

Web-based Meeting Scheduler
Phase 1: Final
CS 6361 Section 001 Spring 2010

Project Plan
Version 1.1

March 25, 2010

Team Awesome

Team Website: <http://www.utdallas.edu/~rhb081000/6361/>

Phase 1 Leaders: Rachel Bock and Victor Isbell

Team member	Student ID	Email Address
Rachel Bock	rhb081000	rhb081000@utdallas.edu
Amy Polcari	ajp081000	ajp081000@utdallas.edu
Ramon Rivera	rar096020	rar096020@utdallas.edu
Chih-Lin "Leo" Cheng	cxc094120	cxc094120@utdallas.edu
Swathi Kandimalla	sxk083300	sxk083300@utdallas.edu
Nikhil Mishra	nkm090020	nkm090020@utdallas.edu
Victor Isbell	vri021000	vri021000@utdallas.edu
Ruben "Gabe" Cavazos	rgc061000	rgc061000@utdallas.edu

Submitted to:
Dr. Lawrence Chung
Associate Professor,
Department of Computer Science
The University of Texas at Dallas

Revision History

Version	Primary Author(s)	Description of Version	Date Completed
1.0	R. Bock	Interim Project Plan	March 4, 2010
1.1	R. Bock	Finalized document	March 24, 2010

Table of Contents

Revision History	i
Table of Contents	ii
1. Introduction	1
1.1 Project overview.....	1
1.2 Project deliverables.....	1
1.3 Evolution of this document.....	1
1.4 References.....	1
1.5 Definitions, acronyms, and abbreviations.....	2
2. Project organization	2
2.1 Process model.....	2
2.2 Organizational structure.....	2
2.3 Organizational boundaries and interfaces.....	3
2.4 Project responsibilities.....	4
3. Managerial process	4
3.1 Management objectives and priorities.....	4
3.2 Assumptions, dependencies, and constraints.....	4
3.3 Risk management.....	5
3.4 Monitoring and controlling mechanisms.....	5
4. Technical process	5
4.1 Methods, tools, and techniques.....	5
4.2 Software documentation.....	6
4.3 Project support functions.....	6
5. Work elements, schedule, and budget	6

1. Introduction

1.1 Project overview

For any organization to function efficiently, personnel must be able schedule meetings without wasting vast amounts of time determining a time and location to fit everyone's schedule. This is a project plan describing the Web-based Meeting Scheduler (WMS), a program that will assist in automating the process of determining the best times for various participants to meet.

The project involves creating a scheduler which will allow users to initiate meetings and acquire information about potential meeting attendees' time preferences to find an optimal meeting time. WMS users will update their schedules with times they are unavailable as well as times they prefer to meet to help guide users who are initiating meeting to find the best available meeting time.

1.2 Project deliverables

The deliverables for this project and their deadlines are:

Phase	Deliverable Name	Deadline
0	Preliminary Project Plan Minutes of Meetings	01/28/2010
1.1	Project Plan Software Requirements Specification Minutes of Meetings Presentation Team Emails and Agendas	03/02/2010
1.2	Project Plan Software Requirements Specification Minutes of Meetings Team Emails and Agendas	03/25/2010
2.1	Interim Project 2	04/15/2010
2.2	Final Project 2	4/27/2010

1.3 Evolution of this document

The following sections will be upgraded as the project goes on:

1. References: update as necessary
2. Definitions, acronyms, and abbreviations: update as necessary
3. Project Organization: updated when team leader is assigned to different project phases.
4. Technical Process: update as necessary

1.4 References

1. Course Homepage: <http://www.utdallas.edu/~chung/CS6354/>
2. Project Overview: <http://www.utdallas.edu/~chung/CS6354/Project.doc>
3. Team Website: <http://www.utdallas.edu/~rhb081000/6361/>
4. Prototype Website: <http://ramon.grugel.com/>

1.5 Definitions, acronyms, and abbreviations

WMS: Web-Based Meeting Scheduler

DR: Domain Requirements

FR: Functional Requirements

NFR: Non Functional Requirements

Exclusion set: A set of times when a Potential Meeting Attendee cannot attend the meeting.

Preference set: A set of times when a Potential Meeting Attendee prefers to have meetings scheduled.

Date range: A time interval selected by the Meeting Initiator in which to schedule a meeting.

Strong conflict: When there are no times within in the date range when all potential meeting attendees are able to attend the meeting. All times in the date range fall within at least one Potential Meeting Attendee's exclusion set.

Weak conflict: When there are no times within the date range when all Potential Meeting Attendees prefer to have a meeting scheduled. No times in the date range fall within all of the Potential Meeting Attendees' preference sets.

No Conflict: The state when neither a Strong or Weak conflict exists.

Meeting Initiator (MI): A person who creates and schedules a meeting through the WMS.

Important Participant (IP): A person whose presence is important for the meeting but who is not actively participating (for example, a host).

Active Participant (AP): A person who will be actively engaged in the meeting (for example, a speaker or presenter) and is necessary for the meeting to take place.

Regular Participant (RP): A person who will be at the meeting but not actively participating or hosting.

Potential Meeting Attendee (PMA): An important, active, or regular participant who has been invited to the meeting.

2. Project organization

2.1 Process model

For this project, Team Awesome will adopt an iterative software development process and an Agile-like philosophy of teamwork. The WMS project consists of two phases. During the first phase, the team will attempt to determine stakeholder needs, define software requirements, and build a prototype of the WMS software. The requirements and prototype will be used to validate and improve the team's understanding of stakeholder needs. Using the knowledge gained from phase one, the team will iteratively improve upon the proposed WMS solution during phase two. Additionally, the team will detail a more in-depth WMS solution than in phase one.

Throughout the project, Team Awesome will operate using Agile Methods in order to promote a collaborative, flexible, and productive environment for all team members. This includes frequent team meetings -- at least one team meeting a week; varying and flexible roles -- an individual contributes as reviewer, developer, and leader; parallel work effort -- requirements, prototyping, and documentation occur simultaneously; and simulated customer interaction -- all team members are asked to think and act like a customer.

2.2 Organizational structure

The project team members are:

- Rachel Bock

- Amy Polcari
- Ramon Rivera
- Chih-Lin "Leo" Cheng
- Swathi Kandimalla
- Nikhil Mishra
- Victor Isbell
- Ruben "Gabe" Cavazos

Rachel Bock is the team leader for Phase 1.1 of the project. Victor Isbell and Rachel Bock will be co-leaders for Phase 1.2, 2.1, and 2.2 of the project. The team leaders will assign individual responsibilities for each phase of the project.

Phase 0 Deliverables	Leader	Developers	Reviewers
Preliminary Project Plan	R. Bock	R. Bock, A. Polcari, R. Rivera, S. Kandimalla, V. Isbell, N. Mishra, R. Cavazos, C. Cheng	

Phase 1 Deliverables	Leader	Developers	Reviewers
Project Plan	A. Polcari	A. Polcari, R. Bock	C. Cheng, R. Cavazos, V. Isbell, R. Rivera, S. Kandimalla, N. Mishra
Software Requirements Specification	R. Bock	R. Bock, A. Polcari, V. Isbell, S. Kandimalla, N. Mishra, C. Cheng	R. Rivera, R. Cavazos
Prototype	R. Rivera	R. Rivera, R. Cavazos	R. Bock, A. Polcari, V. Isbell, S. Kandimalla, N. Mishra, C. Cheng
User Manual	V. Isbell	V. Isbell, R. Cavazos	R. Rivera, C. Cheng, R. Bock, N. Mishra, A. Polcari, S. Kandimalla

2.3 Organizational boundaries and interfaces

The team leaders will assign other team members as sub-leaders who are responsible for overseeing the completion of specific deliverables. This will allow multiple team members to act in a leadership capacity.

The team leaders will divide Team Awesome into groups based on each sub-phase of the project plan. Based on team progress and project needs, the team leaders may adjust responsibilities weekly. The overall goal is to have each team member contribute to every component of the project as a sub-leader, a developer, or a reviewer. The role of a sub-leader is to oversee the

creation and/or updates of a deliverable. The sub-leader is responsible for planning and organizing the completion of their deliverable on schedule. Additionally, the sub-leader is available to developers and reviewers for questions. If decision conflicts arise, the sub-leader has the authority to decide the outcome of the conflict. Developers are responsible for creating the content of the deliverable. Reviewers are responsible for checking the deliverable for spelling, grammar, consistency, and accuracy. Since an agile-like philosophy is practiced, the organizational structure exists to guide the team rather than to define rigid boundaries. Thus, any team member can consult with the team leaders about questions or conflicts.

2.4 Project responsibilities

All team members will be involved in each sub-phase of the project as a team leader, sub-leader, developer, or reviewer. An individual may serve in multiple roles during each project sub-phase. The team leaders will assign roles and responsibilities at the beginning of each sub-phase. Assignments of roles and responsibilities are based on project needs, team member skill sets, contributions in previous phases of the projects, and an individual's desire to contribute.

3. Managerial process

3.1 Management objectives and priorities

Management's primary objective is to ensure Team Awesome delivers a quality WMS system to the customer that meets the needs of the identified stakeholders and is on time and on budget. As Team Awesome's reputation and ability to win future business depends on the success of this project, understanding and documenting a complete and accurate set of software requirements is management's top priority. To successfully do this, management has outlined the problem statement and a high-level schedule including milestones, goals, and deliverables. The schedule offers a set of boundaries for Team Awesome to operate within. Additionally management acts as a resource for guidance on conflict resolution, project scope, customer feedback, and organizational goals. Management has given Team Awesome the latitude to organize and operate as the team prefers, but recommends the following principles be followed:

- Encourage team leadership, participation, and communication
- Be critical of work but not the work's creator
- Resolve conflicts within the team
- Distribute and assign work equally
- Adapt an agile-like work environment

3.2 Assumptions, dependencies, and constraints

Assumptions:

- Since no actual customer exists, the role of the customer is a synthesis of management and team members' opinions.
- Since Team Awesome has accepted this project pro bono, budget is not an issue.

Dependencies:

- Customer/user input is provided by management on behalf of the customer.
- Customer/user input can occur throughout the process.
- Customer/user input may change throughout the process.

Constraints:

- No flexibility exists in the project deliverable schedule.

- Team members' availability to attend team meetings is restricted.

3.3 Risk management

<i>Risk</i>	<i>External?</i>	<i>Probability</i>	<i>Possible Impact</i>	<i>Description</i>
Resource shortfall: human	Internal	Likely	High	Shortage of human resources, such as knowledge
Resource shortfall: other	External	Likely	Low	Shortage of other resources, such as face-to-face meeting time
Work loss	External	Unlikely	High	Data loss due to computer/human error
Lack of commitment	Internal	Unlikely	High	Group member does not appear to be contributing fully
Workload sharing	Internal	Unlikely	Low	Individual unable to complete task due to emergency/other
Team deadline missed	Internal	Unlikely	High	Team misses a deadline
Requirements change	External	Unlikely	Low	Instructor changes assignment specification

3.4 Monitoring and controlling mechanisms

<i>Risk</i>	<i>Monitoring and Controlling</i>
Resource shortfall: human	Actively identify known unknowns Seek aid from instructor/TA
Resource shortfall: other	To work around meeting time shortage, online forum might be helpful to post and discuss issues
Work loss	Normal operating routine for backing up data
Lack of commitment	Set and follow professional standards If necessary, stage leader discusses matter in private with individual
Workload sharing	Identify emergencies as soon as possible to team Stage leader distributes work in chunks among remaining team members
Team deadline missed	Analyze difficulties or workload bottlenecks Revise future schedule and work distribution
Requirements change	Analyze necessary changes to existing work Alter future schedule and workload assignments to fit new needs

4. Technical process

4.1 Methods, tools, and techniques

Methods employed for successful execution of the project are:

- Each phase has a team leader or leaders responsible for addressing project requirements, assigning work to the team and conducting weekly meetings.
- Team members have to complete the work assigned on time.
- Team leader or leaders will review the final project that is to be delivered.

Tools used for application development, and communication among team members and documentation:

- Google docs will be used for the composing and editing of project documents.
- All project related documents will be posted on our team's website:
<http://www.utdallas.edu/~rhb081000/6361/>
- Project documentation will be done in Microsoft Office Word.
- Project presentation will be done in Microsoft Office PowerPoint.
- Google groups will be used for project discussions and collaboration between team members.
- Rational Rose software will be used for representing architectural and conceptual drawings.
- PHP scripting language will be used for application development.

Techniques to be employed vary according to the deliverable and will be addressed as needed.

4.2 Software documentation

The following documentation will be produced:

- Preliminary Project Plan
- Project phase 1 Interim updated Project Plan
- Project phase I Interim Software Requirements Specification document
- Project phase I Interim presentation document
- Project phase 1 Final updated Project Plan
- Project phase I Final Software Requirements Specification document
- Project phase II documents
- Minutes of Meetings
- Emails and Agendas for Team Meetings

4.3 Project support functions

Project Support Functions will can be broken down in the following categories:

- Technical
 - Testing
 - Software Quality Control
 - Verification and Validation
- Non Technical
 - Creating Work Documents

These will be executed at the appropriate stages by members of the team as assigned by the team leaders.

5. Work elements, schedule, and budget

The work elements (packages) are outlined by the course and are scheduled as follows:

Preliminary Project Plan: Due 01/28/10

Project I Interim: Due 03/02/10

Project I Final: Due 03/25/10

Project II Interim: Due 04/15/10

Project II Final: Due 04/27/10

They consist of work documents to be turned in to the instructor. Each work document will rely on work done within the project and will act as milestones. Additional scheduling will be done by team leaders.

Budgetary constraints are not applicable for this project as we will be developing and working on the project without payment. Any computer resources necessary will be provided by team members.