Initial Design Report

CENG 491 – Computer Engineering Design I
Fall 2011

BEE-TECH

1. Fatih Semiz_1752476
2. Güner Orhan_1631050
3. Çağlar Seylan_1631126
4. Tuğba Demir_1630722

Sponsored by ASELSAN

20.12.2011

Contents

1. Introduction

Intruder detection system is the security-based project that is essential for detection of the intruder entering to the specified confidential area. The aim of this document is that showing the more detailed representations of the requirement specifications and giving the structured data design of the project in more detailed manner.

1.1. Problem Definition

Military service of any country has confidential information about country or any other equipped military soldiers. Therefore, this secret information must be protected from any other country or person. This can be achieved by protecting the area which surrounds these places. Another use area of this intruder detection system is to protect area for soldier gunned exercises, because, entrance of the any person can cause to injure the intruder.

Due to the security necessity of the specified area, intruder detection system will cope with the intrusion detection. The most important problem is that detecting the intrusion can be difficult due to the noise which results from the animal entrance, wind, or other climate conditions.

Another important problem that our system will handle is false alarm rates. Since, operators will control the alarm detected area, this kind of problem will cause waste of money and time.

Moreover, the detection range is a problem and also the constraint of the system. The working range of the system is important, because if we can increase it, military service can protect the area with less number of seismic sensors.

1.2. Purpose

We have stated the functional and non-functional requirements of the intruder detection system in Software Requirements Specification document. At this document, we will give the detailed system design properties of these requirements.

With the help of this document, reader can understand the system and the data architecture of the system. In addition, this document includes all necessary information for a programmer to develop an intrusion detection system.

1.3. Scope

The scope of document includes hardware and software requirements, data design, system architecture, tools, libraries, user interface design, and time planning of our project.

1.4. Overview

We can divide the document into nine parts, including introduction part as first. In the 2nd section, we will mention about the general details of the system by giving its goals, benefits and objectives. 3rd section generally includes the design issues which are dependencies, constraints, goals, and guidelines. In section 4, we will give the data design properties which are the properties of the storage of data into the database system, or SD card with data features. 5th part includes the detailed description of program architecture. In 6th section, user interface design and its screen images will be presented. In the 7th section, we will give the required tools and libraries to develop the system. In the 8th section, we will give the finalized Gantt chart diagram for 1st and 2nd terms. In the last section, we will conclude the initial design report.

1.5. Definitions, Acronyms and Abbreviations

IDR: Initial Design Report

IDE: Integrated Development Environment

GPS: Global Positioning System

GPRS: General Packet Radio Service

SD: Secure Digital

SRAM: Static Random Access Memory

1.6. References

[1] http://en.wikipedia.org/wiki/Flow-based_programming

[2] http://jpaulmorrison.com/fbp/

[3] http://c2.com/cgi/wiki?CouplingAndCohesion

2. System Overview

We use mainly Waspmote IDE application for writing and embedding the code to the Waspmote.

2.1. Waspmote IDE

Waspmote IDE is a well-organized development environment, programmed with Java, to embed the code to the waspmote board. It contains all required libraries to handle the operations of the sensors, GPS, GPRS, or any other equipped hardware that can be integrated on a board. Its libraries are created, effectively. All of the functions are understandable for a programmer.

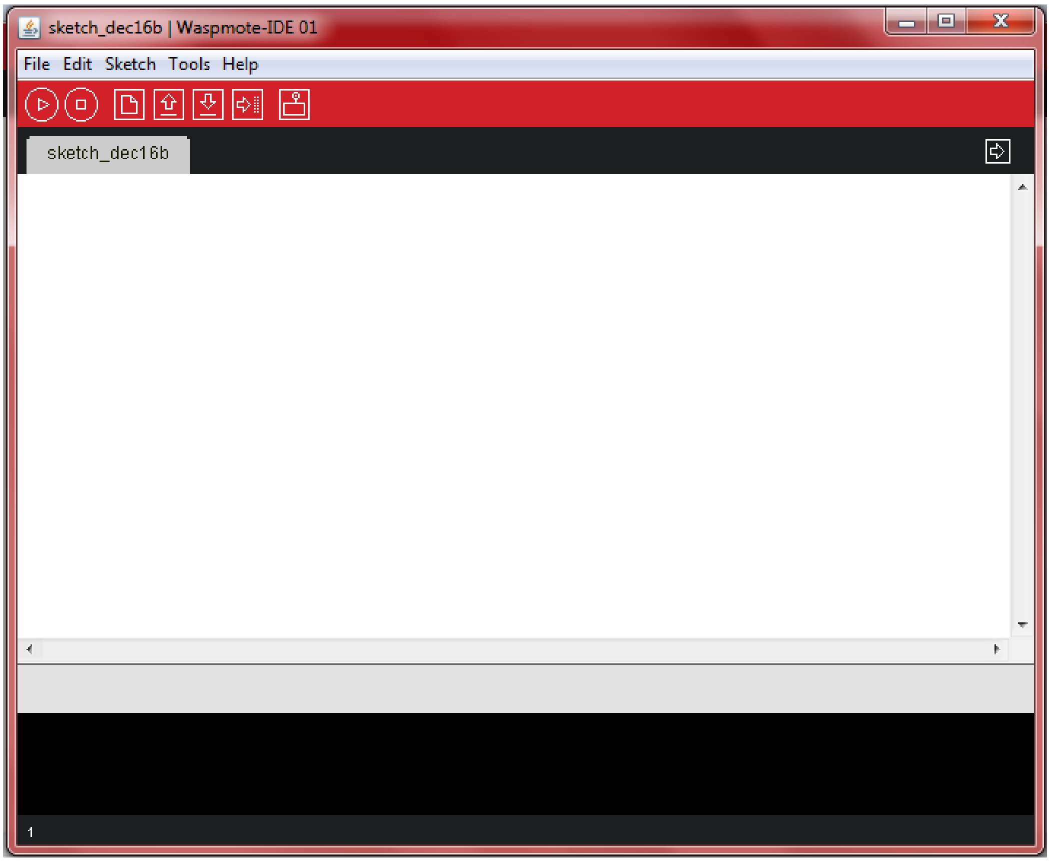


Figure 1. Waspmote IDE

It includes own grammar syntax written with Java ANTLR library.

3. Design Considerations

3.1. Design Assumptions, Dependencies and Constraints

- Intruder detection system code will be embedded to the waspmote board.
- Waspmote has ATmega1281 microprocessor with 8MHz frequency and 8 KB SRAM.
- The program must use less memory as much as possible due to the small size of memory on board.
- The microprocessor has low frequency so the time complexity of our system must be very low.
- The required database management system is PostgreSQL.
- The system has GPS to get the latitude, longitude, altitude, and the time from the satellite.

- When Waspmote is not connected to the PC, it will record the alarms to the SD card which can be integrating on it.
- Our project assumption is that each waspmote with seismic sensors will have own code inside and all seismic sensors will give an alarm independent from each other.

3.2. Design Goals and Guidelines

3.2.1. Performance

Due to the small memory size and low frequency of the microprocessor inside the board, our intruder detection algorithm must work fast. All of the calculations which decide the intrusion detected must be performed in short period of time.

3.2.2. Reliability

First of all, the most important issue that we will handle is reducing the false alarm rate meaning that giving an alarm when there is no intruder or not giving any alarm that there is an intruder. False alarm rate will decrease the reliability of our system. It also causes the waste of time for military service. We can think that our system has a zero tolerance to the errors.

4. Data Design

4.1. Data Description

This part consists of two subparts: description of data entities, where the major data entities of the system will be identified and described one by one and database of the system where the responsibilities of the databases will be discussed and details how to create it will be given.

4.1.1. Description of Data Entities

In this part, descriptions of the data entities in the information domain of the intruder detection system are given. There are 7 major data entities in the system: Alarm data entity, Geophone data entity, DatabaseRecords data entity, User data entity, Map data entity, AlarmTable data entity, Analysis data entity.

4.1.1.1. Alarm Data Entity

The Alarm data entity holds the information needed to describe an alarm.

Field Name	**Data Type**	**Description**
AlarmID	Long	ID number of the alarm
AlarmTime	Time	The time when the alarm is taken
AlarmDate	Date	The date when the alarm is taken
GeophoneID	Long	ID number of the geophone from which the alarm is get
AlarmPos	Position	Position at which the alarm is get

4.1.1.2. Geophone data entity

The Geophone data entity holds the information needed to describe a geophone.

Field Name	**Data Type**	**Description**
GeophoneID	Long	ID number of the geophone
GeophonePos	Position	Current position of the geophone
Status	Integer	Status of the geophone

4.1.1.3. DatabaseRecords data entity

The DatabaseRecord data entity holds the information needed to describe an entry of the database.

Field Name	**Data Type**	**Description**
Alarms	Array of Alarm	A list of alarms on the database

4.1.1.4. User data entity

The User data entity holds the information needed to describe a user.

Field Name	Data Type	Description
UserID	Long	ID of the user
Passwd	String	Password of the user
UserName	String	Name of the user
UserMap	Map	Map of the user
UserAlarmTable	AlarmTable	Alarm table of the user

4.1.1.5. Map data entity

The Map data entity holds the information needed to represent a map on the GUI.

Field Name	Data Type	Description
Geophones	Array of Geophone	A list of geophones on the map

4.1.1.6. AlarmTable data entity

The AlarmTable data entity holds the information needed to describe the alarm table on the GUI.

Field Name	Data Type	Description
Alarms	Array of Alarm	A list of alarms on the alarm table

4.1.1.7. Statistics data entity

The Statistics data entity holds the information needed to represent the statistics those will be computed by the analysis component.

Field Name	Data Type	Description
Records	DatabaseRecords	Records in the database will be printed as the statistics

4.1.2. Databases and Data Storage Items

4.1.2.1. SD Card

There will be a SD card in the sensor, more specifically, integrated with the Waspmote board. With this card, the sensor will be able to store the alarms. When user wants to read the alarms from the sensor, he/she will send a command requesting the alarms in the card and the alarms will be kept in the hard disk, namely, the storage item of control center. More specific information about SD cards is given in section 7.5.

4.1.2.2. Hard-disk of the Control Center

A common hard-disk will be the main information storage item of the system. It will hold the records of the past alarms. We keep past alarms because user may want to create a document showing the statistics about the alarms kept.

The database will be created with PostgreSQL, which is an open source, object-relational database system. More specific information about PostgreSQL can be found in section 7.3.

4.2. Data Dictionary

Name	Type	Refer to Section
Alarm	Data entity	4.1.1.1
AlarmTable	Data entity	4.1.1.6
Analysis	Module	5.2.7
Communication	Module	5.2.5
Database	Module	5.2.2
DatabaseRecords	Data entity	4.1.1.3
Geophone	Data entity	4.1.1.2
Map	Data entity	4.1.1.5
Reporting	Module	5.2.4
Sensor Managing	Module	5.2.3
Simulation Core	Module	5.2.1
Statistics	Data entity	4.1.1.7
Timeline	Module	5.2.6
User	Data entity	4.1.1.4

5. System Architecture

In this section a general description about the Intruder Detection System will be provided. First we will provide information about relationships between modules to understand behavior of the whole program better. Then we will analyze each of the modules' behavior individually by providing information about their processing details and interface descriptions. Finally to understand each module better, we will present dynamic behavior of the modules.

5.1. Architectural Design

We design our system in a way that each of the modules sees every other module as black boxes. The whole system organized with the help of data exchanges between these black boxes. This type of programming paradigm is called as flow based programming [1].

Our system has seven main modules that compose the overall system. These modules are namely;

- Simulation core module,
- Database module,
- Sensor managing module,
- Timeline module,
- Reporting module,
- Analysis module,
- And communication module.

Simulation Core Module: Simulation core module is the module which is placed at the center of the whole system. Every other module will be in contact with simulation core module. They will send their data to simulation core module. The only module that has direct access to database module is simulation core module. This module is created to make system more modular, with this way the situations like more than one request to reach database at same time period is avoided.

Sensor Managing Module: Another module is the sensor managing module. This module is responsible from the operations on the sensors. These operations are mainly divided into two sub operations namely changing sensor settings and activating or deactivating the sensors. By sensor settings we mean changing the mode of a sensor (sleep mode or normal mode).

Database Module: Database module is responsible from providing connection with database as the name implies. When simulation core module makes a request to the database, the module will store data to database or retrieve data from database, according to the type of the request.

Time Line module: Time line module is responsible from creating time tables and marking the intruder actions on these time tables. In some time periods the user will want to see this time table information. For this reason this information is send to simulator core module to save them to database. With this way when user wants to the time information this information will be retrieved form the database by the help of database module.

Reporting Module: From time to time the user may want to see some report information about the intruder action as a written document. To fulfill this kind of needs of the user reporting module will be used. This module is responsible from creating doc or pdf files that contains the recent intruder action information.

Analysis Module: Analysis module is the module which makes analysis from the information gained with timeline and reporting modules.

Communication Module: The last module of the system is communication module. This module is responsible from taking signal information from the sensors. Before requesting signal information from the sensors, first it takes sensor information from the sensor managing module (i.e. which of the sensors are currently active) then it request the information from the sensors which are appropriate.

To see the whole system and data exchanges between them better the following component diagram of the system is provided at figure 1:

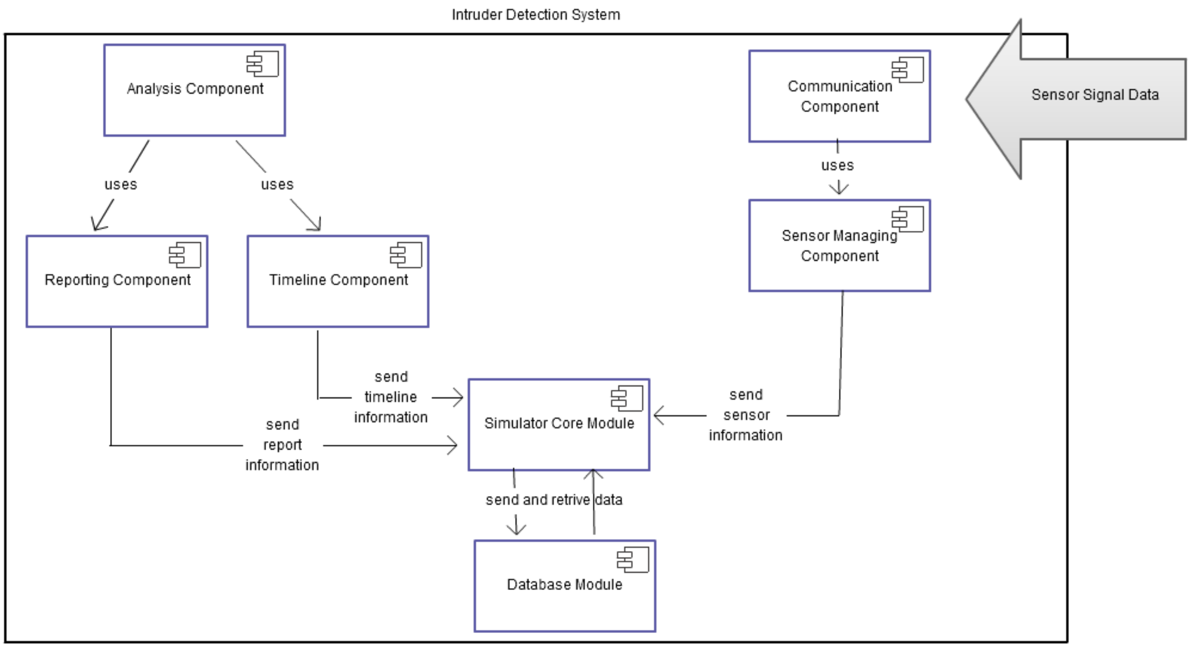


Figure 2. Component diagram of the system.

5.2. Description of Components

5.2.1. Simulator Core Component

The main module of the system is this module. All of the components in the system are connected to this module. This module is used as a communication interface among all the components. In addition to that the only component which has interaction with database (via database module) is simulator core module.

5.2.1.1. Simulator Core Component Processing Narrative

When we designing the system we followed flow based programming paradigm. And this module is created to make the system more suitable for flow based programming paradigm [2].

This module provides an interface between other methods. When one of the modules wants access another modules member function it makes this request to simulator core component then simulator core component makes the function call for that function. This provides a minimum coupling for the system [3].

Moreover all the data transfer is done on this module. For instance, if a module wants to save information to database, it first sends its data to simulator core component. Then simulator core component saves the information to database for that module.

5.2.1.2. Simulator Core Component Interface Description

This module provides a data interface to system. The data between the modules is carried with the help of this interface. Also all the data flow of the system is done on this interface.

5.2.1.3. Simulator Core Component Processing Details

In our system design there is no communication between modules. They only communicate with simulator core component. Simulator core module sees all of the other modules as black boxes. It takes data from that black boxes and send data to another black box which needs it. When one of the modules want to send data to another module. It first writes that data to relevant port of the simulator core component. Then simulator core component sends that data to specified component.

5.2.1.4. Dynamic Behavior

Simulation core component has the responsibility to read data from database and write data to database. The components like sensor managing component, timeline component and reporting component makes changes in the data and want to save this data to database. For this reason simulation core component reads data from these modules.

On the other hand, the modules like analysis module and communication module need to read data from database and make inferences on this information. In order to read data from database they send database reading requests to simulation core component. After these requests simulation core component reads data from database and sends these data to requesting modules as a response.

A sequence diagram that shows all of these interactions of simulation core component is provided below:

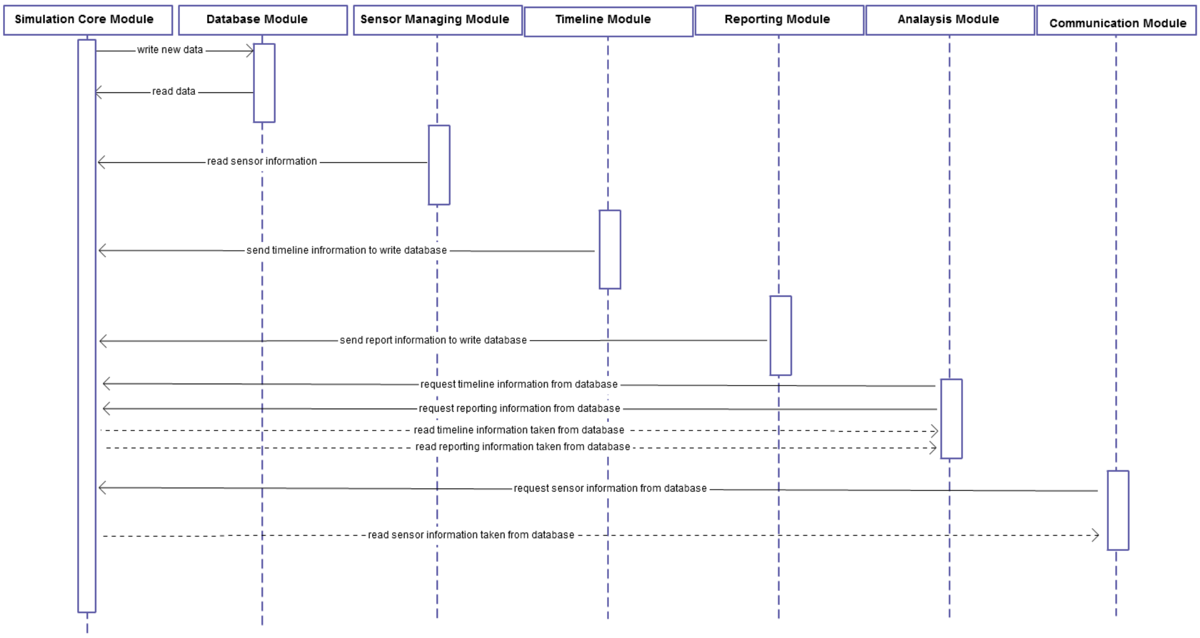


Figure 3. Sequence diagram of the simulation core component.

5.2.2. Database Component

This module is designated to coordinate writing and reading processes that will be done on the database. Accessing to the database when needed is the responsibility of this module.

5.2.2.1. Database Component Processing Narrative

As indicated previously, the only module that interacts with database is simulation core component. Simulation core component reaches database via database module.

This module provides an interface between simulation core component and database system. All the other modules make their database operations over simulation core component. Simulation core component accesses to this module for them. Thus, database component can be seen as a “bridge” between the simulation core component and the physical database.

5.2.2.2. Database Component Interface Description

The database component will be an interface between the simulation core component and the physical database. The simulation core component will make requests from the database component instead of making requests directly to the physical database. The database component will perform these requests.

5.2.2.3. Database Component Processing Details

The alarms get from the SD card of a sensor is recorded in the database. There are two main operations with the database, new alarms may be written into it, or previously written alarms may be read from it.

When the user wants a document of the alarms in the database via the reporting module, the simulation core module sends a request to the database module. Then, a read operation is performed; the database module reads the alarms recorded in the database and sends them to the simulation core component.

When the user wants to save the records get from the SD card of a sensor into the database via the timeline module, the simulation core module takes the alarms from the timeline module and them to the database module. Then, a write operation is performed; the database module writes the alarms sent by the simulation core module to the database. A write operation may also be requested by the sensor managing module.

5.2.2.4. Database Component Dynamic Behavior

The responsibility of the database component is described in the previous section. At the very basic, it performs read and write operations from or to the database requested by the simulation core module. The sequence diagram for the database module is given below:

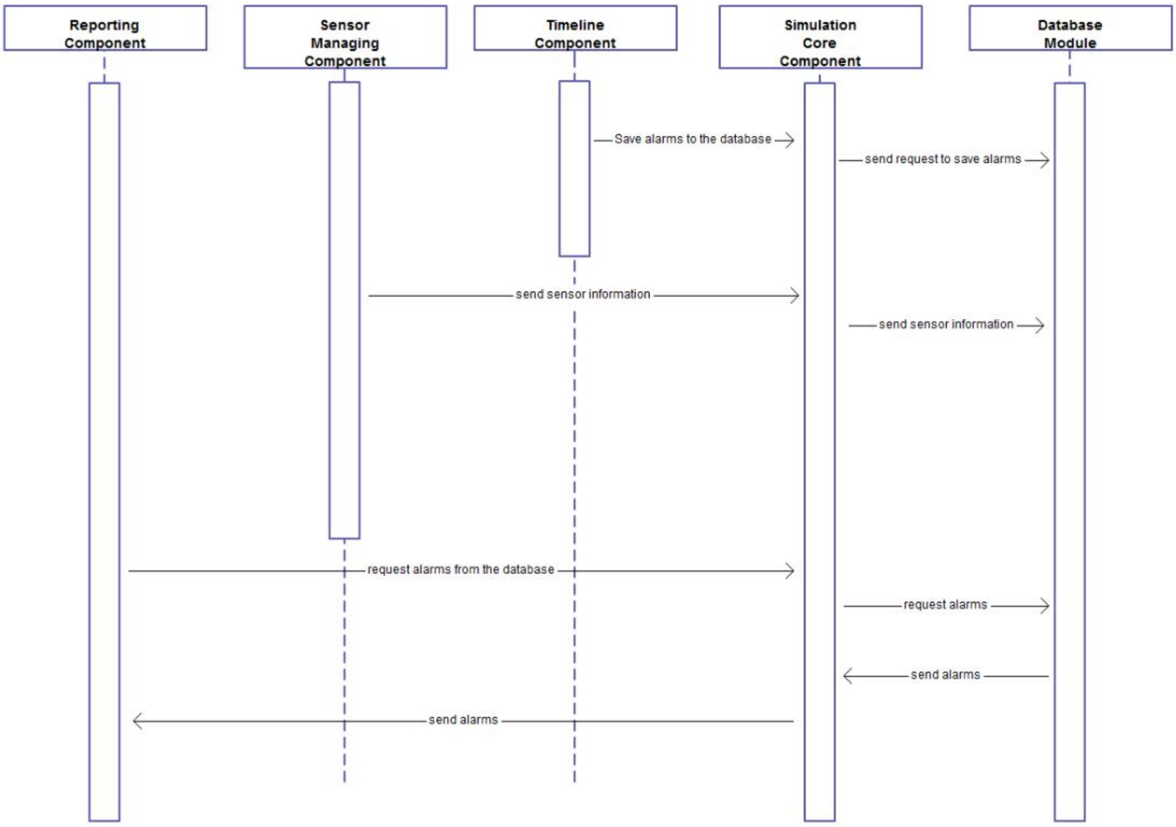


Figure 4. Sequence diagram for the database module.

5.2.3. Sensor Managing Component

In this sub section an explanation of the sensor managing component will be provided. To make a better understanding a sequence diagram will also be provided.

5.2.3.1. Sensor Managing Component Processing Narrative

As the name implies sensor managing component is responsible from all the sensor related organizations. All the processes like changing sensor settings, activating or deactivating sensors are completed inside this component.

The user has direct access to this component. This module takes its input from the user and applies the orders on the data information related to sensor. Also it can read the current status of the sensors with the help of simulation core component as well as writing new data to database.

5.2.3.2. Sensor Managing Component Interface Description

This component provides an interface between the user and the sensors. The user may make all the sensor related tuning and status change information within this module.

5.2.3.3. Sensor Managing Component Processing Details

When user wants to interact with the sensors he/she must use sensor managing component because all of the sensors related information updates are done here.

The user may want to change sensor status. He/she may want a long battery life with activating the sleep mode or he/she may want to take better sensor signal with normal mode. This information is hold in sensor managing module and from time to time the information is written to database. Each time this module called and an order of status change is requested, the module requests related information from simulator core component and simulator core component fetches the information from database. After changing the status, the status information is send and written to the database with the help of simulator core component.

Other information that can be changed with the help of this module is activating or deactivating the sensors. When user wants to activate or deactivate a sensor user makes a request to this module. The component saves the new status and then it sends the command to sensors to apply changes of them. With this way sensor settings can be changed from outside. Then the new information is saved to database as in the case of the status information.

5.2.3.4. Sensor Managing Component Dynamic Behavior

The sensor managing module has the responsibility to make changes on the sensors. It can reach them or make changes on them according to the commands taken from the user. This module can also communicate with the user. It can take new orders to change sensor information from the user. Another communication of the sensor managing module is with the simulation core component. When sensor managing module wants read some information from the database or write some information to the database it communicates with the simulation core component in order to achieve this process. Furthermore sensor managing module can have communication with communication component. Communication component may request sensor information from the sensor managing module to see the current status of the sensor managing component.

A sequence diagram that shows all of these interactions of sensor managing component is provided below:

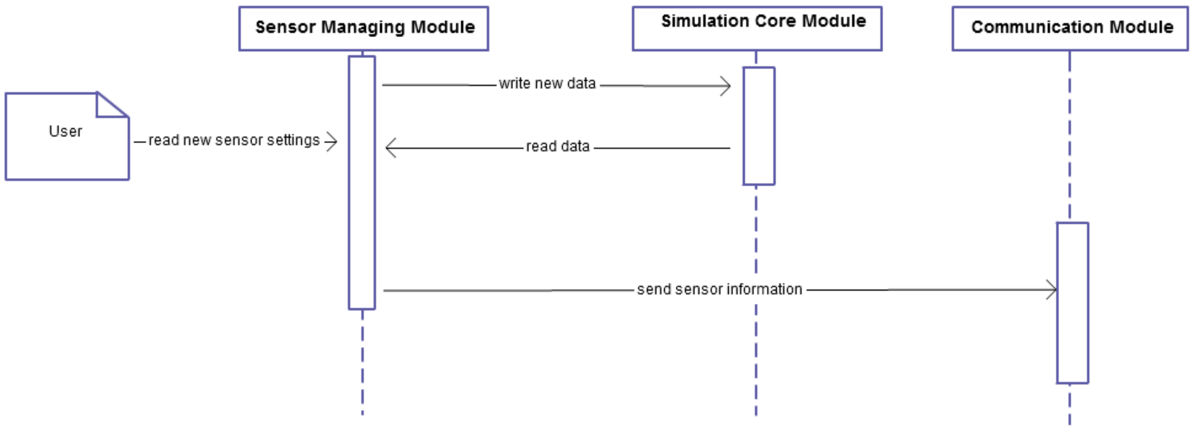


Figure 5. Sequence diagram for the sensor managing module.

5.2.4. Reporting Component

In this subsection responsibilities of the reporting module, its communications with other components and detailed processing descriptions will be provided. Furthermore a sequence diagram that describes the dynamical behavior of this component will be provided.

5.2.4.1. Reporting Component Processing Narrative

This module is responsible from creating a status report of the system. By status report we indicate recent intruder actions on a schedule. The information like when the intruder actions happened, from which of the sensors the signal is taken will be read from the database with the help of simulation core component. Then before creating the written document, if user choses a type, doc or pdf the document will be published accordingly. If user does not choose any type the report will be created as pdf by default.

5.2.4.2. Reporting Component Interface Description

This component will provide and interface to analysis component. It will send data when analysis component requests.

5.2.4.3. Reporting Component Processing Details

This component is responsible from creating reports about the system. It will be interaction with analysis component and user.

From time to time user may want a report of the system to see the progress of the system. The user may want this report for a time period or from the beginning of the system. This preference will be made by the user. According to preference of the user the data that is needed will be requested from database over simulation core component. Reporting component will read the data from the database and will prepare this data to be suitable for the report.

Another responsibility of the report component is sending report information to analysis component. When analysis component requests data the related data shall be read from the data base and send to analysis component.

5.2.4.4. Reporting Component Dynamic Behavior

Reporting component will be in communication with analysis component and the user as well as simulation core component.

Reporting component may take commands from user to specify details of the report. Reporting component will also communicate with simulation core component to reach the database. Furthermore this module will have relations with analysis component. It will send data to analysis component when requested.

A sequence diagram that shows all of these interactions of reporting component is provided below:

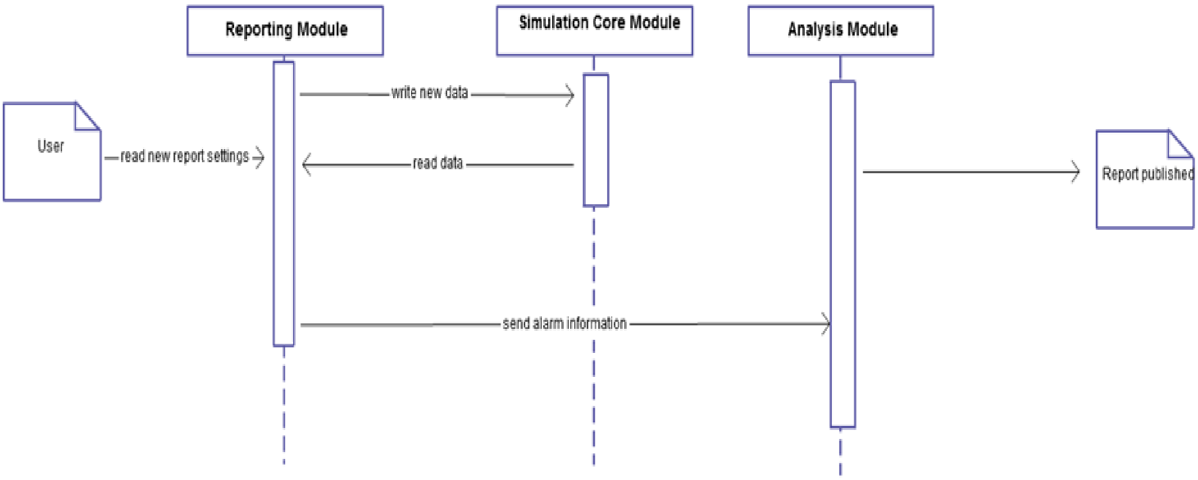


Figure 6. Sequence diagram for the reporting module.

5.2.5. Communication Component

This module is the place where signal data from the sensors is requested, saved and send to the database. The signal information of the system is provided from this component. To illustrate better processing details of the component and a sequence diagram that shows communications of this component will be provided.

5.2.5.1. Communication Component Processing Narrative

Communication module is the module that is responsible from taking signal information from the sensors. Before requesting signals, this module will request sensor status information from the sensor managing module. The signal data taken from the sensors will processed to make the information have a more structured look. Then it will be written to the database.

5.2.5.2. Communication Component Interface Description

Communication component will provide an interface between the sensor signal information and the whole system. After taking signal information from the sensors it will send it to requested places. Also this information will be send to database over the simulation core component.

5.2.5.3. Communication Component Processing Details

This component is responsible from taking signals from the sensors. It will continuously request information from the sensors and will save them to database in a structured manner. For each time stamp it should control which of the sensors are available (activated). To do this the communication component will continuously request information from sensor managing component to control the status of the sensors. After learning which sensors to take information it will send commands to each of the sensors. After taking the data from the sensors it need to restructure the information because they came in a situation that is not possible to save in the database. So it will reshape the structure and send the information to the simulation core component to save to database.

5.2.5.4. Communication Component Dynamic Behavior

The communication component will be in close relationship with sensors and sensor managing component. It will continuously exchange data with them.

A sequence diagram that shows all of these interactions of communication component is provided below:

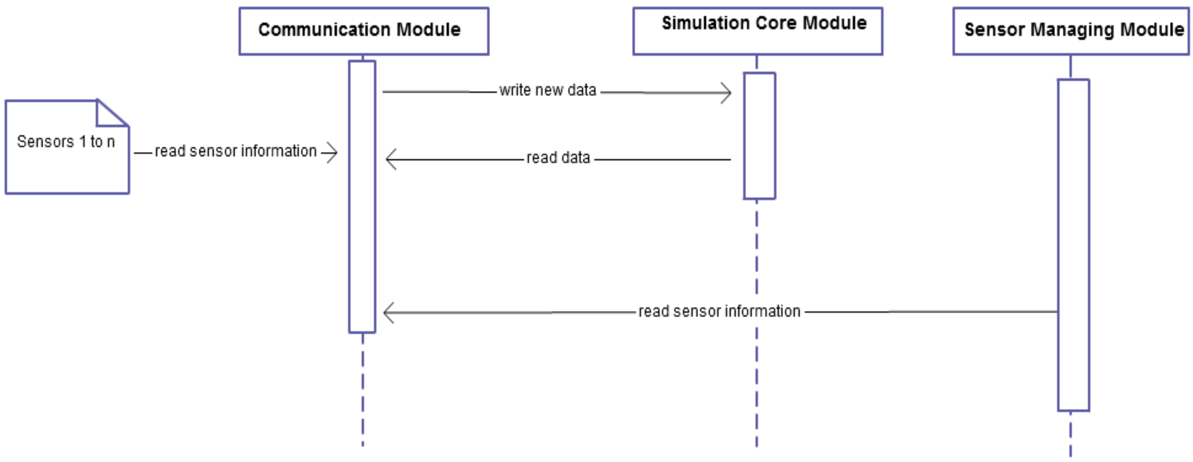


Figure 7. Sequence diagram for the communication module.

5.2.6. Timeline Component

5.2.6.1. Timeline Component Processing Narrative

Timeline module is represented with graphical user interface which is friendly to use. Commonly, it is used to put a sign to the alarm detected timestamp of the timeline. That will give the information about the detection not handling with database management system and core component. Basically, when report button is pressed from main GUI, then the timeline chart will be opened on window and waits a sensor selection for getting the alarm timestamps of the corresponding sensor being available with sensor id. After the selection, alarms are pointed out on the timeline chart. In addition this module sends the timeline information to the simulation core component to save the records into the database.

5.2.6.2. Timeline Component Interface Description

We are planning to create not sophisticated GUI instead we will create a Basic Sensor selection menu and timeline Chart. This module takes the inputs from database with using database management system and the core component. The output is the points on the timeline chart. We are planning to create timeline chart as the following figure.

5.2.6.3. Timeline Component Processing Details

Timeline module firstly sends required information to the core component to take the desired database records. Alarm table of the database includes the signal id, which is a foreign key with Sensor table, date, hour, signal id, signal value, and message about signal this information are taken from the alarm table by core component and it sends these to the timeline module. It analyzes the records and puts a mark on the timeline chart with respect to the timestamp information. Moreover, it sends the taken timeline to the database to save into the database system.

5.2.6.4. Timeline Component Dynamic Behavior

Timeline module is association with the simulation core component, directly, and the database module with the help of simulation core component. Timeline module gives a request to the simulation core component to save the records into the database and the reading the records from the database. This simulation core component will also interact with the database module to get or save the records form or into the database management system.

Sequence diagram of the timeline module as follows;

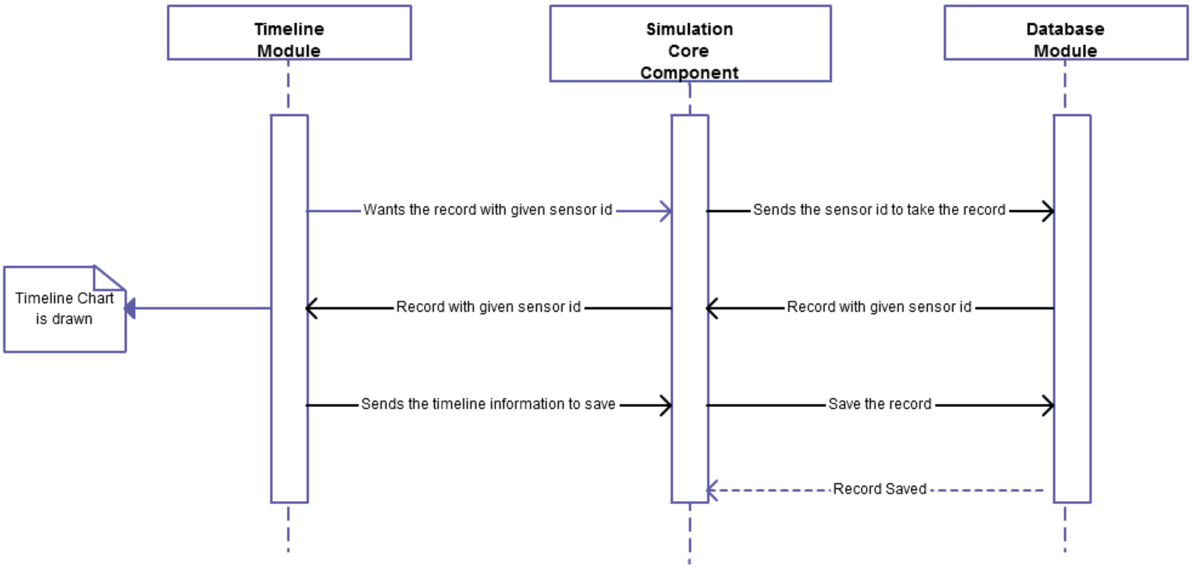


Figure 8. Sequence diagram for the timeline module.

5.2.7. Analysis Component

5.2.7.1. Analysis Component Processing Narrative

Analysis module is such a statistical calculation module for user. It is mostly used for report module to give some statistical values to interpret the intruder detection. The mainly calculation that this module makes is;

- The day and hour which the intruder detection is the most.
- The sensor which the intruder detection is the most.
- The most vulnerable part of the restricted area.

We can calculate any other statistical values with respect to the sensor id. This is beneficial for giving detailed report for the user. With the help of this module, some necessary precautions can be taken into account by the military service to increase the protection of the confidential area.

5.2.7.2. Analysis Component Interface Description

This module takes the input from the database by using the simulation core component. This analysis module is generally the intermediate module between the simulation and the reporting module. Therefore, it takes the input from simulation core component and sends the output to the reporting module.

5.2.7.3. Analysis Component Processing Details

- If the user clicks the report button from the interface, then this module is activated.
- It sends requests for information for all sensors which are available
- Module calculates the predefined statistical values to give the detailed information about the detections
- Reporting module takes this information and writes a piece of report which is PDF or text document.

5.2.7.4. Dynamic Behavior

The sequence diagram of the analysis module is as follows;

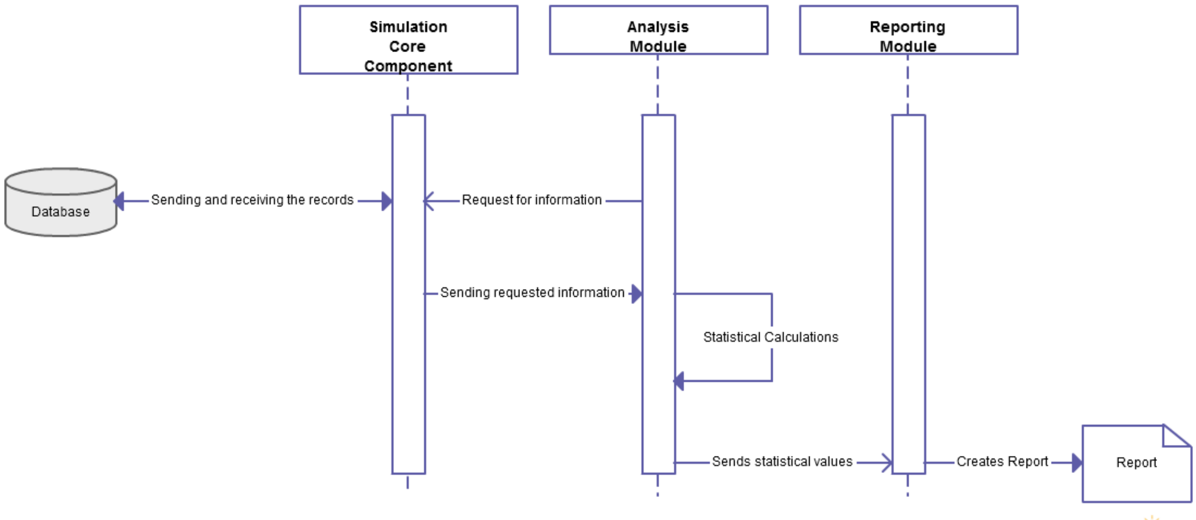


Figure 9. Sequence diagram for the analysis module.

6. User Interface Design

6.1. Overview of User Interface

In this part, we will explain how users will be able to use the system. We will have two types of users and interfaces for each.

6.1.1. Tester Interface

Tester is the person who checks data send by sensors. These type of users will have limited authority.

6.1.1.1. Login Page

To be able to access the system, user must enter a valid username and password to two text boxes provided. Once a valid username/password combination has been entered, the user may attempt to login by pressing enter button. For new users, there is Register link. This link gives access to the Registration page that allows new users to register themselves on the system.

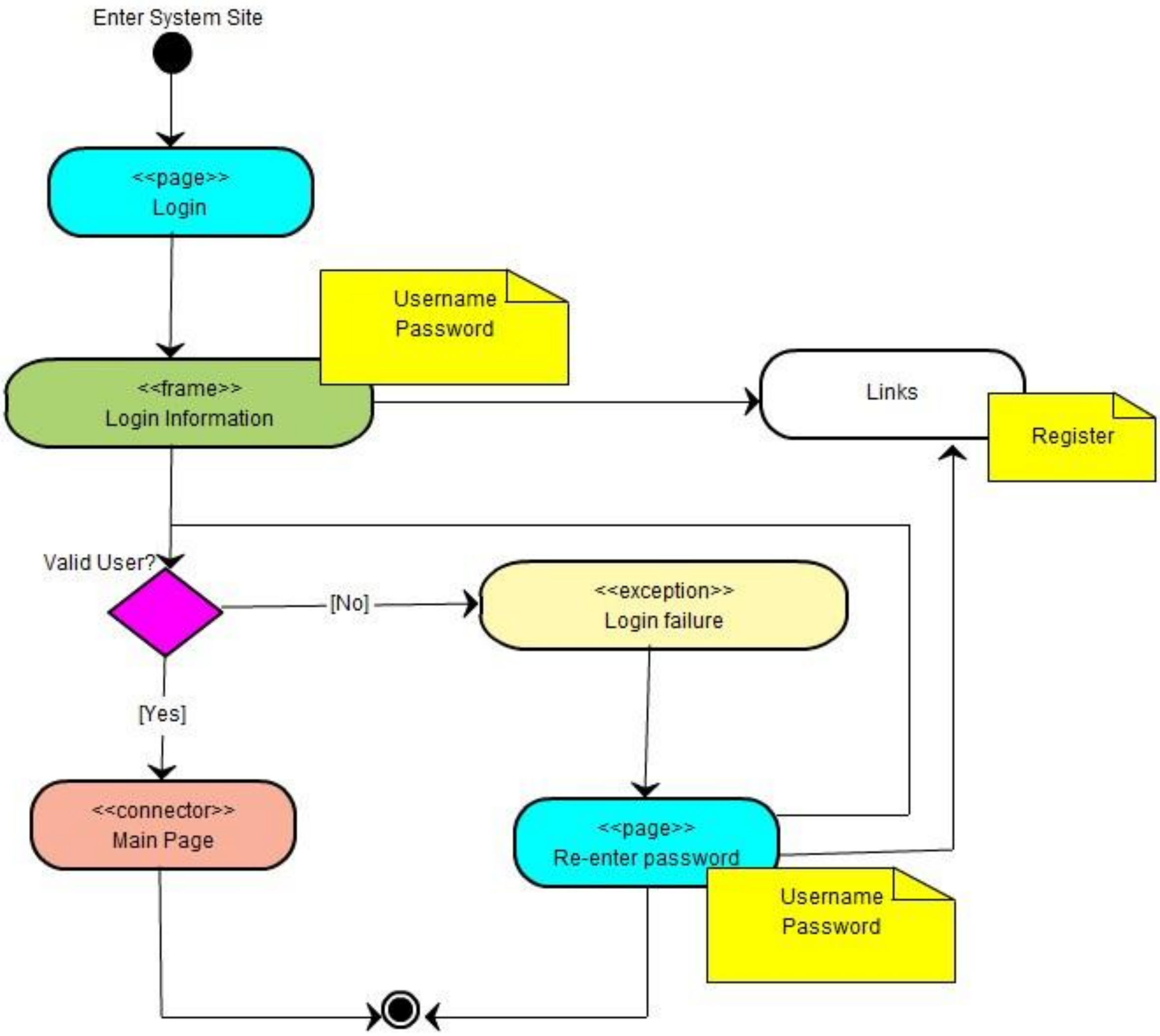


Figure 10. Activity diagram for the login page.

6.1.1.2. Registration Page

This page provides a form. In order to register, the user must fill out the fields which are marked required and click the “submit” button. If username entered exists already, then user will be directed to this page again. After making successful registration, user will be directed to Login page.

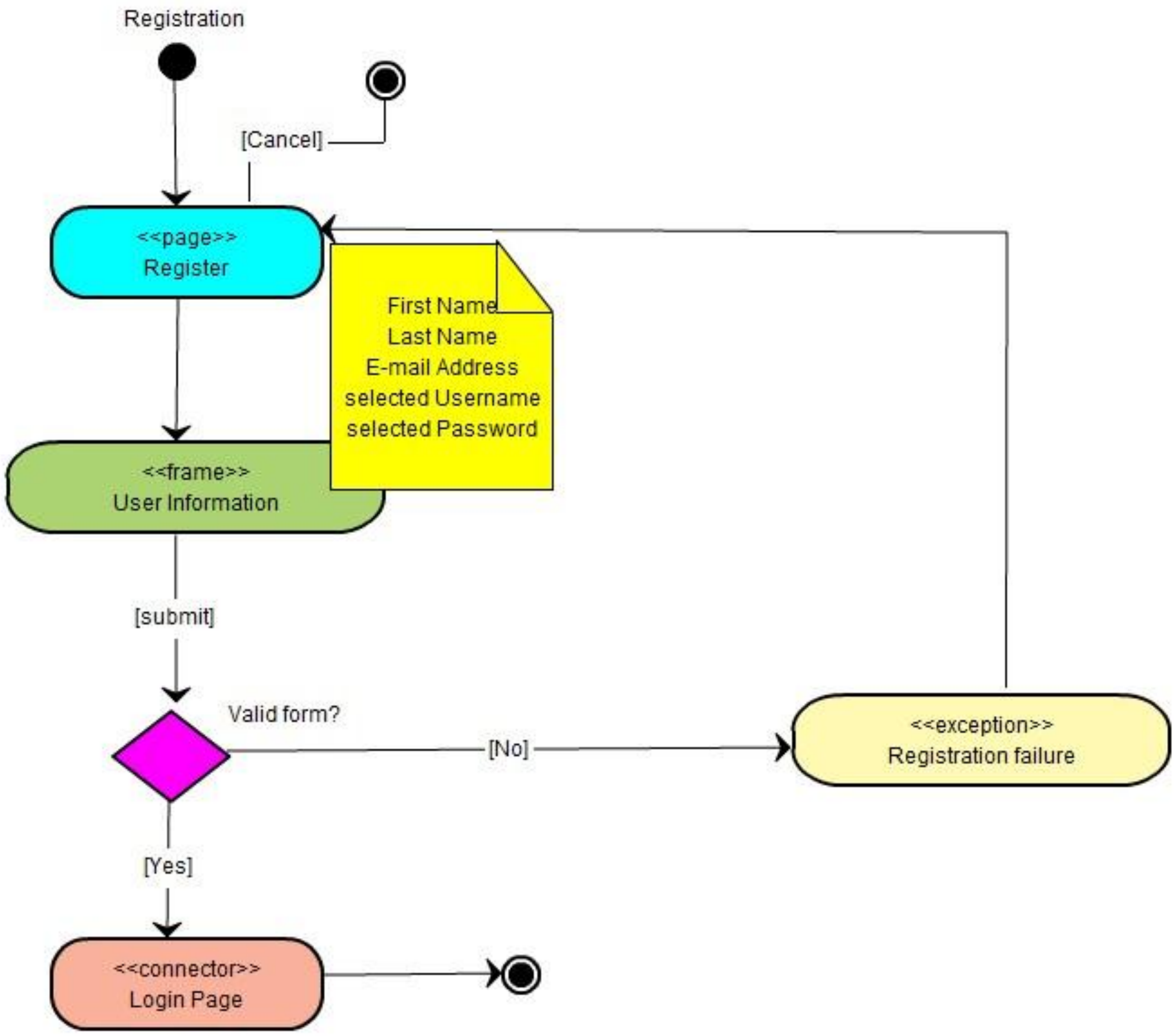


Figure 11. Activity diagram for the registration page.

6.1.1.3. Main Page

When login is successful, this page will be opened in front of user. In this page, a marked map will be available. Each rectangular-shaped marker will show the place of separate geophones. Colors of markers display important information to user:

RED: Last alarm is get from sensor in less than 15 minutes

YELLOW: Last alarm is get from sensor in less than 24 hours but more than 15 minutes

GRAY: Last alarm is get from sensor in more than 24 hours

In addition, when user clicks on one of the markers, information about geophone (such as coordinates, last data time) will be shown to user.

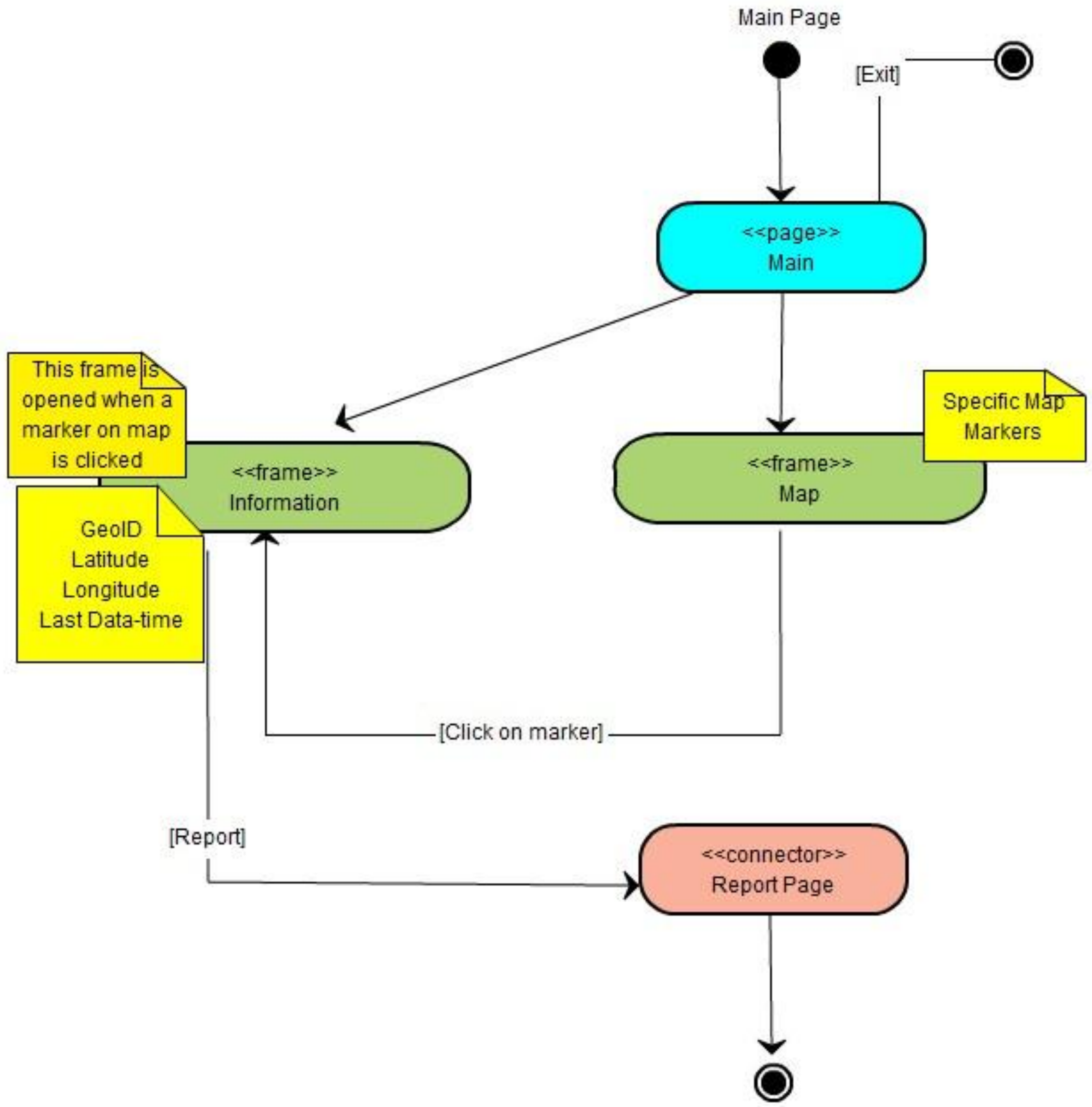


Figure 12. Activity diagram for the main page.

6.1.1.4. Report Page

This page contains some specific information about sensor which is clicked lastly. This information will be ID, place Information, and last data information of geophone. There will be a printout button to get a hard copy of report.

6.1.2. Admin Interface

Admin is the person who has more authority on the system. Admin will be able to make the same operations as tester does. In addition, he will be responsible for adding new user and geophone to the database of the system.

6.1.2.1. Login Page

Login page of admin will be the same as testers except there will not be register link. Admin is already registered himself to database.

6.1.2.2. Control Page

On this page, there will be two parts. First part will be used for adding new geophone to the system. In this part, there will be a form to be filled. This form contains necessary information for geophones to be registered to database. After pressing add button, geophone will be added to database. Second part will contain multiple rows showing users who want to register to the system. Using Add Person / Delete Person, admin will response to these requests.

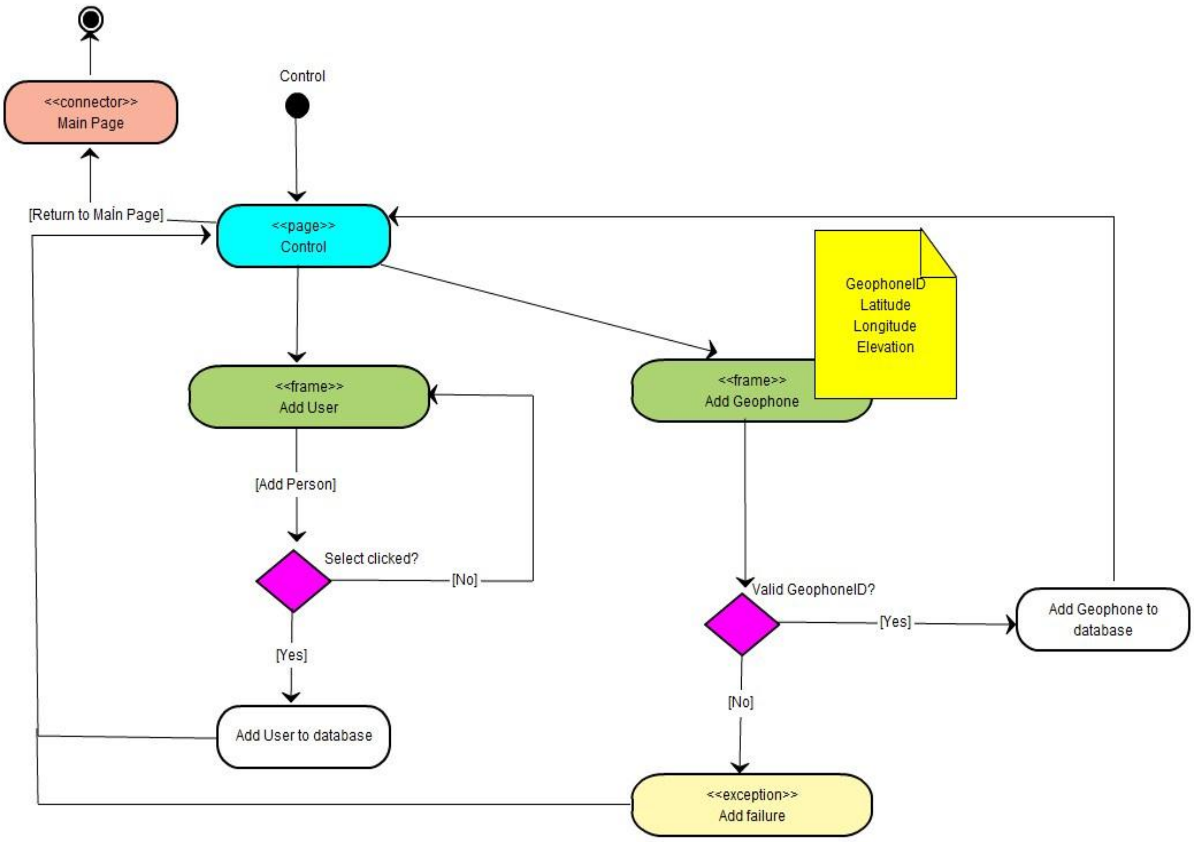


Figure 13. Activity diagram for the control page.

6.1.2.3. Main Page

This page will be the same as tester's Main Page (explained in part 6.1.1.3).

6.1.2.4. Report Page

This page will be the same as tester's Report Page (explained in part 6.1.1.4).

6.2. Screen Images

6.2.1. Login Screen

Figure 14. Login screen

6.2.2. Register Screen

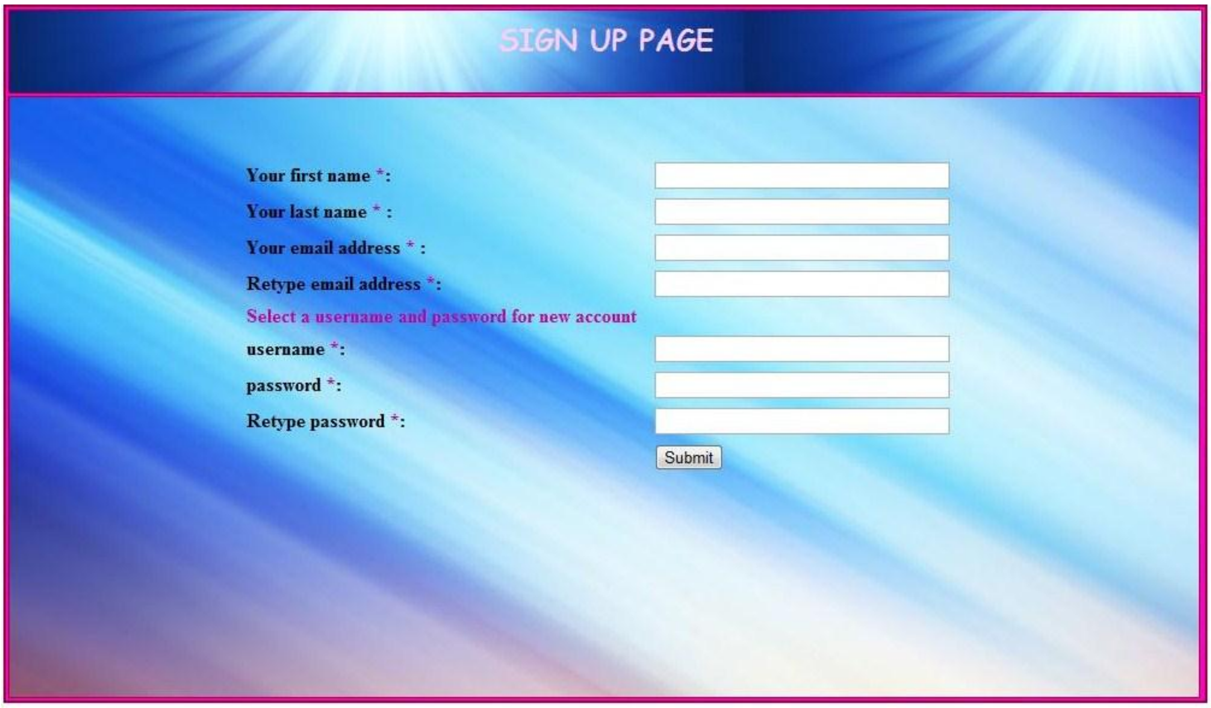


Figure 15. Register screen

6.2.3. Main Screen

Figure 16. Main screen

6.2.4. Control Screen

Figure 17. Control screen

6.3. Screen Objects and Actions

On Control Screen, there will be a link "Return to Main Page". Using this link, user can return to the Main page where the map is available.

On Main Page, there will be a map which is explained in part 6.1.1.3. Double clicking on map, user can zoom in. Double right click on mouse leads user to zoom out map. Using left, right, back, forward icons, user can travel on the map. Zoom in /Zoom out can also be made using +/- icons on the map.

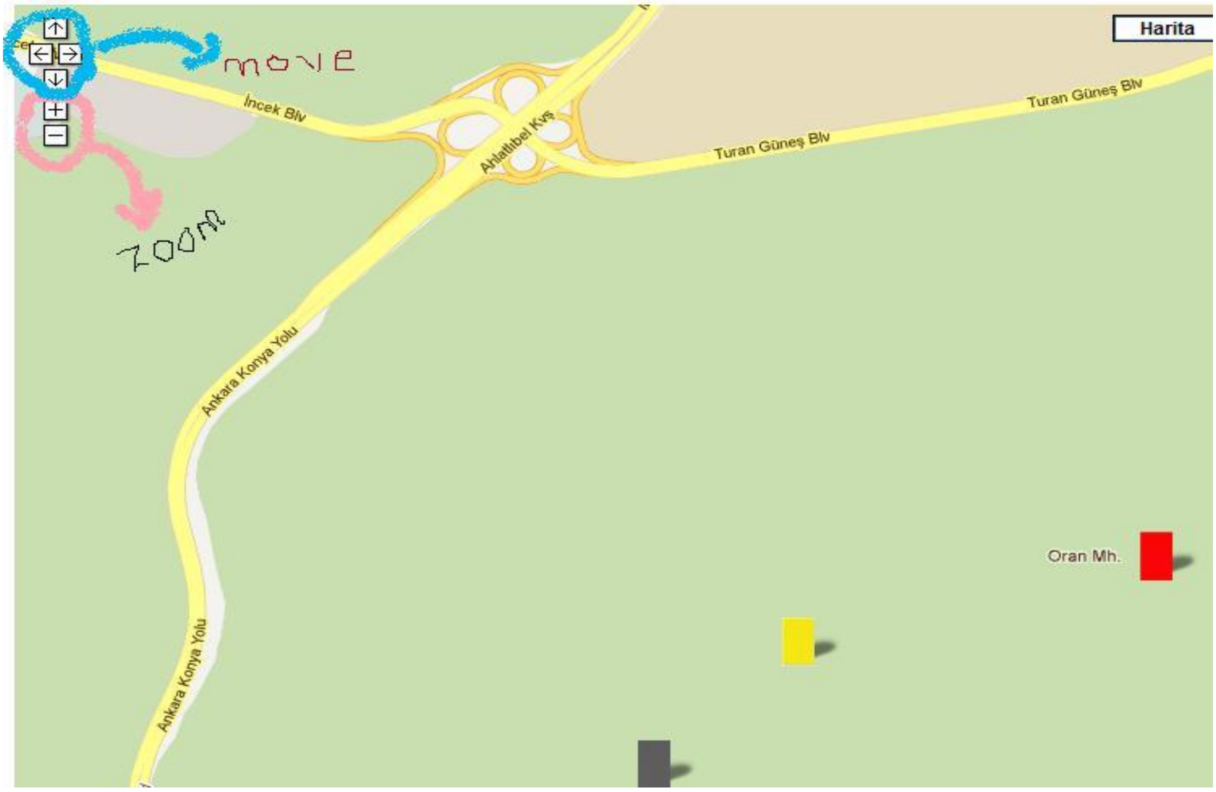


Figure 18. Zoom in/Zoom out and Movement

On map, there are also alternative map types. By clicking on desired map type, user can change the type of map.

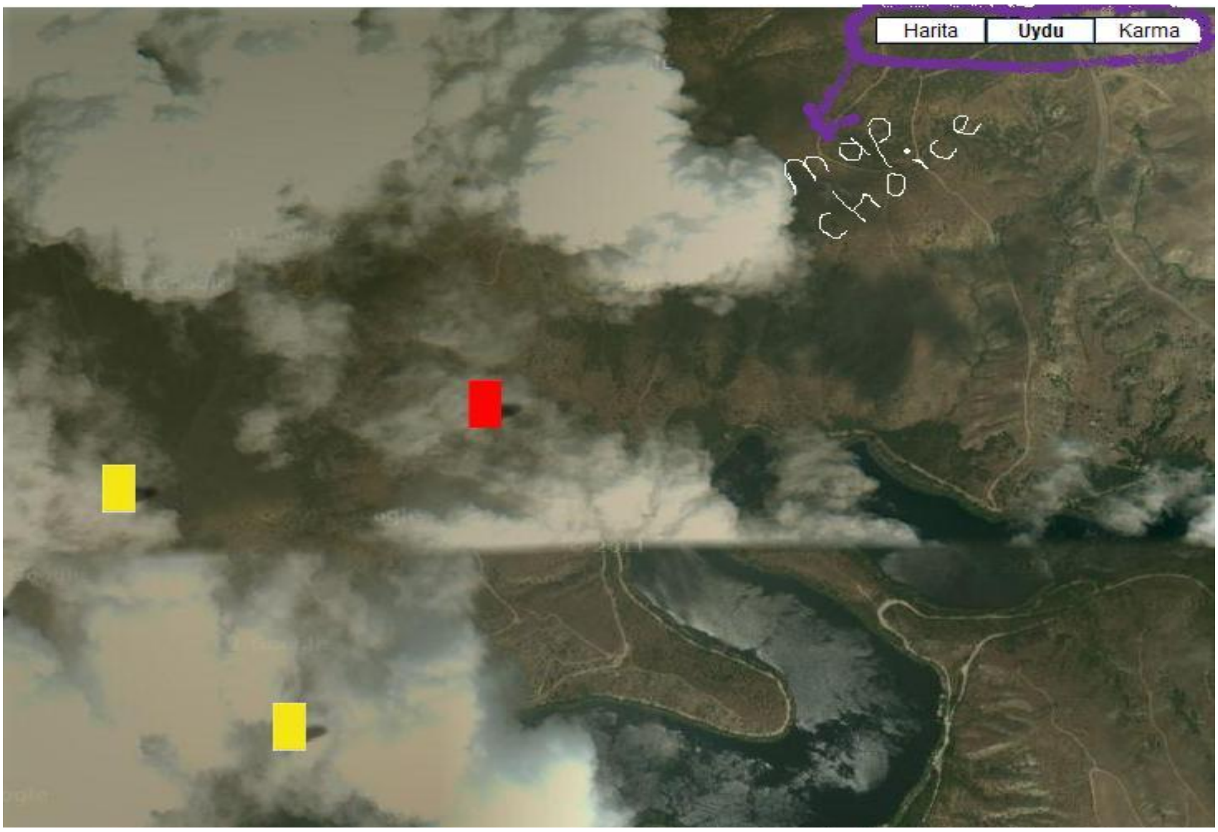


Figure 19. Map type choice

7. Libraries and Tools

The libraries and tools that will be used while developing the intruder detection system are given in this section. For each library and tool, we first give its description and then explain its usage in developing the system.

7.1. Waspmote and Waspmote IDE

7.1.1. Description

Waspmote is an integrated board developed by Libelium Company on which sensors can be integrated. Its usage is extremely wide, it can be used in agriculture, health, industrial processes, marketing etc.

Waspmote IDE is the environment to write the code which will be embedded into the Waspmote. It is developed with Java ANTLR Library. Although it has its own syntax and libraries, its usage is quite similar to C programming language.

7.1.2. Usage in the Intruder Detection System

We are going to use Waspmote integrated board in our design along with a GPS. The GPS is going to tell us the position of the sensor in terms of latitude, longitude and altitude. The main algorithm (The algorithm which determines whether a signal is due to an intruder or not) is going to be embedded in the Waspmote.

7.2. Eclipse IDE

7.2.1. Description

The Eclipse IDE provides libraries and tools for Java developers to build Java applications. It provides validation, incremental compilation, cross-referencing, code assist etc. It also provides debugging tools.

7.2.2. Usage in the Intruder Detection System

The control center part of the intruder detection system will be developed with Java programming language. Thus, we will use Eclipse IDE which is considered by many developers to be one of the best Java development environments.

7.3. PostgreSQL

7.3.1. Description

PostgreSQL is a powerful, open source object-relational database system. It has been developed actively for 15 years. It runs on all major operating systems including Linux, UNIX, and Windows.

It has full support for foreign keys, joins, views, triggers and stored procedures (in multiple languages). It includes most SQL:2008 data types. It supports storage of binary large objects, including pictures, sounds or videos. It has native programming interfaces for C/C++, Java, .Net etc.

7.3.2. Usage in the Intruder Detection System

We are going to use PostgreSQL while creating the database of the system. We are going to take advantage of it by using its programming interface with Java.

7.4. XBee

7.4.1. Description

XBee is a radio module is designed for point-to-point and point-to-multipoint communications by Digi International Company. It is mainly used for wireless communications. An image of XBee Pro is provided below:

Figure 20. XBee Pro

7.4.2. Usage in the Intruder Detection System

The sensor for intruder detection will send the alarms to the control center via wireless communication. Because of its practical and simple usage, XBee will be used for wireless communication infrastructure. It will be integrated with Waspmote board.

7.5. SD Card

7.5.1. Description

SD is a non-volatile memory card format developed for use in portable devices. As its practicality and small dimensions, it is being used for data transfer between devices like flash disks.

7.5.2. Usage in the Intruder Detection System

Besides sending the alarm to the control center, they will also be recorded in the sensor. The alarms will be saved in SD card which is also going to be integrated with Waspmote board. The user will be able to read the alarms saved in the SD card at a later time. In other words, it will be used as the main storage device by the sensor itself.

In the following figures, we present two images of the sensor. Please note that, the hardware consists of Waspmote with GPS, XBee and an SD card.

Figure 21. The sensor (not hidden)

Figure 22. The sensor (hidden)

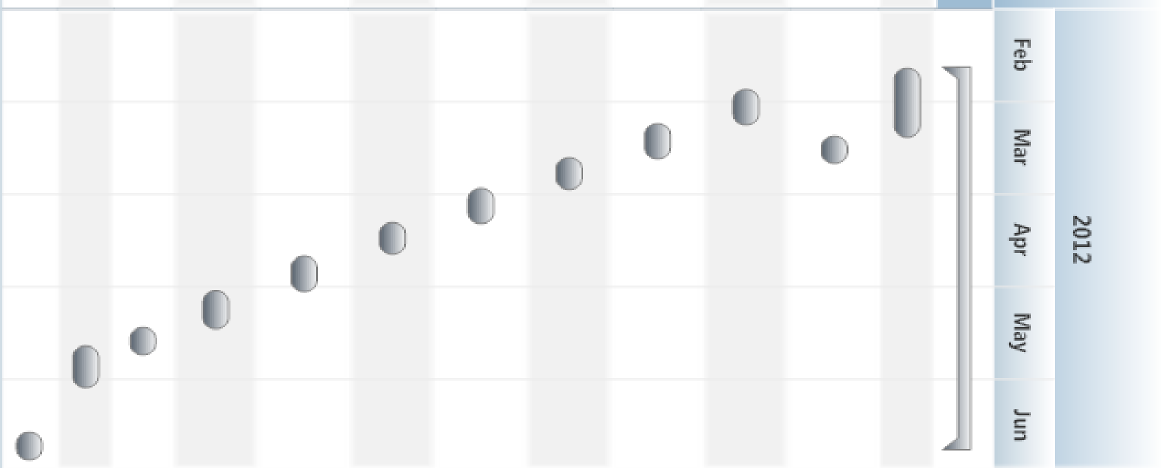
8. Time Planning

8.1. Term 1 Gannt Chart

	Task	Assigned To	Start	End	Dur	%
	Project (Intruder Detection System)		1/10/11	25/1/12	83	
1	Topic Selection		1/10/11	10/10/11	6	
2	Literature Search		10/10/11	17/10/11	6	
3	Marketing Search		10/10/11	17/10/11	6	
4	Project Proposal		17/10/11	31/10/11	11	
5	Software Requirements Spec.		17/11/11	25/11/11	7	
6	Testing with Sensor Equipment		25/11/11	20/12/11	18	
7	Initial Design Report		8/12/11	20/12/11	9	
8	Designing User Interfaces		17/12/11	9/1/12	16	
9	Detailed Design Report		20/12/11	9/1/12	15	
10	Creating the Database		17/12/11	9/1/12	16	
11	Prototype Demo		21/1/12	25/1/12	3	

8.2. Term 2 Gannt Chart

	Task	Assigned To	Start	End	Dur	%
	Project (Intruder Detection System)		20/2/12	23/6/12	90	
1	Search for Detection Algorithms		20/2/12	12/3/12	16	
2	Implementation of Detection Algorithm		12/3/12	19/3/12	6	
3	Implementation of Sensor Managing Module		27/2/12	8/3/12	9	
4	Implementation of Database Module		8/3/12	19/3/12	8	
5	Implementation of Timeline Module		19/3/12	29/3/12	9	
6	Implementation of Reporting Module		29/3/12	10/4/12	9	
7	Implementation of Analysis Module		10/4/12	20/4/12	9	
8	Implementation of Communication Module		21/4/12	2/5/12	8	
9	Implementation of Simulation Core Module		2/5/12	14/5/12	9	
10	Assembling the Modules		14/5/12	20/5/12	5	
11	Testing and Debugging		20/5/12	3/6/12	10	
12	Presentations		18/6/12	23/6/12	5	



9. Conclusion

This documentation provides implementation details of the intruder detection system up to a point. In this document, basics of data design, modules and user interfaces of the system are described, respectively. The tools and libraries that will be used while designing the system are also described. Moreover, timeline describing the design process is provided.

As this is the initial design report, design of the system is not deeply described; instead, a brief overview of the design of the system is given. Detailed design of the system is going to be described in the detailed design report. Thus, this report can be seen as a preparation for the detailed design report.