

WEB BASED MEETING SCHEDULER

**A Paper
Submitted to The Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science**

By

Mayukh Sharma

**In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE**

**Major Program:
Software Engineering**

May 2010

Fargo , North Dakota

North Dakota State University
Graduate School

Title

WEB BASED MEETING SCHEDULER

By

MAYUKH SHARMA

The Supervisory Committee certifies that this *disquisition* complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

North Dakota State University Libraries Addendum

To protect the privacy of individuals associated with the document, signatures have been removed from the digital version of this document.

ABSTRACT

Sharma, Mayukh, M.S., Department of Computer Science , College of Science and Mathematics , North Dakota State University , May 2010. The Web Based Meeting Scheduler. Major Professor : Dr. Kendall E. Nygard.

In this paper I present a Web based Meeting Scheduler which is a web-based easy to use group calendar application for scheduling events, reserving meeting rooms and scheduling meetings. Meetings and events may be scheduled for invitees within an organization as well as prospective customers and clients from outside the organization by adding the following website (www.meetingscheduler.org) to the address bar of a browser.

Any user with an account on the system can choose a meeting date from the calendar and provide the subject for the meeting, a description, the location and any specific conference room information. A meeting may be set up for one or more dates and include multiple time frames from available time slots. A user selects the attendees, enters the RSVPs and sends the invitations to mail boxes with a verification link. A user may also share files including the agenda with the attendees.

This Web Based Meeting Scheduler is developed on a modular Internet/Intranet application framework. It features a scheduler, meeting calendar, messaging, an address book, file upload and download and feedback. The system employs an easy to intuit “wizard like” interface to encourage online collaboration.

The objective of Web Based Meeting Scheduler is to provide users a web based tool with effective planning and scheduling of a meeting, provide monitoring of the meeting by an organizer, and promote effective use of an online system to communicate and collaborate with the participants/invitees of the meeting.

This Web Based Meeting Scheduler application allows the building of collective event calendars by providing the following features:

- Calendar view with bold marking of dates with events ;
- 3 month quick planning view with the ability to schedule events far into the future;
- Unlimited number of attached files tied to an event;
- Events that may span multiple days;
- User based control for creating, reading and updating file attachments;

This Web Based Meeting Scheduler consists of a login screen from which the users and administrator logon to the system. In this System, the users are assigned roles. Each role is associated with a given set of services. Various roles include the administrator, organizers, and invitees. An administrator may add other roles to the system as needed. The administrator is responsible for user management and system administration.

This Web Based Meeting Scheduler (www.meetingscheduler.org) process starts with users registering on the system and the administrator assigning roles to users. Once users are registered, the system allows an organizer to set up a meeting and maintain all meeting related activities such as adding invitees, sending invitations, uploading and downloading the files, and tracking invitation response and feed back by the invitees.

ACKNOWLEDGMENTS

I would like to thank Dr. Kendall Nygard for his continued support, help, and direction. I would like to thank my family and friends who supported and motivated me to accomplish this task.

TABLE OF CONTENTS

ABSTRACT.....	iii
ACKNOWLEDGMENTS	v
LIST OF TABLES.....	x
LIST OF FIGURES	xi
1. INTRODUCTION	1
2. RELATED WORK	3
3. SYSTEM ANALYSIS.....	4
3.1. Existing System.....	4
3.2. Drawbacks.....	4
3.3. Proposed System	5
3.4. Advantages.....	5
4. HARDWARE AND SOFTWARE REQUIREMENTS	7
4.1. Hardware Requirements.....	7
4.2. Software Requirements.....	7
5. FEASIBILITY STUDY	9
5.1. Economic Feasibility.....	9
5.2. Operational Feasibility	9
5.3. Technical Feasibility	10
6. FUNCTIONAL & NON-FUNCTIONAL REQUIREMENTS	11
6.1. Functional Requirements	11

6.2.	Non-Functional Requirements	12
6.2.1.	Constraints	13
6.2.2.	Guidelines	14
7.	EXECUTION METHODOLOGY	15
7.1.	Requirements Analysis & Process	15
7.2.	System Requirements Specifications	16
7.3.	Design and Development	16
7.4.	Testing & Debugging	16
7.5.	Implementation	16
8.	PHYSICAL DESIGN	17
8.1.	UML Diagrams	17
8.2.	Sequence Diagrams	20
8.3.	Activity Diagram	23
8.4.	Class Diagram	25
9.	INTERNAL VIEW	26
9.1.	WebSchedulerEvents Class	26
9.2.	Invitee Class	28
9.3.	DAL Class	28
10.	DATA DICTIONARY	34
10.1.	Data Modeling Overview	34
10.2.	Methodology	35
10.3.	Data Modeling in the Context of Database Design	35

10.4. Components of a Data Model	36
10.5. Why is Data Modeling Important?.....	36
10.6. Summary	37
11. TABLES	38
12. TECHNOLOGICAL REQUIREMENTS.....	49
12.1. HTML	49
12.2. Attributes.....	51
12.3. JavaScript.....	52
12.4. .NET Technology.....	54
12.5. Reasons to use the .NET Framework.....	55
13. EXTERNAL VIEW	61
13.1. Problem Statement	61
13.2. Web Meeting Application Design.....	61
13.3. The Application UI	62
13.4. Organizer's UI.....	62
13.5. Create New Event	64
13.6. View Details of Existing Events	66
13.7. Invitee's UI.....	67
13.8. Admin UI	68
14. TESTING AND DEBUGGING	70
14.1. Testing Process-an Overview.....	70
14.2. Application is Tested in the Following Aspects	75

15. CONCLUSIONS	76
16. REFERENCES	77

LIST OF TABLES

<u>Table</u>	<u>Page</u>
1. Functional Requirements.....	12
2. Constraints.....	13
3. Guidelines.....	14
4. Aspnet_Applications.....	38
5. Aspnet_Membership.....	39
6. Aspnet_Roles.....	40
7. Aspnet_Users.....	41
8. Aspnet_UsersInroles.....	42
9. EventDateTimes.....	42
10. EventFiles.....	43
11. EventInvitee.....	44
12. EventInviteeResponse.....	45
13. Events.....	46
14. Events_History.....	47

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
1. Use Case Diagram	19
2. Sequence Diagram1	20
3. SequenceDiagram2	21
4. Sequence Diagram3	22
5. Activity Diagram	24
6. Class Diagram.....	25
7. User Interface Diagram.....	61
8. Web Scheduler Calendar	63
9. Web Scheduler Event Creation Page	64
10. Web Scheduler Event Invitee Page.....	65
11. Web Scheduler Event Details View Page.....	66
12. Web Scheduler Response Page.....	68
13. Web Scheduler Administrators Page	69

1. INTRODUCTION

This Web Based Meeting Scheduler is an easy-to-use group calendar application for scheduling meeting rooms, reserving meeting rooms and scheduling meetings. From the intuitive web interface, a user may schedule meeting rooms, search for available rooms, confirm attendance and reserve meeting equipment.

The group calendars are an important means of event divulgation and resource sharing. Shared resources include meeting rooms, conference rooms, and projection equipment. These tools are an extension of many of the features provided by time management software such as desk accessory packages and computer office automation systems.

The schedule view gives users a choice of multiple calendar views, so they can view daily, weekly or monthly availability at a glance, and schedule meetings quickly and effectively. This Web Based Meeting Scheduler is a web base application that can run on a shared server. All user's and meeting data are stored on the server so the user can access and view their data / meeting details from a simple browser via the website over the internet (www.meetingscheduler.org).

The software may be a local package designed for individual use. This is similar to products such as the Lightning extension for Mozilla Thunderbird [02], Microsoft Outlook without Exchange Server [03], or Windows Calendar [03]. The system may be a networked package that allows for the sharing of information between users similar to Mozilla Sunbird [02], Windows Live Calendar [03], Google Calendar [05], or Microsoft Outlook with Exchange Server [03].

A meeting event is characterized by the following data attributes:

- i. *Meeting Subject* : The subject of the meeting
- ii. *Meeting Place* : The location of the meeting
- iii. *Meeting Date* : One (or multiple) dates of the meeting
- iv. *Meeting Times* : One (or multiple) time intervals
- v. *Meeting Invitees* : A list of meeting invitees
- vi. *Meeting Files* : Any number of attached files on any meeting/event

2. RELATED WORK

There are many group calendar software packages available on the web. In this section, we discuss related web based scheduler and compare them with our web based meeting scheduler.

In concurrent work [05, 06], Doodle and Microsoft detailed the process model supported by them: 1) easy to use, 2) Convenient, 3) Shareable, 4) Flexible, 5) Secure. A browsing instance contains all interconnected (or inter-referenced) windows including tabs, frames and subframes *despite* their origins. A scheduler instance is a group of same –sites pages within a browsing instance.

Doodle from Google [05] uses the process of scheduler instance along with all the models mentioned at the beginning of this section by default.

All Process models used in Doodle or any other web based scheduler are insecure since they don't provide memory or other resource protection across multiple principals in a browser instance.

Our scheduler has overcome this issue and proves to be better in terms of performance, flexibility and convenience when it comes to schedule meeting online. Below are some examples of existing meeting schedulers.

- Google Calendar [05]
- Microsoft Exchange [03]
- Microsoft Outlook [03]
- Doodle [05]

3. SYSTEM ANALYSIS

System analysis is a basic activity in software development. Traditionally, it is carried out by a separate group of practitioners, who gather the system requirements, analyze them and prepare the specifications documents to be handed to the development group. In large-scale software projects, these activities are highly significant.

3.1. Existing System

Here the existing system is referring to other web-based meeting schedulers. The web-based design of existing scheduling tools results in slower performance across different time zones. Adding to that it also provides lesser security in terms of a system crash and recovery. The interfaces of meeting scheduling tools seem to be easier as compared with conventional calendaring software.

The greatest disadvantage afforded by existing meeting schedulers is lack of automation. The new Web Meeting Scheduling tool automatically updates calendars maximizing the coordination of new meeting and appointment times.

3.2. Drawbacks

The drawbacks of existing scheduling systems include:

- Lack of simplicity;
- Lack of role based authentication;
- Lack of options for rating or providing inputs;
- Doesn't allow the administrator to monitor overall activities of the meeting.

The drawbacks of existing systems lead to the present web based application called the Web Meeting Scheduler.

3.3. Proposed System

The proposed system is a web-based solution for the existing system. It is based on a modular Internet/Intranet application framework which provides superior co-ordination between the meeting organizer and end-user. In addition, it allows the administrator to effectively track activities over the website. It features a scheduler, a calendar of meetings, messaging, file upload and download and feedback. The system is designed to promote online collaboration.

3.4. Advantages

The key advantages of the new Web Meeting Scheduler are:

- Provide effective co-ordination between different areas regarding work schedules through the scheduler.
- Improve the quality in planning and managing work activities.
- Provide a simple to use and quick way of managing various accounts.
- Provide a facility for the administrator to track overall activities within the scheduler.
- Provide good communication channels for the organizer to interact with invitees.
- Provide upload and download facilities to share documents required for meetings.

- Provide a facility to collect feedback from invitees.
- Provide a facility for the invitees to rate an event on a scale of 1 to 5.

4. HARDWARE AND SOFTWARE REQUIREMENTS

Hardware and software availability affect the configuration of any web based meeting scheduler environment. The following specifications are the minimum hardware and software requirements for our Web Based Meeting Scheduler system. Since the languages used in the development is ASP.NET and Java, the system requirement are derived based on the facts provided by Microsoft [15] standard system requirement details.

4.1. Hardware Requirements

Minimum hardware requirements for running This Web Based Meeting Scheduler Server are as follows:

- Pentium IV processor
- 512 MB RAM.
- 10 GB Hard Disk Space.
- Ethernet card.

4.2. Software Requirements

Software needed to operate the system includes:

Database : Microsoft SQL Server 2005 or SQL Server Express 2005

Web Server : Microsoft Internet Information Server (IIS) 5.0 or above

Client : .NET Framework 3.0, ASP.NET 3.0/C# 3.0, AJAX Toolkit 2005,
HTML, Java Script, JQuery 1.3

Web Browser : Internet explorer IE 6.0 or higher, Firefox 2, Google Chrome.

5. FEASIBILITY STUDY

The feasibility study is an important phase in the software development process. It enables the developer's to have an assessment of the product being developed. It refers to the feasibility of the product in terms of outcomes of the product, the operational use and the technical support required for implementing it [06].

The feasibility study should be performed on the basis of various criteria and parameters. The various types of feasibility studies are:

- Economic Feasibility
- Operational Feasibility
- Technical Feasibility

5.1. Economic Feasibility

Economic feasibility refers to the benefits or outcomes that are derived from the product as compared to the total cost spent on developing the product [06]. If the benefits achieved by the new system are the same or less than the current system, then it is not feasible to develop the product. In the product we have developed, the expectation is to reduce the amount of time spent in organizing, planning and monitoring meeting schedules. Improvements in meeting scheduling will generate productivity savings within the company.

5.2. Operational Feasibility

Operational feasibility refers to the feasibility of the product to solve the business problem at hand [07]. Some products may work very well at design and implementation

but may fail in a production environment. It includes the study of human resources required to maintain and operate the system as well as identify their technical expertise. The new application will operate in most system environments with few anticipated problems since the application is being developed using java based languages.

5.3. Technical Feasibility

Technical feasibility refers to whether the software that is available in the market fully supports the present application. It studies the pros and cons of using particular software for the development and its feasibility. It also studies the additional training needed to be given to the users to make the application work. The anticipated training required to use This Web Based Meeting Scheduler is 1 hour. That will provide enough knowledge for a casual user to work with application. Since this application uses a web-based interface with “wizard like” functionality, anyone can access the application and start working/utilizing it.

6. FUNCTIONAL & NON-FUNCTIONAL REQUIREMENTS

Functional & Non- Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system is required to perform.

Functional requirements, in product development, is useful to distinguish between the baseline functionality necessary for any system to compete in that product domain, and *features* that differentiate the system from products, and from variants in any product development. Features may be additional functionality, or differ from the basic functionality along some quality attribute (such as performance or memory utilization).

Non-functional requirement include, but are not limited to **Performance, Reliability, Fault Tolerance, Frequency, and Priority**. Each requirement is expressed in natural language or an appropriate formalism.

6.1. Functional Requirements

The main purpose of functional requirements within the requirement specification document is to define all the activities or operations that take place in the system [09]. These are derived through interactions with the users of the system. Since the Requirements Specification is a comprehensive document & contains a lot of data, it has been broken down into different sections in this report.

General functional requirements arrived at the end of the interaction with the Users as shown in (Table 1) below. A more detailed discussion is presented in the upcoming topics, which talk about the Analysis & Design of the system.

Table 1. Functional Requirements.

Requirement #	Description
1.	The system allows an administrator to manage users and roles.
2.	The system allows a meeting organizer to prepare meeting schedules and send them to different users.
3.	The system allows a meeting organizer to post the meeting details which will be displayed for all the invitee's.
4.	The system allows a meeting organizer and invitee to share the documents using Upload and download facilities.
5.	The system allows an invitee to post their feedback on the meeting or event.
6.	The system allows a meeting organizer to view invitee feedback.
7.	The system allows a meeting organizer to send a message to other users or a group of user's at once.
8.	The system allows a meeting organizer to view all the meeting invites.

6.2. Non-Functional Requirements

The non-functional requirements consist of constraints that the system is required to operate under and guidelines for improving the usability of the system [10].

6.2.1. Constraints

These are the requirements that are not directly related to the functionality of the system [10]. These should be considered as mandatory when the system is developed. The following Constraints as shown in (Table 2) were identified for our Web Based Meeting Scheduler.

Table 2. Constraints

Constraint	Description
1.	The system should be available over the intranet and internet so that the administrator & user can use the system from their respective locations.
2.	For gaining entry into the system, the user should be registered by an administrator and should be able access the system using a login id and password.
3.	The users should be able to change their passwords for increased security.
4.	The system should be easy to understand and organized in a structured way. The users should also receive appropriate messages about any errors that occur.
5.	The system should run readily available hardware. System owners should not have to purchase any specialized hardware beyond what is required to run a typical server based application.
6.	Data integrity within the system should be maintained if an error occurs or the system comes down.

6.2.2. Guidelines

Guidelines consist of non-mandatory requirements for the system. The requirements in this section are suggestions and recommendations to further enhance the usability of the system. Guidelines for This Web Based Meeting Scheduler as shown in (Table 3) are as follows:

Table 3. Guidelines

Guideline	Description
1.	The system should display a menu for users to choose from.
2.	The system should display users' requests in a reasonable time.
3.	Services of the system should be available 24 hours a day.
4.	The system should be designed in such a way that it is easy to enhance it with more functionality. It should be scalable and easily maintainable.

7. EXECUTION METHODOLOGY

A systematic approach to project execution is imperative for assuring desired project outcome. The approach must take into account schedule impacting issues that arise in every project. In this project, the execution methodology is the cornerstone of the success.

The development of Web based meeting scheduler application was purely based on software driven methods. There was no involvement of any human , animal or biohazards subjects in any phase of the application development or execution methodologies.

The implementation of the project is broken down into five different phases. The following is a high level description of the phases:

Phase 1 - Business Process & Requirements analysis

Phase 2 - System Requirements Specifications

Phase 3 - Design and Development

Phase 4 - Testing & Debugging

Phase 5 - Implementation

7.1. Requirements Analysis & Process

Business Process & Requirements analysis is the phase when the relevant business area is studied in detail [11]. This process brings out the gaps between the existing systems and identifies the areas where the business operations should be modified, keeping in view the way it needs to be carried out to encounter the problems during this phase the required documents are prepared defining the existing and required setup for the project [11].

7.2. System Requirements Specifications

The information about the requirements is collected which contains the information about the current user system and the proposed system as seen from the user perspective. At the end of this phase, a detailed requirement specification document is prepared and approved.

7.3. Design and Development

In this phase, the framework for the design of the proposed product is designed to meet the requirements specifications documented. The product is developed as per the framework to meet the objectives of the system requirement specifications approved.

7.4. Testing & Debugging

This phase contains the preparation of test cases and the standards of testing. The end users using the dummy User Ids carry out the testing.

7.5. Implementation

The project enters the implementation phase when the product is ready to be implemented or piloted on the production environment. After thorough training to all the end users, the product is implemented.

8. PHYSICAL DESIGN

The physical aspects of any information resources are well understood by developers; e.g., computer hardware/software, DBMS files, manual files, the physical components describe "how" a logical design be implemented. For example, companies have been implementing licensing systems for many years prior to the advent of the computer. We have seen licensing implemented manually, then with time clocks and punch cards, followed by automation on mainframes, minis and PC's. All of these devices are simply a physical variation of a theme. It's a lot like music; the composition represents the logical and the orchestration represents the physical.

The physical design is ultimately based on available technology and, as such, changes dynamically. In contrast, the logical side represents the inherent nature of a project and only changes if the requirement changes, which is considerably less volatile than the physical.

8.1. UML Diagrams

Unified Modeling Language (UML) is a standardized general purpose language used in the field of software engineering [10]. The standard is managed and was created by the Object Management Group [10]. The following UML diagram shows the key interfaces and processes in the Web Meeting Scheduler.

Use Case Diagram:

The **use case diagram** shown in (Figure 1) below provides a graphical overview of the functionality provided by This Web Based Meeting Scheduler system. The diagram shows different system functions performed by each actor. The actors of the system are Administrator and User. The user of a system can perform all system functions that an administrator can except for User Management and Website Management.

A **use case diagram** in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

UML **use case diagram** can be used to describe the functionality of a system in a horizontal way. That is, rather than merely representing the details of individual features of your system, UCDs can be used to show all of its available functionality. It is important to note, though, that UCDs are fundamentally different from sequence diagrams or flow charts because they do not make any attempt to represent the order or number of times that the systems actions and sub-actions should be executed.

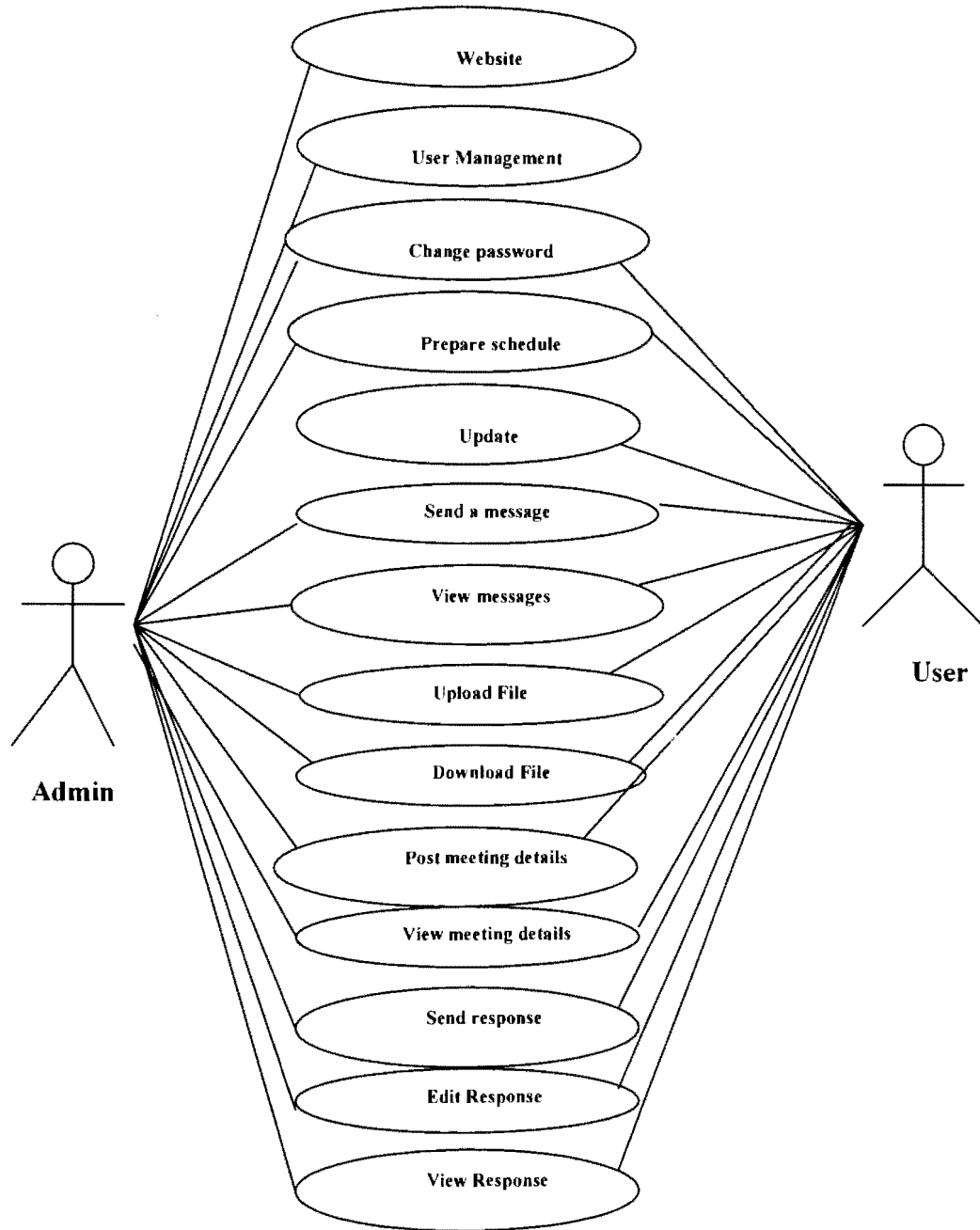


Figure 1: Use Case Diagram is a UML graphical overview of the functionality provided by our Web Based Meeting Scheduler system.

8.2. Sequence Diagrams

The sequence diagram below represents events and actions between the objects or different components of the system.

Sequence Diagram 1:

The Administrator of a System can assign roles, view details, add, edit and delete a user. Below shown (Figure 2) describes various roles as mentioned earlier.

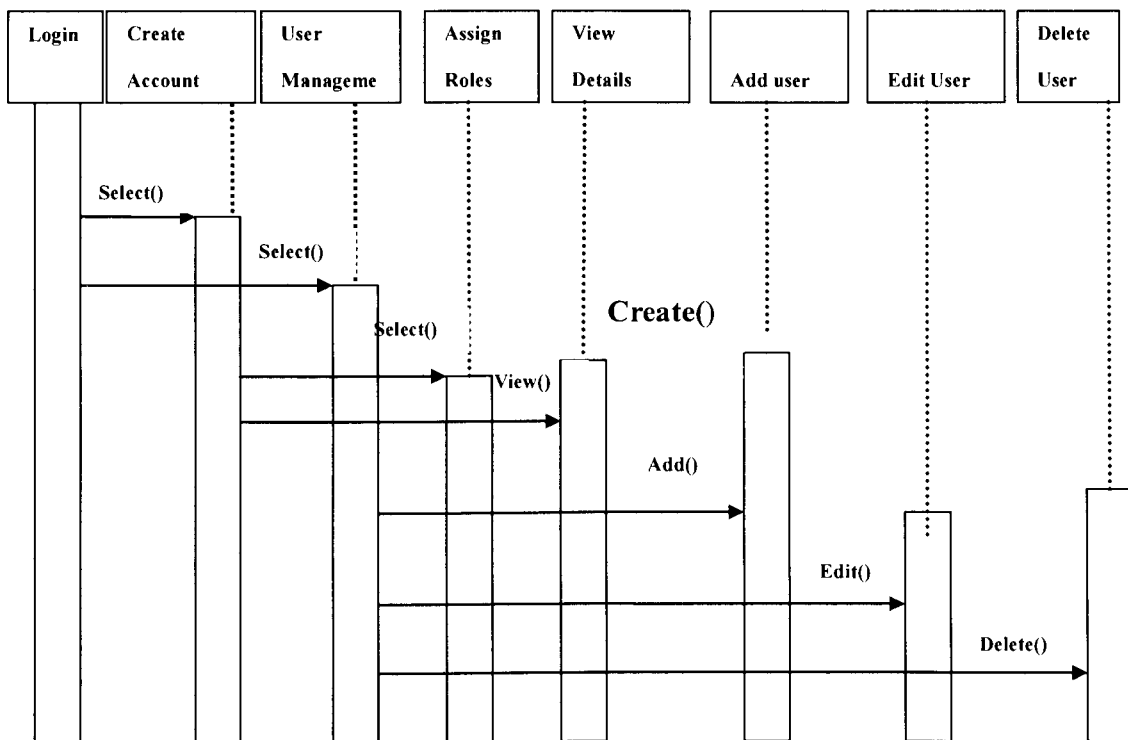


Figure 2: Sequence Diagram 1 is a Sequential representation of administrator's roles.

Sequence Diagram 2

The authorized users of the system can Create tasks, view tasks and send emails to all the attendees. User ability to perform various can be seen in (Figure 3) as shown below.

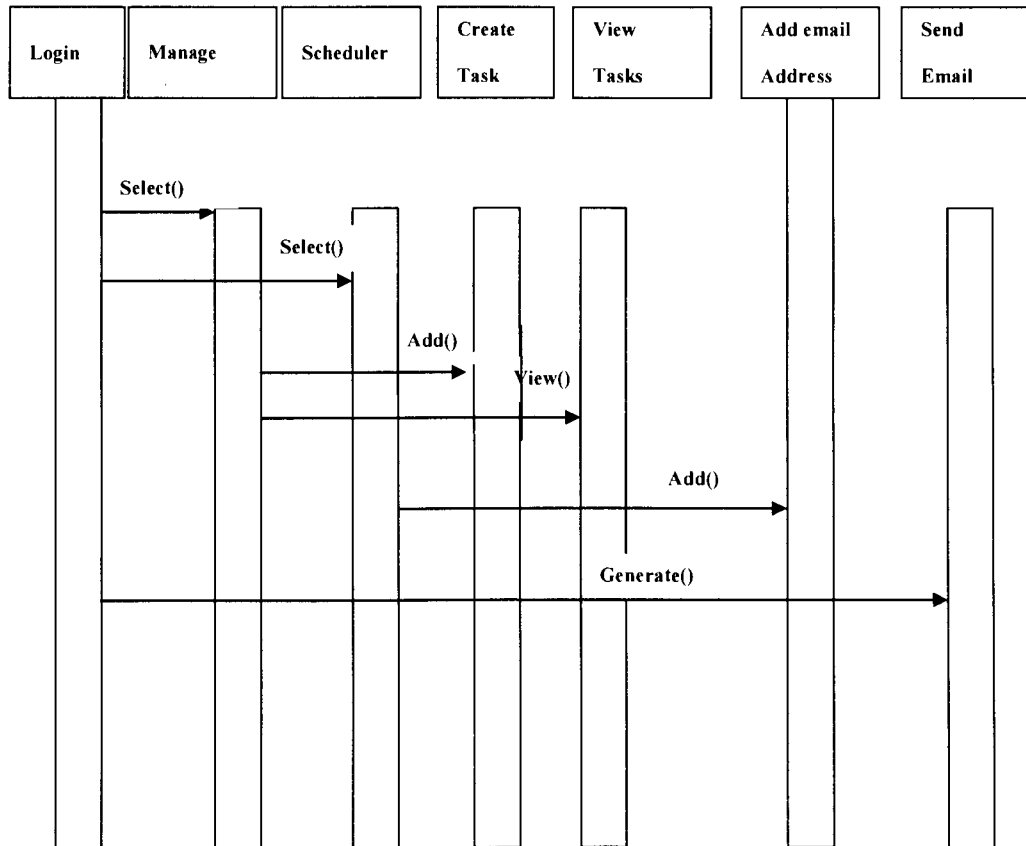


Figure 3: Sequence Diagram 2 represents activity and user roles.

Sequence Diagram 3:

The authorized users of the system can manage contacts as well as manage, upload and download files. Below mentioned (Figure 4) describes users activity and functions.

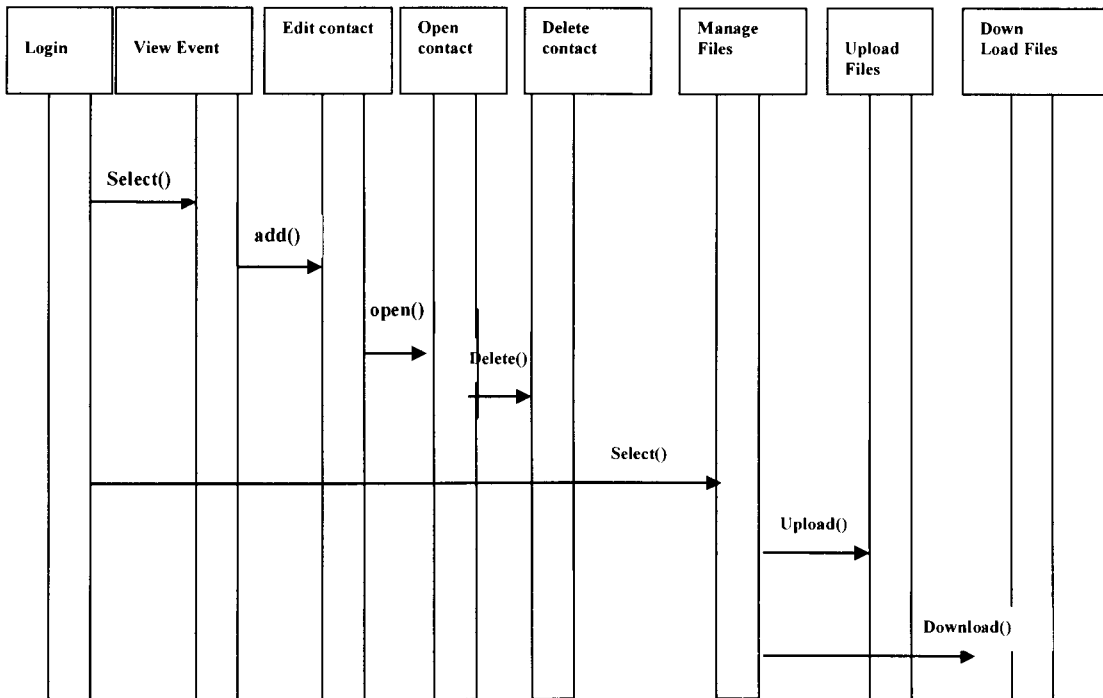


Figure 4: Sequence Diagram 3 explains sequence diagram representation of user roles .

8.3. Activity Diagram

The diagram (Figure 5) below gives an operational workflow of the components involved in This Web Based Meeting Scheduler system. The diagram shows the behavior of the system by outlining the sequence of activities performed.

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

Activity diagrams should be used in conjunction with other modeling techniques such as interaction diagrams and state diagrams. The main reason to use activity diagrams is to model the workflow behind the system being designed. Activity Diagrams are also useful for analyzing a use case by describing what actions need to take place and when they should occur; describing a complicated sequential algorithm and modeling applications with parallel processes.

Activity diagrams represent the business and operational workflows of a system. An Activity diagram is a dynamic diagram that shows the activity and the event that causes the object to be in the particular state. An Activity diagram talks more about these transitions and activities causing the changes in the object states.

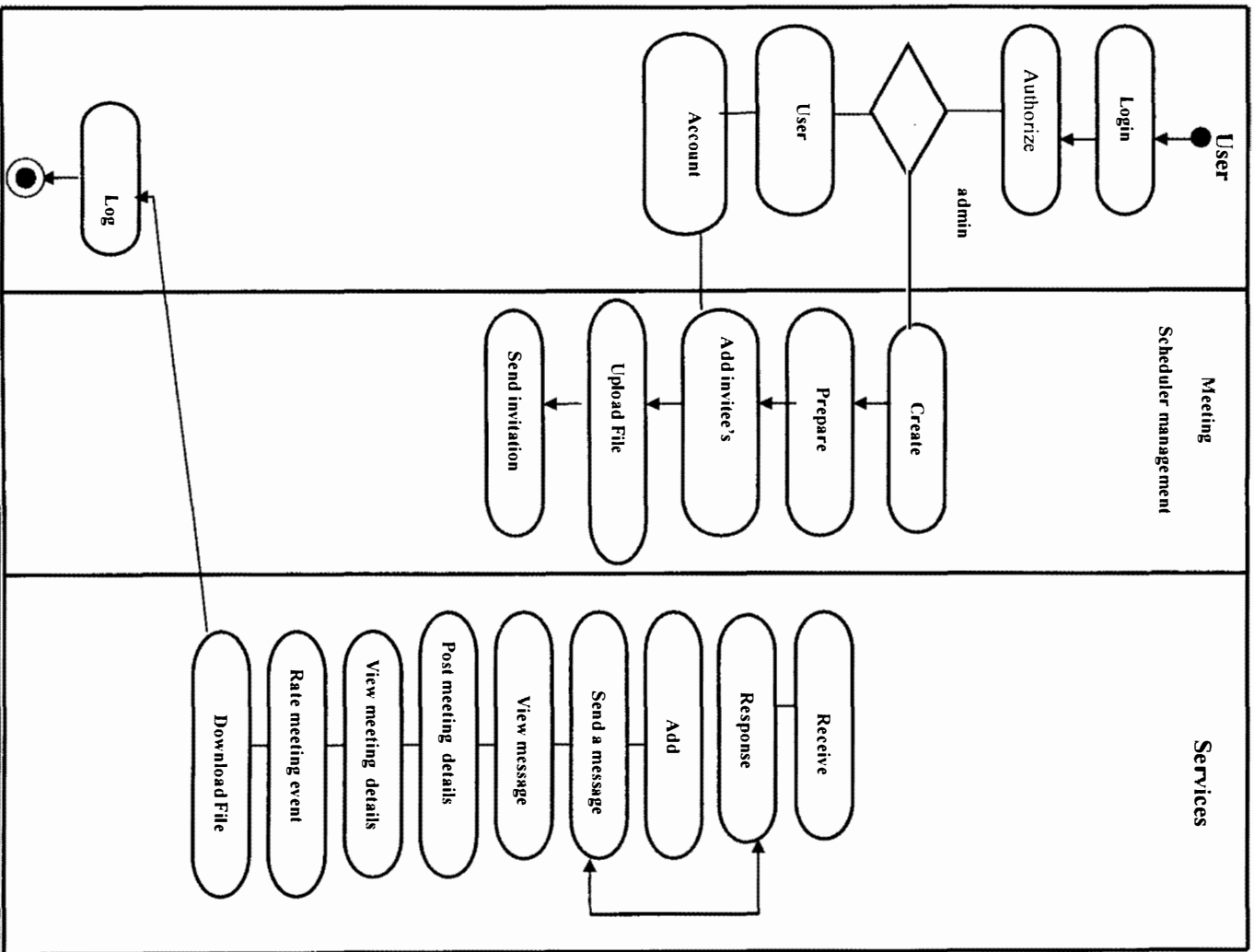


Figure 5: Activity Diagram explains operational workflow of the components involved in this Web Based Meeting Scheduler system.

8.4. Class Diagram

The diagram (Figure 6) below shows the classes and the operations of our Web Based Meeting Scheduler system.

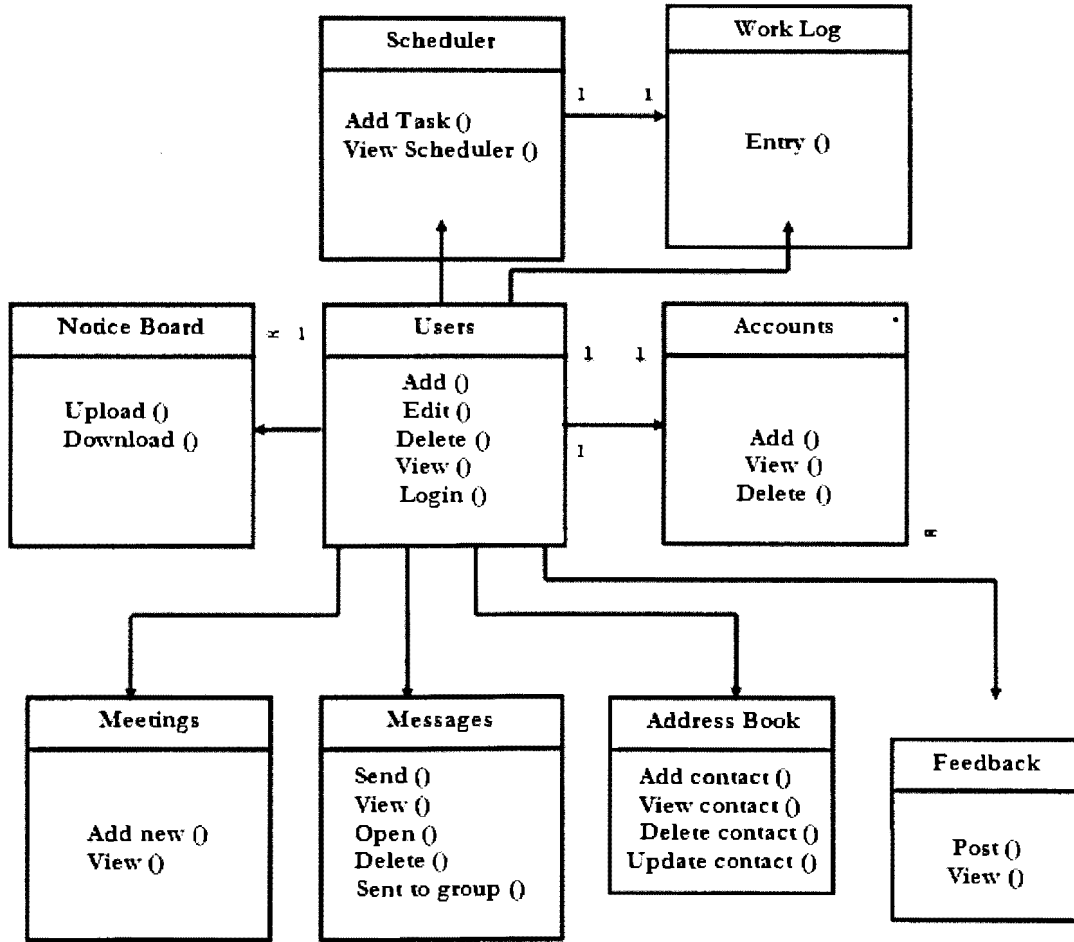


Figure 6: Class Diagram shown above describes various classes and the operations performed by them.

9. INTERNAL VIEW

In this section, we describe the sample problem with which we work with and walk through the class relationships and the key implementation code

9.1. WebSchedulerEvents Class

This class contains the methods necessary to perform all functionalities related to a meeting schedule.

The Code description of the WebSchedulerEvents Class methods is given below.

1) GetAllEvents()

This method is used to get the all events.

2) GetAllEventsByUser()

This method is used to get the all events by the user(organizer)

3) GetAllEventsWithDateAndTimesByUser()

This method is used to get the all events with date and time by user (organizer)

4) GetAllEventsDetailsWithInviteesResponseByUser()

This method is used to get the event details with the invitee response.

5) GetAllEventsInviteesByEvent()

This method is used to get the event's invitees.

6) GetOrganizerByEvent()

This method is used to get the organizers for all events.

7) IsEventDataExpired()

This method is used to check that the event date is expired or not.

8) SaveEvent()

This method is used to save the events

9) UpdateEvents()

This method is used to update the existing event.

10) SaveEventDataAndTime()

This method is used to save the date and time frame of the event.

11) SaveEventInvittess()

This method is used to save the event invitees and their details.

12) SaveEventFileDetails()

This method is used to save the details of the uploaded files of the event.

13) SaveEventOrganizerDateTimeResponse()

This method is used to save the organizer response and rating about the event date and times.

14) GetAllEventsFilesByEvent()

This method is used to get the event files for download.

15) GetEventDetailByEventID()

This method is used to get the details of given event.

16) DeleteInviteeById()

This method is used to delete the invitee of the given event.

17) DeleteEventFileById()

This method is used to delete the related file(s) of the given event.

18) DeleteEventById()

This event is used to delete the given event.

9.2. Invitee Class

This class handles the invitee's related task of the application. The Code description of the Invitee Class methods is given below.

1) GetGetEventsInviteesByCode

This method is used to get the invitee of the event by verifying the given code from the database.

2) SaveEventInviteesDateTimeResponse

This method is used to save the date and time response and there rating about the time in database.

9.3. DAL Class

This class handles the database related task of the application. The Code description of the DAL Class methods is given below.

1) static DAL()

```
{  
    strConn = ConfigurationManager.ConnectionStrings["sqlcon"].ConnectionString;  
}
```

This method is used to initialize the connection string.

2) public static int ExecuteNonQuery(string sql)

```
{  
    return ExecuteNonQuery(sql, null);  
}
```

This method is used to execute the sql query and returns the integer indicated the number of effected rows.

```
3) public static int ExecuteNonQuery(string sql, SqlParameter[] p)
{
    SqlConnection cnn = new SqlConnection(strConn);
    cnn.Open();
    SqlCommand cmd = new SqlCommand(sql, cnn);
    cmd.CommandType = CommandType.StoredProcedure;
    if (p != null)
    {
        for (int i = 0; i < p.Length; i++)
        {
            cmd.Parameters.Add(p[i]);
        }
    }
    int retval = cmd.ExecuteNonQuery();
    cnn.Close();
    return retval;
}
```

This method is used to execute the sql query with given parameters and returns the integer indicated the number of effected rows.

4) Public static object ExecuteScalar(string sql)

```
{  
    return ExecuteScalar(sql, null);  
}
```

This method is used to get the calculated single result by the given query.

5) public static object ExecuteScalar(string sql, SqlParameter[] p)

```
{  
    SqlConnection cnn = new SqlConnection(strConn);  
    cnn.Open();  
    SqlCommand cmd = new SqlCommand(sql, cnn);  
    cmd.CommandType = CommandType.StoredProcedure;  
    if (p != null)  
    {  
        for (int i = 0; i < p.Length; i++)  
        {  
            cmd.Parameters.Add(p[i]);  
        }  
    }  
    object retval = cmd.ExecuteScalar();  
    cnn.Close();  
    return retval;  
}
```

This method is used to get the calculated single result by the given query with parameters.

```
6) public static SqlDataReader ExecuteReader(string sql)
{
    return ExecuteReader(sql, null);
}
```

This method is used to get the data reader by sql query.

```
7) public static SqlDataReader ExecuteReader(string sql, SqlParameter[] p)
{
    SqlConnection cnn = new SqlConnection(strConn);
    cnn.Open();
    SqlCommand cmd = new SqlCommand(sql, cnn);
    cmd.CommandType = CommandType.StoredProcedure;
    if (p != null)
    {
        for (int i = 0; i < p.Length; i++)
        {
            cmd.Parameters.Add(p[i]);
        }
    }
    SqlDataReader reader =
cmd.ExecuteReader(CommandBehavior.CloseConnection);
    return reader;
}
```

This method is used to get the data reader using sql query with parameters.

```
8) public static DataSet ExecuteDataSet(string sql)
{
    return ExecuteDataSet(sql, null);
}
```

This method is used to get the data set by using the sql query.

```
9) public static DataSet ExecuteDataSet(string sql, SqlParameter[] p)
{
    SqlConnection cnn = new SqlConnection(strConn);
    cnn.Open();
    SqlCommand cmd = new SqlCommand(sql, cnn);
    cmd.CommandType = CommandType.StoredProcedure;
    if (p != null)
    {
        for (int i = 0; i < p.Length; i++)
        {
            cmd.Parameters.Add(p[i]);
        }
    }
    SqlDataAdapter da = new SqlDataAdapter();
    da.SelectCommand = cmd;
    DataSet ds = new DataSet();
    da.Fill(ds);
}
```

```
cnn.Close();  
return ds;  
}
```

The above method is used to get the data set using sql query using parameters.

10.DATA DICTIONARY

When developing programs that use the data model, a data dictionary can be consulted to understand where a data item fits in the structure, what values it may contain, and what the data item means in real-world terms. For example, a bank or group of banks could model the data objects involved in consumer banking. They could then provide a data dictionary for a bank's programmers.

A data dictionary is a collection of descriptions of the data objects or items in a data model for the benefit of programmers and others who need to refer to them. A first step in analyzing a system of objects with which users interact is to identify each object and its relationship to other objects. This process is called data modeling and results in a picture of object relationships.

10.1. Data Modeling Overview

A data model is a conceptual representation of the data structures that are required by a database[08]. The data structures include the data objects, the associations between data objects, and the rules which govern operations on the objects[08]. As the name implies, the data model focuses on what data is required and how it should be organized rather than what operations will be performed on the data. To use a common analogy, the data model is equivalent to an architect's building plans.

A data model is independent of hardware or software constraints [08]. Rather than try to represent the data, as a database would see it, the data model focuses on representing the data as the user sees it in the "real world". It serves as a bridge between the concepts that

make up real-world events and processes and the physical representation of those concepts in a database.

10.2. Methodology

There are two key methodologies used to create a data model: the Entity-Relationship (ER) approach and the Object Model.

10.3. Data Modeling in the Context of Database Design

Database design is defined as "design the logical and physical structure of one or more databases to accommodate the information needs of the users in an organization for a defined set of applications" [08]. The data design process follows five steps:

Planning and analysis

Conceptual design

Logical design

Physical design

Implementation

The data model is one part of the conceptual design process [11]. The other is the functional model. The data model focuses on what data should be stored in the database while the functional model deals with how the data is processed. To put this in the context of the relational database, the data model is used to design the relational tables. The functional model is used to design the queries, which will access and perform operations on those tables.

10.4. Components of a Data Model

The data model gets its inputs from the planning and analysis stage [11]. Here the modeler, along with analysts, collects information about the requirements of the database by reviewing existing documentation and interviewing end-users. The data model has two outputs. The first is an entity-relationship diagram which represents the data structures in a pictorial form. Because the diagram is easily learned, it is valuable tool to communicate the model to the end-user. The second component is a data document. This is a document that describes in detail the data objects, relationships, and rules required by the database. The dictionary provides the detail required by the database developer to construct the physical database.

10.5. Why is Data Modeling Important?

Data modeling is probably the most labor intensive and time consuming part of the development process [12]. Why bother developing a data model if the developer is pressed for time? A common response by practitioners who write on the subject is that you should no more build a database without a model than you should build a house without blueprints [12].

The goal of the data model is to make sure that all data objects required by the database are completely and accurately represented. Because the data model uses easily understood notations and natural language, it can be reviewed and verified as correct by the end-users.

The data model is also detailed enough to be used by the database developers to use as a "blueprint" for building the physical database [12]. The information contained in the data

model will be used to define the relational tables, primary and foreign keys, stored procedures, and triggers. A poorly designed database will require more time to develop and is more expensive to maintain in the long-term. Without careful planning, you may create a database that omits data required to create critical reports, produces results that are incorrect or inconsistent, and is unable to accommodate changes in the user's requirements.

10.6. Summary

A data model is a plan for building a database. To be effective, it must be simple enough to communicate to the end user the data structure required by the database yet detailed enough for the database designer to create the physical structure [12].

11. TABLES

This section contains the tabular view of database tables involved in the development of web based meeting scheduler.

1) Table: aspnet_Applications

Table 4. Aspnet_Applications

Column Name	Data Type	Is the Key?	Is Nullable?	Is Computed?	Is Autokkey?	Default
ApplicationName	nvarchar(256)					
LoweredApplicationName	nvarchar(256)					
ApplicationId	uniqueidentifier	Yes				(newid())
Description	nvarchar(256)		Yes			

Foreign Keys

Name	Foreign Table	Primary Key
FK_aspnet_Me_Appli_145C0A3F	aspnet_Membership	PK_aspnet_Applicati_7E6CC920
FK_aspnet_Pa_Appli_45F365D3	aspnet_Paths	PK_aspnet_Applicati_7E6CC920
FK_aspnet_Ro_Appli_32E0915F	aspnet_Roles	PK_aspnet_Applicati_7E6CC920
FK_aspnet_Us_Appli_0425A276	aspnet_Users	PK_aspnet_Applicati_7E6CC920

2) Table: aspnet_Membership

Table 5. AspNet_Membership

Column Name	Data Type	Is in Key?	Is Nullable?	Is Computed?	Is AutoKey?	Default
ApplicationId	uniqueidentifier					
UserId	uniqueidentifier	Yes				
Password	nvarchar(128)					
PasswordFormat	int					((0))
PasswordSalt	nvarchar(128)					
MobilePIN	nvarchar(16)		Yes			
Email	nvarchar(256)		Yes			
LoweredEmail	nvarchar(256)		Yes			
PasswordQuestion	nvarchar(256)		Yes			
PasswordAnswer	nvarchar(128)		Yes			
IsApproved	bit					
IsLockedOut	bit					
CreateDate	datetime					
LastLoginDate	datetime					
LastPasswordChangedDate	datetime					
LastLockoutDate	datetime					
FailedPasswordAttemptCount	int					
FailedPasswordAttemptWindowStart	datetime					
FailedPasswordAnswerAttemptCount	int					
FailedPasswordAnswerAttemptWindowStart	datetime					
Comment	ntext		Yes			

(Table 2 Continued)

Foreign Keys

Name	Foreign Table	Primary Key
FK_aspnet_Me_Appli_145C0A3F	aspnet_Membership	PK_aspnet_Applicati_7E6CC920
FK_aspnet_Me_UserI_15502E78	aspnet_Membership	PK_aspnet_Users_03317E3D

3) Table: aspnet_Roles

Table 6. Aspnet_Roles

Column Name	Data Type	Is in Key?	Is Nullable?	Is Computed?	Is AutoKey?	Default
ApplicationId	uniqueidentifier					
RoleId	uniqueidentifier	Yes				(newid())
RoleName	nvarchar(256)					
LoweredRoleName	nvarchar(256)					
Description	nvarchar(256)		Yes			

Foreign Keys

Name	Foreign Table	Primary Key
FK_aspnet_Us_RoleI_37A5467C	aspnet_UsersInRoles	PK_aspnet_Roles_31EC6D26
FK_aspnet_Ro_Appli_32E0915F	aspnet_Roles	PK_aspnet_Applicati_7E6CC920

4) Table: aspnet_Users

Table 7. Aspnet_Users

Column Name	Data Type	Is in Key?	Is Nullable?	Is Computed?	Is AutoKey?	Default
ApplicationId	uniqueidentifier					
UserId	uniqueidentifier	Yes				(newid())
UserName	nvarchar(256)					
LoweredUserName	nvarchar(256)					
MobileAlias	nvarchar(16)		Yes			(NULL)
IsAnonymous	bit					((0))
LastActivityDate	datetime					

Foreign Keys

Name	Foreign Table	Primary Key
FK_aspnet_Me_UserI__15502E78	aspnet_Membership	PK_aspnet_Users__03317E3D
FK_aspnet_Pe_UserI__5070F446	aspnet_PersonalizationPerUser	PK_aspnet_Users__03317E3D
FK_aspnet_Pr_UserI__29572725	aspnet_Profile	PK_aspnet_Users__03317E3D
FK_aspnet_Us_UserI__36B12243	aspnet_UsersInRoles	PK_aspnet_Users__03317E3D
FK_aspnet_Us_Appli__0425A276	aspnet_Users	PK_aspnet_Applicati__7E6CC920

5) Table: aspnet_UsersInRoles

Table 8. Aspnet_UserInRoles

Column Name	Data Type	Is in Key	Is Nullable	Is Computed	Is Identity	Default	Description
UserId	uniqueidentifier	Yes					
RoleId	uniqueidentifier	Yes					

Foreign Keys

Name	Foreign Table	Primary Key
FK_aspnet_Us_RoleI_37A5467C	aspnet_UsersInRoles	PK_aspnet_Roles_31EC6D26
FK_aspnet_Us_UserI_36B12243	aspnet_UsersInRoles	PK_aspnet_Users_03317E3D

6) Table: EventDateTimes

Table 9. EventDateTimes

Column Name	Data Type	Is in Key	Is Nullable	Is Computed	Is Identity	Default	Description
ID	int	Yes			Yes		
Date	datetime						
Times	nvarchar(200)						
EventID	int						

(Table 6 Continued)

Foreign Keys

Name	Foreign Table	Primary Key
FK_EventInviteeResponse_EventDateTimes	EventInviteeResponse	PK_EventDateTimes
FK_EventDateTimes_Events	EventDateTimes	PK_Events

7) Table: EventFiles

Table 10. EventFiles

Column Name	Data Type	Is in Key?	Is Nullable?	Is Computed?	Is AutoKey?	Default	Description
ID	int	Yes			Yes		
Title	nvarchar(50)		Yes				
Name	nvarchar(50)		Yes				
Size	nvarchar(50)		Yes				
Comments	nvarchar(0)		Yes				
EventID	int		Yes				

Foreign Keys

Name	Foreign Table	Primary Key
------	---------------	-------------

8) Table: EventInvitee

Table 11. EventInvitee

Column Name	Data Type	Is in Key?	Is Nullable?	Is Computed?	Is AutoKey?	Default	Description
ID	int	Yes			Yes		
Email	nvarchar(50)						
EventID	int						
MessageNote	ntext		Yes				
ResponseID	int		Yes				1 for accepted, 2 for rejected, 3 for tentative
PriorityRating	int		Yes				1 for Low, 2 for Normal, 3 for above normal, 4 for high, 5 for very high
ValidationCode	nvarchar(0)		Yes				
IsInvitationSent	bit		Yes			((0))	
LastUpdateOn	datetime		Yes			(getdate())	
IsOrganizer	bit		Yes			((0))	
FullName	nvarchar(300)		Yes				

(Table 8 Continued)

Foreign Keys

Name	Foreign Table	Primary Key
FK_EventInviteeResponse_EventInvitee	EventInviteeResponse	PK_EventInvitee
FK_EventInvitee_Events	EventInvitee	PK_Events

9) Table: EventInviteeResponse

Table 12. EventInviteeResponse

Column Name	Data Type	Is in Key?	Is Nullable?	Is Computed?	Is AutoKey?	Default	Description
ID	int	Yes			Yes		
EventInviteeId	int		Yes				
EventDateTimesId	int						
Response	int						
Rating	int						
IsOrganizer	bit		Yes			((0))	

Foreign Keys

Name	Foreign Table	Primary Key
FK_EventInviteeResponse_EventDateTimes	EventInviteeResponse	PK_EventDateTimes
FK_EventInviteeResponse_EventInvitee	EventInviteeResponse	PK_EventInvitee

10) Table: Events

Table 13. Events

Column Name	Data Type	Is in Key?	Is Nullable?	Is Computed?	Is AutoKey?	Default	Description
ID	int	Yes			Yes		
Subject	nvarchar(50)						
Description	nvarchar(300)		Yes				
Place	nvarchar(100)		Yes				
ConferenceRoom	nvarchar(100)		Yes				
StartDate	datetime		Yes				
StartTime	nvarchar(50)		Yes				
EndDate	datetime		Yes				
EndTime	nvarchar(50)		Yes				
Duration	real		Yes				
IsAllDayEvent	bit		Yes			((0))	
UserId	uniqueidentifier						
CreatedOn	datetime		Yes			(getdate())	
IsDeleted	bit		Yes			((0))	
IsModified	bit		Yes			((0))	

(Table 10 Continued)

Foreign Keys

Name	Foreign Table	Primary Key
FK_EventDateTimes_Events	EventDateTimes	PK_Events
FK_EventInvitee_Events	EventInvitee	PK_Events

11) Table: Events_History

Table 14. Event_History

Column Name	Data Type	Is in Key?	Is Nullable?	Is Computed?	Is AutoKey?	Default	Description
ID	int	Yes			Yes		
EventID	int						
Subject	nvarchar(50)						
Description	nvarchar(300)		Yes				
Place	nvarchar(100)		Yes				
ConferenceRoom	nvarchar(100)		Yes				
StartDate	datetime		Yes				
StartTime	nvarchar(50)		Yes				
EndDate	datetime		Yes				
EndTime	nvarchar(50)		Yes				
Duration	real		Yes				
IsAllDayEvent	bit		Yes				
UserId	uniqueidentifier						
CreatedOn	datetime		Yes				
IsDeleted	bit		Yes				

(Table 11 Continued)

CONCISE TECHNICAL REQUIREMENTS

Foreign Keys

Name	Foreign Table	Primary Key
------	---------------	-------------

12.1. HTML

HTML is the standard language for creating web pages. It is a markup language that uses tags to describe the structure and content of a document. The tags are placed between the opening and closing tags of the element they describe. The browser interprets the tags and displays the content accordingly.

The HTML document is a text file with a .htm or .html extension. It contains a series of tags that define the layout and content of the page. The browser reads the file and renders the page in the user's browser window.

The HTML document is a text file with a .htm or .html extension. It contains a series of tags that define the layout and content of the page. The browser reads the file and renders the page in the user's browser window.

The HTML document is a text file with a .htm or .html extension. It contains a series of tags that define the layout and content of the page. The browser reads the file and renders the page in the user's browser window.

The HTML document is a text file with a .htm or .html extension. It contains a series of tags that define the layout and content of the page. The browser reads the file and renders the page in the user's browser window.

12. TECHNOLOGICAL REQUIREMENTS

A technical requirement pertains to the technical aspects of this system that must fulfill, such as performance-related issues, reliability issues, and availability issues. These types of requirements are often called quality of service (QoS) requirements, service-level requirements, or non-functional requirements.

12.1. HTML

HTML, an acronym of Hypertext Markup Language, is the predominant markup language for web page development [09]. It provides a means to describe the structure of text-based information in a document — by denoting certain text as headings, paragraphs, lists, and so on. It also provides a structured language to supplement text with interactive forms, embedded images, and other objects [09]. HTML is written in the form of labels (known as tags), surrounded by angle brackets. HTML can also describe, to some degree, the appearance and semantics of a document, and can include embedded scripting language code which can affect the behavior of web browsers and other HTML processors [11].

HTML is also often used to refer to content of the MIME type text/html or even more broadly as a generic term for HTML whether in its XML-descended form (such as XHTML 1.0 and later) or its form descended directly from SGML [09].

HTML is not a programming language but it is an application of ISO Standard 8879, SGML (Standard Generalized Markup Language) [11], but specialized to hypertext and adapted to the Web. The idea behind Hypertext is that instead of reading text in a rigid linear structure, the user can easily jump from one point to another point. The user of a

hypertext document navigates through the information based on interest and preference. A markup language is simply a series of elements, each delimited with special characters that define how text or other items enclosed within the elements should be displayed. Hyperlinks are underlined or emphasized words that link to other documents or some portions of the same document.

HTML can be used to display any type of document on the host computer, which can be geographically at a different location. It is a versatile language and can be used on any platform or desktop. HTML provides tags (special codes) to make the document look attractive. HTML tags are not case-sensitive. Using graphics, fonts, different sizes, color, etc., can enhance the presentation of the document. Anything that is not a tag is part of the document itself.

Basic HTML Tags:

- <! -- --> specifies comments
- <A>..... Creates hypertext links
- Formats text as bold
- <BIG>.....</BIG> Formats text in large font.
- <BODY>...</BODY> Contains all tags and text in the HTML document
- <CENTER>...</CENTER> Creates text
- <DD>...</DD> Definition of a term
- <DL>...</DL> Creates definition list
- ... Formats text with a particular font
- <FORM>...</FORM> Encloses a fill-out form
- <FRAME>...</FRAME> Defines a particular frame in a set of frames
- <H#>...</H#> Creates headings of different levels (1 – 6)
- <HEAD>...</HEAD> Contains tags that specify information about a document
- <HR>...</HR> Creates a horizontal rule

<HTML>...</HTML>	Contains all other HTML tags
<META>...</META>	Provides meta-information about a document
<SCRIPT>...</SCRIPT>	Contains client-side or server-side script
<TABLE>...</TABLE>	Creates a table
<TD>...</TD>	Indicates table data in a table
<TR>...</TR>	Designates a table row
<TH>...</TH>	Creates a heading in a table

12.2. Attributes

The attributes of an element are name-value pairs, separated by "=", and written within the start label of an element, after the element's name. The value should be enclosed in single or double quotes, although values consisting of certain characters can be left unquoted in HTML (but not XHTML). Leaving attribute values unquoted is considered unsafe.

Most elements take any of several common attributes: id, class, style and title. Most also take language-related attributes: lang and dir.

The id attribute provides a document-wide unique identifier for an element. This can be used by style sheets to provide presentational properties, by browsers to focus attention on the specific element or by scripts to alter the contents or presentation of an element. The class attribute provides a way of classifying similar elements for presentation purposes. For example, an HTML document (or a set of documents) may use the designation class="notation" to indicate that all elements with this class value are all subordinate to the main text of the document (or documents). Such notation classes of elements might be gathered together and presented as footnotes on a page, rather than appearing in the place where they appear in the source HTML.

An author may use the style non-attributed codes presentational properties to a particular element. It is considered better practice to use an element's id and select the element with a style sheet, though sometimes this can be too cumbersome for a simple ad hoc application of styled properties. The title is used to attach sub textual explanation to an element. In most browsers this title attribute is displayed as what is often referred to as a tool tip. The generic inline span element can be used to demonstrate these various non-attributes.

The preceding displays as HTML (pointing the cursor at the abbreviation should display the title text in most browsers).

Advantages:

- A HTML document is small and hence easy to send over the net.
- It is small because it does not include formatted information.
- HTML is platform independent.
- HTML tags are not case-sensitive.

12.3. JavaScript

JavaScript is a script-based programming language that was developed by Netscape Communication Corporation [13]. JavaScript was originally called Live Script and renamed as JavaScript to indicate its relationship with Java. JavaScript supports the development of both client and server components of Web-based applications. On the client side, it can be used to write programs that are executed by a Web browser within the context of a Web page. On the server side, it can be used to write Web server programs that

can process information submitted by a Web browser and then update the browser's display accordingly.

Even though JavaScript supports both client and server Web programming, we prefer JavaScript at Client side programming since most browsers support it. JavaScript is almost as easy to learn as HTML, and JavaScript statements can be included in HTML documents by enclosing the statements between a pair of scripting tags.

```
<SCRIPT>.. </SCRIPT>.
```

```
<SCRIPT LANGUAGE = "JavaScript">
```

JavaScript statements

```
</SCRIPT>
```

Some functions that can be performed with JavaScript are:

- Validate the contents of a form and make calculations.
- Add scrolling or changing messages to the Browser's status line.
- Animate images or rotate images that change when the mouse is moved over them.
- Detect the browser in use and display different content for different browsers.
- Detect installed plug-ins and notify the user if a plug-in is required.

JavaScript and Java are entirely different languages. A few differences when programming in these languages are:

- Java applets are generally displayed in a box within the web document; JavaScript can affect any part of the Web document itself.
- While JavaScript is best suited to simple applications and adding interactive features to Web pages; Java can be used for incredibly complex applications.

There are many other differences but the important thing to remember is that JavaScript and Java are separate languages. They are both useful for different things; in fact they can be used together to combine their advantages.

JavaScript can be used for Server-side and Client-side scripting. It is more flexible than VBScript. JavaScript is the default scripting languages at Client-side since all the browsers supports it.

12.4. .NET Technology

As the software technologies advance, it makes it difficult for an analyst to decide which software to use for development. The programming team selected the .NET technology for software development for the Web Meeting Scheduler.

The key advantages of the .NET Framework include:

- Provides native support for Web Services.
- Assists with the deployment of software to both users and Web servers.
- Facilitates the development of software with improved reliability, scalability, and performance.

- Helps developers be more productive by:
 - ✓ Making it easier for them to reuse existing code.
 - ✓ Enabling them to more easily integrate components written in any of the more than 20 supported programming languages.
 - ✓ Helping them more easily build software for a wide range of devices using same skills and tools.

12.5. Reasons to use the .NET Framework

The .NET Framework is an integral component for building and running the next generation of applications and XML Web services [07]. The .NET Framework delivers business value with faster time-to-market, improved systems flexibility, and reduced costs.

Ten reasons for developers to build applications using the .NET Framework [07] include:

IMPROVED RELIABILITY

The .NET Framework takes the core achievements originally made in Windows 2000 and brings them to new levels [14]. With advanced ways of monitoring the health of running applications, as well as isolating applications from each other, applications built using the .NET Framework are more reliable than those developed on previous platforms such as COM or COM+ [14].

INCREASED PERFORMANCE

Thanks in part to advanced compilation and caching techniques, server applications have never been faster than with the .NET Framework and its ASP.NET technology [14]. Customers who have moved from ASP to ASP.NET are seeing significant increases in speed on the order of 300- to 500-percent improvement [14].

DEVELOPER PRODUCTIVITY

Developers of all backgrounds are finding that they can rapidly get up to speed on the .NET Framework. The intuitiveness of the programming model, the amount of code already provided in the class libraries and the amount of work that the .NET Framework handles behind the scenes in areas such as memory management have enabled .NET Framework developers to reap huge productivity gains [14].

POWERFUL, GRANULAR SECURITY

The code access security technology in the .NET Framework was designed for today's Internet environments. The .NET Framework can collect evidence about the origin and author of an application. The .NET Framework run-time environment can then combine that evidence with administrator-set or default security policies to make fine-grained decisions about whether to run that application or enable it to access a particular resource. It can even "negotiate" with the application, for example, denying it the permission to write to a protected directory and enabling the application to choose whether it will run, given that it has been denied that permission [14].

INTEGRATION WITH EXISTING SYSTEMS

The COM interop technology in the .NET Framework generates a wrapper around your existing COM components and Windows-based applications (such as Microsoft Office), enabling you to program against them as though they were originally written using the .NET Framework. Applications built using the .NET Framework can connect with existing systems and packaged applications—regardless of their underlying platform. .NET provides native XML Web Services support, or other system-specific connectors. The Native XML Web Services Support, available with Microsoft Visual Basic® .NET 2003, and the Java Language Conversion Assistant (JLCA) help to convert your existing Visual Basic 6.0 and Microsoft Visual J++® code to run on the .NET Framework [14].

EASE OF DEPLOYMENT

The .NET Framework makes it easy to deploy, run, and manage applications. Application isolation and automatic version control of components can help prevent versioning conflicts. Applications built using the .NET Framework can be deployed to a client or server machine simply by copying the application directory to the target machine—no registration is required. With No-Touch Deployment, Windows-based smart client applications can also now be deployed to and updated on target PCs simply by copying the necessary components to a Web server that can be accessed by your end users [14].

MOBILITY SUPPORT

The .NET Framework provides one unified programming model for developing smart client and Web applications for both PCs and mobile devices such as personal digital assistants (PDAs) and mobile phones [14].

NATIVE XML WEB SERVICE SUPPORT

The .NET Framework was designed from the ground up to support XML Web services, a model for cross-platform, distributed computing based on standard protocols such as XML, SOAP, and HTTP. Web services can be used to integrate applications running on different platforms, or to offer software as a service. With the .NET Framework, an application can be transformed into a Web service with just one simple line of code [14].

SUPPORT FOR MORE THAN 20 PROGRAMMING LANGUAGES

The .NET Framework supports the integration of over 20 programming languages in a way unimagined previously, enabling developers to choose the right programming language for the task at hand. All programming languages target a single, extensive, and extensible set of class libraries. Components written in different languages supported by the .NET Framework can interact seamlessly, with no COM plumbing required [14].

FLEXIBLE DATA ACCESS

The .NET Framework technology for interacting with data, ADO.NET, is designed for today's Web-based style of data access. Using ADO.NET, developers have the option of working with a platform-neutral, XML-based cache of the requested data, instead of

directly manipulating the database. This approach to data access frees up database connections and results in significantly greater scalability [14].

The .Net Framework (Dot Net) is a new Microsoft initiative directed to the modification of computer world [14]. It is the foundation of the next generation of Windows-based applications that are easier to build, deploy, and integrate with other networked systems.

More specifically, it is a large set of development tools, servers, software, and services. Its main advantages for the developer are the creation of an integrated information space connecting him or her with computers and programs, as well as connection software applications together. For developers, the value of dotNet lies in interoperability and the seamless connectivity of multiple systems and sources of data [14]. This empowers them to quickly and easily create required products.

The IT department manager of every company has a dream -- an enterprise that performs all business transactions with partners exceptionally over the Internet, with no headaches about the business processes [14]. For this to happen, the processes must be well designed, stable, and easily customized and controlled both from the local network and from any computer in the Internet. All company's employees should have general access to work information, Email and personal documents no matter if they use mobile phone, Pocket PC, Notebook or high-end workstation.

Nowadays, in an age of rapid development of E-commerce, the existing tools for creating digital marketplaces do not always handle the business needs. By developing a new means for this field a major breakthrough belongs to XML Web services [14].

For a long period, they were used by program engineering services provided by external software [14]. When it became clear that it is easier to once create a universal information storage facility and to integrate it into different programs than invent each time a new one, there appeared first Database Management Systems [14]. Next step was the creation of messaging and collaboration systems, e.g. Lotus Notes and Exchange, which simultaneously served as development platforms [14]. Then products providing messaging delivery (Message Oriented Middleware), such as IBM MQSeries and MSMQ [14] came into use. They allowed applications to organize message exchange in a distributed system with manifold (and often unreliable) communication links. Their difference from mail servers lay in the fact that they were oriented on information exchange not between people but various parts of program systems. Finally, one of the last tendencies became Application Servers and Enterprise Application Integration Servers. Initial application servers allowed developers to create scalable solutions from simple software components giving them a ready means of supporting distributed transactions and controlling access to resources [14]. An Enterprise Application Integration Server acts as glue, being the intermediate among existing program systems and helping them to process data and exchange references. Web services enhance and extend the value of these existing technologies. In addition, the use of HTTP as the transport mechanism allows remote calls to these services to pass through corporate firewalls without compromising security. Web services existed before .Net was introduced, but the .Net Framework makes creation of web services far easier to work with than they otherwise would be [14].

13.EXTERNAL VIEW

The external view is an essential part of the IT system model. Here, it is determined what future users expect from the IT system. The functionality that is defined in this view should ultimately be used to verify if the IT system fulfils the requirements.

The external view consists of the problem statement, application design, and interface prototypes. At first sight, these elements appear strange. Some analysts may be tempted to allege that it is sufficient to record user requirements in the form of prose. However, practical experience shows again and again that this is not true. Prose can be inconsistent, imprecise, and incomplete, without it becoming obvious to the future user when he or she reads it. The IT systems developed accordingly and the programmer interprets the prose as a third party, from his or her own viewpoint, and implements the system accordingly.

13.1. Problem Statement

The problem is to develop a calendar application that will be used for scheduling meeting rooms, reserving meeting rooms and scheduling meetings.

13.2. Web Meeting Application Design



Figure 7: User Interface Diagram.

Users of the Web meeting create or access events using a website (www.meetingscheduler.org). The request from the web browser (Figure 7) goes to the web server, which then utilizes business logic to communicate with the SQL Server database .

13.3. The Application UI

There are two parts of the application, One for the organizer for creating and managing the events/meeting and other for the invitees to response on the event.

13.4. Organizer's UI

After the necessary login phase, the main application web page appears on (Figure 8). This page displays the current list of events and supports following main functionalities:

- Create new event by clicking on the calendar date.
- View details of the current/existing events by clicking on the details link under the event subject.
- Perform the administrative tasks.

Web Scheduler

[Home](#) > Manage Events

Your Events

Click on date to create a new invite.

August		September 2009					October
Sun	Mon	Tue	Wed	Thu	Fri	Sat	
30	31	1	2	3	4	5	
6	7	8	9	10 test of mult... Details	11	12	
13	14	15	16	17 testssssads Details new Details test of mult... Details	18 testssssads Details	19 testssssads Details	
20	21	22	23	24 test of mult... Details	25 testssssads Details	26 test of mult... Details	
27	28	29	30	1	2	3	
4	5	6	7	8	9	10	

© 2009 Web Scheduler.

Figure 8: Web based meeting scheduler calendar.

13.5. Create New Event

Multiple dates can be selected by clicking on the calendar dates as shown (Figure 9) below. Once the dates are selected and clicked on the next button on the screen the event details are asked by the user with the time frame selection.

Home > Manage Events > Add new event

Add Events

Subject:

Description:

Place:

Conference Room:(if any)

Select Time Frames:

<input type="text" value="08:00 AM - 09:00 AM"/>	<input type="checkbox" value="13-Oct-2009"/>	<input type="checkbox" value="21-Oct-2009"/>
<input type="text" value="09:00 AM - 10:00 AM"/>	<input type="checkbox" value="13-Oct-2009"/>	<input type="checkbox" value="21-Oct-2009"/>
<input type="text" value="10:00 AM - 11:00 AM"/>	<input type="checkbox" value="13-Oct-2009"/>	<input type="checkbox" value="21-Oct-2009"/>
<input type="text" value="11:00 AM - 12:00 PM"/>	<input type="checkbox" value="13-Oct-2009"/>	<input type="checkbox" value="21-Oct-2009"/>
<input type="text" value="12:00 PM - 01:00 PM"/>	<input type="checkbox" value="13-Oct-2009"/>	<input type="checkbox" value="21-Oct-2009"/>
<input type="text" value="01:00 PM - 02:00 PM"/>	<input type="checkbox" value="13-Oct-2009"/>	<input type="checkbox" value="21-Oct-2009"/>
<input type="text" value="02:00 PM - 03:00 PM"/>	<input type="checkbox" value="13-Oct-2009"/>	<input type="checkbox" value="21-Oct-2009"/>
<input type="text" value="03:00 PM - 04:00 PM"/>	<input type="checkbox" value="13-Oct-2009"/>	<input type="checkbox" value="21-Oct-2009"/>
<input type="text" value="04:00 PM - 05:00 PM"/>	<input type="checkbox" value="13-Oct-2009"/>	<input type="checkbox" value="21-Oct-2009"/>

Figure 9: Web Scheduler Event Creation Page.

After the event details are given and time frames are selected the user is asked to save and proceed next for the invitees' insertion (Figure 10). Once the invitees are inserted organizer can sent the invitations to the invitees mail boxes. The meeting organizer can also share files for the invitees in the event creation process.

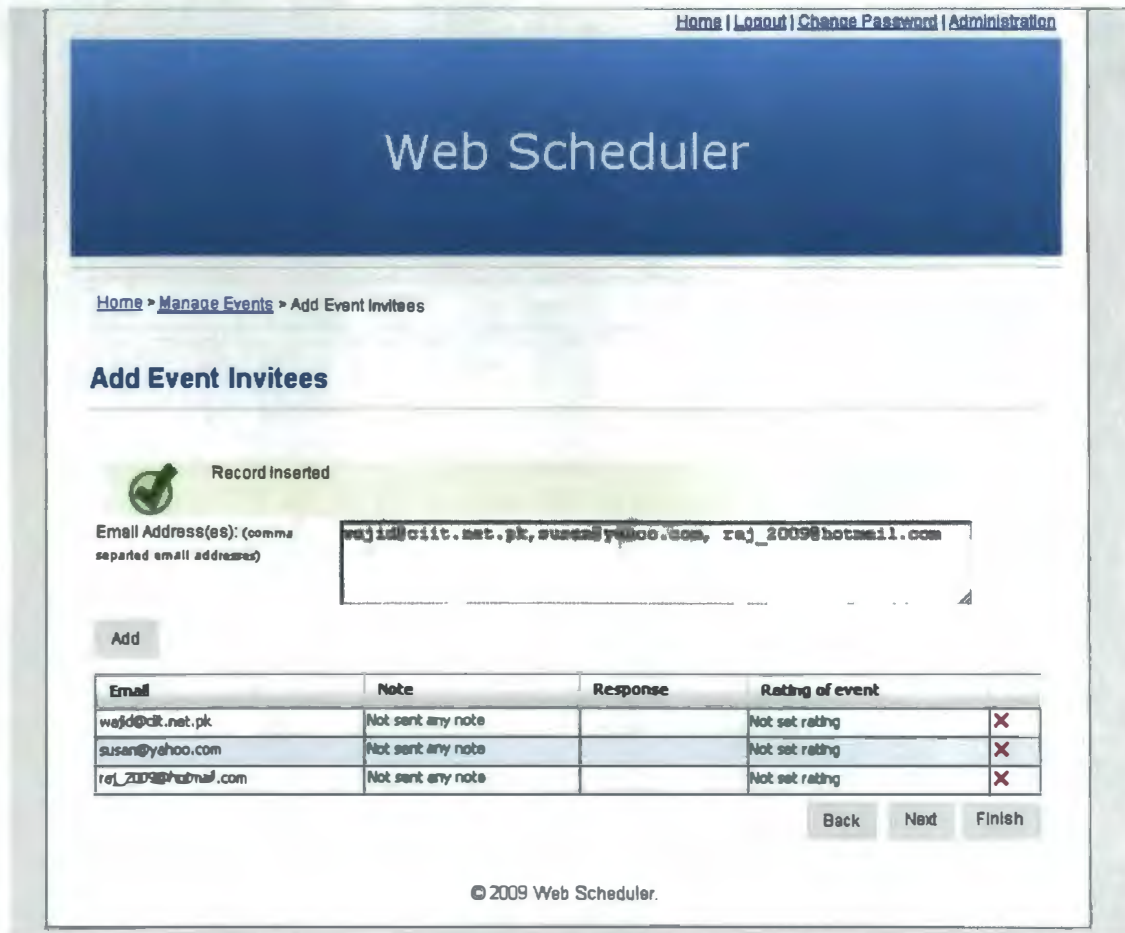


Figure 10: Web Scheduler Event Invitee Page.

13.6. View Details of Existing Events

By clicking on the “Details” link on the calendar, the user is directed to the Event Detail page (Figure 11). This page displays the details of the selected event. Events can be changed easily using a dropdown list on the page. There are different navigation links on the page for the different tasks related to the selected event.



Figure 11: Web Scheduler Event Details View Page.

On this page, a meeting organizer can do the following tasks:

Give his/her rating about the particular time frame by clicking on the “Give your Response” link.

View the responses from the invitees in summarized form by clicking on “View Response Summary”.

Send the invitations to new invitees or can resend the invitation to all or a selected invitee.

Add or delete the invitee(s).

Add or remove the shared files.

13.7. Invitee's UI

All invitees are sent a unique link to their email address for a specific event. By clicking that link, the user is directed to a web page where the user can give his response by accepting or rejecting the event (Figure 12). The invitee can also send a note or message to the organizer and rate a specific time by giving the number of stars. The invitee can also view the response of other invitees by viewing the response summary. The invitee also has the ability to download shared files for the event.

Web Scheduler

[Home](#) > [Manage Events](#) > Event Details

Confirm Invitation

Denotes "YES"
 Denotes "NO"
 Denotes "Level of Interest"

Print Your Name :

Please select one:

17/09/2009	18/09/2009
08:00 AM - 09:00 AM <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	10:00 AM - 11:00 AM <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>
09:00 AM - 10:00 AM <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	11:00 AM - 12:00 PM <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>
10:00 AM - 11:00 AM <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	12:00 PM - 01:00 PM <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>
04:00 PM - 05:00 PM <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	04:00 PM - 05:00 PM <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>
19/09/2009	25/09/2009
12:00 PM - 01:00 PM <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	01:00 PM - 02:00 PM <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>
01:00 PM - 02:00 PM <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	02:00 PM - 03:00 PM <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>
02:00 PM - 03:00 PM <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	03:00 PM - 04:00 PM <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>
04:00 PM - 05:00 PM <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	04:00 PM - 05:00 PM <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>

Please write a note for the organizer :

I like the event and was waiting for the event for long time.

Figure 12: Web Scheduler Response Page.

13.8. Admin UI

From the Admin UI the administrator of the system can manage the users of the system. An Administrator (Figure 13) can approve or reject user accounts. When performed an administrator action is sent to the user in their mail boxes.

Edit User

General user information

UserName: rbarbleri
E-mail: rbarbleri@yahoo.com
Registered: Thursday, September 24, 2009 6:40 AM
Last Login: Thursday, September 24, 2009 6:40 AM
Last Activity: Thursday, September 24, 2009 6:40 AM
Online Now:
Approved:
Locked Out:

Edit user's roles

Administrator Invitee Organizer

Create new role:

Figure 13: Web Scheduler Administrators Page.

14. TESTING AND DEBUGGING

Quality is incorporated into a web application as a consequence of good design. It is evaluated by applying a series of technical reviews that assess various elements of the design model and by applying a testing process.

14.1. Testing Process-an Overview

The Testing Process- the testing process for Web engineering begins with tests that exercise content and interface functionality that is immediately visible to end-users [01]. As testing proceeds, aspects of the design architecture and navigation are exercised. The user may or may not be cognizant of these elements in the software. Finally, the focus shifts to tests that exercise technological capabilities that are not always apparent to end-users such as WebApp infrastructure and installation/implementation issues [01].

Content Testing attempts to uncover errors in content [04]. This testing activity is similar in many respects to copy-editing for a written document. In fact, a large web site might enlist the services of a professional copy editor to uncover typographical errors, grammatical mistakes, errors in content consistency, errors in graphical representations, and cross referencing errors. In addition to examining static content for errors, this testing step also considers dynamic content derived from data maintained as part of a database system that has been integrated with the WebApp [02].

Interface Testing exercises interaction mechanisms and validates aesthetic aspects of the user interface [02]. The intent is to uncover errors that result from poorly implemented

interaction mechanisms or omissions, inconsistencies or ambiguities that have been introduced into the interface inadvertently.

Compatibility testing is to define a set of “*commonly encountered*” client-side computing configurations and their variants [02]. In essence, a tree structure is created, identifying each computing platform, typical display devices, the operating systems supported on the platform, the browser available, likely Internet connection speeds, and similar information. The intent of these tests is to uncover errors or execution problems that can be traced to configuration differences.

When a user interacts with a WebApp, the interaction occurs through one or more interface mechanisms [02]. They are given below:

1. Links- Each navigation link is tested to ensure that the proper content object or function is reached (by the `` Tag).
2. Forms- At a *microscopic level*, tests are performed to ensure that
 - a. Labels correctly identify fields within the form and mandatory fields are identified visually for the user;
 - b. The server receives all information contained within the form and no data are lost in the transmission between the client and server;
 - c. Appropriate defaults are used when the user does not select from a pull-down menu or set of buttons;
 - d. Browser functions (“back” arrow) does not corrupt data entered in a form; and
 - e. Scripts that perform error checking on data that is entered work properly and provide meaningful error messages.

3. Forms- At a more *targeted level*, tests should ensure that
 - f. Form fields have proper width and data types;
 - g. The form establishes appropriate safeguards that preclude the user from entering text strings longer than some predefined maximum;
 - h. All appropriate options for pull-down menus are specified and ordered in a way that is meaningful to the end-user;
 - i. Browser “*auto-fill*” features do not lead to data input errors; and
 - j. The tab key initiates proper movement between data.
 - k. For example, when an application of uploading a file, it must ensure that the *form* tag must consists the *enctype* to be set to the *rfc-1867* standard “*multipart/form-data*” otherwise it will send a file path only (<*form* *action*= “*to where*” *enctype*= “*multipart /form-data*” *method*= “*post*>).
 - l. Client-side scripting, black-box tests are conducted to uncover any errors in processing as the script (e.g., JavaScript) is executed. These tests are coupled with the forms testing, because script input is often derived from data provided as part of forms processing.

Navigation testing applies use-cases, derived as part of the analysis activity, in the design of test cases that exercise each usage scenario against the navigation design [02]. Navigation mechanisms implemented within the interface layout are tested against use-cases and NSUs (Navigation Semantic Unit) to ensure that any errors that impede completion of a use-case are identified and corrected.

The job of the navigation test is:

1. To ensure that the mechanisms that allow the WebApp user to travel through the WebApp are all functional and
2. To validate that each navigation semantic unit (NSU) can be achieved by the appropriate user category.

Component testing exercises content and functional units [01]. When WebApps are considered, the concept of the unit changes [02]. The “*unit*” of choice within the content architecture is the Web page. Each Web page encapsulates content, navigation links, and processing elements (forms, scripts, applets). A “*unit*” within the WebApp architecture may be a defined functional component that enables the WebApp to provide service directly to an end-user [01]. Optionally a unit may be an infrastructure component that is tested in much the same way as an individual module tested in conventional software. In most cases, tests are black-box oriented. However, if processing is complex, white box tests may also be used. In addition to functional testing, database capabilities are also exercised.

Integration testing is performed as the WebApp architecture is constructed; navigation and component testing are used as integration tests [01]. The strategy for integration testing depends on the content and WebApp architecture that has been chosen. If the content architecture has been designed with linear, grid, or simple hierarchical structure, it is possible to integrate Web pages in much the same way as integrating modules for conventional software. However, if a mixed hierarchy or network architecture is used, integration is similar to the approach used for Object Oriented systems [01]. Thread based testing can be used to integrate the set of Web pages required to respond to a user event.

Each thread is integrated and tested individually. Regression testing is applied to ensure that no side effects occur [01].

Each element of the WebApp architecture is unit tested to the extent possible [01]. For example, in MVC architecture the *model*, *view*, and *controller* components are each tested individually. Upon integration, the flow of control and data across each of these elements is assessed in detail.

Configuration testing attempts to uncover errors that are specific to a particular client or server environment [01]. A cross reference matrix that defines all probable operating systems, browsers, hardware platforms, and communications protocols is created. Tests are then conducted to uncover errors associated with each possible configuration.

Security testing incorporates a series of tests designed to exploit vulnerabilities in the WebApp and its environment [01]. The intent is to demonstrate that a security breach is possible.

Performance testing encompasses a series of tests that are designed to access

1. How WebApp response time and reliability are effected by increase user traffic,
2. Which WebApp components are responsible for performance degradation and what usage characteristics cause degradation to occur, and
3. How performance degradation impacts overall WebApp objectives and requirements.

14.2. Application is Tested in the Following Aspects

Dummy users are created and tested. Testing the work flow defined in all the forms and action buttons route the document to next level was tested. Dummy messages are sent to remote users. In the user interface all options are verified by checking at each level. The admin functionalities have been tested by using dummy users database. Validations: Java script is used for validating the data at the client side, which saves the server from extra processing.

Add user/update user - If the admin is trying to enter the userid, which is already existed, he will get a message that the user is already existed.

Delete user - Confirmation message is displayed asking whether or not to delete the user that has been selected. If the change password and the confirm password are not same, the user will be intimated. Text Information - Mostly all text boxes are checked to enter only characters in the name fields (like in user profile etc.) and the fields which would contain numbers only validated at the client side itself. Implementation testing was performed. All the test case criteria points taken up are tested in the live scenario and found complying to the requirements

Testing of application of portal devices- As mentioned earlier This Web Based Meeting Scheduler application was tested on the various portable devices (Ex: Mobile phones, Apple I-Pod touch). Full features were also available while using the application on a mobile device. Synchronizing meeting invitations and reminders to mobile devices and other wireless devices (Palm & Black berry) can be achieved.

15.CONCLUSIONS

The efficiency of any system depends on cooperation during the implementation stage and flexibility of the system to adapt itself to the requirements. This Web Based Meeting Scheduler System (www.meetingscheduler.org) has been developed to overcome the problems with existing communication and collaboration between organizer and the participants.

As evidence of the success of this system, all the data regarding meetings stored at server side through which the organizer can make decisions in time, and the meetings are scheduled in a well refined manner. This Web Based Meeting Scheduler will be best suited to any organization or individual for effective and well managed meetings and meeting resources in a real time environment with the power of internet for access anytime and anywhere. The online scheduling approach is simple and quickest available way to find the right time for a group of people to meet. An online display of optional meeting times allows all participants to indicate their preferred times and enable the organizer to choose the final slot. There is no toggling between participant's calendars and no inefficient email chains.

This web based meeting scheduler makes scheduling transparent and very flexible , regardless of whether the user uses offline calendars or paper planners at all times. When connected to the personal calendar, our system selects available time slots from participant's calendar, allows them to specify their preferences and re-syncs to places the final meeting time on participant's calendars. By using our tools, meeting planners can administer invitations and confirmation even reminders-no more typing email addresses.

16. REFERENCES

[01] Pressman, Roger S (2005) (QA76.758.P935S 2005). Software Engineering: A Practitioner's Approach. New York: Mc Graw-Hill.

[02] Thunderbird Features (2005-2010) Mozilla

Available:[<http://www.mozillamessaging.com/en-US/thunderbird/features/>]

[Accessed Aug 11, 2009]

[03] Windows Calendar (2009). Explore the features: Windows Calendar.

Microsoft Corp.

Available:[<http://www.microsoft.com/windows/windowsista/features/calendar>]

[Accessed Aug 7, 2009]

[04] Online Meeting Scheduler (2007). Comparative Guide and Review of the Best Collaborative Tools: Creative Commons Limited

Available:[<http://www.masternewmedia.org/online-meeting-and-appointment-schedulers-comparative-guide/>]

[Accessed July 24, 2009]

[05] Doodle (2009). Doodle from Google: Web based meeting scheduler.

Available:[<http://www.doodle.com/>]

[Accessed July 10, 2009]

[06] Bertino, E., Ferrari, E., and Atluri, V. The specification and enforcement of authorization constraints in workflow management systems.

ACM Trans. Info. (Feb. 1999), 65–104.

[07] Wikipedia (2008) . Web based encyclopedia: Wikimedia Foundation Inc.
Available: [http://en.wikipedia.org/wiki/Main_Page]

[Accessed Aug 10, 2009]

[08] Ferraiolo, D.F., Barkley, J.F., and Kuhn, D.R. A role-based access control model and reference implementation within a corporate intranet. ACM Trans. Info. (Feb. 1999), 34–64.

[09] Garfinkel, S. and Spafford, E.H. Web Security and Commerce. O’Reilly and Associates, Sebastopol, CA, 1997.

[10] Bertino, E., Pagani, E., Rossi, G.P., and Samarati, P. Protecting information on the Web. Commun. ACM 43, 11 (Nov. 2000), 189–199.

[11] Harrison, M.H., Ruzzo, W.L., and Ullman, J.D. Protection in operating Systems. Commun. ACM 19, 8 (Oct. 1976), 461–471.

[12] Wikipedia (2008). Data Modeling: Wikimedia Foundation Inc. Florida:
Available: [http://en.wikipedia.org/wiki/Data_medeling]

[Accessed Dec 10, 2009]

[13] Wikipedia (2008). Netscape Corporation: Wikimedia Foundation Inc. Florida:
Available: [<http://en.wikipedia.org/wiki/Nerscape>]

[Accessed Dec 22, 2009]

[14] Microsoft (2009). Dot NET framework: Microsoft Corporation.
Available: [<http://msdn.microsoft.com/en-us/netframework/default.aspx>]

[Accessed Dec 30, 2009]

[15] Microsoft (2009). Dot NET framework system requirement.
Available: [<http://msdn.microsoft.com/en-us/library/8z6watww%28VS.71%29.aspx>]

[Accessed June 22, 2009]