RealOffice Requirements Document

CS14B023, Rahul Kejriwal
CS14B007, Suhas
CS14B045, Malireddi Sunil Kumar
CS09B043, Shanker Lal Sharma
CS09B031, Chandrakanth

Table Of Contents

I. Abstract

RealOffice is an application software intended for use in the CSE department office to simplify routine procedural tasks. In essence, it shall allow office staff to schedule different kinds of meetings, track meeting requirements, manage room allotments, file meeting reports and also cancel arrangements for cancelled meetings. It shall also remind the staff regarding upcoming events that need attention and facilitate routine backups as replacement for hard paperwork. It shall also integrate data from the CSE department calendar about meetings scheduled from other platforms.

The protocols used by the CSE department office can be easily adapted to a digitized workflow. This document proceeds to build the requirements for such a system, highlighting the current workflow and possibilities for the upcoming software project.

Keywords: meetings, meeting requirements, scheduling, organizer, staff user, requirements specification, use cases, scenarios, UML.

II. Introduction

RealOffice is a software primarily intended for use by department office staff to schedule different kinds of meetings: research scholar meetings, faculty meetings, project evaluations, poster evaluations, visitor seminars, lectures(regular and invited), selection examinations etc. and to manage the unique requirements for each meeting and appropriately allot venues. The project shall allow extensive reporting and backup facilities. The focus for this project has been the CSE department office but it may be generalized to suit any generic office.

The current workflow requires staff to manually collect meeting requests along with prerequisites for infrastructural capabilities and other resources. They are also responsible for scheduling the meeting and it’s venue in addition to arranging the required resources. Post-meeting they are responsible for collecting the meeting report etc. As the number of meetings increases, this process becomes tedious and inefficient for manual operation. This is the motivation behind digitizing this process.

The software shall allow the administrator to configure department details: rooms and their infrastructure support, participating entities like faculty, student and staff members along with time schedules. It shall provide a web interface for scheduling meetings and requesting infrastructure and resource support.

There are other software products like Taskworld available off-the-shelf but are much more complex and are overkill for the intended purpose. The product is designed with a minimal philosophy and is intended to be used with minimal user training.

The remainder of the document will focus on

1) General Description (User Requirements)
2) System Requirements
3) Use Cases
4) Scenarios

III. User Requirements

The main purpose of the software is to automate CSE department office processes which includes scheduling meetings and courses.

The CSE Office deals with various participating entities like students, faculty members, and staff members. The main kinds of meetings organised are:

1) Lectures (Guest lectures will be organized/managed through faculty)
2) Student evaluations of batches
3) Faculty meetings
4) Research Scholar meetings
5) Project evaluations
6) Poster presentations
7) Selection examinations

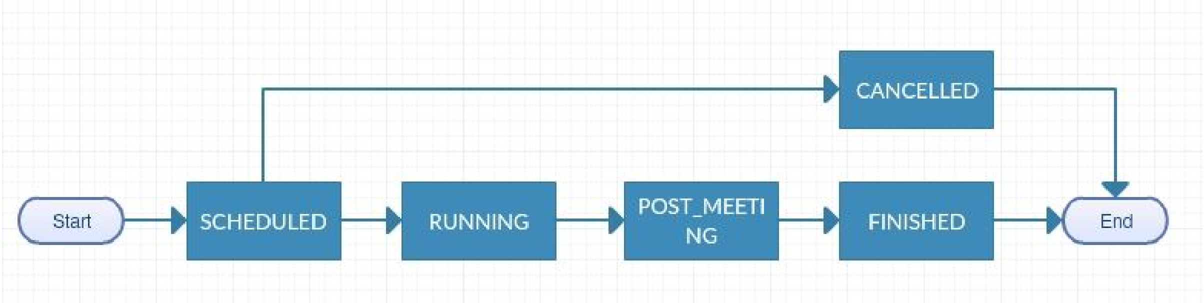
These meetings require different kinds of resources - stationery, snacks, student files, evaluations, classrooms with specific infrastructure etc. This software project shall enable the office staff to schedule meetings, track the resources required and book rooms for a single meeting or for courses.

IV. System Requirements

This section is divided into *functional* and *nonfunctional requirements*. The software is centered around the scheduling of meetings. Each meeting entity has a definite life cycle and may be in any of the following states:

1. **SCHEDULED**: State of meeting just after scheduling.
2. **POST_MEETING**: State of meeting after completion of actual meeting but pending completion of formalities (gathering signature forms, reports etc.).
3. **FINISHED**: Post completion state of entire meeting lifecycle.
4. **CANCELLED**: State of meeting if cancelled.

The life cycle may be summarized as:



A. Functional Requirements:

The software product should:

1. Allow only authorized office staff to access the software.
    a. The software shall allow an admin user to add more users to the platform.
    b. The software shall allow an admin user to remove existing users from the platform.
2. Allow storage of details about:
    a. Infrastructure (Venue, seating capacities, projector availability, air-conditioning etc.)
    b. Participants
3. Allow scheduling of meetings.
    a. In cases where a valid schedule cannot be generated, the user shall be notified with the appropriate reason.
    b. This includes all the constraints: time, meeting room, participants' availability etc.
    c. It shall also provide suggestions regarding which rooms are free and support all the organizer’s requirements.
    d. Suggestions shall try to recommend rooms not in use by prior meetings to allow scope for meeting extensions.
4. Allow meetings requirements to be entered.
5. Auto-add reminders for requirements satisfaction and validation/checking/testing prior to meeting (staff , technician etc.).
6. Send reminders to meeting participants.
7. Send a specialized reminder to the meeting organizer with options for requesting additional resources.

    a. Organizer responses shall be collected by the software and shown to the staff in a dedicated section for approval.

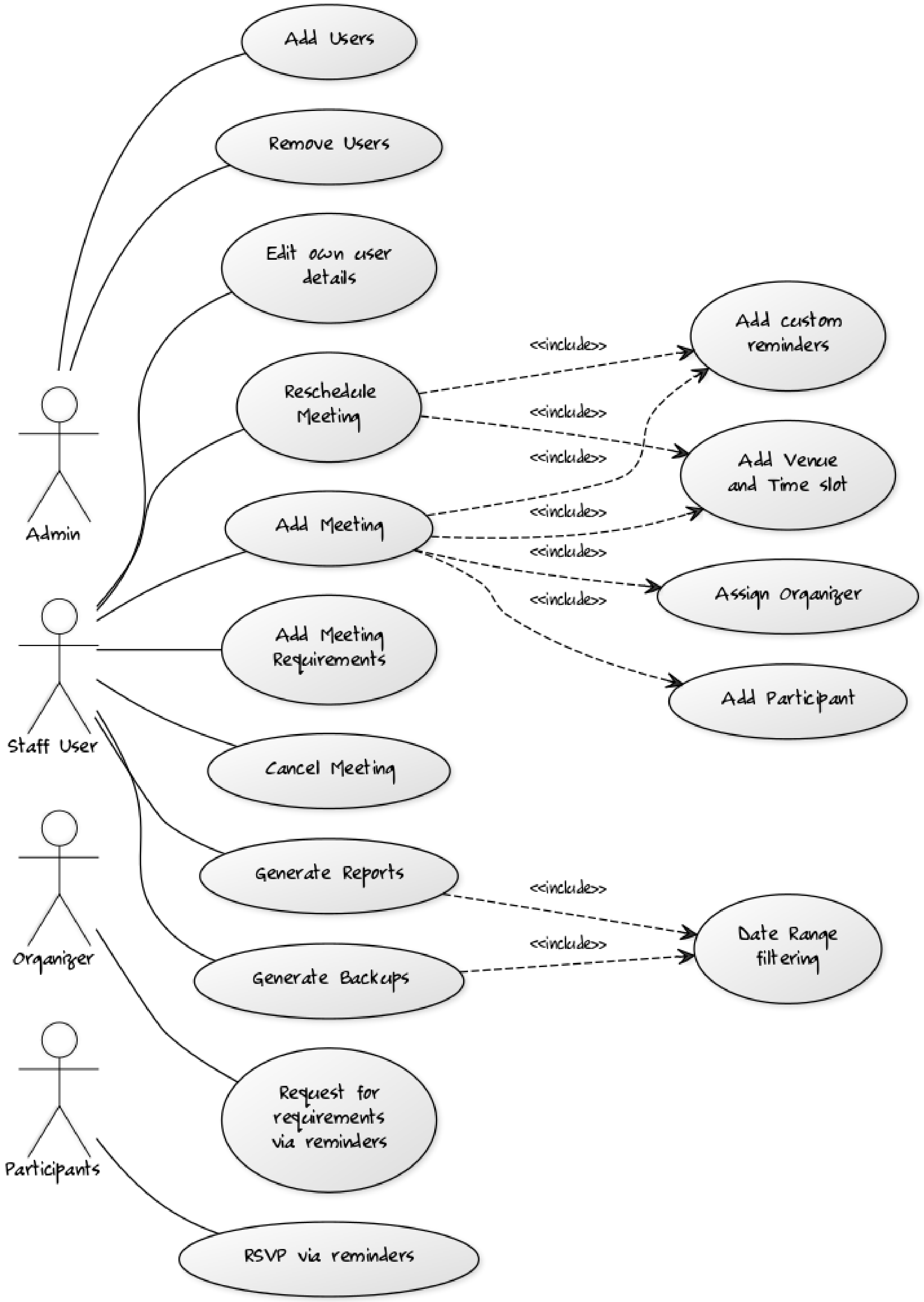
8. Remind the staff for collecting reports, signature forms etc. (see status POST_MEETING)
9. Allow report generation. (see status FINISHED)
10. Manage cancellation workflow for meetings. (see status CANCELLED)
11. Allow backup generation.
12. Allow meeting participants to respond with whether or not they will attend the meeting.
    a. This data shall then be provided to the staff in terms of headcounts.
13. Allow rescheduling of existing meetings.
14. Allow calendar view of all scheduled meetings.
    a. This shall include meetings not scheduled via this platform but available on the CSE department calendar data.
15. Provide cumulative meeting reports.
    a. Should take in date ranges and build a report for meetings taking place in the range.
    b. Should provide info regarding participants, organizers, descriptions etc.

B. Non-Functional Requirements

1. The software should not use any proprietary code or platform.
2. It should be platform independent and run on platform-agnostic web interfaces.
3. It should ensure reliability and security.
4. The software should be well-documented.

V. Use Cases

The use cases may be summarized by the following UML diagram:



VI. Scenarios

This section is divided into 4 parts based on the category of the user:

A. Admin

Scenario #1: Add Users	
Initial Assumption	Admins knows new user details and there is space for adding more users in the system.
Normal	Admin enters username and password for the new user and makes the request to the server.
What can go wrong	1. If username exists, the server will send an error message. The admin then has to remake the request with new credentials. 2. Network error.
System state on completion	New user record is stored in the database. New user can now login to the system.

Scenario #2: Remove Users	
Initial Assumption	Admin knows the user credentials to be removed.
Normal	Admin enters username and confirms the removal.
What can go wrong	1. User doesn't exist. Server informs the admin. 2. Network error.
System state on completion	User record is removed from the database. Removed user cannot login to the system hereafter.

B. Staff User

Scenario #1: Add meeting	
Initial Assumption	Staff user knows the meeting details (Obtained from the organizer).
Normal	User enters the meeting title, scheduled time and venue, participants and other details in the interface and submits it to the server.
What can go wrong	1. Another meeting may already have been scheduled at the requested venue at an overlapping time. The server responds with an error message and alternate suggestions. 2. Some of the participants are already scheduled to attend other meetings at the same time. The server responds with an error message and alternate suggestions.
System state on completion	A new meeting record is added to the database. Reminders are added to queue.

Scenario #2: Add meeting requirements	
Initial Assumption	Staff user receives meeting requirements.
Normal	User approves/adds/modifies the meeting requirements in the interface and submits it to the server.
What can go wrong	1. Network error
System state on completion	Meeting requirements are added to the corresponding meeting record.

Scenario #3: Reschedule meeting	
Initial Assumption	Staff user gets request for rescheduling from the organizer. Relevant meeting was already scheduled on the platform.
Normal	User enters the rescheduled time and venue of the meeting and submits it to the server.
What can go wrong	1. Some of the participants are already scheduled to attend other meetings. The server responds with the error message and alternate suggestions. 2. Another meeting has already been scheduled at the requested venue at an overlapping time slot. The server responds with the error message and alternate suggestions.
System state on completion	The meeting record is updated in the database. New reminders are added to queue.

Scenario #4: Cancel meeting	
Initial Assumption	Staff user is requested to cancel the meeting by the organizer. Relevant meeting was already scheduled on the platform.
Normal	User selects the meeting for deletion in the interface and submits it to the server.
What can go wrong	1. Relevant meeting record does not exist. User will not be able to find it on the interface. 2. Network error
System state on completion	The meeting record is modified in the database and shifted to the **CANCELLED** state. New cancellation reminders are added to the queue.

Scenario #5: Generate backups	
Initial Assumption	The data is already in the system.
Normal	User selects the generate backup option and selects the appropriate date range. The server then delivers the backup in a consolidated file.
What can go wrong	1. The system may not have any records to backup. Server responds with an appropriate message. 2. The date range may be invalid. The client interface raises an error asks the user to correct the range. 3. Network Error.
System state on completion	The client system obtains a consolidated file containing backup records for the selected date range.

Scenario #6: Generate Reports	
Initial Assumption	The data is already in the system.
Normal	The staff selects the generate report option and selects the appropriate date range. The server responds with relevant information in an easy-to-interpret format.
What can go wrong	1. The system may not have any records for the given date range. Server responds with an appropriate message. 2. The date range may be invalid. The client interface raises an error asks the user to correct the range. 3. Network Error.
System state on completion	The client system displays the generated report.

Scenario #7: Edit own user details	
Initial Assumption	Staff user exists in the system (staff is logged in).
Normal	User enters his new password and reenters it for confirmation. He then submits the request.
What can go wrong	1. If the new password is identical to the old one, the server responds with a error message. The user can then change his new password. 2. The confirmation password may not be identical to the new password field. The client interface raises an error and asks the user to correct. 3. Network Error
System state on completion	The user record is modified in the database. The next time the user logins, the old password is no longer valid. The new password must be used.

C. Participants

Scenario #1: RSVP via Reminders	
Initial Assumption	Participant has received a reminder mail from the system. Participant has also decided whether or not to attend. By default, the server maintains status of not coming.
Normal	Participant chooses the appropriate option: will attend or will not attend. The server responds with an acknowledgement message. If this is not the first response from the user, it acknowledges a modification to the earlier response.
What can go wrong	1. The participant may not be invited for the relevant meeting. Server responds with an appropriate error message. 2. Network Error.
System state on completion	Participant arrival status is appropriately set. Head counts of arriving participants is updated.

D. Organizer

Scenario #1: Request Requirements via Notification	
Initial Assumption	Organizer has received a notification with a form for requirements. By default, ‘No Requirements’ is considered.
Normal	User fills the form and submits it.
What can go wrong	1. Organizer doesn’t respond, server sets default requirements. 2. Network error.
System state on completion	Requirements field for the relevant meeting record is updated and can been seen by the staff user.

VII. Test Cases

To be added in the next report.

VIII. References

1. UML Diagrams, https://yuml.me/diagram/scruffy/class/draw
2. “Software Requirements Specification of A University Class Scheduler” *Deanna M. Needell, Jeff A. Stuart, Tamara C. Thiel, Sergiu M. Dascalu, Frederick C. Harris, Jr*.
3. “Software Engineering” *Somerville*.