Contents

Chapter One

GPIT Competence Development Tool Documentation Preface

This chapter covers the document purpose, evaluation of the documents, overview of the document and some preliminary idea about the project and documents.

1.1 Purpose of the documentation

The purpose of this document is to present a detailed description of the Competence Development Tool. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be proposed to the GPIT Human Resource Management Team for its approval.

It also serves as a contract between the customer and the supplier to approve the requirements contained here, as specified in the acceptance criteria.

The audience of this document (Software Requirements Specification) primarily includes the Project Team, the Client and the Project Supervisor.

1.2 Intended Audience

This Document is intended both for Department Head and coordinator who may contribute additional requirements or commentary, and for software project managers and developers who will implement the requirements. As such, it aims for a high level of readability for a non-technical audience, while providing enough specificity to be useful to a software developer.

It is assumed that when software development occurs, it will be in a highly collaborative and iterative environment in which end-users have multiple

opportunities to review and refine the user interface and software functionality.

It is also assumed that the reader has a general understanding of Competence Development Training services and processes and does not require definition of common Competence Development Training related words.

1.3 Evaluation of this Document

Our Team will endeavor to elicit all of the requirements for the proposed system in the early stages of the project life-cycle. However, it may be impracticable for some requirements to be specified during the initial stages.

As such, this document has been produced as thoroughly as is possible at this time. Any changes deemed necessary by the client in conjunction with our Team will be formally documented by revising this document in a traceable manner.
In order to prevent alterations to the requirements at a later phase in the production process, our Team will provide the client with a prototype of the system, showing all interfaces, and allowing the client to suggest any changes.

1.4 Glossary

Term	Definition
Competence Development Tool	A web portal for GPIT to perform their training through web based system.
Database	Collection of all the information monitored by this system.
Software	A document that completely describes all of the functions of a proposed system and the constraints under which it must

Requirements Specification	operate.
Stakeholder	Any person with an interest in the project who is not a developer.
User	The person who uses this system.

1.5 Overview of Document

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

1.6 Conclusion

In this chapter we discuss about the document purpose, intended audience, overview of this document and definition of some used term in the document which provide a clear view to the reader of this document about the purpose of this document.

Chapter Two

GPIT Competence Development Tool Project Plan

This chapter covers the project proposal and the feasibility of the proposal along with background study and some preliminary plan about the project.

2.1 Background Study

Training is a vulnerable part of a company specially the Software Firms because technology is changed rapidly. So, to cope up with present technology every firm arranges workshop or training for their Employee. GPIT perform their training manually. They have not any online tool to manage employee training. They use excel file to keep up training record. With the development of technology it's possible to build an automated training management tool. An automated training management tool makes it easy to manage training and related information.

2.2 Project Scope

This software system will be a Competence Development Web based Portal for GPIT Human Resource Management Team. This system will be designed to develop competency among the employees of GPIT by providing tools to assist in automating the Training Process and collecting Feedback, which would otherwise have to be performed manually.

2.3 Human Resource Plan

From the above analysis, we can justify that we need the following tasks to operate while the system will implement.

- Terminal Server operating
- Database Management
- Application and other functional monitoring
- Admin Assistant
- Technical Assistant
- Website Maintenance

Though there are some various tasks here but this time only an IT Officer is needed to work as Admin. After that we may need some more people like as Assistant IT Officer as Database handling will be complex and so also other issue. It is suggested that, if there are different people engaged in these different tasks the system will run smoothly.

2.4 Project Timelines

Project State	**Start Time**	**End Time**
Project Proposal	15.01.2013	05.02.2013
Requirements gathering and specification	05.02.2013	15.03.2013

Design and Coding	15.03.2013	25.05.2013
Testing and Enhancement	25.05.2013	03.06.2013
Project submission	-	06.06.2013

Table 2.1: Project Timelines

The figure 2.1 illustrates the project scheduling.

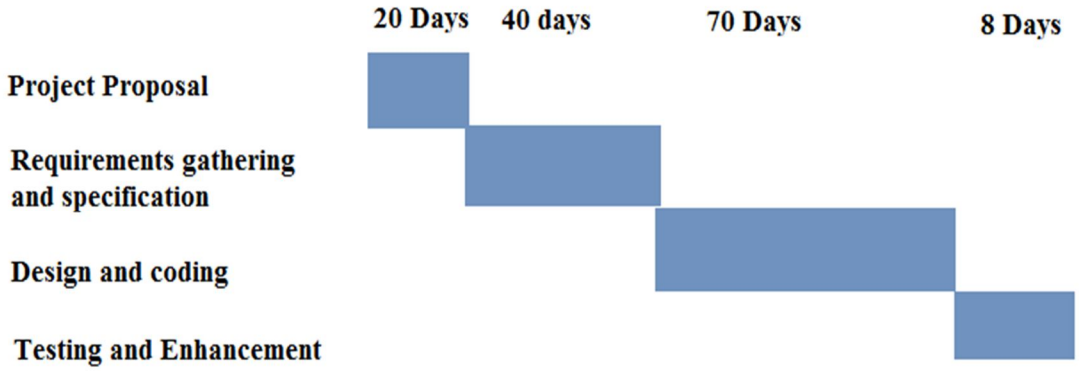


Figure 2.1: Project Scheduling

2.5 Conclusion

In this chapter we discuss the background study of this project, scope of this project and the human resource needed to main this software. Background study and scope must define before building a software project. We also provide a approximated timelines to reach our goal.

Chapter Three

GPIT Competence Development Tool Software Requirements Specification and Analysis (SRS)

3.1 Inception

At project inception, establish a basic understanding of the problem, the people who want a solution, the nature of the solution that is desired, and the effectiveness of preliminary communication and collaboration between the other stakeholders and the software team.

Customer(s) or end users may be located in a different city or country may have only a vague idea of what is required, may have conflicting opinions about the system to be built, may have limited technical knowledge, and may have limited time to interact with the requirements engineer. None of these things are desirable, but all are fairly common, and you are often forced to work within the constraints imposed by this situation.

3.1.1 Stakeholder Identification

Department Head	The Department Head is a controller of Competence Development Tool.
Supervisor	A supervisor supervises employees who work under his/her supervision.
General Employee	General Employees are the trainees or the employees who participate in different training.
Admin	Admin has the right to insert, update, delete above all operate the system.

3.1.2 Recognizing Multiple Viewpoints

Because many different stakeholders exist, the requirements of the system will be explored from many different points of view. The Human Resource Team of GPIT is interested in an automated calendar for training, simple mail sending process.

3.1.3 Working toward Collaboration

If five stakeholders are involved in a software project, you may have five (or more) different opinions about the proper set of requirements. Throughout earlier chapters, I have noted that customers (and other stakeholders) must collaborate among themselves (avoiding petty turf battles) and with software engineering practitioners if a successful system is to result. But how is this collaboration accomplished? The job of a requirements engineer is to identify areas of commonality (i.e., requirements on which all stakeholders agree) and areas of conflict or inconsistency (i.e., requirements that are desired by one stakeholder but conflict with the needs of another stakeholder). It is, of course, the latter category that presents a challenge.

3.1.4 Asking the First Questions

Questions asked at the inception of the project should be “context free”. The first set of context-free questions focuses on the customer and other stakeholders, the overall project goals and benefits.

- Who depends on the project?

  Human Resource Management Department of GPIT depends on this project.

- Who are interested in the outcome of the project?

  GPIT gets the outcome of the project.

- Who will influence the project?

  Our respective teacher Kazi Muhaymen-Us-Sakib Sir influences us about this project. Another reason we are also interested in doing industry work.

- Who can be included in the planning of the project?

  The Developers (S.M Sofiqul Islam, Mostafizur Rahman, Zahidul Kabir, Shanto Rahman)and Competence Development Team of GPIT.

3.2 Eliciting Requirements

3.2.1 Collaborative Requirements Gathering

The goal is to identify the problem, propose elements of the solution, negotiate different approaches, and specify a preliminary set of solution requirements in

an atmosphere that is conducive to the accomplishment of the goal. To better understand the flow of events as they occur, we present a brief scenario that outlines the sequence of events that leads up to the requirements gathering meeting, occur during the meeting, and follow the meeting. We go to GPIT and discuss about their requirements.

3.2.2 Quality Function Development

Quality function deployment (QFD) is a quality management technique that translates the needs of the customer into technical requirements for software. QFD "concentrates on maximizing customer satisfaction from the software engineering process". QFD identifies three types of requirements.

3.2.2.1 Normal Requirements

GPREQ-101 **Priority:** 3

Name: Admin Login System

Description: Support for Admin login accounts; access to modules is granted by use of "roles" or "privileges" that allow Admin access to as many modules as needed.

GPREQ-102 **Priority:** 3

Name: Uploading Employee Information from Excel to Database

Description: Admin can load the excel file into the Database provided that the excel file contains current employee information.

GPREQ-103 **Priority:** 3

Name: Budgeting

Description: At the beginning of each year, the admin will insert budget amount for a particular training and for a particular division.

GPREQ-104 **Priority:** 3

Name: User Creation

Description: The admin will able to create users or sub-admins by giving some required fields.

GPREQ-105 **Priority:** 3

Name: Creating Training Calendar

Description: At the beginning of each year, the admin will create an annual calendar where all training schedule will be inserted. The schedule will include the training names, tentative training dates and desired trainee groups.

GPREQ-106 **Priority:** 3

Name: Sending E-mail Notification from the system

Description: Admin can send E-mail notification to the desired employee group or send E-mail notification to the selected Trainees.

GPREQ-107 **Priority:** 3

Name: Sending Feedback form

Description: Admin can send feedback form to the desired employees and line manager through email.

GPREQ-108 **Priority:** 3

Name: Different Types of Report Generation

Description: Admin can generate different types of reports on the basis of the feedback from the trainees and line manager.

3.2.2.2 Expected Requirements

This project is for GPIT Human Resource Management Team to manage their Training activities. So, we have to strictly follow the requirements provided by GPIT. If any requirements need to complete this project, we will include those requirements later in the documentation.

3.2.2.3 Exciting Requirements

This project is for GPIT Human Resource Management Team to manage their Training activities. So, we have to strictly follow the requirements provided by GPIT. If we are able to provide some features those excite the GPIT Stakeholders, we will include those requirements later in the documentation.

3.3 Requirements Specification

3.3.1 User Story

Competence development tool is a human resource management system to improve competence of employees. This tool is proposed by GPIT. This Project is undergoing on our Software Project Lab (II).

The proposed system will be an interactive web based portal for managing training activities of GPIT. In GPIT three categories of trainings exist. Those are

1. Local External Training – conducted at the local companies like BDJobs, GP etc.
2. Foreign External Training – conducted at the abroad.
3. Internal Training – conducted in house at GPIT.

It is noteworthy that the Internal Training bears no expenses. All those training related information will be available in the proposed system.

The system has two types of users – Admin and general user, Admin will maintain and coordinate all the activities, and users can register for intended trainings and get useful training related information through the portal.

The access of Admin user is restricted through an authentication system. There will be a separate logging interface for the admin, where he/she will fill up some required fields (for example: E-mail and password). Initially, default e-mail and password will be provided to the admin user.

After logging in, the system will give permission to the admin for doing some predefined operations such as Budgeting, Creating Calendar, Sending Notification, Sending Feedback Form, Uploading Employee Information into Database and Generating Report. At the beginning of each year, the admin will create an annual calendar where all training schedule will be inserted. The calendar is visible in the portal and intuitively interactive, that is, if you click any dates of the calendar, the system will show the programs on that day. The schedule will include the training names, tentative training

dates and desired trainee groups (that is people who are suitable for a particular training program).

In this system after creating the training calendar, admin should be able to send e-mail notification to the respective group members. If the admin desires, he/she also can resend the notification in multiple times before the training date. However, by default, the system will remind all the trainees, 3 days before their training program starts.

In Local External Training, expense occurs but here no employee has the capability to show his/her willingness to attend to a particular training. So here who is eligible, get approval from the admin. Before starting the training, admin will insert amount of expenses for that training into the calendar. If a training date expires, a notification will also be sent to the trainees that" All Info Are Correct? Such As Expense Amount, Trainees Name etc.". If admin assures that all info are correct, system will automatically update budget amount, if not then system will not update the expenses.

In Foreign External Training, expense occurs. But the difference between Foreign External and Local External is to the abroad expense

It is also noteworthy that, before a training program starts the admin will send a configuration link to all the prospective trainees. The desired trainees need to click the configuration link to reserve his/her seat. The allocation of the seats will be based on first come first serve basis. However, if it is required, the admin can increase or decrease the initial number of seats of a training program.

Admin can create a user (Department Head or Sub Admin) by giving email and Password. Every User can do some operations. If one user creates one training calendar for that every notification will also be gone to that user.

Admin can load the excel file into the Database (DB) provided that the excel file contains current employee information such as Employee Id, Name, E-mail, Division, Department, Contact No etc. Here three conditions exist -

- If some employees information exists into the DB but not into the excel file, the system will deactivate that employee information.
- If some employees information does not exists into DB, but into the excel file, the system will load the information from that excel file.
- If the excel file cannot be loaded within one month into the DB, a message will be shown that “DB is not updated within one month”.

When training is completed admin manually collect information about employees who attend in the training and who could not. Employee who reserved his/her seat for training but did not attend in the training just delete their info from DB.

This system has a module to get feedback from the trainees who take part in the training. Admin will also send a notification to the employees who took part in the training. The notification includes a feedback form to get feedback from the trainees who attended in the training program. A time slot is set to the system to get feedback from the trainees. If trainees could not fill up the feedback form within the time slot, another remainder will be sent to them.

This system will also provide opportunities to generate report on the basis of feedback. When the feedback information is stored in DB, if admin wants, he/she can generate report such as pie chart, bar diagram and general report on the basis of feedback. If all attended trainees do not fill up the feedback form, admin can generate report, but a message will be shown in the report that "All the trainees did not fill up feedback form. This is not an error free report". If all trainees fill up the feedback form report will be generated without error.

Supervisors/Line managers will also provide feedback on trainee performance. So after three months from finishing training a feedback form will send to the line manager automatically by the system. On the basis of his/her feedback, admin can also generate different types of report.

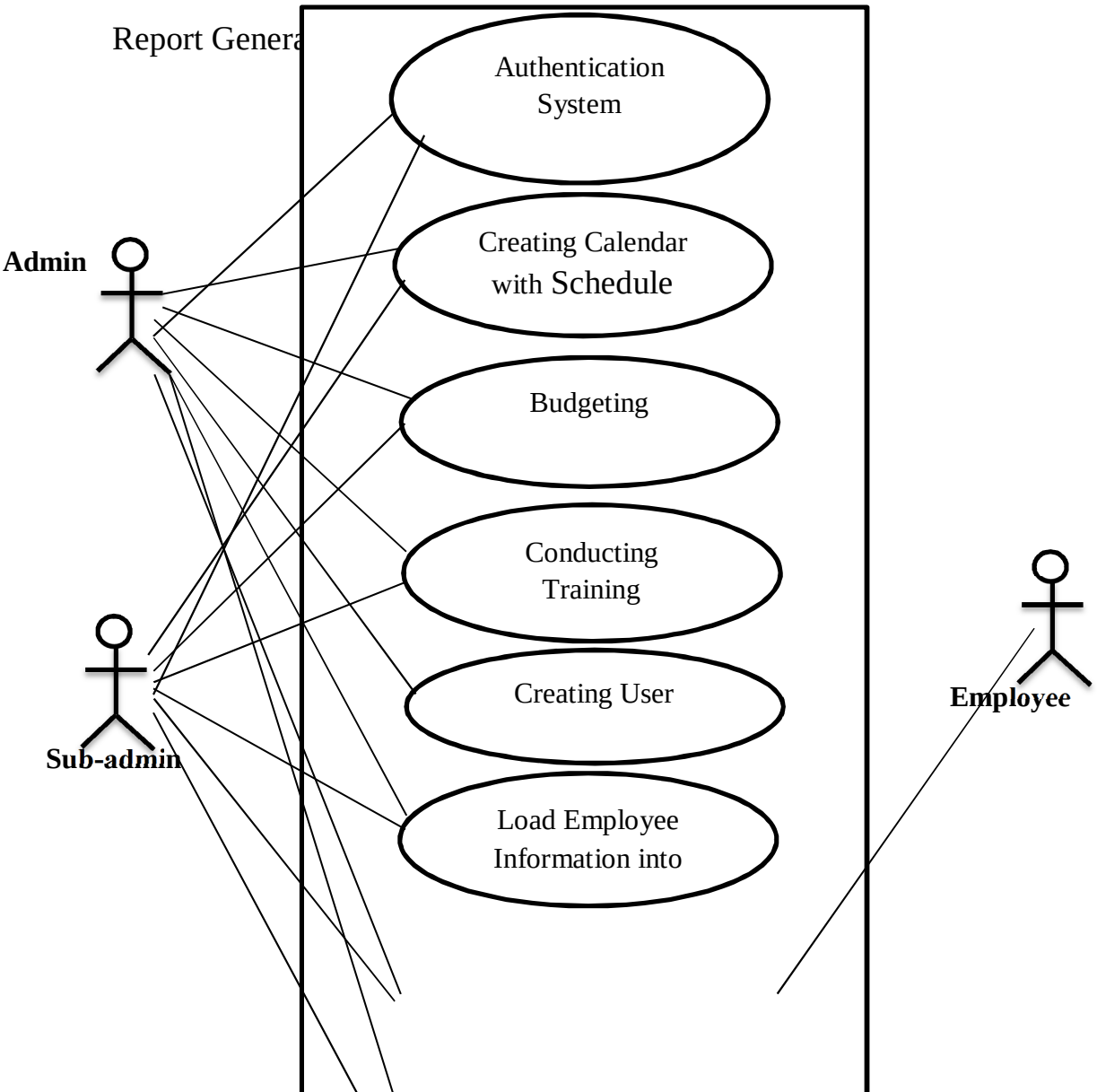
Competence development tool has two types of user such as Admin and General user. Admin have some predefined rights such as inserting budget amount, updating DB, creating training calendar, sending notification, generating report. But general users have the right to apply in specific training and fill up the feedback form through email.

3.3.2 Use Cases

We have written use cases for selected, frequently-performed activities. These are included to supplement the requirements, and to highlight places where good software could bring great improvements in efficiency and ease of use. The steps of the use cases should be considered suggestive rather than prescriptive.

3.3.2.1 Use Case List

	Use Cases	Primary Actor
	Authentication System	Admin
	Creating Training Calendar with Schedule	Admin
3.3.2.2 Use case Diagram (Level – 0)	Budgeting	Admin
	Conducting Training	Admin
	Creating User	Admin
	Upload Employee Information Into DB	Admin
	Sending Notification with a Feedback Form	Admin
	Report Genera	



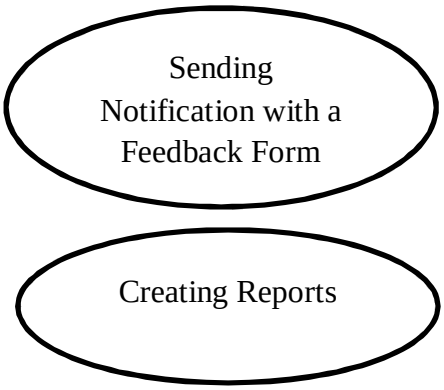
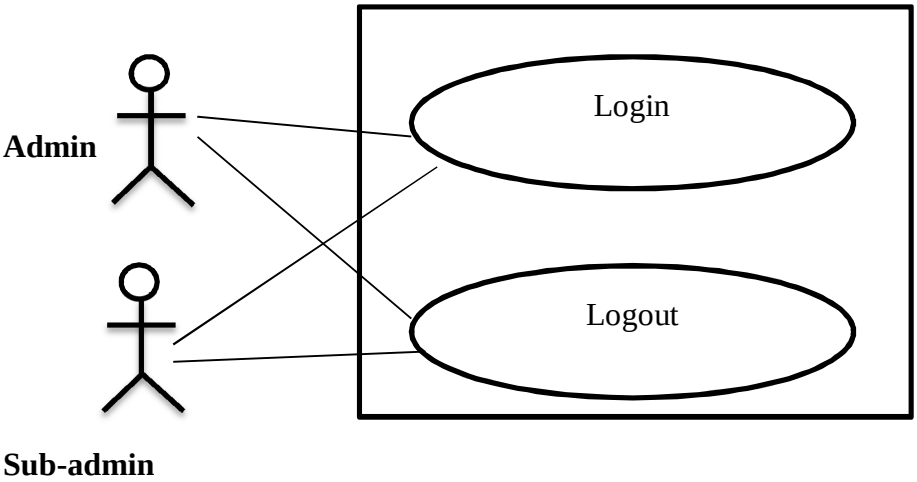
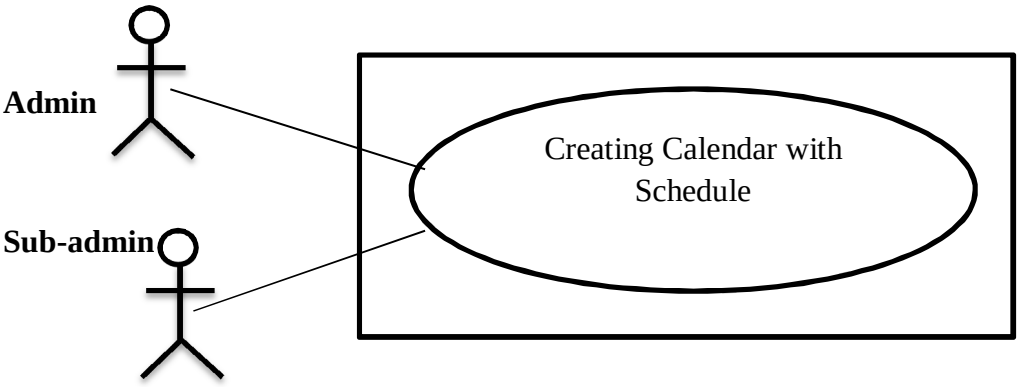


Figure: Level-0 use Case

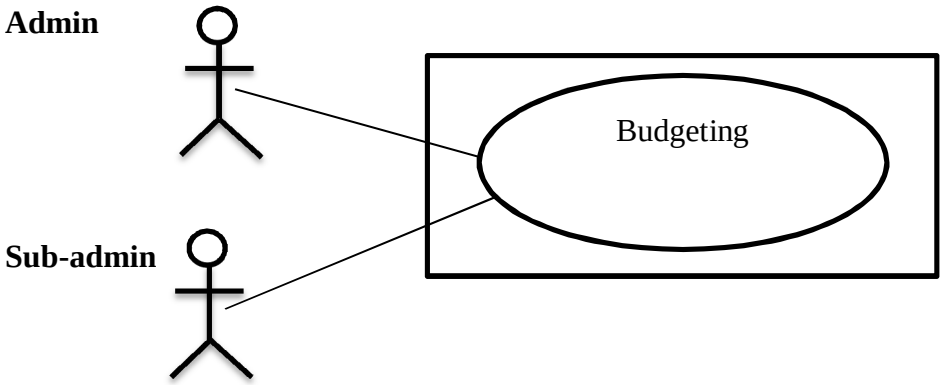
3.3.2.3 Authentication **System**



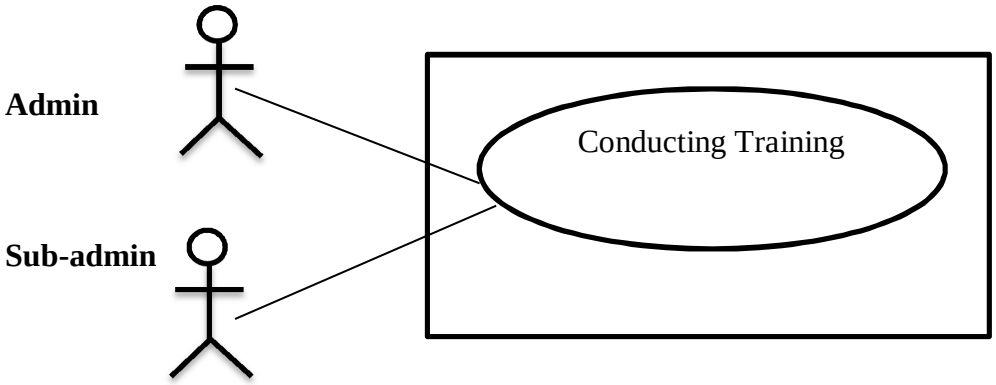
3.3.2.4 Creating Calendar with Schedule



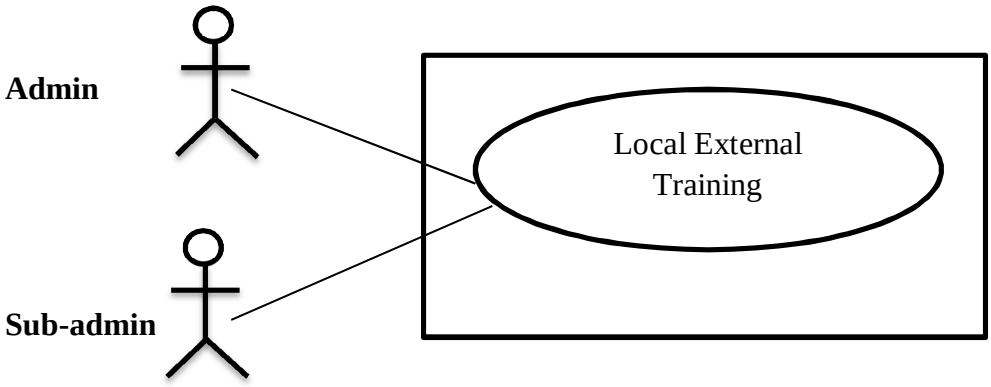
3.3.2.5 Budgeting



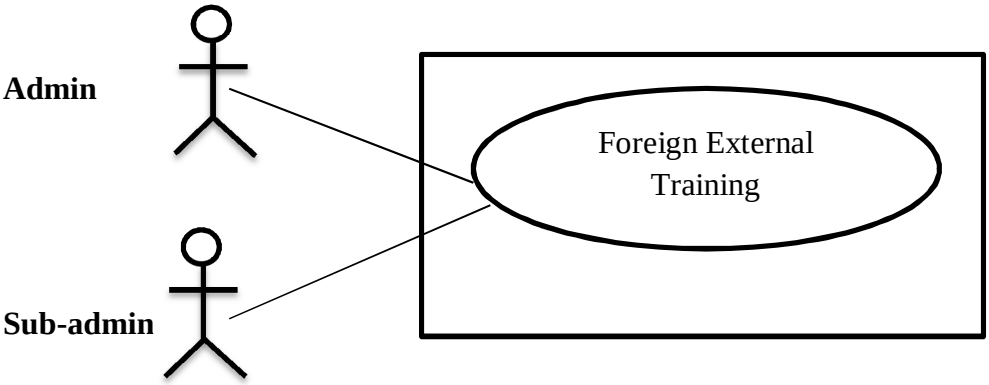
3.3.2.6 Conducting Training



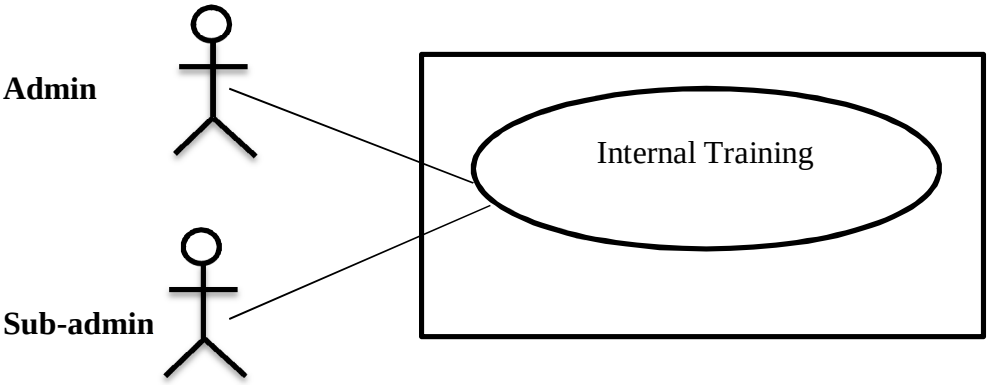
Local External Training



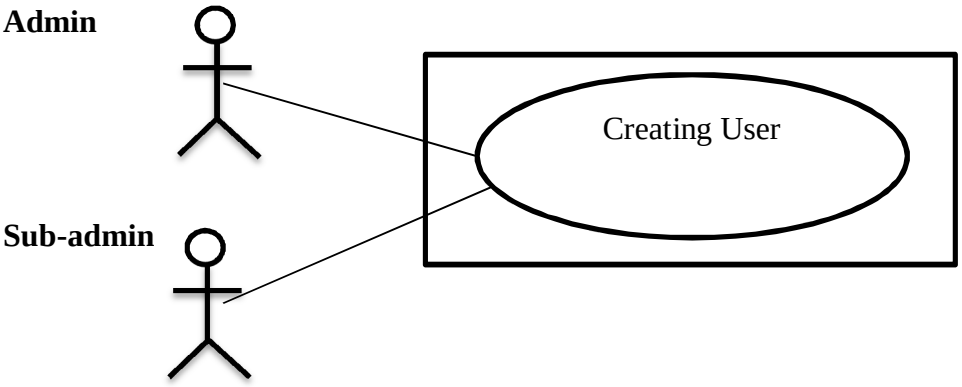
Foreign External Training



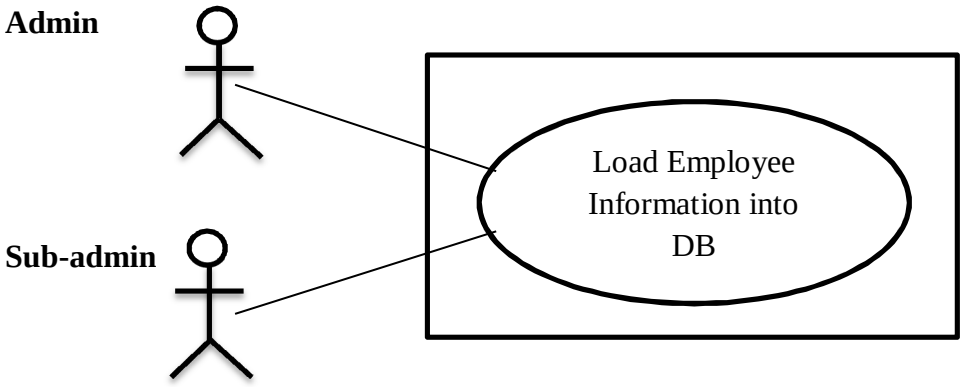
Internal Training



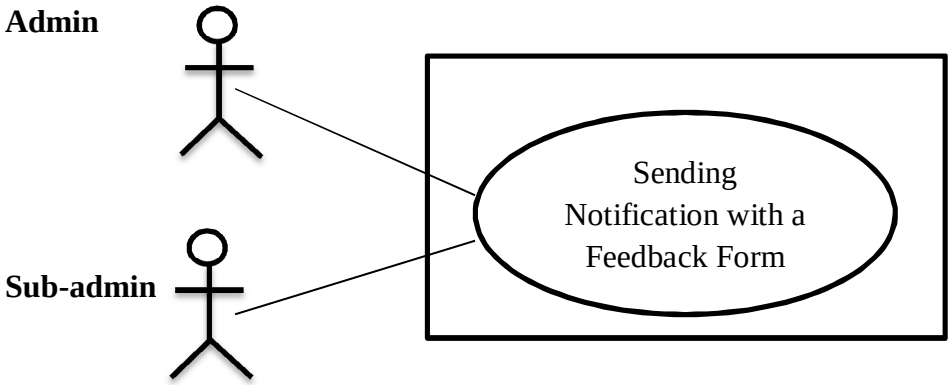
3.3.2.7 Creating User



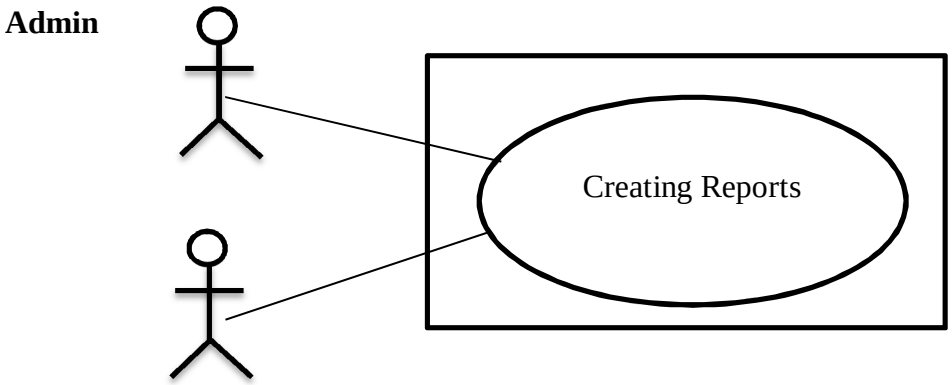
3.3.2.8 Load Employee Information into DB



3.3.2.9 Sending Notification with a Feedback Form to employee

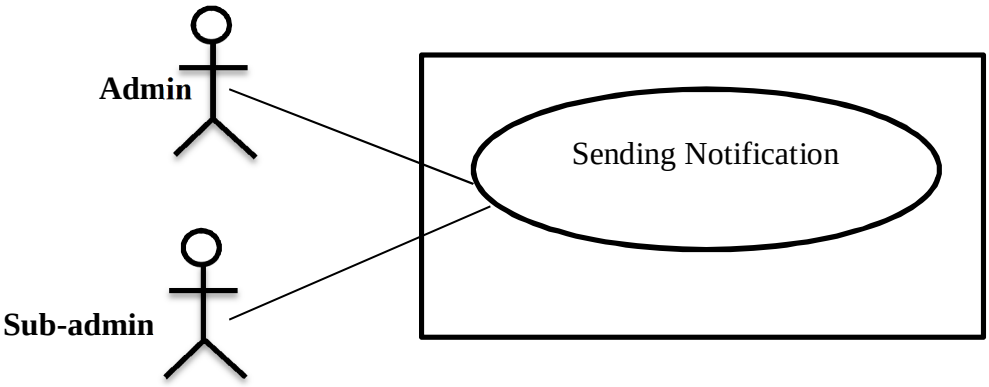


3.3.2.10 Creating Reports

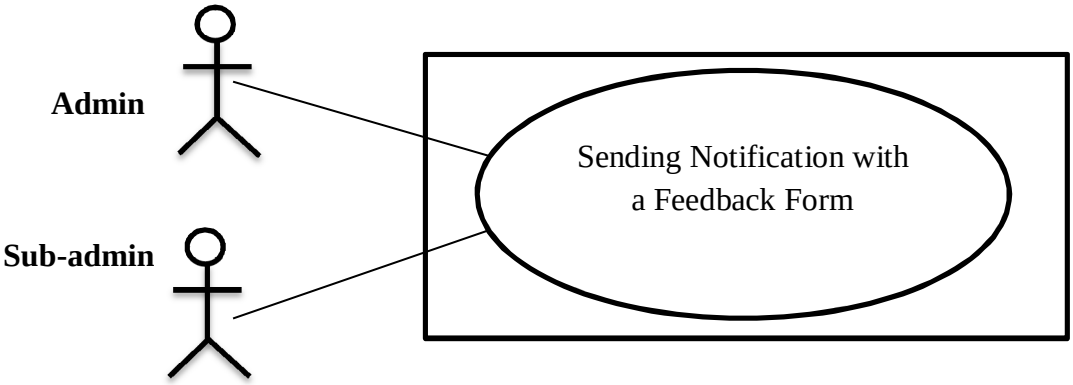


Sub-admin

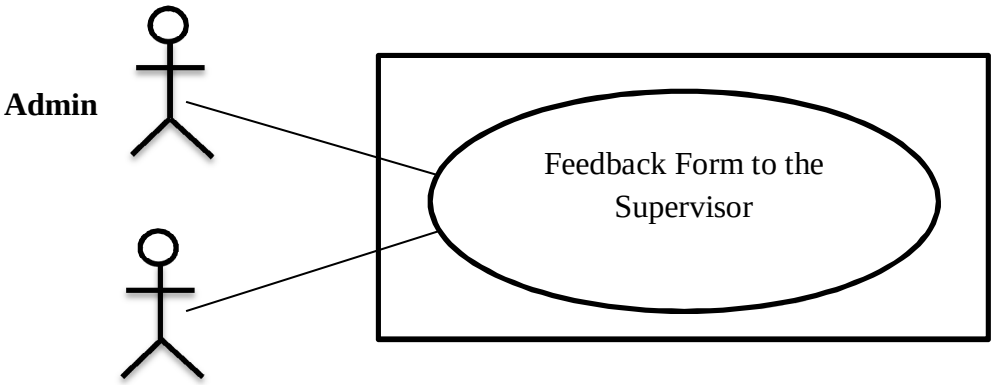
3.3.2.11 Sending Notification



3.3.2.12 Sending Notification with a Feedback Form to supervisor



Feedback Form to the Supervisor



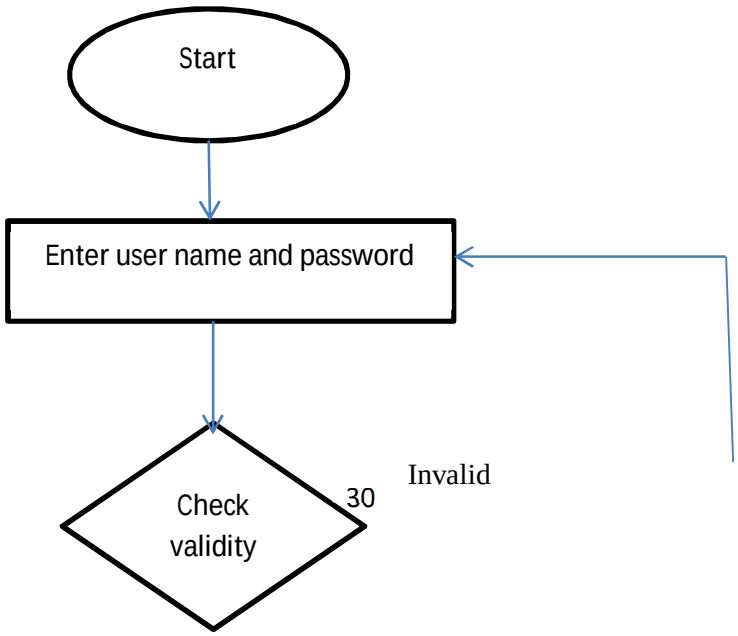
Sub-admin

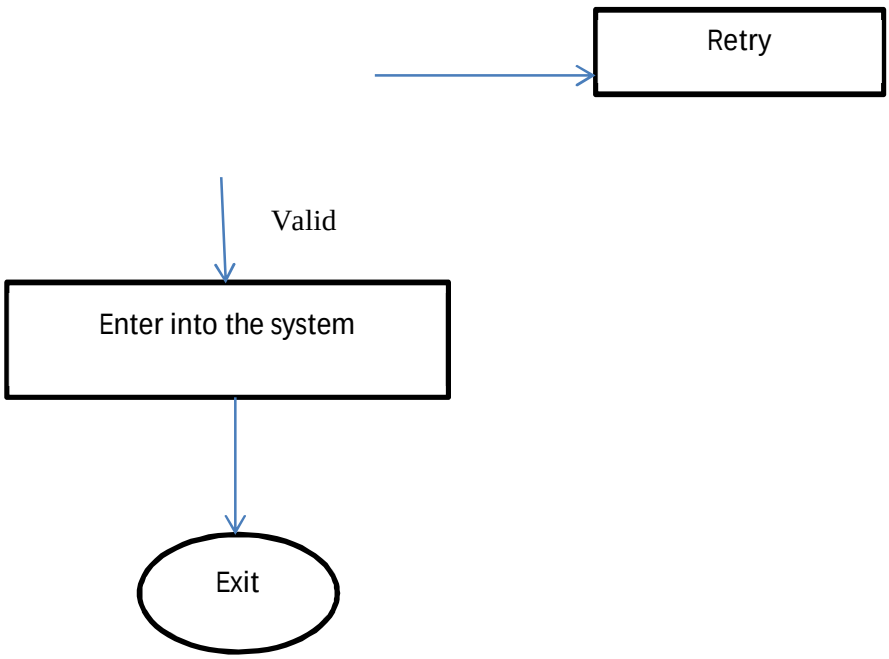
3.3.3 Activity Diagram

The activity diagram is a diagram that shows the activities in a system that is expected to take part. It shows all kinds of possible turnaround of the system.

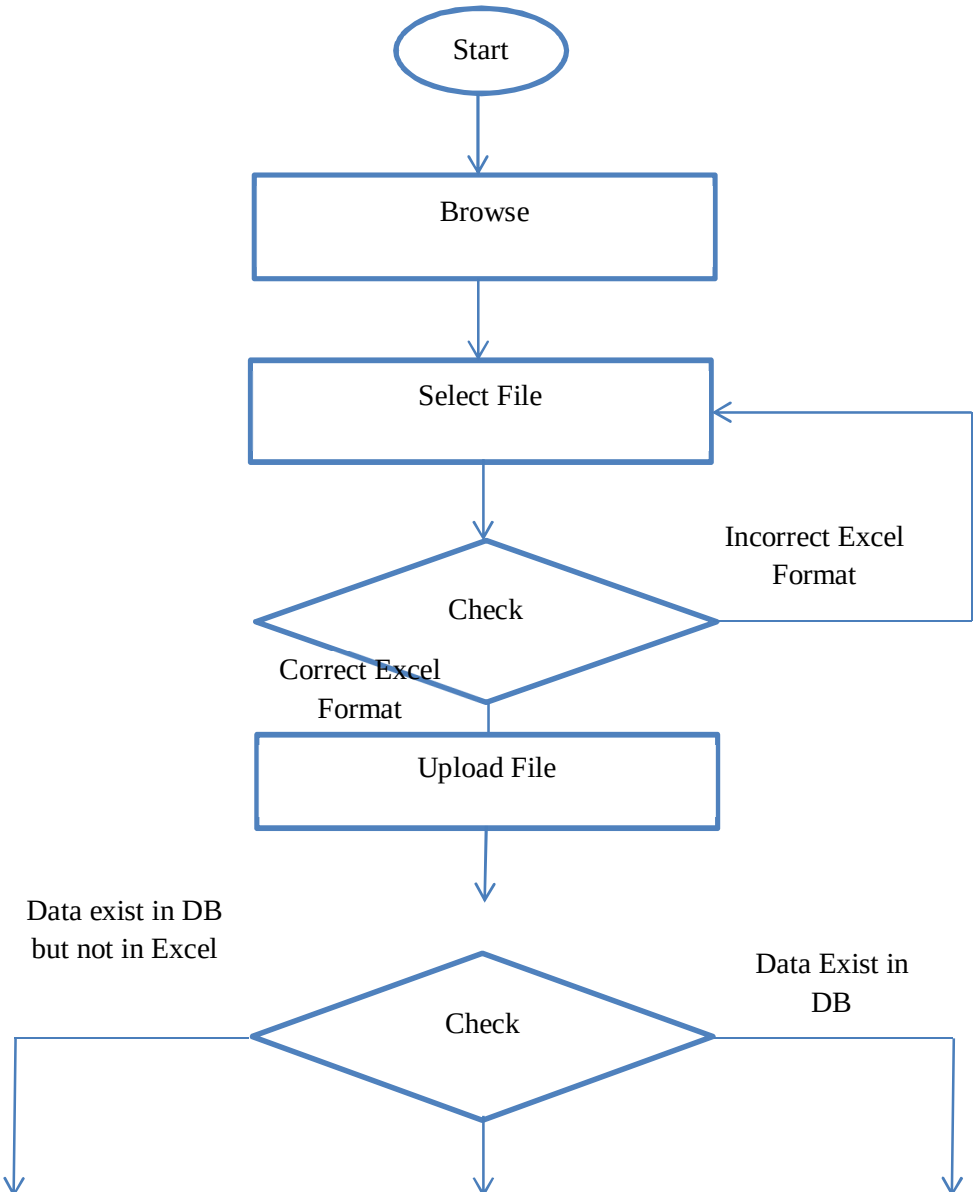
In this diagram there are some special symbols like which means the starting position of the system. After that the symbol means the control flow of the system. The rounded rectangle means the activity nodes where the main operations are done. The symbol denotes the decision node where the decision is made where to go. And finally symbol denotes the end of the process.

3.3.3.1 Authentication System



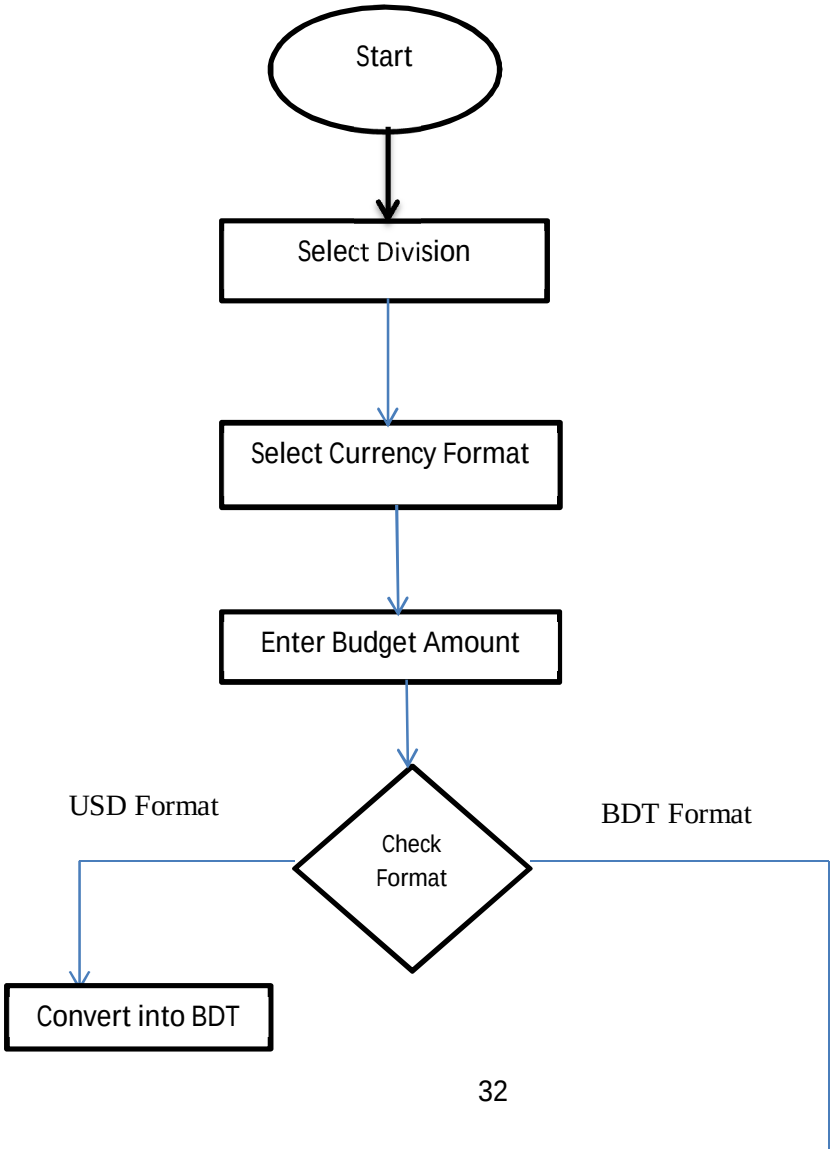


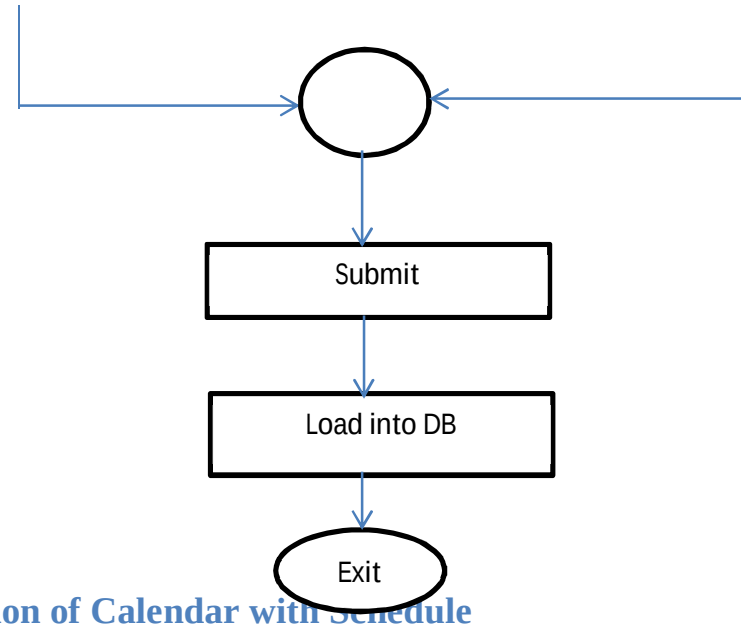
3.3.3.2 Excel File Load in DB



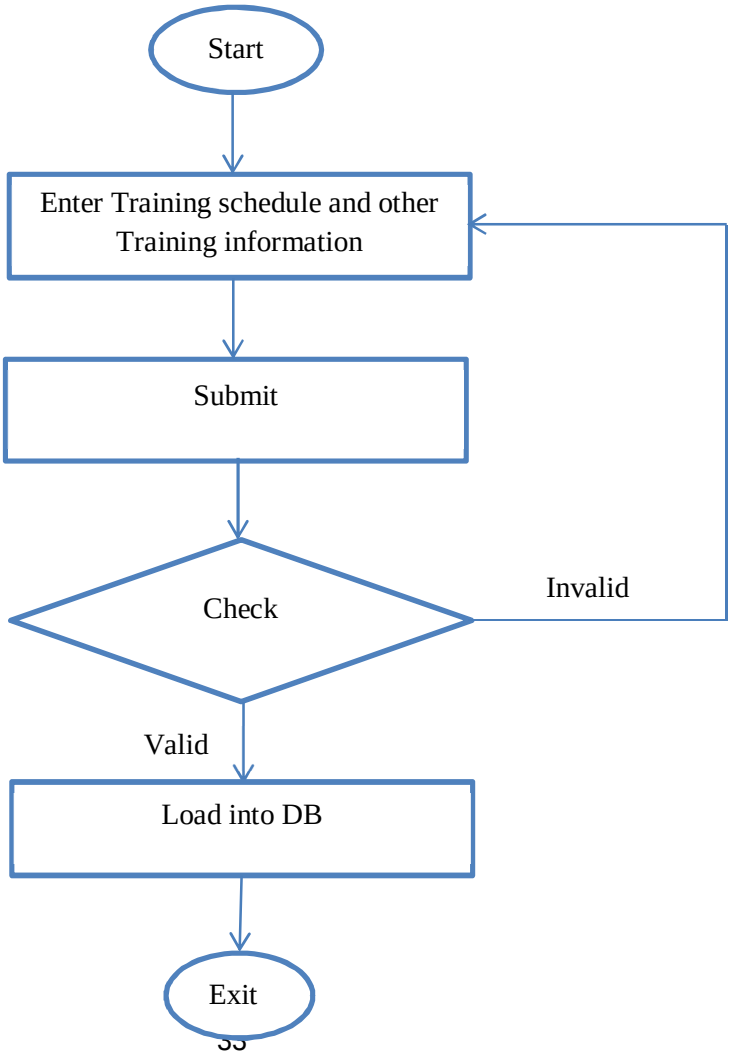
Data do not
exist in DB

3.3.3.3 Budgeting

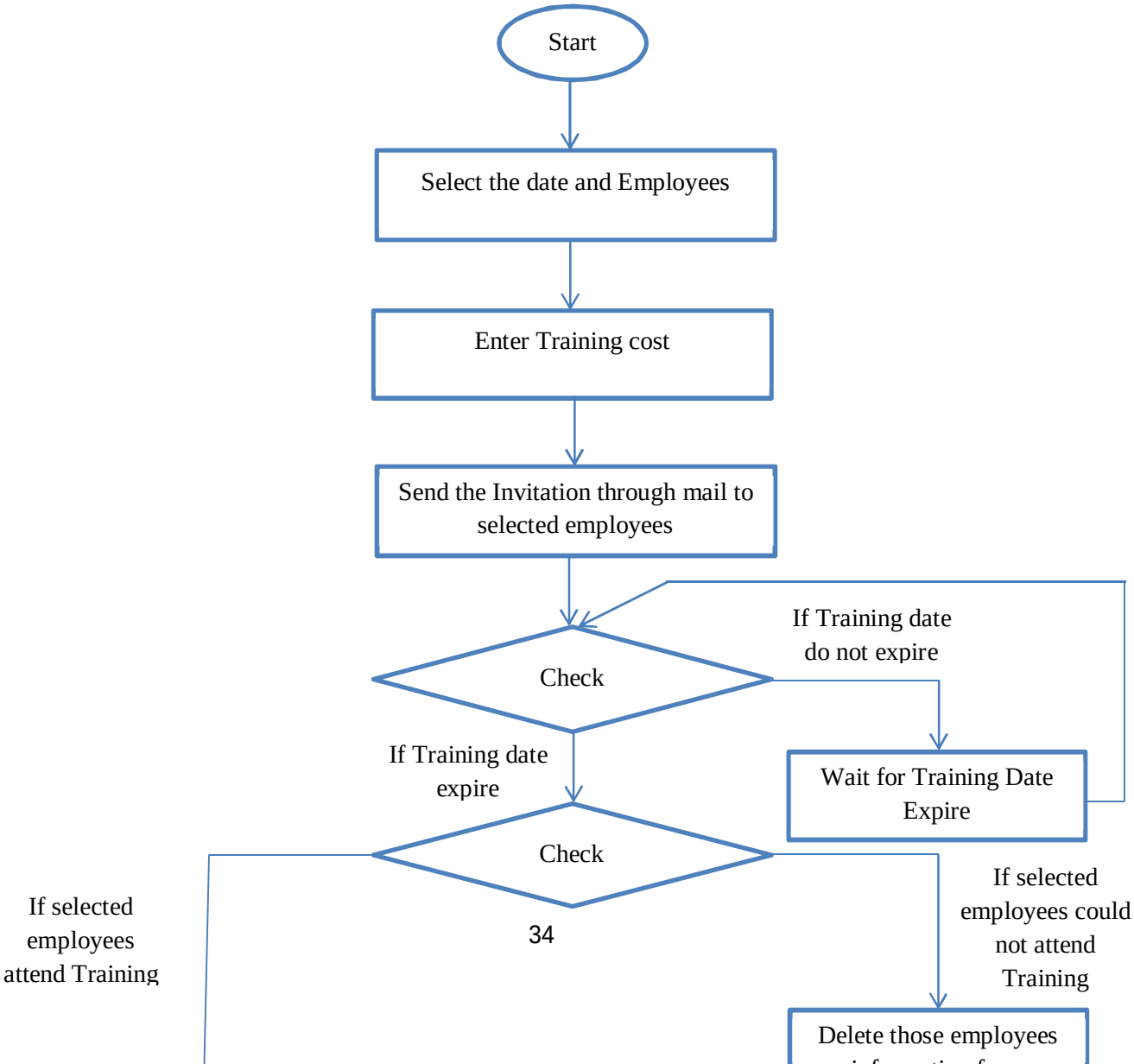




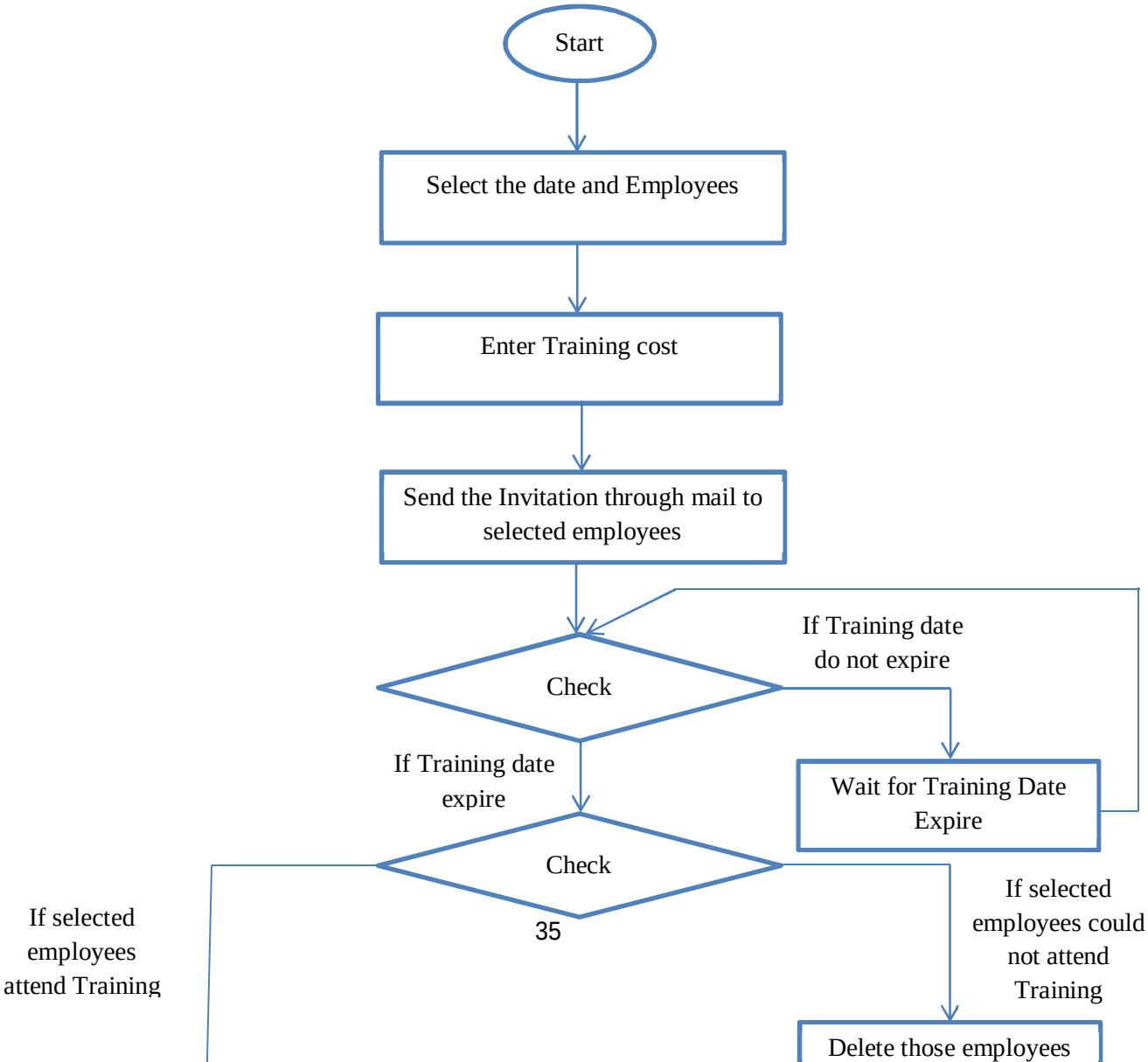
3.3.3.4 Creation of Calendar with Schedule



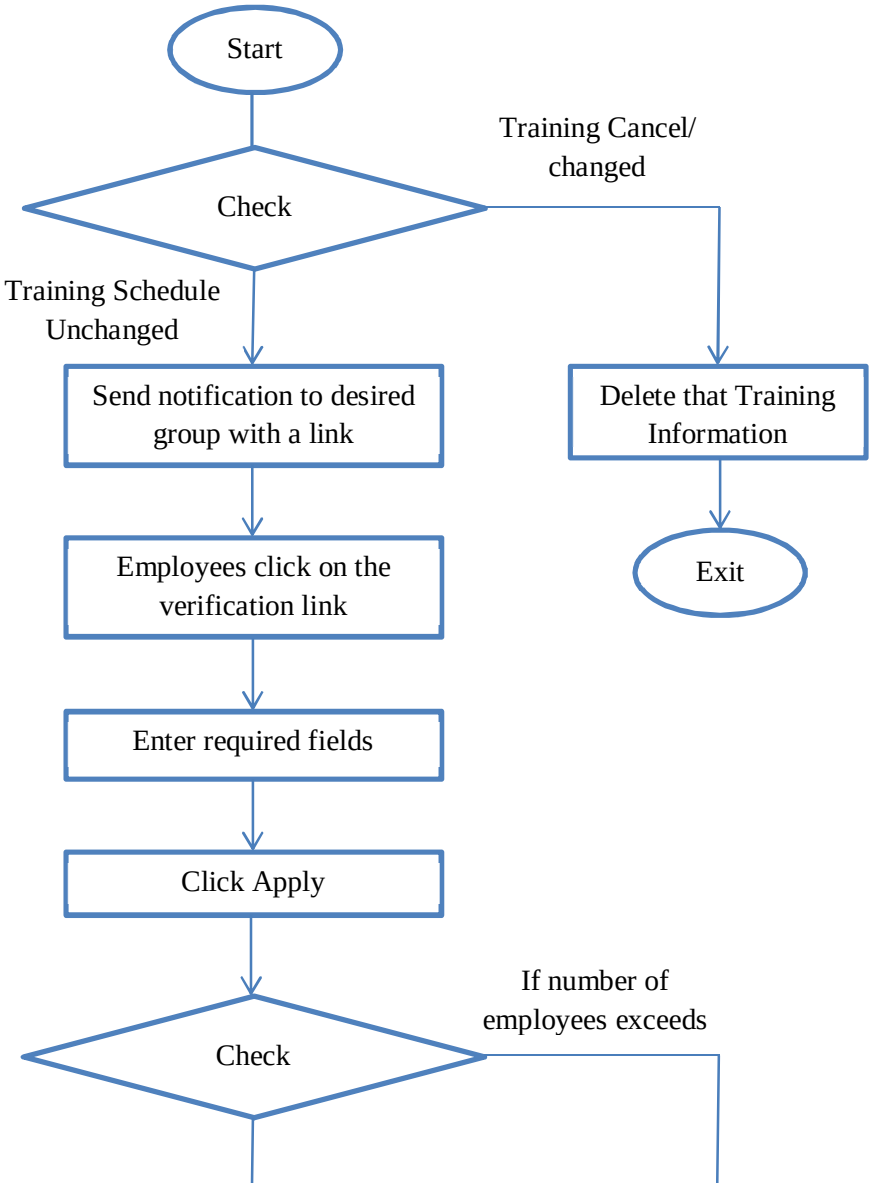
3.3.3.5 Local External Training



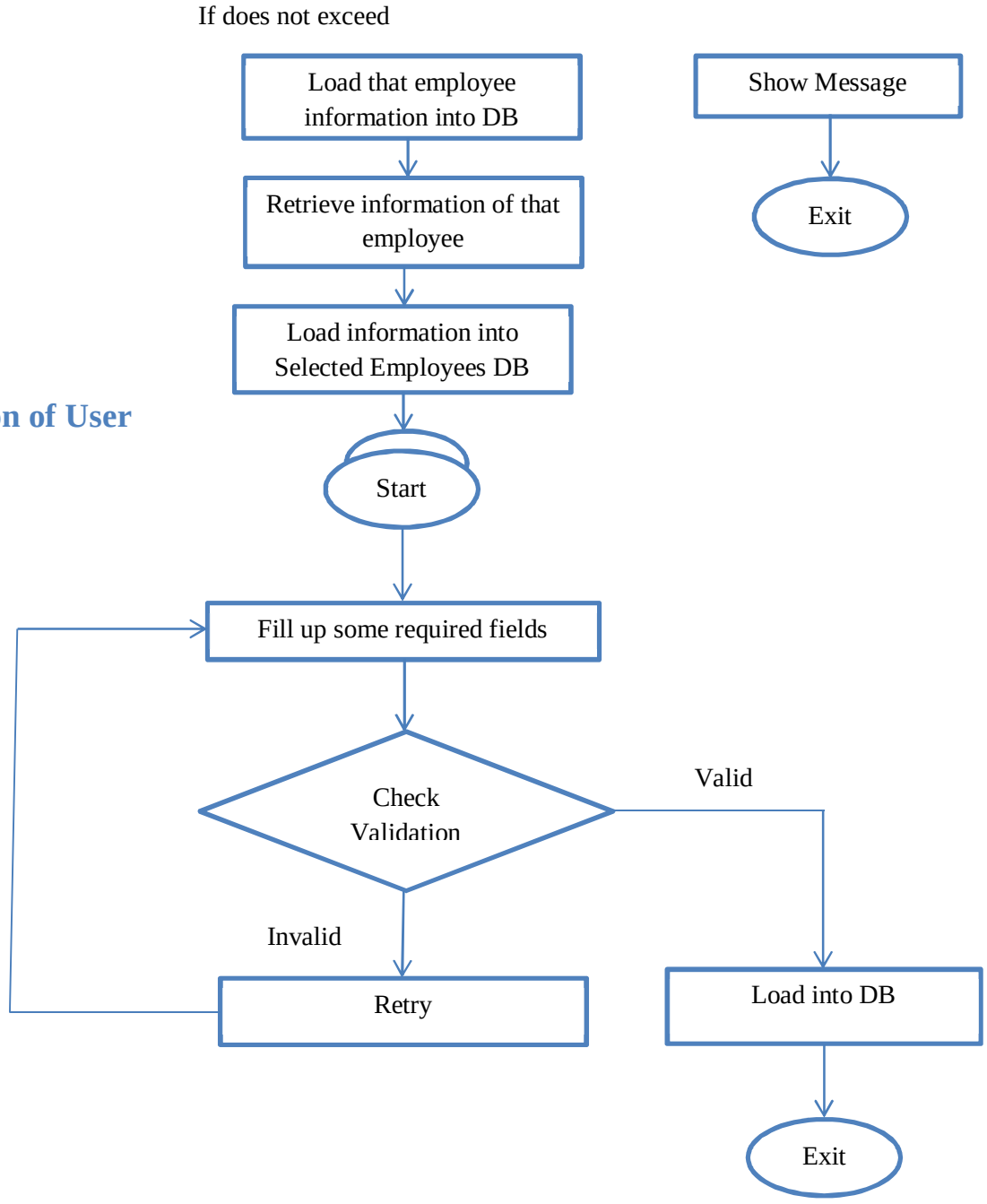
3.3.3.6 Foreign External Training



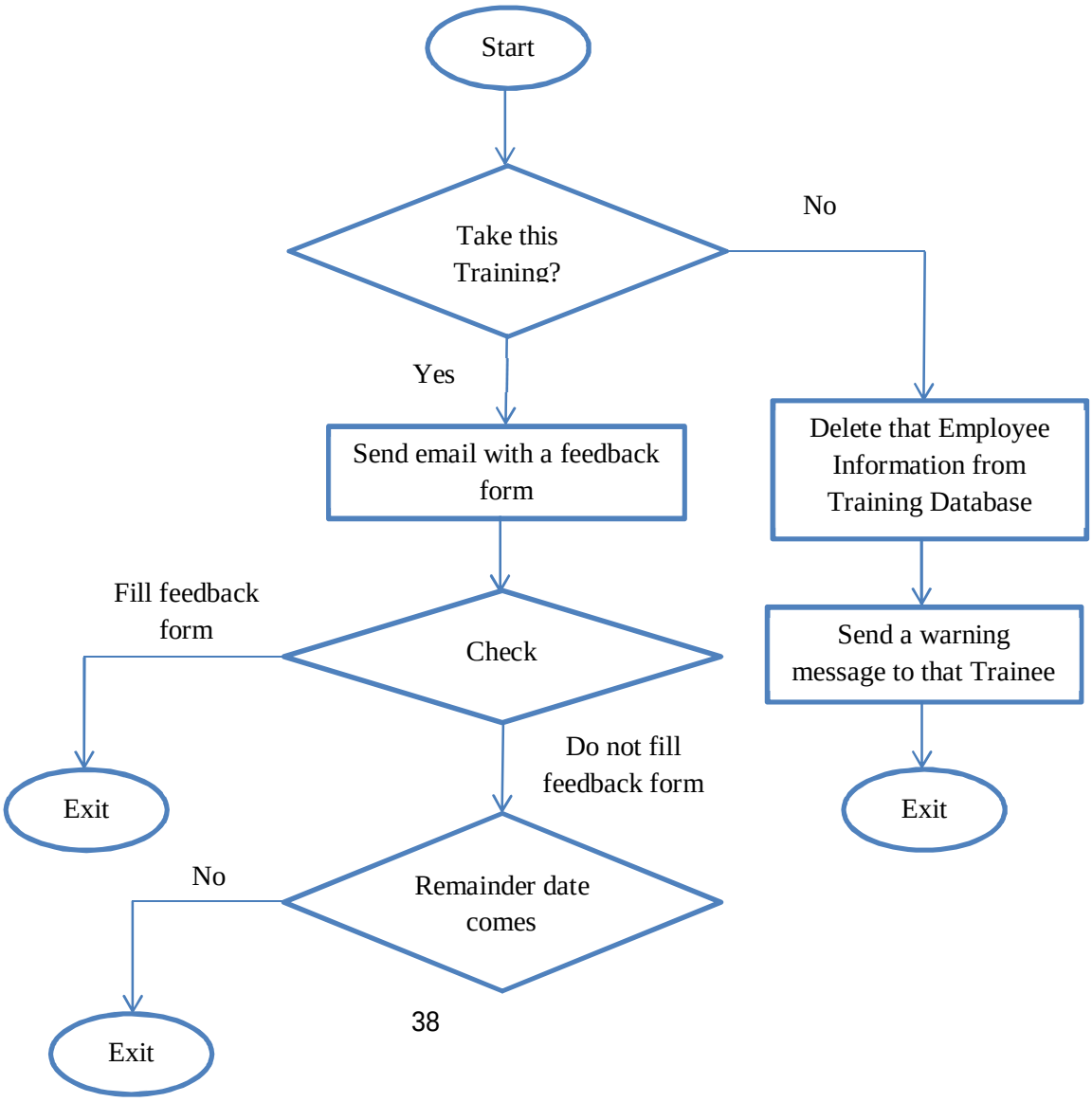
3.3.3.7 Internal Training

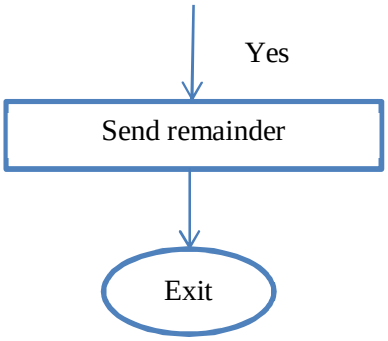


3.3.3.8 Creation of User

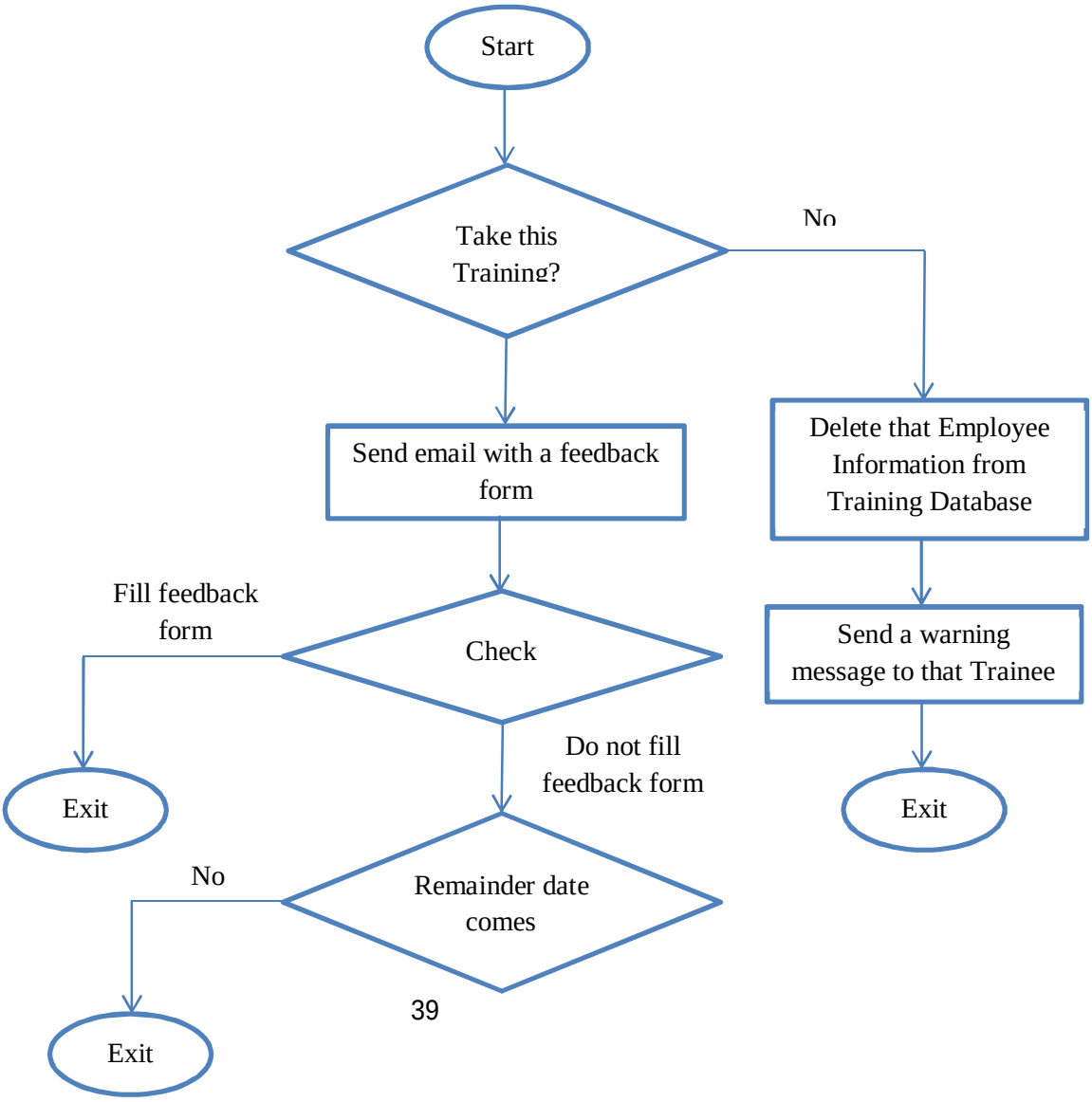


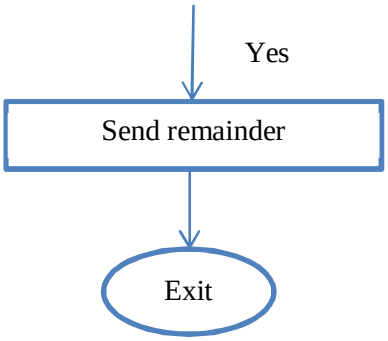
3.3.3.9 Sending notification with a feedback form to Trainees



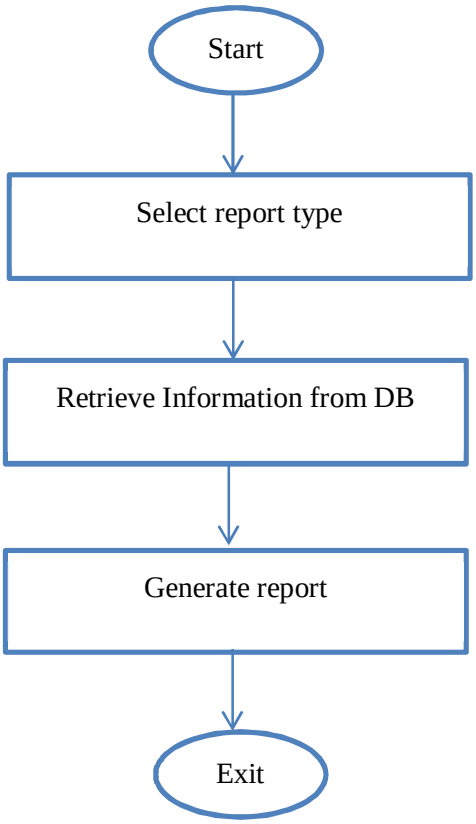


3.3.3.10 Sending notification with a feedback form to Supervisors



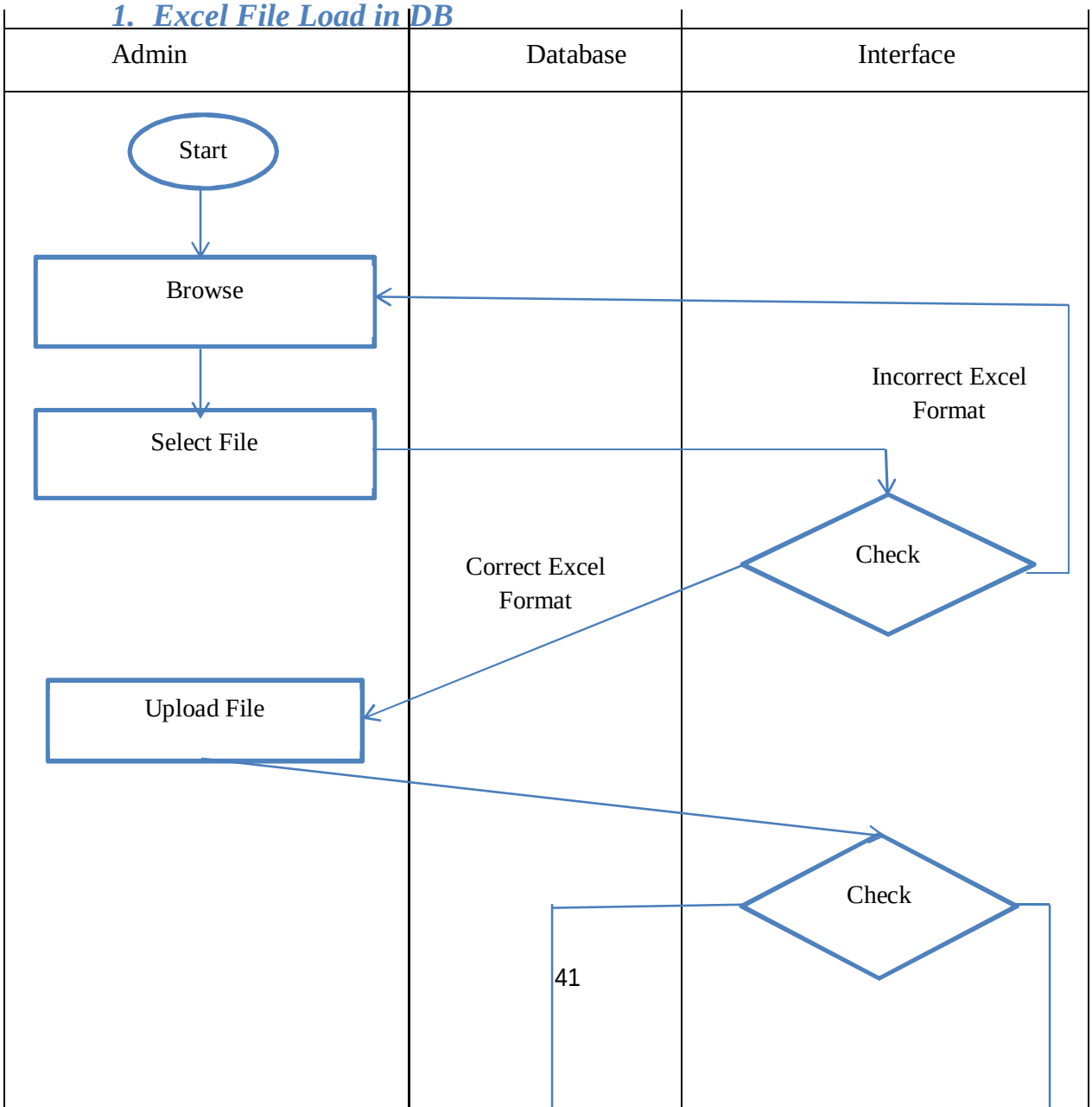


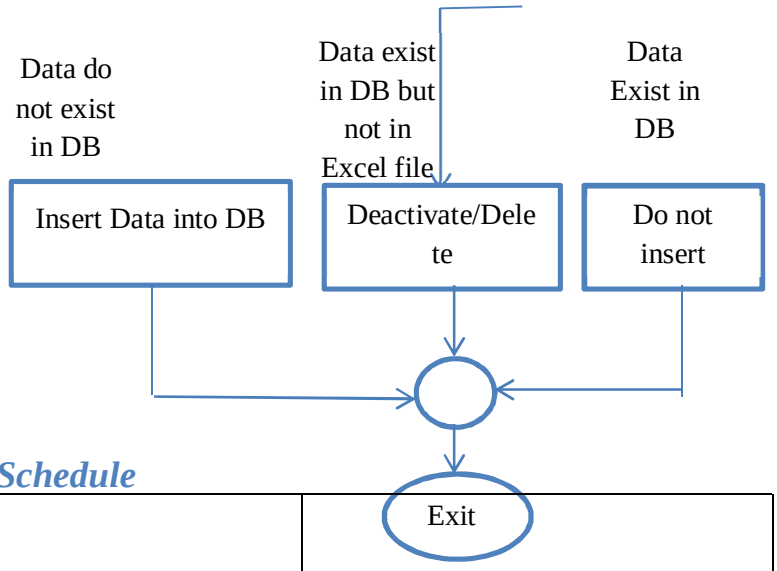
3.3.3.11 Creation of Report



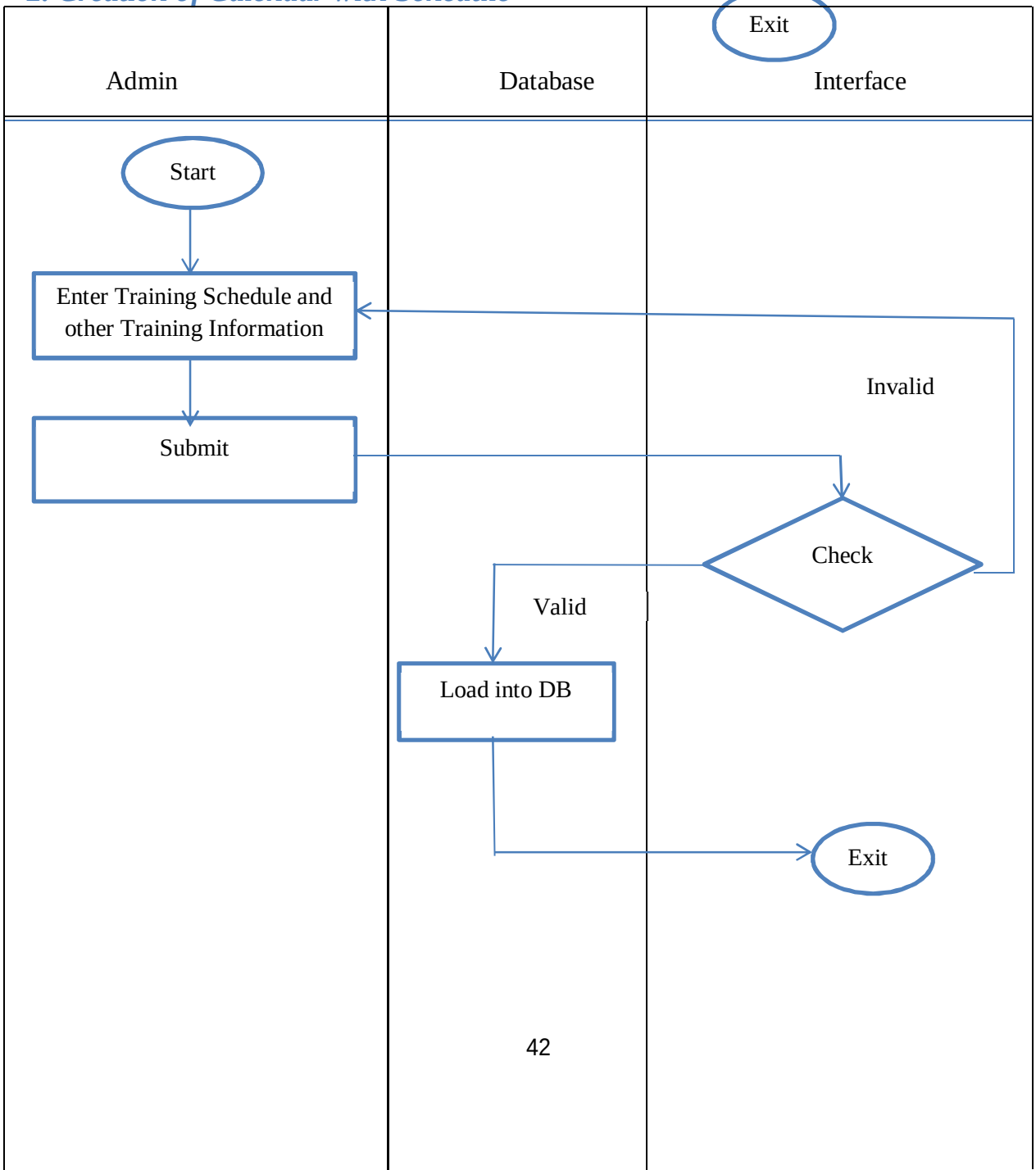
3.3.4 Swim lane Diagram

1. Excel File Load in DB

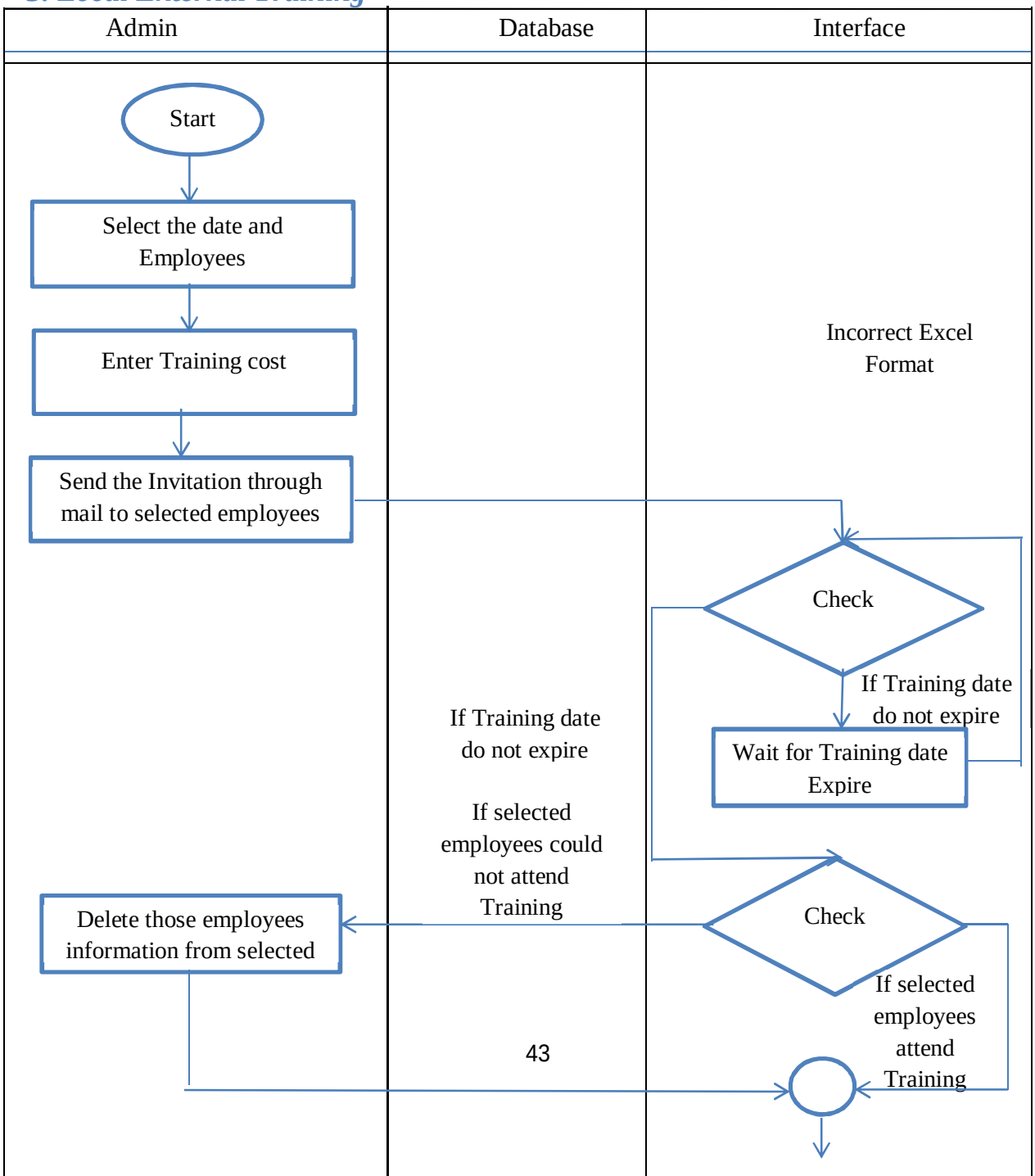


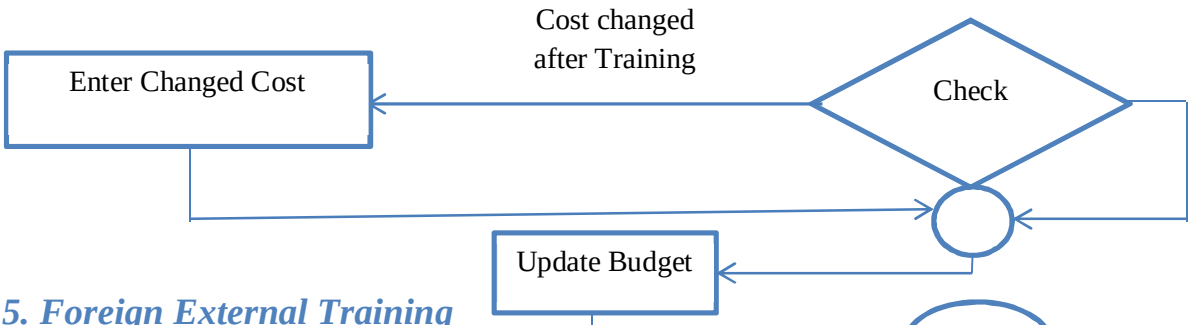


2. Creation of Calendar with Schedule

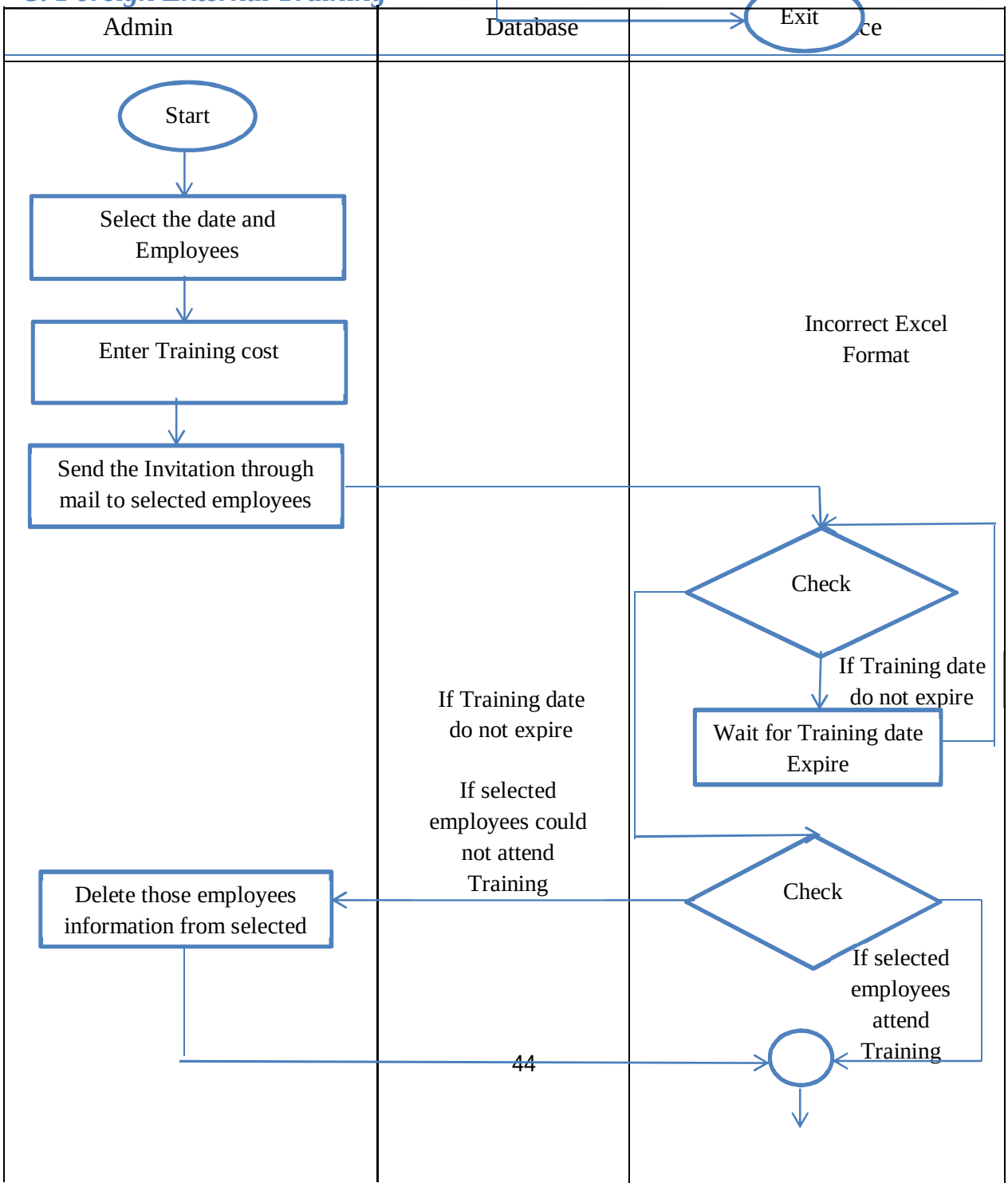


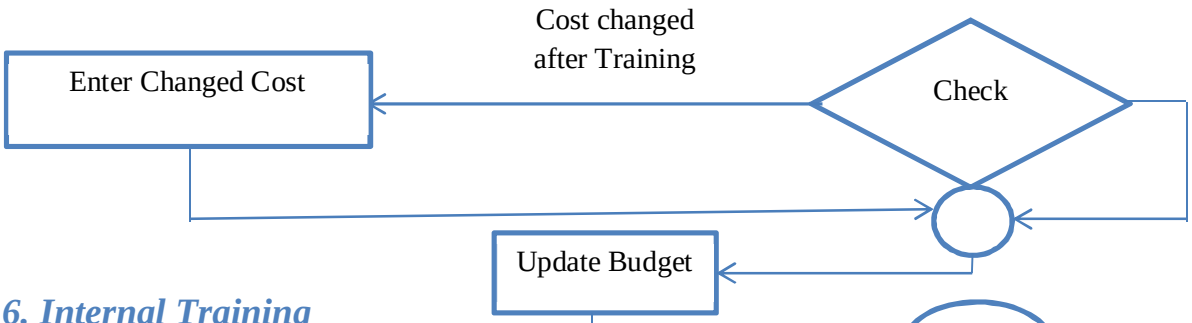
3. Local External Training



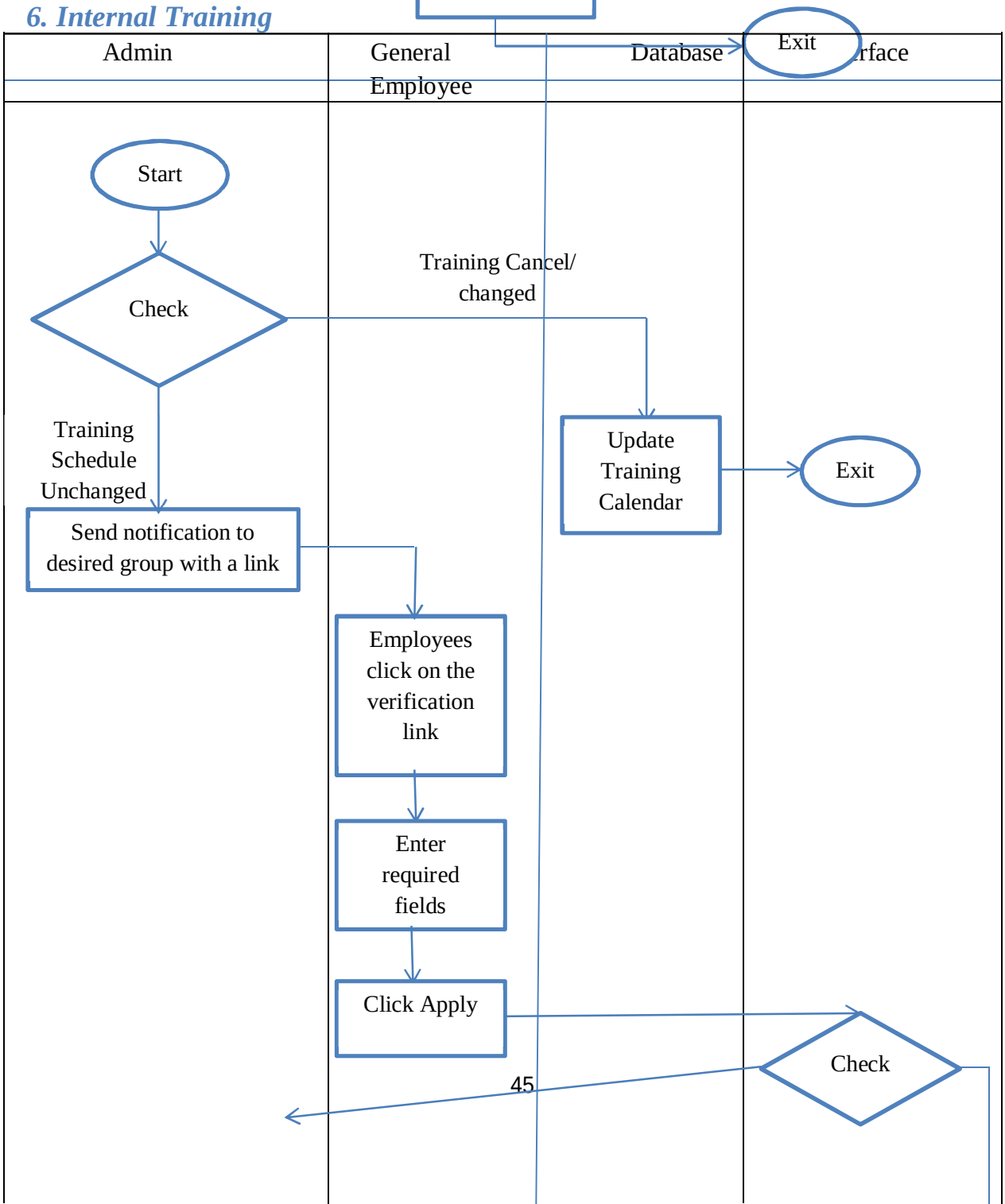


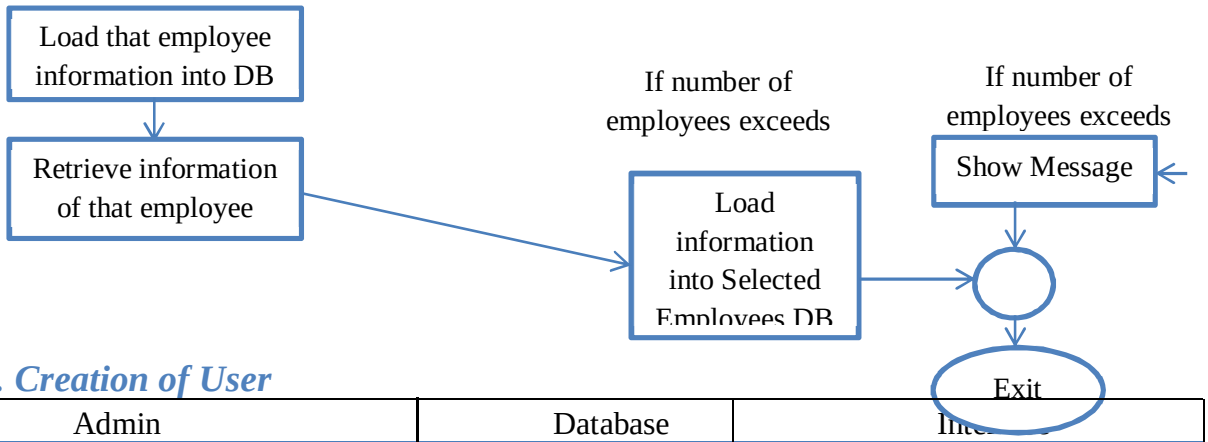
5. Foreign External Training



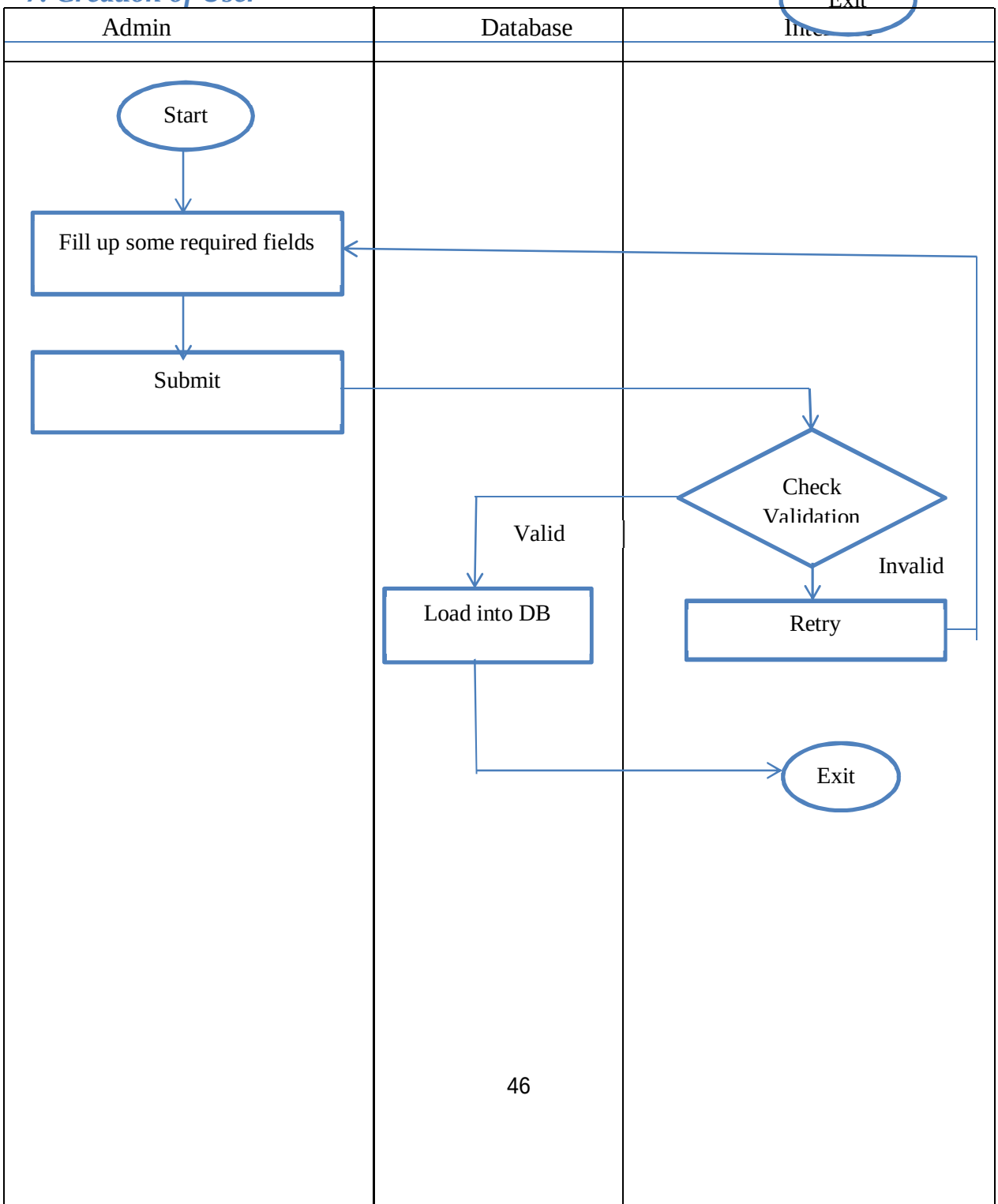


6. Internal Training

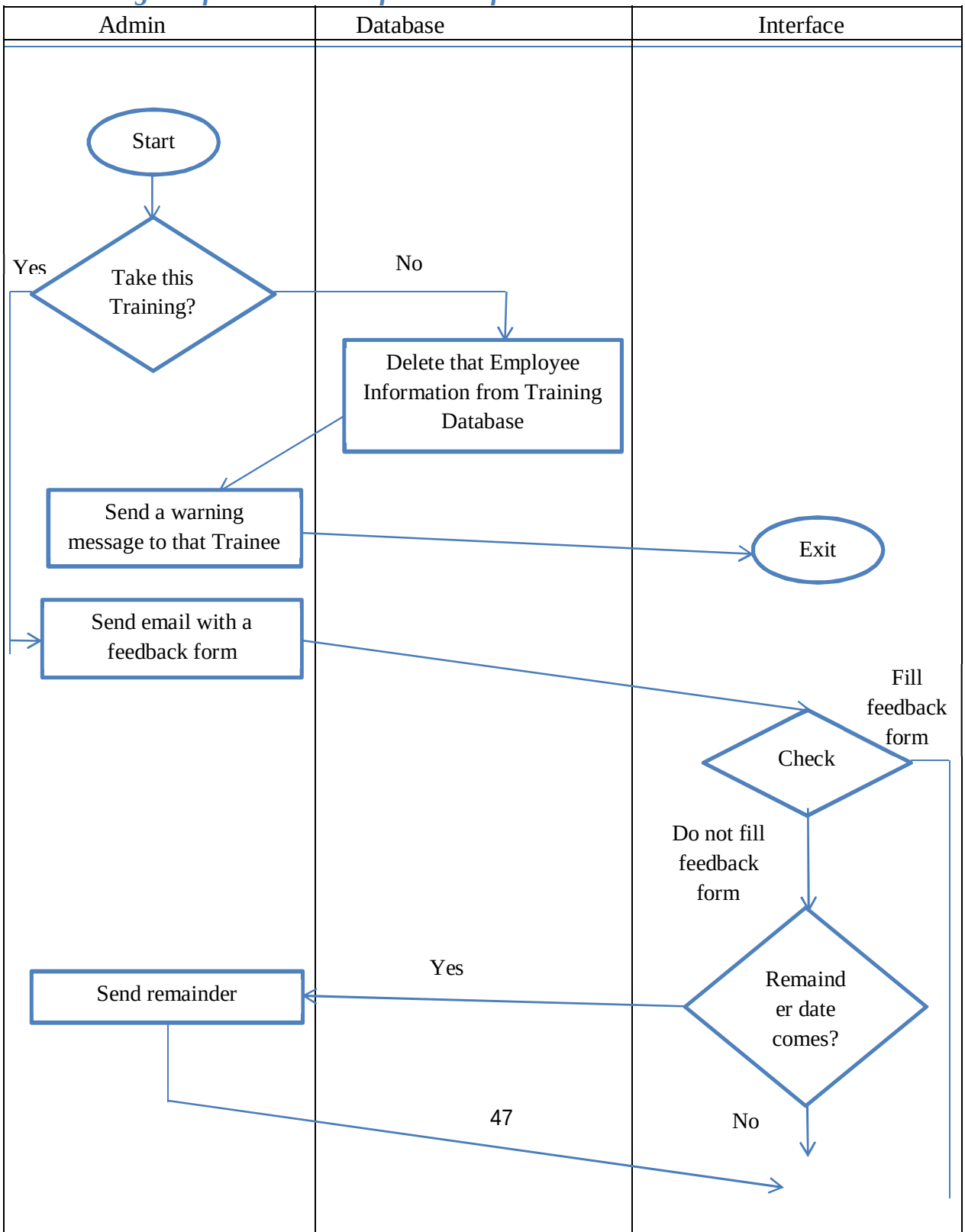


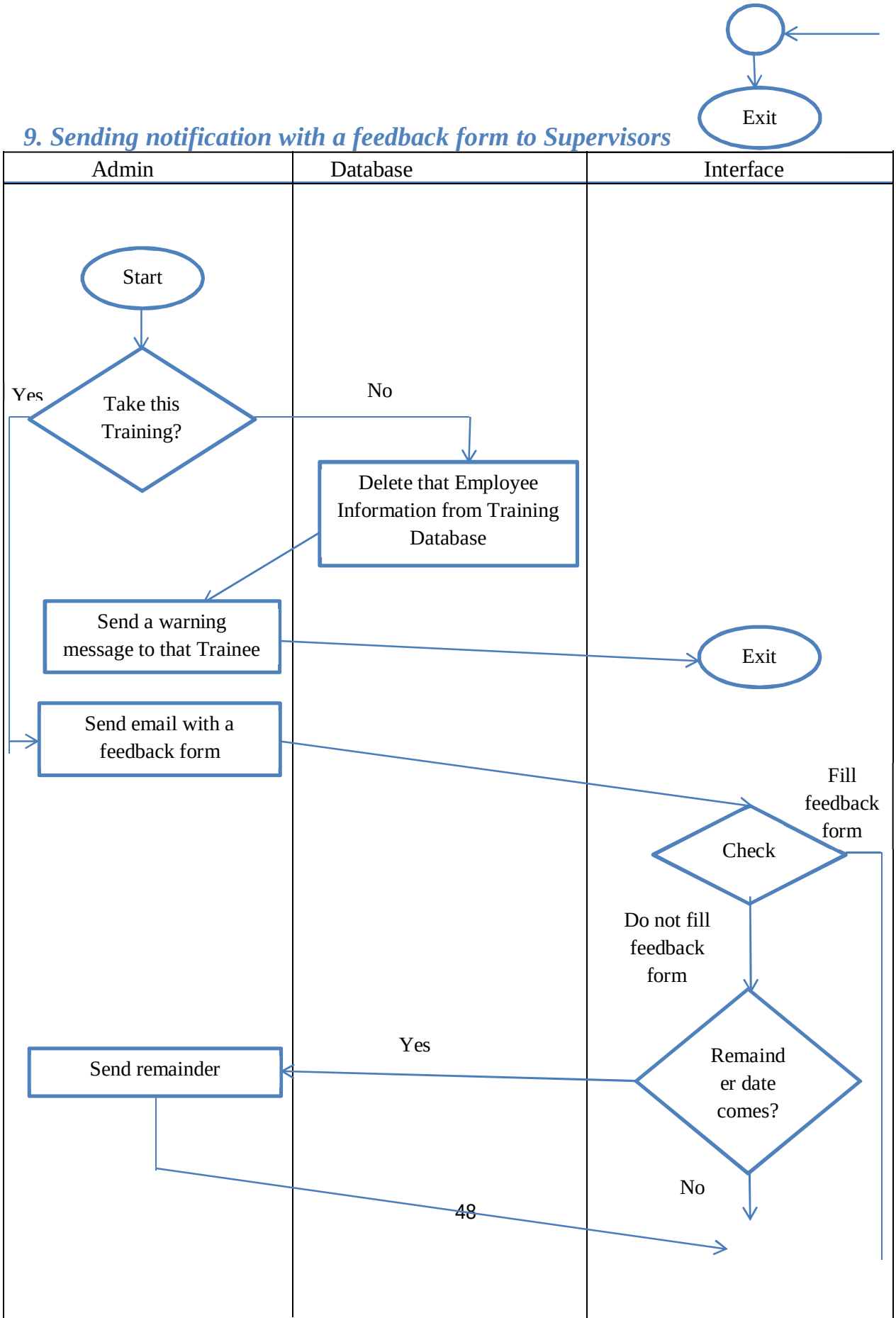


7. Creation of User

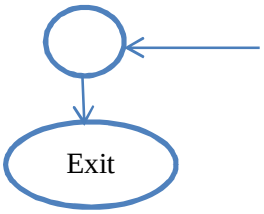


8. Sending notification with a feedback form to Trainees

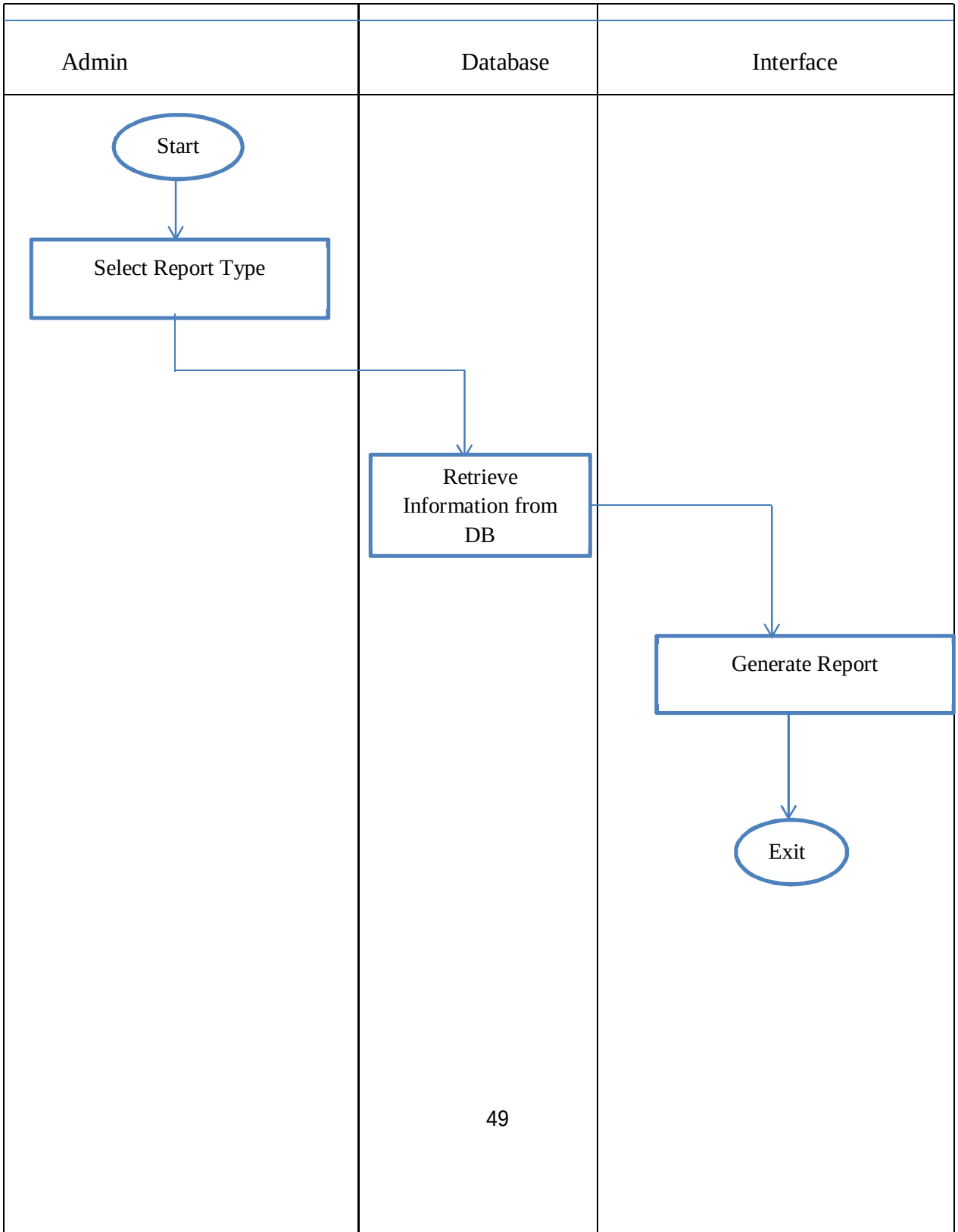




9. Sending notification with a feedback form to Supervisors



10. Creation of Reports



3.3.5 Data Modeling

In data modeling, a software engineer or analyst defines all data objects that are processed within the system, the relationships between the data objects, and other information that is pertinent to the relationships. The *entity-relationship diagram* (ERD) addresses these issues and represents all data objects that are entered, stored, transformed, and produced within an application.

Data object:

A data object is a representation of composite information that must be understood by software.

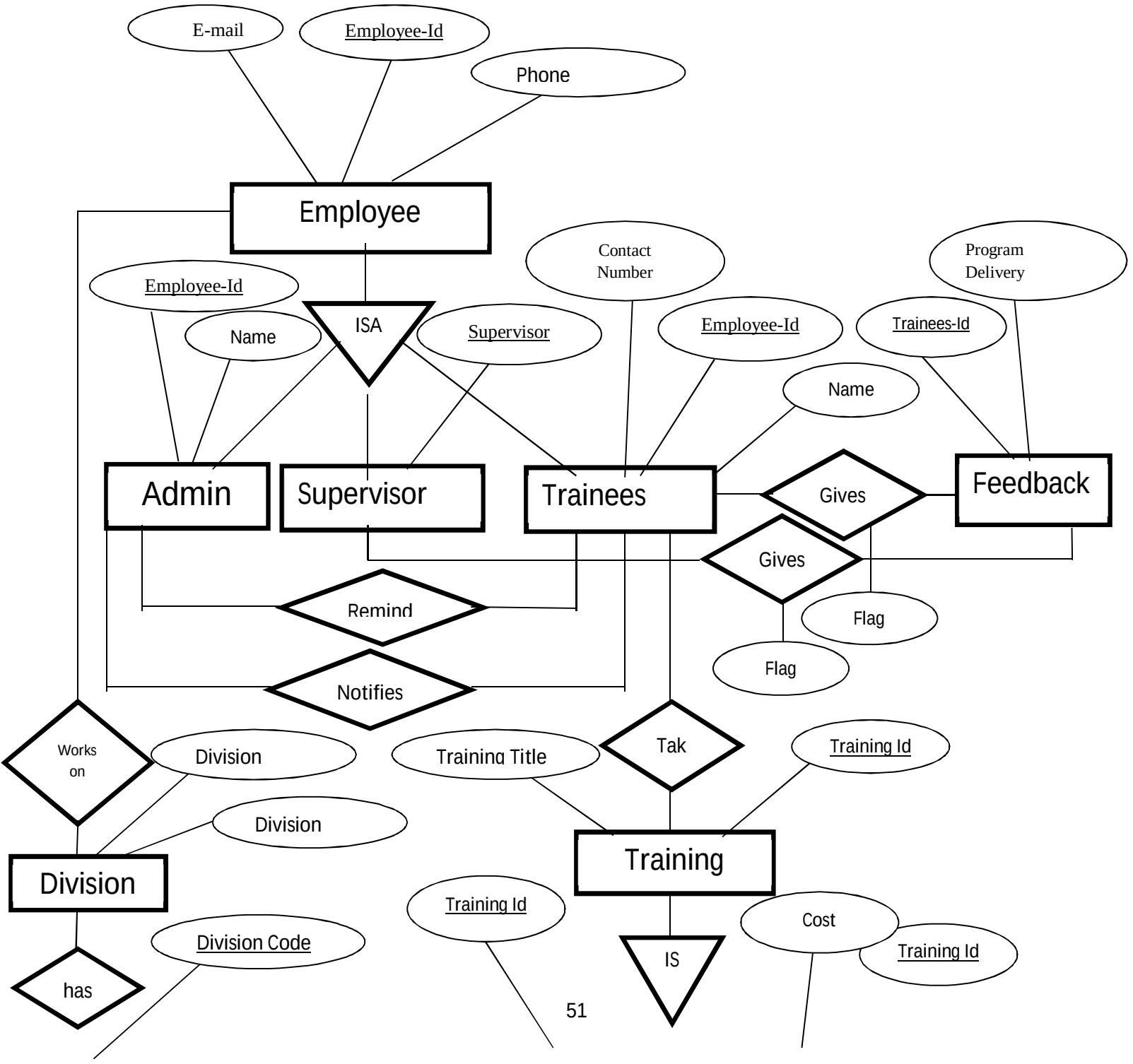
Data in our Use case Scenario

1. Admin
2. Training
3. Feedback Form
4. Report
5. Employee
6. Budget
7. Trainees
8. Supervisors
9. External Training
10. Internal Training
11. Training

Data Attributes:

Data attributes define the properties of a data object and take on one of three different characteristics. They can be used to (1) name an instance of the data object, (2) describe the instance, or (3) make reference to another instance in another table.

3.3.5.1 Entity Relationship Diagram



N.B: Due to insufficient space in page, we cannot give all the attributes.

Fig: E-R Diagram

3.3.5.2 Database Schema

Employee	
1. Employee-Id	Varchar(50)
2. E-mail	Varchar(50)
3. Phone Number	Varchar(50)
4. Employee Name	Varchar(50)
5. Gender	Varchar(50)
6. Date-of-Birth	Varchar(50)
7. Blood Group	Varchar(50)

Admin	
E-mail	Varchar(30)
Password	Varchar(30)
Employee Id	Varchar(30)

Supervisor

Name	Varchar(50)
Employee Id	Varchar(50)
Contact Number	Varchar(50)
E-mail	Varchar(50)
Division	Varchar(50)

Trainees	
Name	Varchar(50)
Contact Number	Varchar(50)
Email	Varchar(50)
Employee Id	Varchar(50)
Trainees-Id	Varchar(50)

Internal Training	
Training Name	Varchar(50)
Employee Id	Varchar(50)
Date	Varchar(50)
Division	Varchar(50)
Training Id	Varchar(50)

External Training

Training Name	Varchar(50)
Employee Id	Varchar(50)
Date	Varchar(50)
Division	Varchar(50)
Training Id	Varchar(50)

Budget	
Division	Varchar(50)
Amount(BDT)	Varchar(50)
Department	Varchar(50)
Section	Varchar(50)
Division Code	

Training	
Priority	Varchar(50)
Training Title	Varchar(50)
Target Group	Varchar(50)
Training Duration	Integer

Training Time	Integer
Manpower Requirement	Varchar(50)
Date	Varchar(50)
Venue	Varchar(50)
Status	Varchar(50)
Training Id	Varchar(50)

Feedback	
Trainees Id	Varchar(50)
Program Delivery	Varchar(50)
Facilitator	Varchar(50)
Facility	Varchar(50)

Sub-Admin	
Name	Varchar(50)
E-mail	Varchar(50)
Employee Id	Varchar(50)

Contact Number	Varchar(50)
User Name	Varchar(50)
Password	Varchar(50)

3.3.6 Class Based Modeling

3.3.6.1 Identifying Analysis Classes

Potential Class		**General Classification**
• Admin	----------	Roles
• Authentication System	----------	Occurrence
• E-mail	----------	Not object, attribute of Admin
• Password	----------	Not object, attribute of Admin
• Operations	----------	Occurrence
• Budget	----------	Occurrence
• Calendar	----------	Things
• Notification	----------	Things
• Feedback Form	----------	Things
• Database	----------	Structure
• Employee Information	----------	Things
• Annual Calendar	----------	External Entities
• Training	----------	Occurrence
• Training Name	----------	Not object, attribute of Training
• Tentative Training Date	----------	Not object, attribute of Training

- Desired Trainee Group ---------- External Entities
- Local External Training ---------- Occurrence
- Foreign External Training ---------- Occurrence
- Internal Training ---------- Occurrence
- Expenses ---------- Occurrence
- Group Member ---------- Roles
- Training date ---------- Not object, attribute of Training
- System ---------- Things
- Selected Trainees ---------- Roles
- Confirmation Link ---------- Things or External Entities
- Allocation ---------- Occurrence
- Seats ---------- Things
- Initial Numbers ---------- Not object, attribute of Trainees
- User ---------- Roles
- Excel File ---------- External Entities
- Message ---------- Things
- Time Slot ---------- Things
- Reports ---------- Things or occurrence
- Pie chart ---------- Things or occurrence
- Bar Diagram ---------- Things
- Line Manager/Supervisor ------- Roles or Organizational Units

Coad and Yourdon suggest six selection characteristics that should be used as you consider each potential class for inclusion in the analysis model:

1. Retained Information
2. Needed Service
3. Common Attributes
4. Multiple Attributes
5. Common Operations
6. Essential Requirements

Potential Class **Characteristic Number That Applies**

- Admin ---------- accepted: all apply

- Authentication System ---------- accepted: all apply
- E-mail ---------- rejected: all fail except 2
- Password ---------- rejected: all fail except 1
- Operations ---------- rejected: 3, 4, 5 fails
- Budget ---------- accepted: all apply
- Calendar ---------- accepted: 1, 2, 3, 4 apply
- Notification ---------- accepted: 1, 2, 3, 6 apply
- Feedback Form ---------- accepted: 1,2 , 3, 4, 6 apply
- Database ---------- accepted
- Employee Information ---------- accepted: 1, 2, 3, 4, 6 apply
- Annual Calendar ---------- accepted: 1, 2, 3, 4, 6 apply
- Training ---------- accepted: 1, 2, 3, 6 apply
- Training Name ---------- rejected
- Tentative Training Date ---------- rejected
- Desired Trainee Group ---------- rejected
- Local External Training ---------- accepted: all apply
- Foreign External Training ---------- accepted: all apply
- Internal Training ---------- accepted: all apply
- Expenses ---------- rejected
- Group Member ---------- rejected
- Training date ---------- rejected
- System ---------- rejected
- Selected Trainees ---------- accepted: all apply
- Confirmation Link ---------- rejected
- Allocation ---------- rejected
- Seats ---------- rejected
- Initial Numbers ---------- rejected
- User ---------- accepted: all apply
- Excel File ---------- rejected
- Message ---------- accepted: 1, 2, 3, 4 apply (Redundant on Notification)
- Time Slot ---------- rejected
- Reports ---------- accepted: all apply

- Pie chart ---------- accepted: 1, 2, 3, 4, 5 apply
- Bar Diagram ---------- accepted: 1, 2, 3, 4, 5 apply
- Line Manager/Supervisor -------- accepted: all apply

3.3.6.2 UML Class Diagram

Annual Calendar	
Attributes	Methods
1) Priority 2) Training Title 3) Target Group 4) Training Duration 5) Training Time 6) Manpower Requirement 7) Data 8) Venue 9) Status	1. Insert Values() 2. Update Calendar() 3. Delete Training() 4. Edit Training() Admin, Database.

Feedback Form	
Attributes	Methods
1) Program Delivery 2) Facilitator 3) Facility 4) Total Score	1. CalcutateTotalScore() 2. ShowResults() 3. InsertValueIntoDB()

5) Overall Rating 6) Comments	Selected Employees, Reports, Pie Chart, Bar Diagram.

Employee Information

Attributes	Methods
1) Employee Number 2) Employee Name 3) Date of Birth 4) Gender 5) Contact Number 6) Date of Join 7) Division 8) Division Code 9) Department 10) Designation 11) E-Mail 12) Line Manager	1. InsertValue() 2. UpdateValue() Sending Notification, Database.

13) Department Code	

Admin	
Attributes	Methods
1. Name 2. E-mail 3. Phone Number 4. Password 5. Employee Id	1. LoginSystem() 2. CreatingUser() 3. Budgeting() 4. Creating Calendar() 5. Sending Notification() 6. Loading Excel File()
	Budgeting, Creating calendar, Sending Notification, Generating Reports, Authentication.

Budgeting	
Attributes	Methods
1. Initial Budget Amount 2. Expense 3. Updated Amount 4. Division Name 5. BDT Format 6. USD Format	1. Insert Budget() 2. Update Budget() 3. Convert into any format()
	Admin, Database.

Notification	
Attributes	Methods
1) E-Mail 2) Content 3) Attached Files	1. SendMail() Selected Trainees, Database.

Internal Training	
Attributes	Methods
1) Training Name 2) E-Mail 3) Employee Number 4) Contact Number 5) Division 6) Confirmation Link	1. RetriveEmployeeInfo() 2. InsertEmployeeInfo() 3. SendNotification() 4. SendFeedbackForm() Notification, Database, Selected trainees.

External Training	
Attributes	Methods
1) Training Name 2) Budget Amount 3) E-Mail 4) Employee Number 5) Division 6) Contact Number	5. RetriveEmployeeInfo() 6. InsertEmployeeInfo() 7. SendNotification() 8. SendFeedbackForm()

Selected Trainees	
Attributes	Methods
1) Name 2) ID 3) Division 4) Contact Number	1. SendingNotification() 2. SendingFeedbackForm() Notification, Sending feedback form, Database.

Month Wise Training Calendar	
Attributes	Methods
1) Training Name 2) Date 3) Division	1. InsertSchedule() 2. EditSchedule() 3. ShowTrainingName() Annual calendar, Database.

Reports	
Attributes	Methods
1) Report Type 2) Report Name	1. GenerateReport()
	Admin, Feedback form, PieChart, BarDiagram.

Bar Diagram	
Attributes	Methods
1. X axis 2. Y axis 3. Height 4. width	1. DrawBarDiagram()
	Reports, feedback form.

User Creation	
Attributes	Methods
1) Employee ID 2) E-mail 3) Contact Number 4) Name 5) Division	1. CreateUser()
	Admin, Database.

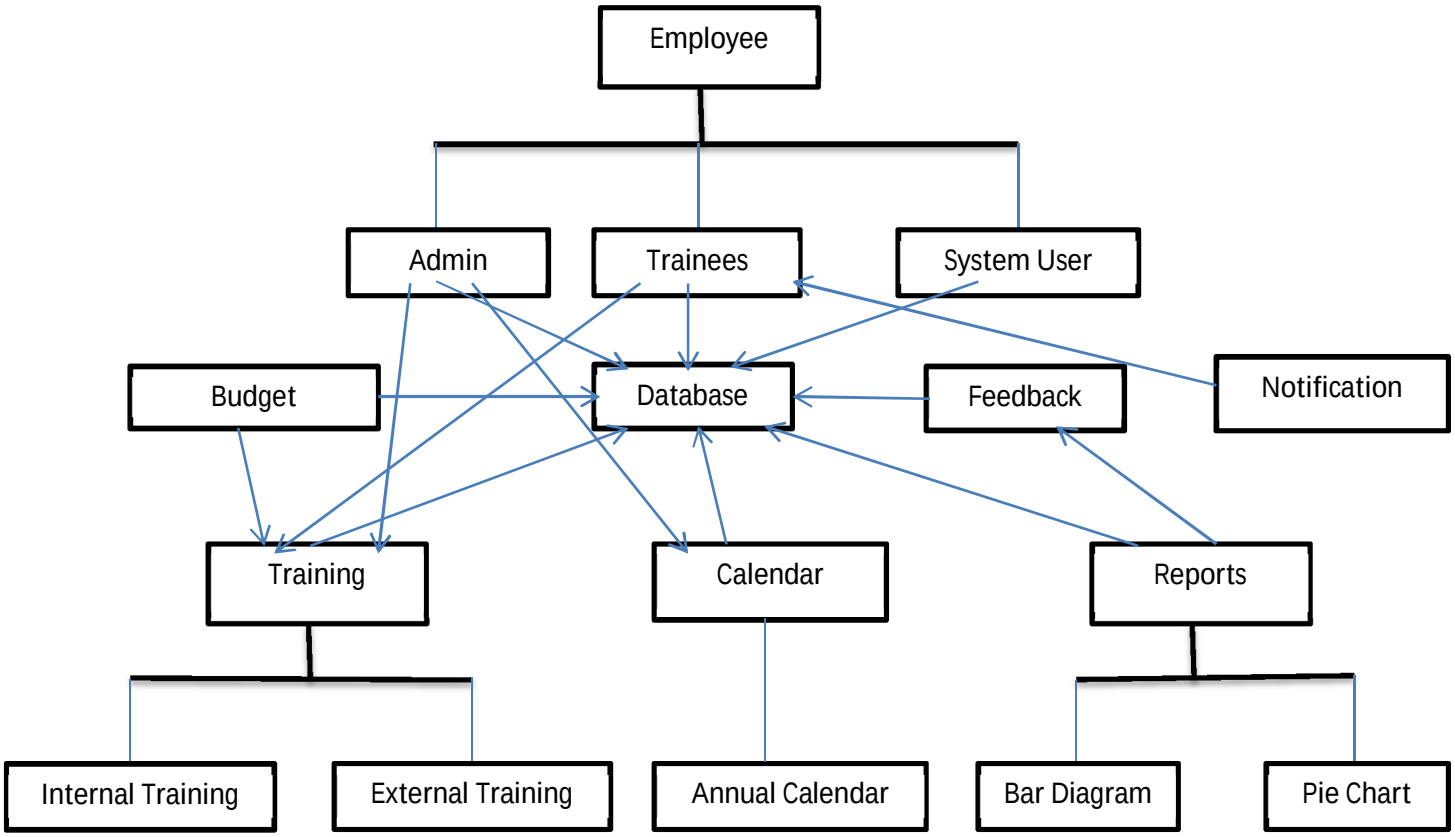
Database

Attributes	Methods
1. User name 2. Password 3. Employee Id 4. Division Name 5. Employee Info	1. Add() 2. Update() 3. Deactivate()

Admin, User creation, Selected Employees, Employees Information, Authentication.

Pie Chart	
Attributes	Methods
1) X Axis 2) Y Axis 3) Radios 4) Color	1. DrawPieChart()
	Reports, Feedback Form

3.3.6.3 Class-Responsibility-Collaborator (CRC)



3.3.7 Dataflow Diagram

Level-0 DFD:

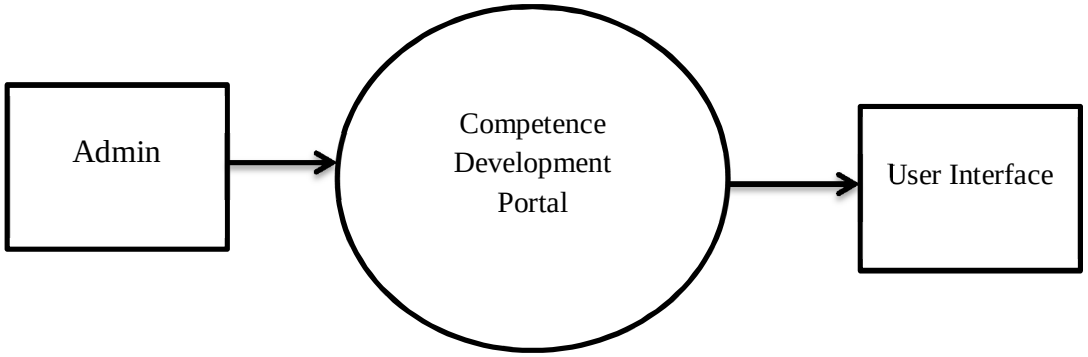


Fig: Level-0 DFD

Level-1 For Authentication:

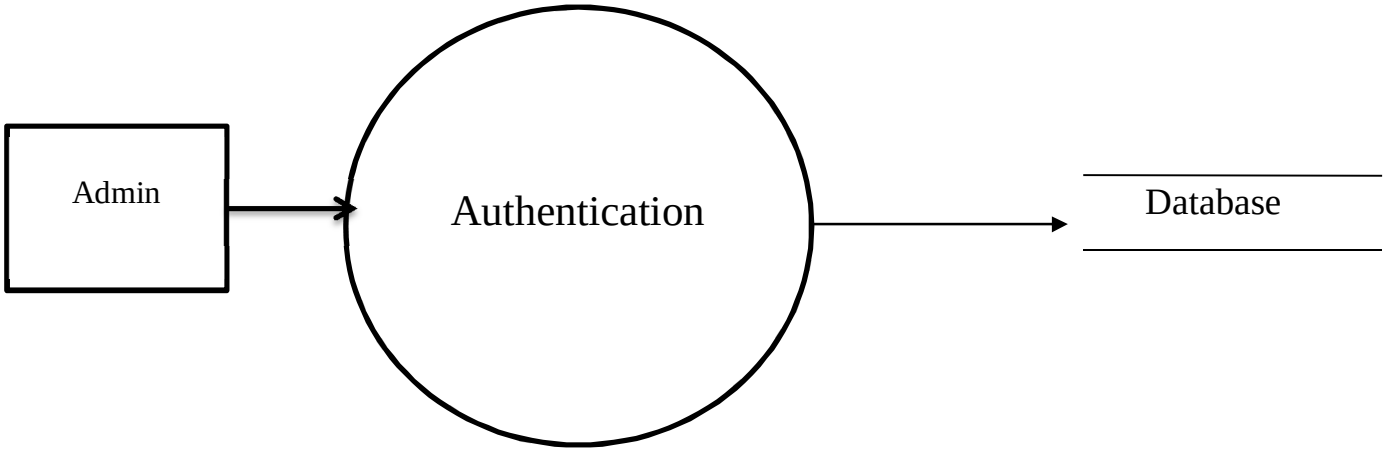


Fig: Level-1 DFD

Level-02:

Authentication:

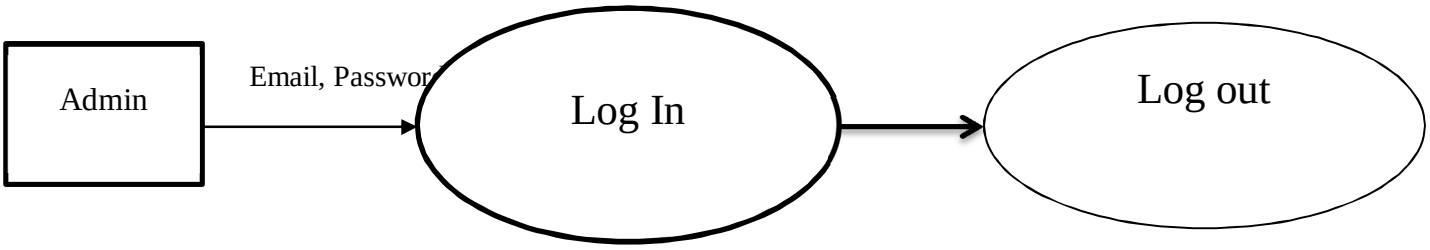


Fig: Authentication (Level-2)

User Creation:

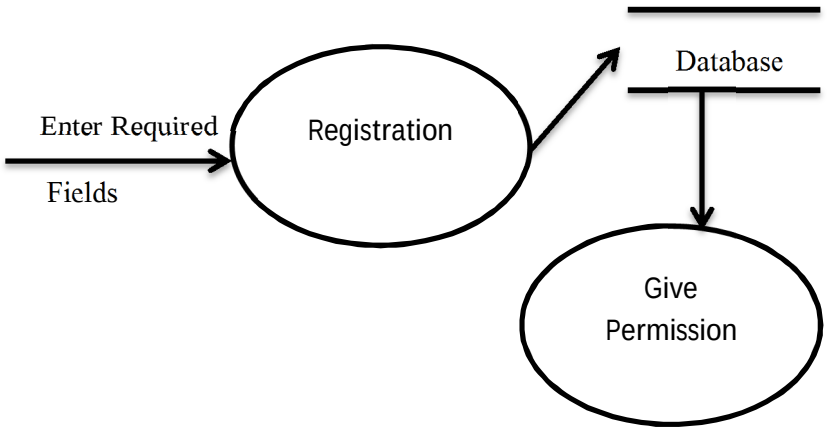


Fig: User Creation (Level-1)

Feedback Form:

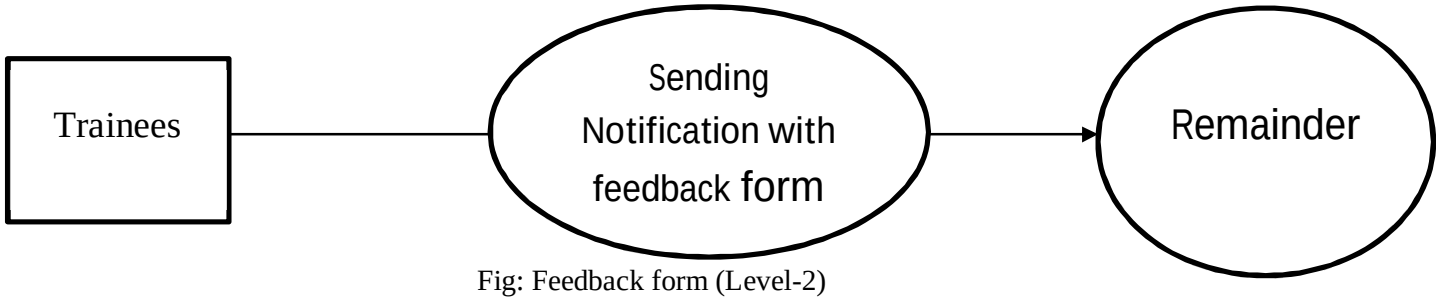


Fig: Feedback form (Level-2)

Local External Training:

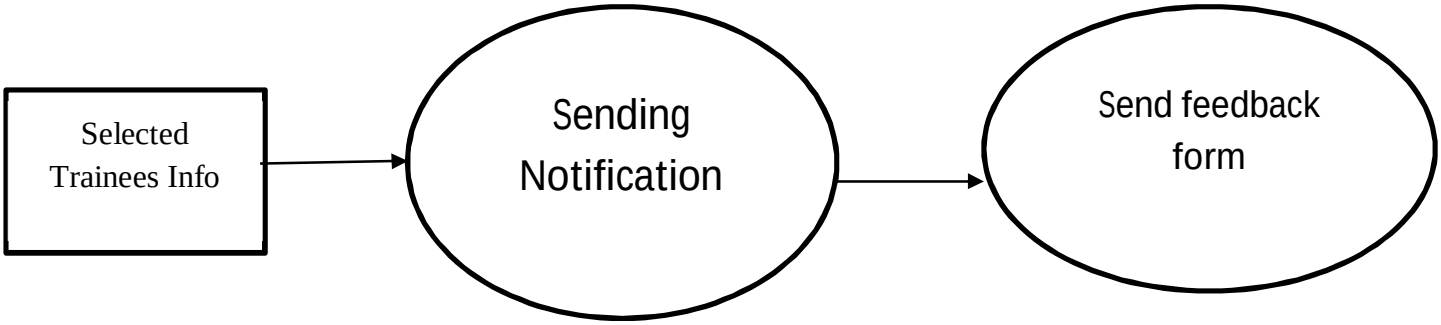


Fig: Local External Training (Level-2)

Internal Training:

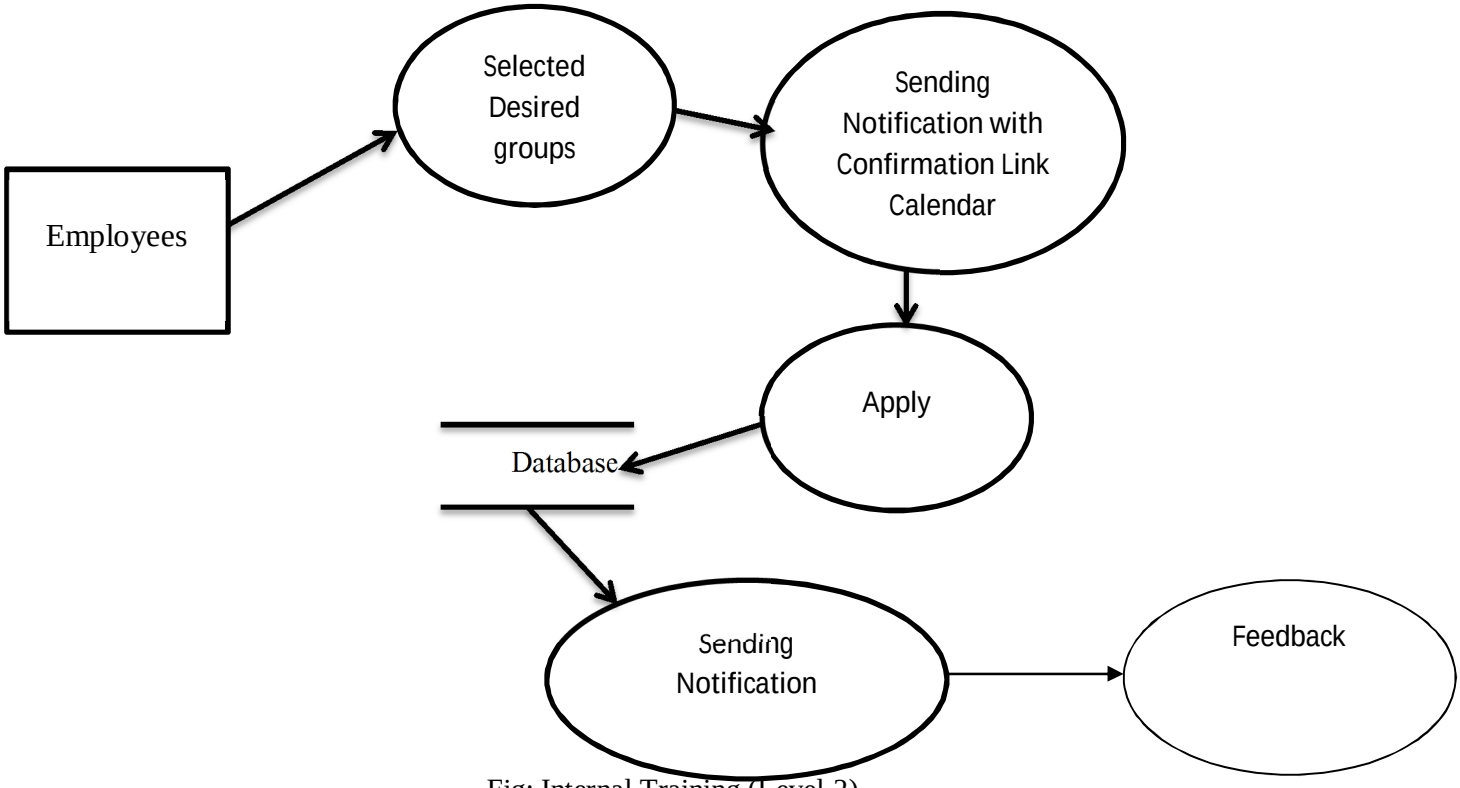


Fig: Internal Training (Level-2)

3.3.8 State Diagram

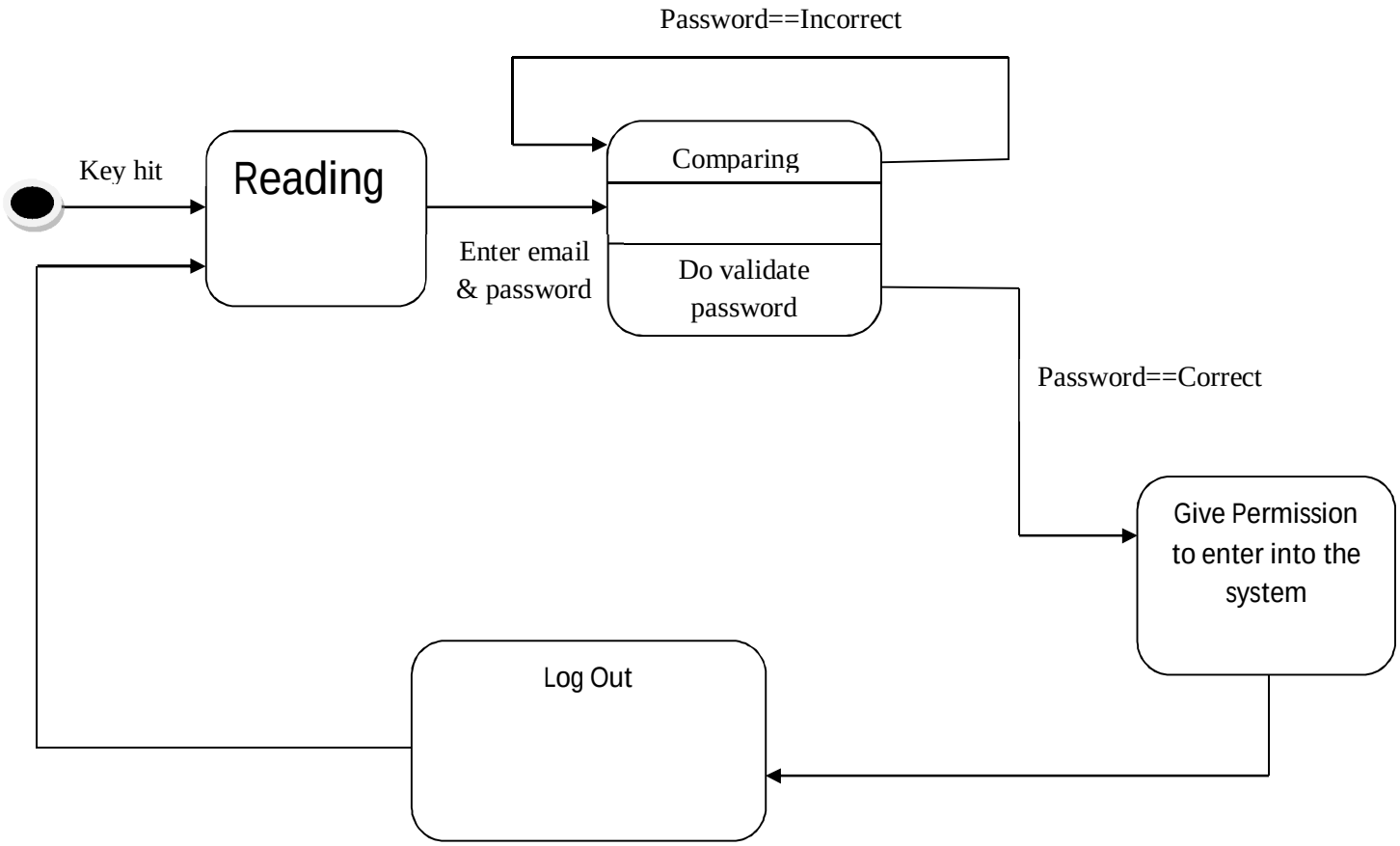


Fig: Authentication System

Creating User:

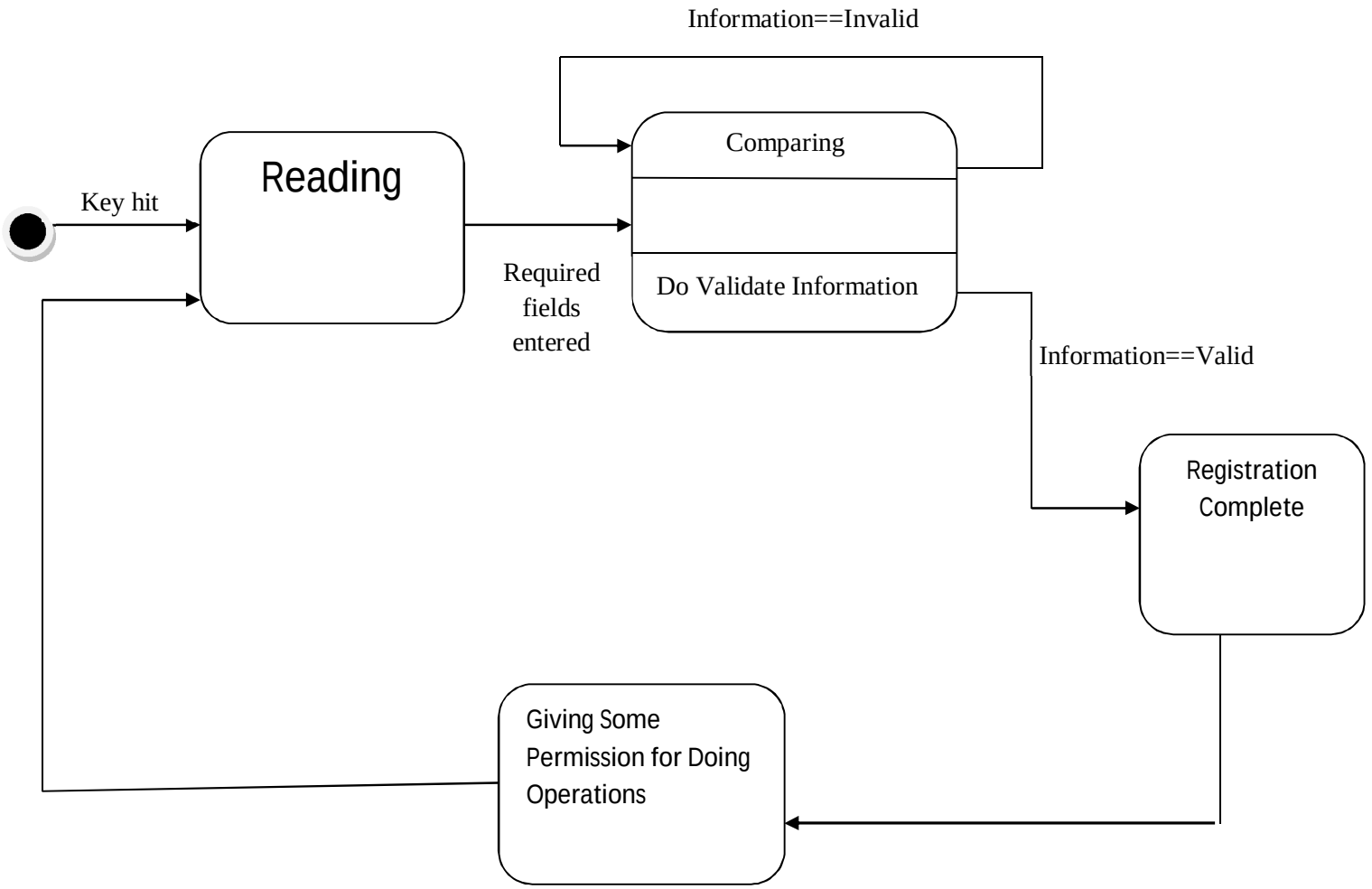


Fig: Creation of User

Conducting Training:

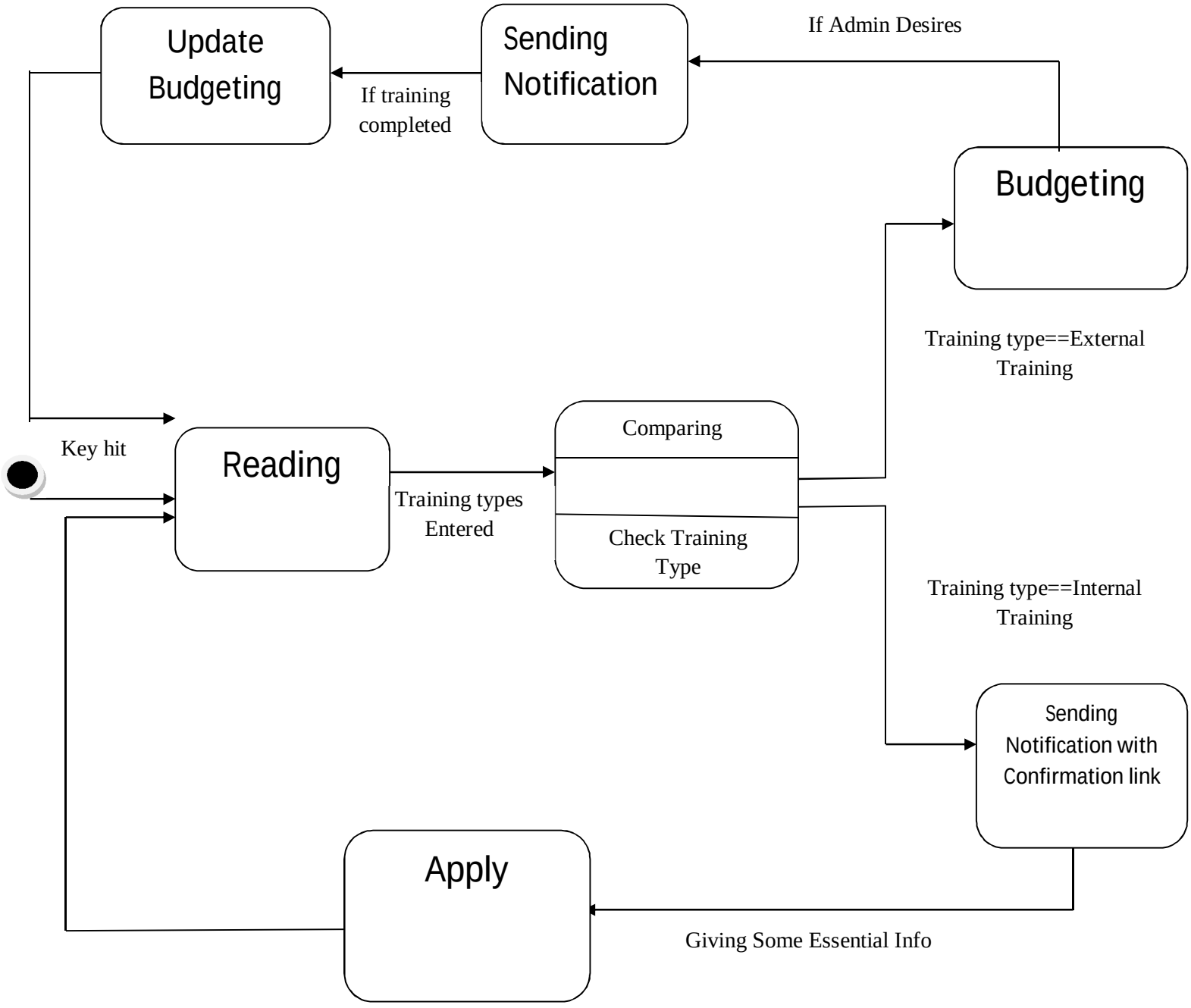


Fig: Conducting Training

Creating calendar:

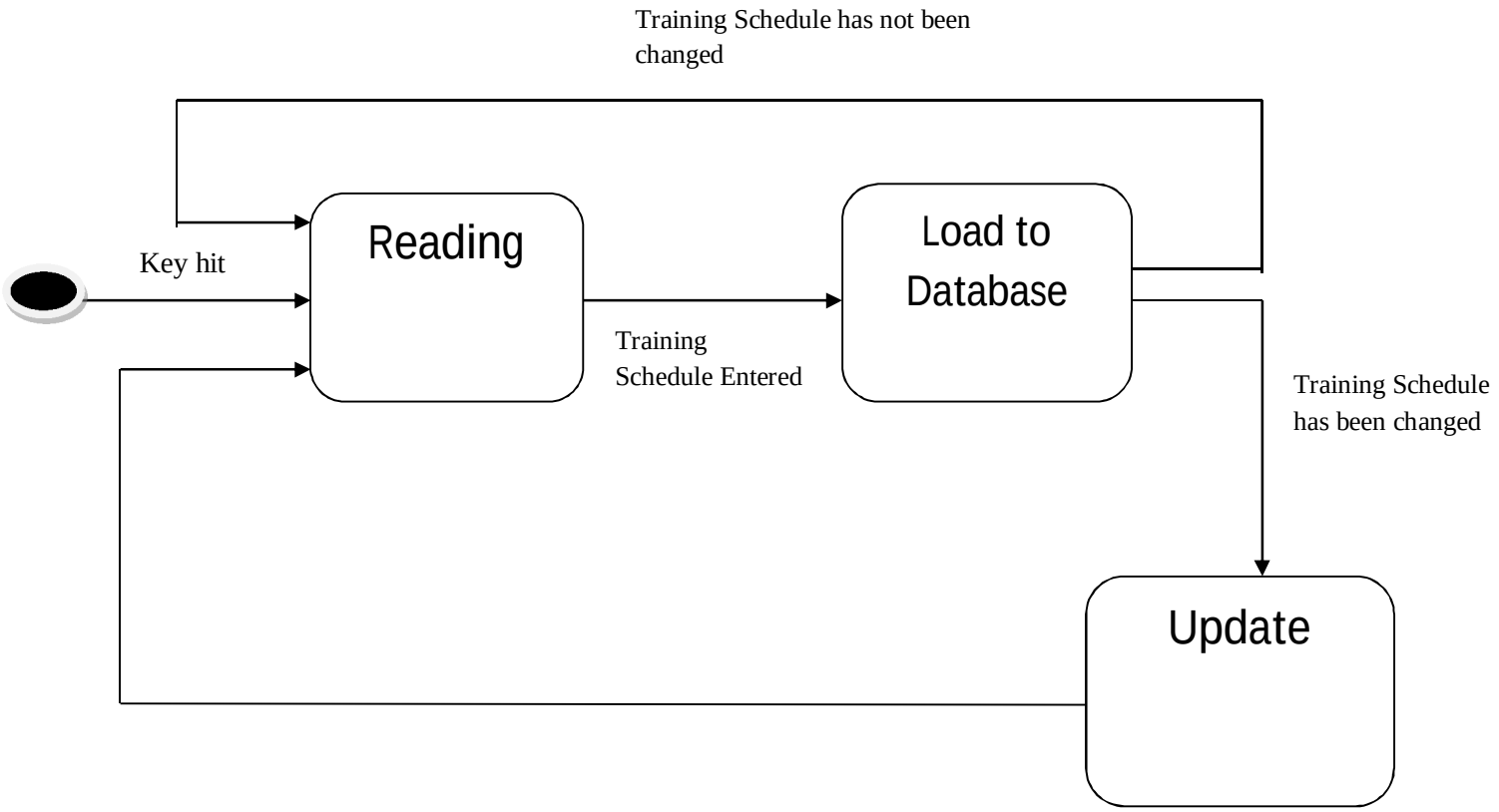


Fig: Creating Training Calendar

Report Generation with Feedback:

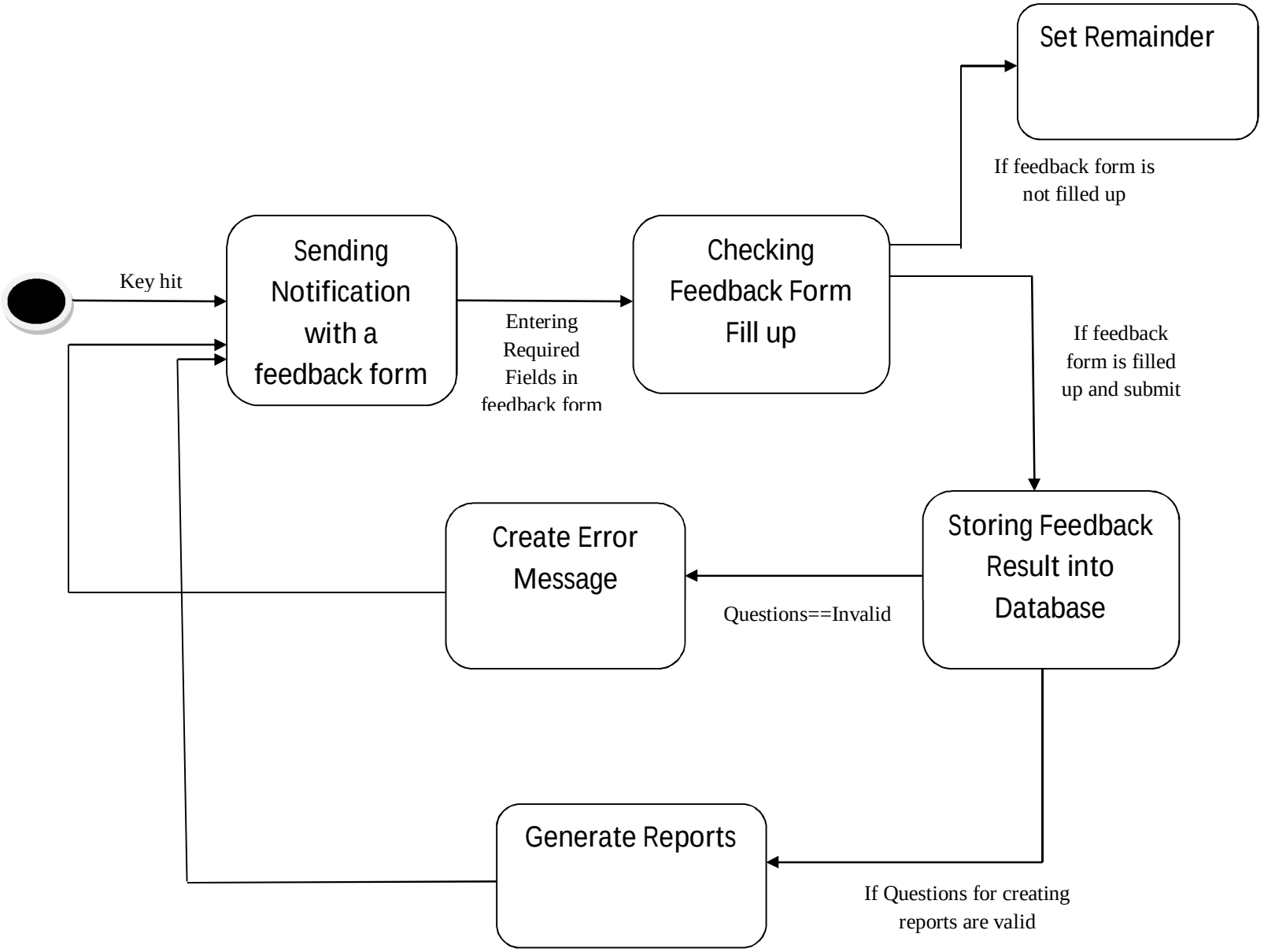


Fig: Creating Training Calendar

3.3.8 Sequence Diagram

Authentication System:

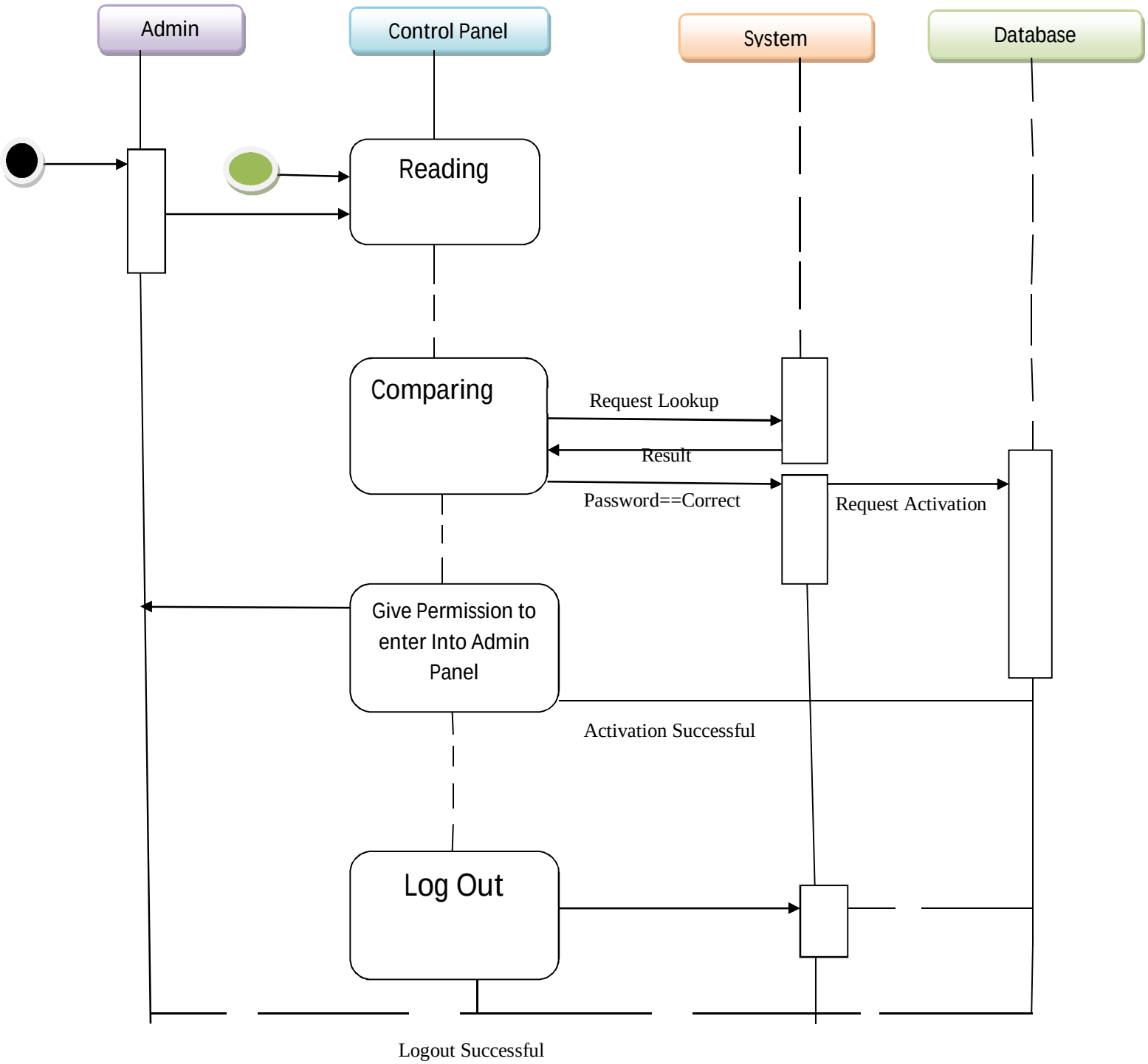


Fig: Sequence Diagram for Authentication

User Creation:

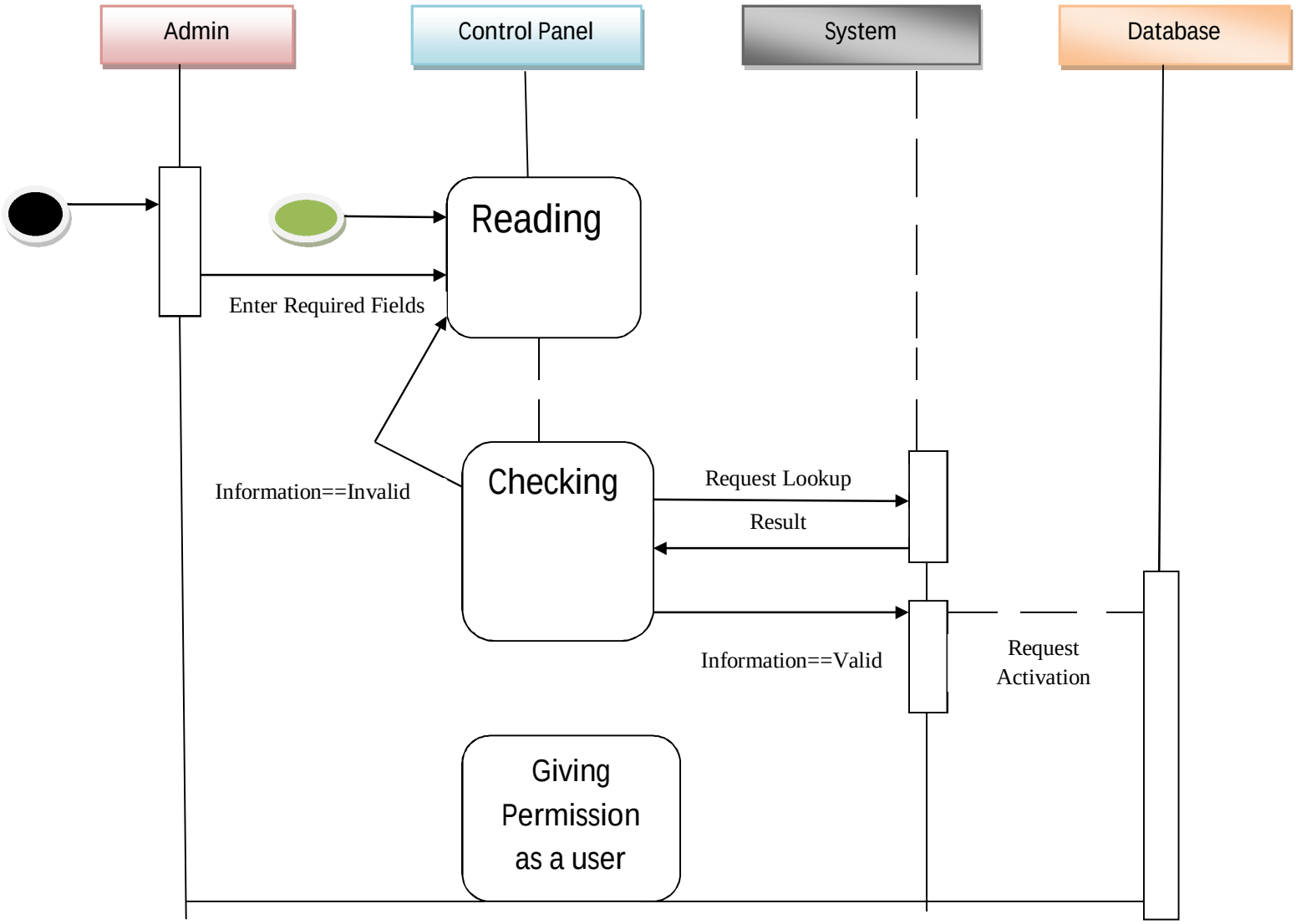


Fig: Creation of User

Conducting Training:

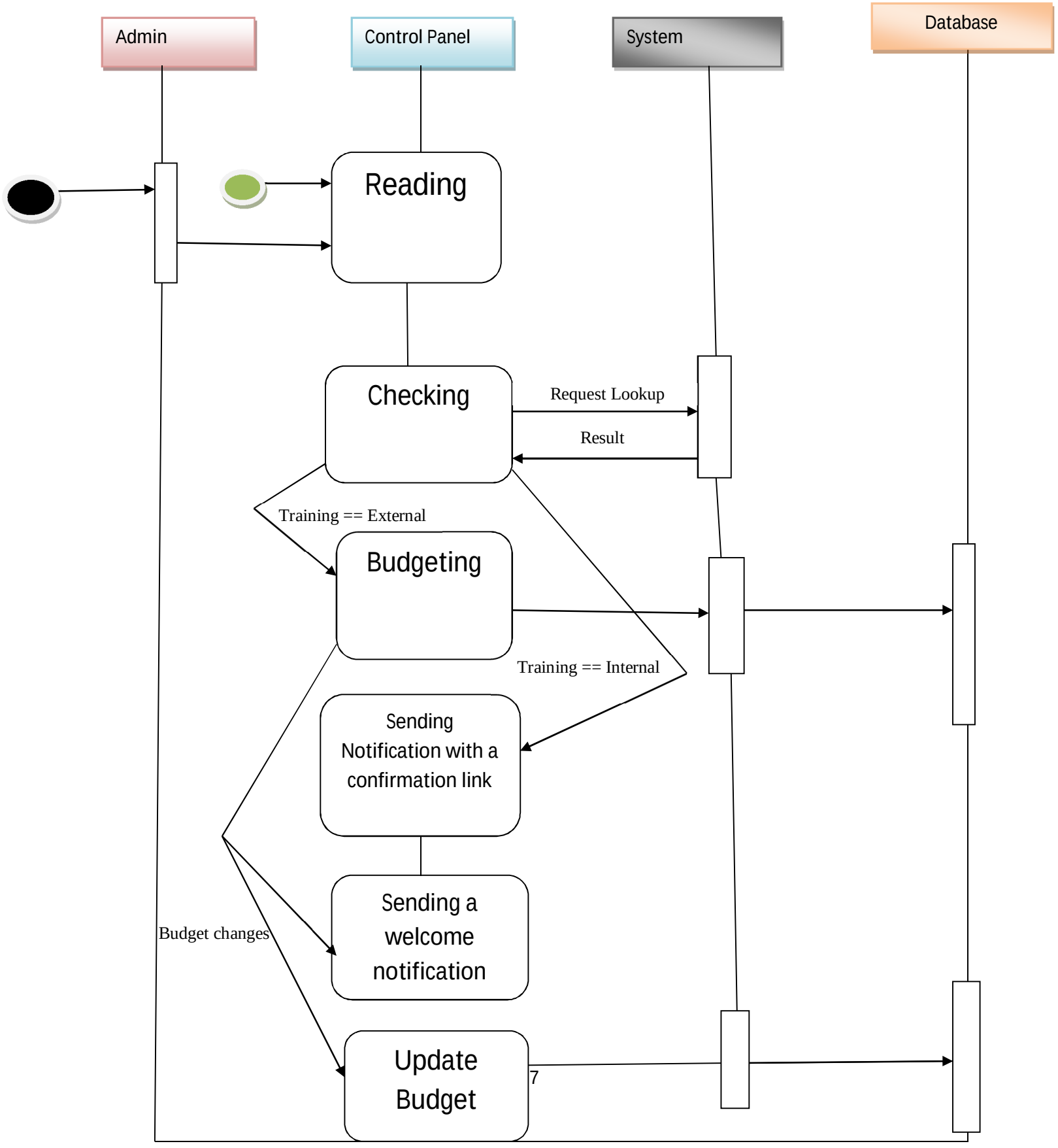


Fig: Conducting training

Calendar Creation:

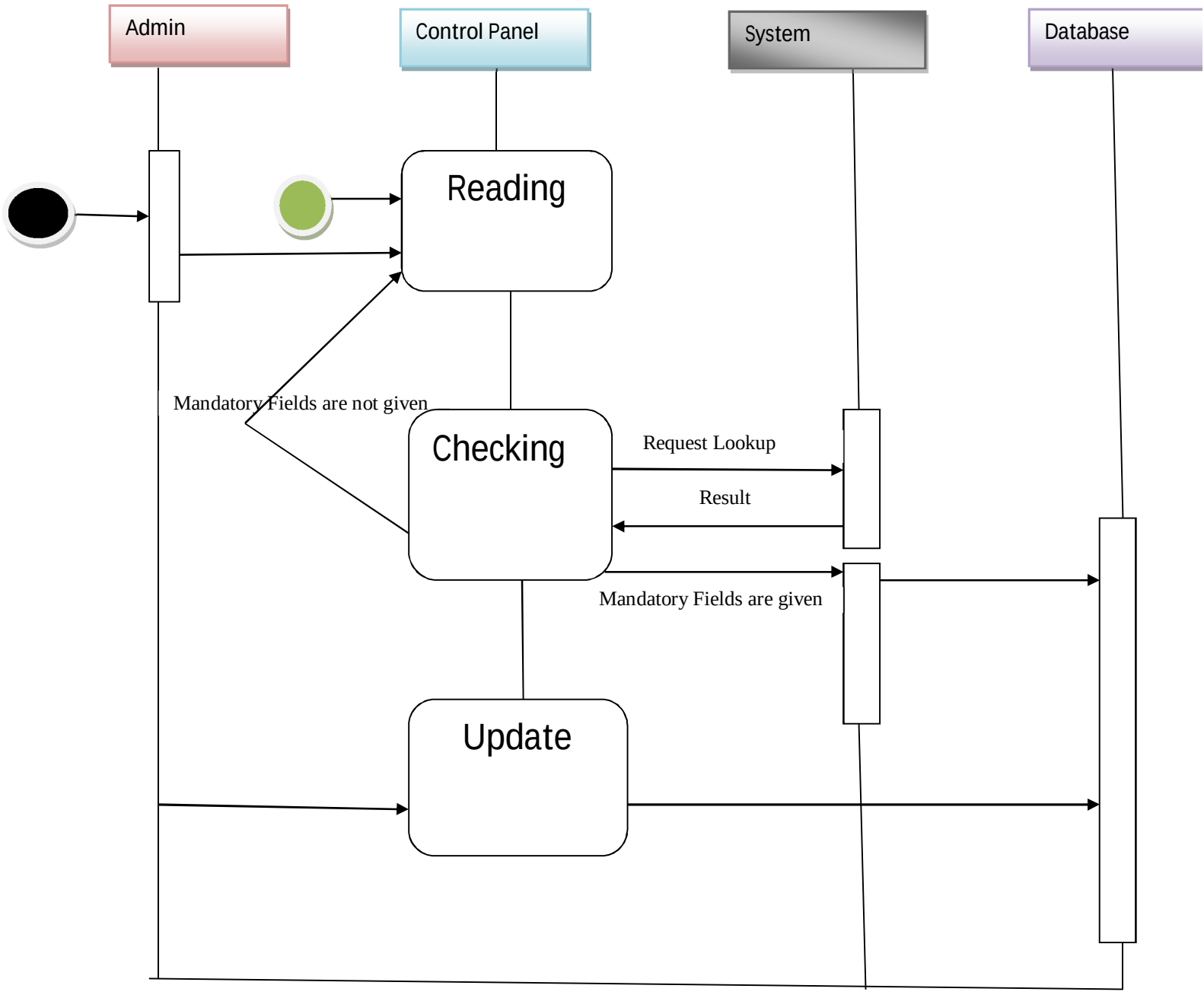


Fig: Training Calendar Creation

Reports with Feedback:

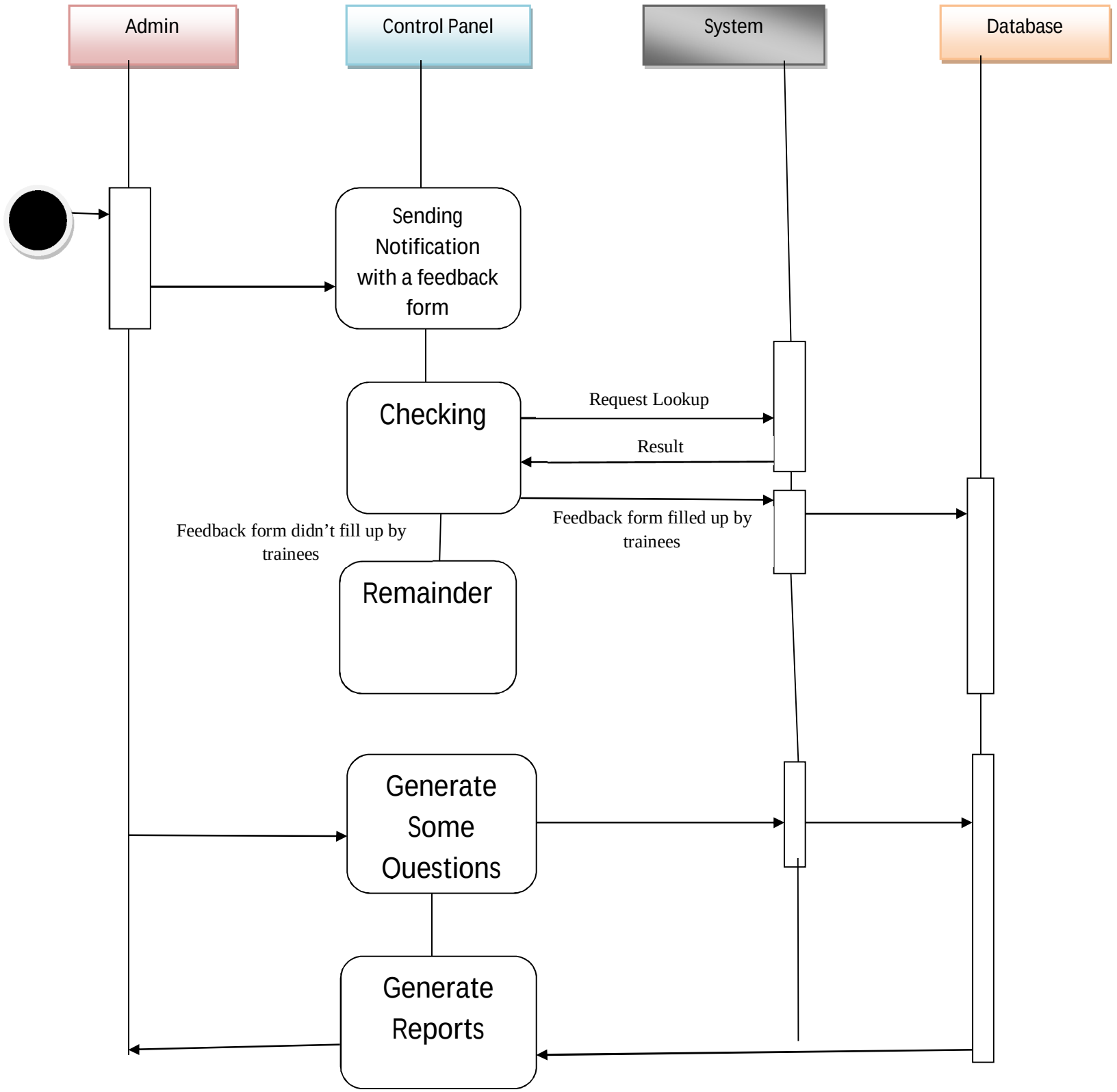


Fig: Sequence Diagram for Reports with feedback form

3.4 Conclusion

A software requirements specification (SRS) is a comprehensive description of the intended purpose and environment for software under development. The SRS fully describes what the software will do and how it will be expected to perform. Software Requirement Specification is an important part of the process of project development. Moreover, it is a prerequisite for creating the following design documentation. In this document, we have provided the information about general product description, data elements that the product deals with, specific requirements like product's interfaces and the functions will be implemented. In addition general behavior of the product has been explained in order to make it easy for the customer to understand to product's usage clearly. Furthermore, structure and working plan of the developing team is given. This document has been created through the help of various researches and depending on the demands of the customer. However, some little specifications are prone to be changed in the future.

Chapter Four

GPIT Competence Development Tool System Design

Design is what every engineer wants to do. It is the place where creativity rules—where stakeholder requirements, business needs, and technical considerations all come together in the formulation of a product or system. Design of Competence Development Tool of GPIT creates a representation or model of the software, but unlike the requirements model (that focuses on describing required data, function, and behavior), the design model of Competence Development Tool provides detail about software architecture, data structures, interfaces, and components that are necessary to implement the system. Different Design Phases of GPIT project are given below.

4.1 Interface Design of GPIT Project

The interface design elements for software depict information flows into and out of the system and how it is communicated among the components defined as part of the architecture. It include the following interfaces –

- User Interfaces
- Communication Interfaces

4.1.1 User Interface Design of GPIT Project

The user interface design of GPIT Project creates an effective communication medium between a human and a computer. Following a set of interface design principles, design identifies interface objects and actions and then creates a screen layout that forms the basis for a user interface prototype.

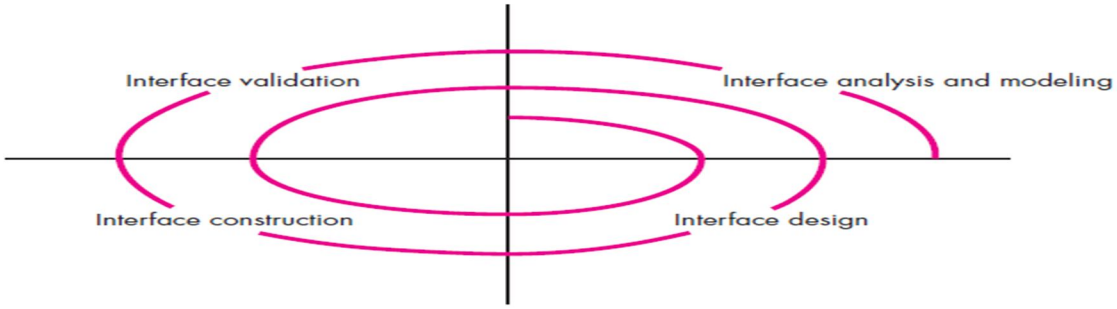


Figure 3.1: User Interface Design Process (Reference: Software Engineering Practitioner Approach, R. Pressmen)

Preliminary User Interface Design of GPIT Competence Development Tool

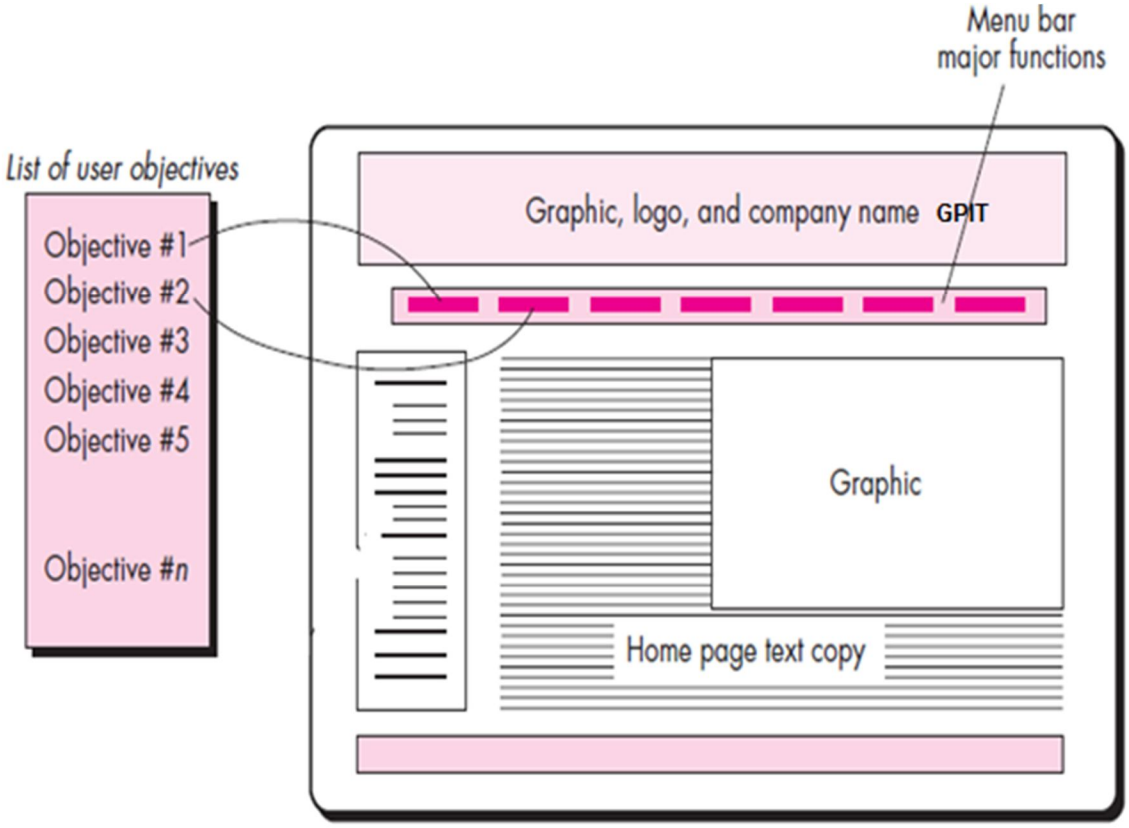


Figure 3.2: Mapping user objectives into interface actions

4.1.2 Communications Interface Design of GPIT project

HTTP: Hypertext Transfer Protocol is a transaction oriented client/server protocol between web browser & a Web Server.

TCP/IP: Transmission Control Protocol/Internet Protocol, the suite of communication protocols used to connect hosts on the Internet. TCP/IP uses several protocols, the two main ones being TCP and IP.

4.1.2.1 Network Diagram of Competence Development Tool

Network Diagram represents the communication process, that is, how GPIT Competence Development Tool and server communicate through Network.

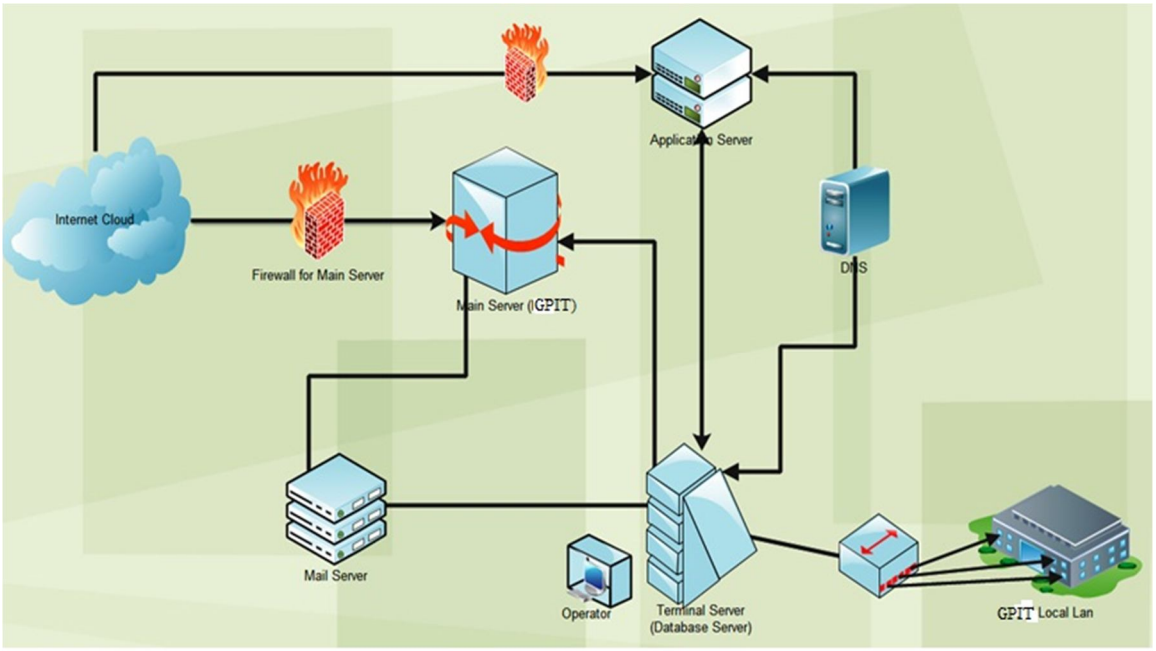


Figure 3.3: Network Diagram of our proposed system

4.2 System Requirements

4.2.1 Minimum Hardware Requirements

Processor	:	Pentium IV
Ram	:	256 mb
Hdd	:	2 GB
Key board	:	normal
Mouse	:	normal

4.2.2 Software Requirements

Operating System	:	Windows 7
Front End	:	C#, ASP.Net 2008(Entity frame work 4.0)
Back End	:	SQL server 2008, SMTP server
Web server	:	IIS

4.3 Architectural Design of Competence Development Tool

Architectural design of GPIT Competence Development Tool represents the structure of data and program components that are required to build the system. It considers the architectural style that the system will take, the structure and properties of the components that constitute the system, and the interrelationships that occur among all architectural components of a system.

4.3.1 System Architecture of Competence Development Tool

As architectural design begins, the software to be developed must be put into context— that is, the design should define the external entities (other systems, devices, and people) that the software interacts with and the nature of the interaction. This information can generally be acquired from the requirements model and all other information gathered during requirements engineering. The system architecture of GPIT Competence Development Tool represents how system works.

The figure 3.4 shows the Architectural Context Diagram (ACD) of GPIT Competence Development Tool.

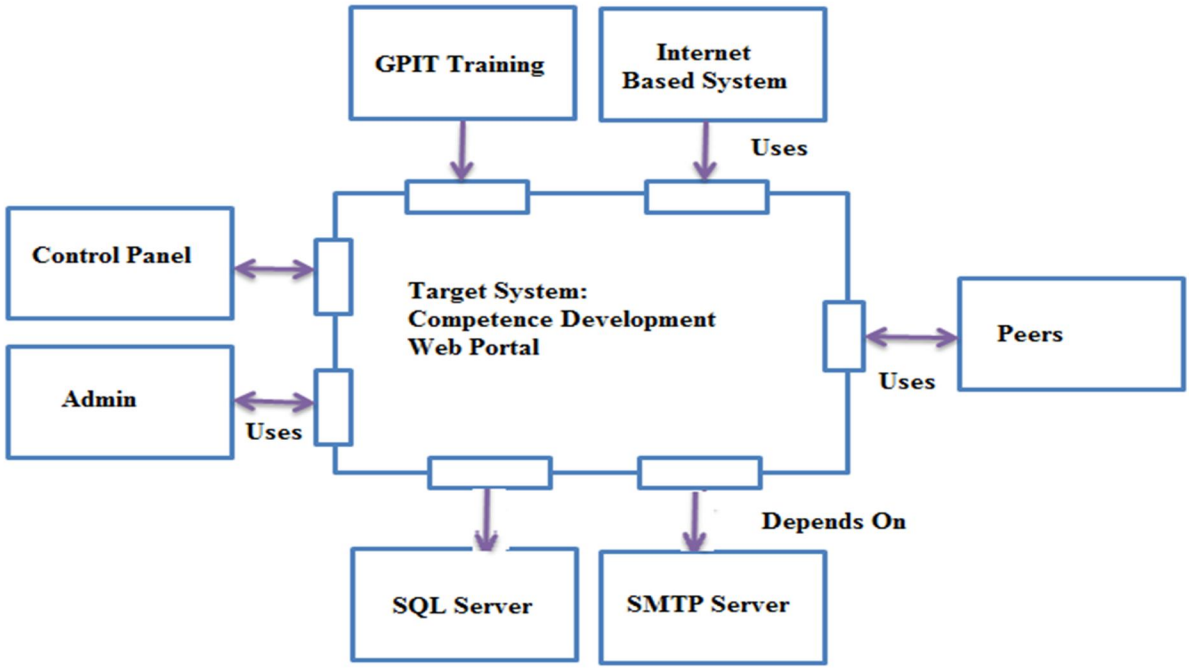


Figure 3.4: Architectural Context Diagram (ACD)

The figure 3.5 illustrates the System Architecture of our proposed system.

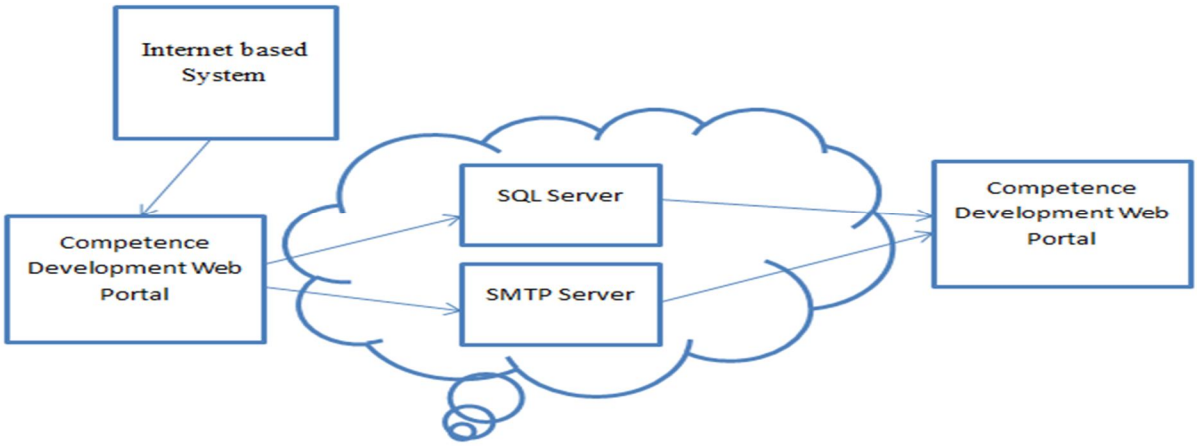


Figure 3.5: System Architecture

4.3.2 Software Architecture of GPIT Competence Development Tool

The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among those. Software architecture must model the structure of a system and the manner in which data and procedural components collaborate with one another. Software architecture of GPIT Competence Development Tool, such as Data Centered Architecture and Data Flow architecture are given below.

4.3.2.1 Data Centered Architecture

A data store (Such as, a file or database) resides at the center of this architecture and is accessed frequently by other components (Such as, GPIT User Authentication, GPIT Training Information, GPIT Employee Information, GPIT Feedback Information) those update, add, delete, or otherwise modify data within the store. Figure 3.6 illustrates a typical data-centered architecture of Competence Development Tool. Client software accesses a central repository to update, add, delete or otherwise modify data within the store. In some cases the data repository is passive, that is, client software accesses the data independent of any changes to the data or the actions of other client software.

Data-centered architectures promote *integrability*, that is, existing components can be changed and new client components added to the architecture without concern about other clients (because the client components operate independently) and client components independently execute processes.

Figure 3.6 illustrates a Data Centered Architecture of GPIT Competence Development Tool.

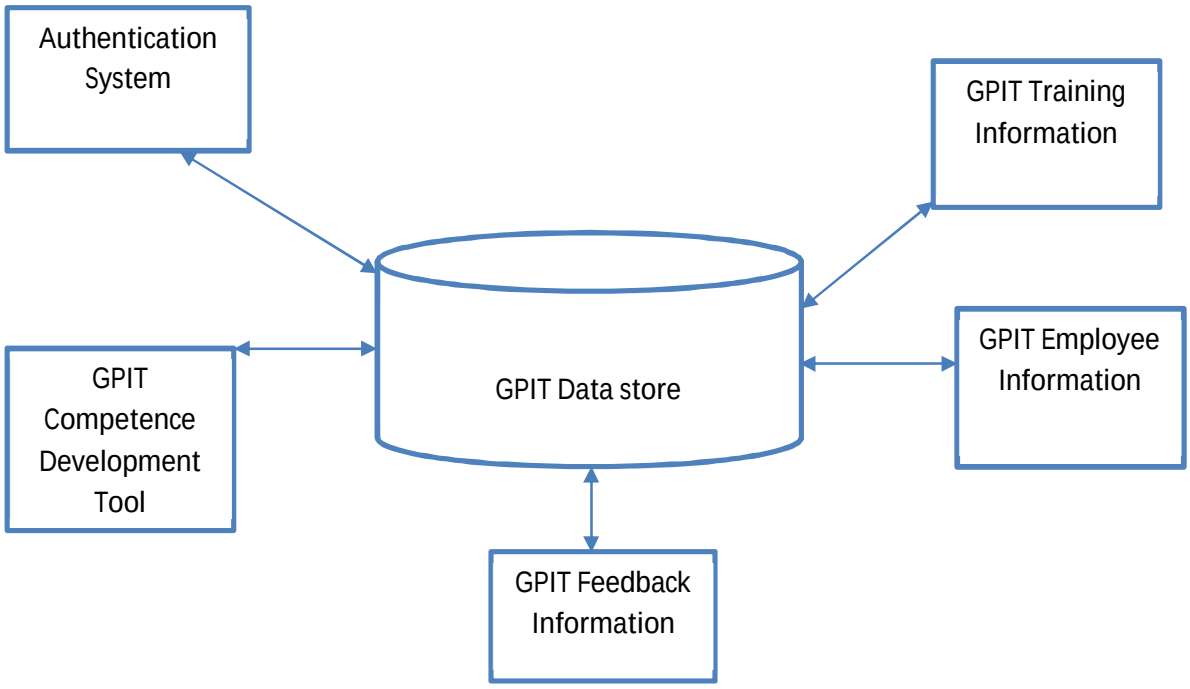


Figure 3.6: Data-centered Architecture

4.3.2.2 Data Flow Architecture

The data flow architecture represents the flow of data module to module. In chapter 2 we show different data flow diagram of our system. Different Modules of our System-

- Authentication System
- Creating Calendar with Schedule
- Budgeting
- Conducting Training
- Creating User
- Load Employee Information into DB
- Sending Notification with a Feedback Form
- Creating Reports

Figure 3.7 illustrates a Data Flow Architecture of GPIT Competence Development Tool

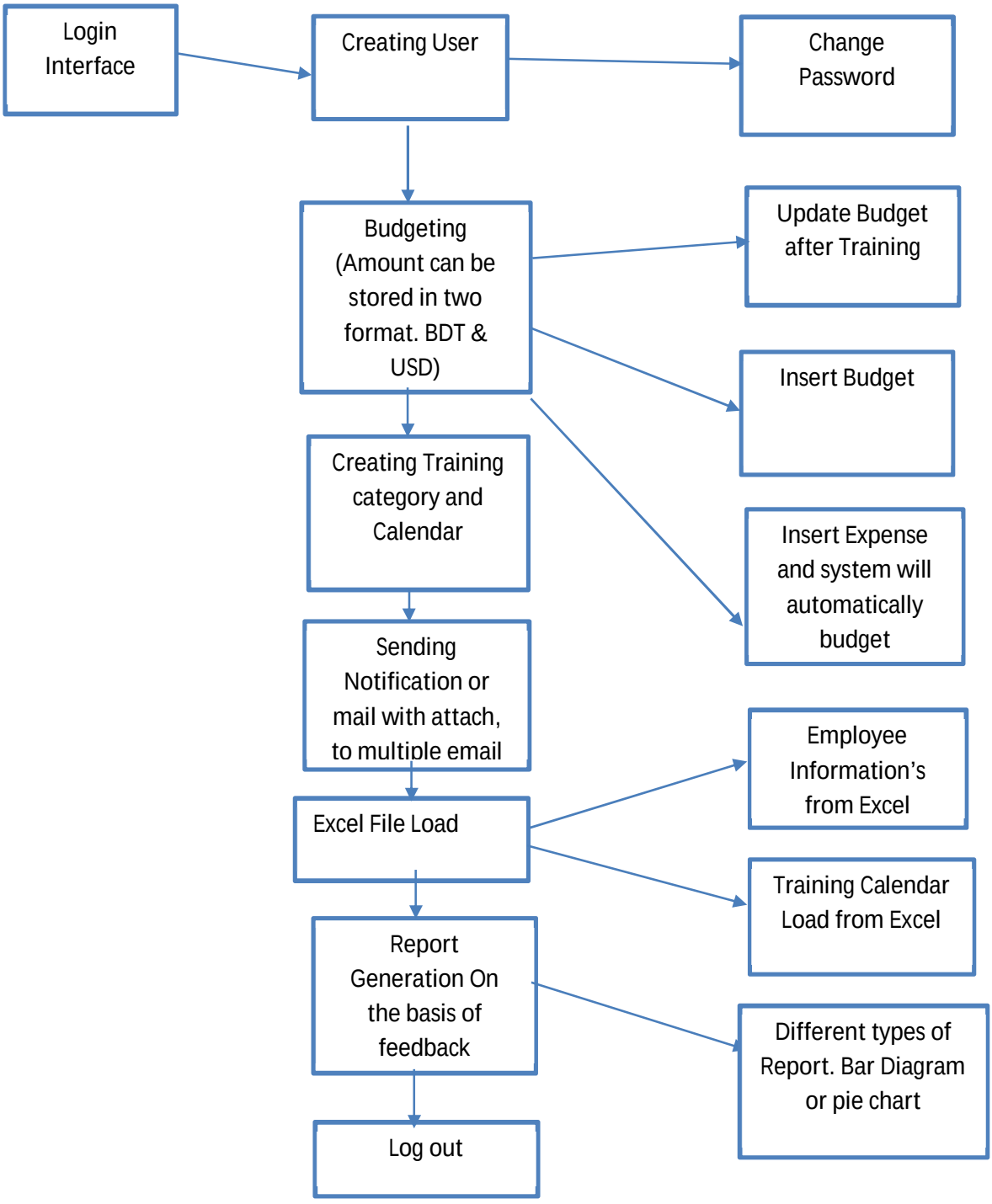


Figure 3.7: Data Flow Architecture

4.4 Conclusion

Design is the most valuable part to develop a software system. As GPIT Competence Development Tool is a well-engineered system, we design this system, although we have not enough knowledge for designing a system. We illustrate Interface Design, requirements design and architectural design to develop GPIT Competence Development Tool. The system may not be well designed, but we try our best to make e good design based on our knowledge and with the help of our respective teacher Dr. Kazi Muheymin-Us-Sakib, Associate Professor, Program Chair (BIT), Institute of Information Technology, Dhaka University.

Chapter Five

GPIT Competence Development Tool Implementation and Testing

This chapter aims to describe the implementation process of Competence Development Tool. Here the technologies that have been used to develop this system and the testing that have been done during this system development will be described in brief.

5.1 Implementation

Implementation is the final and important phase. Implementation is the stage in the project where the theoretical design is turned into a working system. The system can be successfully implemented only after thorough testing.

5.1.1 The Technology

Development technologies are growing very rapidly with the increase of requirements. The technologies that have been used to develop this system is the most recent technologies and also very much appropriate to it.

5.1.1.1 Web User Interface

User interface is the working environment for the user. We used various language and frameworks. Here they are:

ASP .NET Framework:

The ASP.NET Framework is a web application development framework. It can build scalable, standards-based web applications using well-established design patterns.

Cascading Style Sheet (CSS):

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation semantics (the look and formatting) of a document written

in a markup language. There are several versions of CSS in web. We have used CSS3.0 version which is latest version of it.

Hyper Text Markup Language (HTML):

Hyper Text Markup Language (HTML) is the main markup language for web pages. HTML elements are the basic building-blocks of a webpage. We have used the latest HTML5 for developing this system.

Internet Information Service (IIS):

IIS (Internet Information Server) is a group of Internet servers (including a Web or Hypertext Transfer Protocol server and a File Transfer Protocol server) with additional capabilities for Microsoft's Windows NT and Windows 2000 Server operating systems**.** With IIS, Microsoft includes a set of programs for building and administering Web sites, a search engine, and support for writing Web-based applications that access databases. We have used IIS 7.5 express which comes by default with Visual Studio 2010 Professional edition.

5.1.2 Implementation Tools

As types of software are increasing day by day, new implementation tools are also needed for their implementation. Now-a-days, there are many implementation tools. Developers have to choose right tools for each part of their application. If they can utilize tools perfectly, their labor can be reduced.

5.1.2.1 Microsoft Visual Studio 2010

Microsoft Visual Studio is a suite of component-based development tools and other technologies for building powerful, high-performance applications. In addition, Visual Studio is optimized for team-based design, development, and deployment of enterprise solutions. We have chosen Microsoft visual studio because:

- It is lightweight with respect to its ability.
- Database development is very easy and efficient. It supports entity framework.
- Its free and open source.
- It supports various plugins. We can add whatever plugins is needed for our application.
- Its suggestions and auto complete features are excellent.
- Have a user friendly interface.
- Visual Studio includes a code editor supporting IntelliSense as well as code refactoring.
- Visual Studio supports different programming languages by means of language services, which allow the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C/C++ (via Visual C++), VB.NET (via Visual Basic .NET), C# (via Visual C#), and F# (as of Visual Studio 2010). Support for other languages such as M, Python, and Ruby among others is available via language services

installed separately. It also supports XML/XSLT, HTML/XHTML, JavaScript and CSS. Individual language-specific versions of Visual Studio also exist which provide more limited language services to the user: Microsoft Visual Basic, Visual J#, Visual C#, and Visual C++.

5.1.2.2 MSSQL Server 2008

MSSQL Server is a relational database server, developed by Microsoft: it is a software product whose primary function is to store and retrieve data as requested by other software applications, be it those on the same computer or those running on another computer across a network (including the Internet). We have used MSSQL Server because of these:

- It has self-service business intelligence tools.
- Creditable and consistent data.
- Productive development experience.

5.1.2.3 .NET Runtime

The **.**NET Framework (pronounced *dot net*) is a software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a large library and provides language interoperability (each language can use code written in other languages) across several programming languages. Programs written for the .NET Framework execute in a software environment (as contrasted to hardware environment), known as the Common Language Runtime (CLR), an application virtual machine that provides important services such as security, memory management, and exception handling. The class library and the CLR together constitute the .NET Framework.

The .NET Framework's Base Class Library provides user interface, data access, database connectivity, cryptography, web application development, numeric algorithms, and network communications. Programmers produce software by combining their own source code with the .NET Framework and other libraries. We have used the 4.0 version.

5.2 Coding

After completion of designing the system, we start coding to achieve our goal. Using the above tools and technology we start our coding. The figures below show some snapshot of GPIT Competence Development Tools coding part.

Figure 5.1 shows the classes those are we create to develop the Competence Development Tool.

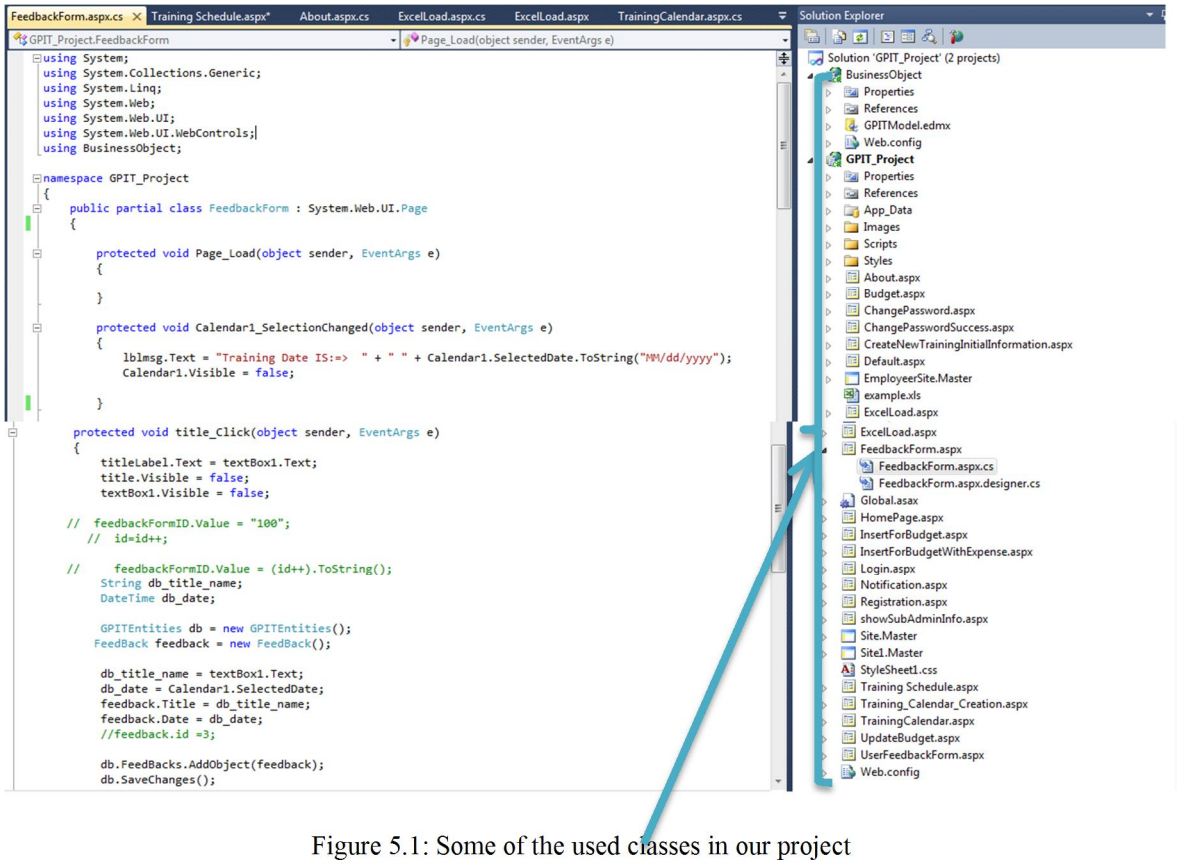


Figure 5.1: Some of the used classes in our project

Figure 5.2 shows the details of BudgetWithExpense class.

InsertForBudgetWithExpense.aspx.cs | InsertForBudgetWithExpense.aspx | InsertForBudget.aspx.cs | InsertForBudget.aspx | FeedbackForm.aspx.cs | Training Schedule.aspx* | About.aspx.cs

GPIT_Project.InsertForBudgetWithExpense — Page_Load(object sender, EventArgs e)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using BusinessObject;

namespace GPIT_Project
{
    public partial class InsertForBudgetWithExpense : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void submitButton_Click(object sender, EventArgs e)
        {

            {
                String divisionName;
                String year;
                String divisionCode;
                float budgetAmount;

                float expense,newBudget;

                if (IsValid)
                {

                    GPITEntities db = new GPITEntities();
                    divisionName = divisionDropDownList.SelectedValue;


                    year = yearDropDownList.SelectedValue;
                    divisionCode = divisionCodeTextBox.Text;

                    expense = (float)Convert.ToInt32( expenseAmountTextBox.Text);
                    Insert_For_Initial_Budget budget = db.Insert_For_Initial_Budget.SingleOrDefault(p => p.year_For_Budgeting == year && p.Division_Name == divis

                    budgetAmount = (float) budget.Budgiting_Aomunt;
                    newBudget = budgetAmount - expense;

                    //budget.Budget_Amount =(double)newBudget;
                    budget.Budgiting_Aomunt = newBudget;
                    db.SaveChanges();
                    expenseAmountTextBox.Text = "";
                    divisionCodeTextBox.Text = "";
                }
            }

        }
    }
}
```

Figure 5.2: BudgetWithExpense class

Figure 5.3 shows the Database Schema Diagram of competence Development Tool which we create in implementation phase.

Figure 5.3: Database Schema Diagram

5.3 Testing

Software testing is an investigation conducted to provide stakeholders information about the quality of the product or service under test. We wanted to test our application perfectly. But unfortunately, we do not have enough knowledge to test such application.

With our limited knowledge of testing, we do some informal testing during development period. As it was informal, we cannot submit any document. This means that this current version is released without any formal testing.

5.4 Conclusion

In this chapter, we discuss two vulnerable phase to develop a software system. Here, we discuss the Implementation and Testing phase. In Implementation phase, we describe what tools we use and how we implement those tools. Although we do not use any formal testing, we use unique test in our development period.

Chapter Six

Competence Development Tool User Manual and Snapshot

This section of this document aims to make the usage of this system easy to the users by providing the major functionalities and how they work from a user's perspective. This section will contain no technical word. It will be described in very simple language which is very well known to the common users of this system.

6.1 Getting Started

Authentication process in the web is very common. Most of the web application has their own authentication system for identifying their user. This process has some common steps like registration, logging in, logging out and changing account password. As our system is a web based application, it also needs this authentication processes and thus we have.

6.1.1 Login

To get access into the system, admin need to Log In. To Log In into the system user name or E-mail address and Password is required. The figure below shows the Log In interface.

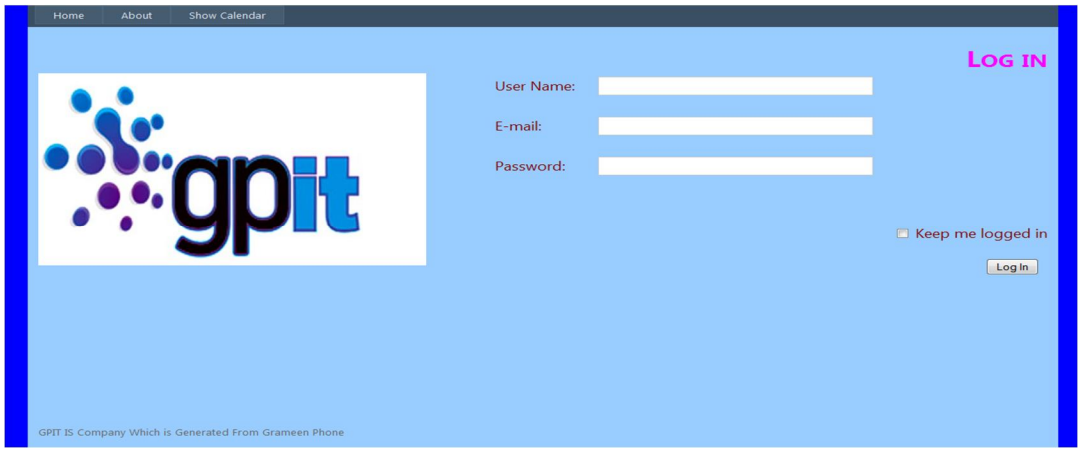


Figure: Log In Interface

Figure: Interface after Logging In

6.1.2 Change Password

Admin or Sub admin can change password when he / she wants. To change password he / she must enter current password and the new password.

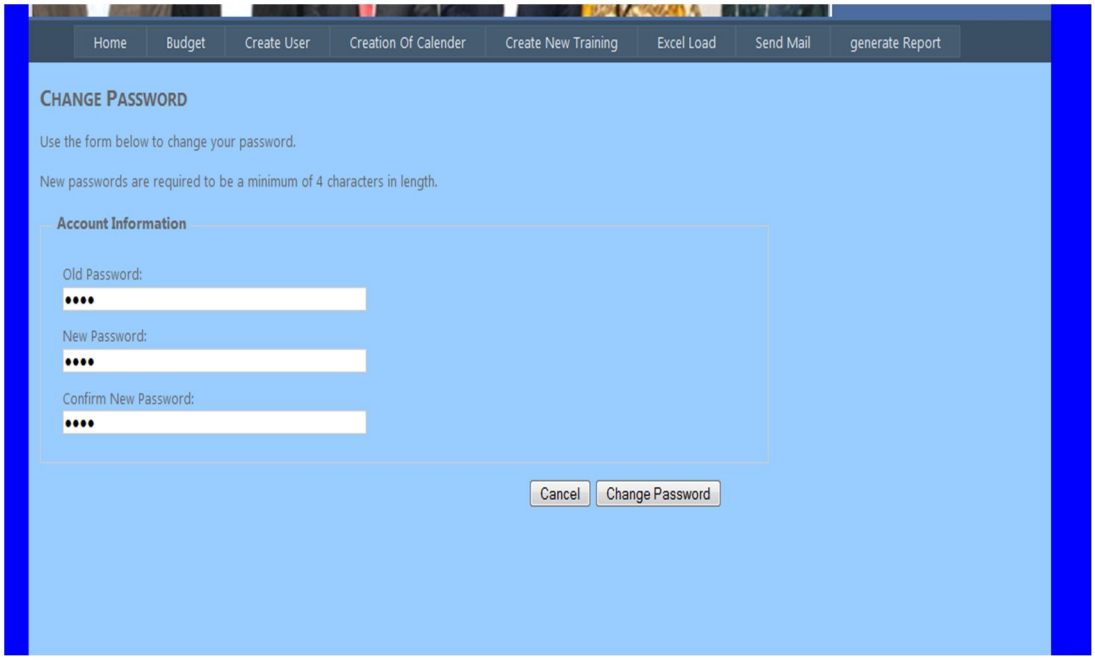


Figure: Change Password

6.1.3 Create Users

Creation of user stands for creation of sub admin. Inserting User name, E-mail, and Password, Mobile No, and Employee Id admin can create new user or sub admin.

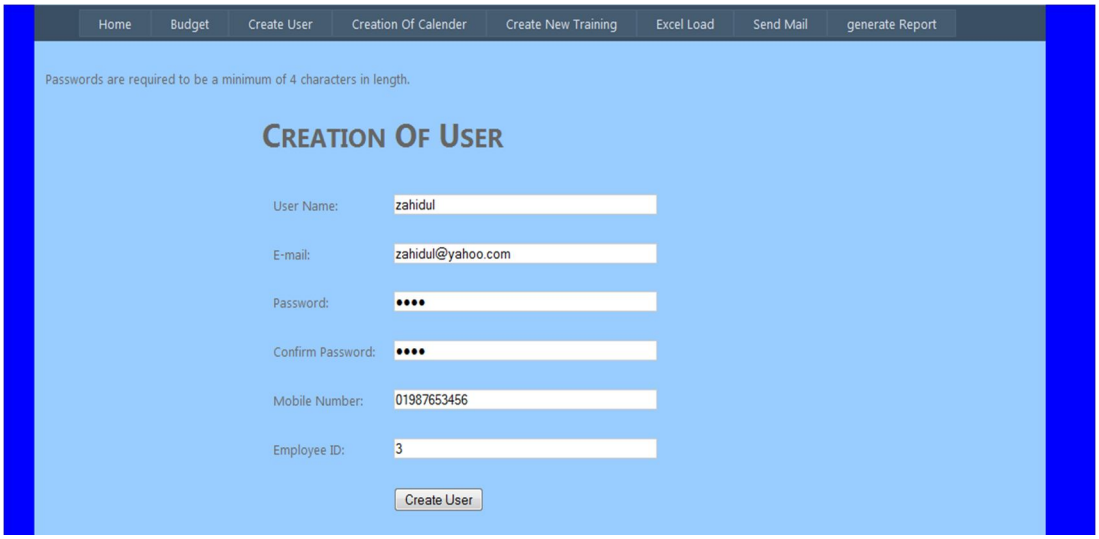


Figure: Creation of User

6.1.4 Logout

When a user logged in into the system, the system provides a logout module for logging out from the system. To log out user only need to click on “Log Out” button.

Figure: Logging out

6.1.5 Insert initial Budgets

On the beginning of the year admin will insert budget according to GPIT's different division's name. He is capable for inserting budget into two formats, one is BDT and another is USD. In every time admin will give how much BDT amount is equal to 1 USD, because amount can vary sometimes. Conversion of USD to BDT, Division Code, and Budgeting Amount is mandatory.

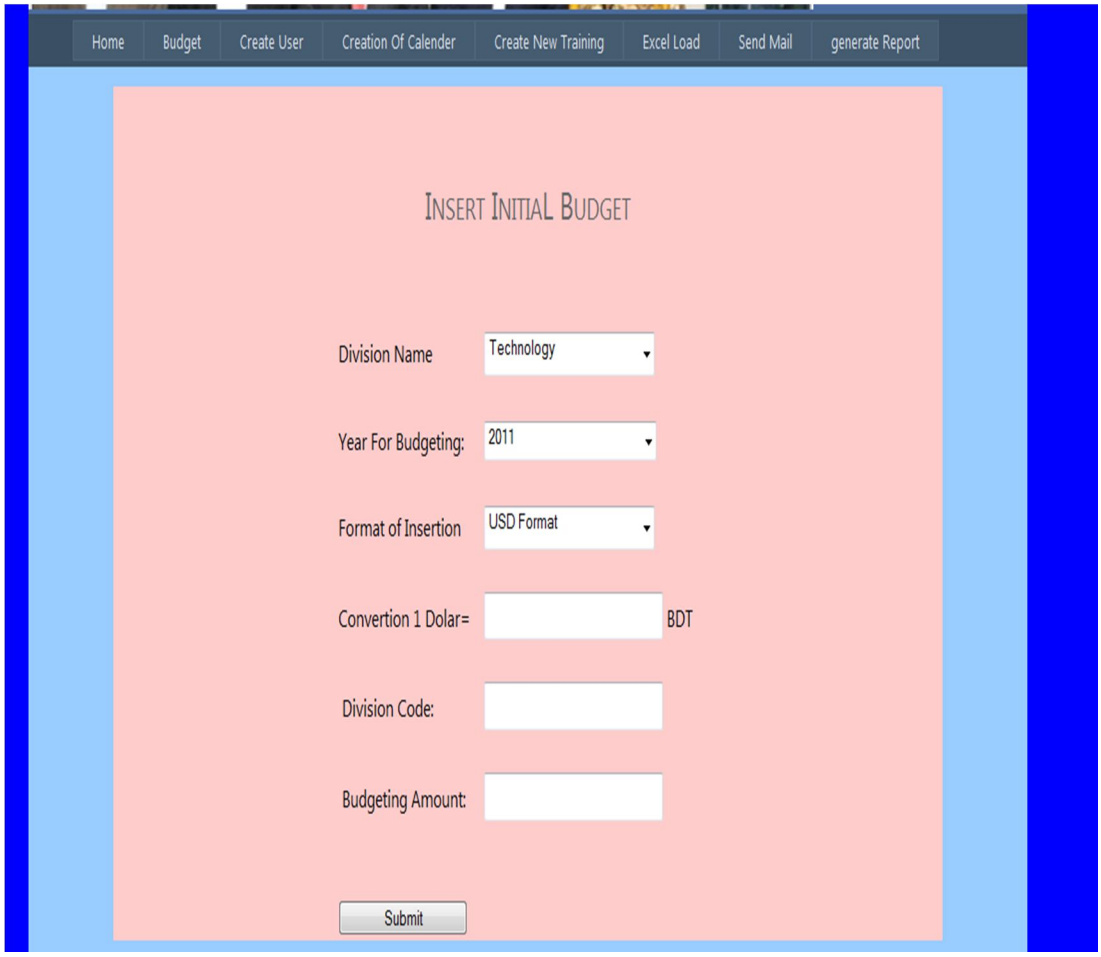


Figure: Inserting Initial Budget

6.1.6 Update budget

After inserting the initial budget amount, if admin thinks one division's budget has been changed; through update budget option admin can change the budget amount of a particular division of a particular year which is previously inserted.

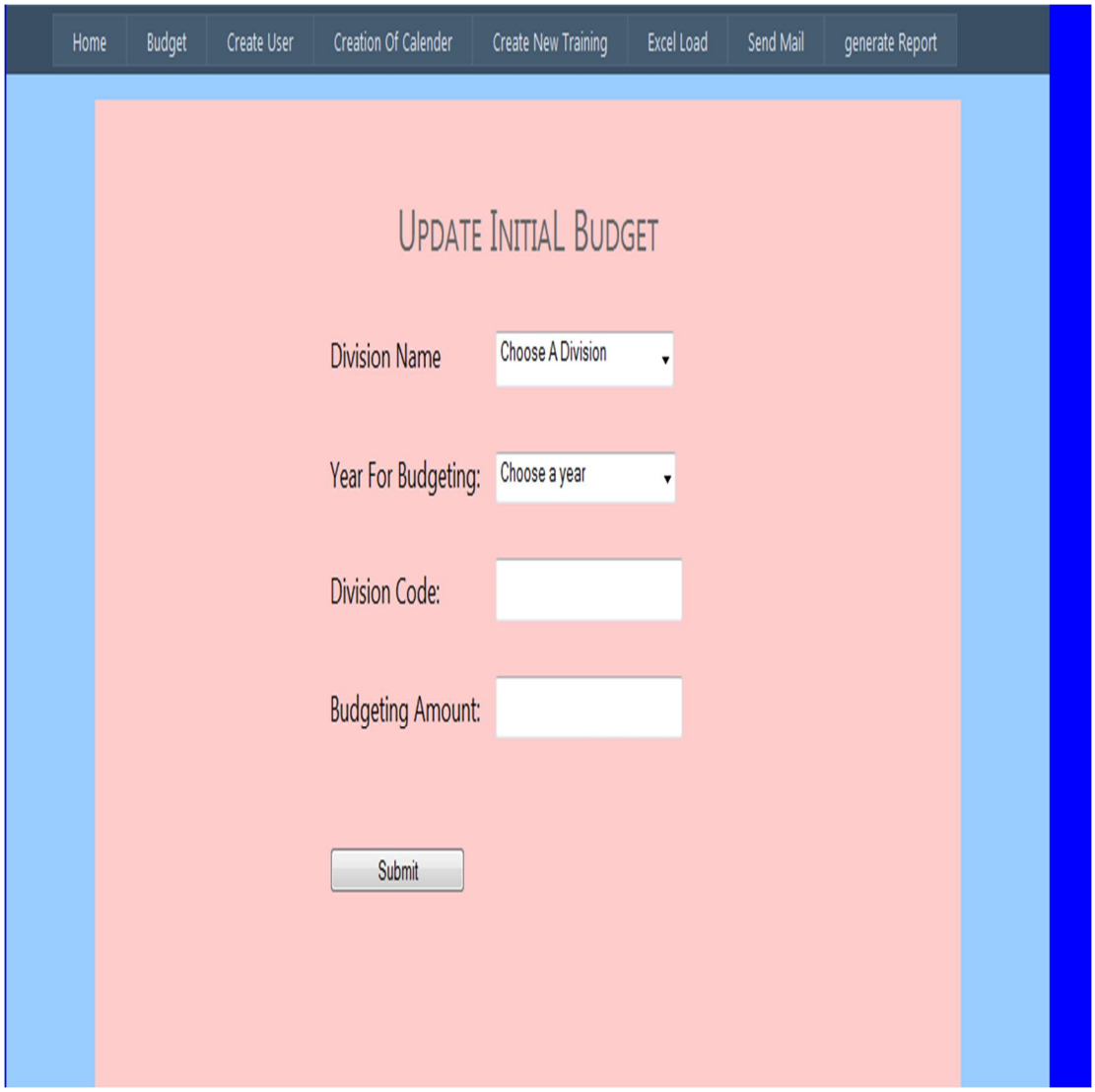


Figure: Updating Initial Budget

6.1.7 Insert Budget with Expense

After completing a training, if cost arises then it is very much reasonable to keep our database in a consistent state. That's why GPIT admin will insert the expense amount of a particular division with a particular year.

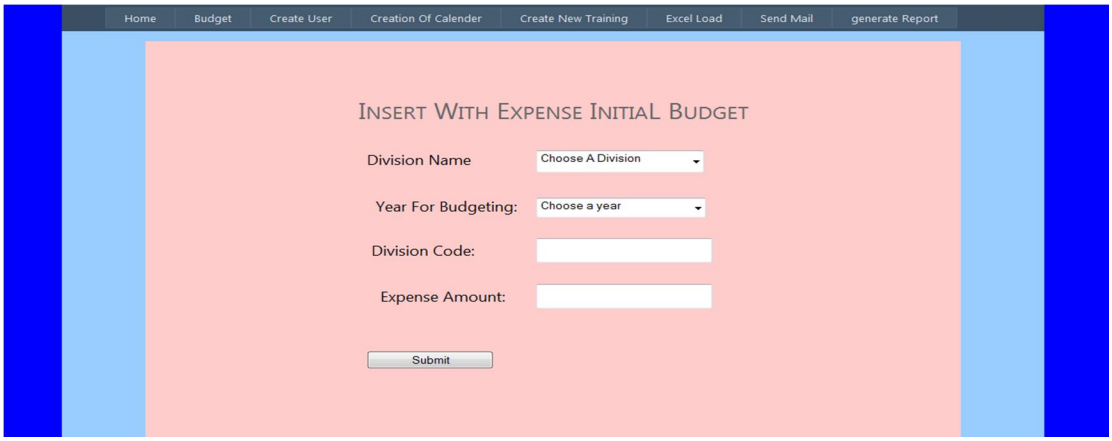


Figure: Inserting Initial Expense amount

6.1.8 New Training Initial Information

When a new training comes some information is about the training and some are about trainees. We divide those training into two phase - one is about training and next is about trainees. When admin or sub-admin creates training, he or she stores information into database.

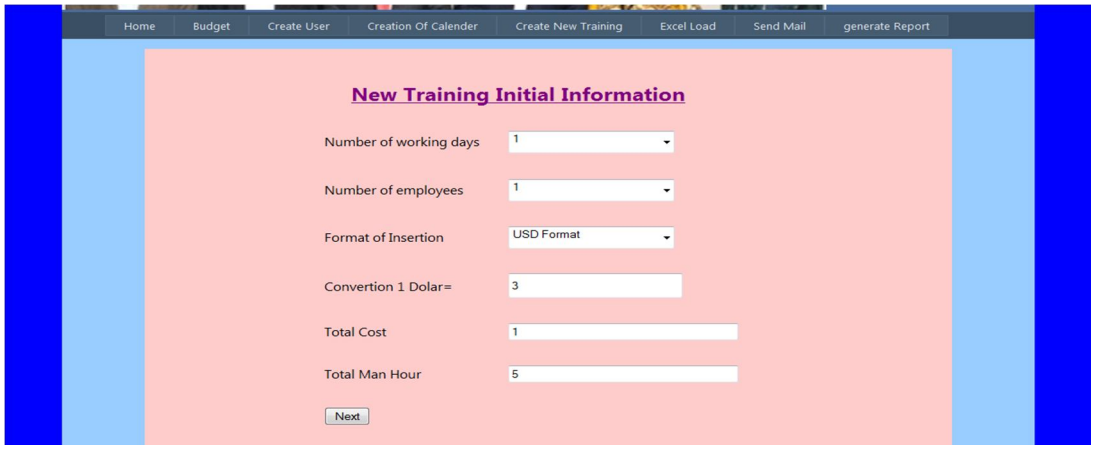


Figure: Inserting New Training Initial Information

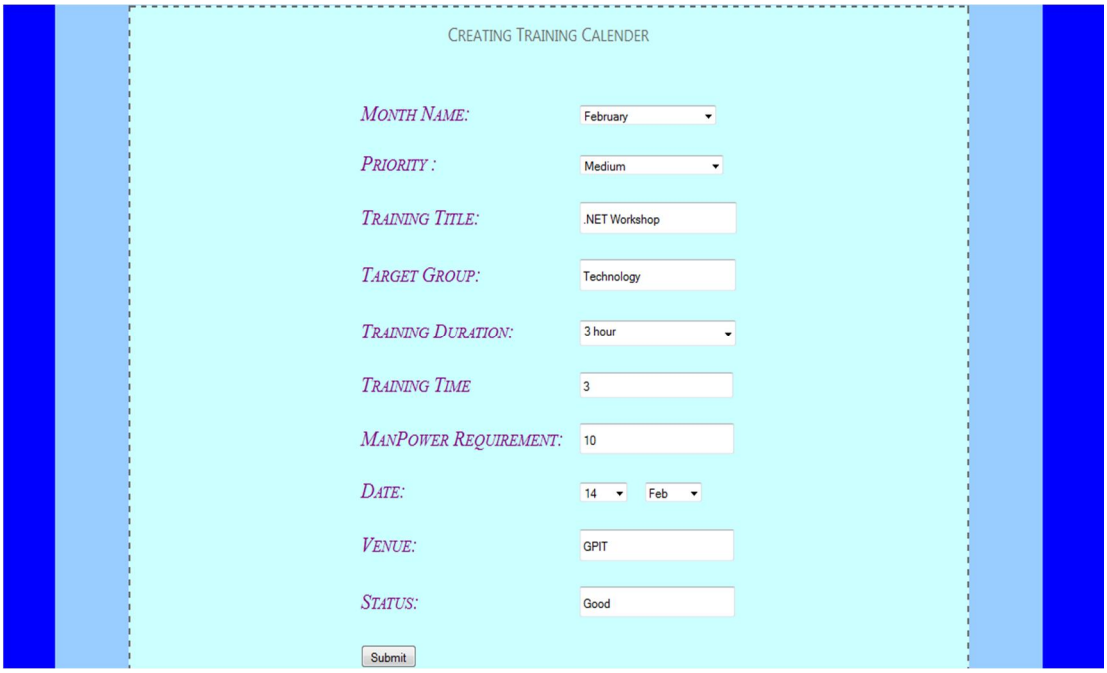


Figure: Inserting Training schedule

6.1.9 Excel Load

Admin can browse an excel file in where GPIT's Employee information or other training information is already stored. For loading an excel file admin have to choose an excel file and then must be clicked insert option. For loading excel, our first page is looks like bellow.

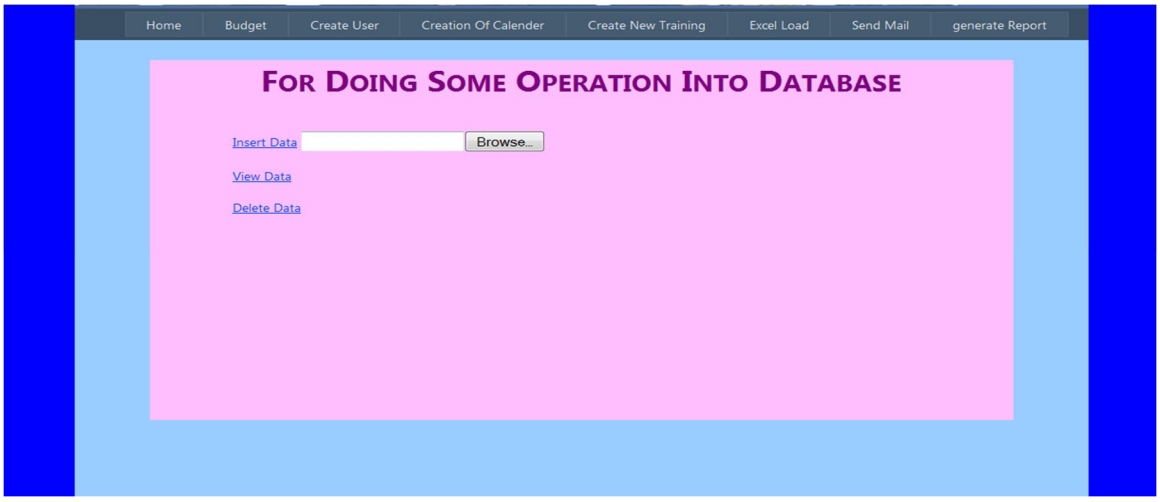


Figure: Inserting Information through Excel File

If without browsing any excel file; user clicks insert button, a message will be shown which is similar to this.

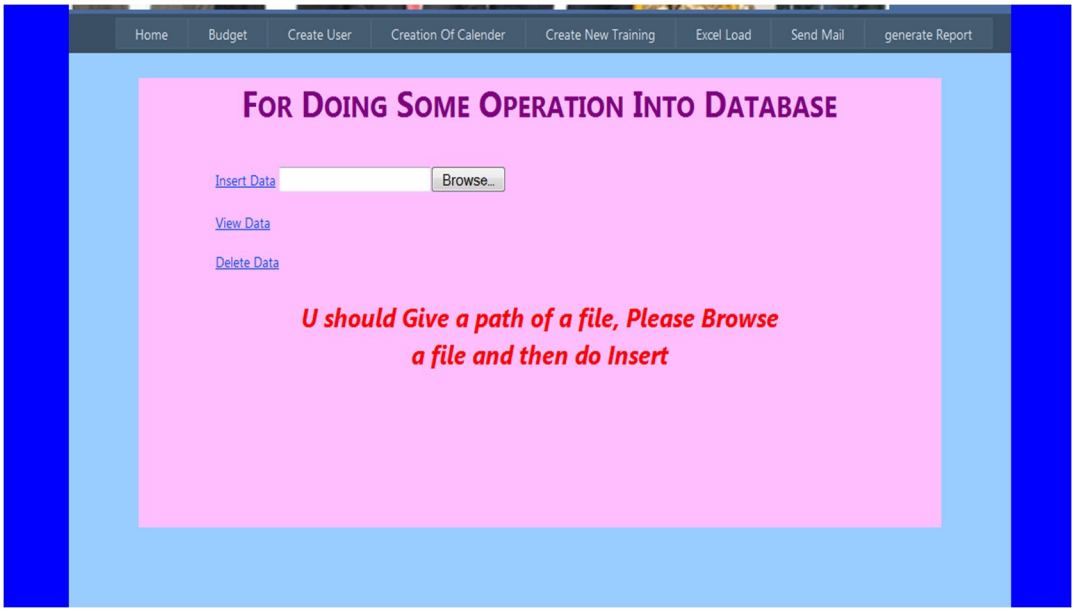


Figure: Result of without choosing any Excel File

After loading an excel file admin can view data any time by clicking a view Data Option. Here we upload a template date and show those data which is given bellow.

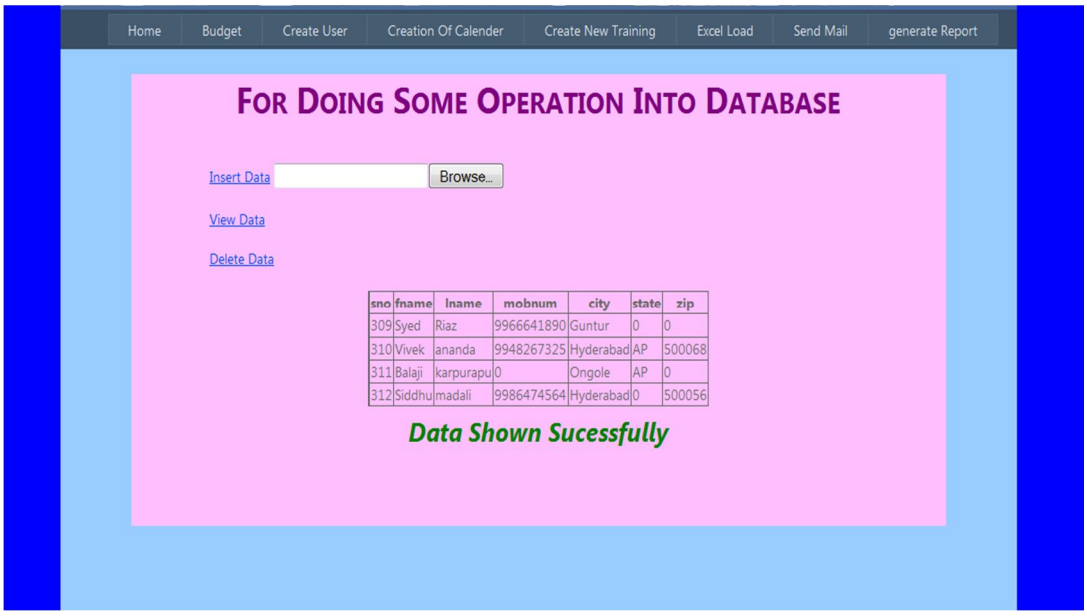
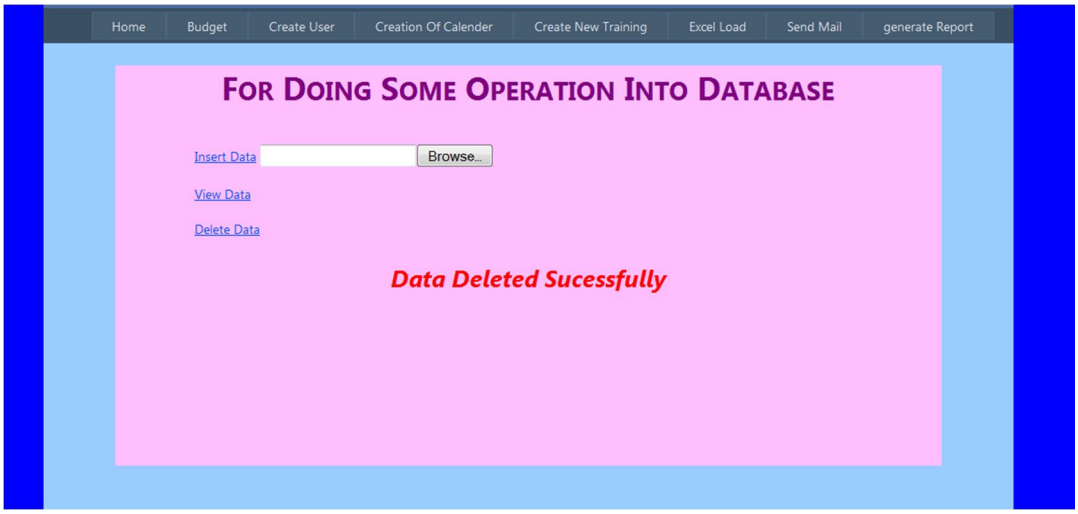


Figure: Showing Data after Insertion

When admin prefers to delete an excel file, he just click a Delete Data and the result is just like.



6.1.10 Sending Notification

Any time admin can send notification to employees through this system. GPIT admin can send an e-mail at a time to multiple employees. It is also possible, if he/she wants to send a file which needs to be attached. Email address is a mandatory field here. If internet is not available, an error message will be shown.

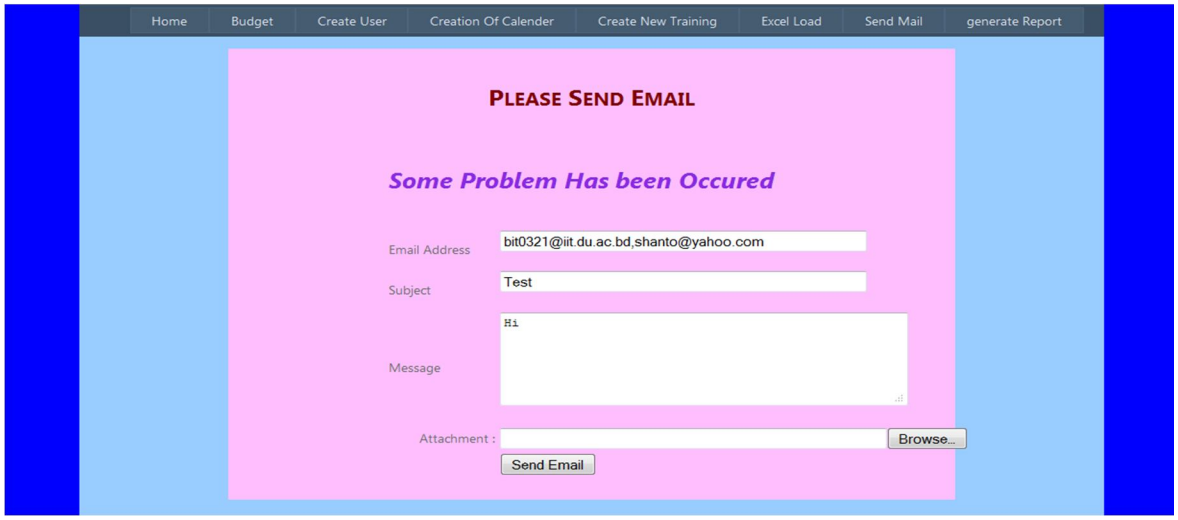


Figure: Error in Sending Notification

If no error occurs, then email has been successfully sent to given email address.

Figure: Sending Notification successfully

6.1.11 Sending Feedback Form

It is very much important for our project. After completing training, it is necessary for authorities to know how training was effective. For those reason a number of selective questions will be sent to all training participant via email. Before giving the title and selecting a date, interface is looks like this.

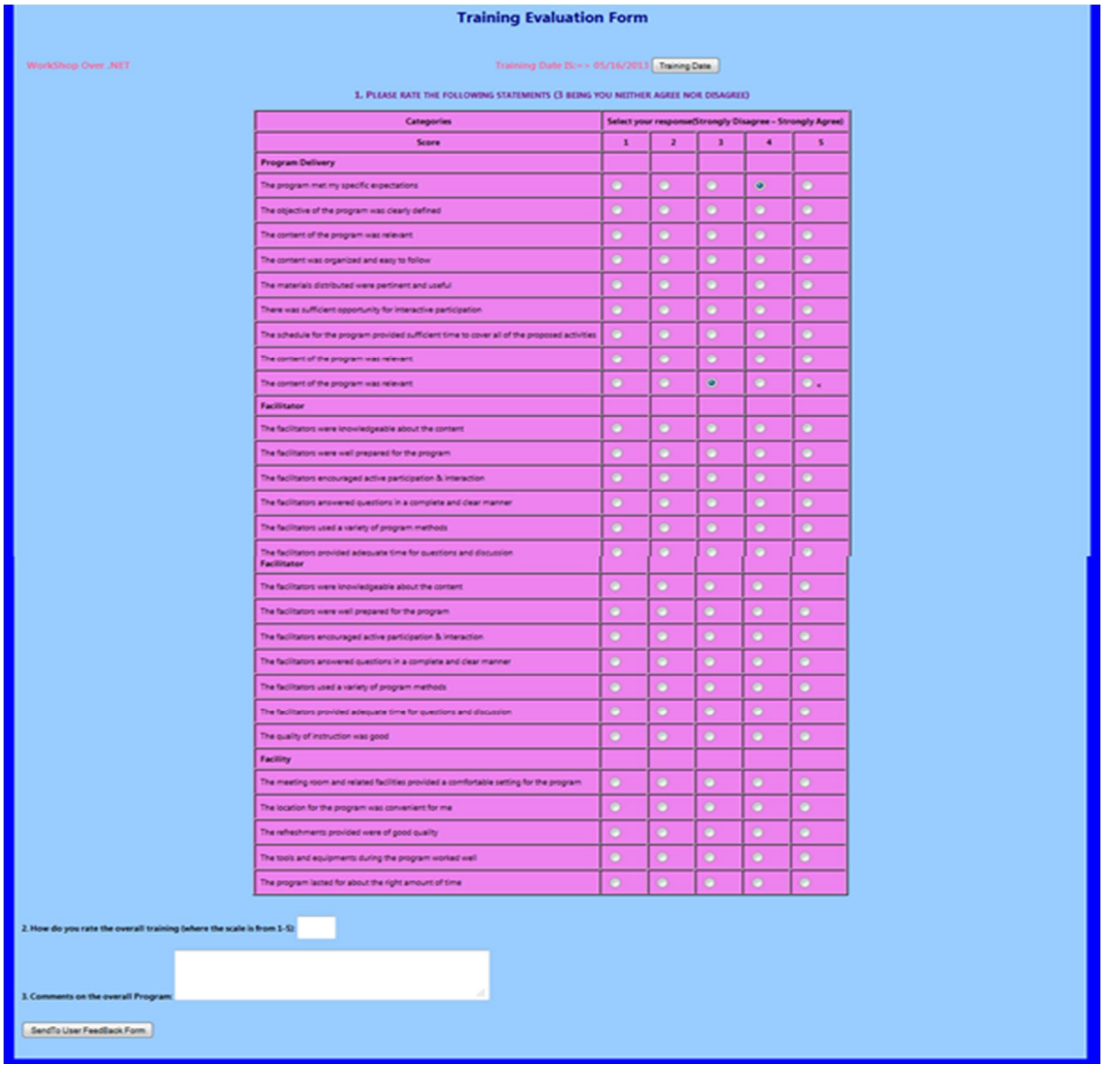
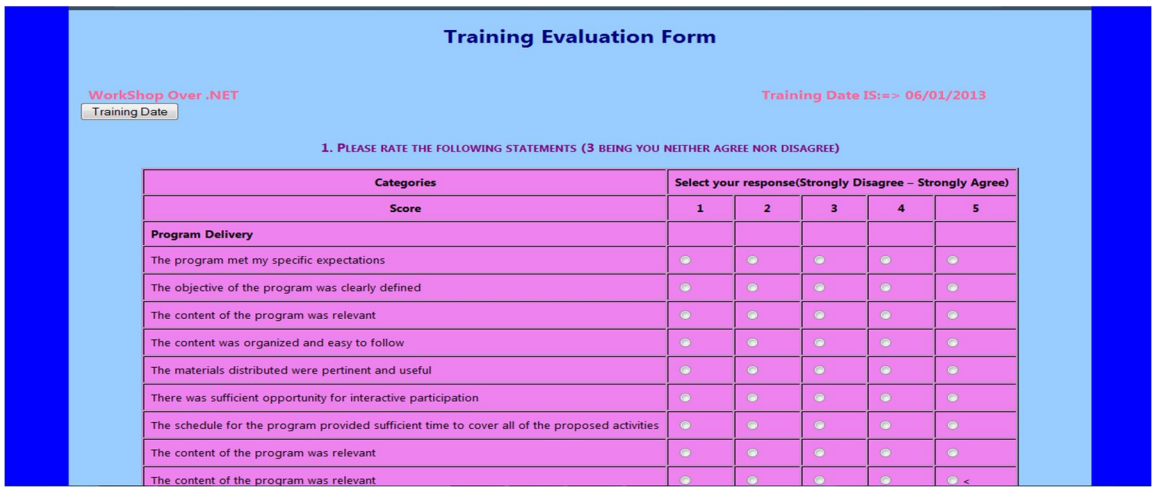


Figure: Sending Notification with feedback form

At first admin will select the training title and date of occurring those training and then send the form to the trainees.



Training Evaluation Form

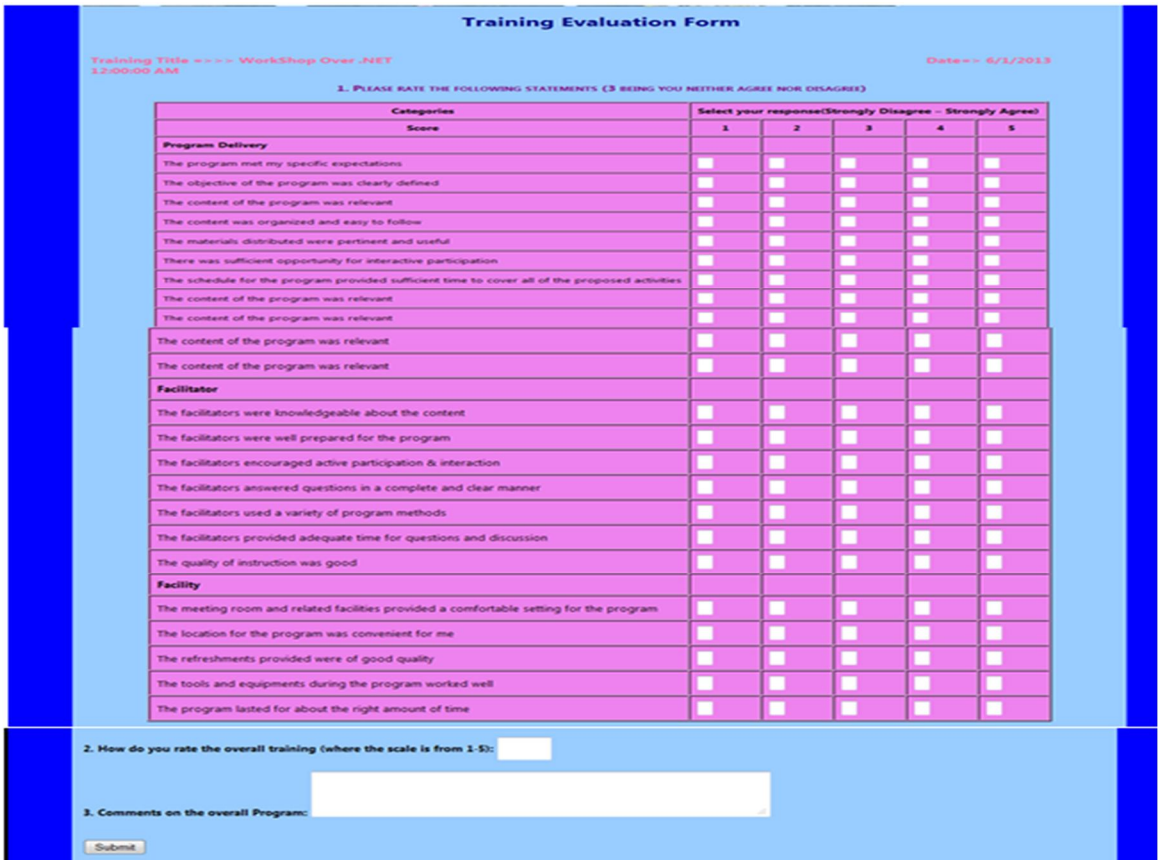
WorkShop Over .NET — Training Date IS:=> 06/01/2013

Training Date

1. PLEASE RATE THE FOLLOWING STATEMENTS (3 BEING YOU NEITHER AGREE NOR DISAGREE)

Categories	Select your response(Strongly Disagree – Strongly Agree)				
Score	1	2	3	4	5
Program Delivery					
The program met my specific expectations	○	○	○	○	○
The objective of the program was clearly defined	○	○	○	○	○
The content of the program was relevant	○	○	○	○	○
The content was organized and easy to follow	○	○	○	○	○
The materials distributed were pertinent and useful	○	○	○	○	○
There was sufficient opportunity for interactive participation	○	○	○	○	○
The schedule for the program provided sufficient time to cover all of the proposed activities	○	○	○	○	○
The content of the program was relevant	○	○	○	○	○
The content of the program was relevant	○	○	○	○	○

Trainees will get this form in their email with a link. By clicking the link trainees can get the feedback form. From trainees form looks like this:



Training Evaluation Form

Training Title =>>>> WorkShop Over .NET 12:00:00 AM — Date=> 6/1/2013

1. PLEASE RATE THE FOLLOWING STATEMENTS (3 BEING YOU NEITHER AGREE NOR DISAGREE)

Categories	Select your response(Strongly Disagree – Strongly Agree)				
Score	1	2	3	4	5
Program Delivery					
The program met my specific expectations	☐	☐	☐	☐	☐
The objective of the program was clearly defined	☐	☐	☐	☐	☐
The content of the program was relevant	☐	☐	☐	☐	☐
The content was organized and easy to follow	☐	☐	☐	☐	☐
The materials distributed were pertinent and useful	☐	☐	☐	☐	☐
There was sufficient opportunity for interactive participation	☐	☐	☐	☐	☐
The schedule for the program provided sufficient time to cover all of the proposed activities	☐	☐	☐	☐	☐
The content of the program was relevant	☐	☐	☐	☐	☐
The content of the program was relevant	☐	☐	☐	☐	☐
The content of the program was relevant	☐	☐	☐	☐	☐
The content of the program was relevant	☐	☐	☐	☐	☐
Facilitator					
The facilitators were knowledgeable about the content	☐	☐	☐	☐	☐
The facilitators were well prepared for the program	☐	☐	☐	☐	☐
The facilitators encouraged active participation & interaction	☐	☐	☐	☐	☐
The facilitators answered questions in a complete and clear manner	☐	☐	☐	☐	☐
The facilitators used a variety of program methods	☐	☐	☐	☐	☐
The facilitators provided adequate time for questions and discussion	☐	☐	☐	☐	☐
The quality of instruction was good	☐	☐	☐	☐	☐
Facility					
The meeting room and related facilities provided a comfortable setting for the program	☐	☐	☐	☐	☐
The location for the program was convenient for me	☐	☐	☐	☐	☐
The refreshments provided were of good quality	☐	☐	☐	☐	☐
The tools and equipments during the program worked well	☐	☐	☐	☐	☐
The program lasted for about the right amount of time	☐	☐	☐	☐	☐

2. How do you rate the overall training (where the scale is from 1-5):

3. Comments on the overall Program:

Submit

Figure: training Evaluation from to trainees

For seeing about developer employer or admin will just click “About” button and can know about ourselves.

Our Supervisor:

Figure: About Us

6.2 Conclusion

In this chapter, we discuss how different modules of our system works, which provide a user guideline to operate the system. We also provide some snapshot of different module. This user manual can help a user of this system to operate the system successfully.

Chapter Seven

GPIT Competence Development Tool Finale Synopsis

A software project means a lot of experience. This chapter summarizes the experience gained by the project team during development of the GPIT Competence Development Tool.

7.1 Critical Evolution

This Competence Development Tool for GPIT is mainly developed with agile development model. So, we need to handle with the requirements during the life cycle of the project which made us complicated to deal with. But we overcome this critical situation with our patient and help of our respective teacher. Moreover, we have faced with the stakeholders confliction as the members of the GPIT have many opinions. Overall, with the agreement of them we can overcome these crucial situations and finally overcome with the product. We had to learn four new technologies ASP.NET, Entity Framework 4.1, ADO.Net, and SQL Server implementation. Learning new technologies cost a lot of work hour of the project team.

7.2 The Achievements and Team Attainment

We have successfully learned the new technologies we needed. Specially, the knowledge of ASP.NET is expected to help us in future. We have gained experience with a new software design approach Entity framework 4.1.

The work was a team work. Every software engineer, being good in team work is not only good, also necessary. The success in distribution of workload and compiling the respective parts together proves that we have learned the habit of team work.

7.3 Limitation and Future Scope

We are considering recovering all known limitations in next versions. Therefore, future works is pointed along with Limitations. The login mechanism is unsecured. Some checking in the web module are done in the server side. However, those checking does not need any database access thus can be done in the client side to optimize server load. As we are developing this software for GPIT, any kind of enhancement they prefer for the software is our future work and scope.

7.4 Last Few Words

The software we developed is intended to serve the GPIT Training Management Team. The success of this project may come to help GPIT, especially those who are maintaining the Training. Therefore, we expect that, anyone using the software will help us reporting any problems/bugs found in it. Also, help us improve the software with constructive comments. For any kind of assistance, feel free to contact any of the developers.

7.5 Conclusion

Finally we can say that our project team of Competence Development Tool have endeavored a lot for the completion of this project and we are very much pleased that this system is going to be run as not only as a pilot project but also run for the GPIT Competence Development. With all other complications at the end we are supposed to fulfill the clients' satisfaction level with the versions and requirements building. Again, from this document we assure that other developer team should be helpful to enhance Competence Development Tool.

References

Book

- **"Software Engineering – A Practitioner's Approach-7th Edition"** by Roser S. Pressman
  Publisher: McGraw Hill

SRS

- IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

Requirements Gathering and Analysis

- http://www.visual-paradigm.com/VPGallery/diagrams/Activity.html
- http://leanlogisticsblog.leancor.com/2010/09/01/how-to-create-a-swim-lane-diagram/
- www.webopedia.com/TERM/E/**entity_relationship_diagram**.html
- www.visual-paradigm.com/product/lz/tutorials/**dfd**.jsp
- www.nikhef.nl/~p63/www/STD.html
- http://www.gatherspace.com/static/software_requirement_specification.html
- http://www.hi-teck.com/soft_development/req_home.jsp
- https://goldpractice.thedacs.com/practices/mr/index.php
- http://www.softwareprojects.org/software-requirement-management-01.htm
- http://www.serena.com/solutions/requirements-management/

Software Development

- http://www.php.net
- http://www.smarty.net/
- http://msdn.microsoft.com/en-us/library/ff649643.aspx
- http://adodb.sourceforge.net
- http://en.wikipedia.org/wiki/.NET_Framework
- http://en.wikipedia.org/wiki/Microsoft_Visual_Studio
- http://en.wikipedia.org/wiki/Microsoft_SQL_Server
- http://en.wikipedia.org/wiki/Internet_Information_Services

Network Diagram

- http://www.edrawsoft.com/Network-Diagrams.php
- http://en.wikipedia.org/wiki/Network_diagram
- http://www.networkworld.com/subnets/cisco/053008-ch1-ccna-prep-library.html