

CHAPTER 3

METHODOLOGY

3.1 Introduction

Figure 3.1 shows the methodology or steps involved in this project. The analysis from the literature review shows the motion detector can be used as detector in the nursery. Problem statement state that the nursery is not protected enough by security system that will cause the baby is in risky situation even they are in the nursery. The baby sitter cannot keep an eye on the baby at all the time in the nursery especially when there have several numbers of babies.

The Baby Monitoring System in Nursery is the solution that will provides security for the babies in the nursery. Here, three phases are required to accomplish the system. The three phases are hardware, software and mechanical design.

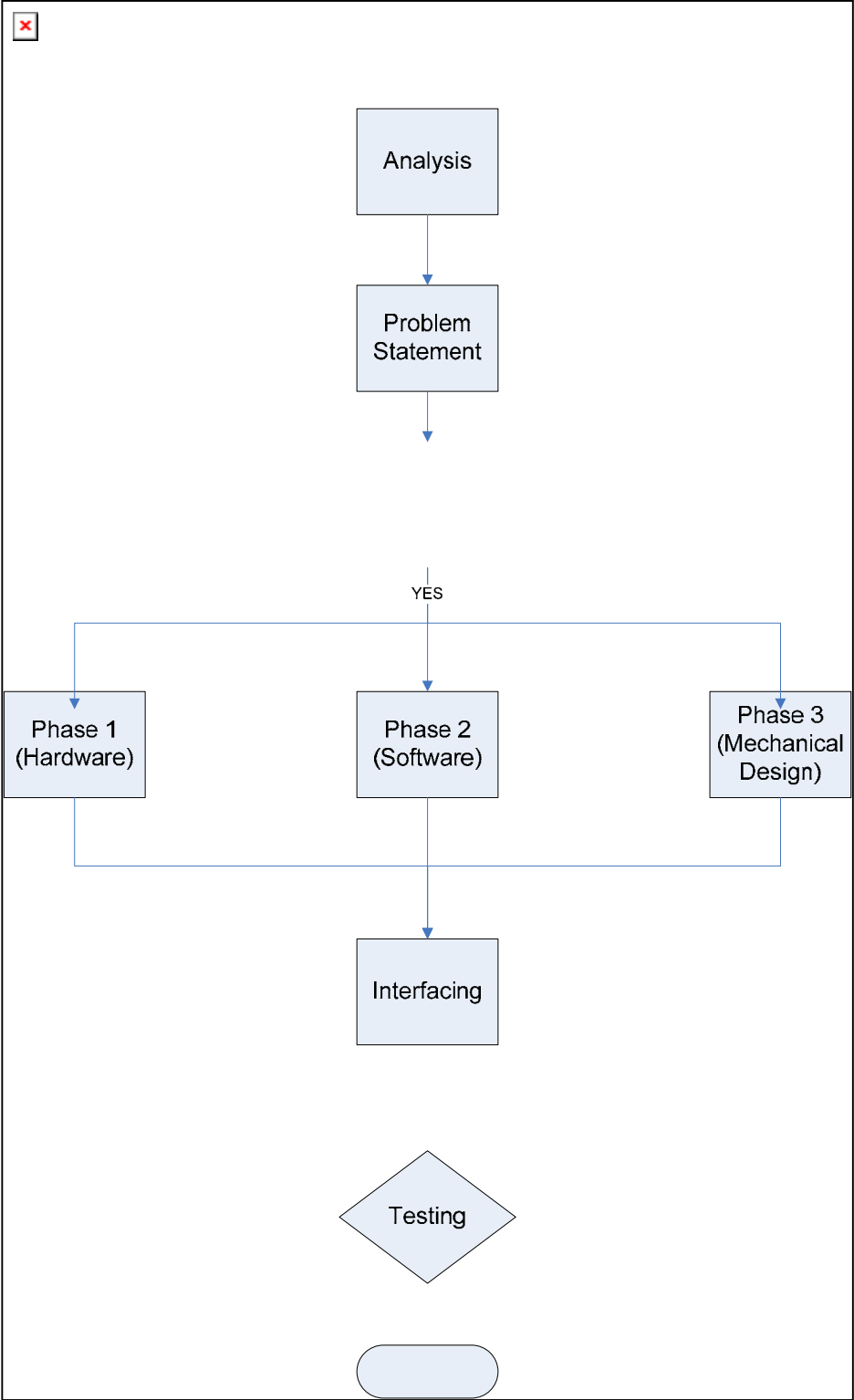


Figure 3.1 : Methodology

3.2 Phase 1 (Hardware)

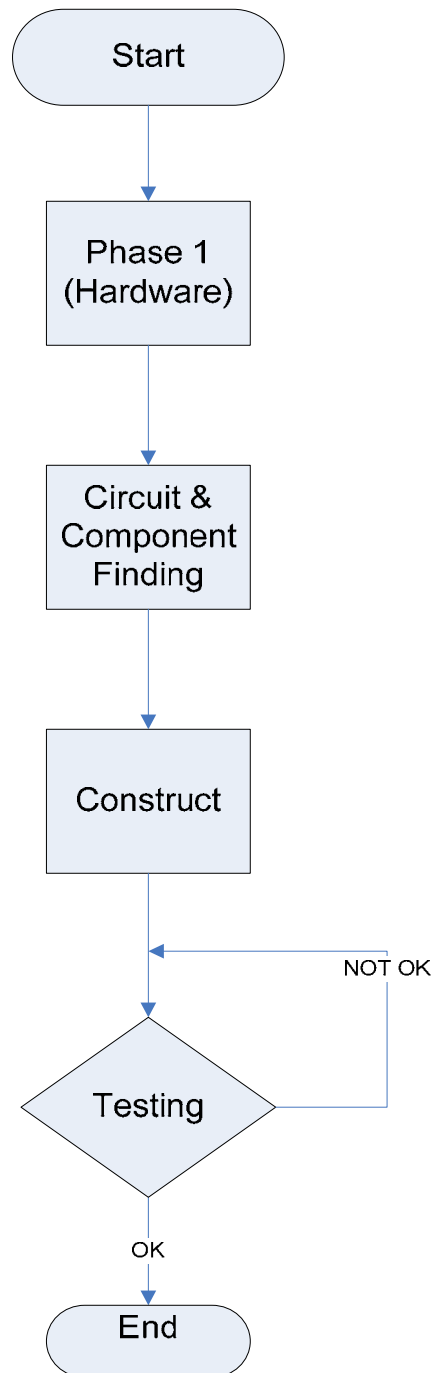


Figure 3.2 : Phase 1

Figure 3.2 shows the phase 1 steps in this project. The first step is to get the component for the circuit that found in the internet. The entire component for the circuit is shown in the Appendix A. When the components are ready, the circuit is constructed on the project board to test whether the circuit is fully functioning.

3.2.1 IR Motion Detector Circuit

Figure 3.3 shows the circuit that used in the project. The circuit is test after constructed on the project board. The circuit is fully functioning. The problem is the output of the circuit produces more than 5V when the circuit is at logic 1 (high). This will be a problem when the circuit connects with the computer by using the parallel port. The computer can just receive 5V maximum if not the computer will cause a problem.

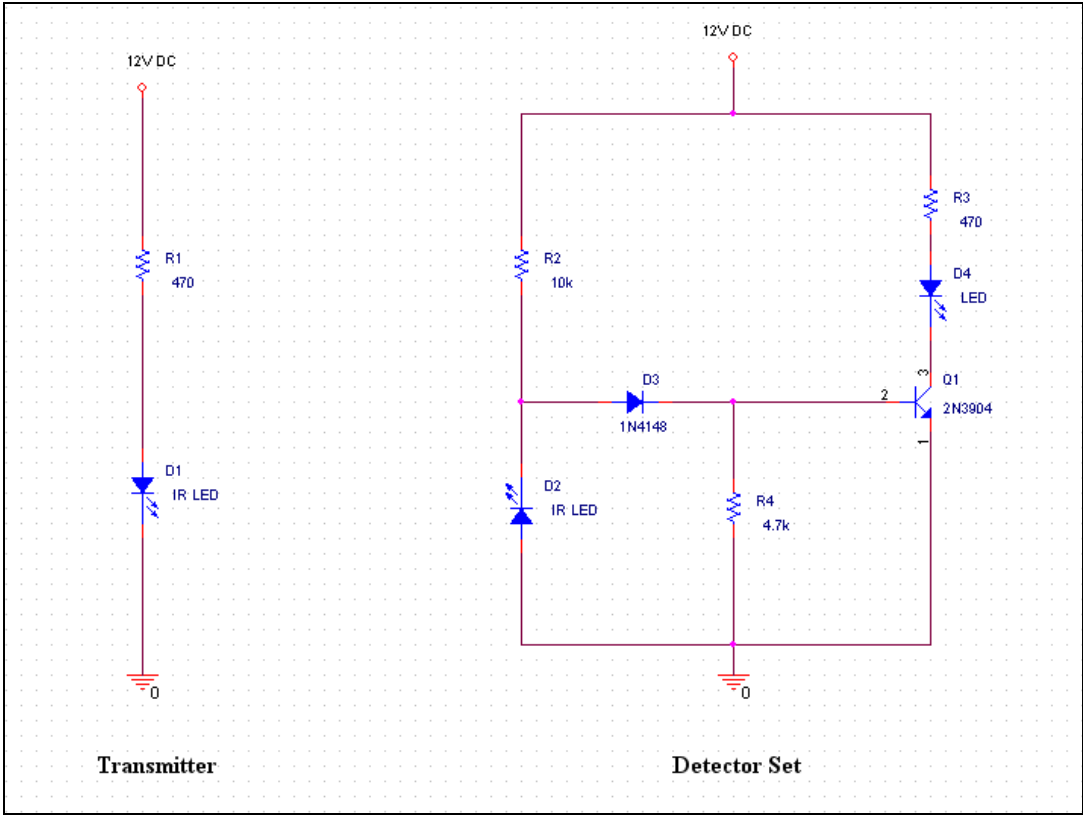


Figure 3.3: IR Motion Detector Circuit

So, the circuit needs to modify to reduce the output voltage. Figure 3.4 shows the circuit that has been modified. This is by applying the voltage divider at the output.

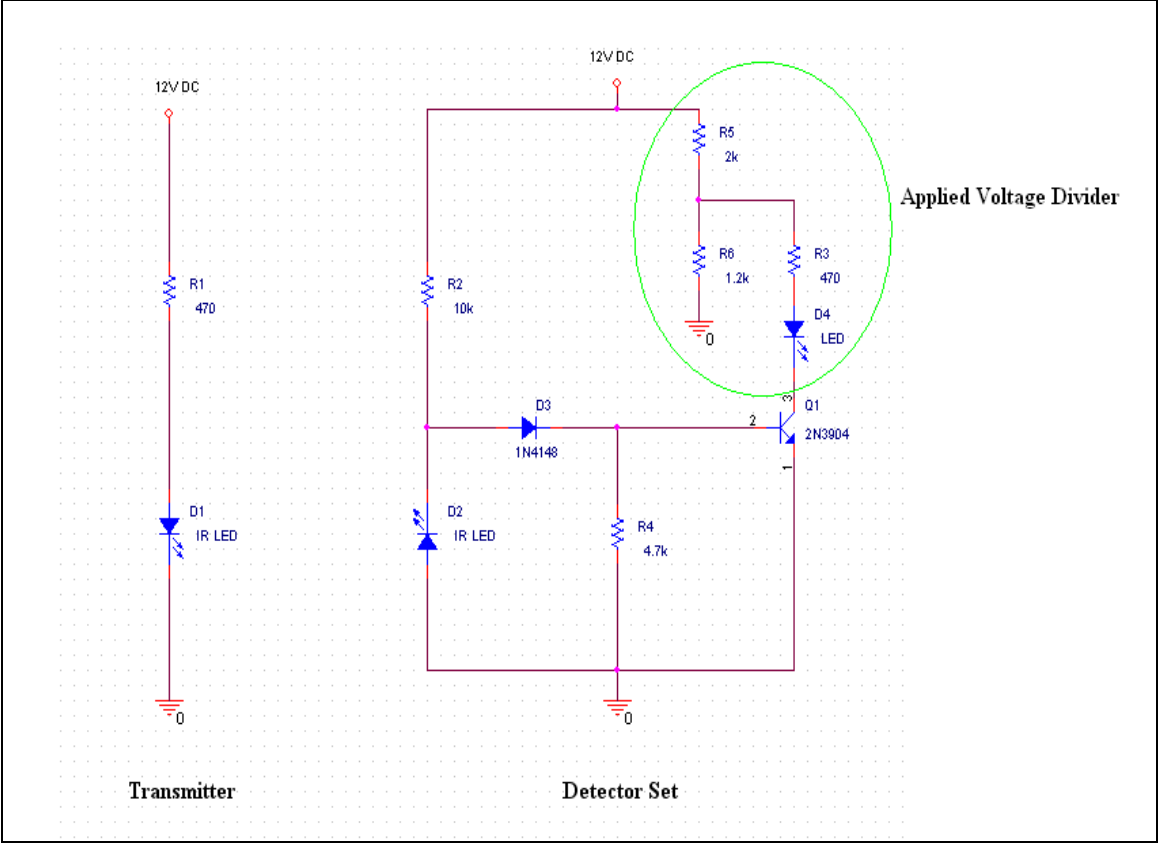


Figure 3.4: Modified IR Motion Detector Circuit

$$\begin{aligned}
 V_{R6} &= \frac{1.2k}{1.2k+2k} \times 12 \\
 &= 4.5V
 \end{aligned}
 \tag{3.1}$$

Applying Voltage Divider as shown in Figure 3.4 will give the result at V_{R6} equal to 4.5V as shown in equation 3.1. Table 3.1 shows the output voltage for the circuit before and after modification. From the table it shows that the output voltage

is reduce when applying the voltage divider. It decreases from 10.45V to 3.5V. The computer will only read logic 1 and logic 0 to communicate. Logic 1 is read from 2.4V to 5V and for logic 0 computers will read from 0V to 0.7V.

Table 3.1 : Output Voltage for Circuit Before and After Modified

	Circuit Before Modified	Circuit After Modified
o/p high (logic 1)	10.45V	3.5V
o/p low (logic 0)	130mV	60.5mV

3.2.2 Circuit Description

Figure 3.4 shows the circuit that used as a sensor to detect babies in the nursery. Here 12VDC is used as input. When there is no blocking (D2 can receive signal from D1) nothing happened. At this time the output is high. At the detector set, current will flow from 12VDC to resistor R2 through the IR LED, D2 and ground. The current will also flow through R5 and R6.

When there is blocking (meaning: there is a baby block the signal), D2 cannot receive the signal. At this time the output is low. Current will flow to the resistor R2 through the D3. At here, D3 is at active high so that current can flow through the D3 until reach the Q1 base. Transistor will active and then will operated. So that, current will flow from R5 through R6 and also R3. Then, it will flow through the LED and then to transistor until the ground because at this time the transistor is function. LED will light ON meaning that there is a baby blocking the signal.

3.3 Phase 2 (Software)

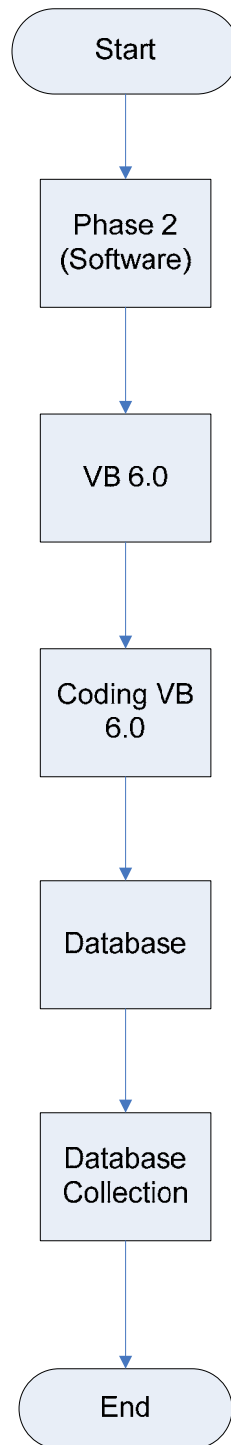


Figure 3.5 : Phase 2

Figure 3.5 shows the step for the software parts. Here, software consist two parts that are VB 6.0 and database. Database is used to store the data in the system. This system will have the database that will recorded all the data for the babies and also for the baby sitter.

3.3.1 Visual Basic 6.0

Visual Basic was designed to be easy to learn and use. The language not only allows programmers to easily create simple GUI applications, but also has the flexibility to develop fairly complex applications as well. Programming in VB is a combination of visually arranging components or controls on a form, specifying attributes and actions of those components, and writing additional lines of code for more functionality. Since default attributes and actions are defined for the components, a simple program can be created without the programmer having to write many lines of code. Performance problems were experienced by earlier versions, but with faster computers and native code compilation this has become less of an issue.

Although programs can be compiled into native code executables from version 5 onwards, they still require the presence of runtime libraries of approximately 2 MB in size. This runtime is included by default in Windows 2000 and later, but for earlier versions of Windows it must be distributed together with the executable.

Forms are created using drag and drop techniques. A tool is used to place controls on the form (window). Controls have attributes and event handlers associated with them. Default values are provided when the control is created, but may be changed by the programmer. Many attribute values can be modified during run time based on user actions or changes in the environment, providing a dynamic application. For example, code can be inserted into the form resize event handler to reposition a control so that it remains centered on the form, expands to fill up the form, etc. By inserting code into the event handler for a key press in a text box, the program can automatically

translate the case of the text being entered, or even prevent certain characters from being inserted.

Visual Basic can create executables(EXE), ActiveX controls, DLL files, but is primarily used to develop Windows applications and to interface web database systems. Dialog boxes with less functionality (e.g., no maximize/minimize control) can be used to provide pop-up capabilities. Controls provide the basic functionality of the application, while programmers can insert additional logic within the appropriate event handlers. For example, a drop-down combination box will automatically display its list and allow the user to select any element. An event handler is called when an item is selected, which can then execute additional code created by the programmer to perform some action based on which element was selected, such as populating a related list.

Alternatively, a Visual Basic component can have no user interface, and instead provide ActiveX objects to other programs via Component Object Model (COM). This allows for server-side processing or an add-in module.

The language is garbage collected using reference counting, has a large library of utility objects, and has basic object oriented support. Since the more common components are included in the default project template, the programmer seldom needs to specify additional libraries. Unlike many other programming languages, Visual Basic is generally not case sensitive, although it will transform keywords into a standard case configuration and force the case of variable names to conform to the case of the entry within the symbol table entry. String comparisons are case sensitive by default, but can be made case insensitive if so desired. [2]

3.3.2 Microsoft Access Database

In computing, a database can be defined as a structured collection of records or data that is stored in a computer so that a program can consult it to answer queries. The records retrieved in answer to queries become information that can be used to make decisions. The computer program used to manage and query a database is known as a database management system (DBMS). The properties and design of database systems are included in the study of information science.

The term "database" originated within the computing discipline. Although its meaning has been broadened by popular use, even to include non-electronic databases, this article is about computer databases. Database-like records have been in existence since well before the industrial revolution in the form of ledgers, sales receipts and other business related collections of data.

The central concept of a database is that of a collection of records, or pieces of knowledge. Typically, for a given database, there is a structural description of the type of facts held in that database: this description is known as a schema. The schema describes the objects that are represented in the database, and the relationships among them. There are a number of different ways of organizing a schema, that is, of modeling the database structure: these are known as database models (or data models). The model in most common use today is the relational model, which in layman's terms represents all information in the form of multiple related tables each consisting of rows and columns (the true definition uses mathematical terminology). This model represents relationships by the use of values common to more than one table. Other models such as the hierarchical model and the network model use a more explicit representation of relationships.

The term database refers to the collection of related records, and the software should be referred to as the database management system or DBMS. When the context

is unambiguous, however, many database administrators and programmers use the term database to cover both meanings.

Many professionals would consider a collection of data to constitute a database only if it has certain properties: for example, if the data is managed to ensure its integrity and quality, if it allows shared access by a community of users, if it has a schema, or if it supports a query language. However, there is no agreed definition of these properties.

Database management systems are usually categorized according to the data model that they support: relational, object-relational, network, and so on. The data model will tend to determine the query languages that are available to access the database. A great deal of the internal engineering of a DBMS, however, is independent of the data model, and is concerned with managing factors such as performance, concurrency, integrity, and recovery from hardware failures. In these areas there are large differences between products. [1]

3.4 Phase 3 (Mechanical Design)

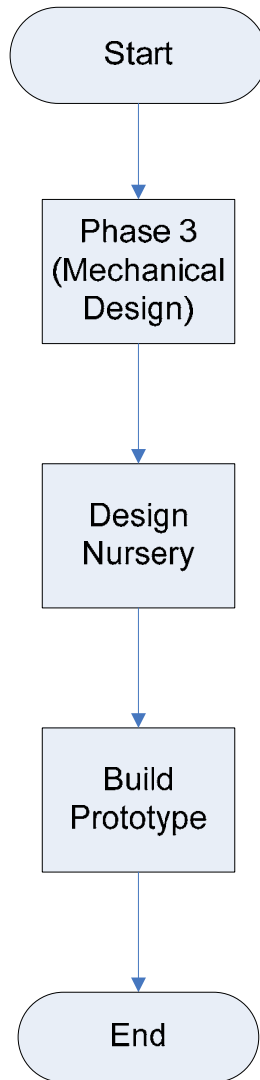


Figure 3.6 : Phase 3

Figure 3.6 shows the step for the mechanical design for the nursery prototype. This is the last part before interfacing all the three phases. The prototype is just made from the card board that can easily found at the stationery. The prototype is used as demonstration on how the system is implementing in the nursery.

3.5 System Overview

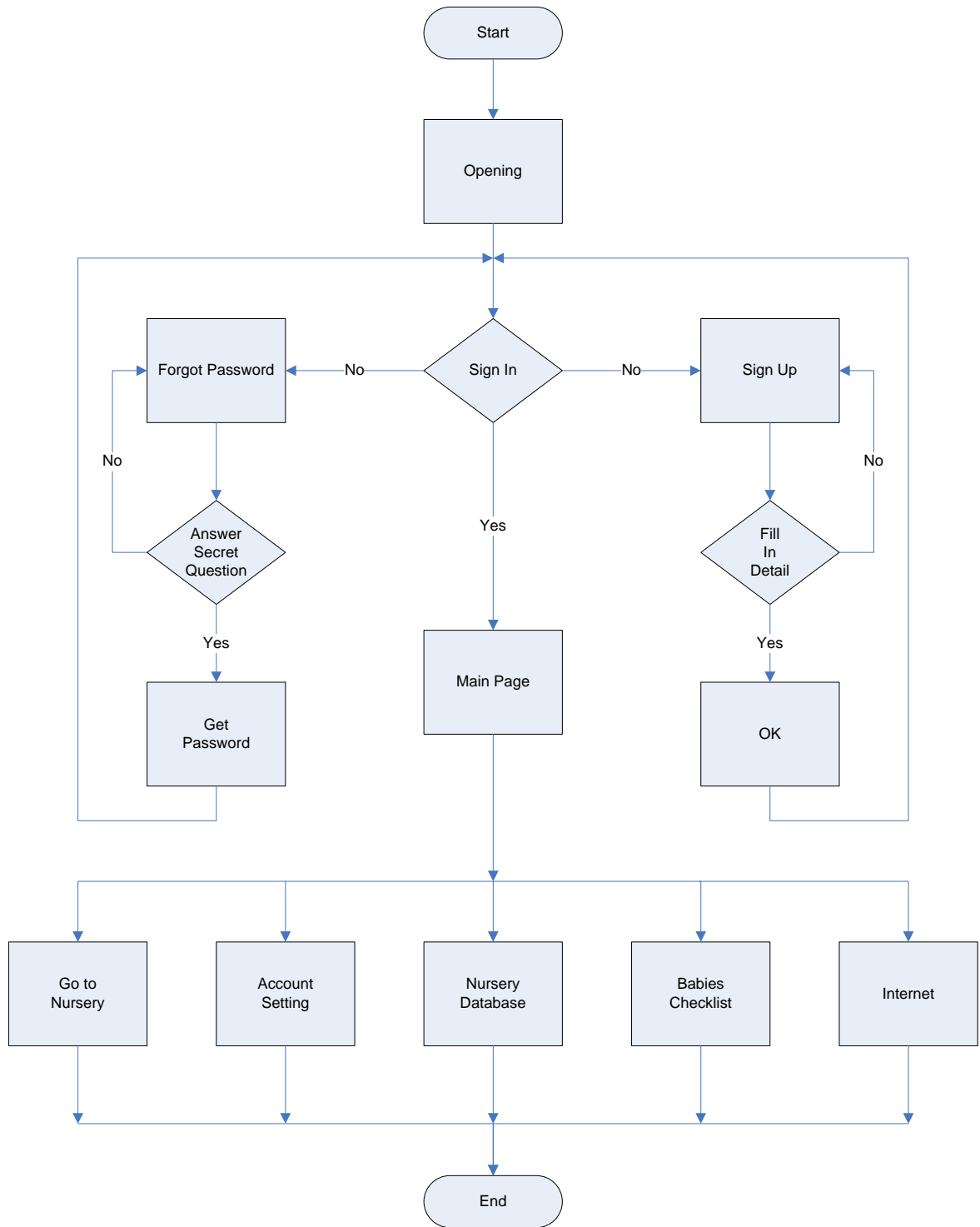


Figure 3.7 : Flow Chart of the System

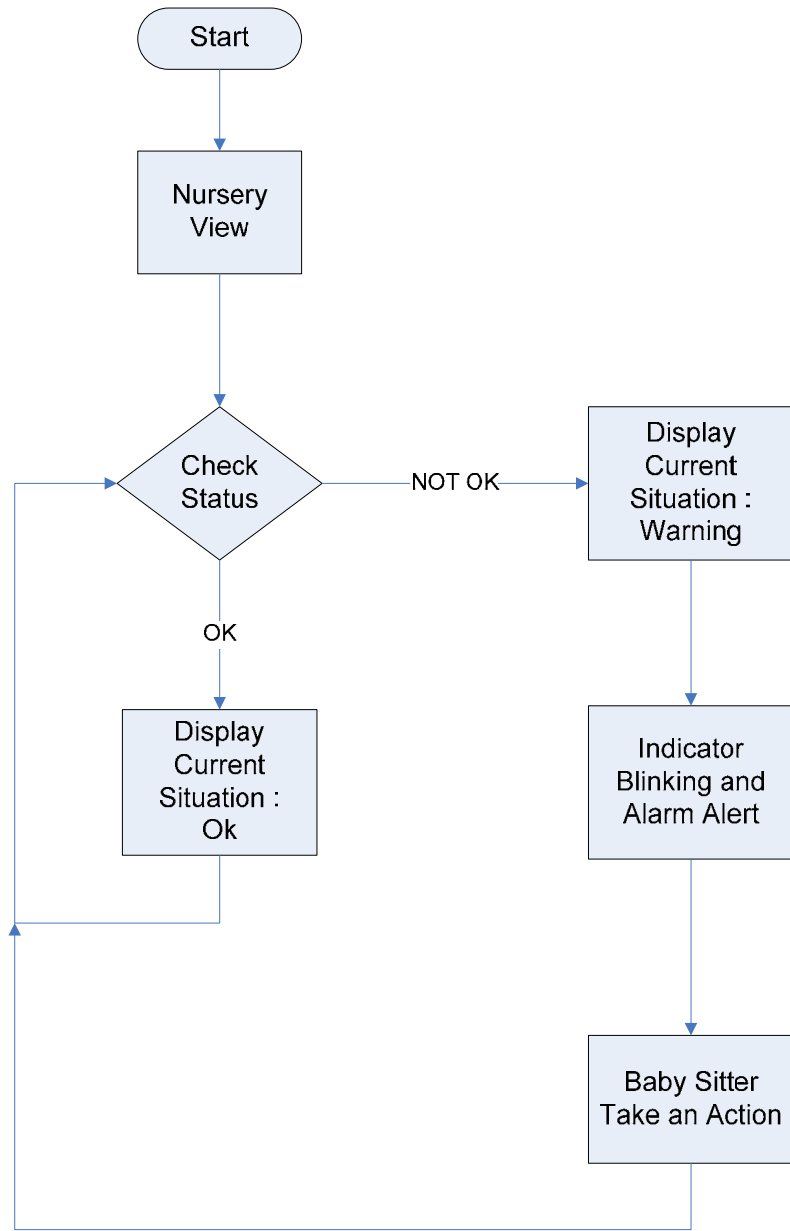


Figure 3.8 : Flow Chart for Checking Status at Nursery

Figure 3.7 shows the system overview of the Baby Monitoring System in nursery. The system in the nursery is handled by the baby sitter at the nursery. The baby sitter needs first to sign in their ID and password to access the system. After signing in, the system will go through to the main page. If the baby sitter is a new comer, they need to sign up first. If

the baby sitter forgot their password, they can go to forgot password section to answer the secret question and then get the password.

At the main page there are five menus to enter. The menu 'Go to Nursery' will go through the system that will monitor the baby at the nursery. The others are the additional menu to make the system more appropriate. Figure 3.8 shows the checking status at the nursery. Baby sitter will check the status by clicking the check status button at the Nursery View's Form. When the status is OK, the display status will appear "OK" and if not "Warning" status will appear and then indicator will blinking and an alarm will alert. Then, baby sitter will take an action based on the display at the monitor.