1. Introduction:

Firstly, this document is prepared to provide a framework for the solution of the wireless-door security problem. In our vision, to define a framework for the solution, the software development team had to do following things before getting into actual technical development process:

- Defining the goal and scope of the project in a detailed and precise manner,
- Gathering as much as possible knowledge related to the project,
- Defining the product that has to be build,
- Specifying the product unambiguously and clearly,
- Modeling and Partitioning the elements of the product,
- Scheduling the order of events and distributing tasks among group members,
- Managing risks, by considering possible threats surrounding the project and possible solutions to them when they occur,
- Defining the resources needed to complete the project and acquiring them,

Consequently, in our conception the whole purpose of preparing this report is to accomplish these tasks and express their results in a clear, lucid way.

To achieve this goal, the analysis document is organized in the following structure:

- **Team Vision & Project Scope:** This section describes the problem that has to be solved and expresses the development team's vision and motivation about the project.
- **Fact Gathering:** In this section the problem domain is explored. Results of market researches and user surveys are represented to the reader. In addition, innovative ideas of group members, which can increase quality of the product and development process, are also included as a separate section.
- **Defining Requirements:** The features that are needed for the success of the project are listed with the suitable category and importance priority. Actually this is the section where "what has to be done "is represented.
- **Product Specification:** Product specification can be seen as a preliminary user manual of the end product. It is a representation of the core product for the end user and the customer. Once product specifications are agreed by both parties, this section can be seen as a contract between developer and customer.
- **Modeling:** This section is the analytical representation of the system. The information gathered on system during previous phases depicted in an analytical manner with the help of diagrams, graphs, and partitioning.
- **Appendix:** This section holds items that do not have direct relationship with the analysis of the project. Items such as Project Plan, Task Distribution, Risk Management, Needed Resources, and Project Estimates (Cost and Time) are presented at this section..

2. Team Vision & Project Scope:

2.1. Team Vision and Motivation:

Although through our university education we don't have much chance to work on computer hardware and develop embedded applications, it is a well known fact that nearly every milestone on computer industry is based on or ignited by the advances in computer hardware (Wireless Technology, USB, Fire wire, SATA Controllers, CPUs with Parallel Processing Capabilities , etc….).Therefore, we think that the experience and knowledge we are going to gain at the end of this hardware project would be very exciting and highly valuable for our future careers. In fact, this is the main reason why we have chosen this topic at the first place.

2.2. Problem Definition and Project Goal:

In our perspective, the essence of the Wireless Door Security problem is to implement a security system serving users at numerous door - controller pairs which are managed by a single master computer through a wireless network. In our vision, a system user, who is an owner of an e-card, can use the system by the following fashion:

1) System user passes his/her card in front of an electronic card reader at a legitimate door-controller checkpoint.

2) Then card reader then transfers appropriate data to the door controller for validation of his/her identity. After the validation, if the user's identity is valid then he/she is permitted to enter. Therefore, controllers can be seen as checkpoints at the doors looking for registered and permitted Ids, which can be managed through the master computer.

In this interaction, we consider the master computer as an interface between the system and administrator, which is the most important human factor of the system. Simply, administrator is the super user of the system, who is capable of viewing stored user activity, and setting the lock behavior for special situations (banning entrance, granting access, etc…).To sum up, all the functionality of the system is under the control of the administrator.

The figure depicted below can be seen as a blue print of the whole system:

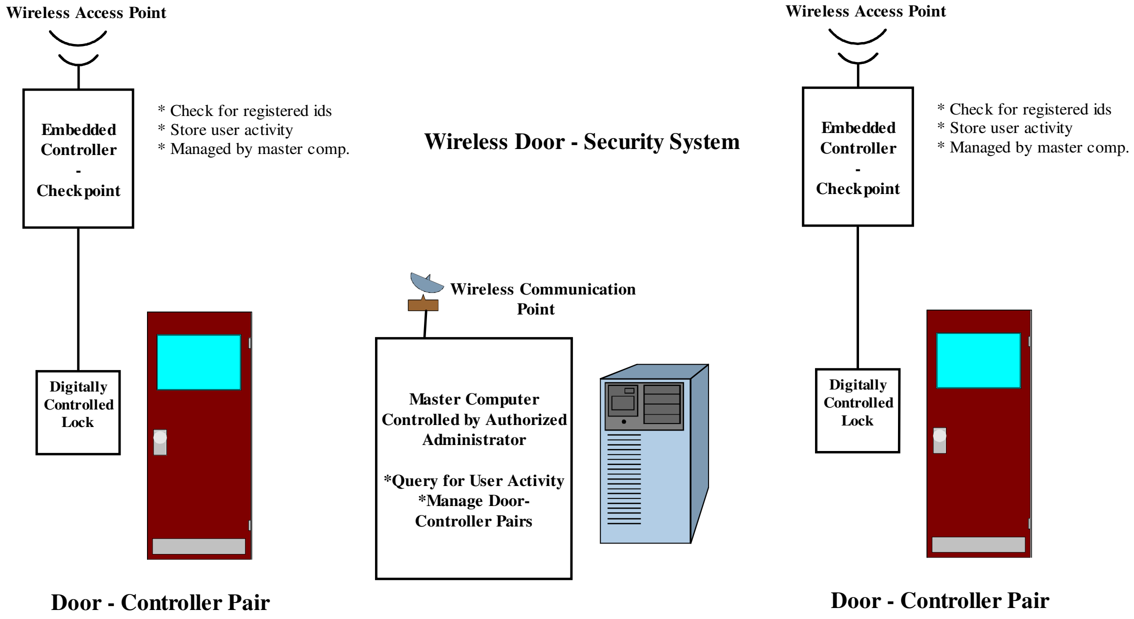

Wireless Access Point
Embedded Controller - Checkpoint
* Check for registered ids
* Store user activity
* Managed by master comp.
Digitally Controlled Lock
Door - Controller Pair
Wireless Door - Security System
Wireless Communication Point
Master Computer Controlled by Authorized Administrator
*Query for User Activity
*Manage Door-Controller Pairs
Wireless Access Point
Embedded Controller - Checkpoint
* Check for registered ids
* Store user activity
* Managed by master comp.
Digitally Controlled Lock
Door - Controller Pair

Figure – 1 Blueprint of the Door – Security System

In addition to these basic features, the performance and reliability of the system is critical since we are building a security appliance. The final product has to be highly reliable, may be even fault tolerant. To achieve this, we will use a basic principle; simplifying the features and removing unnecessary modules. In other words, we are going to use KISS principle throughout the project to make system reliable.

Apart from that, we don't see the problem merely as building the software/hardware which is capable of functioning the way customers want. We think the solution to the problem is building the product with the required features using an efficient process so that the end product would be simple to install, use and support.

To sum up, our eventual goal is to build a robust and simple system that solves the problem, while satisfying customer, end user and developer simultaneously.

2.3. Major Constraints:

There are three major constraints that the development team had to cope with:

- **Current ambiguity and unavailability of the hardware equipment:** Actually, this is not big problem for now, but it can be serious if this situation is prolonged to the next semester. This fact makes hard for team to foresee the future status of the project.
- **Restrictions imposed on the use of hardware equipment**: Imposed restrictions limits the number of available technology to use, therefore the development team had to invest carefully before adapting a new technology.
- **Limited monetary resources to buy spare hardware equipment for testing and system backup:** This fact can slow down the development process. Therefore development team had to find innovative solutions to overcome this problem.
- **Limited number of hardware:** This is not only a constraint but also a serious threat for the project in the case of hardware failure.

3. Fact Gathering:

3.1. Market Research:

Today, security needs of commercial companies, industrial organizations, military bases and state units are very high due to large number of people interacting with these organizations. Therefore to check and record people moving in and out of their premises, many organizations assign electronic cards to the legitimate users and manage them through computer systems consisting of electronic card readers, embedded controllers, smart cards, etc…

When organizations are installing such systems they expect a high level of protection, even when the threat of crime is very low. Therefore, in order to stay in competition, our system must offer the safest solution with the proper price.

Common applications of such security systems can be listed as follows:

- Library Security, Museum Security
- High-rise Office, Building Security
- Campus and Dorm Security
- Hospital and Airport Security.

Considering the previously mentioned facts, we can list the main expectations of customers from door security systems as following:

- System should have Anti-pass back property. In other words ,without passing the main door, the other doors should not open to the user.
- Each door should be programmable on its own to control entrance permissions.
- Entrance permissions at the doors should be manageable by time scheduling.
- System should include an alarm mechanism to prevent information loss due to limited storage. For example, if there is less space in event storage in master computer system should give warning message with sound. Also to prevent a possible information loss., system may self-lock itself temporarily when the storage for the user-activity is full,
- Integration with fire alarm and the system which is regarded as self lock opener system
- Even when the master computer is closed there must be a mechanism that stores the user-activity to the memory.
- There should be a mechanism for reporting and archiving of permitted passes.
- There should be a mechanism for reporting and archiving of failed attempts.
- There should be a mechanism for reporting and archiving according to card readers.
- There should be a mechanism for reporting and archiving according to departments.
- There should be a mechanism for reporting system's technical status.
- System should be capable of integrating to C.C.T.V. (Closed Circuit TV)
- System should have a comprehensive reporting facility providing statistical information.

3.2. Customer Surveys:

Meetings with possible customers arranged to have a better conception about the problem, and requirements for the ideal system. These meetings especially helped us while we were identifying the requirements of the system. The reader can find the outline of these customer meetings in the following sections.

3.2.1. Meeting with Cumhur Büyükakıncı:

Context-Free Questions (process oriented)

Q: Who is the customer?
A: Cumhur Büyükakıncı E-mail: cbakinci@polinas.com.tr Tel: 0236 233 04 70

Q: Who is the user?
A All of the white – collar and blue – collar staff of the company he manages.

Q: Are their needs different?
A: Almost same

Q: What is his current professional position?
A: He is a 54 years old businessman, who graduated from METU Chemical Engineering Department at 1972. Currently, he is the CEO of POLİNAS Plastics Factory at Manisa /TÜRKİYE

Q: How many people are working in the factory? How many doors had to be integrated to the system?
A: 350 people are currently working, and we have 6 important entrance doors.

Q: What is the importance of a successful door security system for your company?
A: Our factory will gain a better security and order.

Q: Have you ever used any electronic security system in factory?
A: No, we haven't.

Q: How much development time do we have?
A: At most 7 months

Q: How much budget are you planning to allocate for this system?
A: We have not decided yet.

Q: Do you want any guarantee?
A: We want full guarantee until we end using the system.

Q: Do you want training for your personnel?
A: Absolutely.

Context-Free Questions (product oriented)

Q: Why do you want a door security system for factory?
A: First of all, factory needs high security due to product formalization. Secondly he wants to control and inspect the staff. For example, he wants to collect information about when and where employees pass in and out.

Q: What are your expectations on usability, reliability and performance?
A: He expects staff to be comfortable with the system after training. And in performance oriented view, he wishes a perfect system.

Q: Do you want to interact with the system in Turkish?
A: Yes, interaction had to be in Turkish, since most of the staff does not know English.

Q: Would you like to integrate the system with another system in the factory?
A: It is a good idea; it can work with alarm system at the same time.

Q: Do you want any extras?
A. I think, system should also report failed attempts.

Context-Free Questions (meta-questions)

Q: Am I asking too many questions?
A: No, I think you're questions are ideal for the first meeting.

Q: Did my questions seem relevant to the subject?
A: Yes, most of the time.

Q: Are you the right person to answer these questions.
A: Definitely, I am exactly the right person to answer these questions.

Q: To assure we understand each other, can I write down the answers to these questions and give you a written copy to study and approve?
A: Yes, sure you can.

Q: Is there someone else who can also give me useful information?
A: My partner Ali Evren Özçam can be helpful too.

Q: Is there anything else that I should also ask to you?
A: I think, you're questions are enough for the first meeting.

Q: Is there anything you want to ask me?
A: So far there is no, but I want to ask that whether you're going to give this product as a present?

Q: Can I ask more questions later?
A: Of course you can.

3.2.2. Meeting with Adnan Seyik(businessman, factory director):

The results of this meeting are listed below;

- Especially he wants a classification of identity cards for different staff members.
- He wants a system which can be improvable through time.
- He expects a reasonable price

3.3. Innovative Ideas:

At this section we provide ideas that can improve the quality of the product and software development process.

Nr.	**Ideas on Development Process**
1	To reduce risks and avoid procrastination we schedule deadlines of the milestones earlier and behave according to new deadline. This attitude provides us extra time to enhance the work we’ve done, in addition to mitigating the impact of the possible problems that can happen during development cycle.
Nr.	**Ideas on Product**
1	Using smart cards instead of magnetic cards to store information on user cards
2	Integrating video cameras to the system for visual archiving
3	Integrating digital photograph machines for visual archiving
4	Integrating an electronic siren to the system
5	Integrating the system with other security systems
6	Integrating the system with police, hospital, firemen hotlines
7	Interfacing small LCD screens to the electronic equipment, to make diagnosing easier.
8	Interfacing a wireless keypad to the door-controller to make debugging easier.

4. Defining Requirements:

In this section, we will list the features, suggested to be included in the possible final product. We partitioned requirements according to the five different categories:

- Functional Requirements: Features related to the system functions
- User Interface Requirements: Features related to user – system interaction
- Maintenance Requirements: Features related to troubleshooting and user support
- Installation Requirements: Features relating to the setup of system
- Validation Criteria: Features that have to be fulfilled in order to have a working system.

Each category can have its own partitions depending on its complexity. Priority values range from 1 to 5, representing increasing priority.

4.1 Functional Requirements:

Nr.	Functional Requirements Related to the System User	Priority
1	When a valid and permitted user passes his card in front of the card reader system should respond to him within reasonable amount of time, and open the door.	5
2	Card Reader should read the cards easily and instantly	5
3	System should behave consistently(pass, block) against cards unless there is change made by the administrator	5

Nr.	Functional Requirements Related to the Administrator	Priority
1	Administrator should have the opportunity to modify the entrance permissions of all the registered users at all the doors at the system. In other words, entrance permissions of users will depend on the door he is trying to enter, which is controlled by the administrator	5
2	Administrator should have the opportunity to group the users for controlling group access	4
3	Administrator should have the opportunity to save permission tables for chunks of users and load them to use previous configurations	3
4	Administrator should have the opportunity to schedule the master computer to change permissions on a specific date and time	3
5	Administrator should have the opportunity to save, view and query user activity at specific doors.	4
6	Administrator should have the opportunity to intervene the system operation at the door controllers with limited functionality.	3

Nr.	General Functional Requirements	Priority
1	System can be programmed to open or close the doors on specific events (Fire Alarm, Theft ...etc).	3
2	System can be programmed to connect to an alarm service on specific events(Theft Alarm ->Police, Fire Alarm-> Firemen)	2
3	System can work when master computer is off	5
4	Door Controllers should give reasonable amount of output while they are working, to check for malfunction. For example, a LCD screen that gives information about the current situation of the system would be informative for both customer and developer.	3

4.2. User Interface Requirements:

Nr.	User Interface Requirements	Priority
1	There has to be a card reader which is self – expressive to use. In other words user can easily understand where to pass the card.	5
2	Door Controller Systems should have proper output devices (LEDs, LCD screens) that give information about the current status of the device.	3
3	Door Controller Systems should have mountable input devices to control the operation and troubleshoot problems.	2
4	There has to be a GUI (Graphical User Interface) on the Master Computer to manage door security system.	5
5	User interface should allow the administrator to use wild cards, and numeric bounds while querying the system, to maximize the functionality.	3
6	All the user interface elements have to be self expressive to ease the experience of gaining skills on the usage of the system.	5
7	All the user interface elements have to be designed in such way that, the steps to complete an instruction should be minimized.	5
8	All the user interface elements and devices on the system have to be attractive and decorative.	4
9	Physical elements should have the proper physical appearance so that they should look fine when they are mounted on a wall or on doors.	4

4.3. Maintenance & Support Requirements:

Nr.	Maintenance & Support Requirements	Priority
1	There must be a user manual of the system that clearly defines the system's functions and how-to use them. Moreover user should be supplied with a technical troubleshooting guide to solve problems.	5
2	An On-line Tech support service should be provided	3
3	A backup utility that performs whole system backup should be included in the final product.	3
4	System should able to work during an electrical power failure	4
5	Door-Controller devices should work with the possible minimum energy	2
6	System should be designed so that it lets the user & developer to troubleshoot in case of a malfunction.	3
7	The software at the door controller and the master computer should be possible to update for patches or new features.	3
8	System should contain some type of a fault tolerance mechanism to work under a hardware failure.	2
9	System should notify the administrator about the failures and abuses against the system.	2

4.4. Installation Feature Requirements:

Nr.	Installation Feature Requirements	Priority
1	From customer’s perspective, system elements, both hardware and software had to be trivial to install and run.	4
2	Customer should be provided with a proper manual that describes installation process clearly.	4

4.5. Validation Criteria:

Nr.	Validation Criteria	Priority
1	System should respond users with a reasonable amount of time when they pass their cards in front of the card reader. Values that are lower than 1.5 seconds would be acceptable.	5
2	System should almost never fail under normal circumstances	5
3	System shouldn't be vulnerable to attacks. It must stand against both physical and software attacks towards the system.	5
4	System should be simple in every aspect.	4
5	System operation shouldn't be scrambled by another system. In other words, it must be closed against inputs coming out of the system.	5

5. Product Specifications:

At this section we provide details on how the system interacts with the user and how it behaves under normal operation. This part can be seen as a preliminary user manual although we are in early stage of the project. Specifications about functional and user interface elements of the system are presented at the following sections. Meanwhile, reader should notice that these specifications are tentative and they may and possibly change through time as the project progress.

5.1. Functional Specifications:

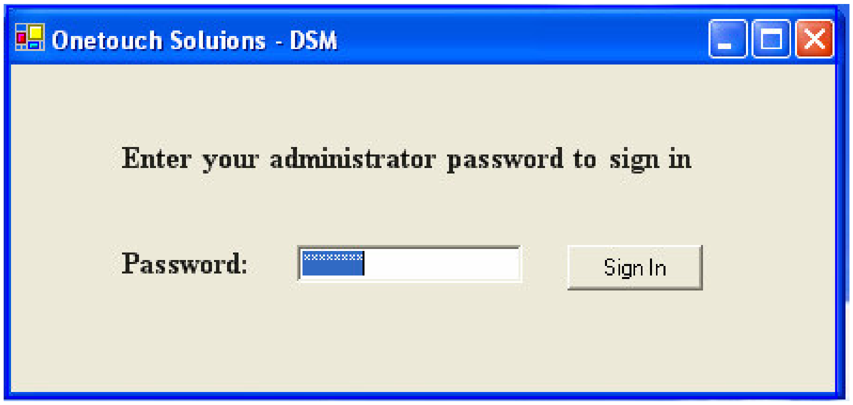
5.1.1. Functional Specifications Related to System User (Card Owner):

A system user has only one option while using the door-security system; passing his card. Then his/her identity is checked by the door controller, to see whether the id is valid or not. Afterwards, if his/her ID is valid then he/she is permitted to enter by the door controller.

5.1.2. Functional Specifications Related to Administrator while using Master Computer

5.1.2.1. Logging In: To login, first administrator has to power up the Master-Computer and run the DSM (Door-Security Manager). Then at the opening screen of the DSM there will be a prompt asking for the administrator password. In order to login, administrator had to type his password and click sign in to resume.

Figure – 2 depicts the possible entrance screen of Door Security Manager:



Enter your administrator password to sign in

Password: xxxxxxxx

Figure - 2

5.1.2.2. Opening a Door Security Project:

Door Security Project is an entity that holds or provides links to all information about a door security management. Configuration. It provides an easy and organized way to handle system backups and different configurations. A new door security project can be created using File->New->Door Security Project, meanwhile a previous can be loaded by its previous settings by File->Open-> Door Security Project Interface.

5.1.2.3. Adding Doors to the Security Project

After creating or opening a door security project, adding doors to the system are done through Doors->Add Doors interface, which lists door controllers in an accessible area with their respective Door IDs. The administrator can select doors he wants to control by selecting them and adding it to the door panel. Selected doors will show up with previously selected doors in the doors panel. Afterwards in the doors panel he can replace the Door ID with a meaningful name. Finally, he can manage and edit the door through doors panel.

5.1.2.4. Removing Doors from the Security Project:

Doors that are previously added on the doors panel can be removed by selecting them and clicking "Remove" button at the doors panel.

5.1.2.5. Managing Doors:

The entrance permissions of the users at the doors are controlled by assignment of permission tables. Permission tables are entities that hold information about user entrance permissions. Assignment is made by first selecting a door and the desired permission table. Afterwards, clicking assign button at the doors panel is enough to complete action. Notice that, assignment of one permission table to multiple doors is possible but conversely, assignment of one door to multiple permission tables is not possible.

5.1.2.6. Opening Permission Tables:

A permission table can be added to an existing door security project either using Permission T.->New interface to add a new table, or using Permission T ->Open interface to load an existing one.

5.1.2.7. Editing Permission Tables:

A permission table can be edited by adding or removing users or user groups. Each entry is permitted to enter the system within respective time bounds that are presented at the table. In this approach, undefined time bounds in the table mean permanent permission to enter the system. Apart from that, administrator can query the permission table to check whether a specified user or group is permitted to enter or not. Moreover, in same sense, administrator can conduct searches among the registered users and groups to add or remove desired items easily.

5.1.2.8. Managing Permission Tables:

Administrator can save, open or assign the permission table. Saving the permission table can be achieved through either saving the whole security project or saving only permission table by using right mouse click + save action. Opening a permission table can be performed by either creating a new table or loading a previously saved one by using Permission Table->Open Interface. The information on assignment of door-security pairs can be found at section 5.1.2.5.

5.1.2.9. Managing Users:

Administrator can add, remove, or edit the users. Users can be added to the project from User-> Add interface by specifying Name, Surname, Group and etc… fields. Moreover, users can be edited and removed by either finding them using User->Search interface or selecting them from permission table.

5.1.2.10. Managing Groups:

In a straightforward manner groups are set of users that can be managed together. In this sense it is possible to add or remove them to/from permission tables, or delete them from whole database using appropriate user interface elements.

5.1.2.11. Loading Security Project:

After saving changes made to project, these changes can be activated by clicking “Load Project” button. Afterwards, the software will make necessary changes on the system elements and verify the administrator about the status of update. Notice that the system can be loaded with only one Door Security Project at a given time.

5.1.2.12. Querying Security Project Archives:

Administrator can query user activity archives for statistical information (passes, failures, abuses etc…) and save the results on a file or print them on paper. Also, the system can provide predefined statistical information to the administrator using “stats” interface. Finally, to use this facility effectively, user has to update the archives in reasonable amount of intervals by clicking the “Archive Update” button.

5.1.2.13. Locking and Opening All The Doors In The System:

Administrator has the opportunity to permanently lock or open the doors included security system project. He/she can lock/unlock all the doors by using lock and unlock all the doors buttons located at the doors panel. In the same sense, she/he can lock/unlock some specific group of doors by first selecting them at the doors list and then clicking to lock/unlock door button to make the desired change. Notice that the doors that are not listed in the doors panel are permanently open.

5.2. User Interface Specifications:

5.2.1. GUI (Graphical User Interface) of Door Security Manager:

Figure - 3 represents a prototype for the user interface of door security manager ,which is capable of performing the functions described at the section 5.1.Functional Specifications.

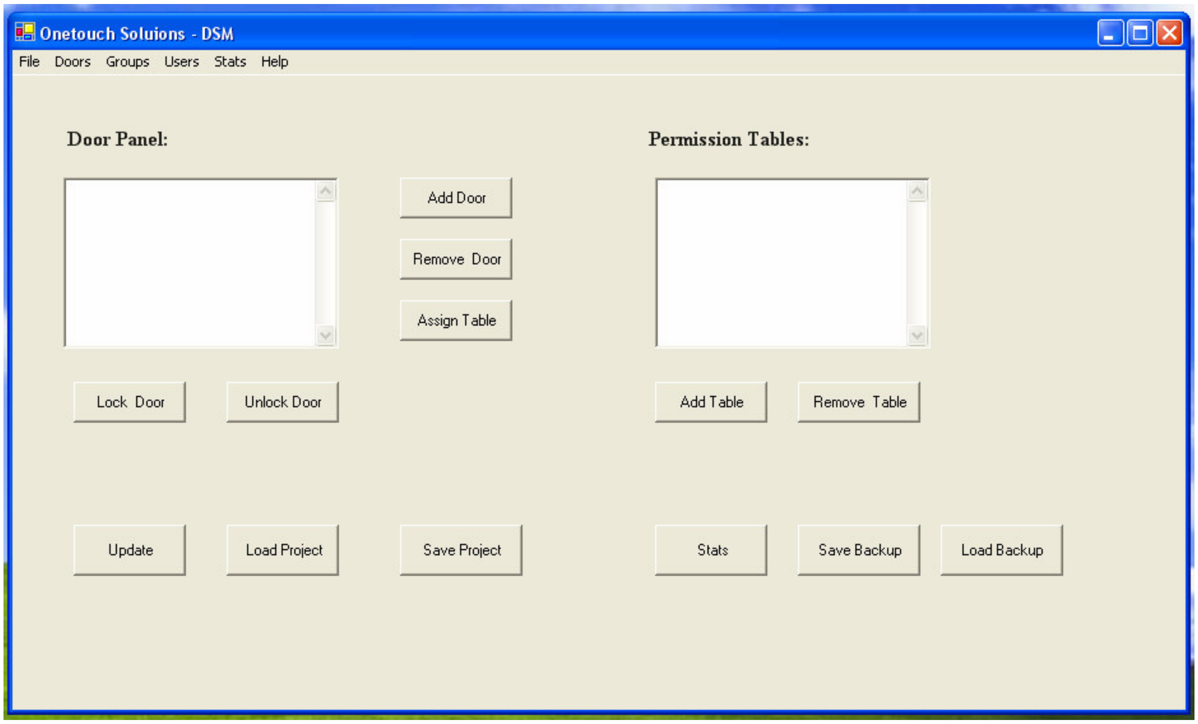


Figure-3

6. Modeling:

6.1. Introduction:

To give a schematic representation of the system and show dynamics, modeling of the system environment is a must. Through the developer's perspective , model of the system is the most helpful element of the analysis The method we preferred to create the model of the system is structured analysis method and the models and diagrams we have build are listed below:

- Data Objects,
- Entity Relationship Diagram,
- Data Flow Diagram,
- State Transition Diagram,

The following sections provide related charts and diagrams of these models. Moreover at the end of the section there will be a part devoted to the data dictionary to provide an organized approach for representing the characteristics of each data object and data mentioned in diagrams.

6.2. Data Objects:

We used following notation to represent data objects:

Name: <Attribute> <Attribute> <Attribute>
Comment

Administrator: <Password>	**Group:** <Title> <Information>
Administrator of the system.	Set of users which are created by the administrator with a specific purpose.
Door: <Door ID> <User Defined Name> <Permission Table>	**Card:** <ID> <Card Owner>
A door representation in a door-security project. Door-Controller is also a viable name for a door.	An identity card belongs to the user.
User: <Name> <Surname> <ID> <Group Memberships> <Address> <Phone>	
User registered to system. Card owner.	

Card Reader:	**Permission Record:** <User> <Start Date- Time> <End Date – Time>
A sensor that reads the identity card.	A permission record that belongs to a user. It permits entrance of the specified user during the defined time limits.
Permission Table: <Title> <Information>	**User Activity:** <User> <Action> <Door-Controller> <Date-Time>
Holds permission records. Created by administrator.	Previous user activity, stored by the system.
User Activity Table: <Title> <Information> <Date – Time>	**Security Project**: <Title> <Information>
A user activity table that stores previous User-activity. Created by administrator through querying the system archives.	Holds all information about the security project.

6.3. Entity Relationship Diagram

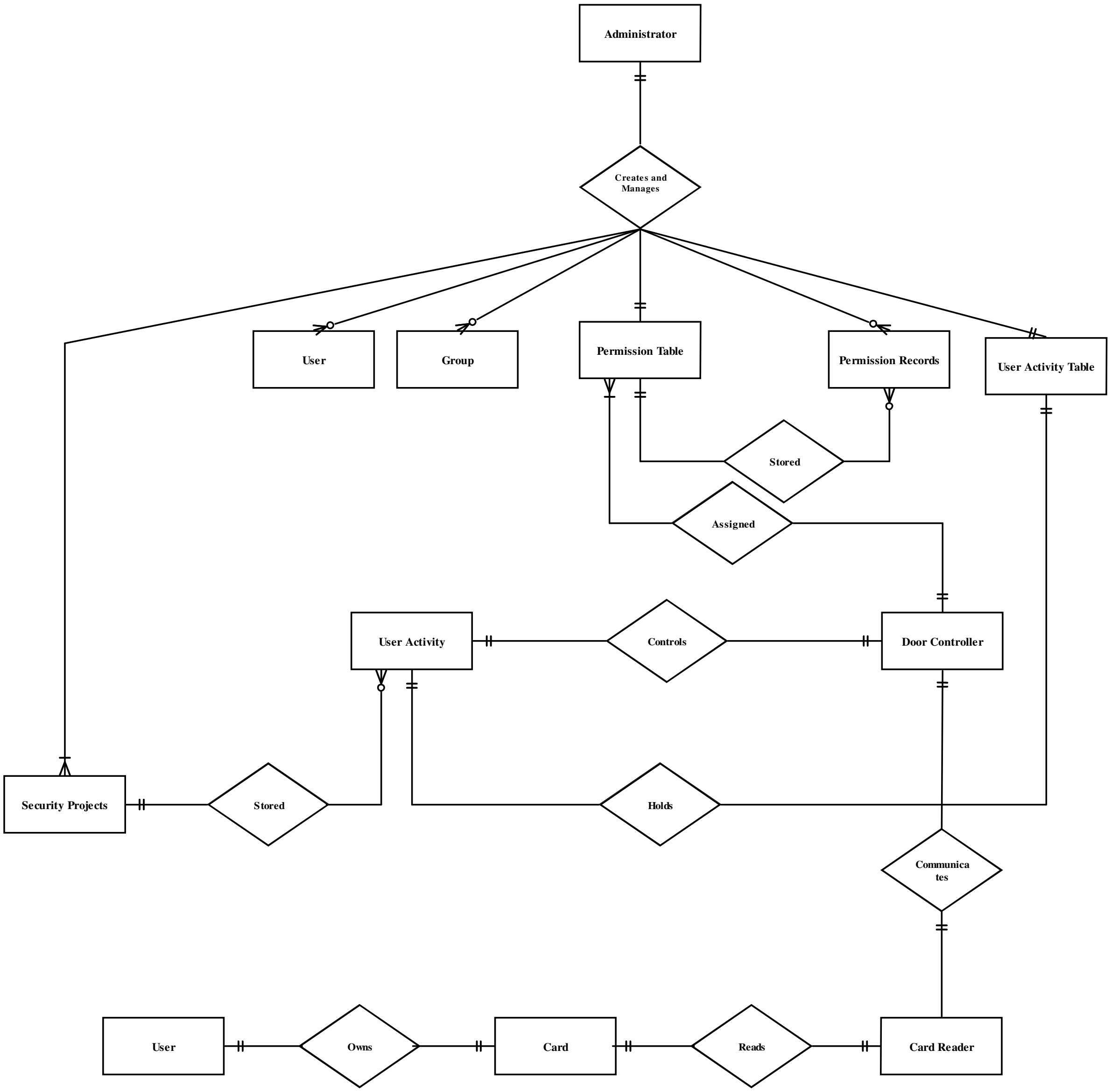
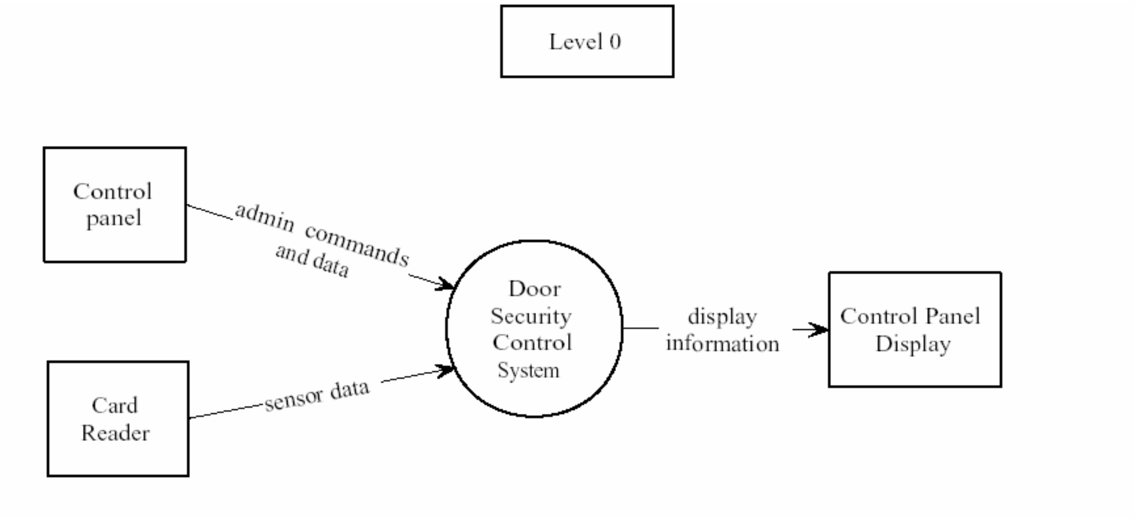
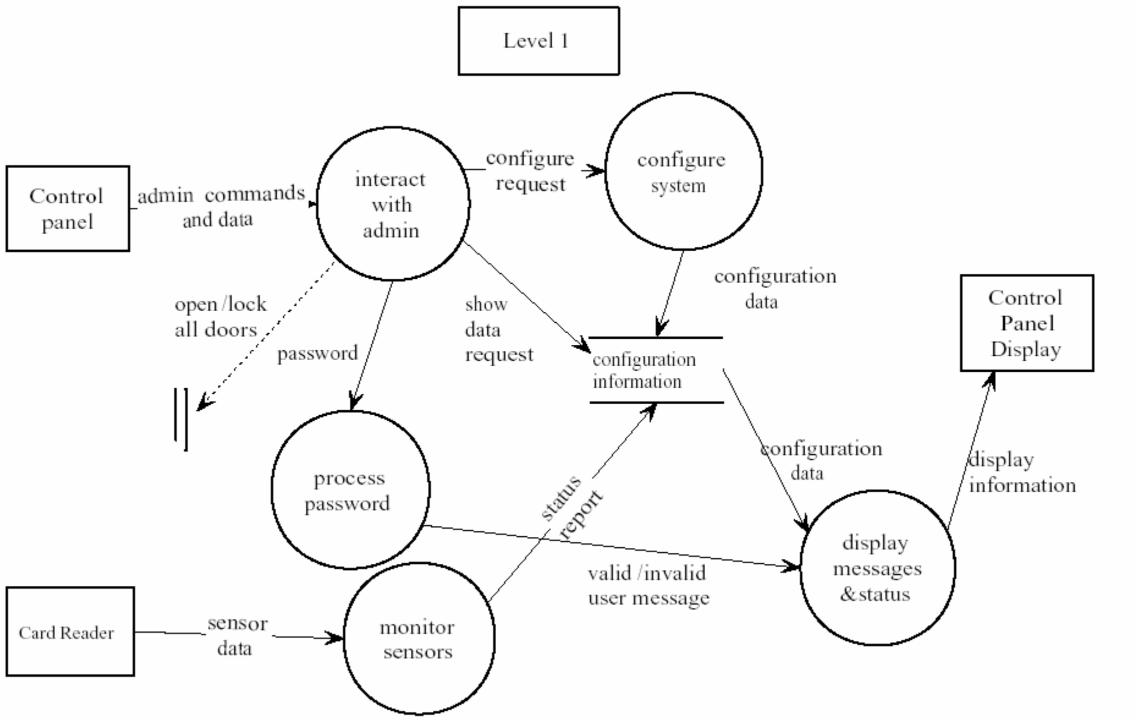
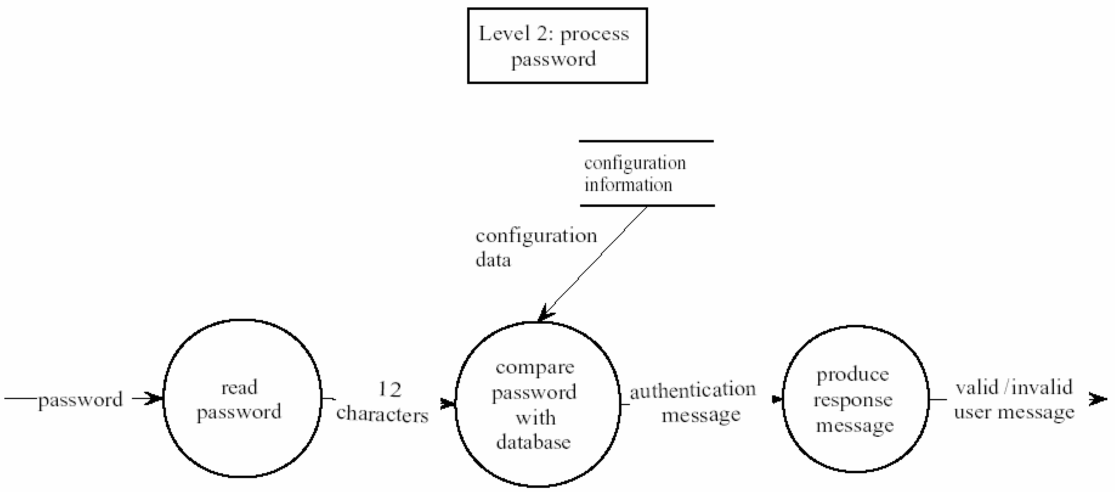


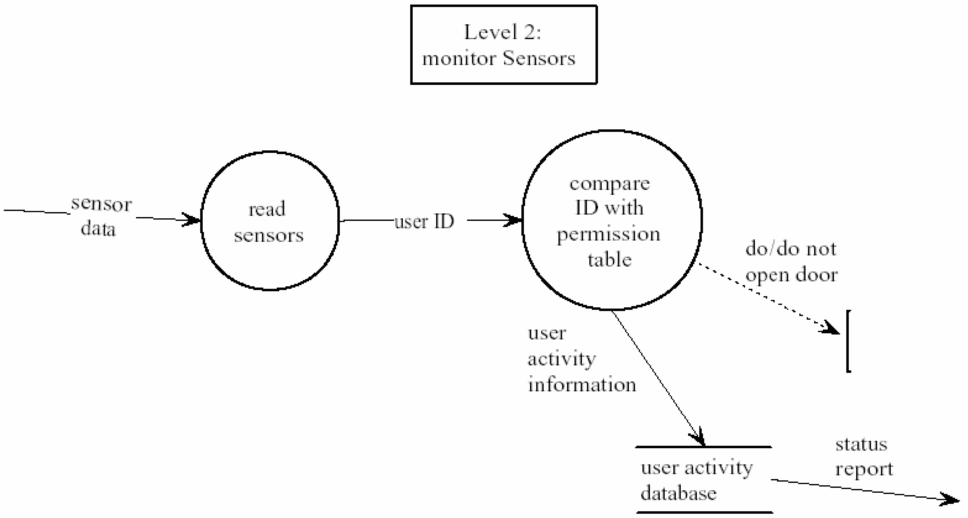
Figure – 5

6.4. Data Flow Diagrams:

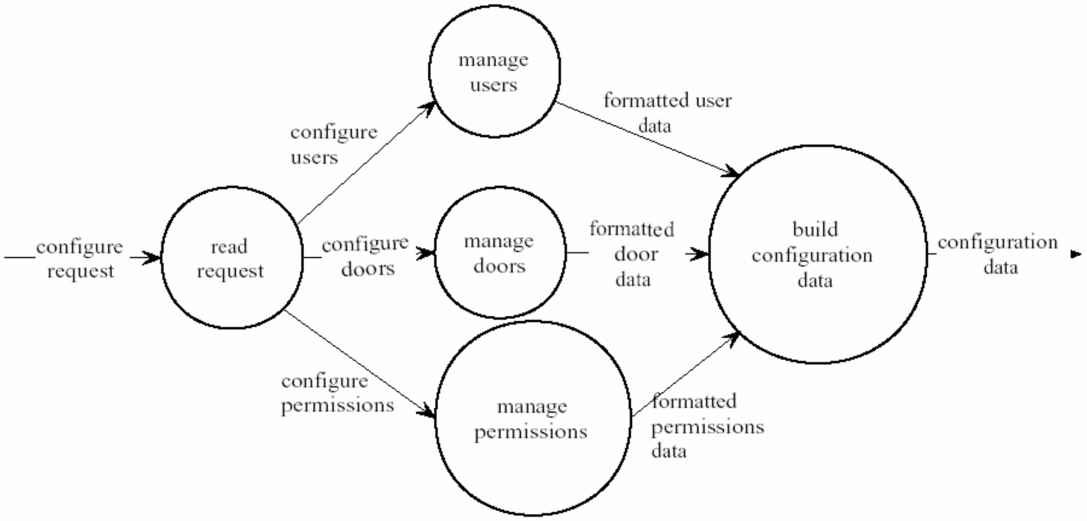




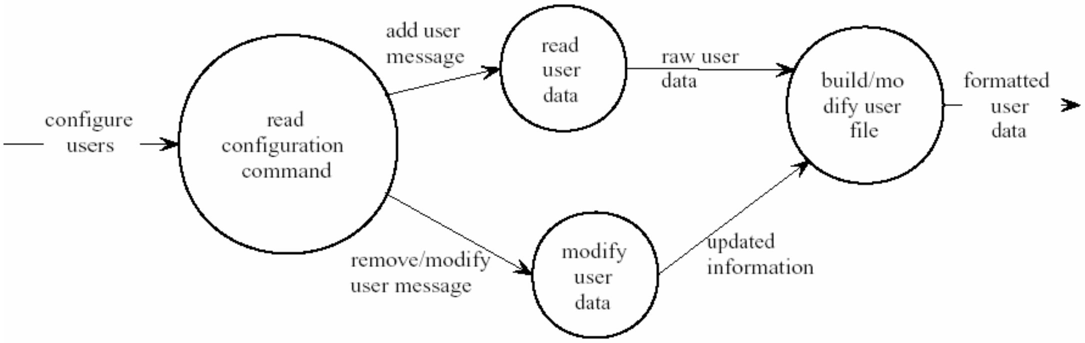

Level 2: process password
configuration information
configuration data
password
read password
12 characters
compare password with database
authentication message
produce response message
valid / invalid user message


Level 2: monitor Sensors
sensor data
read sensors
user ID
compare ID with permission table
do/do not open door
user activity information
user activity database
status report

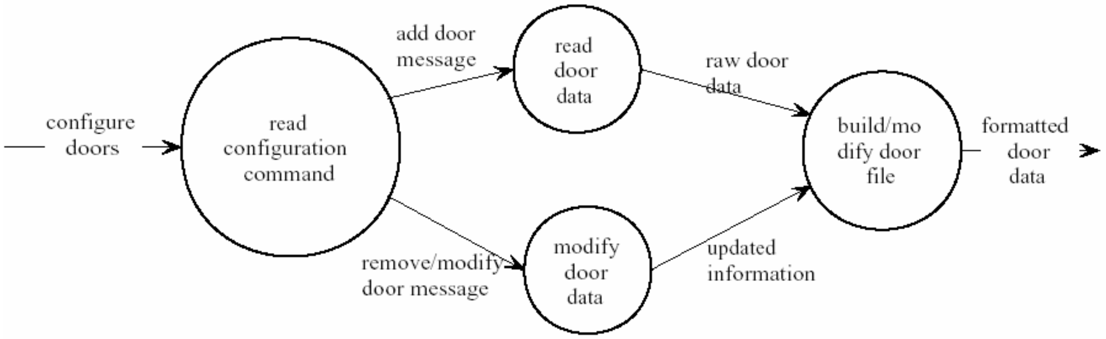
Level 2:
configure system



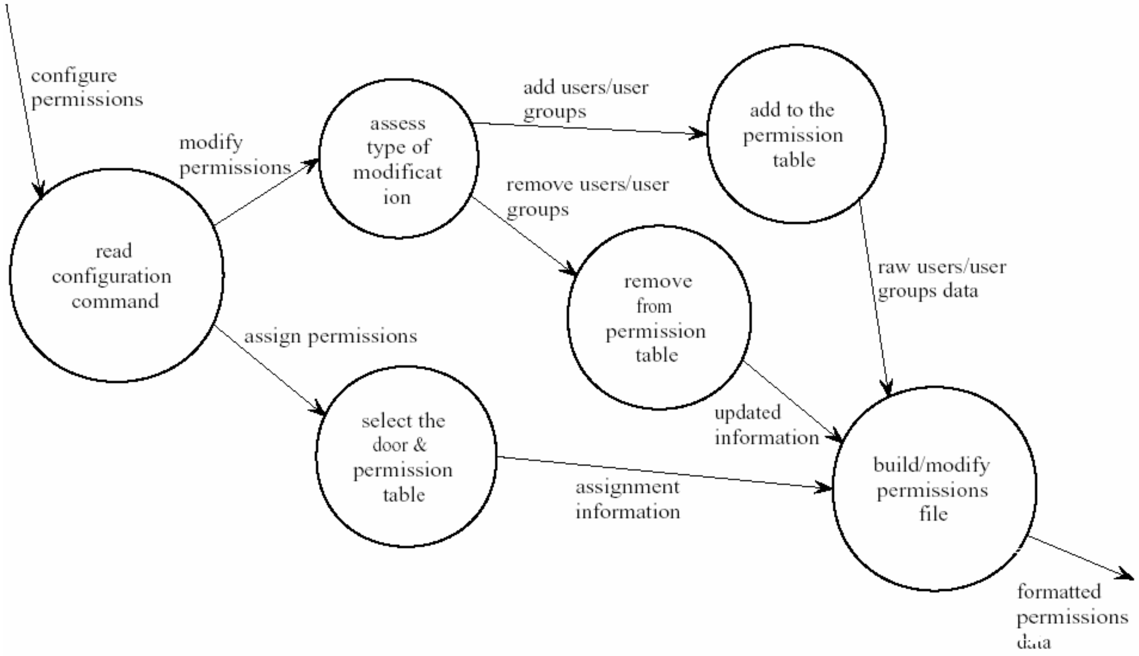
Level 3:
manage users



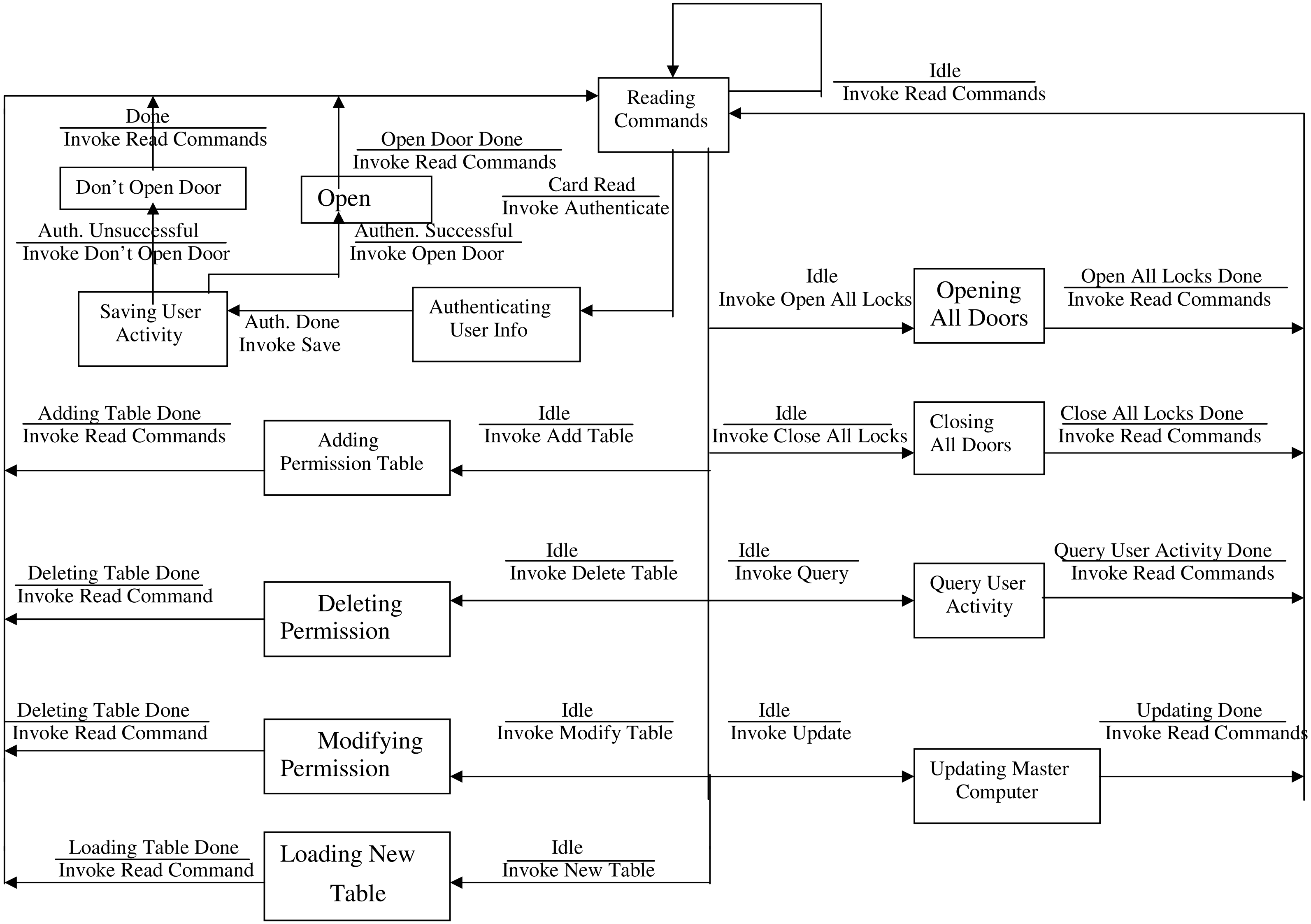
Level 3:
manage doors



Level 3: manage
permissions


configure permissions
read configuration command
modify permissions
assess type of modification
add users/user groups
add to the permission table
remove users/user groups
remove from permission table
raw users/user groups data
assign permissions
select the door & permission table
updated information
assignment information
build/modify permissions file
formatted permissions data

6.5. State Transition Diagram:



6.6. Data Dictionary:

Using the classical notation to represent its elements, in this section we represent data dictionary of the project.

Item: Name

Aliases: Title,Surname,User Defined Name

Where Used/How Used: Group(Attribute),User(Attribute),Door(Attribute),Permission Table(Attribute),User Activity Table(Attribute),Security Project(Attribute),

Description: A 50 character alphanumeric string.

Item: Information

Aliases:-None-

Where Used/How Used: Group(Attribute),Permission Table(Attribute),User Activity Table(Attribute),Security Project(Attribute)

Description: *A variable length alphanumeric string*

Item: Password

Aliases:-None-

Where Used/How Used: Administrator(Attribute)

Description:*12 character alphanumeric string*

Item: Door ID

Aliases:-None-

Where Used/ How Used: Door(Attribute)

Description:*A 32-bit integer*

Item: Card Owner

Aliases:-None-

Where Used/How Used: Card(Attribute)

Description: A link to data object User.

Item: ID

Aliases: User ID

Where Used/How Used: Card(Attribute),User(Attribute)

Description:*A 32-bit integer*

Item: Group Memberships

Aliases:-None-

Where Used/How Used: User Attribute

Description: A linked list consists of variable number nodes which hold group names as keys.

Item: Address

Aliases:-None-

Where Used/How Used: User(Attribute)

Description:*A 160 character alphanumeric string*

Item: Phone

Aliases: Phone Number

Where Used/How Used: User(Attribute)

Description:

Phone =[Local Number | National Number | International Number | Mobile Number]

Local Number = *7 digit number that does not start with zero*

National Number = *10 digit number that does not start with zero*

International Number = *13 digit number*

Mobile Number = *11 digit number starting with a zero*

Item: Date-Time

Aliases: Start Date-Time, End Date-Time,

Where Used/How Used: User Activity(Attribute),User Activity Table(Attribute),Permission Record(Attribute)

Description:

Date-Time = Date + ‘_’ + Time

Date = *Date in DD/MM/YY format *

Time = *Time in HH(0-24) :: MM(0-60) :: SS(0-60) format*

Item: Action

Aliases:-None-

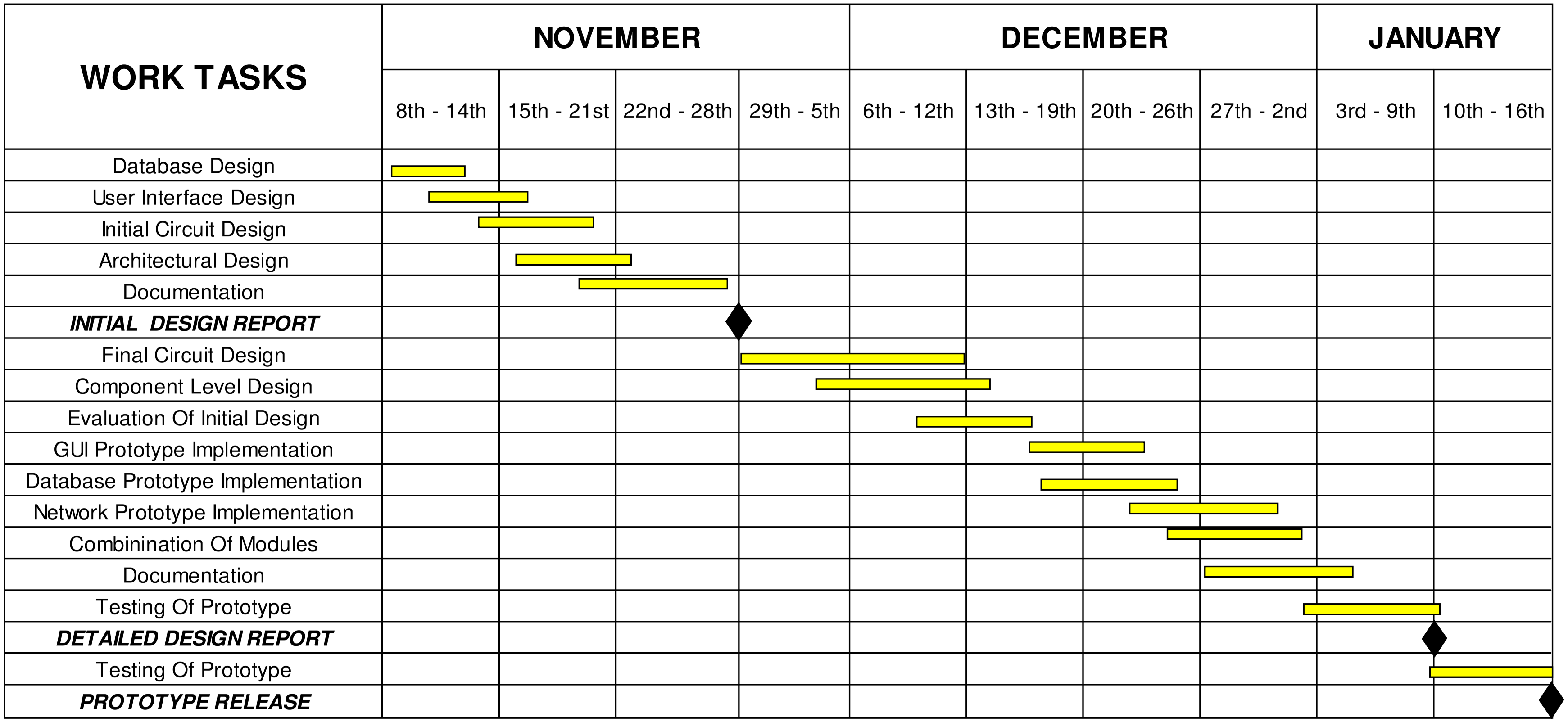
Where Used/How Used: User Activity(Attribute)

Description:[Permitted to Enter | Failed to Enter]

APPENDIX:

1. Project Plan
2. Task Distribution
3. Resources Needed to Complete Project
4. Risk Management
5. Project Estimates

1. Project Plan



2. Task Distribution:

TASKS	Volkan Onur Gursoy	Emin Tolga Akgoz	Doguhan Dikmen	Swaleh Kavuma
Database Design	√		√	
User Interface Design	√		√	
Initial Circuit Design		√		√
Architectural Design	√	√	√	√
Final Circuit Design		√		√
GUI Prototype Implementation		√		√
Database Prototype Implementation	√		√	
Network Prototype Implementation		√		√
Combination Of Modules	√	√	√	√
Testing Of Prototype	√	√	√	√
Documentation	√	√	√	√

3. Risk Management:

3.1. Possible Risks

Risk No.	Category	Risks	Probability	Impact
1	Employee Risk	Temporary unavailability of team member	20%	3
2	Employee Risk	One of the team members drops the course	10%	2
3	Technological Risk	Lack of technical skills	30%	2
4	Employee Risk	Under estimating workload	30%	2
5	Business Risk	Tight delivery deadline	20%	2
6	Development Risk	Hardware failure	10%	1
7	Development Risk	Information Loss	10%	1
8	Development Risk	Absence of hardware	40%	1
9	Technology risk	Investing in wrong technology	40%	2

Impact Values	**Description**
1	Catastrophic
2	Critical
3	Marginal
4	Negligible

3.2. RISK MANAGEMENT STRATEGIES

Nr.	Strategy
1	1) Have more than one person working on each part of the Project to ensure continuity if a team member is unavailable. 2) Dispersing knowledge of technical details of each part of the Project across the team.
2	Well prepared documents explaining everything in detail and putting extra working hours to be ahead of Schedule.
3	We will create a knowledge base with different resources related to the Project e.g GUI programming, Serial Port Communications, Embedded Linux, and Network Programming.
4	Weekly meetings to monitor progress. Each member is given a duty to complete before next meeting to ensure systematic progress as planned.
5	We plan to put in extra working hours during the semester and during the semester break i.e winter holiday.
6	We intend to have a through understanding of the hardware specifications e.g voltage required e.t.c before using it.
7	Every document and source code file will be backed up to avoid losing any information which might lead to collapse of the Project.
8	If for any reason the hardware part of the Project is not available we intend to use software simulation to show the functionality of our product.
9	Doing a thorough research on the Technologies used for products similar to our Project should help avoid this risk.

4. Resources Needed To Complete Project

4.1. Skills Needed to Complete Project:

According to our analysis of the Project we concluded that we will require the following skills:

Digital Circuit Design Skills: This is required to design the hardware that will convert the input from the card sensor to a form that can be understood by the computer’s RS-232 serial communication port.
Serial Interfacing Skills: These skills are required to program the communication between the embedded system and the Wigand converter.
Network Programming Skills: This skill is required to program the network modules (mainly C++) of the Door Security Manager and the embedded system.
Database Programming Skills: This skill is required to program the database that will store the user information.
GUI Programming Skills: This skill is required to program the user interface of the Door Security Manager software.

NB: To require the resources required to learn the required skills we will mainly use resources from the school library and from the internet.

4.2. Hardware Resources Needed to Complete Project:

- Embedded System components provided by the department,
- Required IC chips,resistors,cables,and electronic development tools such as multimeter,and soldering equipment.
- A desktop computer with a wireless NIC capable of running master computer software.
- An electronic lock

4.3.Software Resources Needed to Complete Project(Tentative):

Name:	Usage:
Linux	Development Environment and Embedded OS
C/C++	Programming Language
.NET	GUI and Database Programming
GTK	GUI Programming
XML	Organized file structure for data storage
MYSQL	Database Management System
SQL	Query Language
Windows XP	Operating System
MS Word	Word Processor
SmartDraw	Drawing Tool
Rational Rose	CASE Tool
Electronics Workbench	Digital and Analog Circuit simulator
Shell Programming	For Automating activities at linux environment

5.Project Estimates:

5.1. Estimation Techniques And Results

Two estimation techniques were used:

- Functions Points (FP) → Albrecht and Goffney Model
- Lines of Code (LOC) → Intermediate COCOMO

	Weighting Factor				**FP Count**
Measurement Parameter	**Count**	**Simple**	**Average**	**Complex**	
Number of user inputs	5	3	**4**	6	20
Number of user outputs	5	4	**5**	7	25
Number of files	3	7	**10**	15	30
Number of external interfaces	2	5	7	**10**	20
Count Total					95

Factor (F_i)	**Value**
1- Backup and recovery	4
2- Data communications	5
3- Distributed processing	2
4- Performance critical	5
5- Existing operating environment	3
6- On-line data entry	2
7- Input transaction over multiple screens	2
8- Master files updated on-line	4
9- Information domain values complex	2
10- Internal processing complex	3
11- Code designed for reuse	2
12- Conversation/Installation in design	1
13- Multiple installations	1
14- Application designed for change	4
$\sum$ (F_i)	**40**

$FP = \text{count total} * [\ 0.65 + 0.01 * \Sigma(F_i)]$
$= 95 * [\ 0.65 + 0.01 * 40\ ]$
$= 95 * [\ 1.0\ 4\ ]$
$= 99.75$

$LOC = FP * 64 = 99.75 * 64$
$LOC = 6384$
The estimates for LOC are plugged into the COCOMO formula for effort and duration estimation. The basic COCOMO model is used, for which

- Effort $E = a\ KLOC^{\ b}$
- Duration $D = c\ E^{\ d}$

The project is classified as an organic project, using default values a = 2.4, b = 1.05, c = 2.5 and d = 0.38.

$E = 2.4(KLOC)^{1.05}$
$E = 2.4(\ 6.4\ )^{1.05}$
$E = 2.4 * 7.022$
$E = 16.8528$ person months

$D = 2.5E^{0.38}$
$= 2.5(16.8528)^{0,38}$
$= 2,5 * 2,93$
$= 7.325$ months

Comment*:* From the result of the cost estimate above we can see that our project will take about 7.325 months. The number of people required to complete the project will be 16.8528 / 7.325 = 2.296 ≅ 3 people. To offset the workload required to learn the technologies required for our project, we have one more person in the team i.e a 4 member team.

5.2. Cost Estimation

We have estimated the total cost of the project as follows:

Item Details	**Unit Price**	**Number**	**Total**
Identity Card Sensors	75 $	2	150$
Circuit Elements for wiegand converter			180$
RS-232 Interface	3$	1	3$
Wireless Ethernet Card	45$	2	90$
Embedded Computer	200$	1	200$
Total			**623$**

The total price of the required devices is **623$.**