ABSTRACT

The project, Task Management System on a Pocket PC for the University Center at Texas A&M University-Corpus Christi, has been designed to aid the student workers in performing the tasks for the center. It involved in designing user interfaces for the student workers and administrator. The final system helps in improving the task management process as the existing system is completely a manual process. The project has been implemented using Visual Basic.NET, ASP.NET, ASP.NET Mobile Controls, and Crystal Reports, which forms the front end and SQL Server 2000 has been used as the back end.
# TABLE OF CONTENTS

ABSTRACT ............................................................................................................. ii

TABLE OF CONTENTS ............................................................................................. iii

LIST OF FIGURES .................................................................................................... vi

1. BACKGROUND AND RATIONALE ..................................................................... 1

   1.1 Overview of Existing System ........................................................................ 1

   1.2 Drawbacks in Existing System ....................................................................... 1

   1.3 New System ................................................................................................... 2

       1.3.1 Data Security .......................................................................................... 2

       1.3.2 Fool Proof Operation ............................................................................. 3

       1.3.3 Flexible Design ...................................................................................... 3

       1.3.4 Easy to Use and Maintain ..................................................................... 3

   1.4 Pocket PC ..................................................................................................... 3

       1.4.1 Applications .......................................................................................... 4

2. TASK MANAGEMENT SYSTEM ON POCKET PC ............................................. 5

   2.1 Pocket PC Wireless Computing .................................................................... 5

   2.2 Authentication .............................................................................................. 6

   2.3 Pocket PC Interface ..................................................................................... 7

       2.3.1 Open and Close Mode Tasks ................................................................. 7

       2.3.2 Tasks Mode .......................................................................................... 8

       2.3.3 Tour Mode ............................................................................................ 9

   2.4 Desktop Web Interface ................................................................................ 10
2.5 System Security ...................................................................................................................... 10
2.6 Pocket PC-PC-Database Connection ..................................................................................... 11

3. PROJECT REQUIREMENTS, SYSTEM DESIGN, AND IMPLEMENTATION .................................. 13
3.1 Main Components of the Project ............................................................................................ 13
3.2 Pocket PC Module .............................................................................................................. 13
3.3 Web Interface Module ....................................................................................................... 14
3.4 Project Architecture .......................................................................................................... 14
3.5 System Requirements ......................................................................................................... 15
    3.5.1 ASP.NET ................................................................................................................... 16
    3.5.2 Visual Basic.NET .................................................................................................... 17
    3.5.3 Web Services .......................................................................................................... 17
    3.5.4 Crystal Reports ...................................................................................................... 17
    3.5.5 Database Connectivity ......................................................................................... 18
    3.5.6 SQL Server ........................................................................................................... 19
    3.5.7 NDoc .................................................................................................................... 20
3.6 Database Tables .................................................................................................................. 20
    3.6.1 emp Table .............................................................................................................. 21
    3.6.2 Tasks Table ............................................................................................................ 22
    3.6.3 Location Table ....................................................................................................... 22
    3.6.4 TourLog Table ...................................................................................................... 22
    3.6.5 emsTasks Table ................................................................................................. 22
3.7 Data Flow Diagrams .......................................................................................................... 22
    3.7.1 Level-1 Diagram ............................................................................................... 22
LIST OF FIGURES

Figure 1.1. Flow chart for Login ................................................................. 2
Figure 2.1. Pocket PC .............................................................................. 5
Figure 2.2. Login Screen and Invalid Login ............................................. 6
Figure 2.3. Open Mode Tasks ................................................................. 7
Figure 2.4. Close Mode Tasks ............................................................... 8
Figure 2.5. Regular Tasks ................................................................. 8
Figure 2.6. Tour Mode ................................................................. 9
Figure 2.7. Login for the Web interface ............................................. 10
Figure 2.8. Encrypted Password ............................................................. 11
Figure 2.9. Pocket PC-PC-Database Connection .................................. 12
Figure 3.1. Two-tier Architecture .................................................. 14
Figure 3.2. Three-tier Architecture .................................................. 15
Figure 3.3. ADO.NET Architecture .................................................. 19
Figure 3.4. ER Diagram ................................................................. 21
Figure 3.5. Level-1 Diagram ............................................................. 23
Figure 3.6. DFD for the Login process ........................................... 24
Figure 3.7. DFD-Open Mode Tasks .................................................. 25
Figure 3.8. DFD-Close mode Tasks .................................................. 26
Figure 3.9. DFD-Tour Mode ............................................................. 27
Figure 3.10. DFD-Tasks Mode ......................................................... 28
Figure 3.11. Context Diagram for the Web interface ........................................ 29
Figure 3.12. Menu ......................................................................................... 33
Figure 3.13. NDoc Documentation ................................................................. 35
Figure 7.1. Documentation for Login Class .................................................... 44
1. BACKGROUND AND RATIONALE

The University Services Center at Texas A&M University Corpus Christi is dedicated to providing desired goods and professional services to students, faculty, staff and visitors to the campus. Some of the services include: Campus copies, a full service Post Office, the SandDollars Card service center, University Bookstore etc. Their goals are to assist in enhancing the quality of student life on campus and provide support to the educational, research and public service mission of the University. [TUC 2004]

1.1 Overview of Existing System

The Student Manager of the University Center holds the responsibility of creating the tasks and assigning them to the student workers. It is completely a manual process. There are regular daily tasks like opening the building and rooms, cleaning, arranging the rooms for the meetings, closing the rooms and building at the end. Student workers perform all these tasks in their scheduled job hours. At any time, the student manager can assign a new task. Once the student worker completes the work, the student manager reviews the task and closes the task.

1.2 Drawbacks in Existing System

The existing system is completely a manual process, providing a chance of error in timings of the tasks performed as there is no time stamping facility available. It is a tedious process to note down all the details on a paper. There is no backup of the system. It is very difficult to generate reports based on some queries.
1.3 New System

The proposed system is entirely computer based. An user friendly Graphical User Interface (GUI) has been developed for a Pocket PC, which serves as an application to the student manager and student workers as well. All the data are stored on a remote SQL Server, which has built in security, thereby providing ample security. The system has two interfaces, one is for the Pocket PC and the other one is Web interface. Student workers extensively use Pocket PC interface to accomplish the tasks. The student manager reviews the completed task and closes the task using Pocket PC. Both the interfaces have some common features like creating new tasks, search tasks. The student manager uses Web interface to create new users for the system.

A wireless device such as a Pocket PC is used to enter the staff ID, staff name, task ID, task category, start time, end time, and any special notes. This data can then be immediately transmitted, from on-site, to the server hosting the database. The proposed system has many advantages over the current system.

1.3.1 Data Security

Figure 1.1. Flow chart for Login

As shown in Figure 1.1, the user interface uses a login screen where the user must provide the login details in order to use the application. It also provides multi-level
security access that limits data input and report generation based on the user ID or rank. The entire database is stored in a SQL Server, which has built in security. Only privileged users can access the SQL Server.

1.3.2 Fool Proof Operation

The application uses time stamping so that there will be no time errors recorded. The application uses current system time for the time stamping. There is no scope for the user to enter the time, the application sends the current time to the database as soon as the user clicks the button on the application.

1.3.3 Flexible Design

This application is designed in such a way that any additional features can be incorporated into the existing system by re-using or modifying the code without writing the entire code again for the new features.

1.3.4 Easy to Use and Maintain

The entire application has a Graphical User Interfaces (GUI), which is easy to understand and use for non computer background users by following the navigation links on the application. It is very easy for the administrator to work on the database by using the SQL Server tools like Query Analyzer, and Enterprise Manager.

1.4 Pocket PC

From a technical standpoint, "Pocket PC" is a standard from Microsoft that sets various hardware and software requirements for mobile devices bearing the "Pocket PC" label.

For instance, any device which is to be classified as a pocket PC must

- Run Microsoft's Windows CE operating system
- Come bundled with a specific suite of applications in ROM
- Include a touch screen
- Include a directional pad
- Include a set of hardware application buttons
- Be based on a StrongARM compatible CPU [Wikipedia 2004]

1.4.1 Applications

There are numerous market applications for wireless Pocket PCs. They include doctors and nurses using handhelds to transmit real-time patient data to and from hospital LANs while at the patient's bedside. Students at universities can quickly access the Internet for research information or a professor's Web site for assignments, class announcements, and grade postings.

Wireless database is one of such products which are developed for a Pocket PC. It uses TCP/IP to exchange data between the Pocket PC and database server. One can simply set up the users in the server and list which databases will be available to them. From here clients can connect and edit single tables or stored queries from any database listed in the server and switch on the fly between tables and between databases. One can use wireless database to Add, Delete, Find or Edit any record in a table or stored query. The wireless database supports Text, Memo, Number, Date, Currency, and Yes/No data types. [Kelbran 2004]
2. TASK MANAGEMENT SYSTEM ON POCKET PC

It is not the amount of time one has, but what one do with their time, that counts most. The secret of more effective use of time with greater efficiency lies in organizing and planning our tasks. Task Management application provides an efficient and secured system to implement the tasks in a timely manner.

The application is designed with a Web interface for the Pocket PC and Desktop, and the data is transmitted from the device to the remote server. The user interface design on the Pocket PC enables authentication of the employee to access the database, establishing database connectivity with the remote server, and accessing the database hosted on the remote server. All of the above steps are accomplished by using wireless technology. The desktop version of the system is designed for the use of the student manager.

2.1 Pocket PC Wireless Computing

Figure 2.1. Pocket PC

The real objective of using a Pocket PC as shown in Figure 2.1 is to provide complete mobility to the user—the ability to have information at his fingertips regardless of where he or she is. By adding wireless capabilities, this information-at-anytime-and-in-any-place objective is met. Once mobile users in the organization go
wireless, numerous, and powerful advantages become evident. For example, the staff can easily upload and download relevant data from their organization's databases in real-time.

2.2 Authentication

Each user is provided a username and a password. In order to access any interface, a user must provide login information on the login screen as shown in Figure 2.2. This information is sent to the SQL Server for validation. On authentication by the Web server, the user has access to the database until he/she ends the session by logging out. If the username and password of the user do not match then it displays an error message as shown in Figure 2.2.

![Login Screen](image)

**Your Username or Password is incorrect.**

**Invalid Login**

*Figure 2.2. Login Screen and Invalid Login*
2.3 Pocket PC Interface

Both the student manager and student workers can use the Pocket PC interface. These interfaces have the tasks that users need to perform. Tasks are the activities that the users must accomplish as a part of their job.

The tasks are broadly classified into different categories and performed at different times by different users. These groups are listed below:

1. Open mode tasks
2. Close mode tasks
3. Tour mode tasks
4. Tasks mode

2.3.1 Open and Close Mode Tasks

These are usually the default tasks that are performed at the start of the day and end of the day. These include opening the building, opening the doors, closing the building, locking the doors, turning off the lights etc. as shown in Figure 2.3 and Figure 2.4. All these tasks are performed by the student workers. They carry the Pocket PC to their work area and enter the information immediately with time stamping.
2.3.2 Task Mode

Regular tasks, as shown in Figure 2.5, are the tasks that are done on a routine basis on any given day. Task manager assigns these tasks to the student workers.
EMS (Event Management Scheduling) is software used for scheduling the rooms in University Center. Once a user requests for a room, he or she is informed about the room availability. At the beginning of the day the task manager imports the requests from EMS and assigns the room arrangement tasks to the student workers.

2.3.3 Tour Mode

In this activity, student workers carry the Pocket PC through out the building at regular intervals starting at 8:00 AM to 8:00 PM. All the task details are updated in the database with the current system date as shown in Figure 2.6. The Student Manager monitors all the tasks that are assigned to student workers. Once the task is accomplished, the student manager reviews the tasks and changes the task status to close. [Martin 2004]

![Tour Time Calendar]

**Figure 2.6. Tour Mode**
2.4 Desktop Web Interface

Though there are some administration features available on the Pocket PC, a Web interface, as shown in Figure 2.7, with all the features such as creating new tasks, assigning tasks, maintaining the database, etc. have been developed. This interface is very helpful for the administrator or student manager to monitor all the tasks. It is very easy to add any tasks at any time. In order to maintain the integrity of the tool the database of the finished tasks is updated when the task is completed.

![Figure 2.7. Login for the Web interface](image)

2.5 System Security

All the data is stored in a secured SQL Server. The password that is supplied at login is encrypted and sent to the database where the encrypted passwords are stored as shown in Figure 2.8. When it comes to the Web interface, the administrative office staff
is the only people who have authorized access to the system through a user name and password.

![Figure 2.8. Encrypted Password](image)

### 2.6 Pocket PC-PC-Database Connection

The administrator usually works on either desktop version of the task management or the Web interface as the student workers carry the Pocket PC to the work area all the time. The administrator can use the desktop version to add new tasks and monitor all the completed tasks.

Figure 2.9 shows how the Pocket PC and Desktop system connects to the database. SQL Server provides concurrent access so that the Pocket PC users and desktop users can access the database at the same time. [Mekala 2004].
Figure 2.9. Pocket PC-PC-Database Connection
3. PROJECT REQUIREMENTS, SYSTEM DESIGN, AND IMPLEMENTATION

The project has been developed using Visual Studio.NET IDE. The GUI for the project was developed using ASP.NET and ASP.NET Mobile Development Kit. The business logic has been separated from the GUI and been coded using Visual Basic.NET. ADO.NET was used for communicating with the database, which is SQL Server 2000. Crystal Reports were used for some of the report generation. EMS tasks are imported by calling a Web method of a Web Service.

3.1 Main Components of the Project

The entire project is divided into two modules. Both modules deal with the tasks of the University Center. The modules are:

- Pocket PC Module
- Web Interface Module

3.2 Pocket PC Module

This module was developed for the student manager, student workers and administrator. The following are the main components of the module:

- Login interface
- New Task
- Close Task
- Search Task
- Import Tasks
• Import EMS Tasks
• Tour Log

### 3.3 Web Interface Module

This module was developed for the administrator. The following are the main components of this module:

• Login interface
• Change Password
• New User
• New Task
• Edit Task
• Search Task
• Import Tasks and EMS Tasks
• Tour Log

### 3.4 Project Architecture

![Figure 3.1. Two-tier Architecture](image)

**Figure 3.1. Two-tier Architecture**
A two-tier architecture as shown in Figure 3.1 has been used for the Pocket PC application. The two-tier client/server model typically separates data from the logic of an application. The database server stores the application data while the client interfaces contain the programming logic and process user input.

As shown in the Figure 3.2, three-tier architecture has been used for the administrator Web interface. The client, server, and a database are the three different components in the three-tier architecture for the project. A client sends a request to the web server; the web server in turn requests the database server for the required data, the database server after receiving the request, queries for the required data and sends it to the Web server, which in turn sends this data to the client. The data received from the server is presented in the form of a table. Although the name describes the model as a three-tier, for all practical purposes database server and the web server are generally on the same computer.

![Figure 3.2. Three-tier architecture](image)

### 3.5 System Requirements

The project was implemented on a Pocket PC running Pocket PC 2002 Operating System. A Windows XP with IIS installed on the server was used, which functions as a Web server for the system. The Web server runs the ASP scripts and database modules.
The Pocket PC application was developed in ASP.NET, Visual Basic.NET and SQL Server 2000.

Minimum requirements for running Visual Basic.NET are:

- **Processor**: 450-Megahertz (MHz) Pentium II-class processor, 600-MHz Pentium III-class processor recommended.


- **Hard Disk**: 500 MB of available space required on system drive, 1.5 gigabytes (GB) of available space required on installation drive. Additional 1.9 GB of available space required for optional MSDN Library documentation.

[MSDN 2004a]

### 3.5.1 ASP.NET

Active Server Pages.NET (ASP.NET) is the infrastructure built inside the .NET Framework for running Web applications. An ASP.NET Web application is executed through a series of HyperText Transfer Protocol (HTTP) request and response messages between the client browser and Web server.

In many ways, the ASP.NET is just a successor to ASP 3.0, but on the other hand, it introduces so many new features for enhancing Web application development. ASP.NET supports Object Oriented Programming (OOP) techniques that were traditionally available only to desktop applications. All ASP.NET pages are compiled
before they are executed. Overall performance of compiled code is much better than the interpreted code. [DIWA 2003]

3.5.2 Visual Basic.NET

Visual Basic .NET 2003 provides the easiest, most productive language and tool for rapidly building applications for Microsoft Windows® and the Web. Ideal for existing Visual Basic developers as well as new developers in the Microsoft .NET development environment, Visual Basic .NET 2003 delivers enhanced visual designers, increased application performance, and a powerful integrated development environment (IDE) to get the user on the fast track to application development. [MSDN 2004b]

3.5.3 Web Services

Web Services are Web based enterprise applications that are used to exchange data between distributed applications. They are the fundamental building blocks in the move to distributed computing on the Internet. Web Services expose useful functionality to Web users through a standard Web protocol. In most cases, the protocol used is Simple Object Access Protocol (SOAP). Web Services provide a way to describe their interfaces in enough detail to allow a user to build client application to talk to the services. This description is usually provided in an XML document called Web Services Description Language (WSDL) document. Web Services are registered so that potential users can find them easily. This is done with Universal Discovery Description and Integration (UDDI). [MSDN 2004c]

3.5.4 Crystal Reports

Crystal Reports for .NET provides developers with a fast and highly productive way to create and integrate presentation-quality, interactive reports. Crystal Reports for
Visual Studio .NET provides a comprehensive reporting solution for .NET developers that is thoroughly integrated with both the Visual Studio .NET IDE and the .NET Framework. Crystal Reports supports ADO.NET, XML Web Services, and ASP.NET server controls and caching. It also integrates seamlessly with the Visual Studio .NET Server Explorer, toolbox, and design environment. It has a rich programming model and flexible options for customizing and deploying reports. [BOCR 2003]

3.5.5 Database Connectivity

ADO.NET provides consistent access to data sources such as Microsoft SQL Server, as well as data sources exposed through OLE DB and XML. Data-sharing consumer applications can use ADO.NET to connect to these data sources and retrieve, manipulate, and update data. ADO.NET leverages the power of XML to provide disconnected access to data. ADO.NET was designed hand-in-hand with the XML classes in the .NET Framework.

ADO.NET cleanly factors data access from data manipulation into discrete components that can be used separately or in tandem. ADO.NET includes .NET Framework data providers for connecting to a database, executing commands, and retrieving results. Those results are either processed directly, or placed in an ADO.NET Dataset object in order to be exposed to the user in an ad-hoc manner, combined with data from multiple sources. The ADO.NET Dataset object can also be used independently of a .NET Framework data provider to manage data local to the application.

The ADO.NET classes are found in System.Data.dll, and are integrated with the XML classes found in System.Xml.dll. When compiling code that uses the System.Data
namespace, reference both System.Data.dll and System.Xml.dll. Figure 3.3 illustrates the components of ADO.NET Architecture. [MSDN 2004d]

![Figure 3.3. ADO.NET Architecture](image)

**3.5.6 SQL Server**

Microsoft SQL Server 2000 is a full-featured relational database management system (RDBMS) that offers a variety of administrative tools to ease the burdens of database development, maintenance and administration. It comes with several tools like Enterprise Manager, Query Analyzer, SQL Profiler, Service Manager, and Data Transformation Services.

Enterprise Manager is the main administrative console for SQL Server installations. It provides with a graphical "birds-eye" view of all of the SQL Server installations on the network. One can perform high-level administrative functions that affect one or more servers, schedule common maintenance tasks or create and modify the structure of individual databases.
Query Analyzer offers a quick and dirty method for performing queries against any of the user’s SQL Server databases. It's a great way to quickly pull information out of a database in response to a user request, test queries before implementing them in other applications, create/modify stored procedures and execute administrative tasks. [IMSS 2003]

**3.5.7 NDoc**

NDoc is an open-source documentation generator for .NET projects. It takes project output in the form of .dll or .exe file and an XML file containing custom documentation added within the code and generates MSDN look-alike HTML based help files. The tool uses a mixture of Reflection on the library or executable and the descriptions exported to the XML files to populate this documentation. In C# XML documentation is added by entering three forward-slashes rather than the usual two and following it with xml formatted comments. Visual Studio helps in this process by adding some skeleton tags before methods when one begin to type “///”. [ONCF 2003]

**3.6 Database Tables**

The data from the users are stored in SQL Server database in the form of tables. Figure 3.4 illustrates the Entity-Relation Diagram of the tables used in the project.
The following section describes the tables that were used for both the Pocket PC and Web modules.

3.6.1 emp Table

This table is used to store the username and password of the employees. Username acts as a primary key. When user enters the username and password on the Pocket PC, it sends the request to the server to check whether or not both values match in the login table.
3.6.2 Tasks Table

This was created to store the information about the tasks. Task number acts as a primary key. Based on the task number all the information about the tasks can be retrieved. This table stores information such as who created the task, due date, assigned to etc. LocID acts as a foreign key, which is used to retrieve the location information.

3.6.3 Location Table

This was created to store the location of the tasks. LocID acts like a primary key. This table stores information like floor number, type etc.

3.6.4 TourLog Table

This table is used to store all the tour details that the student worker performs at regular intervals. LocID serves as a foreign key. Combination of LocID, TourDate, and Time act as a primary key.

3.6.5 emsTasks Table

This table is used to store the EMS Tasks. The application retrieves the EMS Tasks from this table by calling the Web method of a Web service. [Martin 2004]

3.7 Data Flow Diagrams

3.7.1 Level-1 Diagram

Figure 3.5 illustrates the context level diagram of the Pocket PC application. User is the external entity, who provides username and password to the log in process. Based on the user position, he or she will be able to access task management process or administrative process or both.
3.7.2 Data Flow Diagram for Login Process

When the user provides username and password, the login process sends the data to the remote database for the validation. Based on the response of the database server, the login process gives the control to the task management or administrative process, if the user is valid, the system takes the user to the tasks interface. If user is invalid then the application asks the user to re-enter the login information as shown in Figure 3.6.
3.7.3 Data Flow Diagram for the Task Management

Figure 3.7 illustrates the data flow of open mode tasks. If the open mode is processed then the beginning of the day tasks are processed. This process imports all the tasks from the remote database. If the open mode is not processed then it asks whether or not to process. If the user does not want to process then the process are terminated.
Figure 3.8 shows the flow of data at the close mode. If the close mode is processed then the end of the day tasks will be processed. This process imports all the tasks from the remote database. If the close mode is not processed then it asks whether or not to process. If the user does not want to process then the process will be terminated.

[Martin 2004]
Figure 3.8. DFD-Close mode Tasks

Figure 3.9 illustrates the data flow of tour mode tasks. If the tour mode is processed then the task manager imports all the tasks for that day and monitors the tasks. This process imports all the tasks from the remote database. If the tour mode is not
processed then it asks whether or not to process. If the user does not want to process then the process will be terminated.

Figure 3.9. DFD-Tour Mode

Figure 3.10 shows the flow of data at the tasks mode where the tasks imported from EMS will be processed. If the tasks mode is not processed then it asks whether or
not to process. If the user does not want to process then the process will be terminated.

[Martin 2004]

3.10. DFD-Tasks Mode
3.7.4 Context Diagram for the Web interface module

Figure 3.11 shows the context level diagram for the Web interface. The administrator will have the access to create/delete user, create/delete tasks. He or she can also import tasks from the remote database. In Assign task process, he or she will assign the tasks to the student workers.

![Context Diagram for the Web interface](image)

Figure 3.11. Context diagram for the Web interface

3.8 Implementation

This section describes the implementation details of the project. As mentioned before, this project contains two modules: Pocket PC and Web interface. The Pocket PC module is mainly for the student workers to accomplish the tour mode tasks. Student workers carry the Pocket PC and tour the building to perform the required tasks. This
module has been developed using ASP.NET mobile controls with VB.NET for the business logic. Administrators and Student managers will use the Web interface to perform various tasks. This module has been developed using ASP.NET and VB.NET. Crystal Reports and Web services are also used for some of the functionality. In both the modules, ADO.NET technology is used to interact with the database, which is SQL Server 2000.

3.8.1 Pocket PC Module

The following section explains the pages that are used in Pocket PC module.

logon.aspx

This is the starting page for this module. When the user enters the URL for the application on the browser, this page is loaded. The user has to provide the username and password to access the system. The password is encrypted and compared with the encrypted password in the database and matches with the username. Upon validation the user account is stored in a session and the user is redirected to loggedM.aspx, which in turn transfers to mainPageM.aspx.

mainPageM.aspx

This is the main navigation page where the user can use all the links available on the page. Depending on the role, the users are given access to the features of the application. Admin and the Student manager can access all the available features whereas access to student workers is limited to certain features.

newTaskM.aspx

Only the Admin and the Student Manager can have access to this page. Access to the student workers is denied. This is the place where the authorized person can create
new tasks. The data that is sent from this page is stored in Tasks table. This page generates random unique TaskID which is used as a primary key for the table.

*closeTaskM.aspx*

Only the Admin and the Student Manager can access to this page. Once the student worker completes the task, the Student manager reviews the task and closes it if everything is done according to the requirement. This page imports all the tasks based on the filter. Here the status of the task is used as the filter. Student manager has the authority to close any of the tasks. So he or she imports the tasks based on open status, completed status or all tasks. Once the student manager closes the task, the status of the tasks is changed to close in the database.

*searchTaskM.aspx*

The users can search the tasks based on the status or mode of the tasks.

*importTaskM.aspx*

The users can use this page to import tasks based on the type. Available types are Open, Closed, and Regular. This page displays all the tasks along with the location. The user can carry the Pocket PC to that location and accomplish the task. After completing the tasks the user can change the status to completed. Also time stamping is used here to reduce the time errors. System time is stored in the database as soon as the user changes the state to completed. This page retrieves the data from the Tasks table and once the task is completed, it updates the table with the task details.

*importEMS.aspx*

These tasks are separated from the open, closed, and regular tasks. The emsTasks table is used in this page. This table is mainly used to store special events that occur in
the building. A Web service is created to accomplish this task. A Web method in that service access the database and stores the data in a dataset. This method returns dataset to the calling clients. This page acts as a client, which creates a proxy class to that Web service and calls the Web method available in that service. Based on the dataset that is returned from the Web method, the user can accomplish the EMS tasks.

*tourLogM.aspx*

This is the main feature of this module. A calendar control is used to select the day of the tour. The Student worker carried the Pocket PC at regular intervals and selects the location where he or she wants to perform the task. For the location of type room, the user also required to store the count of the people in the room.

*logoutM.aspx*

The user can exit from the application. Here all the session values of the user are cleared off.

### 3.8.2 Web Interface Module

The following section explains the pages that are used in the Web interface module.

*index.html*

This is the starting page for this application. It uses a frameset to incorporate a logo, menu, and the content.

*menu.aspx*

This page as shown in Figure 3.12 provides the links to navigate to different parts of the system. It uses a TreeView, an Active X Control, to display the links in a hierarchical way. The TreeView is an ASP.NET server control for authoring user
interfaces representing hierarchical data. The TreeView control delivers HTML based on the browser type. In down-level browsers, the page content sent by the TreeView ASP.NET server controls is HTML 3.2, whereas, in up-level browsers the HTML, pages also use DHTML behaviors, which are client-side components that define custom elements. With Web Controls, an up-level browser is Internet Explorer 5.5 or later, and a down-level browser is Internet Explorer 5.01 or earlier or a browser other than Internet Explorer.

![Figure 3.12 Menu](image)

**Figure 3.12 Menu**

`login.aspx`

The user has to provide the username and password to access the system. The password is encrypted and compared with the encrypted password in the database and
matches with the username. Upon validation the user account is stored in a session and the user is redirected to logged.aspx.

*logged.aspx*

The user comes to this page once the account has been validated. This page provides brief description about the features of this application.

*register.aspx*

Only Admin or the Student Manager can able to create a new user. Student workers have to provide all the required details so that the account can be created. This page generates a random password which is sent to the users email address. This random password is encrypted and stored in the database.

*pwdhelp.aspx*

At times the user might forget his or her password. So there should be a way to reset the password. Here is the place where the user can request for a new password. In order to receive new password, the user has to supply username and email that he or she has provided at the time of registration. The code generates a random password, which is encrypted and stored in the database, and sends it to the email address of the user.

*changepwd.aspx*

Once the user receives the password to the email account, he or she needs to access the application and change the password. This page asks for the old password and new password. The new password is encrypted and stored in the database.

*edittask.aspx*

Only the Admin or Student Manager can edit the tasks. Once the user has chosen the type of task, all the tasks with details are displayed on this page. Now the user has an
opportunity to edit the task and update the database. At any time, the user can cancel the process. Available task types are Open, Closed, and Regular.

searchtask.aspx

This page asks for the status of the task and based on the input, it displays the tasks with all the details. Available task statuses are Open, Closed, and Completed.

importtask.aspx

The users can use this page to import tasks based on the type. Available types are Open, Closed, and Regular. A report will be generated with all the task details in crystal report format.

emstasks.aspx

These tasks are separated from the open, closed, regular tasks. The emsTasks table is used in this page. This page sends a request to a Web method of a Web service and data that has been sent from the Web method is displayed in a Crystal report format.

3.9 Documentation

NDoc tool has been used to generate the documentation files for this project. It generates class library documentation from the assembly file of this project. It used pluggable documenters to generate documentation in MSDN online style Web pages as shown in Figure 3.13. It provides the description of all the classes, members, and objects used in the application.
Figure 3.13. NDoc Documentation
4. TESTING AND EVALUATION

This section describes the usability and user interface design of the project. After the coding part of the project is complete a set of testing schemes have been used to see if the complete project is performing the tasks as required. A Pocket PC has been used to test the flow of the data from the Pocket PC to the remote server. All the user interfaces have been checked for any spelling and grammatical errors. The following is the list of tests that was performed for this project.

- Code Testing
- Usability Testing

4.1 Code Testing

In this testing, all the written code was tested for different input values to see if the final output is the same as expected. The application was tested while it was being developed. The project is divided into modules. Each module was individually tested for its functionality. The successfully tested modules were then integrated progressively as the different functionalities were developed. Every time the modules were integrated, integration testing was carried out to check whether the new feature affects the other functionality. Different concepts of Black Box testing and White Box testing have been used. In white box testing, all the internal workings of the project were tested. The functions of the project were tested in black box testing where the user provides different inputs and observes the outputs.
4.2 Usability Testing

As end users became more diverse and less technical, interactive systems came to be compared and evaluated with respect to usability – the quality of a system with respect to ease of learning, ease of use and user satisfaction. Usability testing is used to identify how users actually interact with the application. In a typical approach, users, one at a time or two working together, use the application to perform tasks, while one or more people watch, listen, and take notes. Throughout the development phase, importance has been given to the ease of use and functionality of the interface. The user can easily navigate the Website without any assistance.

The following are some of the list of activities that were tested for.

- Appearance: All the pages in the application were designed using same color, font, and font style. All the pages were checked for the uniformity in the appearance.

- Ease of navigation: All the pages in the application are connected through navigation links. Users can easily navigate through different pages of the application using these links. The application was tested to see if there are any links that break the application and made sure that all the links are properly connected to the required pages.

- Consistency: The application was tested for consistency of the data that are being displayed on the pages.

- Ease of using the application on a Pocket PC: On a Pocket PC, typing is very difficult without using an external keyboard. So to make the users comfortable,
the application is designed by populating most of the data in a Dropdown box, which makes easy for the user to just select the item instead of typing the input.
5. FUTURE WORK

This project has some scope for enhancement in the future. The following are the some of the ideas that can be incorporated into this project in future.

1. Provide users with different formats while importing tasks. Currently the project displays the data in crystal report. Excel, Word, and PDF are some of the other formats that can be used with some modification of the code by calling the APIs provided by the Microsoft.

2. As most of the departments in the University use MS Office, providing Office automation to the existing project with printing options will ease the work for the student workers and managers. A pre-formatted word or excel documents can be used as a template for displaying the tasks. The office automation feature should fetch the data from the SQL Server and fill in the cells of the word or excel document of the template. This template can be directly sent to the printer.
6. CONCLUSION

The Pocket PC application project is aimed at helping the task manager, student workers at University Center. It provides a reliable and easy way of accomplishing tasks. All the data will be stored securely in the secured SQL Server. The student workers do not write the information of the tasks on a paper. They send all the information regarding the tasks to the SQL Server using the Pocket PC. The system is secured by encrypting the user’s password and storing it in a SQL Server, making it not possible for even the administrator to know the actual password of the user. The task manager does not need to think about the storage of the log books as all the data is stored in a safe and secure SQL Server. The Web interface helps the administrator to directly create, delete, and assign tasks. He or she can directly retrieve the tasks information using the search field.

The major accomplishments that were achieved with this project are an efficient way to handle the tasks, secured data storage, easy way to generate reports. On the whole, the project solves the purpose of staff and student workers.
ACKNOWLEDGEMENTS

I express my sincere thanks to my project committee chairperson Dr. Dulal Chandra Kar, for his guidance, support, and valuable insights all along the project, right from the proposal phase. I also thank the other committee members Dr. Mario Garcia and Dr. Long-zhuang Li for their encouragement and valuable feedback during various stages of the project.

I sincerely thank Mr. Bob Martin, Systems Programmer II, for giving me an opportunity to work on this project.
BIBLIOGRAPHY AND REFERENCES


APPENDIX A – DOCUMENTATION

Figure 7.1. Documentation for Login Class

For more documentation refer the enclosed Compact Disk (CD)
APPENDIX B – SOURCE CODE

Refer the enclosed Compact Disk for the source code.