

**Security
System with
Microsoft
Kinect**

**Software
Requirements
Specifications**

- Erdinç Kaya**
- İlhan Yoldaş Karabulut**
- Alişan Yılmaz**
- Utku Şahin**



Sponsored by Simsoft

2011

Table of Contents

1. Introduction	5
1.1. Problem Definition:	5
1.2. Purpose	6
1.3. Scope.....	6
1.3.1 How Does Kinect Works?	7
1.3.2 What Do We Intend To Do?	7
1.3.3 The Movement Processing System	7
1.3.4 The Security System	8
1.4 User and Literature Survey	8
1.5 Definitions and Abbreviations	9
1.6 References.....	9
1.7. Overview	11
2. Overall Description	11
2.1. Product Perspective	11
2.2. Product Functionalities	12
2.1.1 Motion definition and recognition system (MDRS)	12
2.1.2 Communication System	12
2.1.3 User Interface	12
2.2 Product Functions.....	13
2.2.1 Movement Detection	13
2.2.2 Classifying Movements	13
2.2.3 Detecting Suspicious Movements.....	13
2.2.4 Notification	14
2.3. Constraints, Assumptions and Dependencies	15
3. Specific Requirements	16
3.1 Interface Requirements	16
3.1.1 General View	16
3.1.2 Ownership Page.....	16

3.1.3 Administrator Page	17
3.1.4 Movement Defining Page.....	17
3.1.5 Notification Visualization Session	17
3.2 Functional Requirements	17
3.2.1. User Functional Requirements	18
3.2.1.1. Start application	18
3.2.2. Product Selection Mode Handler Functional Requirements.....	18
3.2.2.1. Add administer.....	18
3.2.2.2. Delete administer.....	18
3.2.2.3. Quit.....	18
3.2.3. Real Time Handler Functional Requirements	19
3.2.3.1. Recognize People	19
3.2.3.2. Recognize Movement.....	19
3.2.3.3. Classify Movement.....	19
3.2.3.4. Recognize Suspicious Movement.....	19
3.2.3.5. Warn User.....	20
3.3. Non-functional Requirements.....	20
3.3.1. Performance requirements	20
3.3.2. Design constraints.....	21
4.Data Model and Description	21
4.1. Data Description	21
4.1.1. Data objects.....	22
4.1.2 Relationships	22
4.1.3 Complete Data Model	24
4.1.4 Data Dictionary	25
4.1.4.1. Admin	25
4.1.4.2 .Movement	25
4.1.4.3. Tree	26
4.1.4.4. Log.....	26
4.1.4.5 Notification	26
5. Behavioral Model and Description	26
5.1. Description for Software Behavior	26

5.1.1. Defining Movements Mode.....	26
5.1.2. Security Monitoring Mode	27
5.2. State Transition Diagram	28
6. Planning	29
6.1. Team Structure	29
6.2 Estimation (Basic Schedule)	30
6.3. Process Model	31
7. Conclusion.....	31

1. Introduction

This Software requirements Specification report is a complete description on project “Security System with Microsoft Kinect”. The project will be developed as a final design project by group 213 at Middle East Technical University Computer Engineering Department. This project sponsored by Simsoft.

1.1. Problem Definition:

Nowadays, it is easy to see security cameras everywhere. But, are there anyone watching behind them, are they recording, are they even working? These are the first questions that come to the mind. The main purpose of security cameras is to discourage the criminals. If a crime happens the security cameras can be used for an investigation. There is usually no person watching behind a security camera. Because employing a person for monitoring through a camera all the time is expensive for many businesses. Moreover, these systems are not really useful because of human factors. In many cases, after a security issue has occurred, the videos that have been recorded have to be watched for hours to find the exact time and event. Even it is not guaranteed that the criminal will be found. Our team intends to solve this problem in every side. The security system with kinect will be able to decrease the active monitoring time and increase the reliability of the camera system. It is going to decrease the spent time across the monitor by catching suspicious movements and instant reporting system. It will also Increase the reliability because our security system can work without the active help of a human.

1.2. Purpose

This document specifies the requirements of security system with kinect project. The specifications to explain our goal and the path that group 213 have drawn for the project, which means what we aim for our system to look and work like. To illustrate more clearly, this document give information about interfaces, functionalities of the system, dependencies, expected results.

This document will give an illustration about the project we aim to build. The target groups are the 213 group, Simsoft members and departmental instructors. The main purpose is to decide the requirements of the project and to decide a common ground between these supporters and developers. The specifications of the system will help to decide that whether the system meets the needs of the customers and producers and supporters. More importantly the document will help us to make changes for better, ordered and error free.

Another purpose is the document will help us to reduce time for new specifications or modifications since it is a complete summary of the system we can see from every angle without hesitation. We can find faster inconsistencies, disagreements for the sake of our success.

1.3. Scope

Our project is called Security system with kinect. As can be understand by it is name, we intend to build a program working with the device kinect, that Microsoft developed. For the make it more clear we will give information about the system in 4 parts.

1.3.1 How Does Kinect Works?

Kinect is one of the newest devices that Microsoft developed. It is pretty similar to a video camera but can give information about all three dimensions of an object within the range. It returns the body positions with twenty[3] coordinates.

1.3.2 What Do We Intend To Do?

The kinect device will be placed to the desired area. Since every place has different security issues new movements will be implemented to the program by The Movement Processing System. Every movement will be categorized as their priority of suspicions. If any of these movements occur, the device will send warning information to both administrator and the owner of the product. These warning may be sent to the system itself, may be sent as an e-mail or text-message according to the importance of the situation. Even a message directly can be sent to the police if it seems necessary.

Our project has two basic parts. The first part will be the core of the system that analyze the movements and categorize them. The second part will be built on the first one.

1.3.3 The Movement Processing System

This is the core part of our project Movement Definition and recognition system (MDRS). With the help of the kinect we will categorize every movement and will be able to recognize what it was. After the hardware implemented we will be able to develop a movement database easily with these system.

1.3.4 The Security System

This is the part where the users manage the system. There are two interfaces. First one is for the owner of the product in order to register the administrators who will use the programs. The second interface for the administrators to use the products. Both interfaces have a video visualization, notification page and a last notification part. They will be able to see the situations occurred by clicking thumbnails. Instant notifications will make handling the situation easier. So, no one will be able to stand across a monitor watching everything.

1.4 User and Literature Survey

In this project there are two sides of the literature survey. One side is what kind of applications are made by using kinect. This side is critical for understanding abilities of kinect and what could or could not do with kinect. The other side is security systems and their limitations. By knowing security systems abilities we could know that how can we improve this systems.

When we look at the applications that are made by using kinect, we saw that they are most games and entertainment purposed applications[2]. However, when we examine these games uses skeletal information that kinect provide, in run time. Kinect give us the ability of processing skeletal information simultaneously while getting this information. Literature survey show us this is a new situation. Most of the research on processing skeletal information based on comparing information sets which previously recorded[4]. In our project we will compare real time information that we obtain and previously recorded data. This ability make us could use Kinect based application as a security system.

There are a lots of camera based on security systems uses different technologies like surveillance, motion triggered and infra-red cameras. Some of them can inform peoples whether there are a movement or not .However none of these system could not identify what kind of a movement detected so this system cause a lots of an important warnings. For instance a mouse can cause a red alarm in a bank with motion triggered system. Our system solves this problem by identifying movement. In addition to this as a part of our application it can be defined usage specific movements, application will not use same data set for every place.

1.5 Definitions and Abbreviations

SRS: Software Requirements Specification.

SDK: Software Development Kit.

MDRS : Movement Definition and Recognition System.

N/A : Not applicable.

1.6 References

Referenced documents for this SRS document is provided in Table X. Each referenced document is listed by title, report number or version (if applicable), date, and publishing organization information. Additionally, the sources from which the references can be obtained are provided.

Table 1: References List

Reference Number	Title	Report Number or Version	Date	Publishing - Organization
1	IEEE Recommended Practice for Software Requirements Specifications	IEEE Std 830-1998 (Revision of IEEE Std 830-1993)	June 25, 1998	IEEE
2	Microsoft Kinect Official Webpage, http://www.xbox.com/en-US/kinect.	N/A	N/A	Microsoft
3	Kinect SDK web page , http://research.microsoft.com/en-us/um/redmond/projects/kinectsdk/	N/A	N/A	Microsoft

4	Ontology- based Classification of 3D Virtual Human Motion Data	N/A	N/A	Veysi İşler
---	---	-----	-----	-------------

1.7. Overview

This document is a detailed explanation of the Security system with Microsoft Kinect project. In the first part the general sides of the project was mentioned. How the project will be, what the expectations are, what the purpose is and how it will be used. In the second part a more detailed explanation of the project about the functional requirements, design concepts and the project functions was mentioned. Later the data model was built and a detailed description was made. In the next part the behavioral model was made and detailed. At the last part the group model, estimation of future works and our process model can be found.

2. Overall Description

This section will give information about product perspective, product functions, constraints, assumptions and dependencies.

2.1. Product Perspective

The Motion recognizing security system has two different visual parts one for expanding the database of the movements and the other one is user interface.

Moreover, it has two more important parts which are the communication system and the motion tree that is used to identify the movements.

Basically, the system is made of four parts. These parts are;

2.2. Product Functionalities

2.1.1 Motion definition and recognition system (MDRS)

This system is the most important part of the product. The ones ,that is going to implement a new movement, has to follow few steps at MDRS page. The usage of the page is pretty simple. The implementer has to start system with a single button after entering the name of the movements. 2 seconds after that the program will start working in a 5 second period. After, 3 time analyzing the new movement will create a new movement object according to our algorithms. The new movement will be listed with the other movements at the list at the MDRS. Next step is to decide the security level of the movement, the movement can be listed as low, medium or high.

2.1.2 Communication System

The communication system is what make the program more useful while handling the situations. After MDRS system has decided that a movement is suspicious, it will send a notification to the user interface. According to the priority of the movement messages will be sent to the authorities such as owner, security guards even to the police via e-mail and text-messages.

2.1.3 User Interface

There are two user interface for the system. One of them is owner and the other one is administrator interface. Owner will have an special owner id and password. He has authority to register the administrators of the system. Both of them will be able to

view every suspicious movements through history windows and will be able to view them with a single window popping through page. The last notifications will be shown in red and will be remaining there. Both will be able to monitor the view.

2.2 Product Functions

2.2.1 Movement Detection

Firstly this function fetches critical points of data from Microsoft Kinect SDK and , process them all. At the end of the process progress this function returns the list of the basic movements with classifying simply, such as arm movement, leg movement, head movement etc.

2.2.2 Classifying Movements

This function classifies movements in high level. The classifying criteria is tagging movements respect to pre-defined movements. The pre-defined movements consist of experimental data and self learned data. The real progress of the function that matches obtained movements with pre-defined movements, and figures out the relation between obtained data and pre-defined data. After figuring out the relation, it classifies the movement, such as walking, running, punching etc. Otherwise, if it appropriate to classifying, function creates a new movement tag or type and again classifies the movement with new tag or type. If it is not appropriate, it ignores the movement.

2.2.3 Detecting Suspicious Movements

This function gets the highly classified movements and searches whether the movement is suspicious or not. If the movement is suspicious then function returns true, and let the progress continue.

2.2.4 Notification

Notification function warns the administrator or user via two different ways. The first way of warning the user or administrator is that printing notification message to the notification area on the user interface. The second way of the warning is sending a short message to the administrator or user's cell phone. The short message includes information of notification message.

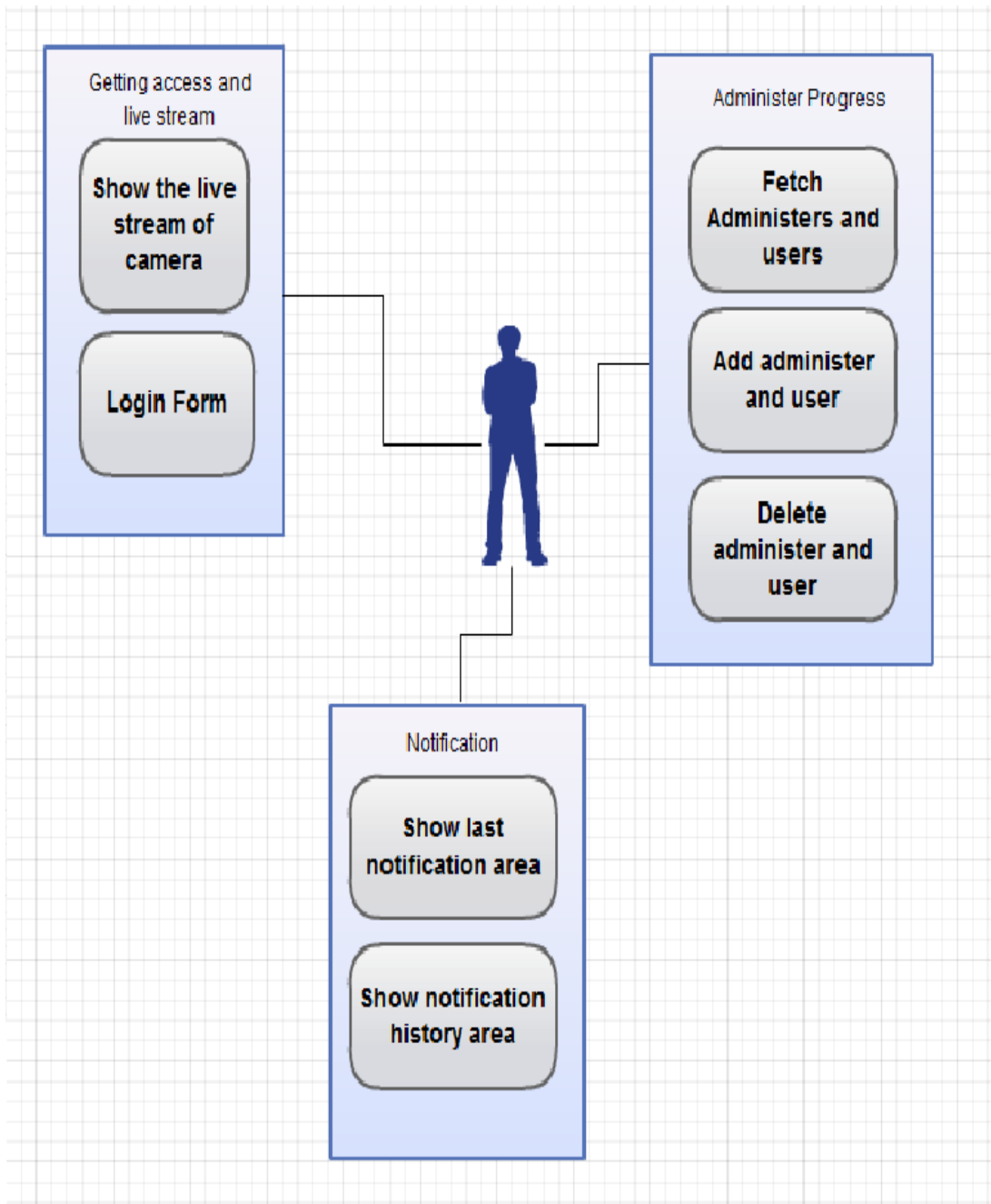


Figure 1 - Use Case Diagram

2.3. Constraints, Assumptions and Dependencies

- Kinect has tree meters width and depth of observing ability. So, our application won't be able to applied in wide areas.

- Kinect Windows SDK only works on Windows 7. So, it is not portable.
- Windows SDK can only be developed with C++ and C#, Visual Basic.
- Kinect doesn't work properly at outside. The sunshine affects its progress. For the better performance it should be inside.
- Kinect can't detect more than two people at a time. So it won't be able to use it in crowded places.

3. Specific Requirements

3.1 Interface Requirements

There will be four main interfaces. First three of them will be used by users. The last one will be for us to develop new movements according to the situations.

3.1.1 General View

This is the very first page the user will reach. This page's main purpose is to direct the owner and the administrator to the different pages. There are going to be a camera view and a login field. According to id and password the user will be directed.

3.1.2 Ownership Page

This is the page that only the customer of the product can reach. The main purpose of this interface is to decide who are going to be the administrators. There will be a form that includes the information of the administrator including contact information, In order to add new administrators. There will be a list of administrators to update and delete the previous administrators. Moreover, there is going to be the last notification and the notification history panels at this page.

3.1.3 Administrator Page

The purpose of this interface is to monitor the situations. There will be a notification history panel and a last notification panel. A visual to monitor the area kinect placed. The administrator page is pretty similar to ownership page. There are only no new administrator insertion panel and administrator list.

3.1.4 Movement Defining Page

This interface is not for users. It is for us developers in order to monitor and visualize the movements just created. There will be a video of kinect visuals showing body positions with dots. There will be a Movements list and insertion panel for new movements.

3.1.5 Notification Visualization Session

Notification history will be at administrator and ownership panels. This will be a list of thumbnails showing recently happened suspicious movements. The user will be able to click thumbnail and watch the video recorded for this event. The information of administrator monitoring the events will be also listed there.

3.2 Functional Requirements

Functional requirements of this application can be categorized in two parts, namely user functional requirement, the functions called with the user activity , and device functional requirements, the functions called automatically.

3.2.1. User Functional Requirements

3.2.1.1. Start application

Description: The user shall start the application.

Assumption: The application is loaded on the device.

How: By running the application.

3.2.2. Product Selection Mode Handler Functional Requirements

3.2.2.1. Add administer

Description: The user shall add the administer to the list.

Assumption: The list is already defined and displayed on the screen.

The screen is on menu mode.

How: The user adds the administer to the list.

3.2.2.2. Delete administer

Description: The user shall delete the administer from the list

Assumption: The list is already defined and displayed on the screen.

The screen is on menu mode.

How: The user deletes the current administer from the list.

3.2.2.3. Quit

Description: The user shall exit the application whenever he wants

Assumption: There will be no assumption for the system to execute this function.

How: Clicking the close button or using alt + F4 button combination.

3.2.3. Real Time Handler Functional Requirements

3.2.3.1. Recognize People

Description: People are recognized by Microsoft Kinect, respect to their critical points, namely, people's skeleton are created.

Assumption: People's skeleton are created by kinect by using infrared technology, and people have to be in the 3 meter x 3 meter area to kinect.

How: When the people in the 3 meter x 3 meter area, people's skeleton are created automatically, and program displays the vision.

3.2.3.2. Recognize Movement

Description: Movements are recognized by processing people's skeleton, respect to their critical points.

Assumption: Recognize the movement via matching and using some algorithms.

How: With obtained people's skeleton program will figure out, which part of the people's skeleton moves, namely, detecting movements of people's member.

3.2.3.3. Classify Movement

Description: Classifies the movement according to movement's behavior.

Assumption: Movements are already defined in the movement database. Every movement has a unique behavior with respect to its class.

How: Movement is matched by searching in the movement database, according to its behavior, then if it appropriates movement is classified.

3.2.3.4. Recognize Suspicious Movement

Description: Recognize suspicious movement.

Assumption: Every movement class has a tag for suspicious or unsuspecting in the movement database, thus tag of the movement tells us the movement's kind.

How: When the classified movement is obtained, just looking its class's kind tag, and detecting.

3.2.3.5. Warn User

Description: Warns user if suspicious movement occurs.

Assumption: When the suspicious movement detects it notifies essential positions.

How: Notifies administrator or necessary positions via sending short message to their cell phone or showing notification history and last notification area.

3.3. Non-functional Requirements

3.3.1. Performance requirements

First, this application will be used by a single user. There will be no multiple user handling since the application runs on a single portable device without needing any server client terminology. However application needs short message technology, therefore application must have an internet connection. The amount of the input is not very huge but it comes repeatedly and application must process them all with respect to an algorithmic sequence. Process of inputs is very important, because this part composes of the huge part of the application, so in performance point of view, movement detection algorithms must be efficient as much as possible. The application is able to detect two people at the same time, this means that application detects the movements of two people, namely, progress multiplied by two. Kinect's position is also important, since kinect's camera works better artificial lighted environment, otherwise kinect's camera turns the sun light, it's detection and generation of critical points rate decreases.

The major issue here is the application should answer in real-time, namely, the detecting and classifying movements has to be handled in less than 1 second. In order to achieve this efficiency rate, application algorithm use efficient approaches such as dynamic programming, which is very useful the repeated calculations.

3.3.2. Design constraints

The application will follow the following design constraints:

Programming language that will be used during the project is C#. Since Microsoft Kinect SDK supports only C# or C++, moreover examples on the internet related to Microsoft Kinect SDK is mostly related to C#, and also most researches is related to C# too. Therefore application will coded in C# language. Because of the language of the application is C# application will developed with Microsoft Visual Studio.

The system shall run on a computer and Kinect. Also, the processor speed of the computer, the quality of the graphics card should be sufficient enough to be able to run the program. In addition to this, computer's operating system must be Windows 7, since Microsoft Kinect SDK works on only Windows 7.

The application needs to send short message to administer or essential positions' cell phone, hence the computer which installed the application, must have an internet connection. Another constraint is Kinect camera will be located in 3 meter x 3 meter area, because kinect can recognize the people only in 3 meter x 3 meter area. Since the application's progress path is very hierarchic, application can handle various suspicious movements. Only thing to handle various cases add a new suspicious movement in the database.

4.Data Model and Description

This section will give information about the data objects, their attributes and the complete data model.

4.1. Data Description

This section will give information about the data objects related to this

project, the relationship among them, the attributes of the data objects and the complete data model with data objects' functions included.

4.1.1. Data objects

The data objects are Admin, Movement, Tree, Log and Notification. Admin data objects has seven attributes called, AdminID, Name, Surname, Username, Password, Email and CellPhone. The second data object, Movement has seven attributes, namely MovementID, Name, Duration, FrameCount, MovementData, IsSuspicious and SecurityLevel. The third data object, Tree has two attributes, namely ParentMovement and ChildMovement. The fourth data object, Log has four attributes, namely LogID, AdminID, LoginDatetime, LogoutDateTime and NotificationCount. Finally, Notification data object has five attributes, namely NotificationID, AdminID, MovementID, DateTime and RecordPath.

4.1.2 Relationships

As explained in the previous section, we have basically five data objects which are namely Admin, Movement, Tree, Log and Notification. Admin and Notification data objects are in a relationship such that Admins monitor Notifications. Moreover, Tree and Movement are in a relationship such that Tree organizes Movements and creates parent-child relationships between movements. Lastly, Admin and Log are also in a relationship such that Log holds Admin session data. ER diagram of the data objects are presented in the following section.

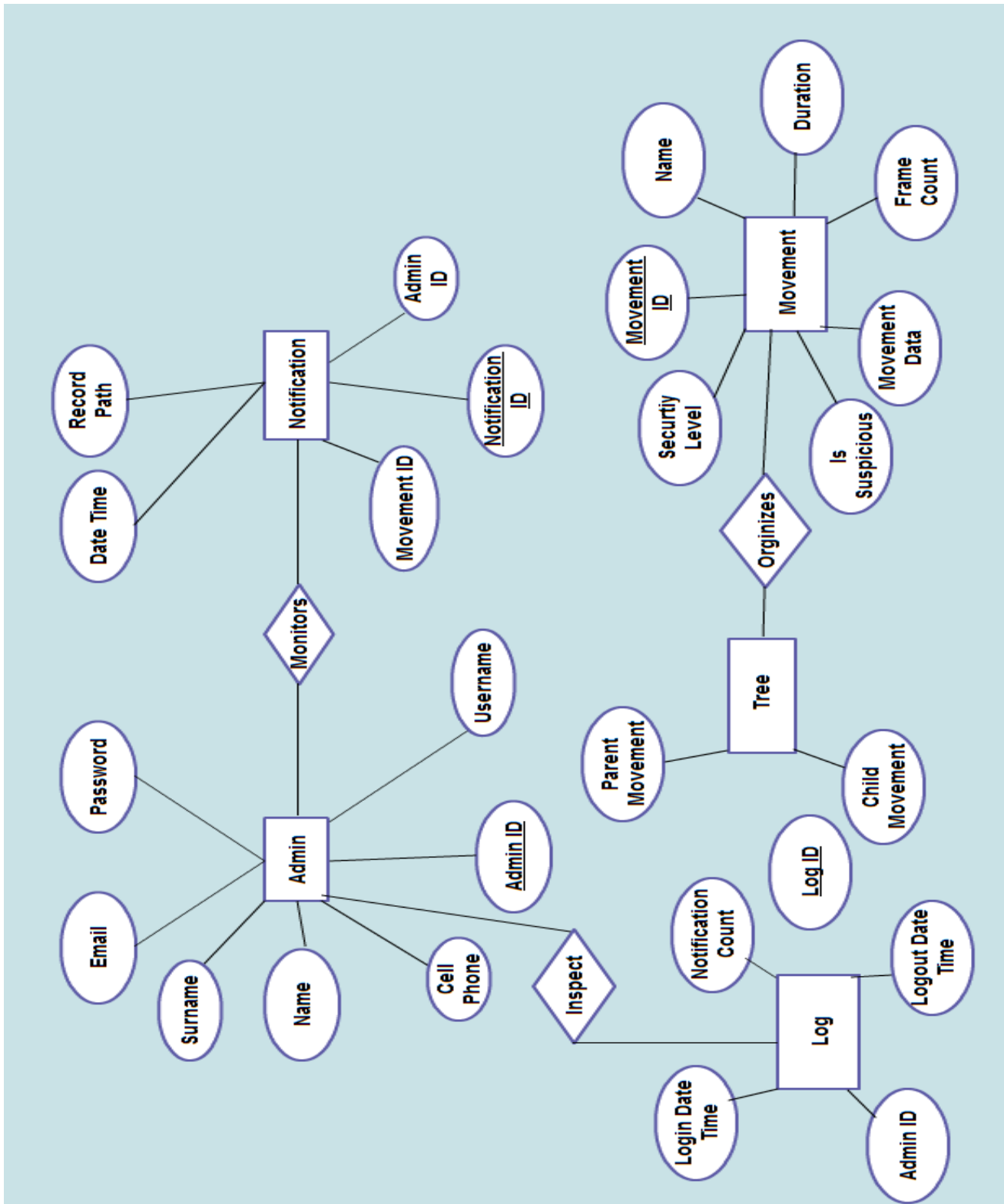


Figure 2 : ER diagram .

4.1.3 Complete Data Model

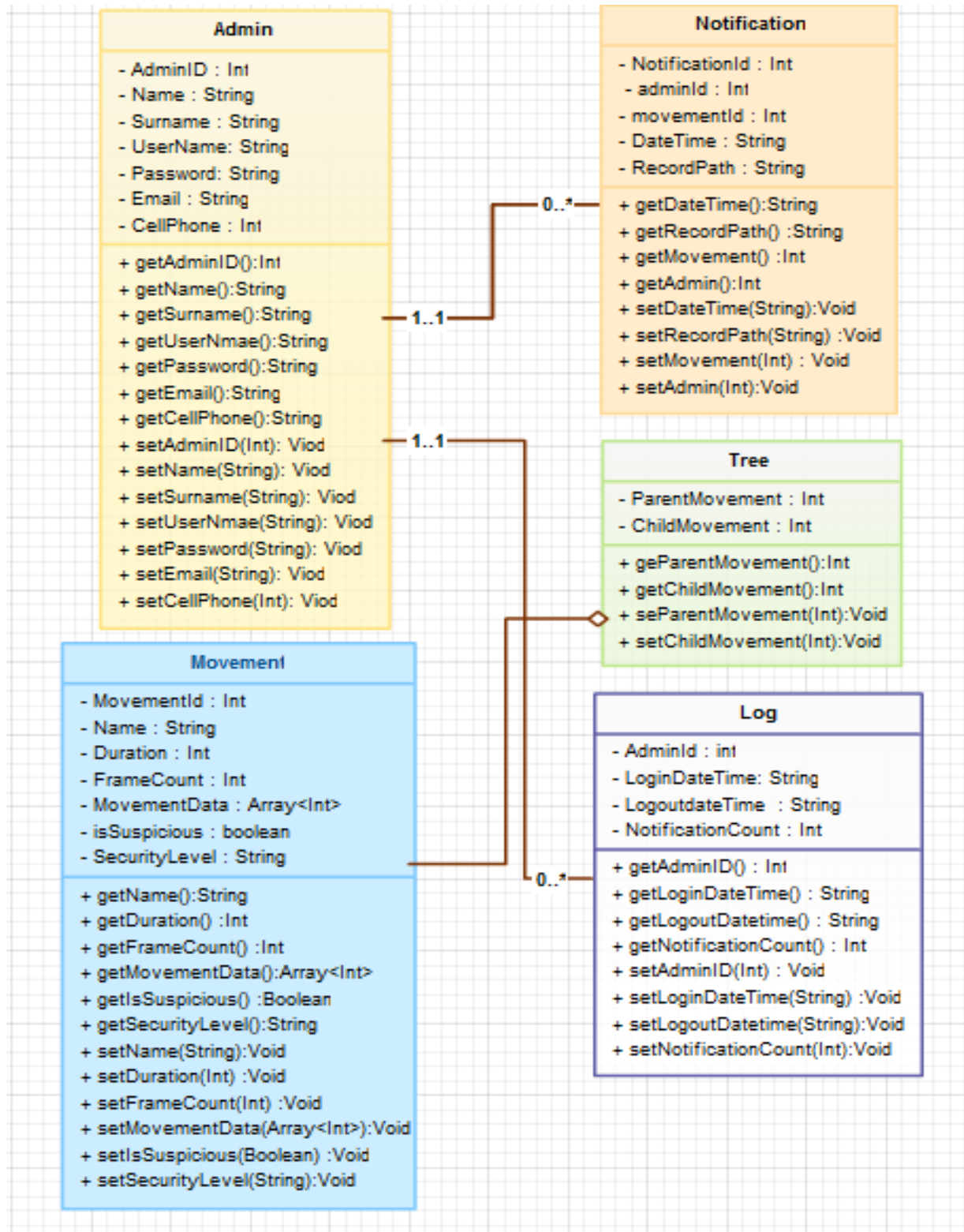


Figure 3: Class Diagram.

4.1.4 Data Dictionary

4.1.4.1. Admin

1. **AdminID:** This number is given to each admin.
2. **Name:** Name of the admin.
3. **Surname:** Surname of the admin.
4. **Username:** a unique username for each admin
5. **Password:** a 7-12 length secure word containing at least one number, one capital letter and one punctuation mark.
6. **Email:** email address of the admin.
7. **CellPhone:** Cell phone number of the admin.

4.1.4.2 .Movement

1. **MovementID:** This number is given to each movement already defined by the system.
2. **Name:** A unique name of the movement.
3. **Duration:** length of the recorded movement by means of seconds.
4. **FrameCount:** number of frames recorded in the specified duration.
5. **MovementData:** structural information of the movement for each frame
6. **IsSuspicious:** a Boolean value that specifies whether the movement suspicious or not
7. **SecurityLevel:** importance value of the movement (low, medium, high)

4.1.4.3. Tree

1. **ParentMovement:** id of the parent movement according to movements tree.
2. **ChildMovement:** id of the child movement that derived from its parent

4.1.4.4. Log

1. **AdminID:** the id of the admin.
2. **LoginDateTime:** login date and time of the admin.
3. **LogoutDateTime:** logout date and time of the admin.
4. **NotificationCount:** suspicious movement count occurring in that session duration.

4.1.4.5 Notification

1. **NotificationID:** notification id number.
2. **AdminID:** admin id that notification sent.
3. **MovementID:** movement id that occurred.
4. **DateTime:** notification date and time
5. **RecordPath:** the path of the 5mins length movie that notification occurred.

5. Behavioral Model and Description

5.1. Description for Software Behavior

The system consists of two main modes. These are namely "Defining Movements Mode" and "Security Monitoring Mode". These modes explained separately below.

5.1.1. Defining Movements Mode

In this mode, the purpose is to define the movements that can be percept as suspicious. Firstly, user performs a movement in front of the kinect to obtain skeleton position specific for that movement. This operation are repeated for 5 times to get more

precise result for detection process, as a result, the movement data explained in “Data Objects” section have been taken. After this stage is over, the other properties of the movement such as name, security level, etc, are entered using this mode. When a movement is recorded in that way, the system will be able to compare any other movements taken from the kinect, and calculates a similarity ratio.

5.1.2. Security Monitoring Mode

When the system starts, the security monitoring operation also starts. In this mode, every n seconds, the system takes a portion of video, for last m seconds ($m > n$). Then compares with the predefined movements that are recorded in the defining movements mode and calculates a similarity ratio. If this ratio is bigger than a threshold value, notifications are sent to the authorized people. While that comparison and calculation processes are being performed, taking a portion of video operation also continues.

5.2. State Transition Diagram

The state transition diagram shown below explains the overall behavior of the system:

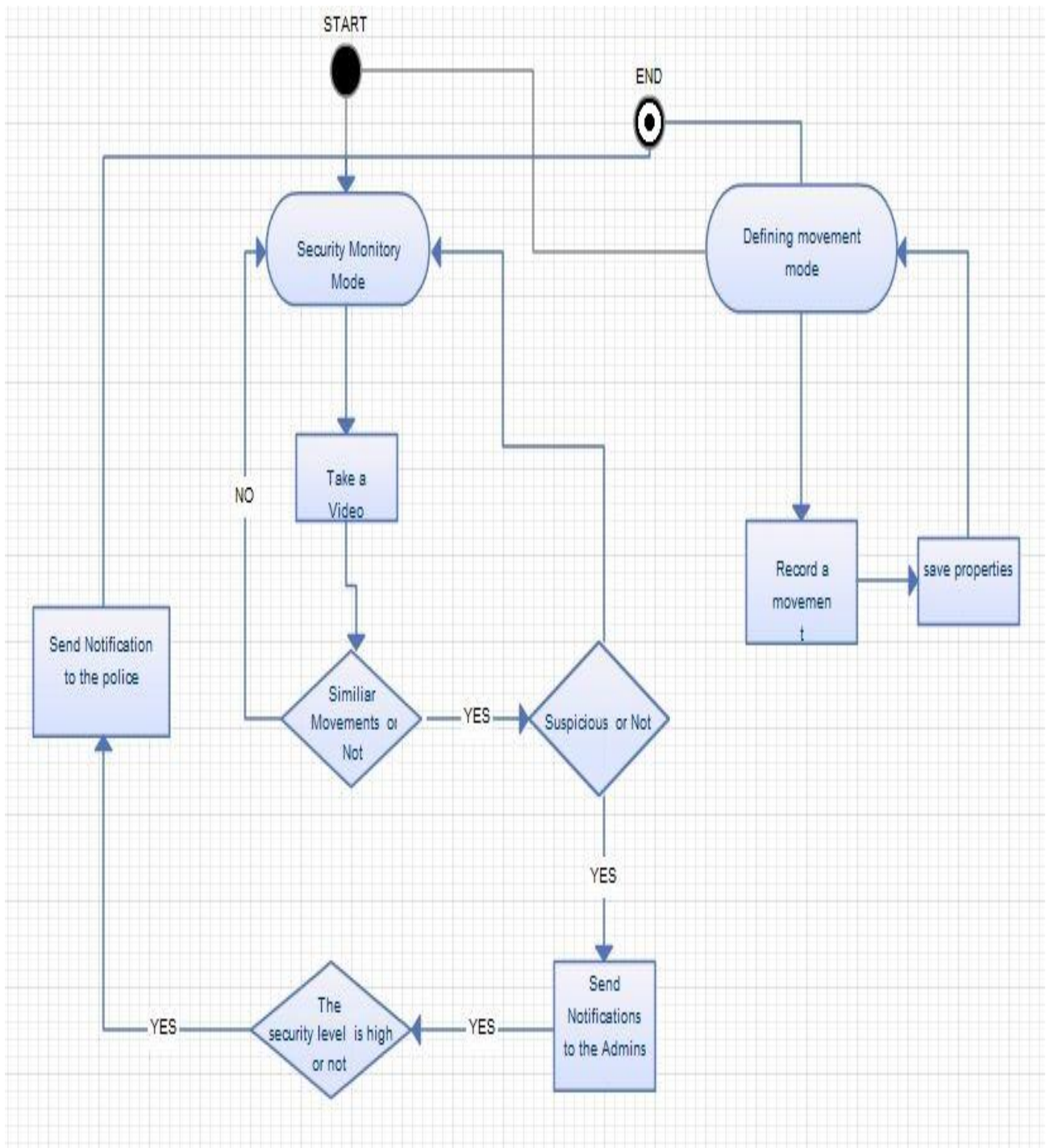


Figure 4 : State Diagram.

6. Planning

In this part of the document, the structure of the team responsible from the project, the basic schedule, and the process model will be presented.

6.1. Team Structure

This team's members have known each other for four years and have experience on division of labor since they have worked together on several projects previously. Moreover this team's members lived in same dormitory room whose name was 213. That's the explanation of our team's name. Every member's opinion is very important to each other ,since this team's members lived together, as mentioned before, this part was exceeded long time before. In addition to this, we can say that our team's governance type is democracy. In every step of our project, we consult each other about every issue and after the reaching agreement we rule the issue. At last we apply our ancient, secret and very powerful method, which is divide and conquer! We divide the workload and distribute them all each other equally, later every member of team begins to conquer their workload like a commander, slowly but efficiently. Another Information about us is that we work order of $n \log n$, namely, the best algorithm.

Erdoğan Kaya -> The Joker, administer.

İlhan Yoldaş Karabulut -> The Maestro, organizer.

Utku Şahin -> The Trigger, encourager.

Alişan Yılmaz -> The Algorithm, ultimate coder.

6.3. Process Model

After revising the process models, our team decided to The Spiral model. The spiral model is a software development process combining elements of both design and prototyping-in-stages, in an effort to combine advantages of top-down and bottom-up concepts. Also known as the spiral lifecycle model (or spiral development), it is a systems development method (SDM) used in information technology (IT). This model of development combines the features of the prototyping model and the waterfall model.

Thanks to the spiral model while we are creating the application, we can have prototypes of our application, therefore we can test our application easily. If an error occurs, we can go back with the help of prototypes. Moreover we can observe the progression of the application. Hence all of these features are provided by The Spiral Model.

To sum up we expect to obtain a prototype every life cycle, thus our application will be more reliable, more testable and includes less bug.

7. Conclusion

This software requirements specification document gives information about the project Security System with Microsoft Kinect. It starts with a definition of the real world problem and explains the solution that the project proposes. This explanations includes basically functionality of the system, interface requirements of the system, performance, attributes, and design constraints imposed on the implementation. In the overall description part, all of the functions that this system will perform is explained one by one. Data objects that are used in the system are modelled and behavioral models are presented. Finally, the team structure, basic expected schedule and the process model are presented.