|  |  |
| --- | --- |
|  | **MINISTRY OF EDUCATION AND TRAINING** |

|  |
| --- |
| **FPT UNIVERSITY** |

Time and Attendance System

Software Design Description

Project code: FUTAS

Document code: FUTAS-SDD

Hà Nội, January, 2013

Table of content

**4.1. Design Overview**

**4.2. System Architectural Design**

4.2.1. Choice of System Architecture

4.2.2. Discussion of Alternative Design

4.2.3. Description of System Interface

**4.3. Class Diagram**

4.3.1. Report

4.3.2. Salary & Man-day

4.3.3. Shift

4.3.4. Employee & Group

4.4. Sequence Diagram

4.4.1. Report

4.4.2. Salary & Man-day

4.4.3. Shift

4.4.4. Employee & Group

**4.5. User Interface Design**

**4.6. Database Design or Data Structure**

4.6.1. Table detail

4.6.2. Design diagram of database

**4.1. Design Overview**

* FUTAS contains many clients and a server. Clients here are reader in office. A client includes reading head and stripe card. Server here stored all client transaction, information scanned card, database.
* Client connect to server via Ethernet only
* Client connect to client via Ethernet only
* Client connect to server
* Server always starts and listens to the connection
* Validate the client connect
* Create a thread for process connection

**4.2. System Architecture Design**

**4.2.1 Choice of System Architecture**

* FUTAS uses client-server model, data centered architecture

**4.2.2 Discussion of Alternative Designs**

1. Server

* Why using C#

The general rule is to always use the highest level language that provides satisfactory performance and stability.

C# defines an extensive set of standard libraries that implement critical functionality that other language library does not even mention.

C# is distributed in forms that are processor architecture and operating system independent. For high end commercial closed source software, this is a major advantage. Most leading edge technology companies would rather die than let their customer look at the source code.

C# is an elegant, simple, type-safe, objected-oriented language that allows enterprise programmer to build a breadth of applications.

C# also gives you the capability of building durable system-level components by virtue if the following features:

* Full COM/Platform support for existing code integration.
* Robustness through garbage collection and type safety.
* Security provided through intrinsic code trust mechanisms.
* Full support of extensible metadata concept.
* You can also interoperate with other languages, across platforms, with legacy data, by virtue of the following features.
* XML support for Web-based component interaction.
* Versioning to provide ease of administration and deployment.
* TCP Server Demo

In this tutorial I’m going to show you how to build a threaded TCP server with C#. If you have ever worked with Window’s sockets, you know how difficult this can sometimes be. However, thanks to the .NET framework, making one is a lot easier than it used to be.

Let’s just see some code. Below is the basic setup for our TCP server class.

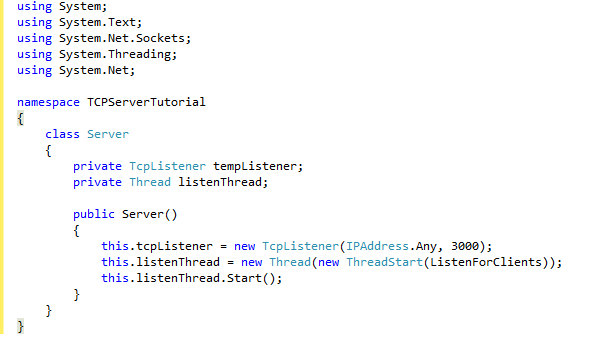


Figure 4.1 TCP server

1. Data

The data centered software architecture is characterized by a centralized data store which is shared by all surrounding software component

**4.2.3 Description of System Interface**

a. Data flow

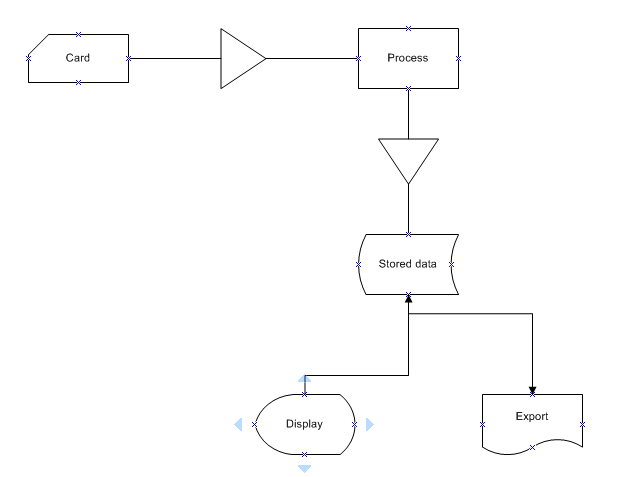


Figure 4.2 Data flow

b. Modules

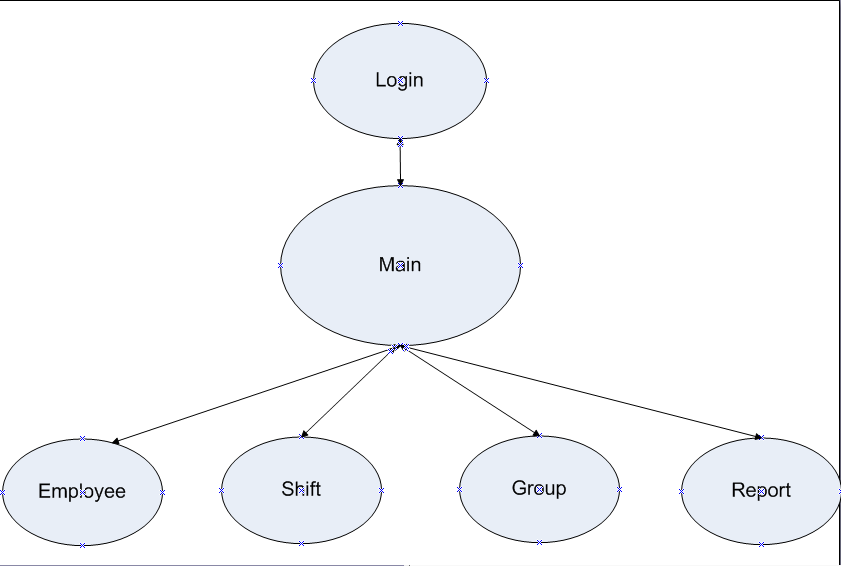


Figure 4.3 Modules

* FUTAS include 4 main module :
* Employee management
* Shift management
* Group management
* Report management

1. Use Case Diagram

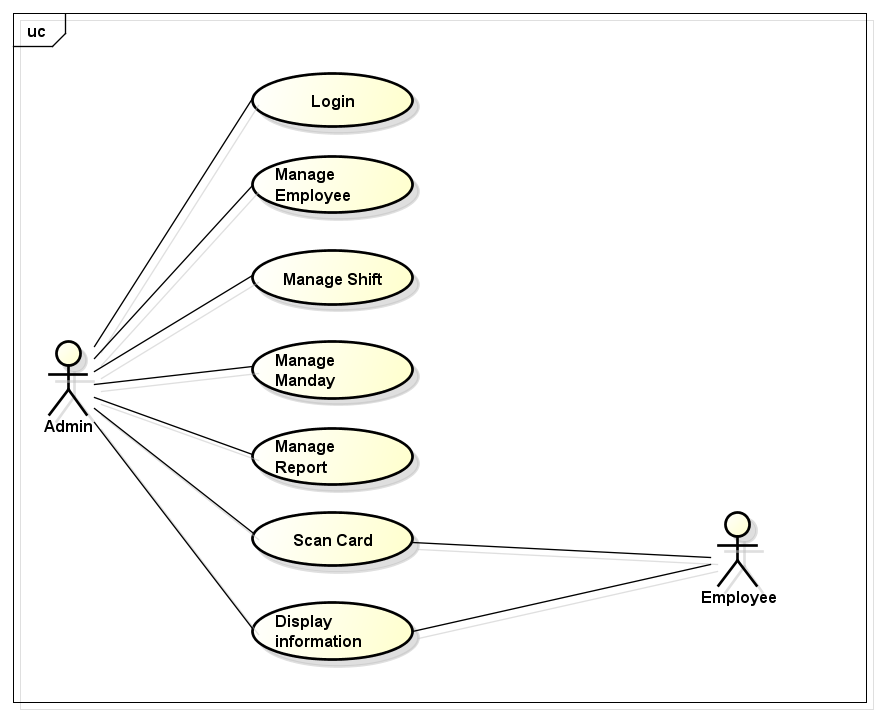


Figure 4.4 Use Case Diagram

* FUTAS is only used by only administrator. Another employees want to view something have to contact to administrator

**4.3. Class Diagram**

**4.3.1. Report**

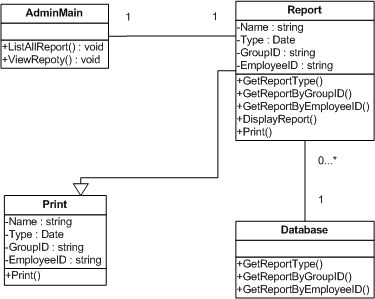
****

Figure 4.5 Report Class Diagram

**4.3.2. Salary & Man-day**

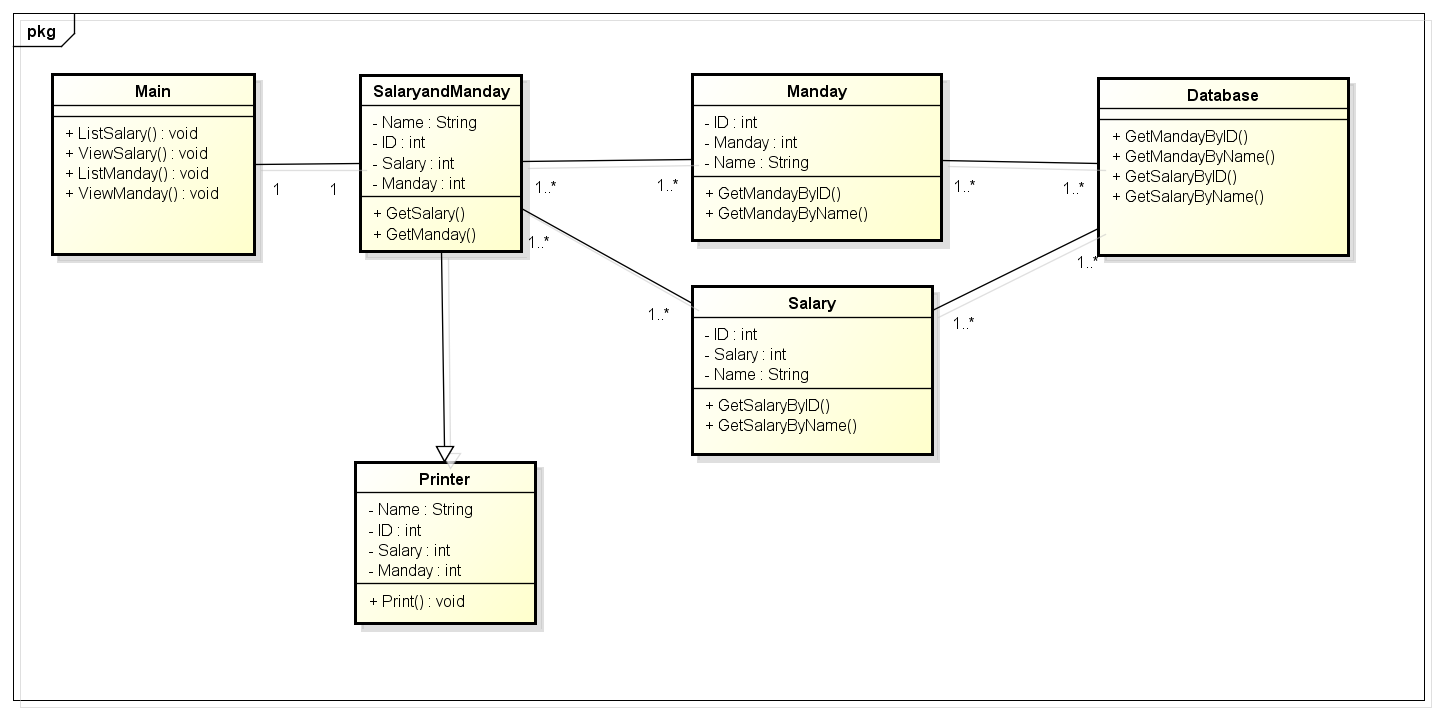
****

Figure 4.6 Salary & Man-day Class Diagram

**4.3.3. Shifts**

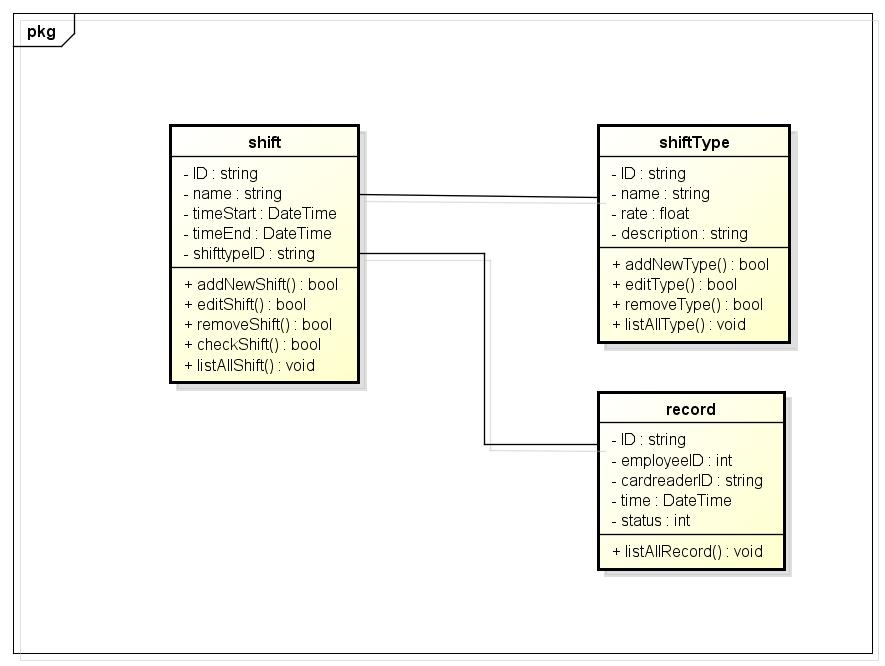
****

Figure 4.7 Shift Class Diagram

**4.3.4. Employee & Group**

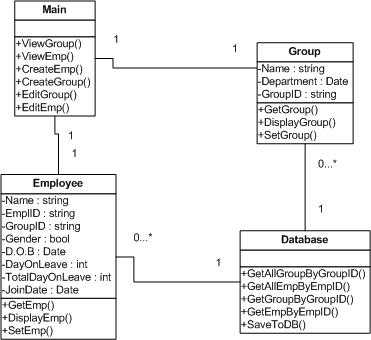
****

Figure 4.8 Employees & Group Class Diagram

**4.4. Sequence Diagram**

**4.4.1. Report**

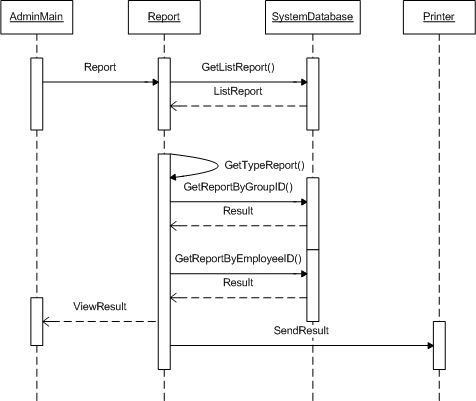
****

Figure 4.9 Report Sequence Diagram

**4.4.2. Salary & Man-day**

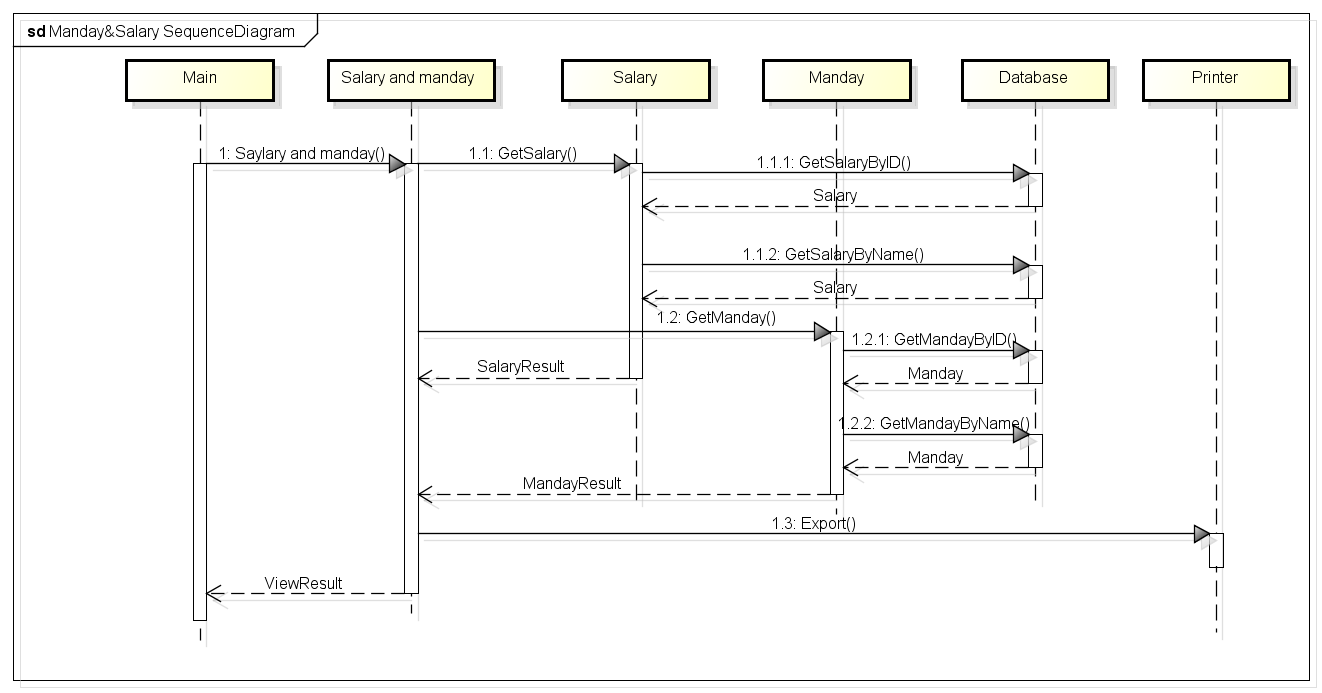
****

Figure 4.10 Salary & Man-day Sequence Diagram

**4.4.3. Shifts**

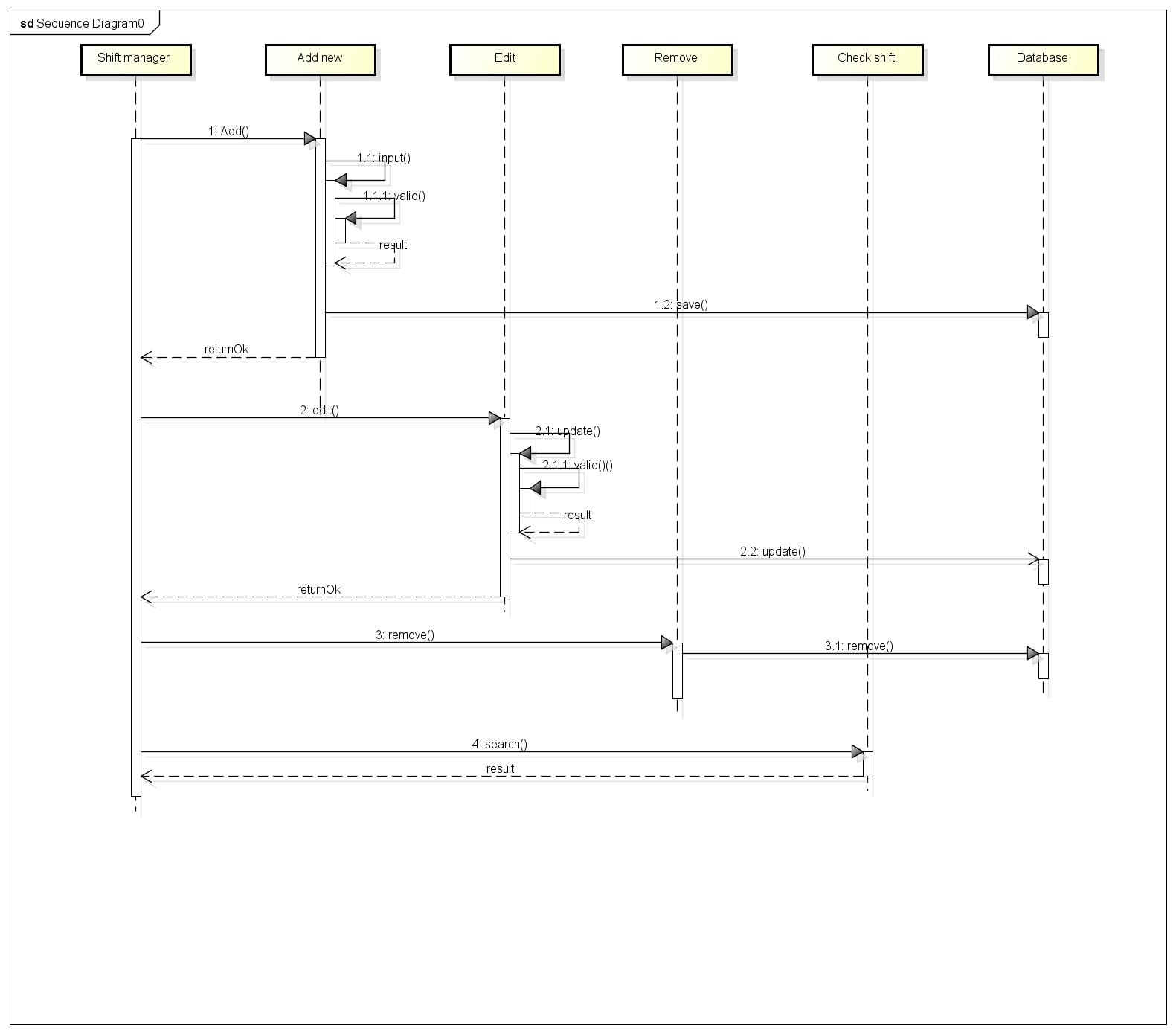
****

Figure 4.11 Shift Sequence Diagram

**4.4.4. Employee & Group**

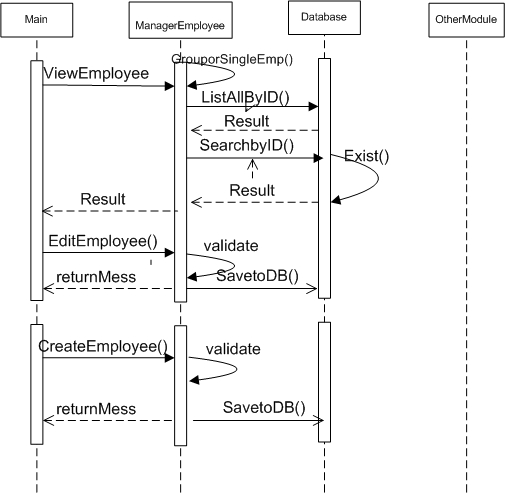
****

Figure 4.12 Employees & Group Sequence Diagram

**4.5. User interface design**

1. **Main menu**

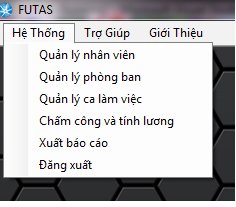
****

Figure 4.13 Main menu screen

* On top menu, FUTAS include:
* Menu : Click and select module
* Help : Click and view guidance
* About

1. **Employee Management**

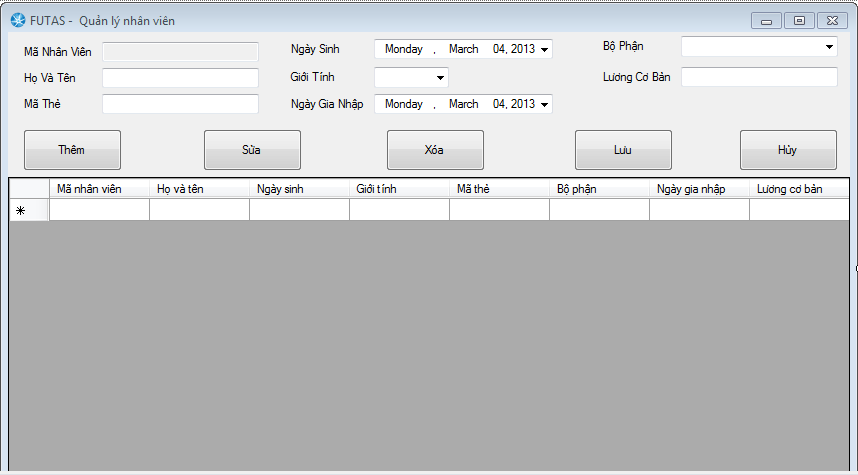
****

Figure 4.14 Employee management screen

1. **Group Management**

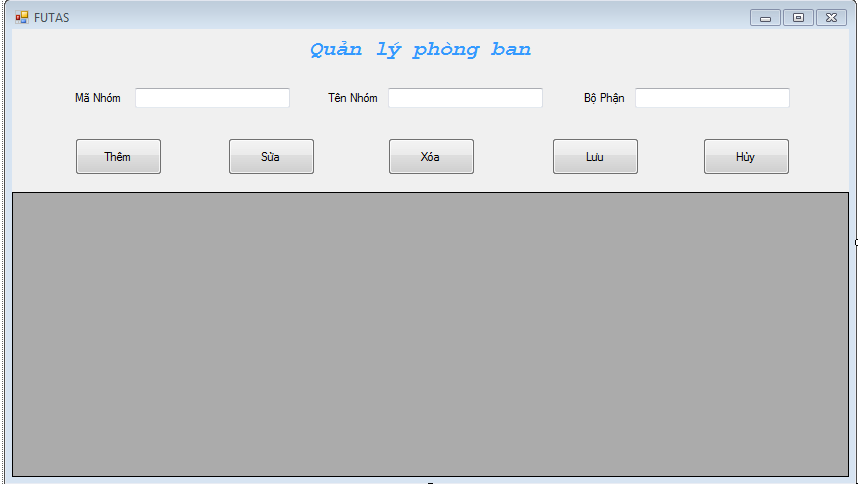
****

Figure 4.15 Group management screen

1. **Shift Management**

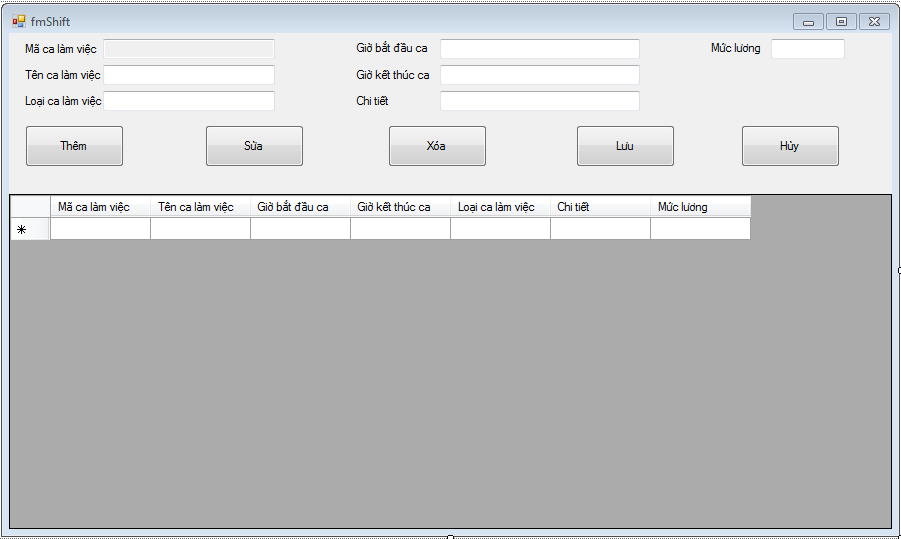
****

Figure 4.16 Shift management screen

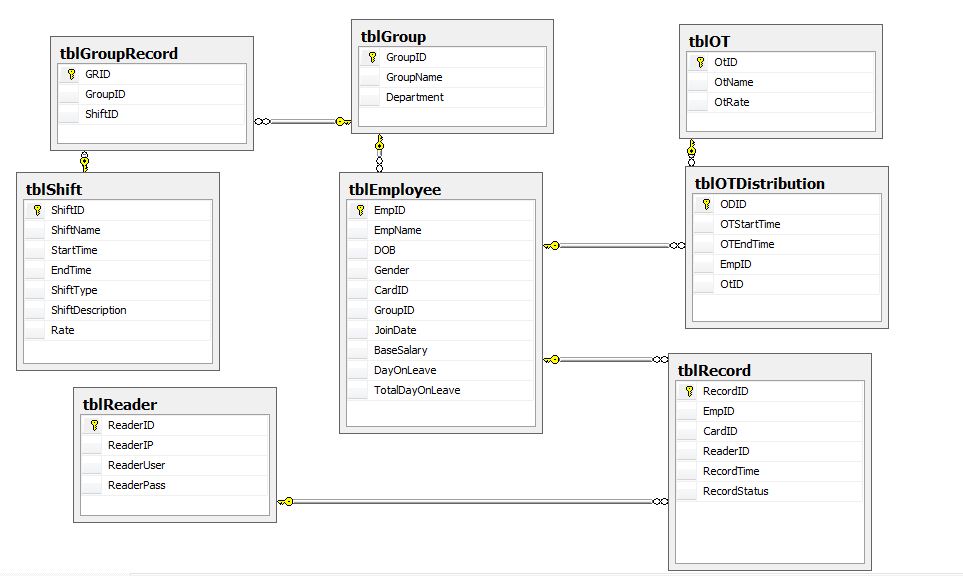
1. **Report Management**
2. **Man-day & Salary Management**

**4.6. Database design**

**4.6.1. Table Detail**

* Employee
  + ID(Primary key)
  + NAME
  + DOB
  + JOINT DATE
  + GROUP ID
  + CARD ID
  + DAY ON LEAVE
  + TOTAL DAY ON LEAVE
  + BASE SALARY
  + GENDER
* GROUP
  + GROUP ID(Primary key)
  + NAME
  + DEPARTMENT
* SHIFT
  + SHIFT ID(Primary key)
  + NAME
  + TIME START
  + TIME END
  + SHIFT TYPE
  + RATE
  + DESCRIPTION
* Group Record
  + ID(Primary key)
  + GID
  + Shift ID
* RECORD
  + ID(Primary key)
  + EMPLOYEE ID
  + READER ID
  + CARD ID
  + TIME
  + STATUS (ONTIME/ LATE)
* READER
  + CARD READER ID(Primary key)
  + USER
  + PASS
  + IP
* OVERTIME
* OVERTIME ID(Primary key)
* NAME
* RATE
* OVERTIME DISTRIBUTION
* OVERTIME DISTRIBUTION ID(Primary key)
* START TIME
* END TIME
* EMPLOYEE ID
* OVERTIME ID

**4.6.2. Design diagram of database**

****