**Meeting Scheduler**

**Software Requirements Specification**

**<11/02/15>**

**Ben Shaw and Nikhil Vedi**

**Lead Software Engineers**

**Prepared for**

**SECM**

**Lecturer:** **Dr Mehmet Bülent Özcan**

**Sheffield Hallam University**

Contents

1. Introduction............................................................................................................................Page 3

* 1. Purpose
  2. Scope
  3. Definitions
  4. Overview

2. General Description...............................................................................................................Page 4

* 1. Product Functions
  2. User Characteristics
  3. General Constraints

3. Requirements........................................................................................................................Page 5

* 1. Functional Requirements
  2. Non – Functional Requirements
  3. External Interface Requirements
  4. Design Constraints
  5. Logical Database Requirements

1. **Introduction**

1.1 <Purpose>

This SRS is intended to inform those working on the Meeting Scheduler program on the requirements they need to meet. It will also be presented to the client before programming begins to ensure that the requirements are accepted and agreed.

* 1. <Scope>

The overall scope for this project is to produce a fully functioning meeting scheduler program. The Meeting scheduler will allow employees to arrange meetings with other employees with ease, the program will inform those who have been scheduled a meeting the date, time and location via the company’s internal email system. The system will only allow meetings to be arranged for employees who have a company login and password. At present a guest being able to use the program is out of scope. This program will improve meeting arrangement within the company massively. The MS should run efficiently enough so that it actually improves meeting arrangements for the client. If it runs in a way that it would be quicker for the client to arrange meetings verbally then the project will have failed, this is our minimum scope.

* 1. <Definitions>

**MS –** This is an abbreviation of Meeting Scheduler

**SRS –** This is an abbreviation for System Requirement Specification.

* 1. <Overview>

The rest of this SRS will contain the requirements needed for this project to be completed to the point that it can be used within the company and the MS will meet all requirements given by the client. The SRS will give the programmers a set of non-functional requirements that will need to be met so that the program will run efficiently enough so that it is actually useful for the client and not just a hindrance.

.

1. **General Description**

2.1 <Product Functions>

The main overall function of the program is for it to arrange meetings based on a set of data provided by the users. User preferences will be submitted to a database and then a meeting will be arranged with selected people at a time and date that best fits all involved. The MS will make all relevant people aware of said meeting via the company email. The MS should be constantly updating whenever a user changes their preferences. When no suitable meeting slot can be found the system will assist the user in resolving the issue so that a meeting can be arranged with as little input from the user as possible, the system should do all the work. The MS will need to allow the client to set the priority of an individual. So attached with their login credentials will be their status within the company. This will allow the Meeting Scheduler to prioritise these individuals when arranging meetings.

* 1. <User Characteristics>

The users of the Meeting Scheduler will on the whole all only have basic computer skills. They will all be able to use simple programs so the MS will need to be kept as simple as possible but still be able to perform all the tasks it needs to. The system needs to be usable with a basic keyboard and mouse as this is what the users will be using and are experienced in doing so. Basic training in using the system will be provided to the users and this will need to be enough to allow the users to work with the MS fully and effectively.

* 1. <General Constraints>

The Meeting Scheduler program needs to be designed in C# using Microsoft Visual Studio. It will need to be a windows program that with minimal work is transferable to other OS's. The application needs to be small enough so that it can be pushed to devices over the network. The application needs to run in a window that fits onto notebooks, laptops and monitors of varying sizes. It also needs to take up minimal memory as it may be running as a background program for prolonged periods of time. The software will link up with the login database using SQL.

1. **Requirements**

**3.1 Functional Requirements**

3.1.1<Allow user to input data>

The meeting scheduler program should allow the users to input data which will let the system know the dates and times they're available for meetings, when they would prefer them and when they're unavailable.

Inputs: User fills in grid showing a number of dates and times, stating whether they prefer that meeting time, can attend but don't prefer it or if they can't make it at all.

Processing: The preferences are passed from the program and stored in a database so they don't have to be resubmitted every time the user logs in.

Outputs: A simply message letting the user know their preferences have been stored.

* + 1. <Arrange Meeting>

The meeting scheduler needs to use data provided from users to arrange meetings that everyone can attend and that satisfies the most people in terms of preferred dates being met.

Inputs: Pre submitted preferences.

Processing: The program uses the pre submitted user preferences to calculate the best available meeting date (if one exists) if a suitable meeting date is not found conflict resolution will occur.

Outputs: A window stating the date, time and location of the meeting.

* + 1. <Resolve conflicts stated by pre-defined resolution policies stated by the client>

The program will need to be able to resolve any conflicts which are stopping a meeting from being arranged. It should let the user know what resolution will remove and conflicts and allow the conflicts to be resolved within the program. The resolution policies agreed are as follows. Remove an employee who is not noted as 'must be present'. Swap out an employee for someone else in their respective department for someone who can attend a date which all other meeting attendees can make. Ask the users creating the conflicts to submit a new set of preferences.

* + 1. <Users preferences should be kept private>

User’s preferences shouldn't be shared with other users automatically. The program will need to allow preferences to be shared but only at the user’s discretion. If they want preferences kept private they should be.

* + 1. <The Meeting Scheduler will need to keep people informed>

The meeting scheduler not only needs to let users know when a meeting has been arranged via email. It also needs to let the users who the information is relevant too in the loop also. Even if a person isn't attending the meeting, if the information is relevant to them they should also receive emails.

* + 1. <Meetings arranged should be physically possible>

When a meeting is arranged it has to be possible for that meeting to actually take place. Meetings shouldn't be arranged in certain locations if a participant isn't anywhere near that location at that time.

* + 1. <People can't be double booked>

If a user has a meeting scheduled at a certain slot, the system should be aware and make that user unavailable for that date and time.

* + 1. <The Scheduler should accommodate for partial attendance>

If a user can only attend part of a meeting the program should be able to accommodate this and make everyone aware of it.

* + 1. <Allow Managers to delegate to staff>

The system should allow managers who are unable to fit a meeting in delegate to their staff who can attend the date.

* + 1. <The Meeting Scheduler should make meetings re – arrangeable>

Once a meeting has been arranged the system should allow for the meeting to be re planned if for some reason an attendee’s availability changes.

* + 1. <The MS needs to allow various sized meetings to be arranged>

The meeting being created should be variable in size depending on how big the user wants the meeting to be. Meeting arranged should not be of a fixed size.

Inputs: User will be prompted to enter the size of the meeting they wish to create.

Processes: MS will then create a form of the same size allowing this to be possible.

Outputs: Relevant meeting will be created with the correct amount of people included.

**3.2 Non - Functional Requirement**

* + 1. <Performance>

The meeting scheduler needs to be able to arrange meetings within a second, if a meeting can't be arranged the resolution window needs to open up within the same time frame. When the program is opened up the time taken between login credentials being entered and the MS loading needs to be kept below 15 seconds. Meetings also need to be arranged with less than 20 user inputs being needed.

* + 1. <Reliability>

The systems downtime must not exceed more than 5 minutes per day (within working hours) Outside of working hours downtime must be kept below 30 minutes a day.

* + 1. <Availability>

The program is available to all employees of the company who require it. The company will be able to add new logins to the database meaning when new employees join the company they can be added to the system straight away with ease.

* + 1. <Security>

The system will be licensed for as many users as the company has paid for. Each user will have a unique log on which they and only they can use. If you don't have a log in you can't use the program.

* + 1. <Maintainability>

The system needs to be easy to maintain so our code needs to be kept as simple as possible. It should be maintained in a way that allows the downtime to be kept to a minimum. Any maintenance on the system should be able to be completed in a single night so that work isn't impacted by system downtime in office hours.

* + 1. <Portability>

The program needs to be portable in the sense that it can be installed on any desktop or laptop. There is two parts to the program, the program itself and then a database which contains all of the user logins. These will be needed to be packaged together and then the application could be pushed too all machines over SCCM or something similar.

**3.3 External Interface Requirements**

3.3.1 <User Interfaces>

The user interface needs to be simple enough so that even inexperienced computer users can use the system with only basic training. The interface needs to be designed in a way so that there is no time wasted using the system. The maximum amount of actions required to arrange any meeting should be kept to 5.

3.3.2 <Hardware Interface>

The MS will be operated using a combination of mouse and keyboard input. It will also be designed in a way so it can be operated solely using the keyboard. However it won't run as efficiently or be as simple to use.

3.3.3 <Software Interface>

The meeting scheduler needs to link up with a database for it to run as this is how it will enable to login to work and also enable preferences to be stored within a database meaning the user doesn't have to enter their preferences every time they load up the software. They can simply just update their preferences when they change.

3.3.4 <Communication Interfaces>

The system needs to be linked in with the company email, when a meeting is arranged it needs to let people know that a meeting has been arranged. This email needs to create a group email that can then be replied to through the company email. Improving the communications throughout the company overall.

* 1. **<Design Constraints>**

The program needs to run on all operating systems of which the company uses. It will also need to be designed in a way so with minimal effort it will be transferable to future OS's. The program will need to run with the login details that employees already use to access their company computer. The program will need to run on both laptops and desktop as this is what the client uses currently. As of yet no request has been made for a smart phone version of the product. This however should be kept in mind when designing the program for the future. If the client ever decides this is what they would like.

* 1. <**Logical Database Requirements>**

The program will have to link up with the database that the company already uses for their employee login. This will mean that the same credentials are needed to log into the system and then to login to the meeting scheduler itself.