ELEN4010 - Software Development III
Meeting Scheduling System
Software Requirement Specification

Ari Croock (718005)
Kanaka Babshet (678851)
Alice Yang (597609)
Daniel Weinberg (547937)

May 14, 2016

1 Purpose

This document details the system requirements specification (SRS) for an online meeting scheduling system for students and academics, commissioned by the School of Electrical and Information Engineering. The project aims to improve the meeting scheduling process by reducing the manual communication (emails) required to agree upon a meeting time.

2 Project Scope

2.1 Basic Scope

The system will at a minimum allow students to select an academic with whom they would like to meet and view the academic's schedule so that a suitable time can be chosen. New meetings can only be booked during the academic's free calendar time. However a student can request to join an existing meeting. Each meeting has additional details associated with it, such as meeting duration, the type of meeting and meeting details.

The academic (the administrator) must be able to view their schedule and manually book meetings, bypassing the scheduling system. Additionally, the administrator should be able to specify times when they are available, which could be fetched from their Google calender. Administrators should be able to edit their own schedules, while students cannot.

The basic scope is required for a Minimum Viable Product (MVP) in order to show that the system will provide some value to the School.

2.2 Extended Scope

Once the basic system scope has been implemented, various additional features will be developed, such as: being able to reschedule meetings, create waiting lists when desired time slots are not available, scheduling rules and integrating with Google Calender.

3 Project Overview

3.1 Assumptions

- All meeting requesters are members of the School of Electrical and Information Engineering
- All meeting requests are valid, i.e., a student will not book a meeting for a course in which he is not enrolled
- Meetings will run according to the meeting schedule
- Academics will not change their free time after a meeting has been scheduled
- All meetings will take place in the lecturer's office
- Adequate storage is available to run the system for the foreseeable future

3.2 Constraints

- No dedicated server-side web development frameworks may be used
- Python must be used as the server-side scripting language
- The Apache Web Server must be used
- The shared Git repository on the provided server must be used

4 System Features

4.1 High Level User Stories (Epics)

1. **View a schedule:** An academic can view his own schedule on a calender
2. **Book a meeting bypassing the scheduling system:** An academic can book meetings directly in case a student does not make use of the online scheduling system
3. **Book a meeting with an academic:** Students can request to meet when an academic is available, and can set details about the meeting
4. **Allocate free time for meetings:** An academic can select time slots when he is available to meet
5. **Join an existing meeting:** Students can join an existing meeting if this is allowed by the academic
6. **Cancel a meeting:** Students can cancel a meeting such that the time slot becomes clear
7. **Set meetings to recur:** An academic can set certain meetings to repeat over a certain period
8. **View reports of time spent on meetings:** An academic can view reports summarising time spent in meetings and with whom

4.2 Low Level User Stories

- **Book a meeting with an academic:**
  - A student must select an academic to meet with
  - A student must enter his name and student number, the meeting subject and meeting duration
- **Allocate free time for meetings:**
  - An academic can set periods of time when he is available to meet
  - (An academic can set recurring periods of time when he is available to meet)
  - (An academic can set periods of free time to repeat so that they don't need to enter the same periods multiple times)
- **View reports of time spent on meetings:**
  - The academic can set the time period of the report so they can view meeting summaries over that period
  - The academic can filter by type of meeting so that they can view a summary of a particular type of meeting

4.3 Use Cases for Primary Features

- **Book a meeting with an academic:** See Table 1
- **Allocate free time for meetings:** See Table 2
- **View reports of time spent on meetings:**

5 External Interface Requirements

6 Other Non-Functional Requirements

7 Testing

7.1 Acceptance Tests

Table 1: Table illustrating the details of the use case for the booking of a meeting with an academic

Use Case	**Book a meeting with an academic**
Description	The purpose of this feature is to enable users within the faculty to book a meeting with a specific academic
Primary Actor	Student or academic
Scope	School of Electrical and Information Engineering
Level	User goal
Stakeholders and Interests	1. Student - wants to book an appointment with an academic 2. Academic - wants to book an appointment with another academic or allocate time within which they are available 3. University - wants to view all booked appointments
Precondition	The user is registered on the system and is on the required webpage
Minimal Guarantee	The student will be able to determine if a selected timeslot is available or not
Success Guarantee	The academic, student, and school administration can view that the academic was available for discussion within the selected timeslot and the meeting has therefore been booked
Main Success Scenario	1. User opens the webpage 2. User searches for the specific academic with whom the meeting is desired 3. User checks the free time allocated by the academic 4. User selects the desired free timeslot for a meeting 5. The request is approved and the meeting is booked
Extensions	The user is not available during the free times allocated by the academic. Renegotiate meeting time
Variations	User may contact the academic directly (for example via email or SMS) in order to request a meeting and the academic therefore directly accesses the system and books the meeting for the user
Schedule	Due date : Release 1.0
Open Issues	• What if the academic does not allocate any free time? • What if the student requires an urgent meeting? • What if an academic gets multiple meeting requests at the same time?

Table 2: Table showing the use case for allocating free time for students to book meetings

Use Case	**Allocate Free Time for meetings**
Description	The goal behind the allocation of free times by the academic is to provide the students with an indication of when they are able to schedule meetings.
Characteristic Information	
Primary Actors	Academic staff members
Scope	School of Electrical and Information Engineering
Level	User goal
Stakeholders and Interests	Academic Staff - wants to be able to specify free time for students to schedule meetings. Students - want to be able to book meetings with an academic by seeing their free time schedule
Precondition	Students and Academic already have access to the booking system
Minimal Guarantee	No minimal guarantees as the system either functions in its entirety or not at all.
Success Guarantee	Academic can specify free time slots and students are able to view the free time schedule
Main Success Scenario	1/ Academic logs into the system 2/ Academic access calender view 3/ Academic specifies free time slots for students to book meetings 4/ Students can view the academic's free time on the system
Extensions	1/ If for whatever reason, the free time allocated is invalid, the academic is notified
Variations	Academic can fully specify the duration of the meetings available if only set periods are allowed
Related Information	*Priority*: Top *Performance Target*: 2 minutes for academic to allocate free time and for system to update the calender *Frequency*: Relatively infrequent but is depending on how often the lecturer's schedule changes *Superordinate Use Cases*: *Subordinate Use Cases*: *Channel to primary actor*: Web application *Secondary actors*:
Schedule	*Due Date*: Release 1.0
Open Issues	What if academic does not allocate any free time slots

Table 3: Table showing the use case for viewing the reports for the meeting summaries

Use Case	Trigger Switches
Description	The goal behind the report for the meeting summaries is to give a brief description of the time that has been spent by the academics for meetings with various students for a specified period.
Actors	Academic staff members
Scope	School of Electrical and Informational Engineering
Level	User goal
Stakeholders and Interests	The academic - wants to view the compressed format of the meetings for a specified period.
	The head of school - wants to brief summaries of the academics meetings to ensure that the students and academics are communicating aside from lecture times.
Precondition	The academic must be registered as an Electrical (Information) Engineering staff member.
	The academic must have their own calender open.
Minimal Guarantee	The system generates a report with the basic information in the calender, regardless whether a specified period is given or not.
Success Guarantee	The system generates a compressed form of the academics meetings, in which it is displayed to the academic through the platform.
Main Success Scenario	1. Academic opens calender and selects the summary report option 2. The system prompts the user to enter a time period that the academic would like a summary for 3. The academic enters the dates for the desired period 4. The system exacts data from the given time period 5. The system places the extracted data into a simple format that summarises the meetings 6. The system displays the summarised meetings to the academic through the same platform
Extensions	1.a. Academic is unable to open the calender due to bad internet connection: Open the calender at a later stage when internet connection is available 2.a. System unresponsive and does not prompt for user input: Academic must resubmit request 3.a. Academic submits the wrong period dates generate report and then allow user to resubmit the period 4.a. System is unable to exact information: Generate error message to both academic and system administrator
Variations	Given that the system is heavily reliant on the database of the system there is no variations