

Build an Inventory Tracking System

Hanzheng Zou

Build an Inventory Tracking System

Hanzheng Zou
School of Mathematics and Systems Engineering
Vaxjo University

Abstract:

This thesis paper introduces you about the process of how to build an inventory tracking system in a local Swedish company. The related project is to support the thesis paper, and is also for the company' using. The software product of this project is an application that works for managing various types' instruments in the company of SWECO-Vaxjo. It will play an important role in the further management work of the company.

In this thesis paper, the candidate techniques and theories for implementing this system are discussed. And in the end a good solution for this problem will be presented in the paper.

Keywords: Database, Microsoft Access, ODBC, Visual studio 2005, VC++, MFC, UPEDU, DBMS.

Acknowledge:

First I should say thank you to my supervisor Jesper Andersson. I am very happy that I get a so practical project from him, and then I can do the interesting work, do a lot of research about this topic and use it as my thesis paper. I can use my knowledge about UPEDU and C++ programming to do a real job. And I get many suggestions and advices from Jesper when I am doing the project. When I started to write my thesis, he gave me much idea but how the structure should be and help me to get start. He spent a lot of time on reading and correcting my thesis. Without his help, I can't finish my thesis on time.

Then I will give my regards to Andreas Petersson and Daniel Ottosson who are from the Swedish company. We had a nice communication and free talking, help me to understand the requirements and build the system for them. They are good men and even some times come to the University to talk about the project with me. They really did a lot and help me a lot for this project.

About writing a thesis paper, I didn't have much experience. Mathias Hedenbog gave me a lot of useful suggestions about writing a paper. I am appreciated that he spent a lot of time to read my paper. My paper opponent Francisco also helps me a lot to correct the mistakes in the paper.

Table of Contents

TABLE OF CONTENTS	V
LIST OF FIGURES	VII
LIST OF TABLES	VIII
1 INTRODUCTION	1
1.1 INTRODUCE THE PROJECT	1
1.2 SWECO COMPANY BACKGROUND	1
1.3 <i>Research on the problem</i>	2
1.3.1 <i>General Research</i>	2
1.3.2 <i>Constraints & Delimitations</i>	2
1.4 GOALS AND MOTIVATIONS	2
1.5 OUTLINES.....	3
2. THEORY (RELATED WORK)	4
2.1 POSSIBLE TECHNIQUES.....	4
2.1.1 <i>Database</i>	4
2.1.2 <i>Database Connection Technique</i>	6
2.2 TOOLS AND PROGRAM LANGUAGES FOR DOING THE PROJECT	8
2.2.1 <i>Java & Eclipse</i>	8
2.2.2 <i>C++ & Visual studio</i>	8
2.2.3 <i>Visual Basic</i>	9
2.3 CANDIDATE SYSTEM SOLUTIONS.....	9
2.3.1 <i>Client- Sever System</i>	9
2.3.2 <i>System running in Server Machine</i>	9
2.4 METHOD AND PLANNING OF THE PROJECT	10
2.4.1 <i>Method of dong the project</i>	10
2.4.2 <i>Planning of the project</i>	11
3. SOLUTION OF THE PROJECT	12
3.1 REQUIREMENT.....	12
3.1.1 <i>Requirement Elicitation</i>	13
3.1.2 <i>Software Requirement Specification</i>	15
3.2 SELECT METHOD AND TECHNIQUE.....	17
3.3 ARCHITECTURE	18
3.4 DESIGN	20
3.4.1 <i>Analysis Class</i>	21
3.4.2 <i>Database Connection</i>	23
3.4.3 <i>Add Data</i>	24
3.4.4 <i>Search Data</i>	25
3.4.5 <i>Service Dealing</i>	26
3.4.6 <i>Booking</i>	26
3.4.7 <i>Report</i>	27

<u>3.5 ACCESS DATABASE DESIGN</u>	27
<u>3.6 IMPLEMENTATION</u>	28
<u>3.6.1 Add instrument to database</u>	28
<u>3.6.2 Search</u>	29
<u>3.6.3 Print</u>	29
<u>4. EVALUTATION AND RESULT</u>	30
<u>4.1 PROBLEMS IN THE PROJECT</u>	31
<u>4.2 EVALUATION</u>	31
<u>4.2 RESULT AND DELIVERY</u>	32
<u>5. DISCUSSION & CONCLUSION</u>	33
<u>5.1 CONCLUSION</u>	33
<u>5.2 FUTURE WORK</u>	33
<u>6. REFERENCES</u>	35
<u>APPENDIX</u>	36
<u>1. WORKFLOW OF THE COMPANY</u>	36
<u>2. SOURCE CODE OF HOW TO ADDING DATA USING ODBC & MFC</u>	37
<u>3. SNAP OF THE SOFTWARE PRODUCT</u>	38

List of Figures

<u>FIGURE 3.1 NETWORK ARCHITECTURE IN THE COMPANY</u>	13
<u>FIGURE 3.2 USE CASE DIAGRAM OF THE PROJECT</u>	14
<u>FIGURE 3.3 ARCHITECTURE OF THE SYSTEM.</u>	15
<u>FIGURE 3.4 SYSTEM DESIGN DIAGRAM</u>	16
<u>FIGURE 3.5 ANALYSIS CLASSES DIAGRAM</u>	17
<u>FIGURE 3.6 CLASS DIAGRAM OF DATABASE CONNECTION PACKAGE</u>	19
<u>FIGURE 3.7 CLASS DIAGRAM OF ADD DATA PACKAGE</u>	20
<u>FIGURE 3.8 CLASS DIAGRAM OF SEARCH DATA PACKAGE</u>	21
<u>FIGURE 3.9 CLASS DIAGRAM OF SERVICE DEALING PACKAGE</u>	22
<u>FIGURE 3.10 CLASS DIAGRAM OF REPORT CREATING PACKAGE</u>	23
<u>FIGURE 3.11 DATABASE CONNECTION DIAGRAM</u>	24

List of Tables

<u>TABLE 2.1 TIME SCHEDULE OF THE PROJECT</u>	11
<u>TABLE 3.1 DETAILED DESCRIPTION OF SYSTEM DESIGN DIAGRAM</u>	16
<u>TABLE 3.2 DETAILED DESCRIPTION OF ANALYSIS CLASSES DIAGRAM</u>	18
<u>TABLE 3.3 DETAILED DESCRIPTION OF DATABASE CONNECTION PACKAGE</u>	19
<u>TABLE 3.4 DETAILED DESCRIPTION OF ADD DATA PACKAGE</u>	20
<u>TABLE 3.5 DESCRIPTION OF SEARCH DATA PACKAGE</u>	21
<u>TABLE 3.6 DESCRIPTION OF SERVICE DEALING PACKAGE</u>	22
<u>TABLE 3.7 DETAILED DESCRIPTION OF REPORT CREATING PACKAGE</u>	23
<u>TABLE 4.1 SYSTEM FUNCTION REALIZATION</u>	27

1 Introduction

The paper showing researches about how to build an inventory tracking system for a Swedish company. It introduces you the process of implement database management software, and in the end I give you the solution using Microsoft Access and VC++ are proved to be good one. Also there may be other techniques to realize this project. This thesis paper will based on a database project for a local company, then we can see how to design and realize a project for a company's using. We will see some other related techniques that can realize the project, discuss the advantages and disadvantages of each technique, and you can also see why we choose Access and VC++ to implement this project.

In the end of the project, you can see the software product that comes from the combination of Access and VC++. This will also be an achievement of a Chinese trainee and a Swedish Company.

1.1 Introduce the Project

This is a bachelor degree paper discussing about how to use Microsoft Access as the database and use VC++ to do the GUI stuff.

The Project is for this thesis topic. It is for SWECO VIAK which is Swedish local company. And the product of the project will be used by the worker of the company for managing the instruments, and the instruments are brought from other distributors and will be sold or leased to some of the other customers. Then the software will help the work of the people in the company a lot in their regular works. The software can keep the records of all the instruments for their buying, leasing, selling or repair, and help the people in the company do some planned service in certain time.

During the project, the developer and the company will follow the RUP process, to make it efficient to finish the work.

1.2 SWECO Company Background

SWECO is the Nordic region leading Consult Company with gathered knowledge in technique, environment and architecture. With 4000 employees the company actively participate in the process of develop a persevering society and an effective industry. The company's fellow worker supply solutions that give the functional and cultural buildings and facilities, cost effective energy systems, better environment and efficient and profitable production. The SWECO company' market is the Nordic region but a comprehensive foreign operation manages round the world. SWECO VIAK in Växjö is the central supply of instruments for measure water, waste water and rain. The company both lease and sell instruments. They lease to other SWECO office, other external companies and to social communities through internal assignments. Each instrument has a unique identification number based on the type of instrument and if it's a leasing or an instrument for sale.

The company has a quality system for maintaining the accuracy of each instrument. In periodically controls of people in the company will test different sensors to make sure that they fulfill the accurateness between different measure intervals.

1.3 Research on the problem

This section will discuss about the general problem of the project. You will see some constraints and delimitations that possible to happen when building this project.

1.3.1 General Research

In the company, there is a Microsoft Access instruments database in using. The database is built a while ago and there a few people who have worked with it. The problem of today's database is that there is no built in function for several functions. Then the problem is to add some new functions to the old system, use new techniques to realize a new system that designed for the company.

There are several new functions need to be implement, such as Set Current Status, Traceable history of an instrument, Booking function, Alarm function etc. Our goal is to solve the problems and build a stable system for the company.

The company needs the application to be a Service-side application. That means this application will be run in a local network. And the database in the server side can be visited by different user using different computer. Then we should also consider about the local network condition.

1.3.2 Constraints & Delimitations

Here we will discuss two possible Constraints and Delimitations in implementing the project. One is local network, the other one is server machine.

- **Local Network**

This system should be run in the server and the client visit the database through the local network. Then the local network should be always working. The system working properly depends on the local network working fine.

- **Server Machine overload**

There are several applications running in the server machine, and the clients PC always visit the server machine through the network. Then the server machine may possible be overload some time.

1.4 Goals and Motivations

Here we will discuss about the goals and Motivations about the thesis and the related real project.

The main goal of this thesis paper is how to use Microsoft Access and VC++ to do realize a software product. This means we will use Access play as the database and use VC++ to draw the GUI. Then in the paper we will discuss a lot of things about database techniques, and think about which one is better for the current requirements.

The other goal is learning to use RUP process to solve the software problem. This means think a way to solve the problem, adapt the RUP process to the current condition.

This is also a good chance that the product can benefit the company workers a lot in their regular work.

1.5 Outlines

Here I will introduce the structure of the thesis paper.

The Chapter 1 is introduction part, the background of the project and some problems are discussed here.

The Chapter 2 is something about the Theory and some related work and techniques. In this chapter, we discuss about some other candidate databases and techniques that can also realize the project. We also discuss something about the UPEDU process and schedule of the project.

The Chapter 3 gives the solution for the project. You can see the requirement, design and implementation detail in this chapter. The Database design can be also seen here.

The Chapter 4 you will see the result of the thesis project. Then you can know how well the design and implementation fulfill the requirements.

The Chapter 5 discusses about the conclusion of the thesis.

2. Theory (Related Work)

This chapter I will introduce you some theory of doing a database management system. That means what database we should choose, what programming language we should use, and what is the candidate system solutions. And we will also talk something about UPEDU process.

2.1 Possible Techniques

After doing some researches on this project, the project is known as a typical local network database system. Then there are several possible solutions and database techniques to do this project.

2.1.1 Database

There are several databases are possible to store the system data. We can use Microsoft SQL-server database, MYSQL database, Microsoft Access database, Excel file or even use the Oracle database as the database of the project.

2.1.1.1 Microsoft SQL-server Database

Microsoft SQL-server is a database management system developed by Microsoft Corporation. SQL server uses Transact-SQL as its primary query language. It usually used by business and mostly for the small or medium size database. But as the technique developed quickly, the SQL-server now is more and more adopted to the larger enterprise database. And the new version of the SQL-server is SQL 2005.

And for SQL 2005, we can see:” SQL Server 2005 is a comprehensive database software platform providing enterprise-class data management and integrated business intelligence (BI) tools. The SQL Server 2005 database engine provides more secure, reliable storage for a relational database format or XML. By supporting both a relational database format and XML, the SQL Server 2005 database engine provides the flexibility necessary to support the way you work. It also enables you to build and manage highly available, well-performing database software applications that you and your people can use to take your business to the next level.”¹

It’s very good to use SQL-server database as the database resource if the developer use Visual studio to build the project. We can see:” Close integration with Microsoft Visual Studio, the Microsoft Office System, and a suite of new development tools, including the Business Intelligence Development Studio, sets SQL Server 2005 apart. Whether you are a developer, database administrator, information worker, or decision maker, SQL Server 2005 provides innovative solutions that help you gain more value from your data—whether stored in a relational database or XML data format.”²

So this is very suggested to use this as the database resource, if the developers are

¹ Web reference from Microsoft SQL home Page

² Web reference from Microsoft SQL home Page

familiar with C++, VB and using Visual studio to create and build the project. Still there is some disadvantages to use it, it need the users to install the SQL-server 2000 or even higher version (like version 2005) in their machine. This is not so good for the normal personal users who don't want to have such a professional database management system.

2.1.1.2 MySQL

MySQL is a database management system that owned by MySQL AB (a Swedish company), and it's a multithread, multi-user SQL database Management system. The latest version is MySQL 5.0.

MySQL is now very popular, because it's an open source database management system that having fast performance and high reliability. And another important point is it's very easy to use.

In the homepage of MySQL, we can see:" Not only is MySQL the world's most popular open source database, it's also become the database of choice for a new generation of applications built on the LAMP stack (Linux, Apache, MySQL, PHP / Perl / Python.) MySQL runs on more than 20 platforms including Linux, Windows, OS/X, HP-UX, AIX, Netware, giving you the kind of flexibility that puts you in control. Whether you're new to database technology or an experienced developer or DBA, MySQL offers a comprehensive range of certified software, support, training and consulting to make you successful"³

MySQL is very good database system for the developer to use it. It's an open source database system and user can download the free version of it. Then it saves some money for the user.

2.1.1.3 Microsoft Access

Microsoft Access is a member of Microsoft Office, and it is known as a famous relational database management system. It can use data stored in Access/Jet, Microsoft SQL Server, Oracle, or any ODBC-compliant data container. And it is very normal that developer use Access & ODBC to store the application data. The latest version of Access is Microsoft Office Access 2007.

In the Microsoft home page, we can see:" Microsoft Office Access 2007 provides a powerful set of tools that help you to quickly start tracking, reporting, and sharing information. You can rapidly create attractive and functional tracking applications by customizing one of several predefined templates, converting an existing database, or creating a new database and you can do so without deep database knowledge. By using Office Access 2007, it is easy to adapt database applications and reports to changing business needs. The enhanced support for Microsoft Windows Share Point Services 3.0 in Office Access 2007 helps you to share, manage, audit, and back up your data."⁴

Then it's very good to use Access to do as the database storage resource. But Access 2007 is the new version and there are not so many user of it, then we can take Access 2003 to do the project.

³ Web reference from MySQL homepage

⁴ Web Reference from Microsoft Access Homepage

2.1.1.4 Microsoft Excel

Microsoft Excel can also be used as the database resource for the application. Excel can store the data in the tables, it is usually used for small size applications, and using ODBC or ADO to connect it. It might be a good idea to use the Excel table to store the data, but if the data record increased, this will get slow. It's not so fast and efficient as Access and SQL-server.

2.1.1.5 Oracle

Oracle Database is popular for the Enterprise Database, and it contains a collection of data managed by an Oracle database management system. But it's not so good to take it as the database for small or medium size projects. And it might take the company to pay money to use it. So it's not possible to use it as the database of the current project.

2.1.2 Database Connection Technique

There are several techniques we can use to connect to the database, we will discuss about how to do that using ADO, ODBC, or JDBC.

2.1.2.1 ADO

ADO (Microsoft **ActiveX Data Objects**) is a set of COM objects for accessing data source. "It provides a layer between programming languages and OLE DB (a means of accessing data stores, whether they be databases or otherwise, in a uniform manner), which allows a developer to write programs which access data, without knowing how the database is implemented."⁵

ADO consists of several top-level objects. The most important objects are Connection Objects, Recordset Objects, Command Objects, and Record Objects.

Connection Objects are used to represent the connection to the database.

Recordset Objects are used to represent set of database records.

Command Objects are used to represent a SQL command.

Record Objects are used to represent a set of data, typically from a source other than a database.

ADO techniques are frequently used with high-level language, like VBScript (in ASP environment) and VB. It is also very normal it be used in Delphi and C++ language.

Examples Showing How to Use ADO Technique:

This example is showing how to use ADO to get database from Microsoft Access database in VC++.

```
//Initialize the COM factory.  
//In the CWinApp::InitInstance()  
AfxOleInit();
```

```
//Create Connection object  
_ConnectionPtr m_pConnection;
```

⁵ Web reference from WIKI about ADO

```
HRESULT hr = m_pConnection.CreateInstance("ADODB.Connection");

//Open the database through the Connection object.
m_pConnection->Open("Provider=Microsoft.Jet.OLEDB.4.0;
Data Source=test.mdb", "", "",adModeUnknown);

// Create recordset and execute SQL command
_RecordsetPtr m_pRecordset;
m_pRecordset.CreateInstance("ADODB.Recordset");
m_pRecordset = m_pConnection->Execute ("SELECT * FROM instruments"
&RecordsAffected,adCmdText);
```

Advantages:

The developers do need to be aware of how database is implemented; they only need to care about the database connection. And they can use ADO to execute arbitrary SQL commands

Disadvantages:

The user should be familiar with SQL commands. And it introduces a dependency upon the type of database have been used.

2.1.2.2 ODBC

ODBC (Open **D**atabase **C**onnectivity) provides a standard software API method for using database management systems. It is designed to be independent of programming languages, database systems, and operating systems

In WIKI, we can see:” The ODBC specification offers a procedural API for using SQL queries to access data. An implementation of ODBC will contain one or more applications, a core ODBC library, and one or more "database drivers". The core library, independent of the applications and DBMS systems, acts as an "interpreter" between the applications and the database drivers, whereas the database drivers contain the DBMS-specific details. Thus a programmer can write applications that use standard types and features without concern for the specifics of each DBMS that the applications may encounter. Likewise, database driver implementors need only know how to attach to the core library. This makes ODBC modular.”⁶

Examples Showing How to Use ODBC Technique:

This example is showing how to use ODBC technique to get database from Microsoft Access database in VC++.

```
// Use CDatabase and CRecordset
CString sDSN= "ODBC; DRIVER=Microsoft Access Driver
(*.mdb); DSN=' ';DBQ=SwecoDB.mdb";
CDatabase m_database = new CDatabase;
if(!m_database->Open(NULL,FALSE,FALSE,sDSN,FALSE))
{
    return FALSE;
```

⁶ Web Reference from WIKI

```

        AfxMessageBox(_T("open database fail!"));
    }
    // m_ InstrRecordset is an instance of class CRecordset
    CString strSQL1="select * from Instruments";
    m_InstrRecordset=new CInstrRecordset(this->m_database);
    m_InstrRecordset->Open(CRecordset::dynaset,strSQL1);

```

Advantages:

The developers do not need to be worried about the programming language and the operation system. And this technique can be used in any kind of database resource.

2.1.2.3 JDBC

For the Java programming, the developer should use JDBC to connect the database resources. And now, JDBC has been part of the Java standard edition after JDK1.1, and the JDBC classes for Java are in the java.sql package.

In WIKI page, we can see: "JDBC is an API for the Java programming language that defines how a client may access a database. It provides methods for querying and updating data in a database. JDBC is oriented towards relational databases.

The Java Platform, Standard Edition includes the JDBC API together with an ODBC implementation of the API enabling connections to any relational database that supports ODBC. This driver is native code and not Java, and is closed source."⁷

2.2 Tools and Program Languages for doing the project

We can use several kinds of programming languages and IDE to build the project. In the below we will discuss about the IDE and programming languages.

2.2.1 Java & Eclipse

We can use Java as the programming language and Eclipse as the building project tool. Then we should have JDK 1.5 (or higher version) installed, and we can choose the latest version of Eclipse (like 3.2 or higher). Then we use java.swing package to do the GUI work, and for the connection technique, we can use JDBC. And the classes related to JDBC techniques are in the java.sql package.

Java is a high level programming language and easy to use, and Eclipse is a nice IDE to write a Java project. But Java is not as efficient as C++, and for doing the GUI things, the java.swing package is not that good as Microsoft Visual Studio tools. And for the developer's personal experience (the author), he is much more familiar with C++ programming. Then Java might not be used in this project.

2.2.2 C++ & Visual studio

We can use C++ as the programming language and Visual Studio as the IDE to

⁷ Web reference from WIKI

implement the project. C++ is a high level programming language as popular as Java, but it is more efficient than the Java. And Microsoft Visual Studio is a very powerful IDE; we can use visual studio and MFC to implement the GUI very quickly and efficiently by drawing and dragging. Visual Studio .Net has the latest version of 2005, then we can use English version for the doing the project.

For the personal experience of the developer and good performance of C++ & Visual studio programming, and because of the closed interaction of ODBC and Microsoft Access and SQL-server, this combination should be the best solution to do this project.

2.2.3 Visual Basic

Visual Basic is a powerful tool to write this project. VB is very simple language and very easy to use, it's usually used for the small size project with SQL-server or Access database. It's very easy to do the GUI things and connect to the database by the ODBC techniques. Then VB is also a good way to do this project. But actually VB is not so efficient as Java or C++, and then if the computation increased, it will not be so good to use it.

2.3 Candidate System Solutions

By doing researches about the problems, we have two candidate system solutions to solve the problem. We can use client-server system to realize the requirements, or we can use the other way, building a server machine system. In the below the two candidate solutions will be discussed.

2.3.1 Client- Sever System

We can use the Client-server pattern to build the system. In the company's needing, there may be several computers work on one database. Then there may be one solution to do this system: a client side application and a server side application. The client side is mostly about the GUI work and accepting the system users' input, and in the background of the client, the data is get from the GUI and then send to the server side application through the system connection part (need the local network working).

Then the server side gets the user's input or operate request, and the do these operations to the database.

This might a good way to do this project, but it will waste a lot of work to build client side and server side. And both the client side and the server side should have some part responsible for the connection, this will take some time to do it. If we do in this way, the application in the client side actually can not do anything without the local network.

2.3.2 System running in Server Machine

This is a good way to do the project. The company have server machine running and some other application can access the local server machine by local network. Then the best way to do this project is put the system in a sharing disk place of the server machine, give the privileges for the different PCs to access and operation the database on the server machine. Then the whole system is most simple and more efficient.

It is a very good way for doing the project, but in this way, we should pay attention to the privileges of the guest-machine user from the other PCs, the user from other PC can both read and write in this system directory of the local server machine.

2.4 Method and Planning of the project

Here we will discuss about the project process we want to follow. That is UPEDU process. In the below we talk about some basic idea of the UPEDU process and also make plan of the project.

2.4.1 Method of doing the project

The project will basically follow the UPEDU process (Unified Process for Education), and make an efficient work to building the system that feed the company's needing. In the process, there may be some changes adapt the current situation.

2.4.2 Planning of the project

Following the UPEDU process the project will be divided into 4 phases.

Inception phase:

The most important goal of the inception phase is to achieve concurrence among all stakeholders on the lifecycle objectives for the project.

Then the work of this phase is to establish the project's software scope and boundary conditions, do software requirement elicitation, and make plan for the whole project.

Elaboration phase:

From the UPEDU website we can see: "The goal of the elaboration phase is to baseline the architecture of the system to provide a stable basis for the bulk of the design and implementation effort in the construction phase"⁸

Then in this phase the much more attention should be paid on the architecture, and software analysis and design. The software requirement validation should also be done. In the end of this phase, most of the design work should be done, and then the project will go into the construction phase to implement the design.

Construction phase:

In the UPEDU we can see: "The goal of the construction phase is on clarifying the remaining requirements and completing the development of the system based upon the base-lined architecture."⁹

In this phase, most of the effort should be paid on the implementation. At the end of this phase, a stable release system should be done, and then this system can be given to the transition phase.

Transition phase:

⁸ Web Reference from UPEDU

⁹ Web Reference from UPEDU

In the UPEDU definitions, we can see:” The focus of the Transition Phase is to ensure that software is available for its end users. The Transition Phase can span several iterations, and includes testing the product in preparation for release, and making minor adjustments based on user feedback.”¹⁰

Then in this phase, we focus on testing the current systems and do some bug-fixing work.

According to the time-split suggestion from the UPEDU, and some adaptations for this project’s need, the time schedule is made below.

Phase	Time
Inception phase	2007-4-13 to 2007-4-21
Elaboration phase	2007-4-22 to 2007-5-3
Construction phase	2007-5-4 to 2007 -5-26
Transition phase	2007-5-26 to 2007-6-3

Table 2.1 Time Schedule of the Project

This table will show the schedule of the project. The thesis project started in the beginning of April 2007.

Then the project should be finished in the beginning of the June, 2007.

¹⁰ Web Reference from UPEDU

3. Solution of the Project

This chapter we will mainly discuss about the solution of the project. We talk about the software requirements, system designs and implementations. The design of the database is also quiet important, it also have been discussed below.

3.1 Requirement

Today an instrument database exists in Microsoft Access. The database is built a while ago and there a few people who have worked with it. The problem of today's database is that there is no built in function for:

Set Current Status: Where is the instrument at this moment (physical e.g. in stock, leased, sold etc.)

Traceable history of an instrument: Each status that an instrument is passing through its lifetime, before it's deleted is important to make traceable. It should for example be easy to make a list of how many times a instrument has been repaired or leased.

Booking function: Each instrument that is in stock should easily be booked by the system. It's only the leasing instruments that should be possible to make a reservation for.

Alarm function: Some of the controls for each instrument are done periodically but the problem is to know when things should be done. Some flag that notifies the service men, which instruments should be tested in the near future are desirable.

The customer needs the application to be a Service-side application. That means this application will be run in a local network. And the database in the server side can be visited by different user using different computer.

Then, in the customer's mind, the system should run like:

Server with Database: This Server machine is connected to the local network, and have database in his disk.

Printer: Used to print out the report created by the system.

PC Client: The employees of the SWECO Company are working in the PC machines. They may operate on the Database through the local network. And print the report by the Printers which are connected to the network.

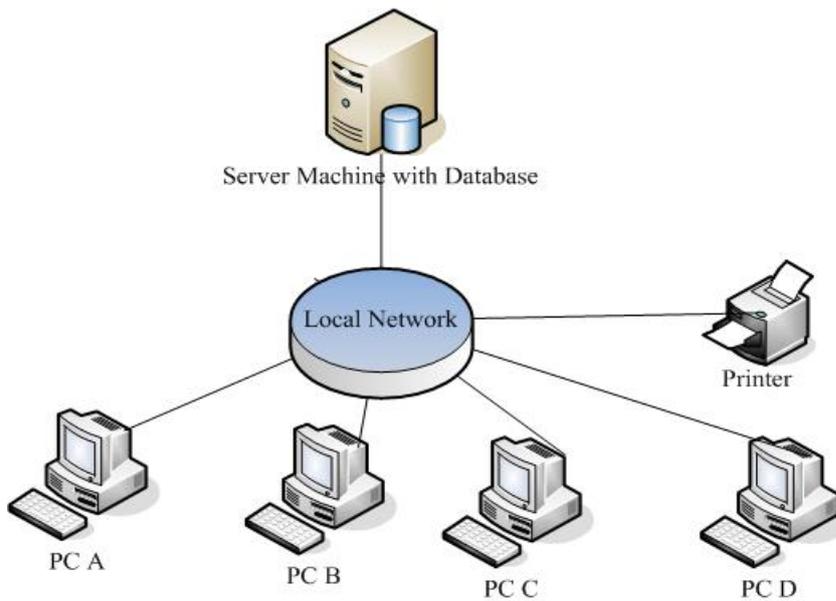


Figure 3.1 Network architecture in the company

The figure shows you the network architecture in the company. There are one or several server machines that connected to the local network. The other personal computers and some printer machines are also in the local network.

3.1.1 Requirement Elicitation

During the customer meeting with the SWECO Company, Much Researches have been done. And there will be some use-cases in the project.

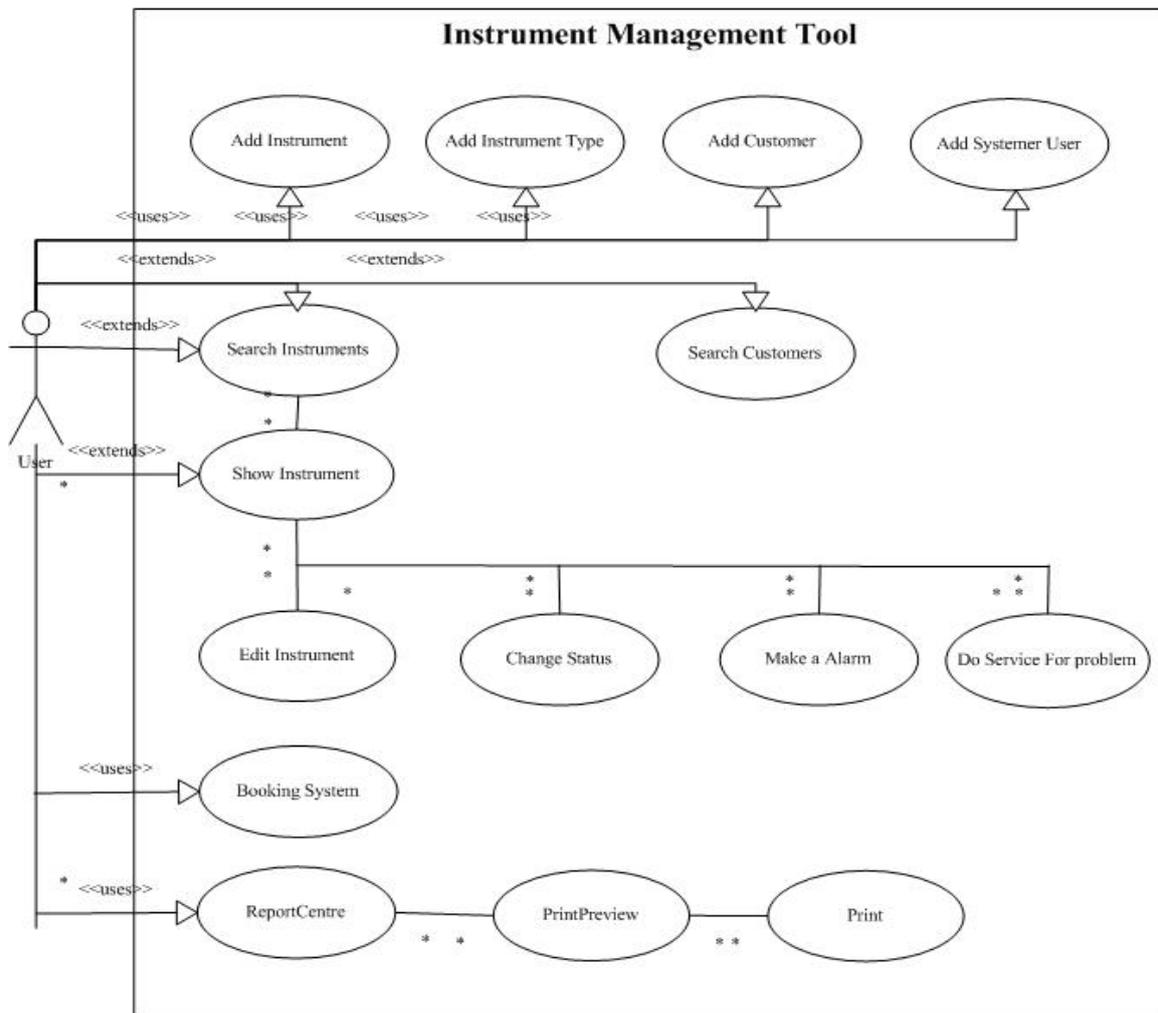


Figure 3.2 Use case Diagram of the project

The figure shows you the Use cases in the project. There will be 16 use cases in the system.

Use Cases:

UC1: Add New Instrument

UC2: Add new Instrument type.

UC3: Add customer

UC4: Add System User.

UC5: Search instruments

UC6: Search Customers

UC7: Show Instrument information

UC8: Edit Instrument information

UC9: Change the instrument current status.

UC10: Make an Alarm for certain service.

UC11: Do a Service to solve some problem

UC12: Show customer information

UC13: Edit customer information

UC14: Book the instrument for leasing

UC15: Make a report

UC16: Print report preview.

3.1.2 Software Requirement Specification

Requirement prioritizing:

The requirements that are presented on the following pages are prioritized by 3 different levels and will be handled from this priority. The levels are the following:

- **Essential**
- **Preferred**
- **Optional**

System Functions:

The Requirements will be listed below.

- **SR1: Install the software system.**

Priority: **Essential**.

The System should be able to be installed in the Server machine side and the client machine side. And the system should be able to run in Windows XP, Swedish version, English version, Chinese version, and any other system with in different language version should all workable.

- **SR2: Uninstall the software system.**

Priority: **Essential**

The whole system can be removed from the server side and the client side. But the database will be still there, then it can keep the records of the system have done before.

- **SR3: Add an instrument(Register an instrument)**

Priority: **Essential**

The system user is able to add a new instrument into the database, and this instrument is bought from some other distributor. This requirement is very important, since some of the other requirements like showing instrument information and edit instrument information must rely on it. It's very important to create an instance of instrument in the database for the future's management.

Then it is taken as essential requirement.

- **SR4: Add an instrument type**

Priority: **Preferred**

This requirement is taken as preferred one, because right now, the Company hasn't bought new type of instrument in the stock. But somehow it might be used in the future, when the company wants to buy new instrument type.

- **SR5: Add a customer**

Priority: **Essential**

This requirement is taken as Essential one. It's very important for the company to keep the information of the customers. And the customer information should containing contact person, phone numbers, and some short description about the customer company's background.

- **SR6: Add system user**

Priority: **Preferred**

This requirement is taken as preferred one. There will be 4 or 5 people in the company who are acting as system user for managing the instruments. Then it's possible in the latter time, the user number will increase, and then the system can register a new system user. And adding a system user to the system is mostly for keeping the records about which person has done some services. The registered users don't need to write down their full names, it will save much time.

- **SR7: Search instruments**

Priority: **Essential**

The search instruments requirement is an essential one of the system. The system user can search the instruments through the ID, instrument type, Distributor name, sales person or current status. There may be some possible status like: "In stock", "Leased", "sold", "deleted", "Problem" or "Repair". The "In stock" status can be the instrument that is still in the company's stock, or it is a new one. The searching result should be showed after the user chose some specified condition.

- **SR8: Search Customers**

Priority: **Preferred**

This requirement is taken as preferred one. The system user of the company can do quickly search about their customers. Then the customer can be show after the user search through the customer name, contact person name and some other useful information.

- **SR9: Show instrument information**

Priority: **Preferred**

The instrument information can be showed after system users choose one instrument from the search instrument results.

- **SR10: Edit instrument information**

Priority: **Preferred**

This requirement is taken as a preferred one. The user can edit the instrument information when the user reading the instrument information.

- **SR11: Change the instrument current status**

Priority: **Essential**

This requirement is taken as an essential one of the system. The system user can change the current status of an instrument. When the user do some change about current status, the change logs should be stored in the database, and the changed status and change log should be show immediately in some way.

- **SR12: Make an alarm for certain service**

Priority: **Essential**

This requirement is taken as an essential requirement of the system. The system user can arrange some service for some instruments before hand. Then the alarm for doing the service should be showed when the planned date is coming nearby.

- **SR13: Do a service to solve certain problem**

Priority: **Essential**

This requirement is taken as an essential requirement of the system. When the instrument is in the status of "Problem", the system of the user should be able to do a service to solve the problem for the instrument. Then the system keep the records about what the problem is and how much money or effect have been used to solve the

problem.

- **SR14:Show customer information**

Priority: **Preferred**

This requirement is taken as a preferred requirement of the system. The user can see from the system about the detailed customer information after he does a searching and choose one of the results.

- **SR15:Edit customer information**

Priority: **Preferred**

This requirement is taken as a preferred requirement of the system. The user can see from the system about the detailed customer information after he does a searching and choose one of the results.

- **SR16:Book instrument for leasing**

Priority: **Optional**

This requirement is taken as an optional requirement of the system. The system user can do some pre-booking work for the instruments which will be leased. Then the company knows better about how to arrange the coming months' work.

- **SR17:Make a report for all tables**

Priority: **Essential**

This requirement is taken as an essential requirement of the system. The system should create report for all the tables of the database.

- **SR18:Show Print preview**

Priority: **Preferred**

This requirement is taken as a preferred requirement of the system. It will be very good if the system have the preview function, and the user can see if the report is ok or not, the user can choose to cancel or print the report.

- **SR19:Print the reports**

Priority: **Essential**

This requirement is taken as an essential requirement of the system. It's very important that the user can print out the report through the print machine.

3.2 Select Method and Technique

After research about different kinds of database management systems, database connection techniques and programming language and tools, we should choose the suitable method and techniques for this project. Considered about the company's needing about the application, we choose the local server machine to run the system, and then the users using different PCs can use this application through the local network.

The Microsoft Access database is chose because of the server machine don't have other database management system except Microsoft Access. Then it won't need the company to pay money to buy new software and install new database system just for this project. This solution is for the "cost-down" and for the easily setting up the instrument register system. The ODBC techniques are used because it's easier for adapting different machines.

And Visual studio 2005 and C++ will be used to build the project and do the programming work. Because MFC is efficient for doing the GUI and it has closed interaction with the ODBC and Microsoft Access database. Also, for a personal reason,

the developer is familiar with the C++ programming.

Then, for doing this project, these Methods and Database techniques are used:

- System running in the local Server Machine (Window Xp environment)
- Use Microsoft Access as the database resource.
- Use ODBC to connect the database resource.
- Use C++ programming.
- Use Microsoft Visual Studio 2005 to build the project

Then the whole system will looks like the figure down.

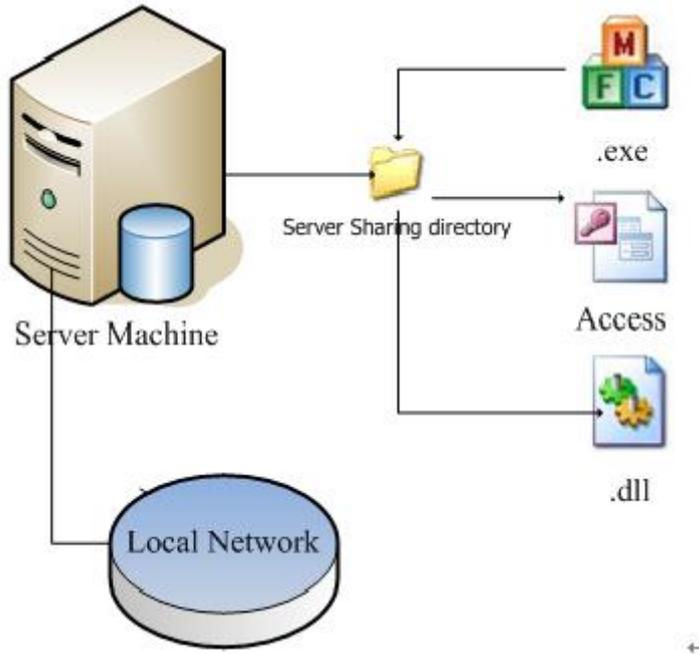


Figure 3.3 Architecture of the system

The figure shows you the architecture of the system. We will have the server application running in the server machine. The application files are in a sharing file of server machine. The other clients can run the application through the local network.

3.3Architecture

The project will use Microsoft Access as the database and ODBC to connect the database. The graphic user interface (GUI) will be written in MFC and using Microsoft visual studio 8.0. The whole system will put in a server machine, and other computers can run the application through the local network.

The high level architecture is showing down.

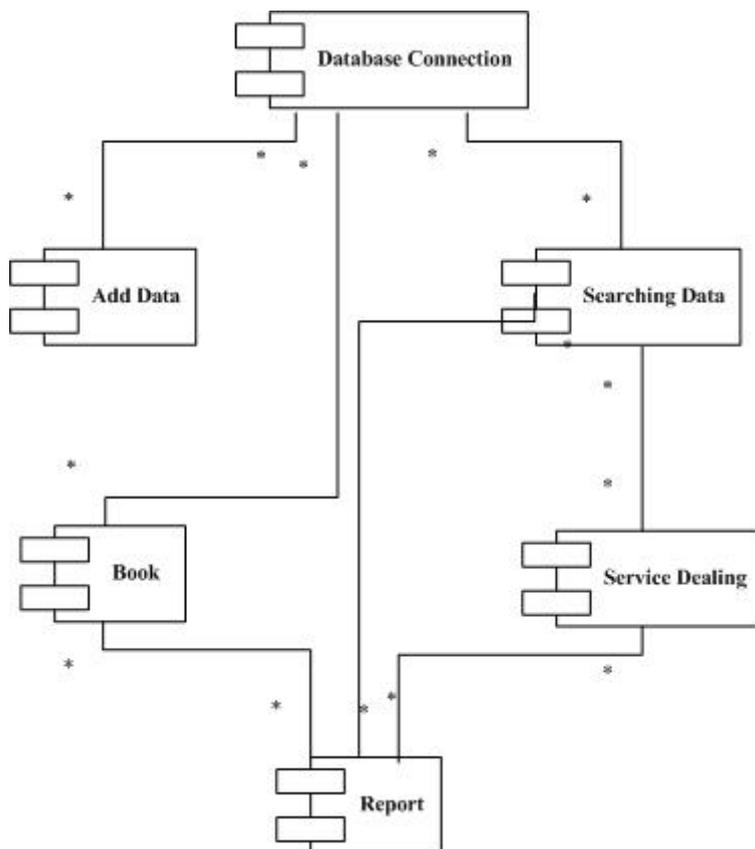


Figure 3.4 System design diagram

This figure shows the high level system design of the project. The system is divided into six packages. Each package is responsible for some work.

Table 3.1 Detailed Description of System Design Diagram

This table shows you the detailed description of each part of the system design diagram.

Name	Description	Responsibilities
Database Connection	Database Connection Package	This part is responsible for connection the database with the system. All The classes that extending CRecordset class will be contained in this package.
Add Data	Add Data Package	This part is for the adding new data in the database. It Containing adding new instruments, customers, system users and instrument types.
Search Data	Search Data Package	This package is for user to search the instruments and customers through some key words. The result showing also be contained in this package.
Service Dealing	Service Dealing Package:	This package is responsible for doing service or planning alarm for a choosing instrument. The user can edit the instrument information here, change the instrument status, and do a service

		to solve certain problems.
Booking	Booking Package:	This package is for the system user to planning the leasing work.
Report	Report Package	This package is for the system user to create report for the database. And the user can choose to see the print preview then print it out in the printers which are connected in the local network.

3.4 Design

Here will discuss about the design of the whole system. The high level analysis class will be talked first, and then we analyze the designing one part after another. You will see the main structure of the system solutions here.

3.4.1 Analysis Class

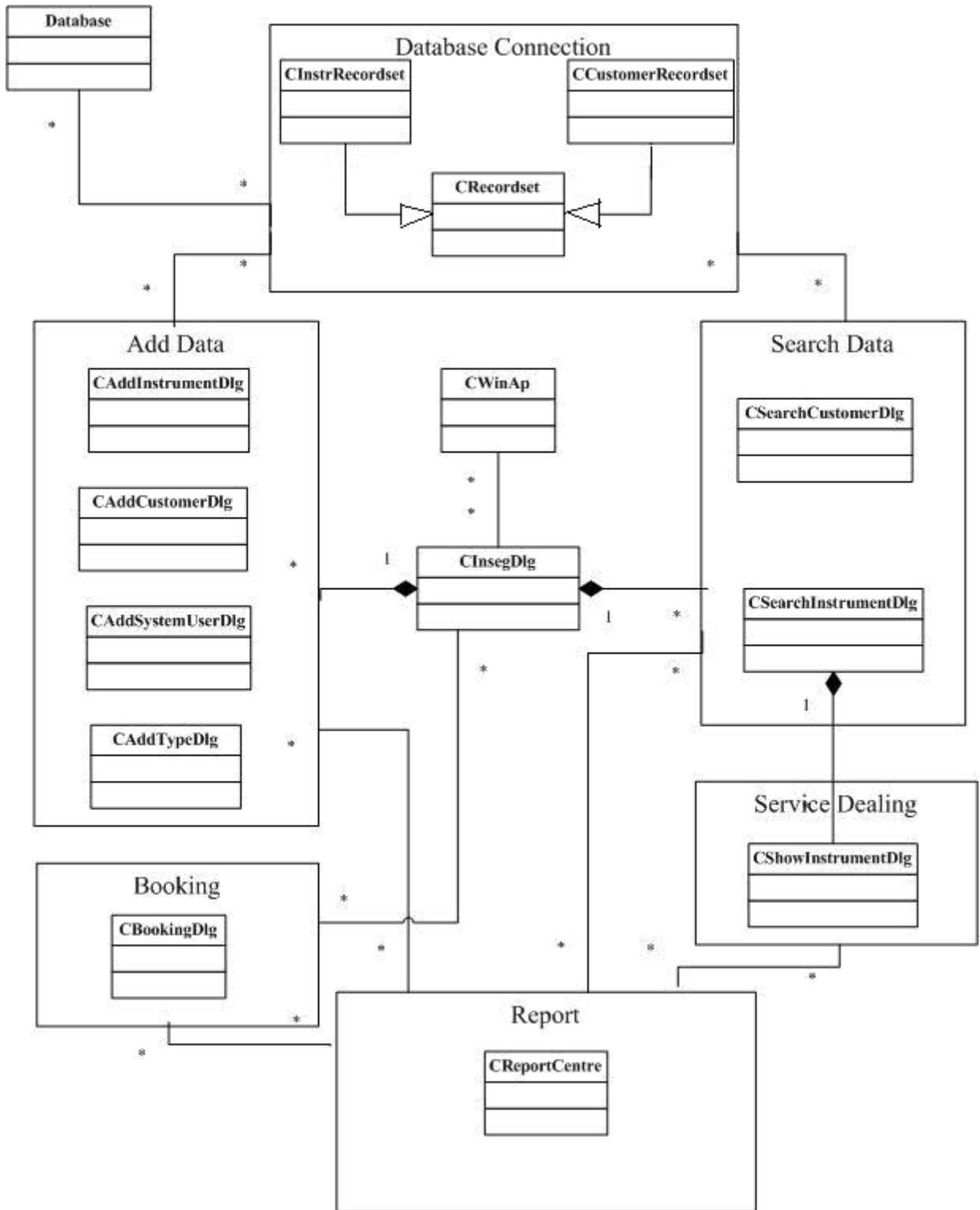


Figure 3.5 Analysis Classes Diagram

This figure shows you the analysis classes of the project. Not each classes are showing here, there all important classes below.

Table 3.2 Detailed Description of Analysis Classes Diagram

Name	Description	Responsibilities
------	-------------	------------------

CWinApp	The application instance for this project will be CInsRegApp, it is derived from CWinApp	This is a system initializing class. And the start point of running the application.
CInsRegDlg	Dialog class for showing the main Menus.	This dialog is for showing main Menus.
CRecordset	CRecordset from MFC.	Some other class will derived this one.
CAddInstrumentDlg	Derived from CDialog class	This class is for adding instrument to the database
CAddCustomerDlg	Derived from CDialog class	This class is for adding customer to the database
CAddSystemUserDlg	Derived from CDialog class	This class is for adding system user to the database.
CAddTypeDlg	Derived from CDialog class	This class is for adding instrument type to the database
CSearchInstrumentDlg	Derived from CDialog Class and do searching instrument	This class is for searching instrument
CSearchCustomer	Derived from CDialog Class and do searching customer	This class is for searching customer and also responsible for showing customer information.
CShowInstrumentDlg	Derived from CDialog Class and do showing instrument information.	This class is for showing instrument information and doing service for solve problems.

This table shows you the detailed description of Analysis classes Diagram.

3.4.2 Database Connection

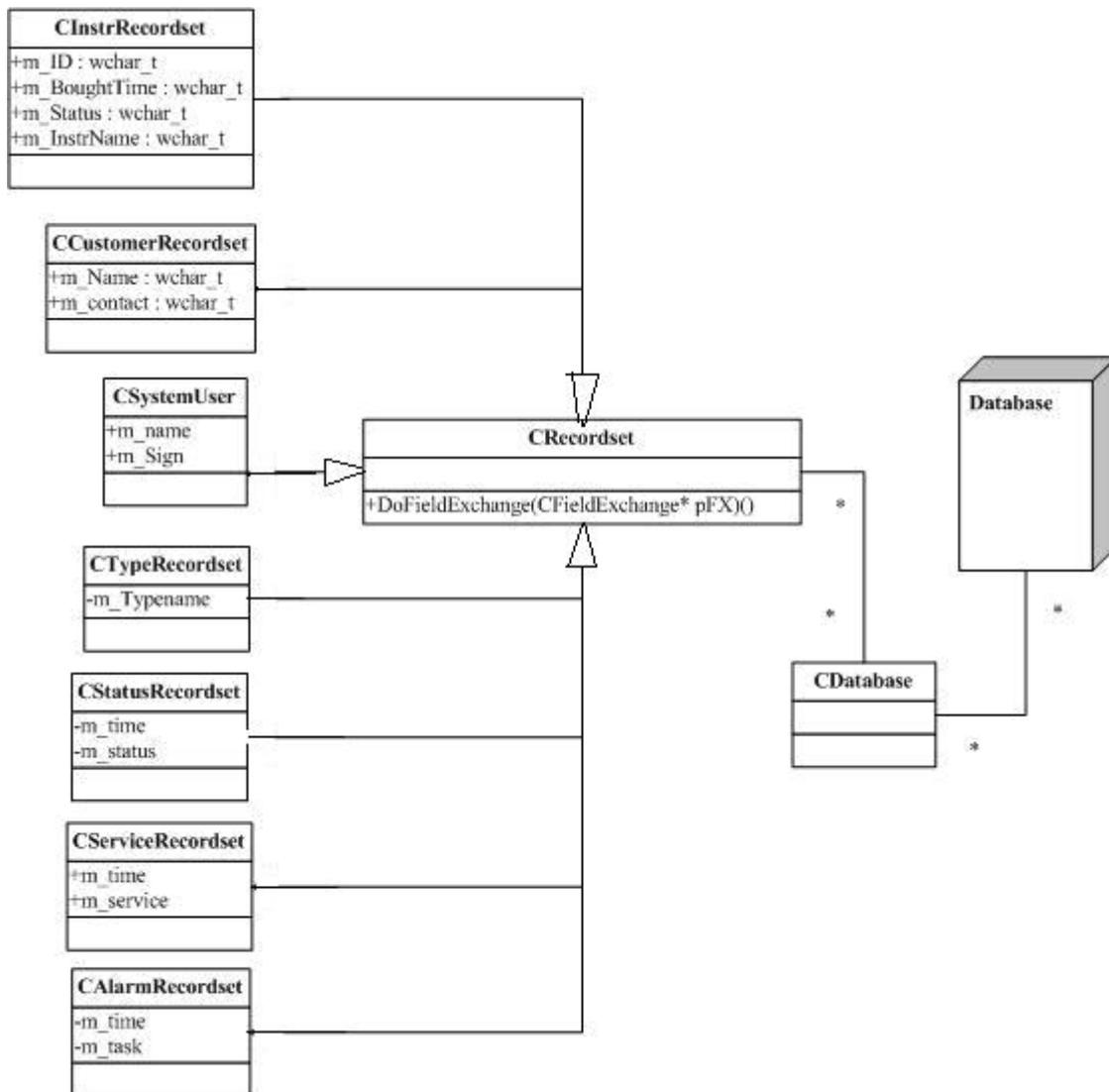


Figure 3.6 Class Diagram of Database Connection Package

This figure shows you the classes that are related to the Database Connection. Most of the classes are inherited from CRecordset.

Table 3.3 Detailed Description of Database Connection Package

This table shows you the detailed description of Database Connection Package.

Name	Description	Responsibilities
CRecordset	CRecordset from MFC.	Some other class will derive this one.
CInstrRecordset	Derived from CRecordset	This one will be used to operate instrument table in the database
CSystemUserRecordset	Derived from CRecordset	This one will be used to operate System user table in the database
CInstrTypeRecordset	Derived from CRecordset	This one will be used to operate instrument type table in the database

CStatusRecordset	Derived from CRecordset	This one will be used to operate instrument status change log table in the database
CServiceRecordset	Derived from CRecordset	This one will be used to operate instrument service table in the database
CAlarmRecordset	Derived from CRecordset	This one will be used to operate alarm table for doing some service in the database

3.4.3 Add Data

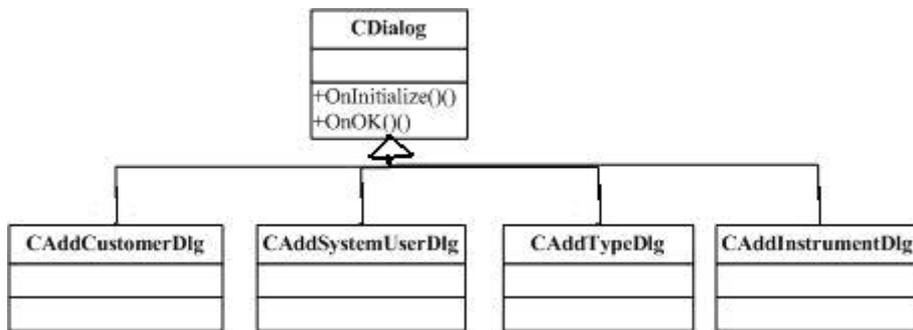


Figure 3.7 Class Diagram of Add Data Package

This figure shows you the classes that responsible for adding data to the system.

Table 3.4 Detailed Description of Add Data Package

This table shows you the detailed description of adding Data diagram.

Name	Description	Responsibilities
CAddCustomerDlg	Derive from CDIALOG of MFC.	This dialog is used for adding customer into the database.
CAddSystemUserDlg	Derive from CDIALOG of MFC	This dialog is used for adding system user into the database
CAddInstrumentTypeDlg	Derive from CDIALOG of MFC	This dialog is used for adding new instrument type into database
CAddInstrumentDlg	Derive from CDIALOG of MFC	This dialog is used for adding new instrument into database.

3.4.4 Search Data

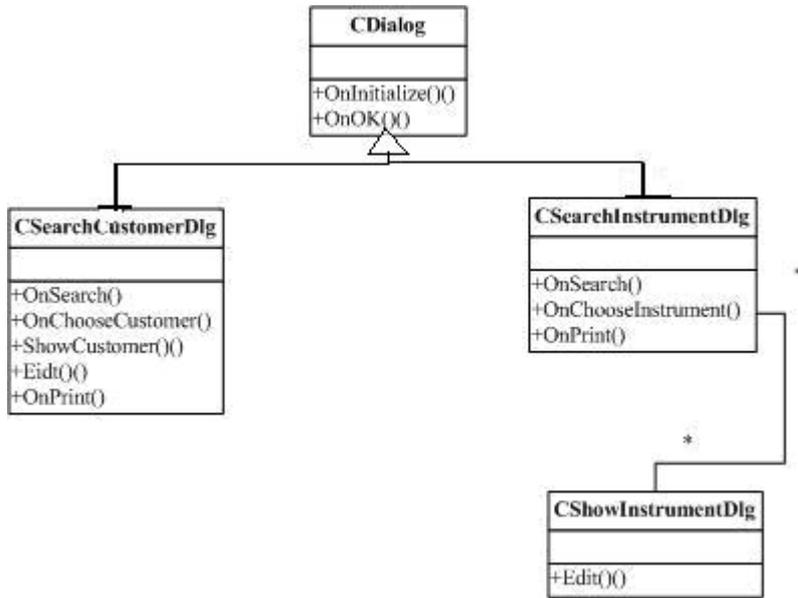


Figure 3.8 Class Diagram of Search Data Package

This figure shows you the classes used for searching data in the system.

Table 3.5 Description of Search Data Package

Name	Description	Responsibilities
CSearchCustomerDlg	CDialog from MFC.	Search customer here. And user can edit customer and print out customer.
CSearchInstrumentDlg	CDialog from MFC.	Search instrument here. And user can choose an instrument and print out instrument table.
CShowInstrumentDlg	CDialog from MFC.	When user chooses an instrument, it will show the instrument by this dialog.

3.4.5 Service Dealing

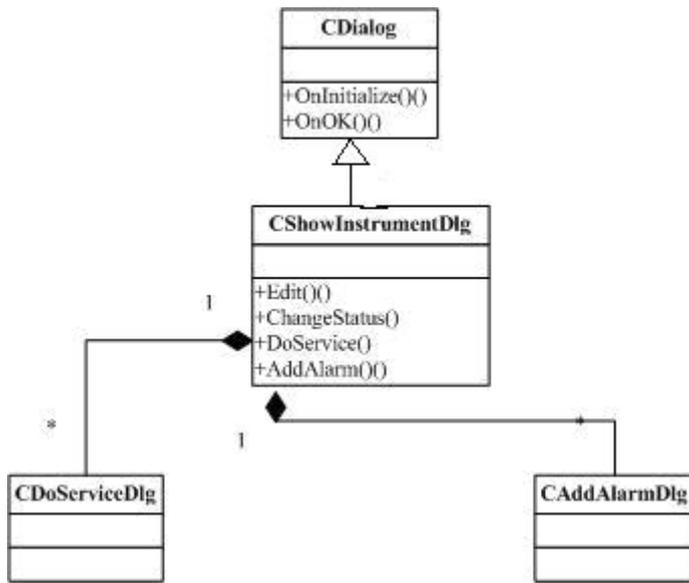


Figure 3.9 Class Diagram of Service Dealing Package

This figure shows the classes that are responsible for service dealing.

Table 3.6 Description of Service Dealing Package

Name	Description	Responsibilities
CShowInstrumentDlg	CDialog from MFC.	Instrument information will be shown here. And user can also change the status, doing a service or plan a service alarm here
CDoServiceDlg	CDialog from MFC.	User can do a service to solve problems here.
CAddAlarmDlg	CDialog from MFC.	User can add an alarm for doing some service.

3.4.6 Booking

Description:

The user can choose to do booking things here. This is only internal booking, for the company to know which instruments are planning to be leased.

3.4.7 Report

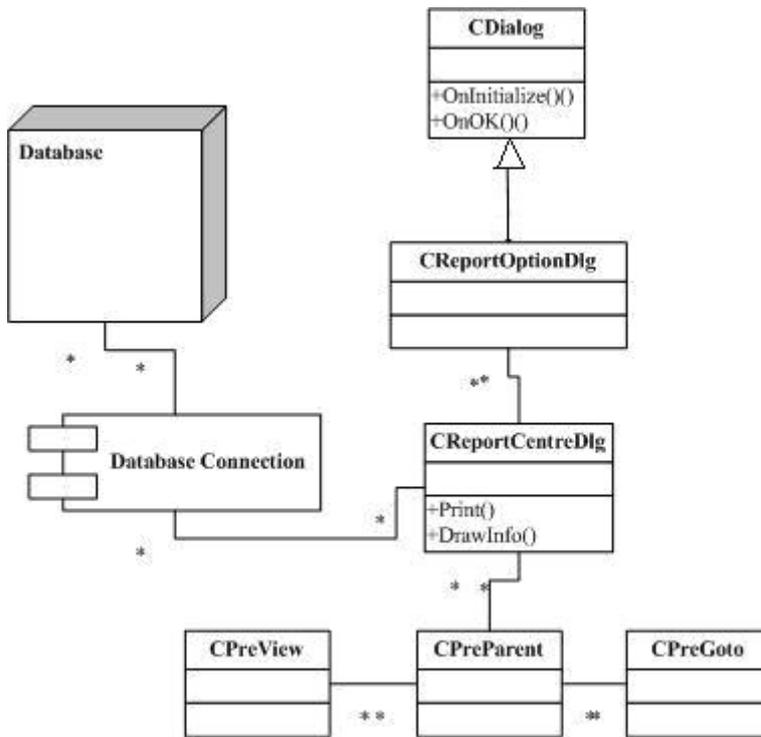


Figure 3.10 Class Diagram of Report Creating Package

This figure shows you the classes that are responsible for creating reports. In the CReportCentreDlg, we can open a window to preview the report, if we choose to print out; it will get data from the database and then print the report in the printer that connected to the local networks.

Table 3.7 Detailed Description of Report Creating Package

Name	Description	Responsibilities
CReportOptionDlg	CDialog from MFC.	The User can choose which kind of report he wants to create
CReportCentreDlg	CDialog from MFC.	The user can see the report and choose to print it out.

3.5 Access Database Design

In the Project, We use Microsoft Access as the database resource. This choice comes from the negotiation between the developer and company worker. They only have Microsoft Access installed in the company's machines, and don't want to pay more money to buy other database management software. Then the Microsoft Access Database is the best choice for doing this project.

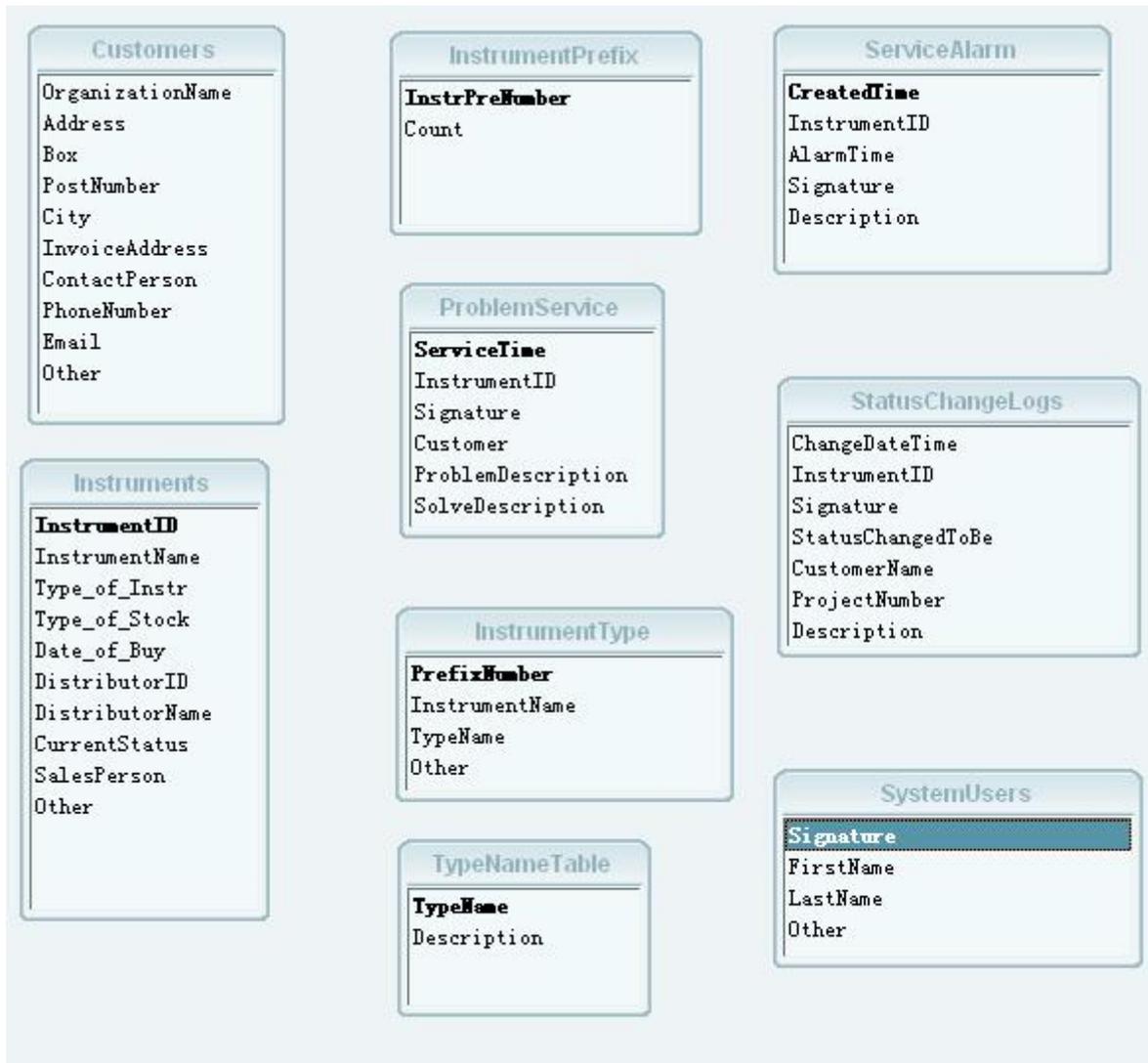


Figure 3.11 Database Connection Diagram

This figure shows you the Database design of the system. The bold word is the primary key of each table. And in the customers table, we have OrganizationName and Contactperson as combined primary key.

As you can see from the figure, there will be nine tables in the access database. “Instruments” table will store the basic information about the instrument in the company. It will have a unique number called “Instrument ID” for each instrument. In the Customers table, the “OrganizationName” and “ContactPerson” will act as the combined primary key.

3.6 Implementation

The project is built in Visual studio .net 2005, and it’s a dialog based project. The graphic user interface is written by using MFC. And I will introduce some implementation detail in the below.

3.6.1 Add instrument to database

In the implementation, ODBC techniques are used to connect the Microsoft Access

database.

First we should have an instance of CDatabase. And use the CRecordset class to access the database.

After it invoked CRecordset::Update() function, the new Recordset will be added into the database. If it returns true means update successful, or it returns false and meaning update failed.

And in the Adding new instrument requirement, the company user would like the application create a unique number automatically through the combination of instrument name, instrument type and the stock type of the instrument (leasing or selling type).Then the application should first get the prefix number of the instrument type from the database, then the application should check the database again and compute a available unique number for the a new instrument.

3.6.2 Search

For doing the searching work, we should first have SQL search string. The SQL searching string can be got by the user's inputting or by easily choosing in combobox. Then in the project, put the SQL searching string to the CRecordset::m_strFilter , and then do the Requery().Then the new Recordset containing searching results will be got from the database.

Code clipping:

```
this->m_InstrRecordset->m_strFilter=schStr;
if(m_InstrRecordset->CanRestart())
{
    m_InstrRecordset->Requery();
}else{
    ::AfxMessageBox(_T("Cannot requery"));
}
```

The result will be in the m_InstrRecordset(an instance of CInstrRecordset). If it can not execute Requery(), it will return the error.

3.6.3 Print

In the design, we used 3 classes responsible for the printing preview and print work. They are CPreParent class , CPreView class, CPreGoto class. The CPreParent is used for the printing preview showing screen. The CPreView is responsible for preview and printing environment initialize. The CPreGoto is used for the use to go some certain pages he wants.

When we click the print-preview button, the system will run into this code:

```
////////////////////////////////////
void CSearchInstrumentDlg::OnBnClickedButtonPrintpre()
{
    // TODO: Add your control notification handler code here
    if(this->m_InstrList.GetItemCount()<= 0)        return;
    PRNINFO PrnInfo = {0};
    PrnInfo.hListView = m_InstrList.m_hWnd;
```

```

PrnInfo.hWnd = this->m_hWnd;
PrnInfo.IsPrint = FALSE;
PrnInfo.nCurPage = 1;
PrnInfo.nMaxLine = m_InstrList.GetItemCount();
CPreParent DlgPreView;
DlgPreView.SetCallBackFun(DrawInfo, PrnInfo);
DlgPreView.DoModal();
}

```

And in the CPreView class, we use a CPrintDialog class (from MFC) to get the printer that are connected to the local network. User can choose the printer , then the code below will create an HDC for the printer device.

```

////////////////////////////////////
CPrintDialog printDlg(FALSE);
//Use CPrintDialog to create printer environment
if (printDlg.DoModal() == IDCANCEL) return ;
HDC hdcPrint=printDlg.CreatePrinterDC();

```

Then in the DrawInfo() method, we do the preview drawing and printing drawing by using CPen and CFont classes.

4. Evaluation and Result

This is the evaluation and result part of the thesis paper. Here we discuss the problems I met in the project and how I solved it. In the evaluation part, I made a table to show how the system solutions fulfill the customer's requirements, and then it is obviously how this project has been finished. We also discuss something about how to delivery the project to the company.

4.1 Problems in the project

Several problems are occurred during the project. The problems come from the communications, the different version of operation systems, and the different system setting and ODBC versions. The application runs perfectly in several personal computer, but when it running in the Swedish operating system in the company's machine, it crashed and give an error saying:"Ogiltig kopplingssekvens som inte kan hanteras av operativsystemet ", in English it means:" Invalid connection sequence can not be handled by the operating system".

In the first, I thought it should be the difference between English, Chinese and Swedish operation system that have caused this problem. Then I used a Swedish WinXP machine to rebuild the project, but found out be to the same result.

Then I tried the ADO techniques to connect the Access database, but still, the same error came out.

Finally, the result that causes this problem is come from the old version Microsoft Access that installed in the company's machine. The company's machine have Access 97 , and it considered the SQL search string to be invalid. Then we had to convert the Access 97 file into the Access 2003 file, then the application works fine.

4.2 Evaluation

The requirement classification table will be provided for evaluate the system and showing how well the system fulfills the company' needing. The requirements have been given a color encoding to indicate their priority as seen from the stakeholder's perspective. Green requirements are essential, blue requirements are preferred and red requirements are optional.

System Functionality	Type	Status
SR1: Install the software system.	Essential	Fully
SR2: Uninstall the software system.	Essential	Fully
SR3: Add an instrument(Register an instrument)	Essential	Fully
SR4: Add an instrument type	Preferred	Fully

SR5: Add a customer	Essential	Fully
SR6: Add system user	Preferred	Fully
SR7: Search instruments	Essential	Fully
SR8: Search Customers	Preferred	Fully
SR9: Show instrument information	Preferred	Fully
SR10: Edit instrument information	Preferred	Fully
SR11:Change the instrument current status	Essential	Fully
SR12:Make an alarm for certain service	Essential	Fully
SR13:Do a service to solve certain problem	Essential	Fully
SR14:Show customer information	Preferred	Fully
SR15:Edit customer information	Preferred	Fully
SR16:Book instrument for leasing	Optional	Not implemented
SR17:Make a report for all tables	Essential	Partially
SR18:Show Print preview	Preferred	Partially
SR19:Print the reports	Essential	Partially

Table 4.1 System Function Realization

This table shows you about if the software product fulfill the company's requirements. The green color means essential requirements, blue means preferred ones, and red means optional ones.

4.3 Result and Delivery

For giving the product to the company, the installation and operation manual will be provided. The system will be given to the company by CD, and the developer will go the company to set up the system for the company using. The developer will also be responsible for the maintenance work of the system.

5. Discussion & Conclusion

This chapter is about the conclusion of the project. It discusses about what I have done and what I have learned during the project. The conclusion and discussion part gives you an over view of the whole project. It is mainly about the achievements. I also put some paragraphs to discuss about the future work of the project, and then it is a good way to make it better.

5.1 Conclusion

Hard working comes to the good results. The most part of the project is done right now. The thesis paper and theory researches are also done. Now I am starting to thinking about the achievement of the thesis topic and project.

I am very happy I got over the hard time of communication in English in the beginning. And this is a personal project for my bachelor thesis; I was doing all the things myself. I met many problems when I was doing my researches and implement the project, but my confident and spirit of never giving up help me to get over the hard time. It is very good experience for my future's software engineer career.

During the thesis and project process, I did a lot of research about the database technique and how to build a good looking GUI. I used VC++ to draw color List controller and buttons and did much work about improving the software appearance.

I used C++ programming in Visual Studio, and used Access as the database resource. This is found out to be a nice solution for the project. The company needs a stable system for managing their instruments, and the local network solution makes the multi-user to operate on the database. The process during the project, we used UPEDU process and did some adaptation for the current project. Then, nice communication and good working environment contributes a lot for doing the project perfectly.

"The main goal of this thesis paper is how to use Microsoft Access and VC++ to do realize a software product."

In the beginning of the project, I set my goals to discuss how to use Access and VC++ to build stable software for practical using. In the process, I did researches to prove my solution is the best one to solve the company's problem.

"The other goal is learning to use RUP process to solve the software problem."

When doing the project, we followed the UPEDU process, and adapted and simplified the process to the project's condition. This is a real project; it gives me quite a lot of experience and helps me understand the UPEDU process.

So choosing the right techniques, using the good process and setting up a good communication is very important for building a software system for the enterprise using.

5.2 Future work

The future work of the project will mostly be some maintenance work. And for making the system to be more efficient and more stable, the following work might be done.

- **Improving GUI**

The Graphic User Interface is very important and more time should be spent on improving the GUI. Nice GUI layout and design will let user work more easily and feel comfortable.

- **Complete the booking function**

The booking function is not fully implemented right now, and there is only an architecture how it will look like. Then the booking function for showing the planned leasing-work can be designed and implemented soon.

6. References

1. Web Reference from the Microsoft SQL server Homepage,
<http://www.microsoft.com/sql/prodinfo/overview/what-is-sql-server.mspx>.
Last visited time on 2007-6-24
2. Web Reference from the Microsoft SQL server Homepage,
<http://www.microsoft.com/sql/prodinfo/overview/what-is-sql-server.mspx>
Last visited time on 2007-6-28
3. Web Reference of MySQL home page;
<http://www.mysql.com/why-mysql/> Last visited time on 2007-7-2
4. Web Reference from Microsoft Home page;
<http://office.microsoft.com/en-us/access/HA100241851033.aspx>
Last visited time on 2007-7-2
5. Web Reference from UPEDU
<http://www.upedu.org/upedu/index.asp?TruY=860887062919126>
Last visited time on 2007-8-3
6. Web Reference from UPEDU
<http://www.upedu.org/upedu/index.asp?TruY=860887062919126>
Last visited time on 2007-8-2
7. Web Reference from UPEDU
<http://www.upedu.org/upedu/index.asp?TruY=860887062919126>
Last visited time on 2007-8-12
8. Referenced in WIKI; http://en.wikipedia.org/wiki/ActiveX_Data_Objects
Last visited time on 2007-6-22
9. Reference from WIKI page, <http://en.wikipedia.org/wiki/ODBC>
Last visited time on 2007-7-5
10. Web Reference from WIKI page, <http://en.wikipedia.org/wiki/JDBC>
Last visited time on 2007-6-26

2. Source Code of how to Adding data using ODBC & MFC

In the below it will show u how to adding and updating data using ODBC in MFC.

Code clipping:

```
CString sDSN;
sDSN= "ODBC;DRIVER=Microsoft Access Driver (*.mdb);DSN='';DBQ=SwecoDB.mdb";
CDatabase m_database = new CDatabase;
if(!m_database->Open(NULL, FALSE, FALSE, sDSN, FALSE))
{
    CDialog::EndDialog(0);
    return FALSE;
    AfxMessageBox(_T("open database fail!"));
}
CString strSQL1;
strSQL1="select * from Instruments";
m_InstrRecordset=new CInstrRecordset(this->m_database);
m_InstrRecordset->Open(CRecordset::dynaset, strSQL1);
////////////////////////////////////
////In the add CAddInstrumentDlg
this->m_InstrRecordset->AddNew();
this->m_InstrRecordset->m_instrID=this->m_valInstrID;
this->m_InstrRecordset->m_instrName=this->m_valInstrName;
this->m_InstrRecordset->m_typeofInstr=this->m_valType;
this->m_InstrRecordset->m_typeofStock=this->m_valTypeStock;
this->m_InstrRecordset->m_distributorID=this->m_valDistriID;
this->m_InstrRecordset->m_distributorName=this->m_valDistrName;
this->m_InstrRecordset->m_buyDateTime=this->m_valBuyDate;
this->m_InstrRecordset->m_currentStatus="In Stock";
this->m_InstrRecordset->m_salesMan=this->m_valSaleMan;
this->m_InstrRecordset->m_comments=this->m_valComments;
this->m_InstrRecordset->Update()
```

Then After the code “m_InstrRecordset->updateae()” is executed , the data will be add into the access database if no errors return.

3. Snap of the Software product

Here you can see some of the screen snaps of the instrument management tools. Then you can have a very quickly understand of what the product look like.

1. Main Screen.

Instrumentregister för SWECO VIAK

Buttons: Nytt instrument, Ny kund, Ny systemanvändare, Ny instrumenttyp, Sök instrument, Sök kund, Booking

SWECO logo

Problem- och tvävtlista

InstrumentID	InstrumentName	Type_of_Instr	Type_of_Stock	CurrentStatus	SalesPerson	Other
1520-0001	Metrolog 420mA	Datalogger	SellingType	Problem	PEAK	This one is for sold.
1520-0004	Metrolog 420mA	Datalogger	SellingType	Washing	PEAK	This one is for sold.
1520-0007	Metrolog 420mA	Datalogger	SellingType	Washing	PEAK	This one is for sold.
1520-0008	Metrolog 420mA	Datalogger	SellingType	Washing	PEAK	This one is for sold.
1815-0004	Mainstream	VH_meter	LeasingType	Washing	DOTT	That's a nice one
1805-0001	Casella	Rain_Collector	LeasingType	Washing	LAXE	Good one
1805-0005	Casella	Rain_Collector	LeasingType	Problem	LAXE	good one

Alarmlista

InstrumentID	Alarm_Time	Signature	Description
1805-0001	2007-06-10 21:25:29	DOTT	Edit
1805-0003	2007-06-11 22:40:46	DOTT	Show this Instrument, and wash it
1805-0004	2007-06-11 22:40:46	DOTT	Show this Instrument, and wash it
1805-0004	2007-06-11 22:42:21	DOTT	Alarm to do some work.
1520-0001	2007-06-13 22:32:41	DOTT	Some work.
1520-0001	2007-06-14 16:49:31	LAXE	Make a Service
1520-0008	2007-06-18 22:55:07	LAXE	some work.
1805-0001	2007-06-28 22:31:21	DOTT	do some work.

Buttons: Visa tvätt, Visa problem, Visa tvätt_problem, Rapport, Avsluta

2. Search Instrument Screen.

Sök instrument

Sökkriterier

Instrument ID:

Leverantörsnamn:

Leverantörs ID:

Typ av instrument:

Lager tillhörighet:

Aktuell status:

Inköpare:

Sökresultat

InstrumentID	InstrumentName	Type_of_Instr	Type_of_Stock	Date_of_Buy	DistributorID	DistributorName	CurrentStatus	SalesPerson	Other
1505-0001	Casella	Rain Collector	SellingType	2007-06-06 ...	232	diddid	In Stock	DOTT	fdididididid
1520-0001	Metrolog 42...	Datalogger	SellingType	2007-04-25 ...	12345678	Technolog	Problem	PEAK	This one is f...
1520-0002	Metrolog 42...	Datalogger	SellingType	2007-04-25 ...	12345678	Technolog	In Stock	PEAK	This one is f...
1520-0003	Metrolog 42...	Datalogger	SellingType	2007-04-25 ...	12345678	Technolog	In Stock	PEAK	This one is f...
1520-0004	Metrolog 42...	Datalogger	SellingType	2007-04-25 ...	12345678	Technolog	Washing	PEAK	This one is f...
1520-0005	Metrolog 42...	Datalogger	SellingType	2007-04-25 ...	12345678	Technolog	In Stock	PEAK	This one is f...
1520-0006	Metrolog 42...	Datalogger	SellingType	2007-04-25 ...	12345678	Technolog	In Stock	PEAK	This one is f...
1520-0007	Metrolog 42...	Datalogger	SellingType	2007-04-25 ...	12345678	Technolog	Washing	PEAK	This one is f...
1520-0009	Metrolog 42...	Datalogger	SellingType	2007-04-25 ...	12345678	Technolog	In Stock	PEAK	This one is f...
1520-0010	Metrolog 42...	Datalogger	SellingType	2007-04-25 ...	12345678	Technolog	Deleted	PEAK	This one is f...
1541-0000	Xbox	PlayerConsole	SellingType	2007-06-14 ...	12345611	microsof	In Stock	LAXE	
1541-0001	Xbox	PlayerConsole	SellingType	2007-06-14 ...	123998009	Microsoft	In Stock	DOTT	for leasing
1541-0002	Xbox	PlayerConsole	SellingType	2007-06-14 ...	121323	micro	Leased	DOTT	dsdidd
1545-0000	PSP-player	New Play C...	SellingType	2006-07-14 ...	1123	sony	Leased	AEPE	new one
1588-0000	Thinkpad T60	Laptop	SellingType	2007-06-19 ...			In Stock	DOTT	new
1805-0000	Casella	Rain Collector	LeasingType	2007-04-30 ...	TRA 12-567	Digital Illusion	In Stock	LAXE	Have Lease...
1805-0001	Casella	Rain Collector	LeasingType	2007-04-30 ...	TRA 12-567	Digital Illusion	Washing	LAXE	Good one
1805-0002	Casella	Rain Collector	LeasingType	2007-04-30 ...	TRA 12-567	Digital Illusion	In Stock	LAXE	Good one
1805-0003	Casella	Rain Collector	LeasingType	2007-04-30 ...	TRA 12-5678	Digital Illusion	Leased	LAXE	Good one
1805-0004	Casella	Rain Collector	LeasingType	2007-04-30 ...	TRA 12-567	Digital Illusion	In Stock	LAXE	Good one
1805-0005	Casella	Rain Collector	LeasingType	2007-04-30 ...	TRA 12-567	Digital Illusion	Problem	LAXE	Good one
1805-0006	Casella	Rain Collector	LeasingType	2007-04-30 ...	TRA 12-567	Digital Illusion	In Stock	LAXE	Good one
1805-0007	Casella	Rain Collector	LeasingType	2007-04-30 ...	TRA 12-567	Digital Illusion	In Stock	LAXE	Good one

Buttons: Sök, Sök alla, Avsluta sökning, Förhandsgranska lista, Editera valt instrument

SWECO logo

3. Instrument detailed information screen.

Visa/Editera unikt instrument

Instrumentinformation

Typ av instrument: Datalogger
Instrument ID: 1520-0008
Instrument Name: Metrolog 420mA
Leverantörs ID: 12345678
Leverantörsnamn: Technolog
Inköpare: PEAK
Lagertillhörighet: SellingType
Inköpsdatum: 2007- 4-25
Kommentar: This one is for sold.

Spara ändringar



Status på valt instrument

Aktuell status: Washing
Signatur: AEPE
Kundnamn:
Projektnummer:
Description
Kommentar:
Åsidosätt: 2007- 7-30
Ändra status

Alarm - service | kontroll | lagring

Här kan du lägga till kommande händelser och göra service rapporter

(Service | Kontroll | Reparation)

Skapa serviceprotokoll

Ändringslogg (Historisk status)

ChangeDateTime	Instrume...	Signature	StatusChangedToBe	CustomerName	ProjectNumber	Description
2007-06-10 22:54:56	1520-0008	AEPE	Washing			
2007-06-10 22:54:44	1520-0008	AEPE	In Stock			
2007-06-10 22:54:17	1520-0008	AEPE	Washing			
2007-06-06 23:15:24	1520-0008	AEPE	Washing			

Avbryt

OK

Förhandsgranska lista



Matematiska och systemtekniska institutionen
SE-351 95 Växjö

Tel. +46 (0)470 70 80 00, fax +46 (0)470 840 04
<http://www.vxu.se/msi/>