CHAPTER 3

METHODOLOGY

3.1   Introduction

This chapter comprises two major parts to implement the whole system of real time counting project. The first part is hardware system which explains detail about the circuitry of infrared motion detector. The appropriate detector selection for sensor system is important for obtain excellent result and seasonable with the environment of project.

The infrared motion detector was chosen because of the flexibility, dynamic and facile to build. In commonly infrared sensor consist two main parts that are infrared emitter and infrared phototransistor. This circuit is fast to construct with less requirement of component. Basic operation of this type of sensor also is easily to understand. When there is no perturbation of beam signal which transmit from transmitter to receiver the LED will not lit and if occur disturbance of beam the LED will on to show someone pass by the sensor area. This circumstance prove that circuitry of infrared motion detector is work.

The second part is software system that is containing Visual Basic 6 program which functional as database counter system for user reference. Beside that it will display the information that related to this project. Most of the software section explain on how the arrangement of database and allow the user to amend the database for update the latest data. This create the communication between users and program which convenient to reach by user side. The combination of both hardware and software system make the counting system fully function to develop reliability, usability, portability and security system.

3.2   Hardware portion

Construction of the infrared sensor entails the roster of components in table APPENDIX A (I).

3.2.1   Resistor

A resistor as shown in Figure 3.1 is a two-terminal electrical or electronic component that resists an electric current by producing a voltage drop between its terminals in accordance with Ω 's law:

$$R = \frac{V}{I}$$

The electrical resistance is equal to the voltage drop across the resistor divided by the current through the resistor. Resistors are used as part of electrical networks and electronic circuits.

Types of resistors:

- Carbon composition
- Carbon film
- Metal film
- Metal oxide
- Wire wound (usually has higher parasitic inductance)
- Tantalum

APPENDIX D shows the method on how to read the resistor color.

3.2.2 Diode
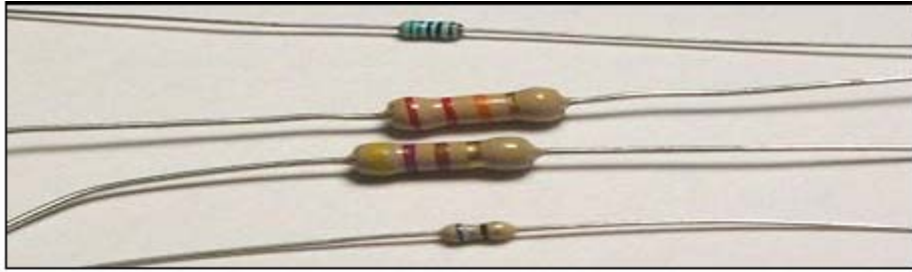
Figure 3.1: Several type of resistor

Figure 3.2: Diode


Figure 3.3: Types of diode

In electronics, a diode is a component that restricts the direction of movement of charge carriers. Essentially, it allows an electric current to flow in one direction, but blocks it in the opposite direction. Thus, the diode can be thought of as an electronic version of a check valve. Circuits that require current flow in only one direction will typically include one or more diodes in the circuit design.

Appendix E shows several types of semiconductor junction diodes and table 3.1 (a) shows the Standard EIA Color Code Table per EIA-RS-279.

3.2.3 Light Emitting Diode (LED)

Figure 3.5: Light Emitting Diode component
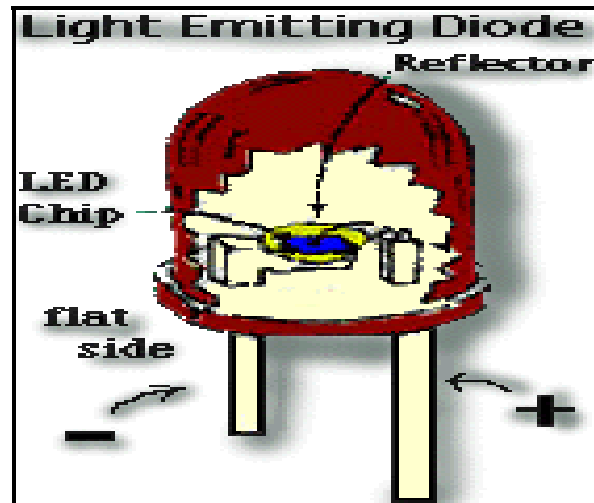
Figure 3.6: Interior design of LED

A light-emitting diode (LED) is a semiconductor device that emits incoherent narrow-spectrum light when electrically biased in the forward direction of the P-n junction. This effect is a form of electroluminescence. LEDs are small extended sources with extra optics added to the chip, which emit a complex intensity spatial distribution. The color of the emitted light depends on the composition and condition of the semi conducting material used, and can be infrared, visible or near-ultraviolet.

When sufficient voltage is applied to the chip across the leads of the LED, electrons can move easily in only one direction across the junction between the p and n regions. In the p region there are many more positive than negative charges. In the n region the electrons are more numerous than the positive electric charges. When a voltage is applied and the current starts to flow, electrons in the n region have sufficient energy to move across the junction into the p region. Once in the p region the electrons are immediately attracted to the positive charges due to the mutual Coulomb forces of attraction between opposite electric charges. When an electron moves sufficiently close to a positive charge in the p region, the two charges "re-combine".

Each time an electron recombines with a positive charge; electric potential energy is converted into electromagnetic energy. For each recombination of a negative and a positive charge, a quantum of electromagnetic energy is emitted in the form of a photon of light with a frequency characteristic of the semi-conductor material (usually a combination of the chemical elements gallium, arsenic and phosphorus). Only photons in a very narrow frequency range can be emitted by any material. LED's that emit different colors are made of different semi-conductor materials, and require different energies to light them.

3.2.4   Circuit Design

Detector circuit distributes two major part (a) IR emitter and (b) IR photo detector. In Figure 3.7 show the overall connection of sensor detector circuit. Value for resistor R1 is 470 Ω , R2 = 10 KΩ and R3 = 4.7 KΩ . D1 is a diode, while L2 is a Light Emitting Diode (LED) and T2 is a transistor. The most important component is L1 and T2 which represent the IR emitter and IR photo detector. This circuit using 12 VDC power supply and 300 mA current. Red lines indicate VCC connection and black line indicate the ground connection in APPENDIX B (Figure 3.7).
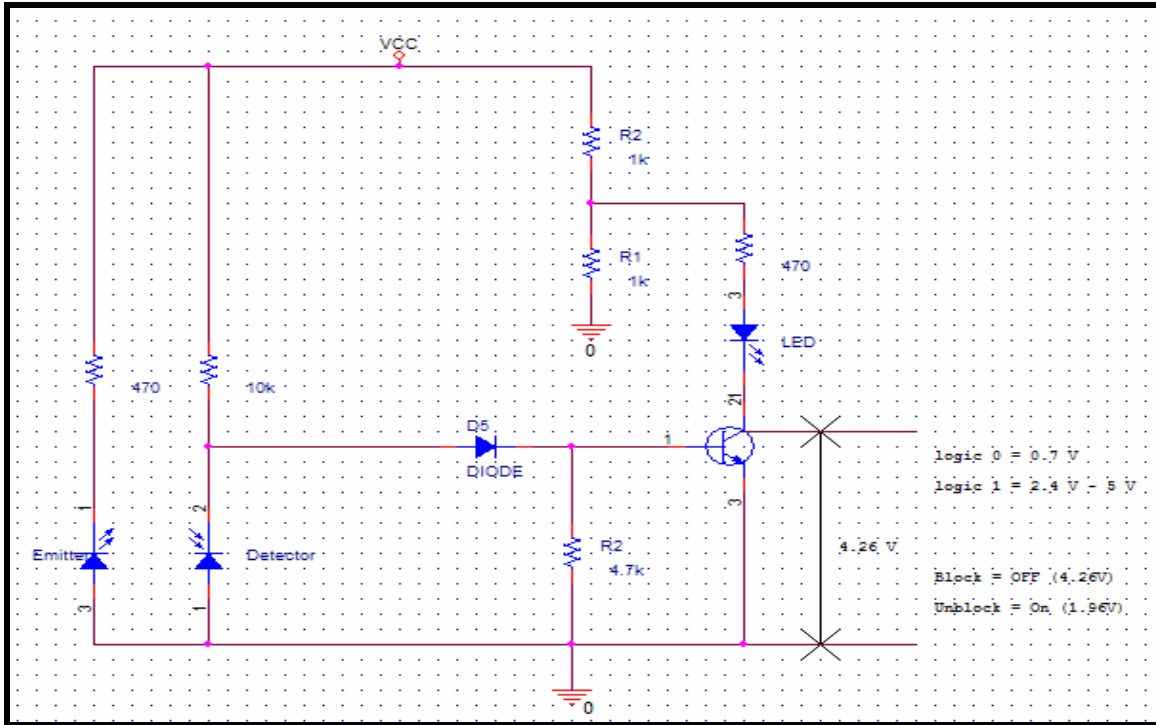
Figure 3.8: Full schematic for detector circuit

(a)   Transmitter circuit part
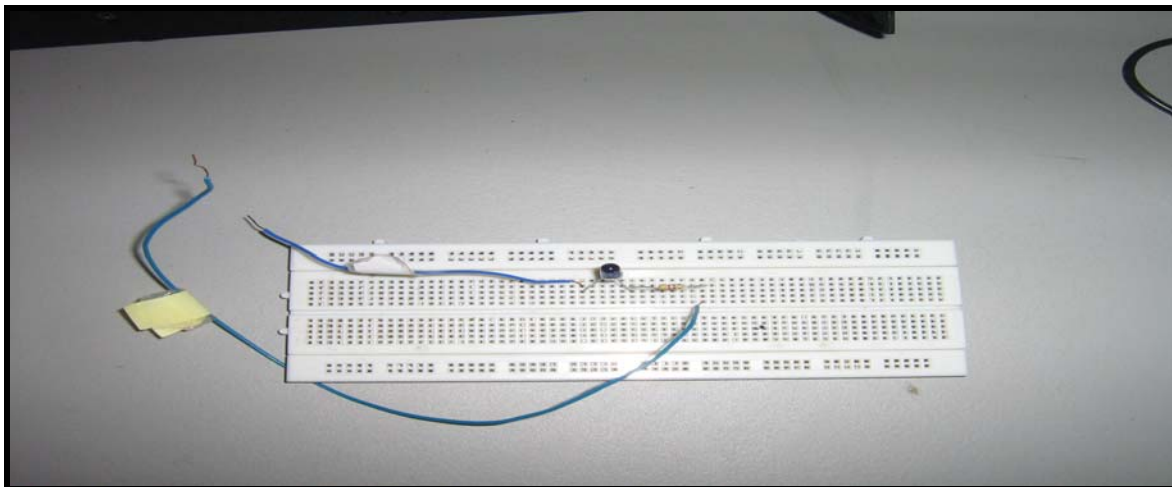
Figure 3.9: Transmitter circuit

Figure 3.9 shows the transmitter contained of 470 Ω resistor and IR Emitter.

(b)   Receiver circuit part
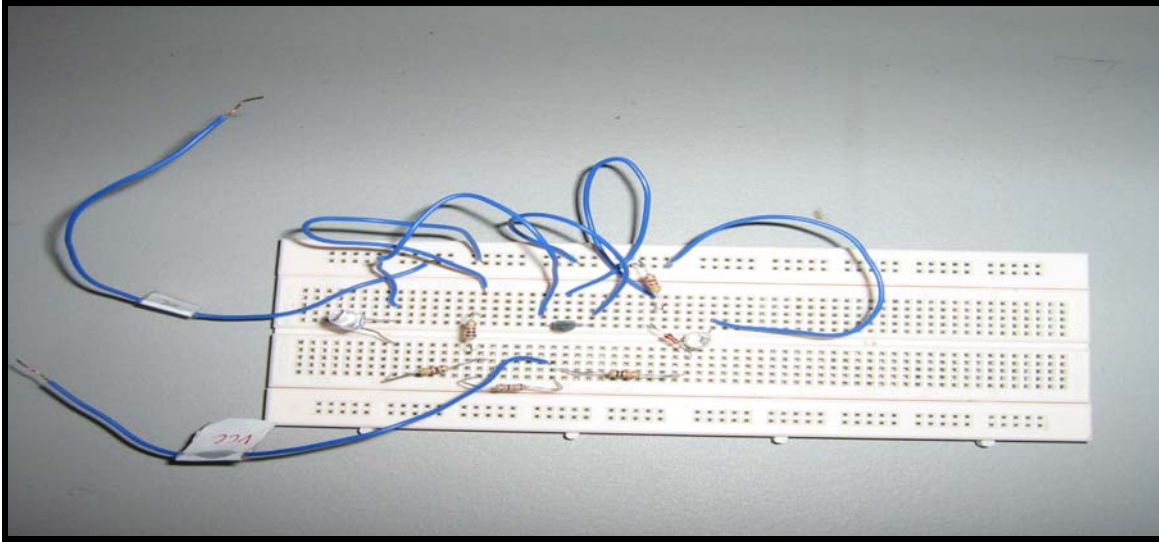
Figure 3.10: Receiver circuit

Figure 3.10 shows a receiver circuit consists of IR photo detector, diode, transistor, LED, 1.8k Ω, 1k Ω, 4.7k Ω and 470 Ω resistor. The above circuit has been modified for origin circuit due to the requirement of parallel port interfacing. Divider voltage circuit was the new additional for stability within circuit. APPENDIX A (II) shows the modification of the sensor circuit.

3.2.5 Process work of determine circuit design

```
┌─────────────────────────────────┐
│    Make research by             │
│    assemble various             │
│    detector circuits.           │
└─────────────────────────────────┘
           │
           ▼
┌─────────────────────────────────┐
│    Simulate the circuit         │
│    detector                     │
└─────────────────────────────────┘
           │
           ▼
┌─────────────────────────────────┐
│    Construct the circuit        │
└─────────────────────────────────┘
           │
           ▼
┌─────────────────────────────────┐
│    Test the circuit             │
└─────────────────────────────────┘
           │
           ▼
NO        ◇ Run? ◇        YES
           │
           ▼
┌─────────────────────────────────┐
│    Detector circuit             │
└─────────────────────────────────┘
           │
           ▼
      (   Done   )
```
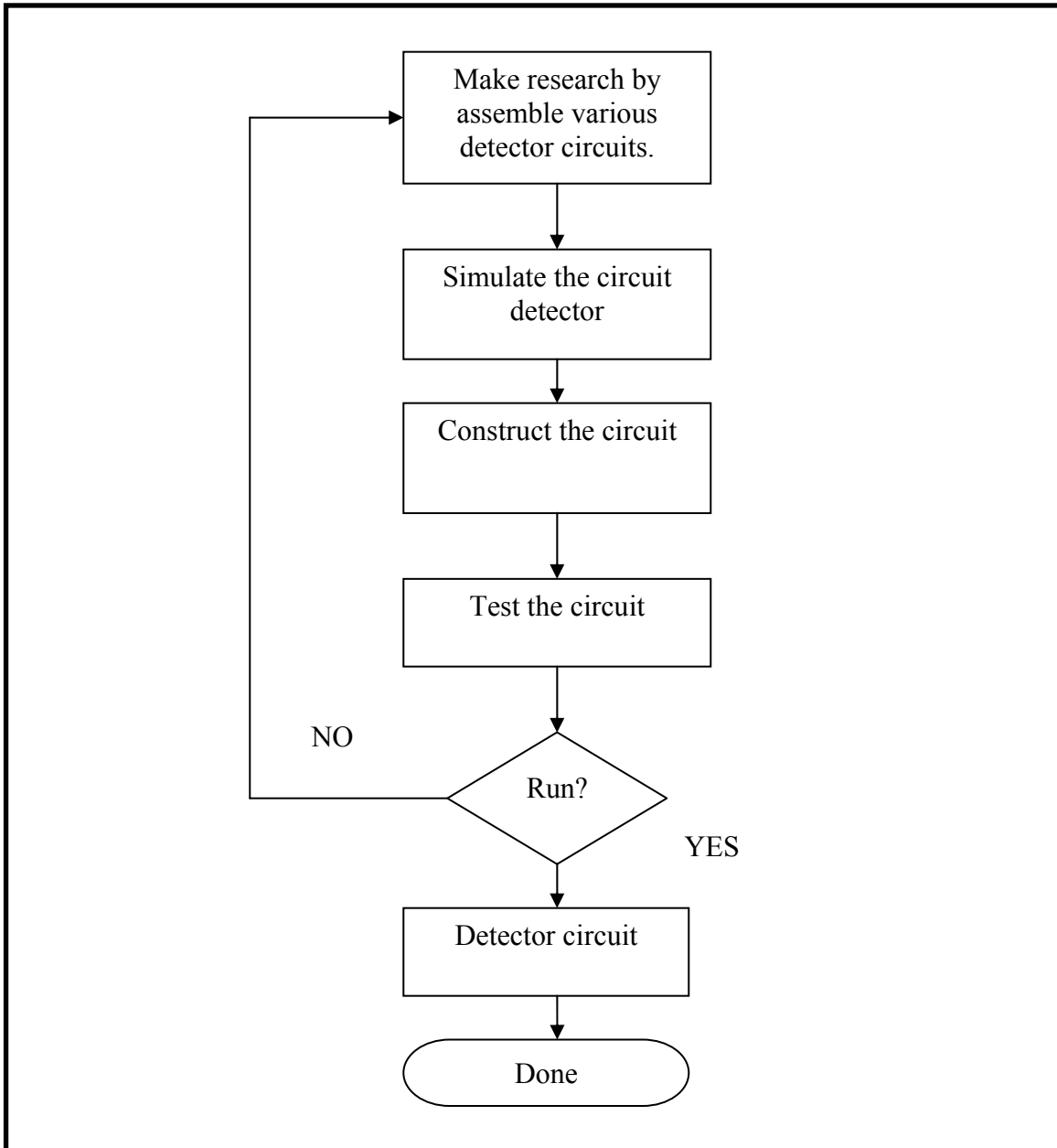
Figure 3.11: Flowchart of process determination of detector circuit

3.2.6   Process flow of circuit detector
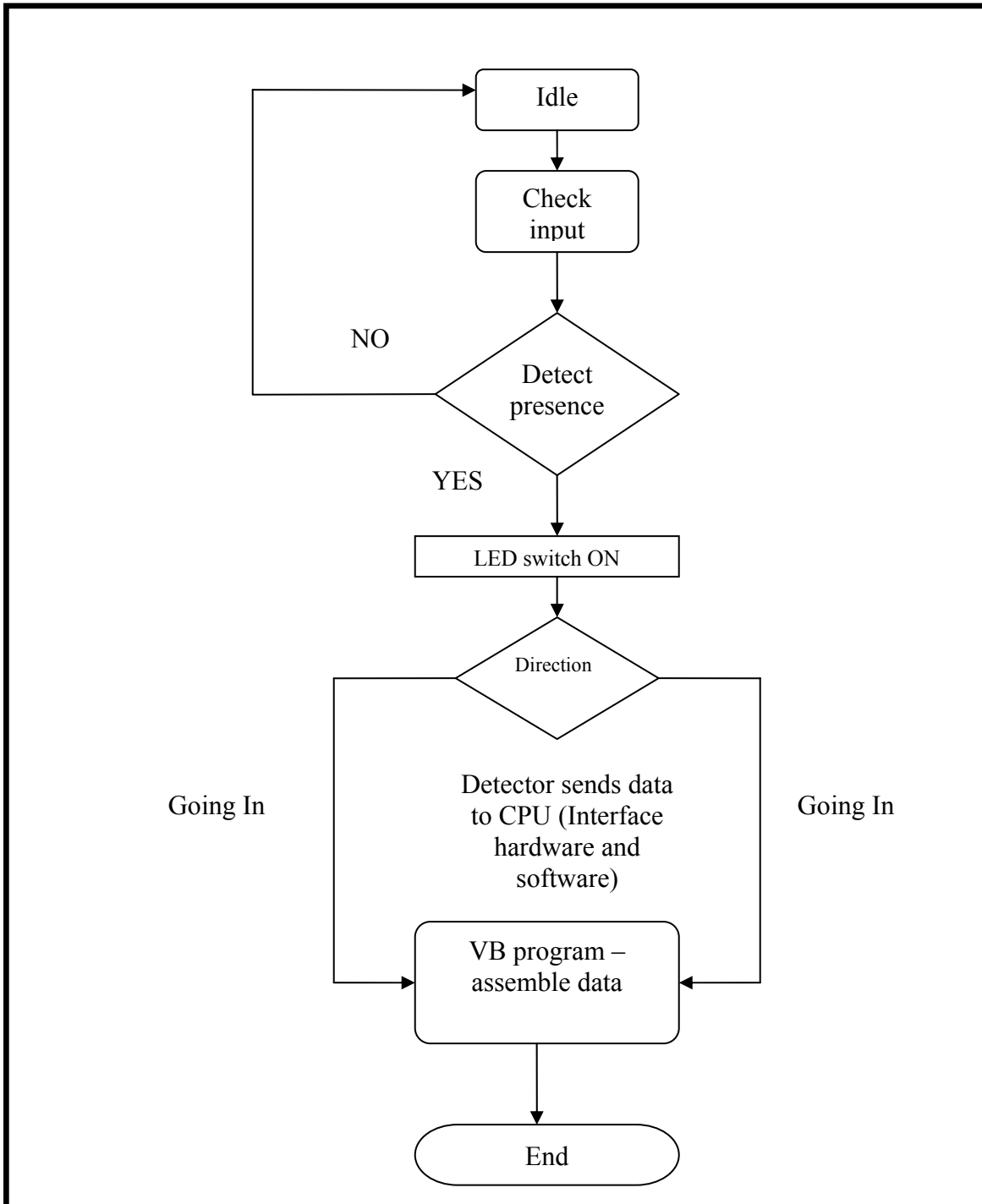
Figure 3.12: Flowchart of process flow of detector circuitry

3.3 Software portion

Whereas for software part, Window 2000 or above installed with Visual Basic 6 programming and Microsoft Access as additional to make the program more systematic and flexible to run.

3.4 Visual Basic 6

Programming is one of essential component to implement this project. This can make the user able to know and more understand the purpose of something about design project. In developing RTCAS, VB 6 was chosen as fit programming language tool the graphical user interface and supports the database system by using MS Access.

Visual Basic is a "visual programming" environment for developing Windows applications. Visual Basic makes it possible to develop complicated applications very quickly. The programmer designs windows graphically, drags program elements from the Visual Basic Toolbox and writes basic code for each element. Visual Basic is "event-driven" which means that procedures are called automatically when the end user chooses menu items, clicks the mouse, moves objects on the screen, etc.

VB 6 is one of the visual basic evolution in develop this software more powerful and compatible with other programming language such as java, C ++, C, VHDL and many more. It also as is an event driven programming language and associated development environment created by Microsoft. It is derived heavily from BASIC. VB enables Rapid Application Development (RAD) of graphical user interface (GUI) applications, easy access to databases using DAO, RDO, or ADO, and easy creation of ActiveX controls. A programmer can quickly put together an application using the components provided with Visual Basic itself. Programmers have undergone a major change in many years of programming various machines. In certain obvious circumstance, what could take days to create an application in other programming languages like C,

C++ could be done in hours with VB 6. VB 6 provides many interesting sets of tools to aid us in building exciting applications.

VB 6 is not only a programming language but also a true graphical development environment. This environment allows programmers with little programming experience to quickly develop powerful Windows applications. VB 6 also has the ability to develop programs that can be used as a front-end application to a database system, serving as the user interface which collects user input and displays formatted output in a more appealing and useful form.

VB 6 is very popular for the ease with which it allows the programmer to create nice looking graphical programs with less coding unlike many other languages that take thousands of lines of programmer keyed code. As the programmer works in the graphical environment much of the program code is automatically generated by the VB 6.

3.4.1   Areas of Use

With using Visual Basic's tools can quickly translate an abstract idea into a program design that can actually see on the screen. Commonly VB program encourages programmer to experiment, revise, correct, and network the design idea until the new project meets the requirements. However, most of all, it inspires programmer imagination and creativity. VB 6 is ideal for developing applications that run on the Windows operating system. There are 3-step approaches when developing applications with VB:

i)   Design the appearance of application

ii)   Assign property settings to the objects of the program

iii)   Write the code to direct specific tasks at runtime

3.4.2   Microsoft Data Report Designer

This is one powerful tool in VB 6 language programming. It allows the programmer to design any type of data report form for certain specific purpose and accordance with latest format of report. Basically there are two way to create the report in Visual Basic. First by using the VB development environment within Microsoft Data Reports and the other way is applying the version of Crystal Reports. To use this can be access in Crystal Report. RPT file within the Crystal's Active X control. Either Data Report or Crystal Report is appropriate for this project and for not become waste time Microsoft Data Report was chosen.

3.4.2.1 Microsoft Data Report (MDR)

In Visual Basic application which involves the Data Report component, the programmer must know how to use and setup the Data Source. Without this Data Report will become trivial because VB program cannot read unknown source. This bind of data and program will allow of programmer to designed, save, edit and compile the project. The advantageous of MDR are dimension of Data Report is fixed and accordance to printed page. Beside that, the MDR have it own report intrinsic controls which no need to bring up ActiveX control.

When using the Data Report, textboxes must be bound to an ADO recordset. In some situations it is necessary to generate this data report at runtime without knowing the column names at design time. In order to use this must have data report included in project with the correct number of controls needed to display the data being retrieved. Then open an ADO recordset and loop through this recordset populating the controls that were placed on the data report. Finally the Data Report can be link to main program and able to achieved by programmer.

3.4.2.2 Process in designing Microsoft Data Report

One of VB 6 tools programming within MDR that is Data Report Designer which enable the programmer to design the data as desired. With using the Microsoft Data Bound (OLEDB), Microsoft Data Control (OLEDB) and Microsoft Data Grid (OLEDB) within VB to display the roster of stadium utility and user log in account. Beside that the user can view the data report and also print directly from page view in visual basic program.

3.4.2.3 Flow chart process of Data Report design
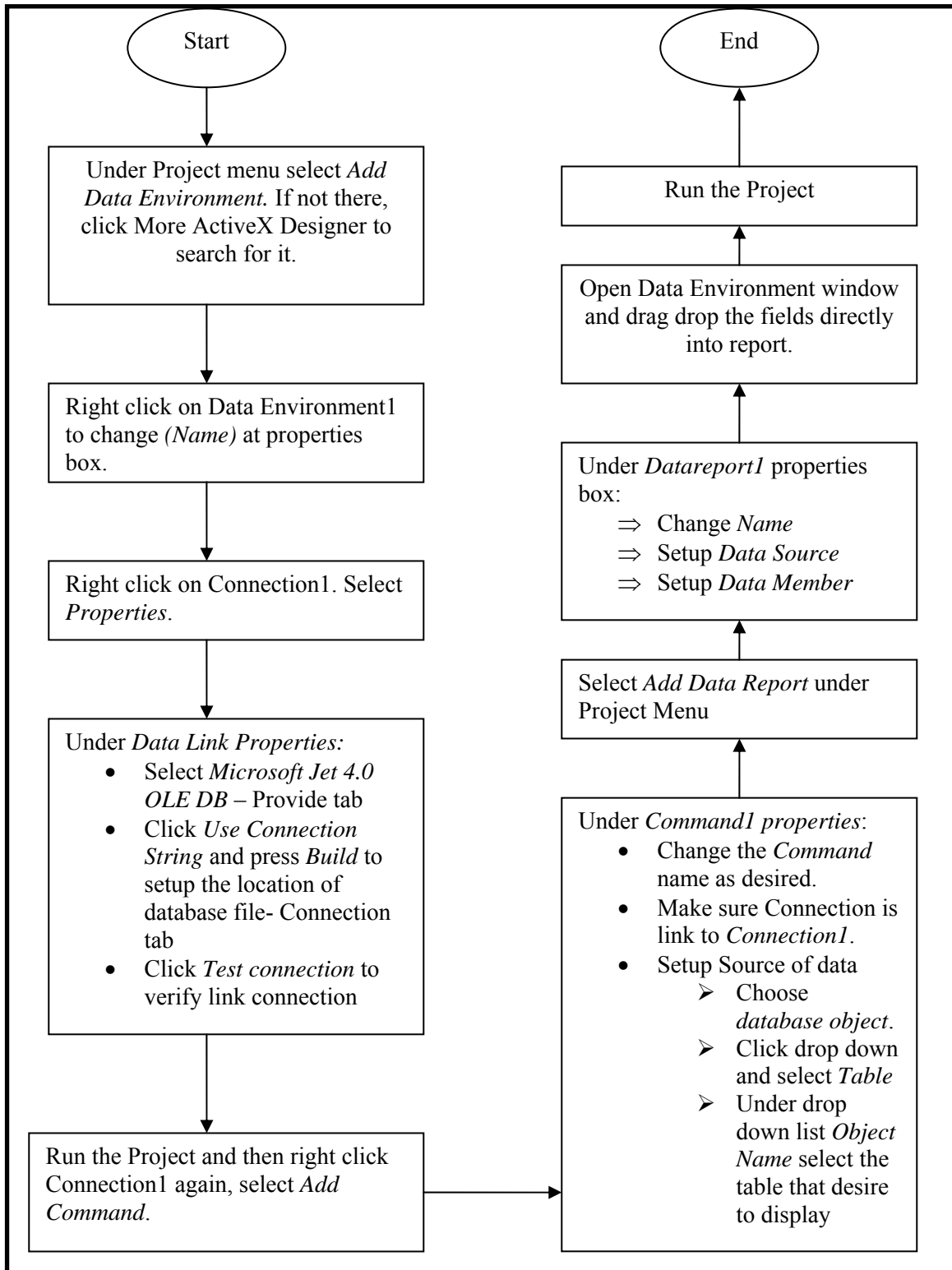
Figure 3.13: Flowchart process for build the Report by using Microsoft Data Report Design (Refer APEENDIX B)

3.4.2.4 Database Design process
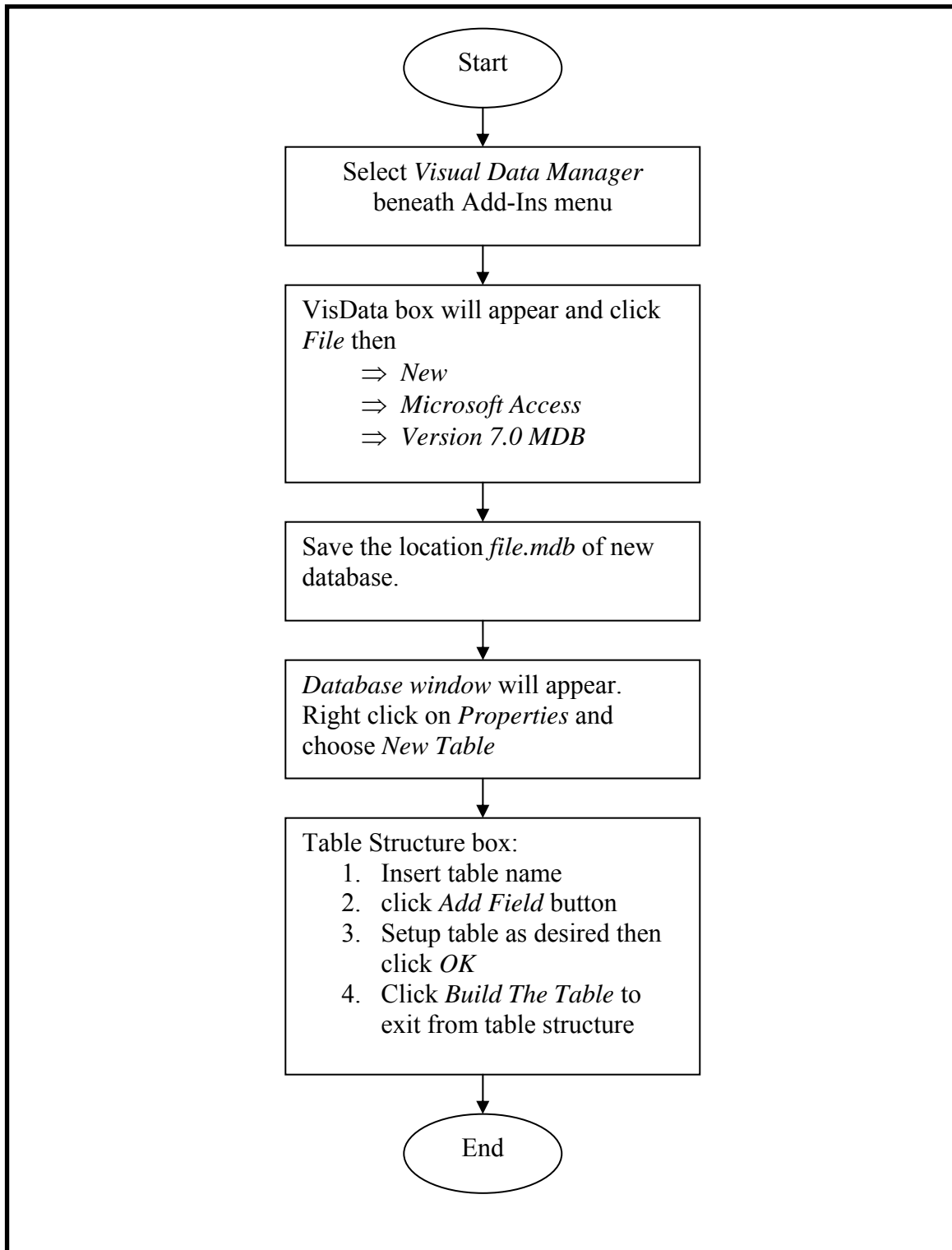
Figure 3.14: Flowchart method in designing database file

3.4.2.5 Applying Data Environment in software procedure

```
┌──────────────────────────────────────────────────────────────────────────────────────────┐
│                                                                                            │
│        ╭─────────╮                              ╭─────────╮                               │
│        │  Start  │                              │   End   │                               │
│        ╰─────────╯                              ╰─────────╯                               │
│             │                                        ▲                                    │
│             ▼                                        │                                    │
│  ┌─────────────────────────┐          ┌─────────────────────────────────┐               │
│  │ Select Standard EXE      │          │ Make a coding to run the software│               │
│  │ project                  │          │ (Refer APPENDIX C)               │               │
│  └─────────────────────────┘          └─────────────────────────────────┘               │
│             │                                        ▲                                    │
│             ▼                                        │                                    │
│  ┌─────────────────────────┐          ┌─────────────────────────────────┐               │
│  │ Save file in same        │          │ Select the Data Grid and Select │               │
│  │ location with database   │          │ Adodc and setup recordsource and│               │
│  │ file                     │          │ connection string               │               │
│  └─────────────────────────┘          └─────────────────────────────────┘               │
│             │                                        ▲                                    │
│             ▼                                        │                                    │
│  ┌─────────────────────────┐          ┌─────────────────────────────────┐               │
│  │ Under Project menu       │          │ Open Data Environment window and│               │
│  │ select Add Data          │          │ drag drop the fields directly   │               │
│  │ Environment.             │          │ into report.                    │               │
│  └─────────────────────────┘          └─────────────────────────────────┘               │
│             │                                        ▲                                    │
│             ▼                                        │                                    │
│  ┌─────────────────────────┐          ┌─────────────────────────────────┐               │
│  │ Right click ...          │          │ Under Datareport1 properties box:│              │
│  └─────────────────────────┘          └─────────────────────────────────┘               │
```
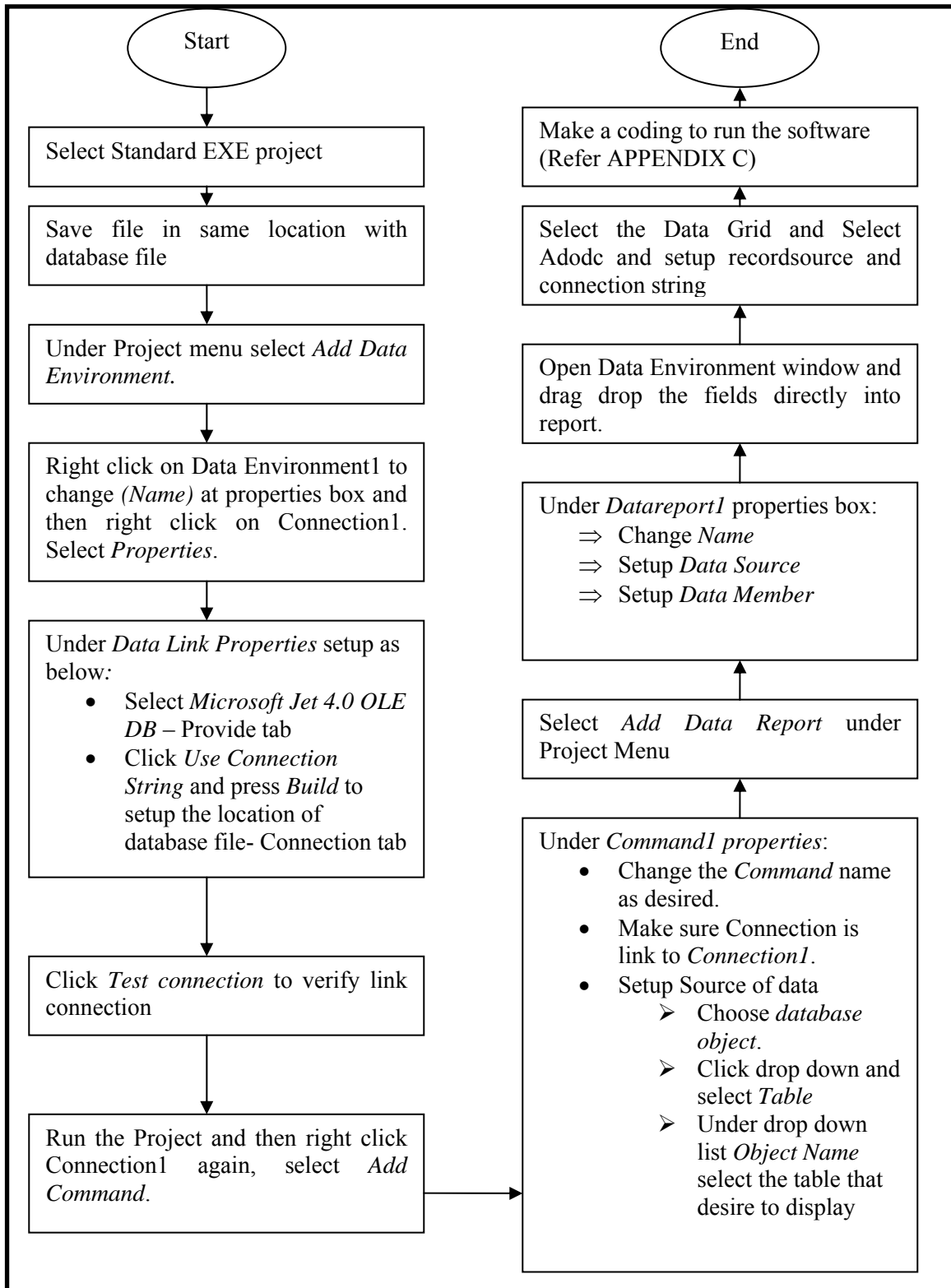
Start

End

Select Standard EXE project

Make a coding to run the software (Refer APPENDIX C)

Save file in same location with database file

Select the Data Grid and Select Adodc and setup recordsource and connection string

Under Project menu select *Add Data Environment*.

Open Data Environment window and drag drop the fields directly into report.

Right click on Data Environment1 to change *(Name)* at properties box and then right click on Connection1. Select *Properties*.

Under *Datareport1* properties box:
⇒ Change *Name*
⇒ Setup *Data Source*
⇒ Setup *Data Member*

Under *Data Link Properties* setup as below*:*
- Select *Microsoft Jet 4.0 OLE DB* – Provide tab
- Click *Use Connection String* and press *Build* to setup the location of database file- Connection tab

Select *Add Data Report* under Project Menu

Under *Command1 properties*:
- Change the *Command* name as desired.
- Make sure Connection is link to *Connection1*.
- Setup Source of data
  ➢ Choose *database object*.
  ➢ Click drop down and select *Table*
  ➢ Under drop down list *Object Name* select the table that desire to display

Click *Test connection* to verify link connection

Run the Project and then right click Connection1 again, select *Add Command*.

**Figure 3.15**: Flowchart process of Data Environment with software

3.4.3   Design of Real Time Counting System (RTCAS)

3.4.3.1 Unified Model Language (UML)

UML is a standard language for documenting, specifying, visualizing and
constructing within software which as a backbone to implement the program system. In
UML comprise several components that make the UML powerful tool such as Data
Modeling concepts (Entity relationship Diagram), Object Modeling, Component
Modeling and Business Modeling (work flow). UML application can be use alone or all
process together.

3.4.3.2 Use Case Diagram

Use case diagram is a set of scenario which emerge from option condition when
the user interaction with program system. Use case diagram is essential to expedite the
flow of program. In RTCAS application, use case will represent the sequences of step
which allow the user to communicate the system from beginning until end of system.
With this, the system can be perform in systematic condition and reduce the system errors.
RTCAS system will begin with login page to enter the login user and then followed by
main system to access the major program of counting system. All action is depend to user
option which each action had been taken will appear new history that had been
programmed.
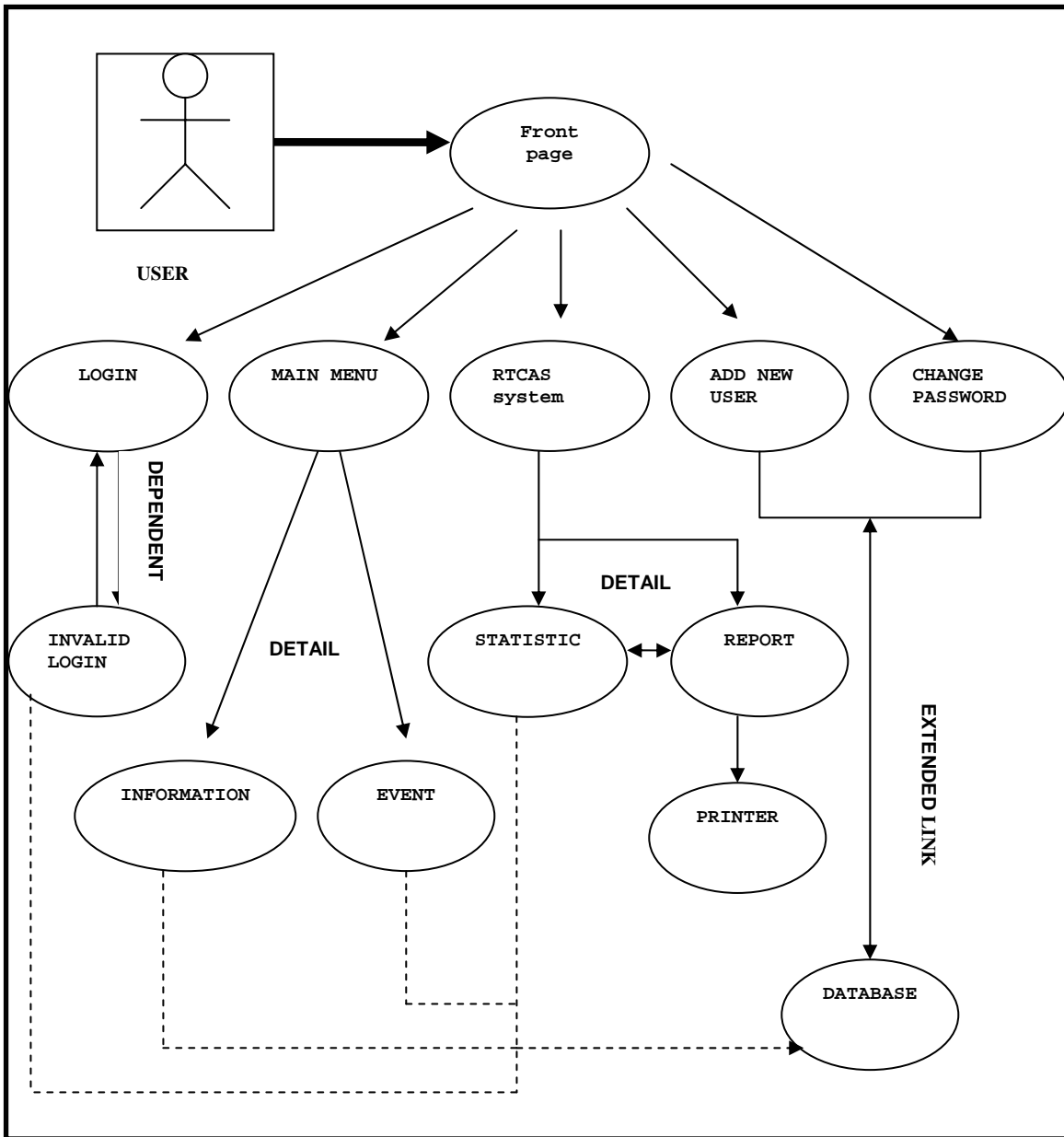
Figure 3.16: Use Case Diagram for RTCAS

3.4.3.3 Enlightenment for Use Case Diagram RTCAS

i. LOGIN

   System security access requirement for every user before can access entire system.
   Login system will check on database system when user tries access this page. When

that user none in database, the system will identify as *unknown user*. Hence the program will appear *invalid code access* message and need to re– login.

ii. MAIN MENU

Comprise the main branch of major system which connects to another window. Spilt into two main part Information and Event.

iii. RTCAS SYSTEM

Reckon system with using sensor can be control in this window. The user can review extra information and statistic of audience. Data report also can be view at here and enable the user to print the report.

iv. ADD NEW USER

Allow to register of new user which tries to access the system. This window will save the detail of user account in database system.

v. CHANGE PASSWORD

This function to allow the old user change the password key as desired. The database system will automatically change accordance to changing of new user password.

3.4.3.4 Sequence Diagram

Sequence diagram is specific diagram for every flow action which relating to use case condition.
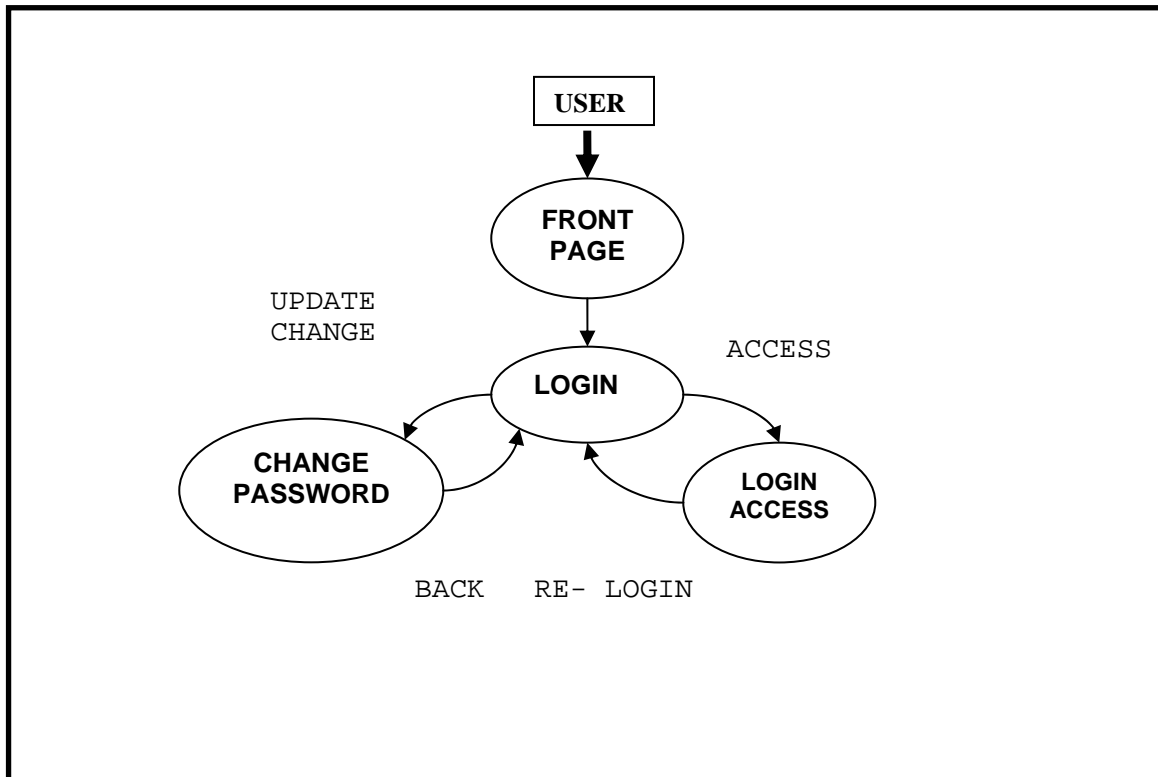
Figure 3.17: Sequence diagram for LOGIN which comprises login access and change password

When users activate the front page button, the LOGIN window will display at screen. There are two main options for user after success access, whether want to access main system or change new password. At LOGIN ACCESS, if failure access user has to re-login. CHANGE PASSWORD allows the old user to alter the new key password as desired.
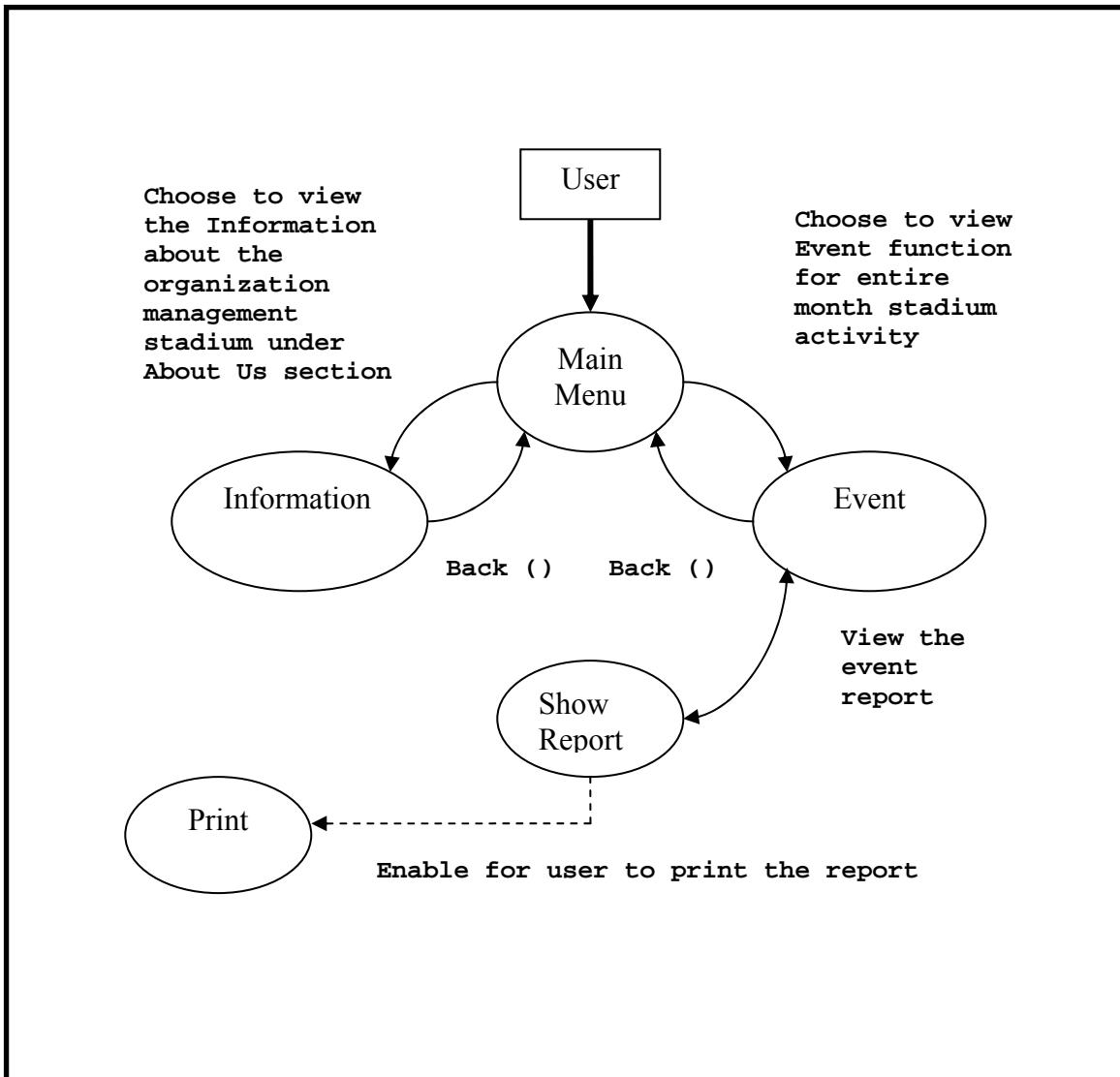
Figure 3.18: Sequence diagram Main Menu

Main menu split into two divisions that are Information and Event. Information contains the management of organization stadium while Event displays the activity of stadium function for every month. This part enables the user to add, edit and delete the database system. Beside that, Event can be view in Report format and also can be print out.
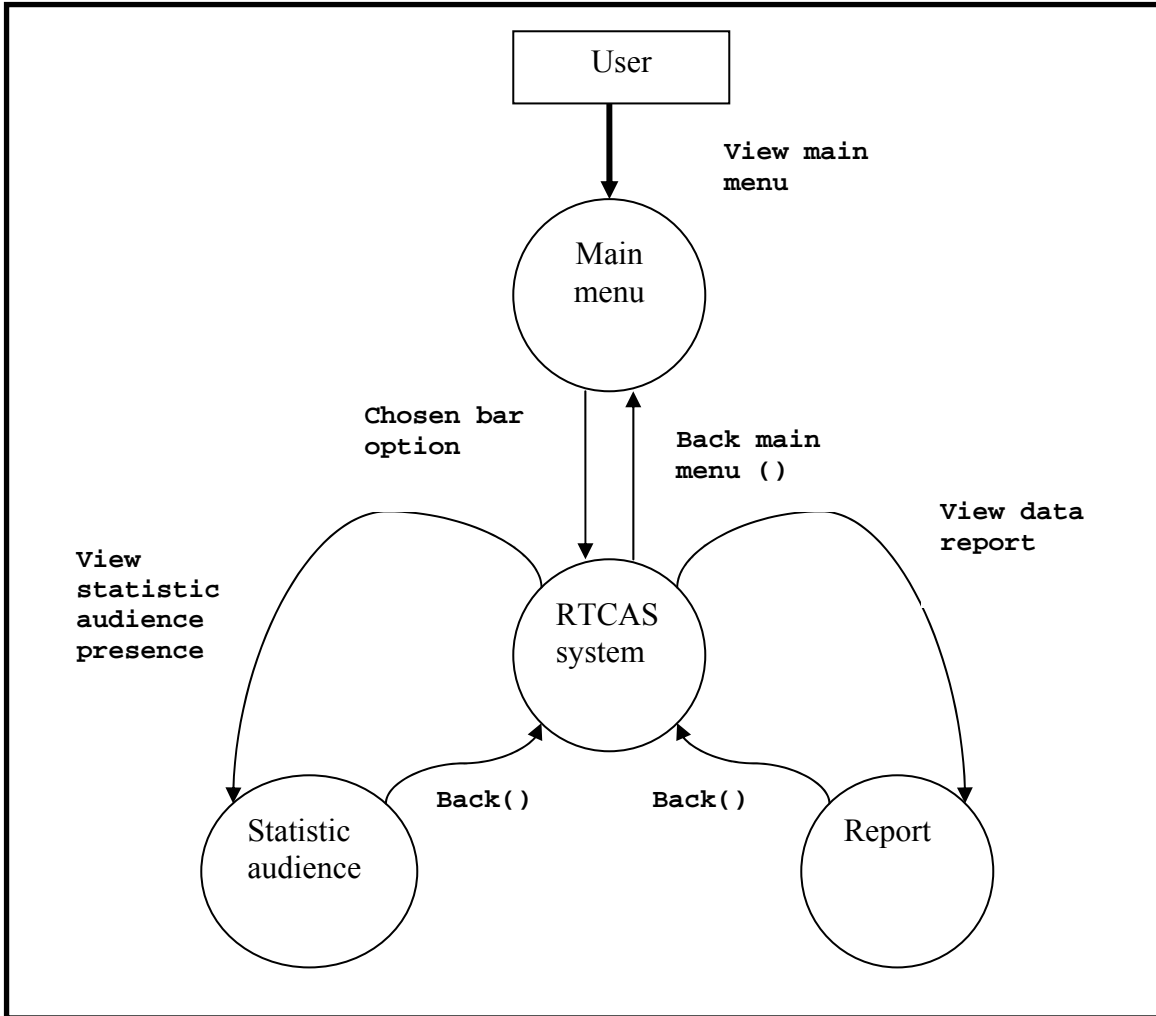
Figure 3.19: Sequence diagram for RTCAS system

This section enables the user to access the counting system and control this system accordance to what instruction user given. Beneath RTCAS system contain statistic of audience and report view. Statistic part shows the statistical of behavior audience (going out or going in), remain available sit and total of bystander attendance at stadium. Report part is to display the entire events that have been recorded in database system. From here user can directly print report to obtain the hard copy.

3.4.3.5 Overall system flow
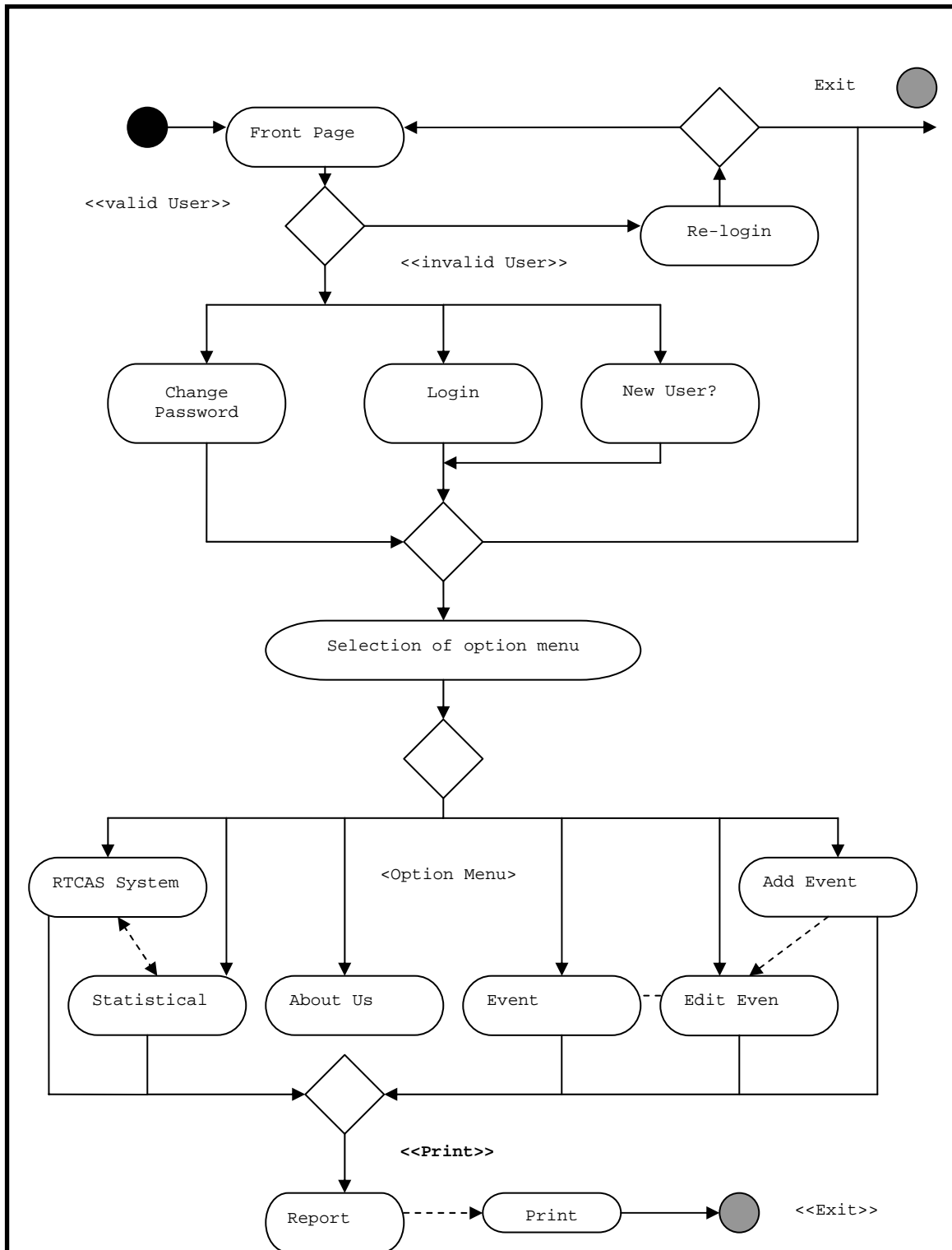
Figure 3.20: Overall of diagram activity for RTCAS system

The overall system for RTCAS is as shown Figure 3.20. This figure describes the sequence of beginning of RTCAS system until end of system. System starts with Login condition where the users have to know their own password and username. Otherwise it will jump to Re - Login condition as access denied state. New user can select New User to register. If through, the next situation is main menu selection which contains six options RTCAS system, Statistical, About Us, Event, Add Event and Edit Event. For RTCAS system, Statistical, Event, Add Event and Edit Event provide the report section and enable user to print the report. Lastly, exit from the system to end the session interaction of RTCAS system.

3.5   Microsoft Access System within database

Microsoft Access is a development environment used to create computer databases for the Microsoft Windows family of operating systems. Usually Microsoft access used in VBA, the programming language that ship with Microsoft Access. A database is a collection of different types of data organized so that its contents can easily be accessed, managed, and updated. In database management comprises such as Access, FileMaker Pro, Oracle or Structure Query Language (SQL) Server equipped with software tools that essential to organize data in flexible manner. The common function that can be utilize in Microsoft Access such add, edit or delete, ask question (or queries) about the data stored in database and yields the data report for display the selected contents.

Figure 3.21: Microsoft Access for 'Function' database

Figure 3.21 shows the database of function stadium which contains four main fields. The first field is Games to identify what type of game will be arrange. Second Venue is to recognize the specific location of stadium. Third Date is tentative available date for prevent the clash of game day. Lastly Capacity is expectation the presence of audience and enables to control the flood of audiences. Rows and columns can be added and reduce accordance to what we desired. In one of MS access file can be also added several tables, queries, Forms, Reports and Pages beneath the Object box.

The capability MS Access combine with database system make the report and data management flexible and systematic to utilize. Database system is only able to preserve the data but with the MS Access capability allow the system open the tables that had been recorded. This system more expedient with access queries that combine the table from multiple types and place the specific condition on the data retrieved.

3.6   Normalization

Normalization is a method to arrange the management of data and prevent the redundancy of error data within database system. Database will split into two or more tables and relationship will be defined. It can be execute with make deleting; addition and modification procedure of the certain field.

Table 3.3: Database design for Login access

Field Name	Data Type	Type of Key	Description
Username	Text	Primary key	Unique code
Password	Text	Primary Key	Unique code

Table 3.4: Database design for January, February and March function

Field Name	Data Type	Type of Key	Description
Games	Text	Primary Key	Unique code
Venue	Text	Primary Key	Unique code
Date	Date/Time	Primary Key	Unique code
Capacity	Text	Primary Key	Unique code

Table 3.5: Database design for RTCAS system

Field Name	Data Type	Type of Key	Description
Total Audience	Text	Primary Key	Unique code
Going Out	Text	Primary Key	Unique code
Going In	Text	Primary Key	Unique code
Time	Text	Primary Key	Unique code
Date	Text	Primary Key	Unique code
Stadium	Text	Primary Key	Unique code

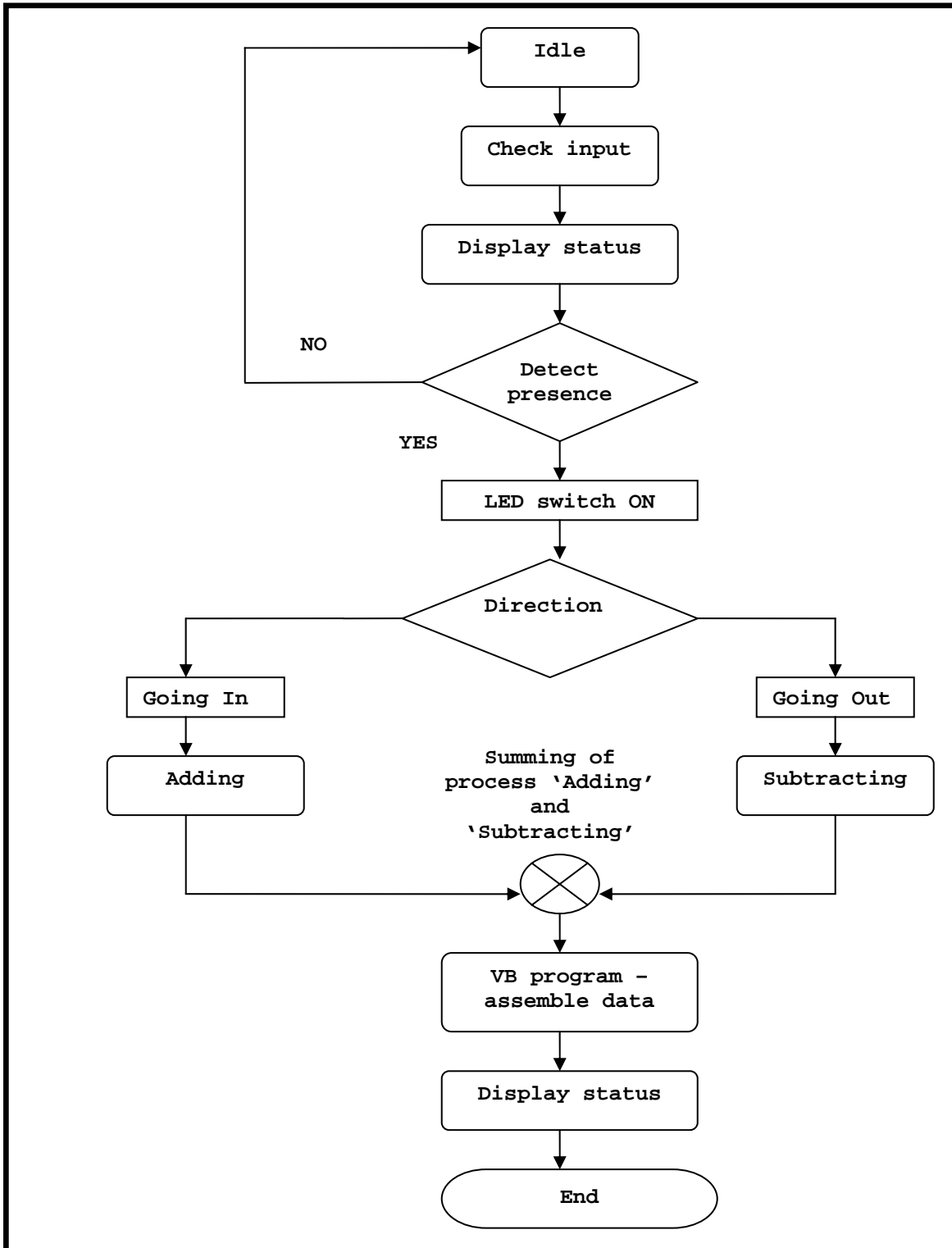## 3.7 Process flow of entire system with interface device and software program

Figure 3.22: Flowchart of process flow of detector circuitry