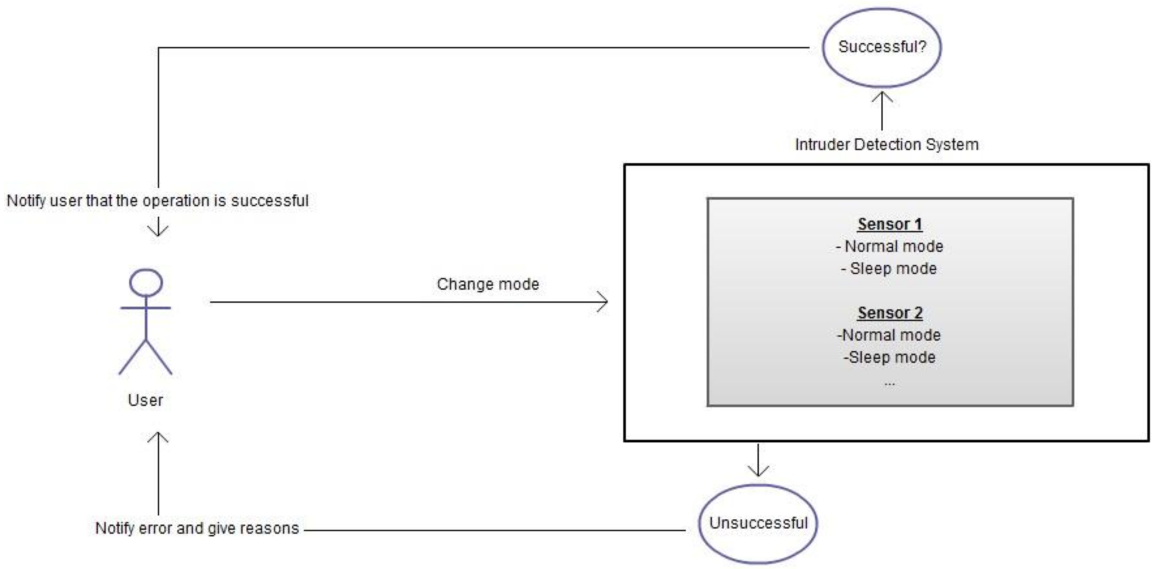


Diagram 8: Use case diagram of changing mode of a sensor

3.2.4.4. Functional Requirements

1. The system shall check whether the sensors changed their mode successfully or not.

3.2.5. Registering a New Sensor to the Control Center

3.2.5.1. Description

With this functionality, administrators of the system will be able to register a new sensor to the system. When a new sensor is registered to the system, a point indicating it is put on the map and a line for it is put into the timeline.

3.2.5.2. Flow of Events

3.2.5.2.1. Normal Flow of Events

1. Administrator opens the new sensor registration page from the main GUI.
2. Administrator fills the fields correctly and clicks on the “Register” button.
3. Administrator gets a message indicating that the sensor registration is successful.

3.2.5.2.2. Alternative Flow of Events

Alternative Flow – 1

2. Administrator clicks on the “Register” button without filling the fields completely or filling the fields in an incorrect manner.
3. Administrator gets a notification message indicating that the sensor cannot registered because of incomplete or incorrect filling of the fields.

Alternative Flow – 2

3. The registration is unsuccessful.
4. The administrator gets a notification message indicating that the registration in unsuccessful with the reason.

3.2.5.3. Diagram

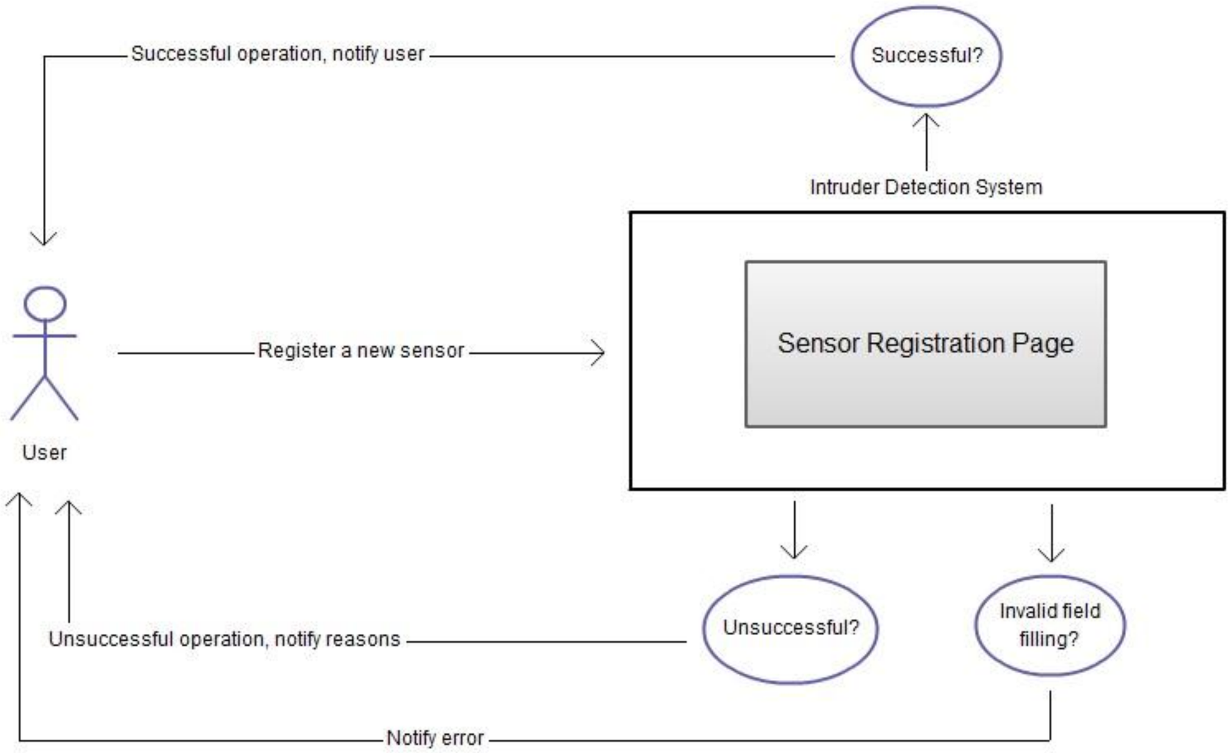


Diagram 9: Use case diagram of registering a new sensor to the control

3.2.5.4. Functional Requirements

1. The system shall check whether the administrator completely and validly filled the areas or not.
2. The system shall check whether the operation is successful or not.

3.2.6. Deleting a Sensor from the Control Center

3.2.6.1. Description

With this functionality, administrators of the system will be able to delete a sensor from the system. When a sensor is deleted from the system, its corresponding point in the map and corresponding line in the timeline is deleted. Moreover, if it has records in the database and in the alarm table, they will be also deleted.

3.2.6.2. Flow of Events

3.2.6.2.1. Normal Flow of Events

1. Administrator opens the sensor deletion page from the main GUI.
2. Available sensors in the system are listed associated with a selection box.
3. Administrator chooses the sensors to be deleted and clicks on "Delete" button.
4. The selected sensors are deleted from the system and the administrator in informed.

3.2.6.2.2. Alternative Flow of Events

Alternative Flow – 1

3. Administrator clicks on "Delete" button without choosing any sensor.
4. Administrator is notified that he/she has to click on "Delete" button after making a selection.

3.2.6.3. Diagram

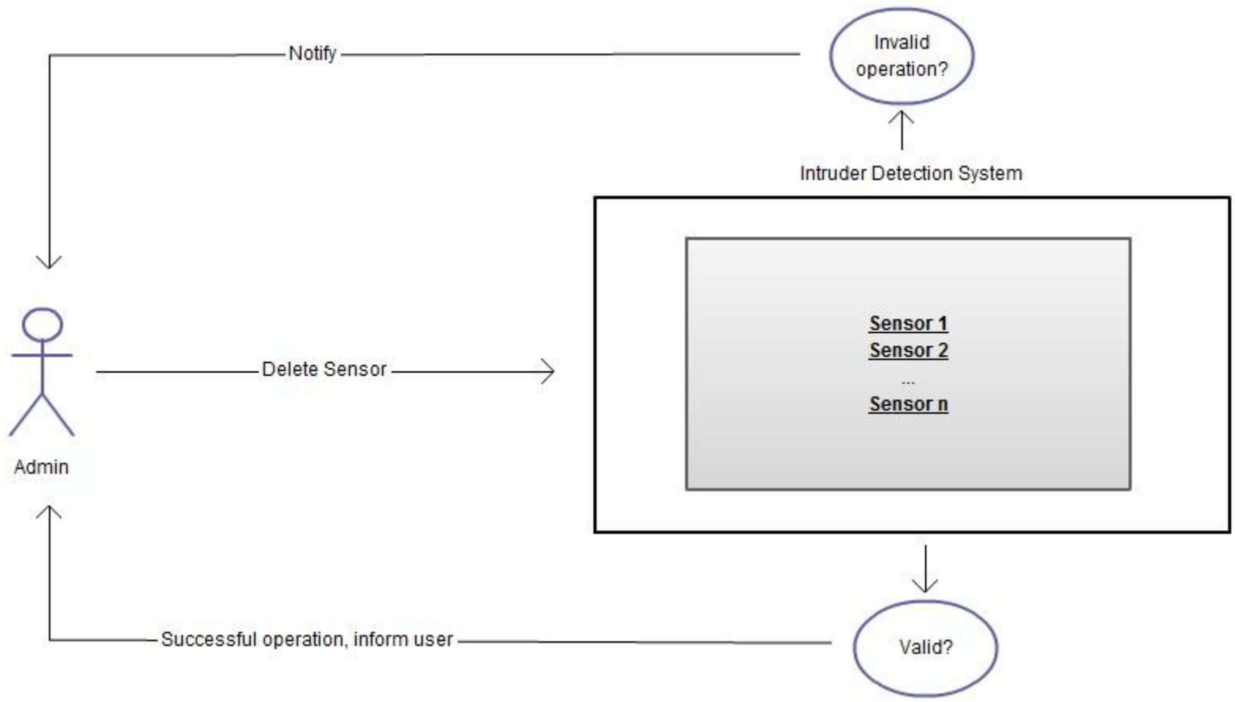


Diagram 10: Use case diagram of deleting a sensor from the control sensor

3.2.6.4. Functional Requirements

1. The system shall check whether the administrator selected at least one sensor or not.

3.2.7. Changing Parameters of Sensors

3.2.7.1. Description

With this functionality, the users will be able to change the parameters of the sensors and the parameters of the algorithms they use from the control center. More specifically, parameters of the sensor and algorithm it uses can be threshold values, sampling rate, frequency etc.

3.2.7.2. Flow of Events

3.2.7.2.1. Normal Flow of Events

1. User opens the page from where the parameters of the sensors are changed.
2. Available sensors in the system along with their parameters are listed.
3. User adjusts the parameters and clicks on “Change” button.
4. The sensors change their parameters successfully.
5. The user is informed that the sensors changed the parameters successfully.

3.2.7.2.2. Alternative Flow of Events

Alternative Flow – 1

3. User adjusts the parameters in an invalid manner and clicks on “Change” button.
4. The user in notified that he/she changed the parameters in an invalid manner.

Alternative Flow – 2

4. The sensors cannot change their parameters successfully.
5. Those sensors send a signal to the control center.
6. The user is notified that those sensors cannot change their parameters successfully with their reasons.

3.2.7.3. Diagram

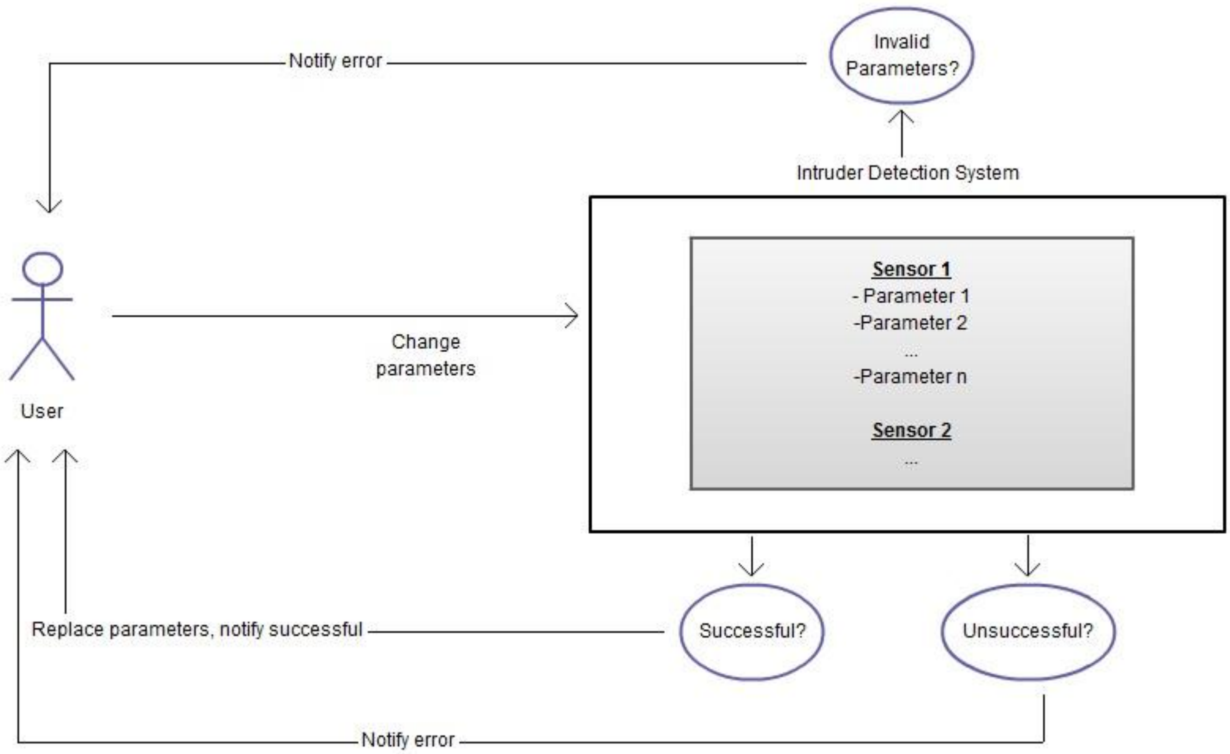


Diagram 11: Use case diagram of changing parameters of sensor

3.2.7.4. Functional Requirements

1. The system shall check whether the parameters that the user gives are valid or not.
2. The system shall check whether the parameters of the sensor are changed or not.

3.3. Non-Functional Requirements

3.3.1. Reliability

Network protocol we implement will be reliable in terms of data accuracy and time accuracy. Data accuracy checking is necessary for making sure that sensors not send wrong data. Programming the sensor nodes, time accuracy is achieved.

3.3.2. Security

User information security is the only constraint. Admins of the system will be able to decide whether giving right for controlling and getting data from a sensor to other users.

3.3.3. Maintainability

The batteries of sensors will run out after some time. Batteries should be replaced regularly.

3.3.4. Usability

The system should have user friendly GUI.

4. Data Model and Description

4.1. Data Description

In this section of the software requirements specification document, we will explain the signal storage of the intrusion according to the sensor. Basically, if no computer is connected to the seismic sensor, then the main storage hardware is SD Card which is the main component of each sensor. Otherwise; the signals are saved to the database system of the computer. Moreover, also new SD Card records are transferred to the database of the computer when computer is connected to the sensor.

4.2. Data Objects

There are four types of data objects which we will use

1. Authenticated User
2. Sensors
3. Signal Data
4. Alarms

Authenticated User:

Due to the fact that our system is for military service, User protection is important. Since, Authenticated user can change the properties of the main computer, sensor and database information. Moreover, all of the changes on database system or sensor properties which are operated by authenticated user must be stored to the system. We will handle this operation log information by storing it according to each authenticated user id. Primary key for database table is authenticated user id. The features of each record are;

- Name
- Surname
- MilitaryRank
- UserID
- Password
- LoginDate
- LogInfo

Sensors:

This table is necessary for storing sensor information and signals table. We can estimate the intruder's trajectory of motion by using the sensor information which is longitude and latitude of the sensor. This table is created on main computer. The primary key for this table is sensor id, and the features of the sensor records are as follows;

- SensorId
- Longitude
- Latitude

Signal Data:

This table is for detected intrusion signals. Signal information is also used on timeline graphical user interface. In addition, this signal data is stored to the SD Card on seismic sensor. The primary key for this table is signal id, and the features of the system are;

- SignalId
- Date
- Hour
- Message
- SignalValue

Alarms:

The table is for the signals which are detected as intrusion action. By looking at the alarm table information, it can be understood when the illegal entrance of the intruder is detected. The primary key for this table is timestamp. This table is created on main computer. The features of this table are as follows;

- SensorId
- Hour
- Date
- SignalId
- SignalValue
- Message

4.3. Relationship

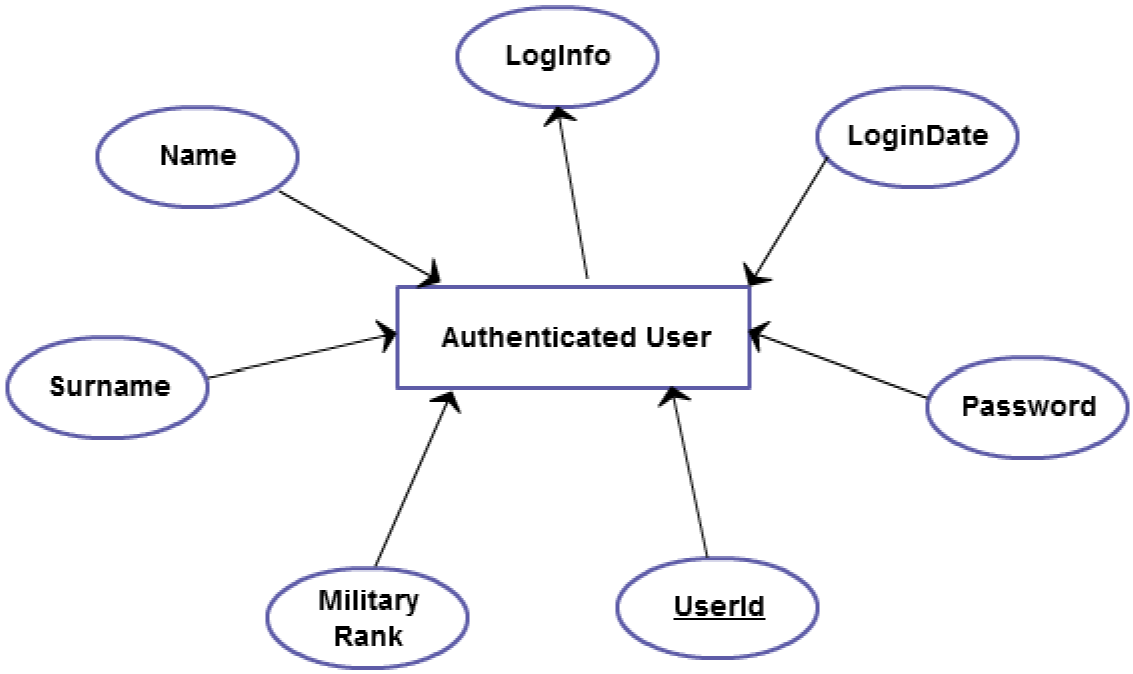


Diagram 12: E/R Diagram for Authenticated User

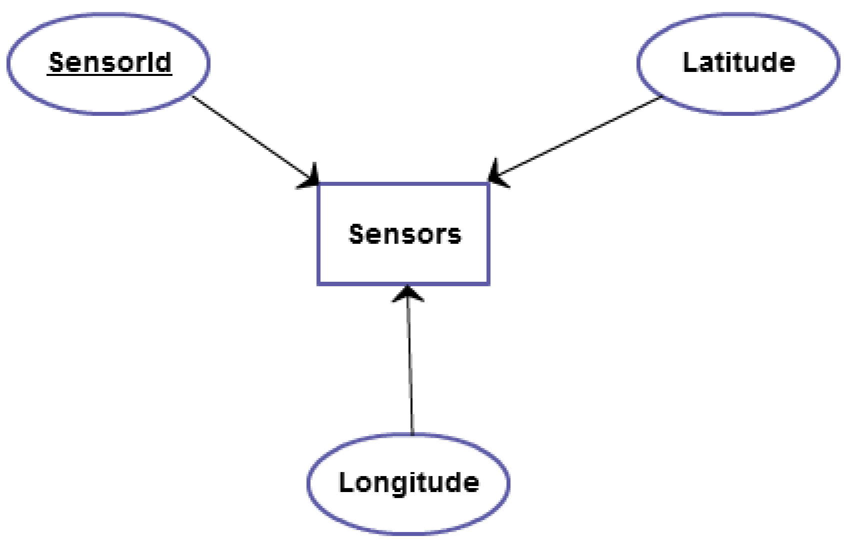


Diagram 13: E/R Diagram for Sensors

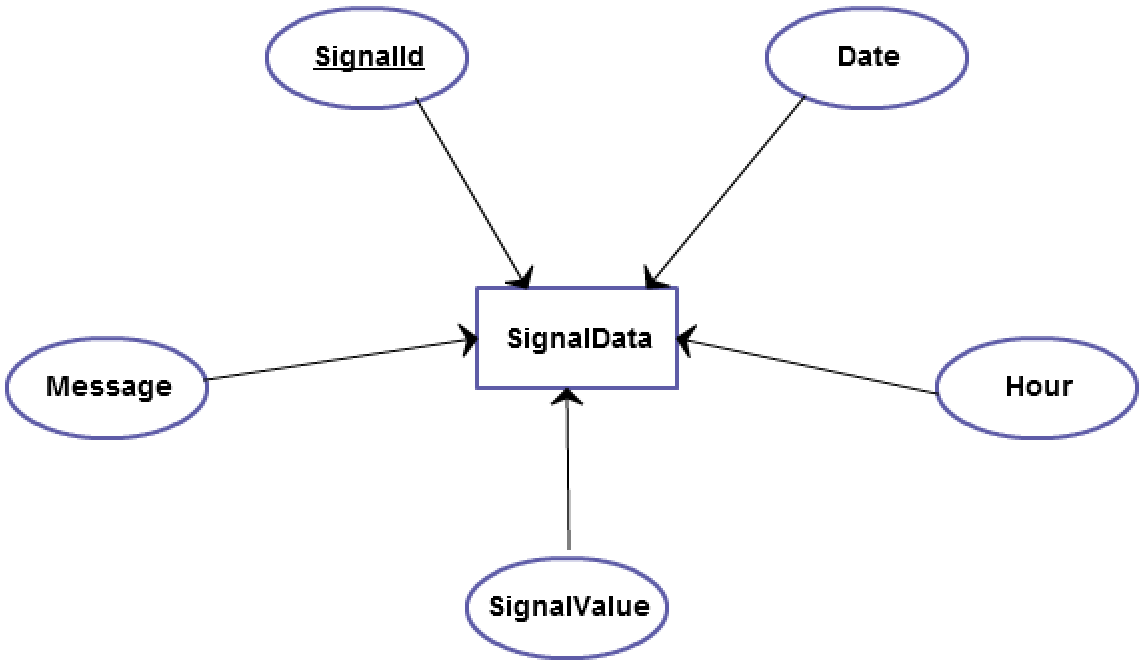


Diagram 14: E/R Diagram for Signal Data

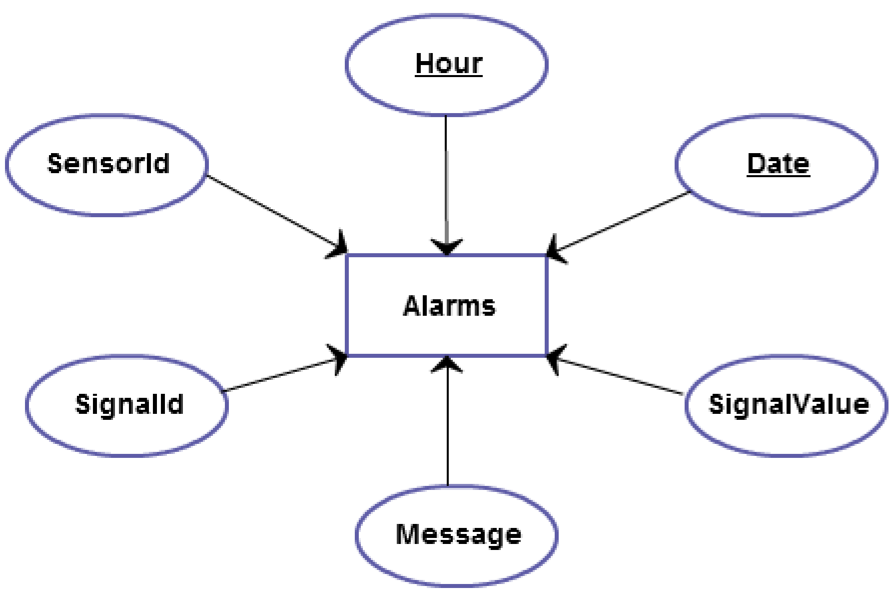


Diagram 15: E/R Diagram for Alarms

5. Behavioral Model and Description

Our intruder detection algorithm will work inside the Seismic sensor operating system which is Linux-based. The reason of that is the seismic sensor number can be changed according the dimensions of the protected area. Therefore, there can be more than fifty sensors which are very costly to compute the detections on one computer processor only. The only problem when we embed this software to the seismic sensor is the power consumption, so we are planning very effective algorithm that the use of power is very low. Now, we would like to mention about the software behavior of our system.

5.1. Description for Software Behavior

Previous State	Initial
Current State	Detection
Conditional – 1	if the system takes the shutdown signal, then the next state is “End” state Else it goes to Conditional – 2
Conditional – 2	If the system detects signal from sensor, then it goes to Conditional – 3 Else the next state is “Detection” state
Conditional – 3	If the signal is false alarm, then it goes to “Detection” state Else the next state is “Save SD Card” state

Previous State	Detection
Current State	Save SD Card
Conditional – 1	If PC is connected to the sensor, then the next state is “Read New Records From SD Card” state Else the next state is “Detection” state

Previous State	Save SD Card
Current State	Read New Records From SD Card
Next State	Communication

Previous State	Read New Records From SD Card
Current State	Communication
Next State	Central Control Unit
Prerequisite	The system takes the corresponding message from message set according to the detected signal.

Previous State	Communication
Current State	Central Control Unit
Conditional – 1	If report request is taken from the user, then the next state is "Take the Detection Records" state Else the next state is "Detection" state
Prerequisite	Central control unit saves the coming message to the database using database management unit

Previous State	Central Control Unit
Current State	Take the Detection Records
Next state	Draw Timeline Chart
Prerequisite	It takes the records of the signals from database by using database management unit.

Previous State	Take the Detection Records
Current State	Draw Timeline Chart
Next State	Detection

5.2. State Transition Diagram

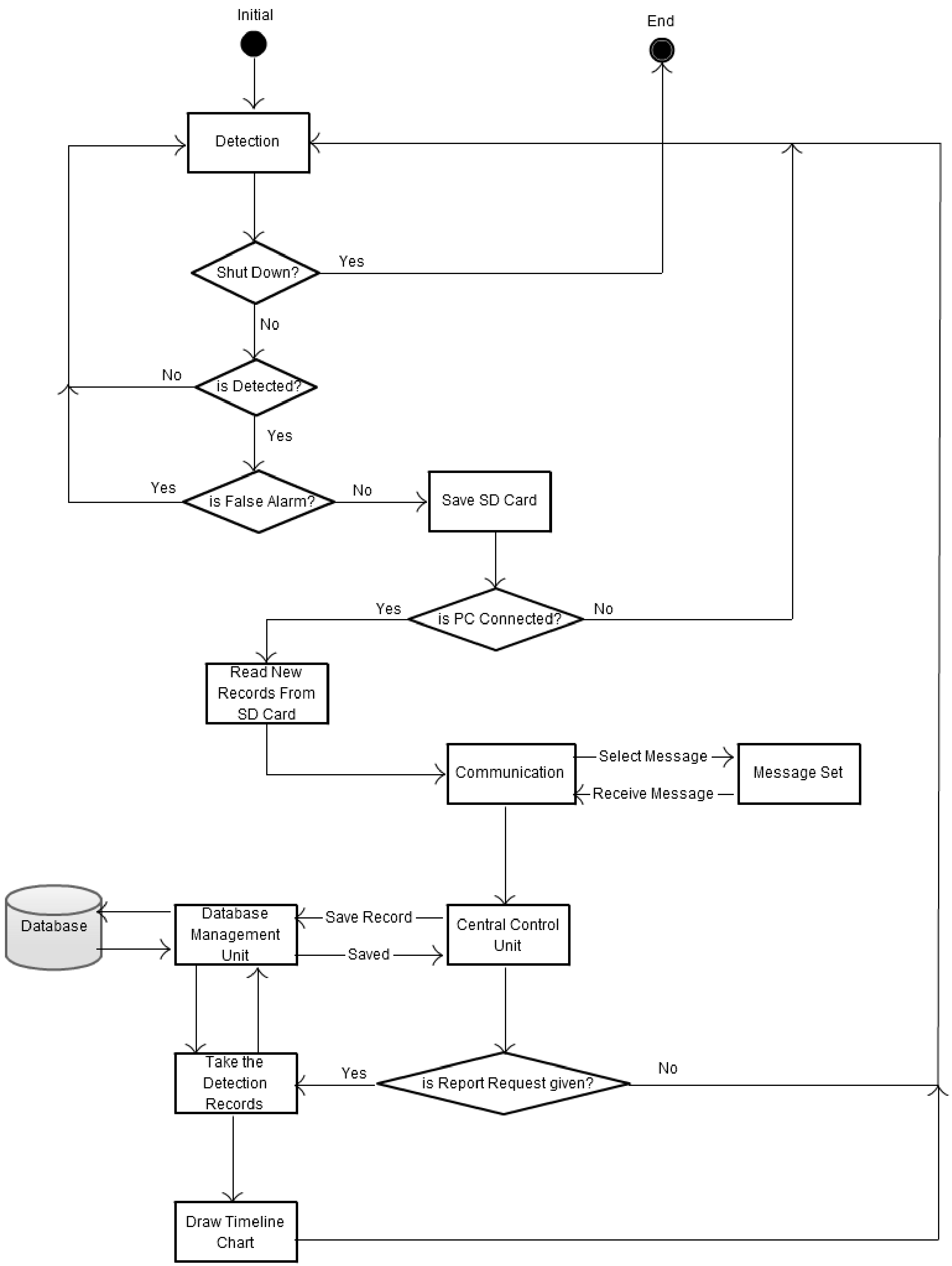


Diagram 16: State Transition Diagram of the system

6. Planning

6.1. Team Structure

The intruder detection project is mainly a research-development project. This project has three sub parts which are implementing a detection algorithm, implementing a simulator and implementing a control system for the system. For implementing a detection algorithm the whole of the group will search the literature to gain knowledge about this research area. However, in the other parts of the project there should be work sharing to proceed faster. Our work sharing will be as the following:

Çağlar Seylan: Software Architect & Head Developer
Fatih Semiz: Project Manager & Developer
Güner Orhan: Software Architect & Developer
Tuğba Demir: User Interface Designer & Developer

Everyone in the group will work on the software of the product but Çağlar and Güner will have more responsibility when it comes to decide software design of the product. The GUI part is assigned to Tuğba. Fatih is the project manager to synchronize and coordinate the work. He will be the one that make contact between the sponsor company and the team.

Apart from these we will give decisions about all other things as a team. Everyone should be aware of others work. For this reason we are having regular meetings every week. Because it is critical for the project to come up together and discuss what will be the best way to continue. Before making decisions we ask every team members idea and choose the idea which suits best for the project.

6.2. Estimation

The Gantt chart below shows the time estimations about the works in our project throughout the year.

	Task	Assigned To	Start	End	Dur	%
	Senior Design Project		1/10/2011	6/6/2012	250	
1	Topic Selection		1/10/11	10/10/11	6	
2	Literature Research		9/10/11	17/10/11	6	
3	Marketing Research		17/10/11	24/10/11	6	
4	Prepare Proposal		25/10/11	1/11/11	6	
5	Requirement Analysis		2/11/11	15/11/11	10	
6	Tests with sensor equipment		16/11/11	25/11/11	8	
7	Initial Design Report		15/11/11	6/12/11	16	
8	Implementation of given algorithm		2/11/11	30/11/11	21	
9	Search for new algorithms		28/11/11	10/12/11	10	
10	Implementing new algorithm		10/12/11	25/12/11	10	
11	Designing user interfaces		26/12/11	8/1/12	10	
12	Detailed design report		6/12/11	5/1/12	23	
13	Prototype Demo		9/1/12	21/1/12	10	
14	Implementing simulation		21/1/12	21/2/12	22	
15	Simulation Interface Design		21/2/12	10/3/12	14	
16	Debug		10/3/12	17/3/12	5	
17	Implementing Command- Control		17/3/12	17/4/12	22	
18	Debug		17/4/12	30/4/12	10	
19	Optimizing the performance and review		1/5/12	8/5/12	6	
20	Unit Testing		6/5/12	17/5/12	9	
21	Integration Testing		17/5/12	24/5/12	6	
22	Testing with real time objects		24/5/12	31/5/12	6	
23	Presentation		31/5/12	6/6/12	5	

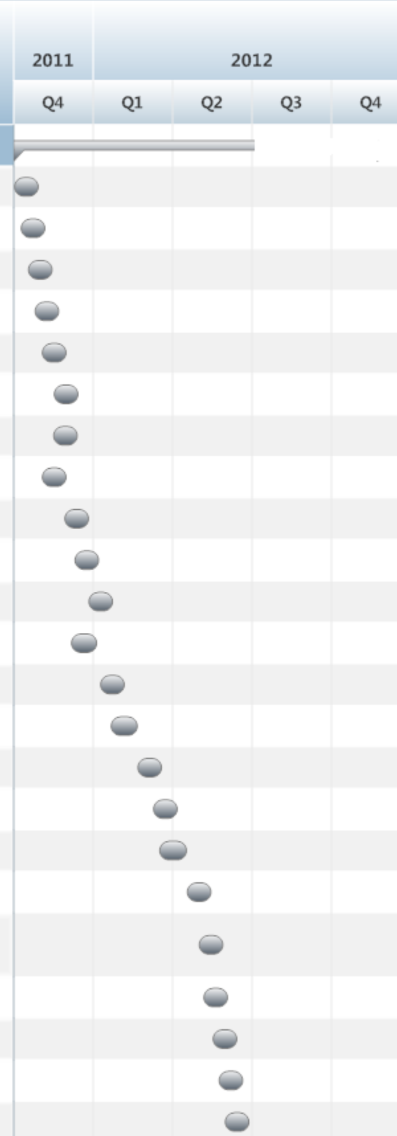


Figure 3: The Gantt chart

6.3. Process Model

The process model that fits best to our project is Waterfall Model. For this reason as process model in this project we are using Waterfall Model. An overview of the Waterfall Model can be seen at picture 1.

We first specified the requirements of the system. After that we will continue with the design of the product. Then we will implement the system that we described. Then testing and maintenance processes will be done on the product. We will apply a waterfall model with back flows. With this way we will be able to change our design when it is needed in the following phases.

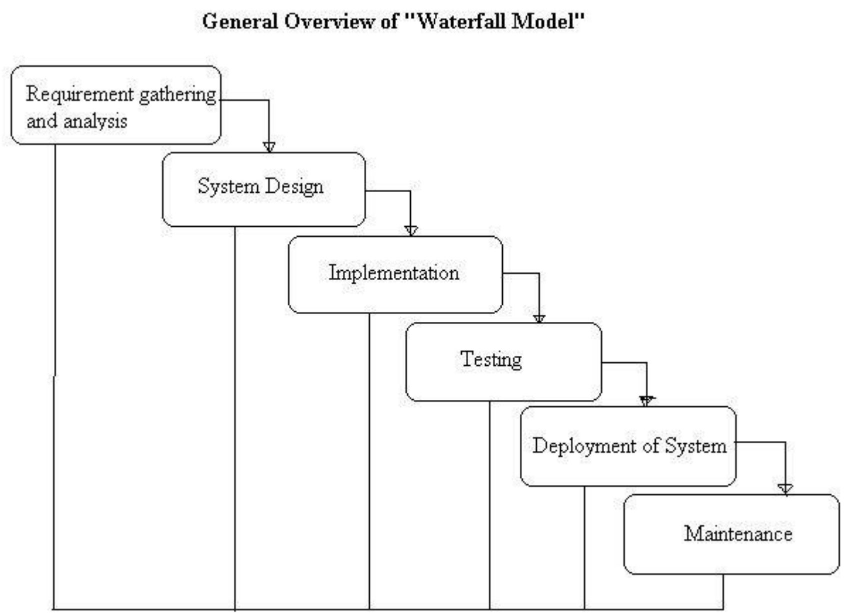


Figure 4: Waterfall Model [*http://www.buzzle.com*]

7. Conclusion

This document provides software requirement details of the Intruder Detection System project. In this document first an introduction which includes some background information is provided. Then overall structure of the system is described. After that detailed information about the system is given with the help of functional and non-functional requirements. Finally planning of the project is provided.

For making a healthy progress understanding the needs of the project is vital. This document describes the needs of the project in a very detailed manner. It gives information about methods needed in the project, limitations of the project and data dependencies of the project. All of this information will be used latter phases of the project as a guideline.

All of the information mentioned in this document is a result of the detailed analysis on user's needs.