# Process Specification and Software Requirements Specification

# for

# Event Management Website

Prepared by Shruti Dongre, Vikram Cothur

December 4, 2017

**Table of Contents 1.Introduction**[**………………….………………………………………………………………...1**](#__RefHeading___Toc441230972)

1.1 Purpose [1](#__RefHeading___Toc441230973)

1.2 Scope 1

1.3 Definitions, Acronyms, and Abbreviations 1

1.4 References 2

1.5 Overview 2

2. Process Specification 3

2.1 Process Overview 3

2.2 Roles and Responsibilities 4

3. Overall Description 4

3.1 Product Perspective 4

3.2 Product Functions 5

3.3 User Classes and Characteristics 6

3.4 Constraints 7

3.5 Assumptions and Dependencies 8

3.6 Requirements Subsets 8

4. External Interface Requirements 8

4.1 User Interfaces 8

4.2 Hardware Interfaces 9

4.3 Software Interfaces [9](#__RefHeading___Toc441230989)

4.4 Communications Interfaces 10

5. System Features 10

5.1 System Feature 1: Plan Meeting 10

5.1.1 Introduction/Purpose of Feature 10

5.1.2 Stimulus/Response Sequence 10

5.1.3 Associated Functional Requirements 10

5.2 System Feature 2: Monitor Meeting 11

5.2.1 Introduction/Purpose of Feature 11

5.2.2 Stimulus/Response Sequence 11

5.2.3 Associated Functional Requirements 11

5.3 System Feature 3: Manage Interactions 12

5.3.1 Introduction/Purpose of Feature 12

5.3.2 Stimulus/Response Sequence 12

5.3.3 Associated Functional Requirements 12

5.4 System Feature 4: Replan Meeting 12

5.4.1 Introduction/Purpose of Feature 12

5.4.2 Stimulus/Response Sequence 13

5.4.3 Associated Functional Requirements 13

5.5 System Feature 5: Modify User Preferences 13

5.5.1 Introduction/Purpose of Feature 13

5.5.2 Stimulus/Response Sequence 13

5.5.3 Associated Functional Requirements 14

6. Other Nonfunctional Requirements 14

6.1 Performance Requirements 14

6.2 Design Constraints 14

6.3 Software System Attributes 15

6.4 Other requirements 16

Appendix A: Analysis Models 17

Appendix B: Prototype 20

## Introduction

## 1.1 Purpose

We are desires to create and market a event management website. This document is our team’s process and requirements document that will capture the necessary requirements of the website, as well as the process we used to create and elicit them. These requirements will allow us to develop the website, or better describe to another group the exact website that should be developed.

The document may be used as a contract between the customer and the designers to formalize the agreement about the specifications and requirements for the product. It may also be used by the designers and developers in aiding them to achieve a better understanding of the system, and to create a product matching the customer’s expectations

## 1.2 Scope

This product, the Event Management Website, here to fore referred to as EMW, aims to create a customizable, web-based system for groups of individuals or organizations that allows them to easily, quickly, and precisely select event organizer for their events.

The website will enable the customer as well as event organizer with maximum input from the customer in terms of their requirements as per their event. Thus the website will perform more intelligently than the products in the past.

Some of the benefits of the website to the customer include providing easy to find a good event organizer among so many event organizer in his locality and for event organizer, new customers aare attracted. For the customer, the benefits include more flexibility with entering event structure and requirements and specifying preferences, as well as being able to event organizer as their ratings on their account for previous events.

## 1.3 Definitions, Acronyms, and Abbreviations

Customer: User who conducts the event.

Event organizer : Organize the event as per requirements of customer.

EMW: Event Management Website is the software system our team is creating to enable users to schedule meetings more easily and efficiently

Concurrency: When meetings are held in parallel, or at the same time

Confirmation: An e-mail notification sent to Attendees by the Initiator confirming the final meeting or event.

Conflict: When no date can be found in the Date range and not in the Exclusion set.

Strong conflict: No date can be found within the Date range and outside all Exclusion sets.

Weak conflict: Dates can be found within the Date range and outside all Exclusion sets, but no date can be found at the intersection of all Preference sets.

Date range: Time interval prescribed by the Initiator that includes the Exclusion and Preference sets.

Date set: A pair of input, including calendar date and time period.

Exclusion set: A set of dates on which the Attendees cannot attend the meeting.

Important participant: A user who is an Attendee, and is required to attend a meeting.

Initiator: A user who has the task of planning a meeting, sending Invites, monitoring the responses, resolving conflicts, and confirming the final event.

Invite: An e-mail notification sent to Attendees by the Initiator, requesting input.

Preference set: A set of dates on which the Attendees would prefer the meeting to take place.

## 1.4 References

The references we used for our requirements specification include:

* Initial requirements
* Customer and event organizer’s interviews—the members of our team assumed the roles of customers and organizer.
* Existing systems—allevents.in, AE organizer, Whova

## 1.5 Overview

The following document contains both the Process Specification and Requirements Specification. The next section describes in detail the process we followed and the roles each of the members in our team played. The section following that is an Overall Description that presents a background of the general factors that affect the website and its requirements, a summary of the major functions the system will perform, general characteristics of the intended users of the product, general description event organizer that will limit the user’s options, factors that will affect the requirements stated in this document, and requirements that may be delayed until future versions of the system.

The sections that follow are Customer – Event organizer Interface Requirements, System Features, and Other Nonfunctional Requirements. They provide a detailed explanation of all inputs into and all outputs from the software system, and a detailed specification of the requirements as they pertain to each of the features the system offers. The sections also specify both the static and dynamic numerical requirements placed on the system or the human interaction with the software as a whole, as well as the design constraints imposed on the system due to hardware limitations or other standards. Discussion on the attributes of the software system also follows.

There is an Appendix at the end which contains the various diagrams referenced throughout the document.

# Process Specification

## 2.1 Process Overview

This section covers the process our team followed in order to create the requirements for EMW.

It was important for us to first of all understand the problem; the purpose and goal of the project needed to be scoped out. Then we determined our specific roles within the group. These roles are further defined in the next section. We then discussed a process to follow to achieve our goals. We mapped the deadlines we would follow tentatively to keep abreast of the project, and be able to accomplish our goal in a timely manner.

We realized that sketching a scenario helped us gain perspective on the problem by specifying what sort of problems a user could be confronted with. The next step was to determine the stakeholders of the system, who are defined in Section 3.3.

Once we had established the ground rules and background on our problem, we started analyzing our problem further to elicit requirements. We assumed the roles of website, Customer and Event organizer to better understand why EMW was necessary, what were the benefits, costs, and risks of this system, and what functionality would the user require or prefer in the system. We then ranked these requirements to differentiate between the most crucial functions and extra requirements.

Then we assumed the role of designers and developers to determine the feasibility of these functions, and ascertain constraints on these functionalities. We utilized dependency diagraphs to aid us in determining the importance of the functions and their constraints.

The elicitation gave birth to a lot of issues and questions we had regarding the requirements. We posed these questions to the Users and Customer, and tried to come to a better understanding. The process was iterative because we had to go back and determine the benefits, costs, risks, as well as feasibility and constraints of the requirements, or make necessary modifications.

In the process, we created a prototype to present our User with a visual example of what the system will offer. This helped us better understand the problems in our requirements understanding, and also helped the customer add to or subtract from the requirements.

We then compiled this document which will be a work in progress throughout as newer requirements or modifications are requested. The next step would be to design and implement the system.

We met several times throughout the span of the Phase I to discuss ideas and clarify questions, as well as to compile our presentation and help each other with problems.

# Overall Description

Our team composed the following scenario to help us understand better the overall problem:

Kim Henderson, administrative assistant to the Senior Vice President of All-State has to schedule a meeting for 200 associates for the Senior Vice President to make an important announcement introducing a new strategy for their line of business in the company. He has made it clear that the two-hour long meeting must be held as soon as possible in the month of October, and that all the 7 Vice Presidents must be there, along with at least 70% of their direct reports. He has also asked her to send out an agenda and some additional information with the invite. She must find a suitable date and time for the meeting, a room that is available and can accommodate a large group, and ensure that she has confirmation from all the important participants that need to attend.

## Product Perspective

The Event Management Website assists users in scheduling a meeting at a proposed date range and location. It also helps in scheduling more than one meeting, and holding distributed meetings in a concurrent manner. This product functionality is carried out by the following interfaces:

1. System Interfaces
2. User Interfaces
3. Hardware Interfaces
4. Software Interfaces

### System Interfaces

The EMW system will come into interaction with memory of the machine it is running on, the network that is used to access the Internet, the operating system that is currently installed on the machine the system is running on, and other system utilities. Network interfaces could include the socket interfaces, memory management interfaces could include memory-mapped file interfaces, memory locking, or memory protection. Similarly, operating system interfaces could include the process interfaces, clocks and timers, as well as threads.

### User Interfaces

User interfaces serve as logical interfaces between software product and its users. This includes screens, layouts, and reports that provide the interface to the user. We have two kinds of users for our system—an Initiator and an Attendee. Following is a description of each of the separate interfaces that the users can expect:

#### Initiator Interfaces

The Initiator should have a *Login screen* to enter the EMW system (Figure 1, Appendix B). Once the Initiator is logged in, he can initiate a meeting through the *Initiate Meeting Interface* on the *User’s Homepage* (Figure 2, Appendix B). The Initiator can also view meeting responses, ongoing progress of invites, and consistency of data through *Meetings Initiated Interface* on the User’s Homepage (Figure 6, Appendix B). As the date and locations responses are sent, the system determines the best date, time, and location for the meeting, and the Initiator can view it through the *Meetings Initiated Interface* as well. Finally, the Initiator can send confirmation of the final event through a *Confirmation Interface.*

#### Attendee Interfaces

Attendees also enter the EMW system using the *Login screen* (Figure 1, Appendix B). If they do are not already registered users, they can become a member using the *Register Interface*. They can view their meeting invites using the *Meetings Pending Response Interface* on the *User’s Homepage* (Figure 2, Appendix B). They can modify and submit their responses using the *Response Interface* (Figure 6 and 7, Appendix B).

### Hardware Interfaces

EMW will require a Processor speed of 133 MHz, with a RAM of at least 60 MB. A mouse or compatible input device will be required to interact with the system. Temporary files might be saved in the local cache or for recovery. The system will use the TCP/IP protocol since it’s most reliable for electronic communication over the internet. A monitor will be required to view the user interface.

### Software Interfaces

EMW will be compatible on Microsoft Windows and Linux. Since EMW is being developed in Java, it is platform independent.

## Product Functions

Our team has determined the following five functions to be most important and pertinent for EMW to provide:

First and foremost, since it is a scheduling system, it should allow an Initiator to be able to Plan a Meeting. Planning a meeting involves being able to specify a tentative date range for the meeting, the time duration it should be held for, the Attendees who are invited, further demarcate the Attendees as active or important if applicable, and choose locations if necessary. To efficiently plan a meeting, an Initiator must have available Attendee information such as e-mail addresses, and their schedules or availability. An Initiator should also be able to preview the invitation at each step, and know what the next step will be.

Next, receiving a response from the Attendees is important for the meeting logistics to be confirmed, and meeting to be held, hence Managing Interactions is an important functionality. This involves the system to duly deliver the Attendees’ responses to the Initiator, allowing flexibility in terms of how often the Initiator wishes to receive responses. The system should also automatically consider the responses and calculate a tentative date for the meeting based on the preference and exclusion sets. The system should also duly deliver the communication between Initiator and Attendees, as well as the final confirmation of the meeting to the concerned user.

An Initiator must be able to view the responses of the Attendees, the date the system is suggesting based on the Attendees’ responses, the availability of the rooms based on the suggested date, and any other suggestions or requests made from the Attendees. Hence the system must allow an Initiator to Monitor a Meeting. This includes the Initiator being able to view an Attendees’ capability to attend virtually if working remotely, whether conference lines are available, and whether any documentation needs to be shared with all the Attendees before the meeting. An Initiator should also be able to view whether an Attendee has received and read the meeting invite or not.

Although the system uses the preference and exclusion sets specified by the Attendees to determine the best possible date to hold the meeting, if conflicts arise, the system must allow the Initiator to efficiently Replan a Meeting. Or the system must alert the Initiator to replan a meeting if a certain threshold of acceptance is not received for the meeting.

Last but not least, the system must allow the user to set certain preferences, especially upon login. This will also include specifying exclusion set and preference sets.

## User Classes and Characteristics

We have determined the following three classes as the most important and active users of the system:

### Initiators

Initiators will be using the system to plan a meeting, send preliminary invitations to the Attendees, receiving responses, replanning meetings or resolving conflicts, and sending confirmation of the meetings. They will be the most important user for our system, and their needs are the most substantial. They may or may not be technically savvy, but we will assume they have the basic knowledge to navigate on the Web and use a PC. A meeting initiator can be any person who has a need to call a meeting, ranging from administrative assistant to an executive of the company. It could include a teacher, a technician, a doctor, lawyer, architect, programmer, etc. They are hard-pressed for time, and they want to minimize their planning time as much as possible.

### Attendees

Attendees are the users who are named as participants in the initial requirements. Our team felt the term Attendees helped differentiate them better for our understanding purposes. The Attendees, similarly to the Initiators, might not be technically savvy, and they are also hard-pressed for time. Hence, their preference would be to minimize the amount of interaction required to respond, and the number of messages received in regards to a meeting. Attendees can fall into four subcategories:

* 1. *Potential Attendees*—any participant who is sent and receives an invite to a meeting.
  2. *Active Attendees*—a participant who has an active role in the meeting, such as moderator, facilitator, speaker, host. They might have specific requirements for the equipment available at the meeting, including microphone, speakers, projector, stationery, etc.
  3. *Important Attendees*—a participant who is required to attend a meeting, and might have preference on where the meeting should be situated. If an important participant cannot attend, then he must designate a representative instead.
  4. *Representative*—a participant who is requested to attend a meeting upon an Important Attendees designation.

### Administrators

Administrators are responsible for setting up user accounts, providing them with login ID and password. They also maintain the various databases with user accounts, room and equipment availability repositories, and preference settings. The Administrators are expected to be relatively technically savvy; they should be able to follow technical instructions, navigate and modify settings on a PC, and troubleshoot technical problems.

## Constraints

Constraint requirements set restrictions on how the user requirements are to be implemented. We determined the following to limit the options for the developers in implementing the system:

### Interface Requirements

#### Communication Interfaces

* Client/server based communication
* TCP/IP protocol
* If email notification is utilized then SMTP must be used for mail delivery

#### Hardware Interfaces

* Windows 2000, 2003, or XP
* Pentium III or better processor recommended
* 512 MB of RAM
* 150 MB of free disk space, plus 1 MB for each user

### Quality Requirements

#### Portability

The system shall be easy to move from one environment to another with minimal time, cost and changes in code.

#### Adaptability

The system shall easily cope with requirements changes.

#### Availability

The average availability of the system shall be 98% over 1 year during working hours and never less than 75% of working hours in any week.

#### Security

No single event, such as a fire, shall cause the loss of more than 1 week's information from the system. Unauthorized users shall be unable to use the system.

#### Safety

No data shall be lost from the system when a power failure occurs.

## Assumptions and Dependencies

### Assumption/Dependency 1:

A third party database system is needed for storing of all data used in the EMW.

### Assumption/Dependency 2:

Every machine accessing the EMW system must have a web browser, and extensions for the language that the EMW shall be written in.

### Assumption/Dependency 3:

There might be a pre-existing set of user logins already in the company (via LDAP for example), if this is the case, the EMW would depend on that login system as the preferred method of logging in. This pre-existing system would already have some user information, like email address, name, office number, etc.

## Requirements Subsets

Allowing users to specify their preferences is an additional functionality that is not required, nor rated high in our requirements. This functionality will be met only when all others have been completed to satisfaction, and when and if time and resources permit.

# External Interface Requirements

## User Interfaces

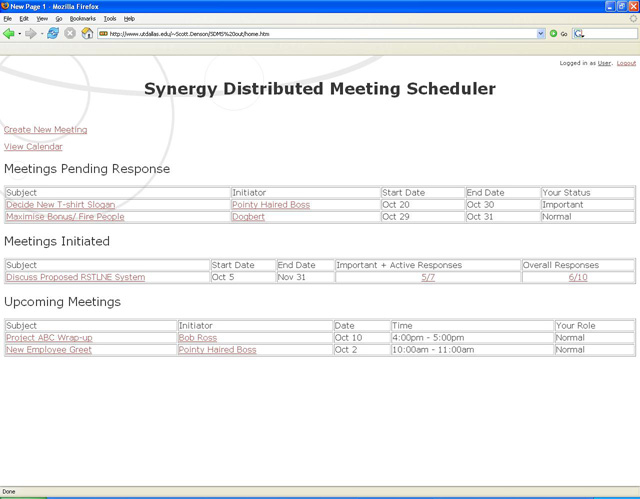
4.1.1 All functionality will need to be handled via a web interface.

4.1.2 Logins will form a critical part of this interface and the ability to logout, as well as see what user is logged in, will be displayed on every screen during a users’ session.

4.1.3 Error screens should detail the error, as well as possibly suggest solutions if it is a user error, or contain contact information for tech support or the system administrator.

4.1.4 The Screen Layout should be configured such that the Internet Explorer browser at Full screen in 800 x 600 resolutions is able to display the entire width of the page.

4.1.5 All screens should provide a consistent look and feel.



Typical view of a user’s home screen.

**Typical view of a user’s home screen**

4.1.6 Additional sample user screens can be found in the Appendix B.

## Hardware Interfaces

The synergy meeting scheduler system will not directly interface with the hardware in any way. All hardware interfacing will be done through the OS system calls or other applications.

## Software Interfaces

4.3.1 Java System Web Server 6.1: The primary software interface will be with the Java Web Server. The web server will manage the web application (.war) deployment as well as serving as the input and output point for end users through the HTTP protocol.

4.3.2 JDBC 3.0: The JDBC library for java will be used to interface the J2EE code with the Oracle Database (third party database). Functionality is described at <http://java.sun.com/javase/technologies/database.jsp>

4.3.3 J2EE 1.4: The system will be built on J2EE and will utilize standard libraries and functionality contained in J2EE.

4.3.4 Oracle 10g: The Oracle Database will be accessed by the software through the JDBC library. Meeting and user information will be stored in the database and accessed through standard database operations as provided by the JDBC

4.3.5 Solaris 6.1: The EMW will operate on the Solaris operating system for x86. All other services and servers will run on top of this platform. Solaris will also provide networking and hardware support.

4.3.6 DataDirect Connect for JDBC: This is a third party driver for the JDBC that is required to connect JDBC to a database. In this case it will connect the JDBC to the Oracle server.

## Communications Interfaces

4.4.1 HTTP: The java web server will provide its functionality over HTTP protocol to clients using a web browser.

4.4.2 SSL: SSL will run on the web server and provide encryption for the HTTP interface as a user logs in and throughout the users session, in order to maintain security and privacy.

# System Features

## Plan Meeting

### Description and Priority

Planning a meeting shall allow the Initiator to specify a tentative date range for the meeting, the time duration it should be held for, the Attendees who are invited, classify the Attendees as active or important if applicable, and choose locations if necessary. The functionality shall allow the Initiator to be able to preview the invitation at each step, and know what the next step will be if there are multiple steps. It should also make available Attendee information such as e-mail addresses, and their schedules or availability.

In a stringent time duration, if the product has to include minimum functions, then Plan Meeting has the highest priority as its purpose is to allow the Initiator to propose date and location, the two most important elements of planning meeting process. Since it directly impacts the most important user, understandably, the benefit of achieving this functionality is high, and the penalty of omission is also high.

The cost for this particular feature will be very high due to the complexity of the requirements and hence the programming involved. There is no code reuse as this is the first module being made and testing will also incur a good amount of time and resource. Thus the total cost for the development of Plan Meeting will be high.

### Stimulus/Response Sequences

* Step 1: A meeting Initiator will setup a meeting and propose a date, time, and duration for the meeting as well as define the Attendees according to if they are important, active, or just regular participants.
* Step 2: The responses will come in from all Attendees, and the EMW will plan the meeting accordingly
* Step 3: Once the responses have been received and the EMW proposes an ideal date and time for meeting based on preference and exclusion sets in the response, the Initiator will send the confirmation for the meeting

### Functional Requirements

* 5.1.3-REQ-1: The system shall be able to schedule a meeting between an Initiator and all Attendees, which will include a meeting subject, proposed date/time range, and location, as well as any additional information/details that the subject might not cover (such as an agenda).
  + When an Attendee is selected, he is informed about the meeting and then chooses whether or not he can attend.
* 5.1.3-REQ-2: The system will allow the Initiator to see all potential Attendee’s current schedules.
* 5.1.3-REQ-3: The system will allow potential Attendees to give a preference and exclusion set to a meeting request, and this will allow the system to automatically detect the “best” date match accordingly.
* 5.1.3-REQ-4: The system may support a level of importance on each Attendee, allowing an Initiator to designate how important they are to the meeting, and if they are a mandatory participant
  + If a mandatory participant cannot make it, the meeting scheduler system alerts the Initiator and suggests a time that is better suited for all mandatory participants
* 5.1.3-REQ-5: The system will be responsible for handling several meetings that occur in parallel and thus must be able to manage this concurrency.
* 5.1.3-REQ-6: The system must have a repository for available locations, size they can accommodate, and equipment they offer

## Monitor Meeting

### Description and Priority

Monitoring a meeting means that the Initiator must be able to view the responses of the Attendees, the date the system is suggesting based on the Attendees’ responses, the availability of the rooms based on the suggested date, and any other suggestions or requests made from the Attendees. Hence the system must allow an Initiator to Monitor a Meeting Invite. This includes the Initiator being able to view an Attendees’ capability to attend virtually if working remotely, whether conference lines are available, and whether any documentation needs to be shared with all the Attendees before the meeting.

Based on our understanding and mutual agreement, our team feels that it is more important for the Initiator to be able to send out a meeting invite, and for the system to deliver the responses, and ascertain the best possible date based on Attendee response. However, it is still important for an Initiator to be able to view the responses and send a confirmation of the meeting or event. Having a virtual meeting is also a customer and user requirement; hence the Initiator must be able to coordinate a virtual meeting as well. Keeping these criteria in mind, we have given Monitor Meeting a Medium priority. It is not critical, but the system will not have very good functionality or performance without this feature.

This feature does not have very high complexity, and we will be able to reuse most of the code developed for the overall User Interface. Testing will incur more cost here as a significant amount of test cases will need to be generated. The resulting cost should be moderate.

### Stimulus/Response Sequences

* Step 1: Initiator initiates a meeting request
* Step 2: The system stores all data such as meeting responses, date system suggests based on preference and exclusion sets, availability of rooms, and other suggestions and dates from Attendees
* Step 3: Initiator can view all the stored system data in a visually pleasing manner

### Functional Requirements

* 5.2.3-REQ-1: The system must be able to store all data from a meeting invite and responses from the Attendees
* 5.2.3-REQ-2: The Initiator shall have the ability to monitor all data from a meeting that he initiates

## Manage Interactions

### Description and Priority

Managing Interactions involves the system to duly deliver the Attendees’ responses to the Initiator, allowing flexibility in terms of how often the Initiator wishes to receive responses. The system should also automatically consider the responses and calculate a tentative date for the meeting based on the preference and exclusion sets. The system should also duly deliver the communication between Initiator and Attendees, as well as the final confirmation of the meeting to the concerned user.

It is critical for the system to deliver messages across for the Initiator and Attendee to be informed of the communication. Hence the priority of this feature is high. The omission of this feature will result in a redundant system; hence the penalty is severe, and thus quite high.

The cost of this feature is dependent on the successful delivery of messages between participants and users. This technology requires relatively simple coding and limited resources, thus the cost should be relatively low.

### Stimulus/Response Sequences

* Step 1: A response will come in from an Attendee, potential, active, or important
  + A typical response might be:
    - Attending
    - Not attending
    - Maybe attending
    - Choose a preferred location (if “Important”)
    - Choose a new preference set for meeting
    - Choose a new exclusion set for meeting
    - Choose a piece of equipment (if “Active”)
* Step 2: The EMW system must log all interactions and deal with them in the proper way.
  + Some actions from the system might be:
    - Cancel a meeting
    - Plan the meeting
    - Alert the Initiator of Attendee’s reactions to meeting

### Functional Requirements

* 5.3.3-REQ-1: The system shall manage all interactions that all users might participate in, such as
  + - Communication requests
    - Replies that come in about the meeting
    - Negotiations and conflicts between participants
    - Alert participants of current status of the meeting

## Replan Meeting

### Description and Priority

Replanning or Cancelling a meeting will allow Attendees to reject a meeting invite, and the Initiators to accordingly replan or cancel a meeting if a threshold of participants are not met, or constraints not fulfilled. This feature shall also support conflict resolution, which is an important requirement for the customer and user. The system shall allow Initiators to modify the meeting constraints, and inform the Attendees accordingly.

The customer has explicitly requested conflict resolution to be a requirement for the system. It is imperative to incorporate conflict resolution in the system for it to have marketable value. Hence the priority of this feature is medium-high.

The feature is complex in nature, and hence requires intricate algorithms and development. Therefore, the cost that will be incurred in achieving this functionality will be high since it requires a lot of computation, developing new code, and rigorous testing.

### Stimulus/Response Sequences

* Step 1: The meeting has reached a threshold that is predetermined OR the Initiator has requested that the meeting be re-planned.
* Step 2: The system will then cancel the meeting and re-plan it accordingly, as well as inform everybody about the actions taken.

### Functional Requirements

* 5.4.3-REQ-1: The system will support a rescheduling feature
  + - The initiator might feel that the meeting needs to be rescheduled based on what he sees in attendance.
    - The system might warn the initiator that not enough people are attending based on a threshold set.
    - If a constraint changes, such as a meeting place must be used by a more important meeting, then a reschedule must occur.
* 5.4.3-REQ-2: The system must support a conflict resolution method that would be stated by a client; therefore, it must be a feature that has a list of pre-made options that a client would pick from

## Modify User Preferences

### Description and Priority

EMW shall enable users to modify their preference sets by adding new date and time to their preference set, or removing old date and time from preference sets. Important Attendees should also be able to change location preferences.

Since this same functionality is provided to the Attendees in the meeting invite, this feature has low priority. It benefits the Initiator because information about Attendee preferences is readily available, yet it does not add value since the Attendee will need to ensure the information is updated regularly. The Initiator will have access to Attendee availability due to the LDAP accounts EMW will link with. Therefore, this feature provides little value to the process.

The cost of this feature should be low since code can be reused from the response feature.

### Stimulus/Response Sequences

* Step 1: User logs into system, and goes into control panel feature
* Step 2: The user may create or update his preferences, and the system will then store those preferences and apply them when needed.

### Functional Requirements

* 5.5.3-REQ-1: The system will have a secure login, which requires a username and password.
* 5.5.3-REQ-2: The system will be online and accessed from a web based interface.
* 5.5.3-REQ-3: The system will allow users to change preferences such as a preferred date set, and an exclusion set (exclusion set, for example, might be vacation time, sick time, etc).

# Other Nonfunctional Requirements

## Performance Requirements

### Requirement 1

Response time of the system shall be less than 1 minute

### Requirement 2

The time interval between the submission of a meeting request, determination of meeting date/location and communication of this information to all participants concerned shall be less than 5 minutes

### Requirement 3

The time interval between the determined and the actual meeting shall be 10 minutes.

### Requirement 4

The system shall be able to handle 10 concurrent meetings for every 1 hour.

### Requirement 5

In case of unexpected failures, system recovery time shall be less than 30 minutes.

## Design Constraints

### Hardware Limitations

6.2.1.1 512 megabytes (MB) of RAM for the operating system and services, plus an additional 32 MB of RAM for each 10 concurrent users

* + - 1. 150 MB of available disk space

## Software System Attributes

## Usability/User-Friendliness

6.3.1.1 The graphical user interface of the system shall be usable by non-experts

6.3.1.2 The system shall accommodate as many decentralized requests as possible. Any authorized user of the system shall be able to make meeting requests regardless of where they are currently located

6.3.1.3 The user interface shall allow intuitive interaction mapped after existing technology, requiring minimal interaction

### Robustness

6.3.2.1 The system shall greatly reduce the amount of overhead involved when organizing meetings where participants are distributed over several different places and still have a means of communication (i.e. the Internet)

6.3.2.2 The amount of interaction among meeting participants shall be kept minimal

6.3.2.3 A meeting shall be accurately monitored, especially when held in a virtual place

### Extensibility

6.3.3.1 The system shall have the ability to be extended to accommodate the following typical variations:

6.3.3.1.1 Handling explicit priorities among dates in preference sets;

6.3.3.1.2 Handling explicit dependencies between meeting date and meeting location;

6.3.3.1.3 Participation through delegation, in which a participant may ask another person to represent him/her at the meeting;

6.3.3.1.4 Partial attendance in which a participant can only attend part of the meeting;

6.3.3.1.5 Different date/address formats, different languages;

6.3.3.1.6 Partial reuse in other contexts such as helping establish course schedule, etc

### Privacy/Security

6.3.4.1 The system shall enforce privacy rules, and participants who do not meet the correct privacy level shall not be aware of constraints expressed by other participants

### Reliability

6.3.5.1 The system shall reflect as closely as possibly the way meetings are typically managed

6.3.5.2 Physical constraints shall not be broken. For example, a person can only be in one place at any time, and a meeting room can only accommodate a single meeting at any given time.

6.3.5.3 The meeting date and location shall be as convenient as possible, and made available as early as possible.

### Customizability

6.3.6.1 The system shall accommodate two different modes for professional or private meetings. Both modes are differentiated by restrictions on the time periods that can be allocated

### Flexibility

6.3.7.1 The system shall be able to handle evolving data. Examples of such evolving data include varying sets of participants, various locations/addresses of participants, participants dropping out of the meeting, etc.

6.3.7.2 A meeting shall be re-planned dynamically and flexibly in regards to time and place.

## Other requirements

### Database Requirement

System backup shall be taken everyday

### Legal Requirement

Personal customer information shall be used only in accordance with privacy laws.

### Internationalization Requirement

6.4.3.1 The System shall be internationalized to work properly in the following countries:

* United States of America
* Canada
* United Kingdom
* Australia
* New Zealand

6.4.3.2 The application shall be internationalized for the following native languages and dialects of the target countries:

* English
* American
* Australian
* British

6.4.3.3 The application shall use Unicode ISO-10646 to properly handle multibyte character sets (MBCS) for the official languages of the target countries

6.4.3.4 The application shall use target country and language conventions for:

* Calendars (i.e., Western)
* Date formatting (i.e., European and USA)
* Time formatting (e.g., 12 hour vs. 24 hour clock)

6.4.3.5 Internationalization shall not require changes to executable software component including user interfaces

### Time Scale Requirement

The system shall be developed with in a period of 1 month.

Appendix A: Analysis Models

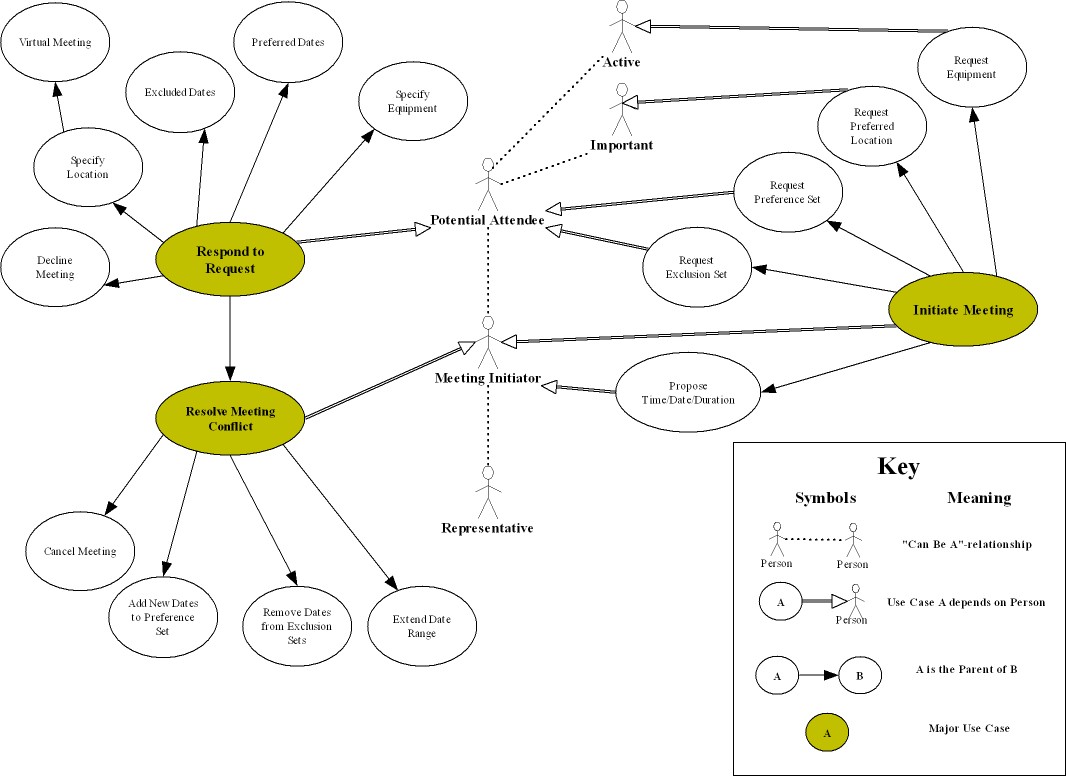


Figure 1: Enterprise Requirements Dependency Diagram

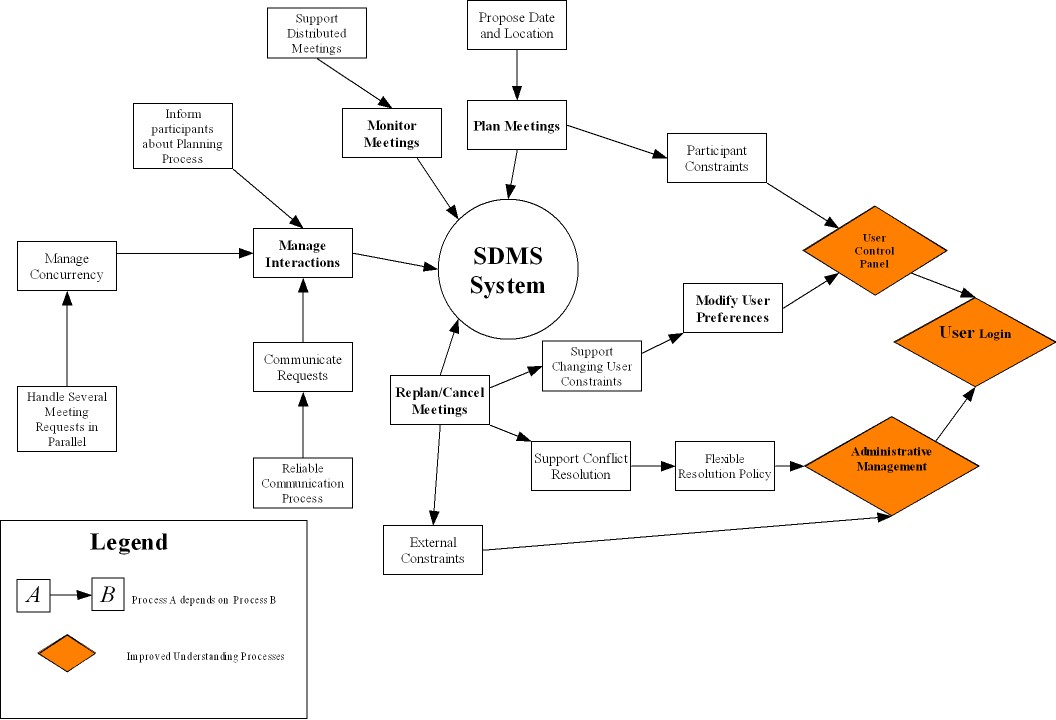


Figure 2: System Functional Requirements Dependency Diagram

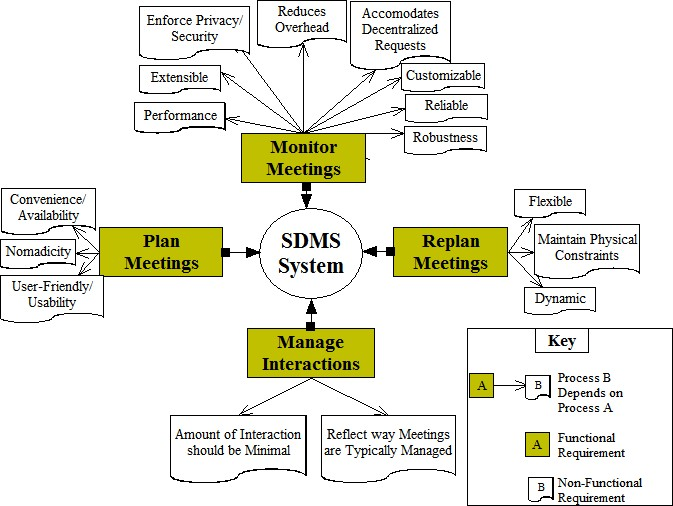


Figure 3: System Non-Functional Requirements Dependency Diagram

Appendix B: Prototype

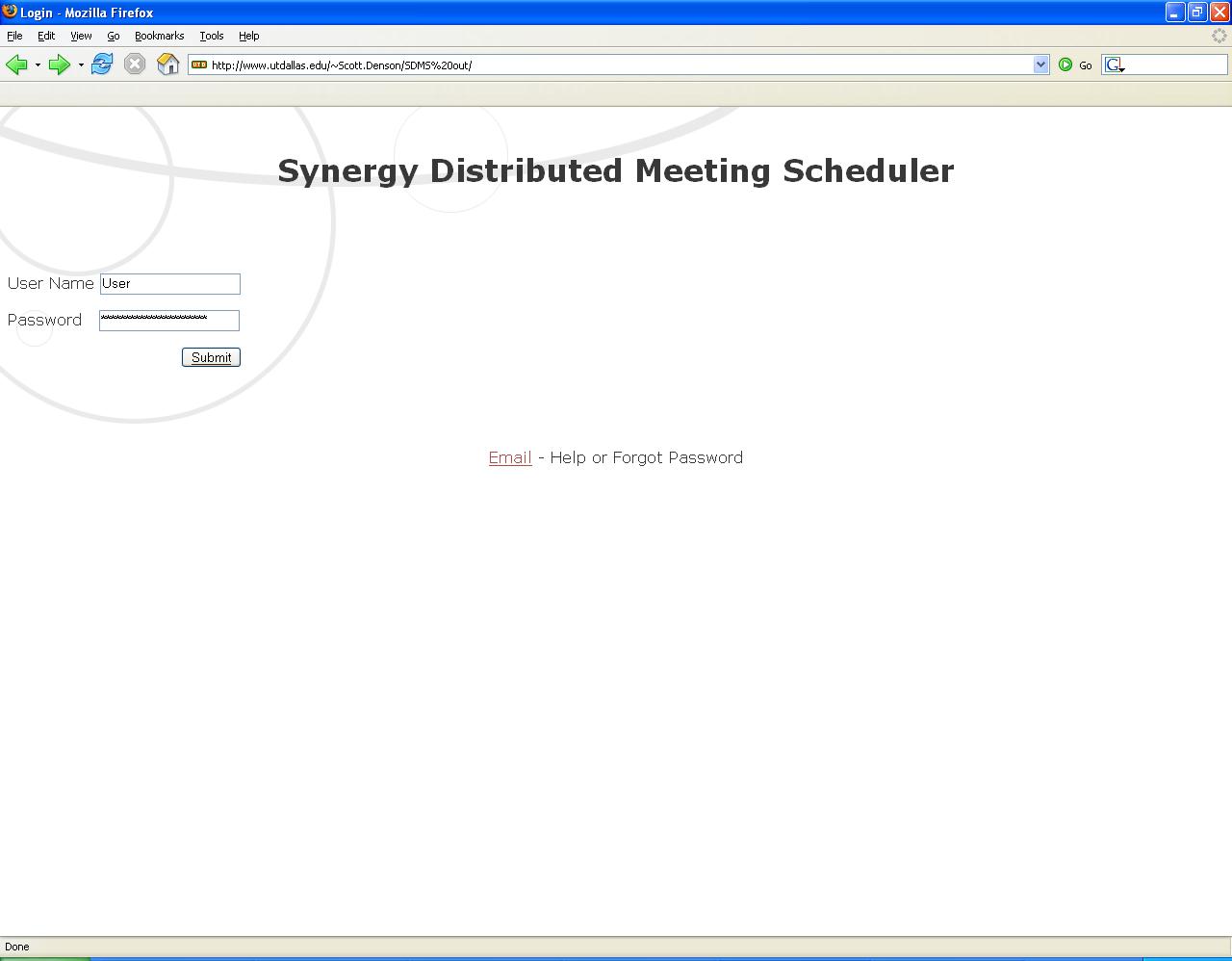


Figure 1: User Login Page

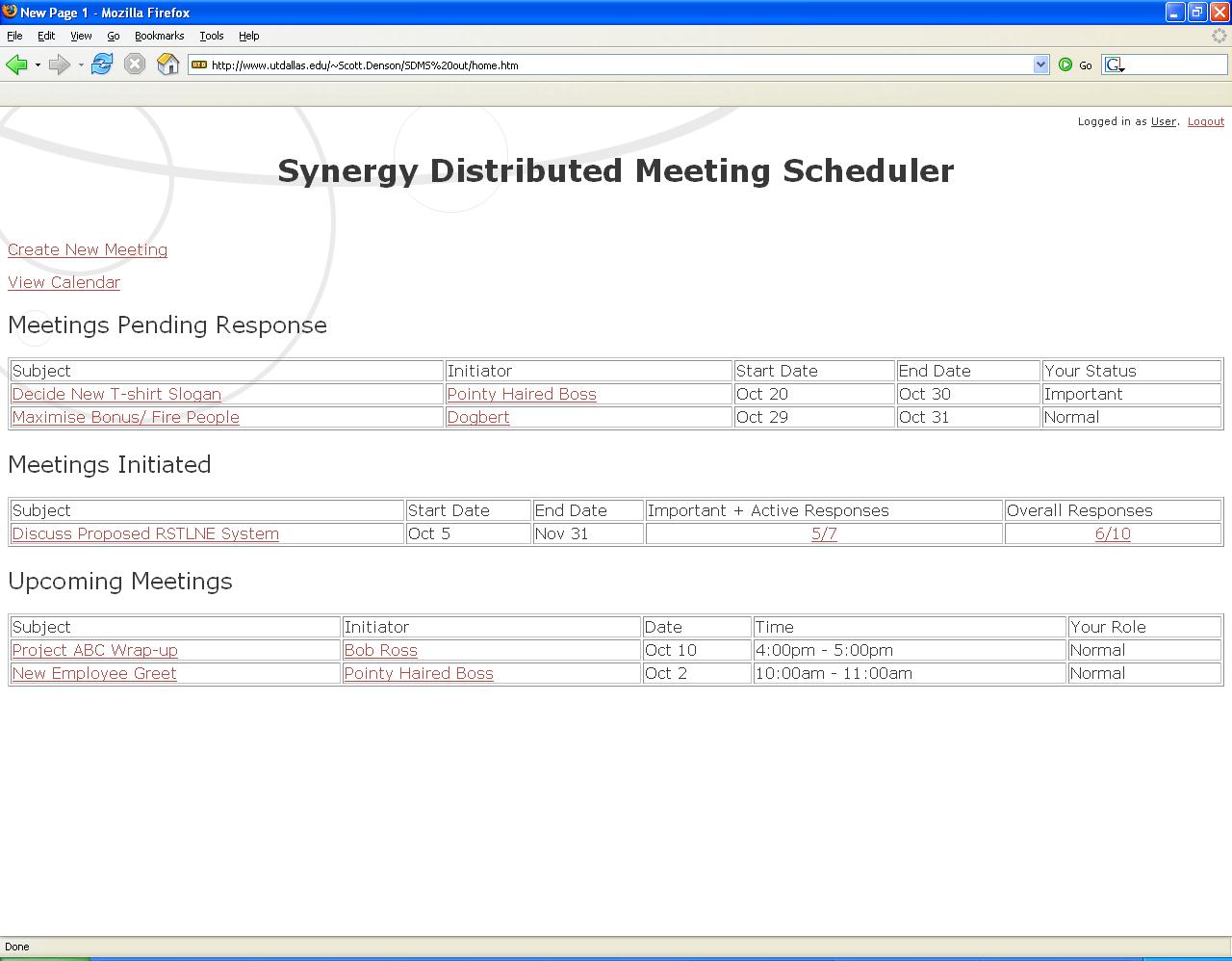


Figure 2: User’s Home Page—displays meetings relevant to the user

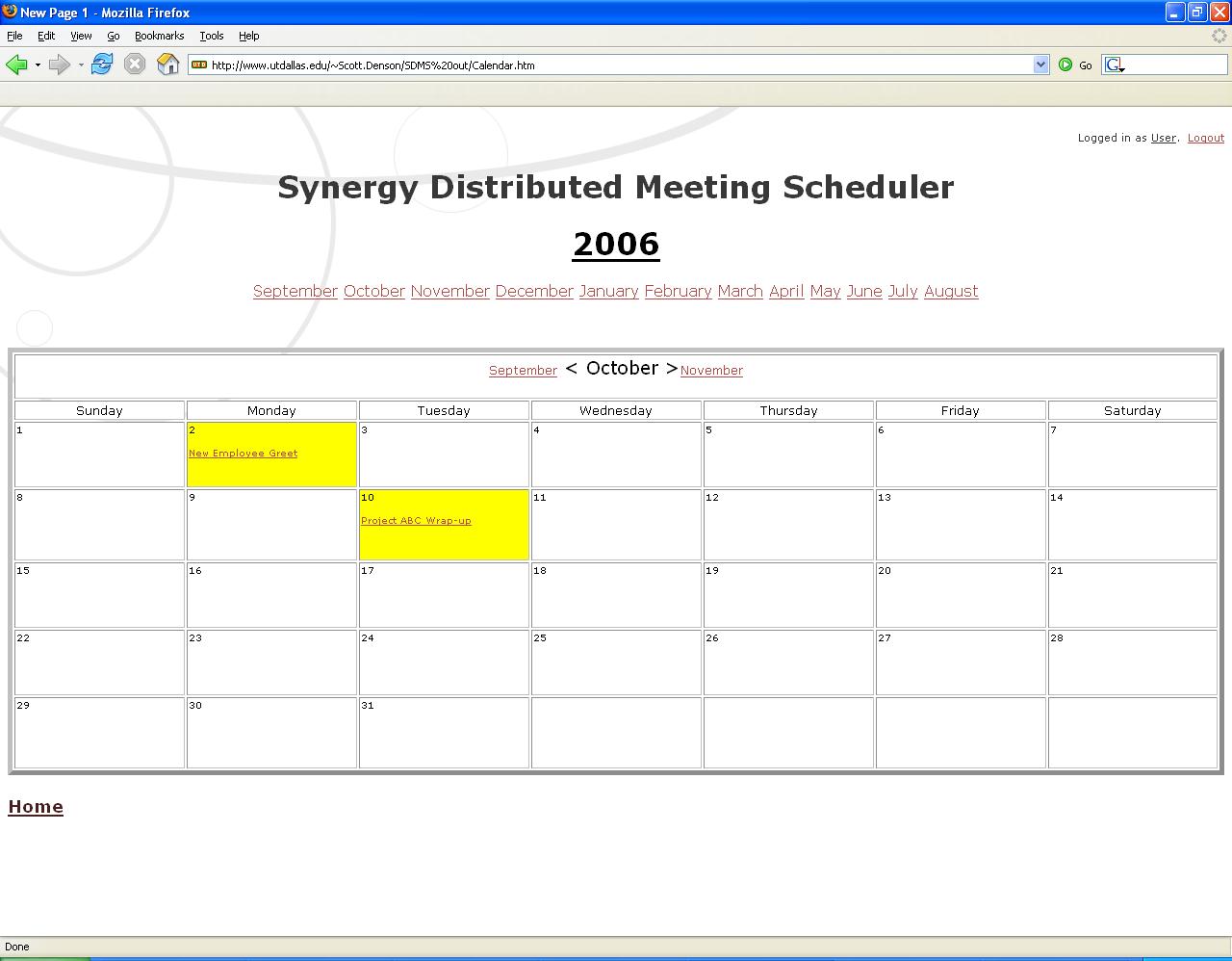


Figure 3: Calendar View—provides user with a pictorial view of meetings

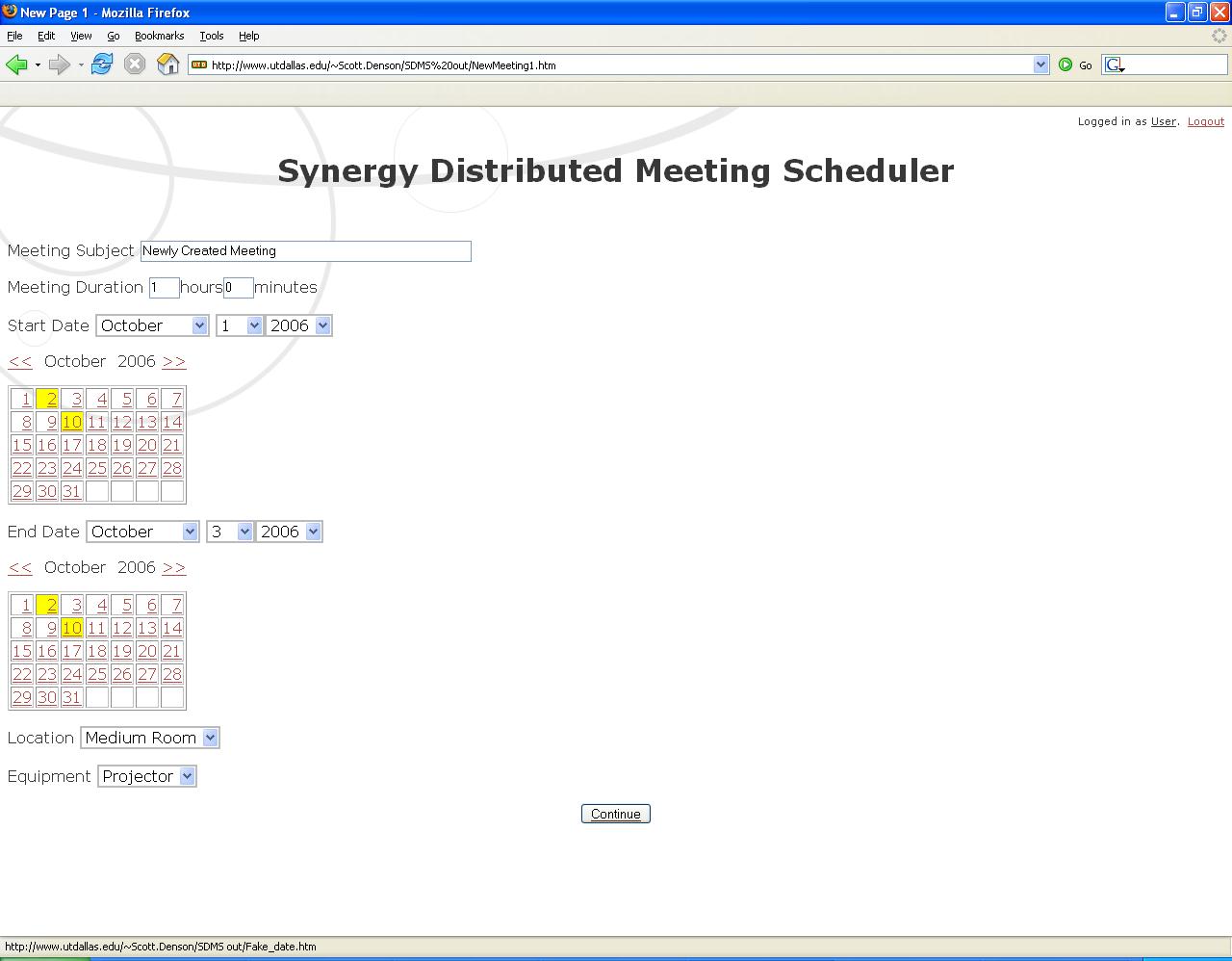


Figure 4: Meeting Creation Page—allows for temporal and physical constraints of the meeting

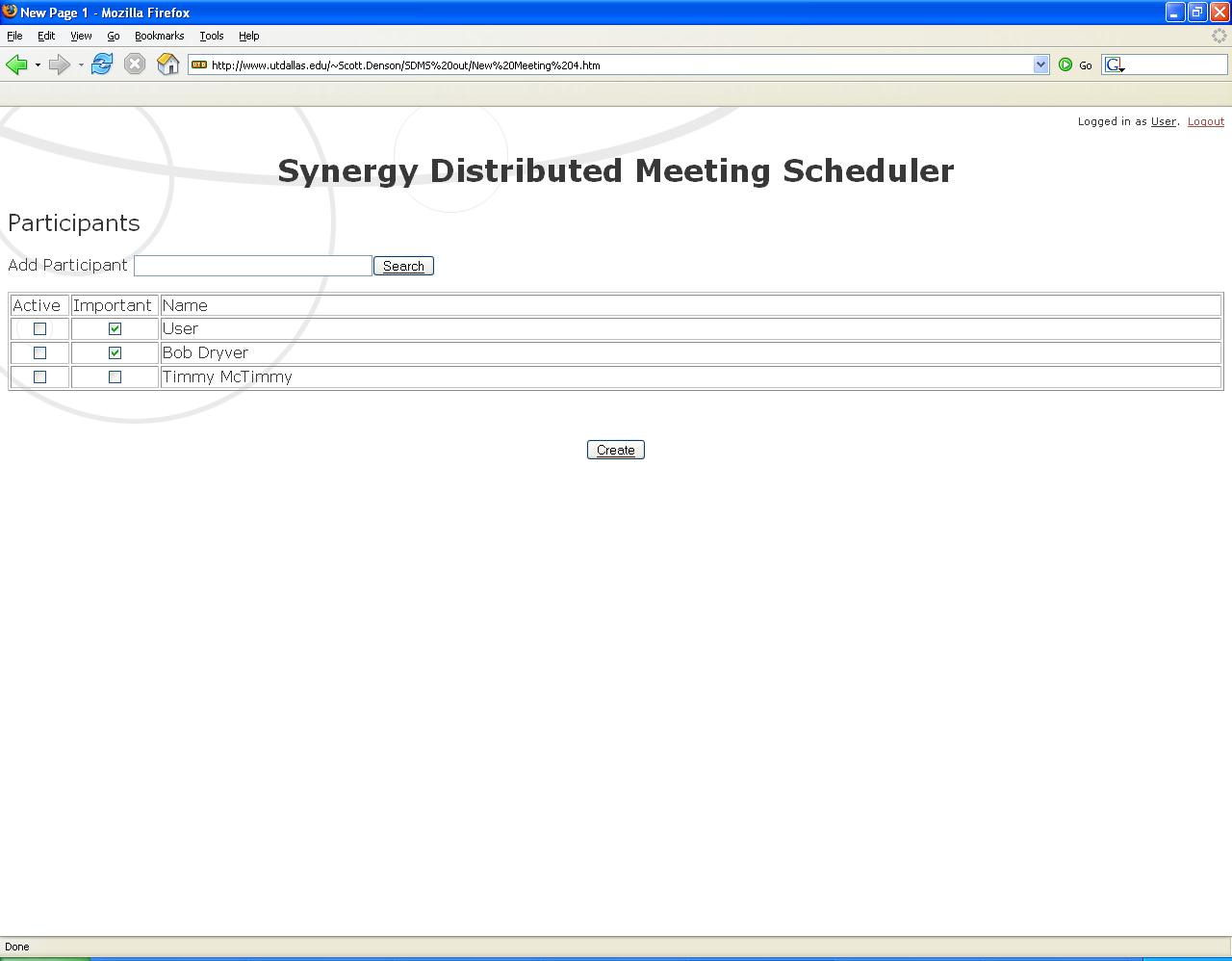


Figure 5: Initiator adds Attendees to the invite started in last step

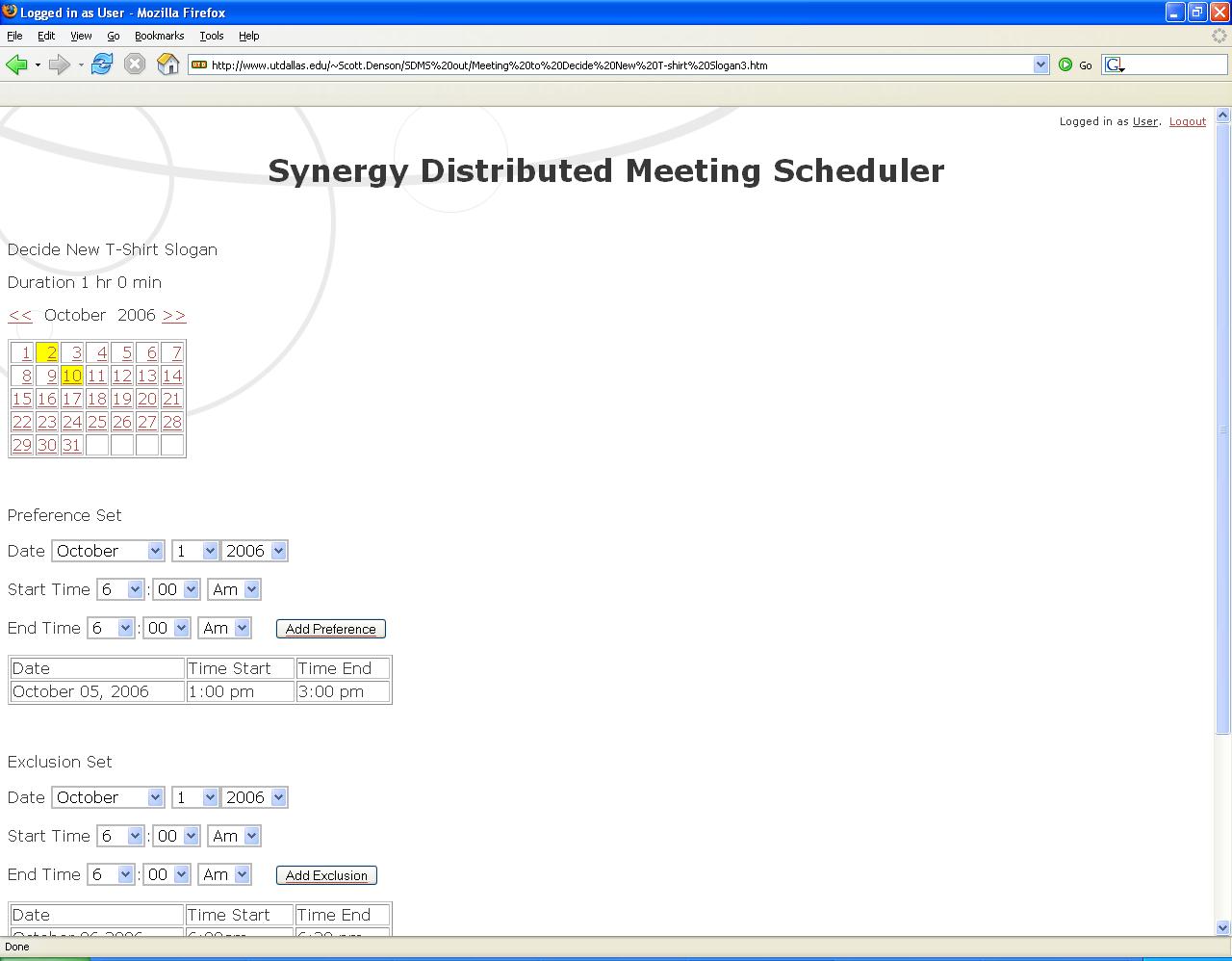


Figure 6: View of User’s Response

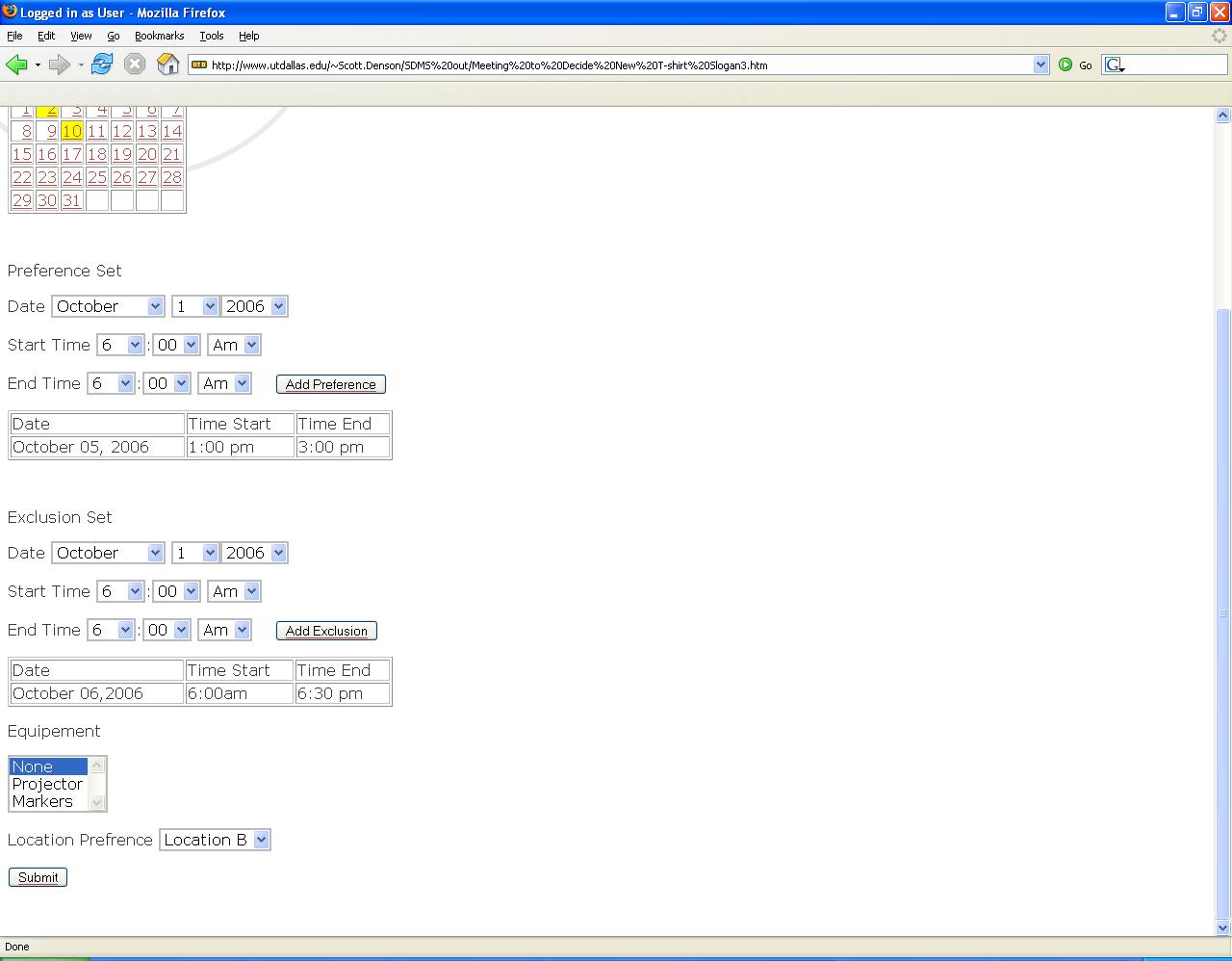


Figure 7: Compl

The document may be used as a contract between the customer and the designers to formalize the agreement about the specifications and requirements for the product. It may also be used by the designers and developers in aiding them to achieve a better understanding of the system, and to create a product matching the customer’s expectations