Software Design Specification

Final Version
<table>
<thead>
<tr>
<th>Document Version</th>
<th>Section</th>
<th>Description</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>Design Consideration</td>
<td>Created and written</td>
<td>Ibrahim Krvavac</td>
</tr>
<tr>
<td>0.2</td>
<td>Design Consideration</td>
<td>Updated</td>
<td>Ali Bahaloo Hore – Seyed Samad Ghasemi</td>
</tr>
<tr>
<td>0.3</td>
<td>System Main Features</td>
<td>Written</td>
<td>Ali Bahaloo Hore – Seyed Samad Ghasemi</td>
</tr>
<tr>
<td>0.4</td>
<td>System architecture And Strategies</td>
<td>Written</td>
<td>Ali Bahaloo Hore – Seyed Samad Ghasemi</td>
</tr>
<tr>
<td>0.5</td>
<td>Interface description</td>
<td>written</td>
<td>Ali Bahaloo Hore – Seyed Samad Ghasemi</td>
</tr>
<tr>
<td>0.7</td>
<td>Data Interface</td>
<td>Written</td>
<td>Ali Bahaloo Hore – Seyed Samad Ghasemi</td>
</tr>
<tr>
<td>0.8</td>
<td>Detailed Design</td>
<td>Written</td>
<td>Ali Bahaloo Hore – Seyed Samad Ghasemi</td>
</tr>
<tr>
<td>0.9</td>
<td>Policies and Tactics</td>
<td>Written</td>
<td>Kuhan Loh</td>
</tr>
<tr>
<td>1.0</td>
<td>Introduction</td>
<td>Written</td>
<td>Daniel Asovic</td>
</tr>
<tr>
<td>1.1</td>
<td>Introduction</td>
<td>Updated / Edited</td>
<td>Kuhan Loh</td>
</tr>
<tr>
<td>1.2</td>
<td>Glossary</td>
<td>Written</td>
<td>Daniel Asovic</td>
</tr>
<tr>
<td>Final</td>
<td>Entire SDS</td>
<td>Proof read, Corrected grammar and spelling mistakes, Formatting</td>
<td>Kuhan Loh</td>
</tr>
</tbody>
</table>
# Table of Contents

Version History .................................................................................................................. 2

1. Introduction .................................................................................................................. 5
   1.1. Purpose .................................................................................................................... 5
   1.2. Scope ....................................................................................................................... 5
   1.3. Intended audience .................................................................................................... 5
   1.4. The product ............................................................................................................. 5

2. Acronyms, Abbreviations & Definitions ........................................................................ 6

3. System Overview .......................................................................................................... 7
   3.1. The system main features: ...................................................................................... 7

4. Design Considerations .................................................................................................. 8
   4.1. Assumptions and Dependencies .......................................................................... 8
       Related software or hardware ....................................................................................... 8
       End-user characteristics ............................................................................................. 8
   4.2. General Constraints ............................................................................................... 8
   4.3. Goals and Guidelines ............................................................................................. 9
   4.4. Development Methods ............................................................................................ 9

5. System Architecture and Strategies ............................................................................ 10
   5.1. System architecture ............................................................................................... 10
   5.2. This Web Application (OSS) is composed of six different components. ............... 11

6. Interface Description .................................................................................................... 13
   6.1. User Interface ....................................................................................................... 13
       Company Schedule page: .......................................................................................... 13
       My schedule page: ..................................................................................................... 13
       Home Page: ............................................................................................................... 13
       Default page: .............................................................................................................. 13
       Change Password: ..................................................................................................... 13
Password Recovery: ...................................................................................................................... 13
Admin Pages: .............................................................................................................................. 13
Messaging pages: .......................................................................................................................... 14

7. Data Interface............................................................................................................................. 15
   E/R Diagram............................................................................................................................... 18

8. Detailed design.......................................................................................................................... 19
   Detailed Design Description: .................................................................................................. 19
   8.2. Class Diagram .................................................................................................................. 33

9. Policies and tactics .................................................................................................................... 33
   9.1. Specific “Products” Used ................................................................................................. 33
   9.2. Plans for testing the software ........................................................................................ 33
1. Introduction

1.1. Purpose
The purpose of this SDS document is to give insight into the design of the web application. This
document will contain information about the system architecture, design considerations and so on.
This is to give the reader, a better understanding of the design process and procedure for this web
application. This document is also used by the developers as a “template” in order to continue with
the coding.

1.2. Scope
This document is written for the OSS. It will give the reader enough information about the design
procedure. The goal with the design is to make a system as easily understandable and as user friendly
as possible.

1.3. Intended audience
The major intended audience of this document is mainly the developers who are going to develop
the web application. However this document will be revised by supervisors and course coordinators
for feedbacks.

1.4. The product
The name of the product is “Online Scheduling System” or” OSS” in short. A web application that will
help the company (ViaMagna) to manage their employees work shifts online. The product will
enable and help the employees to access their schedules and work information online and book the
appropriate shifts easily. The manager with OSS can also assign shifts for company’s workers. Hence
the new system will be replaced for the current system which is a paper-based system.
## 2. Acronyms, Abbreviations & Definitions

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADO</td>
<td>ActiveX Data Objects</td>
</tr>
<tr>
<td>Announcement</td>
<td>A message viewable by everyone</td>
</tr>
<tr>
<td>ASP</td>
<td>Active Server Pages</td>
</tr>
<tr>
<td>.aspx</td>
<td>Active Server Page Extended</td>
</tr>
<tr>
<td>Ajax</td>
<td>Asynchronous JavaScript and XML</td>
</tr>
<tr>
<td>BCL</td>
<td>Base Class Library</td>
</tr>
<tr>
<td>Calendar Component</td>
<td>Weekly Calendar component in Company’s schedule to show available shifts.</td>
</tr>
<tr>
<td>Calendar Control</td>
<td>Next and Prev buttons (label) for the calendar component.</td>
</tr>
<tr>
<td>Content</td>
<td>Next and Prev buttons (label) for the calendar component</td>
</tr>
<tr>
<td>CLI</td>
<td>Common Language Infrastructure</td>
</tr>
<tr>
<td>CLR</td>
<td>Common Language Runtime</td>
</tr>
<tr>
<td>CLS</td>
<td>Common Language Specification</td>
</tr>
<tr>
<td>CSS</td>
<td>Cascading Style Sheet</td>
</tr>
<tr>
<td>Company</td>
<td>Via Magna</td>
</tr>
<tr>
<td>DLL</td>
<td>Dynamic Link Library</td>
</tr>
<tr>
<td>E/R Diagram</td>
<td>Entity-Relationship Diagram</td>
</tr>
<tr>
<td>Employee/Employees</td>
<td>Employees of ViaMagna</td>
</tr>
<tr>
<td>Event</td>
<td>Has the same meaning of shift.</td>
</tr>
<tr>
<td>Html</td>
<td>Hyper-Text Markup Language</td>
</tr>
<tr>
<td>IIS</td>
<td>Internet Information Services</td>
</tr>
<tr>
<td>Manager</td>
<td>The person managing the software at Via Magna</td>
</tr>
<tr>
<td>MasterPage</td>
<td>A single master page defines the look and feel and standard behavior that you want for all of the pages (or a group of pages) in your application</td>
</tr>
<tr>
<td>OSS</td>
<td>Online Scheduling System</td>
</tr>
<tr>
<td>Schedule</td>
<td>Work shift schedule</td>
</tr>
<tr>
<td>Shift</td>
<td>Work shifts for Via Magna employees</td>
</tr>
<tr>
<td>SQL server</td>
<td>Structure Query Language, refers to external software, a Database Management System</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modeling Language</td>
</tr>
<tr>
<td>User/Users</td>
<td>Person using OSS</td>
</tr>
<tr>
<td>VB</td>
<td>Visual Basic</td>
</tr>
<tr>
<td>Visual Studio</td>
<td>Microsoft IDE.</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
<tr>
<td>.NET</td>
<td>Microsoft Development Framework</td>
</tr>
</tbody>
</table>
3. System Overview
The software that “SEVEN Software” is designing is “Online Scheduling System” or “OSS” in short. A web application that will help the company (ViaMagna) to manage their employees work shifts online. The product will enable and help the employees to access their schedules and work information online and book the appropriate shifts easily. The manager with OSS can also assign shifts for the company’s workers. The employees will be able to check their schedules and contract new shifts from the comfort of their own home, with the only requirements being that the user has to have a computer with a web browser installed, an internet connection and account information.

The web application is written in .NET Framework 3.5 and it will be run on a web hosting server that supports .NET Framework 3.5. It connects to the database (SQL Server 2005) through the ADO.NET technology. In terms of design the aim is to make it simple, understandable as well as easy to use.

3.1. The system main features:

- User Authentication
- Different types of company’s schedule presentation – Available and booked shifts
- Presenting amount of worked/planned hours
- Presenting Worker’s Schedule
- Shift booking tool
- Shift management – Creating / Editing / Deleting (manager function)
- Shift assignment tool – From available shifts or by creating a new shift (manager function)
- Style management – Schedule component style (manager function)
- User management – Add / Edit / Delete (manager tool)
- Messaging service – send and receive messages
- Announcement service (manager function)
4. Design Considerations

4.1. Assumptions and Dependencies

Related software or hardware
As a matter of client use, this web application is operating independent but as a matter of web hosting server, there should be server with the operating system such Microsoft Windows XP professional, Windows Vista Business, Windows Vista Ultimate, Windows Server 2008 and Windows Server 2005 which includes Internet Information Services (IIS 6 or IIS 7) that support .NET framework 3.5. The server should also support the SQL Server 2005 database. The reason behind choosing .NET 3.5 is that the latest .NET framework supports Ajax, since we are using Ajax as one of our components; .NET 3.5 is the most suitable framework for our web application that contains Ajax as one its Built-in components and libraries.

For the client machines, OSS will be operating system independent. It will work on all operating systems as long as those systems are equipped with a web browser program that is compatible with html since ASP.NET codes are translated to html in the client machine.

End-user characteristics
Even though the computer skills of most end-users of OSS are identified as intermediate, it cannot be assumed that all users are at that level. Therefore the aim is to make this system as easy to understand as possible. This means that the core functionality of the system will be made simple enough for users with poor computer and internet knowledge.

4.2. General Constraints

End-user environment
Requirement: A webpage based system
Impact: This requirement is by far the one with the biggest influence on the system design. It imposes the web browser limitations on the user interface, server-client considerations such as amount of information that is going to be passed between the two, data encryption etc.

Standards compliance
OSS will have to conform to three standards: HTML 4.01 and .NET framework 3.5 and Microsoft SQL Server 2005.

Data repository and distribution requirements
Requirement: There is necessary data that will have to be stored. For example, the different shift hours, and employee details.
Impact: A database is going to be used to store this data which affects the whole design process because a big part of the system is about retrieving and storing data.

Security requirements
Requirement: Authentication is required
Impact: A login page is necessary. This will be handled by ASP.NET configuration for memberships and roles with 128-bit encryption. Each user has a login username and password. The password should contain at least 7 characters containing 6 alphanumeric and 1 non-alphanumeric.
Capacity limitations
Requirement: Support for minimum of 150 users.
Impact: Server and database provider must be chosen accordingly.

4.3. Goals and Guidelines
There is one major guideline derived from the interview. The company desires that the final web application should be user-friendly and should be supportive for all of the specified requirements. The reason for this is no doubt to minimize the resources needed to teach employees the new system. Making the transition from old to the new system easier is also a reason.

4.4. Development Methods
The iterative approach was identified as the most desirable among the group members and was therefore adopted for the system design. (Reference)

The process method is divided into these parts:

1. Analyzing Customer Requirements
2. Designing the database by sketching the E/R Diagram.
3. UML Designs (Use Case Diagram – System Sequence Diagrams)
4. Analyzing the classes that will be needed for implementation.
5. Designing the class diagram.
6. Designing Virtual Windows
5. System Architecture and Strategies

5.1. System architecture
The architecture used for this web application is the Three-Tier Layered Architecture. The Three-tier Layered architecture has three different tiers being Presentation (Client Tier), Application Logic, and Technical Services. The presentation tier (client tier) is basically the most upper level of the structure which is used to present the information on the client machine. The Application logic is the tier where all the VB.net, ASP.net and ADO.net code are being held and stored. And the Data tier is the tier for holding the database and stored procedures.
5.2. This Web Application (OSS) is composed of six different components.

1. Company’s Schedule:
   This component contains information regarding all the different worker’s shifts. It is decomposed into these parts:
   1. View Available shifts: This contains data about the available shifts. The available shifts are created by the manager in order for the workers to book their own shifts.
   2. View Company’s Schedule: The information regarding all the booked shifts are displayed here.

2. My Schedule:
   This component holds information about the shifts that are booked or assigned to a specific worker.

3. Messaging:
   The messaging component is responsible for holding the information and data with sending and receiving messages. Sub components are listed below:
   1. Composer: Allows the user to send messages to other users.
   2. Inbox: Allow the user to view their message inbox.
   3. Sent: Allows the user to view their sent messages.

4. Announcement:
   This component is responsible for the event of creating and displaying announcements which are created by the manager.

5. Authentication:
   This component is responsible for the logging and logging out events and all other events that is relevant to authentication. It is decomposed into these sub components:
   1. Login / Logout: This function allows users to log onto the web application and log out from it.
   2. Change Password: This allows the user to change his/her password.
   3. Password Recovery: This allows the user to retrieve his/her password in the event that the user forgets his login details.

6. Managing Tool:
   This component is responsible for all the functionalities that only the manager is allowed to use. It is decomposed into these sub systems:
   1. Manage users: This contains the functionalities of adding / editing / deleting users.
   2. Manage shifts: This contains the functionalities of creating / editing / deleting shifts.
   3. Shift assignment: This contains the functionality of assigning shifts to workers either from available shifts or by creating a new shift at the time of assigning.
   4. Manage style: This contains the functionality that allows the manager to change the style and formatting of the webpage.
Note: all parts of the GUI include one .aspx (Visual Part) file and one .aspx.vb (Visual Basic Code Part) file
6. Interface Description

6.1. User Interface

Company Schedule page:
*Purpose:* To retrieve company’s schedule, all shifts and all available shifts. The workers are also able to book shifts through this page.

*Output:* A calendar in weekly view appears. A radio button list is there to filter the mode to change the viewing options between “available shifts” and “Company’s Schedule”. Another small calendar appears underneath the “shift booking” title for the user to select a date. After selecting a date the tool expands and it loads all available shifts into a drop down list. Details of the selected shift will appear after selecting a date. There is also a button to book the selected shift. The button has an Ajax extender for confirmation.

My schedule page:
*Purpose:* To view a person’s working schedule

*Output:* A calendar showing all shifts that are book by a specific person. There are two buttons for navigating between the calendar weeks. Each shift has a green arrow allowing the user to click on it in order to expand and show the full details of that shift.

Home Page:
*Purpose:* To display the announcements that are created by the manager.

*Output:* A table for displaying all made announcements. It also contains a “view” button for the user to click on it and display the full text of the announcement.

Default page:
*Purpose:* Enables users to log in with their username and password.

*Output:* Textboxes for username and password input. A link for “forgotten password” to recover the user’s password is also provided.

Change Password:
*Purpose:* Enables the users to change their login password.

*Output:* Textboxes for the user to input the old and new password.

Password Recovery:
*Purpose:* To recover the password.

*Output:* A text box to enter the username and a next button. When that button is clicked the system asks for the security question and a text box appears for the user to enter the answer.

Admin Pages:

6.1.1. Manage Users:
*Purpose:* To manage the users of the web application

*Output:* Radio buttons to select from the modes for Adding / editing / deleting the users. By selecting the “add new user” radio button the tool expands and brings up text boxes for the manager to fill in with the new user information. By selecting the “Edit User” a drop down list loads all existing users for the manager to select and edit their information. When the
manager selects any user from the dropdown list the table loads up all information of the selected user. The text boxes are editable so the manager can edit the information. When the manager selects “delete user” from the radio button, the system loads a drop down list with the username of the user. Also a button to delete the selected user appears.

6.1.2. **Manage Shifts:**
*Purpose:* To create/assign/edit/ delete shifts
*Output:* A radio button list appears with two selection options. By selecting the “shift management tool” a table appears with the list of all available and booked shifts. A button to create new shifts appears. When the manager clicks on that button the tool expands and displays text boxes and calendars for information such as shift title, start and end date and time and etc. When the manager clicks on “Shift Assignment tool” the tool expands and brings two radio buttons. Then the manager can decide to choose between “Assign to available shifts” or “Assign to new shift”. When the manager selects “Assign to available shifts” the list of all available shifts appears and when the manager selects “Assign to new shift” user name drop down list and other tools appears for the shift’s details.

6.1.3. **Make Announcements:**
*Purpose:* To make new announcements.
*Output:* Text boxes for title and description of the announcement.

6.1.4. **Manage Style:**
*Purpose:* To change the style of the calendar component which shows the company’s schedule.
*Output:* A number of text boxes for the manager to fill in. and a button to save the changes.

**Messaging pages:**

6.1.5. **Inbox:**
*Purpose:* To view messages
*Output:* Message title, sender name and a “View” button to show the full message.

6.1.6. **Composer:**
*Purpose:* To compose a new message
*Output:* A textbox to enter the message, and a button to send the message. A drop down list will appear for the user selection.

6.1.7. **Sent:**
*Purpose:* To view the sent messages.
*Output:* Message title and receiver name in a table with a view “button” to view the message.
Fig: User interface Diagram – Starting from login page, if the user is authorized it redirects to the home page. From there the users can select which pages they want to view. If the user is a manager, it is possible for him/her to use the manager tools.

7. Data Interface
In addition to user interfaces there are VB classes in the App-Code folder that work behind the scene to process the information regarding the schedules, announcements and messaging.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clsCalendars.vb</td>
<td>Visual Basic Class</td>
<td>This Class Handles the different calendar instances. The ID property is the same ID that is used when setting Calendar property of the Calendar control.</td>
</tr>
<tr>
<td>clsControlText.vb</td>
<td>Visual Basic Class</td>
<td>This Class Handles the text for controls on the calendar control, the back button and next button</td>
</tr>
<tr>
<td>clsControlTextList.vb</td>
<td>Visual Basic Class</td>
<td>This class handles the listing functions of the controls in the admin pages</td>
</tr>
<tr>
<td>Class</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>clsEventSortedList.vb</td>
<td>Visual Basic Class This class handles the sorting of events.</td>
<td></td>
</tr>
<tr>
<td>clsRecurEvents.vb</td>
<td>Visual Basic Class This class handles a specific event on the calendar.</td>
<td></td>
</tr>
<tr>
<td>clsRecurEventList.vb</td>
<td>Visual Basic Class This class handles a group of events. Used by the calendar to get the specific dates in the given date range. The admin controls use this for getting all the events for manipulation</td>
<td></td>
</tr>
<tr>
<td>clsRecurTypes.vb</td>
<td>Visual Basic Class The class loads the list of recurrence types from the database. Used in the admin controls</td>
<td></td>
</tr>
<tr>
<td>clsStyle.vb</td>
<td>Visual Basic Class The class holds information for a specific CSS style that is applied to the calendar.</td>
<td></td>
</tr>
<tr>
<td>clsStyleList.vb</td>
<td>Visual Basic Class Loads the list of styles objects (clsStyle.vb) used by the Calendar.</td>
<td></td>
</tr>
<tr>
<td>CurrentName.vb</td>
<td>Visual Basic Class This class handles the retrieval of the User's full name according to its personal number</td>
<td></td>
</tr>
<tr>
<td>Dbo.CreateUser</td>
<td>Stored Procedure Adds new users.</td>
<td></td>
</tr>
<tr>
<td>Dbo.Delete_Selected_Event</td>
<td>Stored Procedure Deletes the booked shifts from the available list.</td>
<td></td>
</tr>
<tr>
<td>Dbo.Deleteuser</td>
<td>Stored Procedure Deleting users from the system.</td>
<td></td>
</tr>
<tr>
<td>Dbo.MakeAnnouncemnet</td>
<td>Stored Procedure Make new announcements and save it to database.</td>
<td></td>
</tr>
<tr>
<td>Dbo.MessageInsert</td>
<td>Stored Procedure Send message through the messaging.</td>
<td></td>
</tr>
<tr>
<td>Dbo.NameOfCurrentUser</td>
<td>Stored Procedure Gets the current full name according to the personal number.</td>
<td></td>
</tr>
<tr>
<td>Dbo.Ups_calendar_Delete</td>
<td>Stored Procedure Deletes a calendar component from the database.</td>
<td></td>
</tr>
<tr>
<td>Dbo.Usp_calendar_GetAll</td>
<td>Stored Procedure Retrieves calendar component.</td>
<td></td>
</tr>
<tr>
<td>Stored Procedure</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>Dbo.Usp_calendar_Save</td>
<td>Saves a new calendar component</td>
<td></td>
</tr>
<tr>
<td>Dbo.Usp_Controltext_GetAll</td>
<td>Retrieves all control text values from the database.</td>
<td></td>
</tr>
<tr>
<td>Dbo.Usp_Controltext_Save</td>
<td>Saves control text into database.</td>
<td></td>
</tr>
<tr>
<td>Dbo.Usp_RecurEvents_by_date_range</td>
<td>Retrieves events by date</td>
<td></td>
</tr>
<tr>
<td>Dbo.Usp_RecurEvents_Delete</td>
<td>Delete events</td>
<td></td>
</tr>
<tr>
<td>Dbo.Usp_RecurEvents_LoadAll</td>
<td>Loads all events regardless of date</td>
<td></td>
</tr>
<tr>
<td>Dbo.Usp_RecurEvents_Save</td>
<td>Save events</td>
<td></td>
</tr>
<tr>
<td>Dbo.Usp_RecurTypes_load</td>
<td>Loads different recurring types</td>
<td></td>
</tr>
<tr>
<td>Dbo.Usp_Style_by_control</td>
<td>Loads style of calendar control</td>
<td></td>
</tr>
<tr>
<td>Dbo.Usp_Style_GetAll</td>
<td>Retrieves calendars control styles</td>
<td></td>
</tr>
<tr>
<td>Dbo.Usp_Style_Save</td>
<td>Save calendar control styles</td>
<td></td>
</tr>
<tr>
<td>Dbo.Usp_text_by_Control</td>
<td>Loads the text of controls</td>
<td></td>
</tr>
<tr>
<td>Dbo.Announcements</td>
<td>SQL Table</td>
<td>Holds announcement data</td>
</tr>
<tr>
<td>Dbo.Calendars</td>
<td>SQL Table</td>
<td>Holds calendar data</td>
</tr>
<tr>
<td>Dbo.ControlText</td>
<td>SQL Table</td>
<td>Holds Text of calendar’s control data</td>
</tr>
<tr>
<td>Dbo.Messaging</td>
<td>SQL Table</td>
<td>Holds messaging data</td>
</tr>
<tr>
<td>Dbo.RecurEvents</td>
<td>SQL Table</td>
<td>Holds information of Events</td>
</tr>
<tr>
<td>Dbo.RecurTypes</td>
<td>SQL Table</td>
<td>Holds types of recurrence</td>
</tr>
<tr>
<td>Dbo.Styles</td>
<td>SQL Table</td>
<td>Holds information about styles</td>
</tr>
<tr>
<td>Dbo.UserInfo</td>
<td>SQL Table</td>
<td>Holds User Information (except the log in information)</td>
</tr>
</tbody>
</table>
E/R Diagram
The E/R Diagram below describes the relation between the OSS database and the ASPNET database. ASPNET database is responsible for holding and managing the user’s membership information and their access (roles).
8. Detailed design

8.1. Architectural Strategies:
Use of a particular type of product:

- Programming language: ASP.Net 3.5 is being used for our project, since it is a web application. Visual Basic 2008 is also being used for the code behind ASP.NET web forms. Ajax is also implemented in our project as extenders.

- Database: Data Sources and data sets that are based off SQL server 2005 databases are being used to store the information.

- Visual Studio 2008 Team System: We are using Visual Studio 2008 that is the official Microsoft IDE that offers an environment to work with .NET Languages.

Detailed Design Description:

1) Root Pages: (../~)
8.1.1. Basic.css
Type: Cascading Style Sheet
Purpose: To maintain the same style through the whole web application

8.1.2. Masterpage.Master
Type: ASP.NET Master Page
Purpose: To maintain the same template though the whole web application

8.1.3. Masterpage.master.vb
8.1.4. Type: Partial class which is the code behind Masterpage.Master
Purpose: To code the master page with Visual Basic code and adds functionalities to the master page.

8.1.5. Default.aspx
Type: ASP.NET Web Form
Purpose: This is the page where the users have to input their login information

![Default.aspx](image)

Fig: Default.aspx

8.1.6. Default.aspx.vb
Type: Partial class which is the code behind default.aspx
Purpose: Holds the VB code for functionalities of Login.

8.1.7. Errorpage.aspx:
Type: ASP.NET Web Form
Purpose: Error page for redirection at the time of error and exception occurrence.
8.1.8. **Errorpage.aspx**
Type: Partial class which is the code behind errorpage.aspx
Purpose: VB code for error page.

8.1.9. **Login.aspx**
Type: ASP.NET Web Form
Purpose: To redirect the user after logging out from the application

8.1.10. **Login.aspx.vb**
Purpose: Partial class which is the code behind login.aspx
Type: VB code for redirection.

8.1.11. **Passwordrecovery.aspx**
Type: ASP.NET Web Form
Purpose: Holds control for recovery of forgotten password.

8.1.12. **Passwordrecovery.aspx.vb**
Type: Partial class which is the code behind Passwordrecovery.aspx
Purpose: Holds the code to recover the forgotten password.

8.1.13. **Default.aspx**
Type: ASP.NET Web Form
Purpose: This the home page for the workers. When they login they will be redirected to this page. The main purpose of this page is to display the announcements.
8.1.14. **Default.aspx.vb**
Type: Partial class which is the code behind default.aspx
Purpose: Contains the code that display the announcements.

8.1.15. **Changepassword.aspx**
Type: ASP.NET Web Form
Purpose: To change the password.

8.1.16. **Changepassword.aspx.vb**
Type: Partial class which is the code behind changepassword.aspx
Purpose: VB code to change the password

8.1.17. **Compose.aspx**
Type: ASP.NET Web Form
Purpose: Allows the users to send messages
8.1.18. **Compose.aspx.vb**
Type: Partial class which is the code behind compose.aspx
Purpose: VB code to send messages.

8.1.19. **CoSchedule.aspx**
Type: ASP.NET Web Form
Purpose: To view the company's schedule, available shifts and book a shift.

8.1.20. **CoSchedule.aspx.vb**
Type: Partial class which is the code behind CoSchedule.aspx
Purpose: VB code to display company's schedule, available shifts and book a shift.

8.1.21. **Inbox.aspx**
Type: ASP.NET Web Form
Purpose: To display a person’s message inbox.
8.1.22. Inbox.aspx
Type: Partial class which is the code behind inbox.aspx
Purpose: VB code to display message inbox.

8.1.23. Messaging.aspx
Type: ASP.NET Web Form
Purpose: This is the main page of messaging. This page holds all the links to Inbox, compose, and sent messages sections.

8.1.24. Messaging.aspx.vb
Type: Partial class which is the code behind messaging.aspx
Purpose: VB code for the messaging page.

8.1.25. MySchedule.aspx
Type: ASP.NET Web Form
Purpose: To display a person’s schedule with their shifts.

Type: Partial class which is the code behind MySchedule.aspx
Purpose: VB code to display a person’s schedule and their shifts.

8.1.27. Redirection.aspx

Type: ASP.NET Web Form
Purpose: To redirect users to their destination according to their membership and role.

8.1.28. Redirection.aspx.vb

Type: Partial class which is the code behind Redirection.aspx
Purpose: VB code to decide where each user should be redirected to.

8.1.29. Sent.aspx

Type: ASP.NET Web Form
Purpose: To display the sent messages of a user.

8.1.30. Sent.aspx.vb

Type: Partial class which is the code behind sent.aspx
Purpose: VB Code to display sent messages.

3) Admin Pages: (../Admin/*)

8.1.31. Default.aspx:
Type: ASP.NET Web Form
Purpose: This is the Manager’s home page, if the user has a manager role, this page will be loaded after logging in. This page is also called “manager tool”

Fig: ../Admin/Default.aspx

8.1.32. Default.aspx.vb
Type: Partial class which is the code behind default.aspx
Purpose: VB code to load the manager tools.

8.1.33. CreateNewUser.aspx
Type: ASP.NET Web Form
Purpose: To manage users. Add / edit / delete functions are being held here.

Fig: ../Admin/CreateNewUser.aspx – Create
8.1.34. **CreateNewUser.aspx.vb**

Type: Partial class which is the code behind createnewuser.aspx
Purpose: VB code to add / edit / delete users.

8.1.35. **MakeAnnouncements.aspx**

Type: ASP.NET Web Form
Purpose: To allow the manager make new announcements.
8.1.36. MakeAnnouncements.aspx.vb
Type: Partial class which is the code behind MakeAnnouncements.aspx
Purpose: VB code allowing the manager to make new announcements.

8.1.37. ManageShifts.aspx
Type: ASP.NET Web Form
Purpose: To manage shifts. Create / edit / delete / assign shifts are being held here.

Fig: ../Admin/ManageShifts.aspx – Create + Edit + Delete

Fig: ../Admin/ManageShifts.aspx – Shift Assignment Tool – Available shifts
8.1.38. **ManageShifts.aspx.vb**

Type: Partial class which is the code behind ManageShifts.aspx

Purpose: VB code to Create / edit / delete / assign shifts.

8.1.39. **ManageStyles.aspx**

Type: ASP.NET Web Form

Purpose: To manage the styles of calendar components
8.1.40. ManageStyles.aspx.vb
Type: Partial class which is the code behind ManageStyles.aspx
Purpose: VB code to manage and change the styles of Calendar components

4) Custom Made Controls :

8.1.41. AdminAssignShifts.ascx
Type: ASP.NET User Control
Purpose: Holds all functionalities that is needed for the manager to assign a shift for a worker.

8.1.42. AdminAssignShifts.ascx.vb
Type: Partial class which is the code behind AdminAssignShifts.ascx
Purpose: VB code to assign shifts.

8.1.43. AdminManageContents.ascx
Type: ASP.NET User Control
Purpose: Enables the manager to change the content (label) of the calendar component.

8.1.44. AdminManageContents.ascx.vb
Type: Partial class which is the code behind AdminManageContent.ascx
Purpose: VB code to change and edit the contents

8.1.45. AdminManageEvents.ascx
Type: ASP.NET User Control
Purpose: To manage events. All functionalities that are needed for the manager to create / edit / delete a shift is being held here.

8.1.46. AdminManageEvents.ascx.vb
Type: Partial class which is the code behind AdminManageEvents.ascx
Purpose: VB code to create / Edit / Delete shifts.

8.1.47. AdminManageStyle.ascx
Type: ASP.NET User Control
Purpose: To manage and edit styles of calendar components.

8.1.48. AdminManageStyle.ascx.vb
Type: Partial class which is the code behind AdminManageStyles.ascx
Purpose: VB code to change and save styles of calendar components.

8.1.49.
8.1.50. Calendar.ascx
Type: ASP.NET User Control
Purpose: This is the calendar component. It loads the company’s schedule as well as the available shifts in the weekly view.

8.1.51. Calendar.ascx.vb
Type: Partial class which is the code behind Calendar.ascx
Purpose: VB code to load shifts into a weekly presentation.

5) VB Classes:
<Reference to Data Interface Section – all classes are described in that section>

6) Stored Procedures:

8.1.52. Dbo.CreateUser
Type: ADO.NET Stored Procedure
Purpose: To add a new user into the UserInfo data tables.
Parameters:
@Persno varchar(50),
@fullName varchar(100),
@HomePhone varchar(50),
@CellPhone varchar(50),
@Address varchar(50),
@ZipCode varchar(50),
@City varchar(50),
@Bank varchar(50),
@AcNo varchar(50),
@EmployeeSince varchar(50)

8.1.53. Dbo.Delete_selected_Event
Type: ADO.NET Stored Procedure
Purpose: To delete the selected shift from the available shifts at the time of booking.
Parameters:
@EID int

8.1.54. Dbo.DeleteUser
Type: ADO.NET Stored Procedure
Purpose: To delete a user from UserInfo data table.
Parameters:
@Persno varchar(50)

8.1.55. Dbo.MakeAnnouncement
Type: ADO.NET Stored Procedure
Purpose: To save a new announcement into announcement table.
Parameters:
@Announcement ntext,
@Subject varchar(225),
@Maker varchar(100),
@aDate datetime,
@MakerPersNo varchar(50)

8.1.56. Dbo.MessageInsert
Type: ADO.NET Stored Procedure
Purpose: To save a new message into the messaging table.
Parameters:
@subject varchar(200),
@sender varchar(100),
@reciept varchar(100),
8.1.57. **Dbo.NameOfCurrentUser**  
*Type:* ADO.NET Stored Procedure  
*Purpose:* To retrieve the full name of current user from UserInfo.  
*Parameters:*  
@personalNumber varchar(50)

8.1.58. **Dbo.Usp_calendars_Delete**  
*Type:* ADO.NET Stored Procedure  
*Purpose:* To delete an existing calendar component.  
*Parameters:*  
@pID int

8.1.59. **Dbo.Usp_calendars_GetAll**  
*Type:* ADO.NET Stored Procedure  
*Purpose:* To retrieve all calendar components.  
*Parameters:*  
N/A

8.1.60. **Dbo.Usp_calendars_save**  
*Type:* ADO.NET Stored Procedure  
*Purpose:* To save a new calendar component.  
*Parameters:*  
@pID int  
@pName varchar(255)

8.1.61. **Dbo.Usp_controltext_GetAll**  
*Type:* ADO.NET Stored Procedure  
*Purpose:* To retrieve calendar control’s text value.  
*Parameters:*  
N/A

8.1.62. **Dbo.Usp_controltext_save**  
*Type:* ADO.NET Stored Procedure  
*Purpose:* To save a new calendar control’s text value.  
*Parameters:*  
@pID int  
@pValue varchar(255)

8.1.63. **Dbo.Usp_recurEvents_by_dateRange**  
*Type:* ADO.NET Stored Procedure  
*Purpose:* To retrieve the events by date range according to the calendar component.  
*Parameters:*  
@pCalendar int  
@pStartDate datetime  
@pEndDate datetime

8.1.64. **Dbo.Usp_recurEvents_Delete**  
*Type:* ADO.NET Stored Procedure  
*Purpose:* To delete an existing event.  
*Parameters:*  
@pID int

8.1.65. **Dbo.Ups_recurEvents_LoadAll**  
*Type:* ADO.NET Stored Procedure
Purpose: To load all events.
Parameters: N/A

8.1.66. **Dbo.Usp_recurEvents_Save**
Type: ADO.NET Stored Procedure
Purpose: To save a new event.
Parameters:
@pID int
@pTitle varchar(255)
@pDescription text
@pStartTime datetime
@pEndTime datetime
@pStartDate datetime
@pEndDate datetime
@pRecurType int
@pCalendar int

8.1.67. **Dbo.Usp_recurTypes_load**
Type: ADO.NET Stored Procedure
Purpose: To load all different recurrence types.
Parameters: N/A

8.1.68. **Dbo.Usp_styles_by_control**
Type: ADO.NET Stored Procedure
Purpose: To load the style of calendar controls
Parameters:
@pControl varchar(255)

8.1.69. **Dbo.Usp_styles_GetAll**
Type: ADO.NET GetAll
Purpose: To retrieve all style attributes of calendar component
Parameters: N/A

8.1.70. **Dbo.Usp_styles_save**
Type: ADO.NET Stored Procedure
Purpose: To save a new style attribute for the calendar component.
Parameters:
@pID int
@pValue varchar(255)

8.1.71. **Dbo.Usp_text_by_control**
Type: ADO.NET Stored Procedure
Purpose: To load the text value of calendar control.
Parameters:
@pControl varchar(255)

7) SQL Data Tables
<Reference to Data Interface Section – All tables are described>
<Reference to E/R Diagram – Attributes are described>
8.2. Class Diagram

The class diagram below describes the classes that are being used in the web application. These classes are only the data interface classes. Partial classes (something.aspx.vb) are NOT assumed as classes in .NET framework, so they can’t be described in the class diagram. In this web application classes are not related to each other, each class handles certain functions and method within itself.

![Class Diagram]

Fig: Class Diagram

9. Policies and tactics

In this section we discuss the different policies/tactics that would not significantly affect the overall organization of the system, but which might have a significant effect on the implementations and interface of certain aspects of our program.

9.1. Specific “Products” Used

The compiler used to generate our software is Microsoft’s Visual Studio 2008 in which the ASP.net website compiler is used. Since our software is online based for easier accessibility, ASP.net with Visual Basic programming language will be used. To handle the database, Microsoft SQL Server 2005 will be used.

9.2. Plans for testing the software

Since our software is mostly going to be user orientated, which means that ease of use for the user is one of our main priorities, there is going to be a lot of testing in this department. The heuristic evaluation will be used in order to make sure that the user will have an easy time navigating around. Another testing plan would be having a couple of staff members from the company test and evaluate our program. By doing this, we can see what our main faults are and hence change the program according to what they think is good or bad.