RealOffice Requirements Document

CS14B023, Rahul Kejriwal
CS14B007, Suhas
CS14B045, MALIREDDI SUNIL KUMAR
CS09B043, Shanker Lal Sharma
CS09B031, Chandrakanth

Note: While all effort has been taken to be as accurate as possible, there may be few errors in the interpretation as we have not yet sat down with the client for requirement elicitation. The following document builds on whatever information was provided and our interpretations.

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.

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.

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-meetings 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 input 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 (periodic and nonperiodic) 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 software engineering process adopted will be an agile approach with incremental development. The database will be modelled using ER diagrams and will be translated to a relational database framework. UML will be used to model use cases and scenarios and develop the system architecture. The software will run on a client-server model with the backend running on Django framework.

The remainder of the document will focus on
1) General description (User Requirements)
2) System requirements
3) Use Cases and Scenarios
4) Software Architecture

III. User Requirements

The main purpose of the software is to automate CSE department office processes which includes scheduling meetings and courses. These are the two broad categorical entities to be scheduled mainly differing in the fact that courses are recurrent (periodic) whereas meetings are nonperiodic.

The CSE Office deals with various participating entities like students, faculty members, and staff members. The main kinds of meetings organised are:
1) Lectures (regular and invited)
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 Specification

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

> SCHEDULING, SCHEDULED, PREPPING, PREPPED, RUNNING, POST_MEETING, FINISHED, CANCELLING, CANCELLED, RECURRING

(For more information regarding this, see architecture section.)

## A. Functional Requirements:

The software product should:

1. Allow storage of details about:
    a. Infrastructure (Venue, seating capacities, projector availability etc.)
    b. Users (Type of user - student, faculty or staff and privilege levels)
2. Allow scheduling of periodic courses or non-periodic meetings.
3. Allow meetings requirements to be entered. (see status SCHEDULING)
4. Auto-add reminders for requirements satisfaction and validation/checking/testing prior to meeting. (see status PREPPING)
5. Send reminders to meeting participants. (see status PREPPED)
6. Auto-demand report submission, signature forms etc. (see status POST_MEETING)
7. Allow report generation. (see status FINISHED)
8. Manage cancellation workflow for meetings. (see status CANCELLING)
9. Allow backup generation.
10. Allow courses to be scheduled. (see status RECURRING)

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.

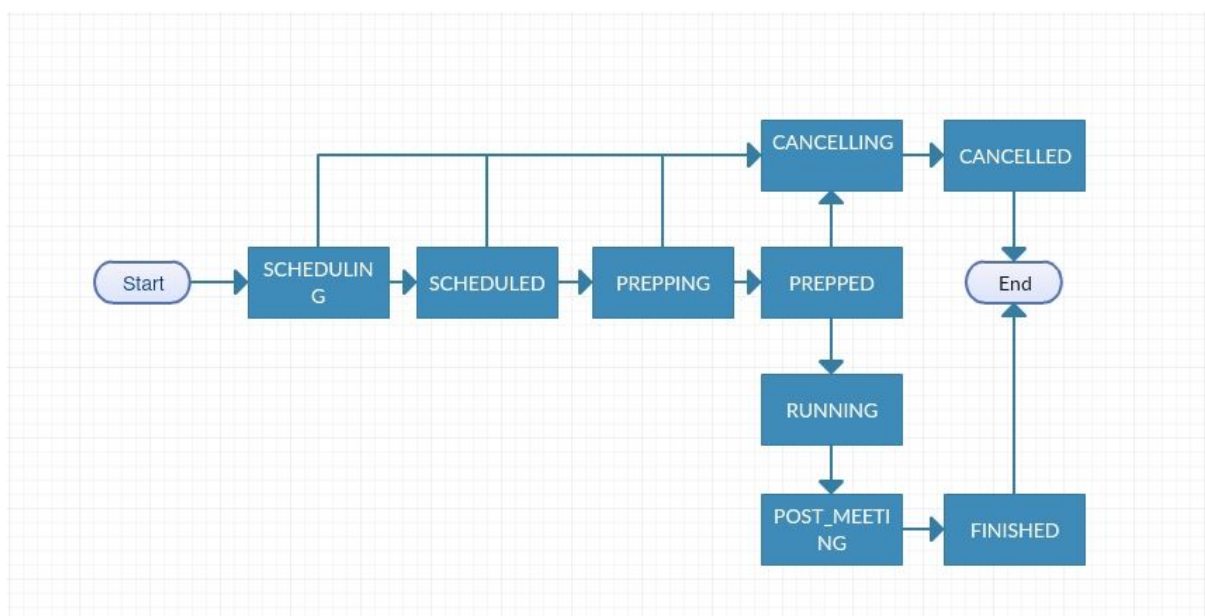# VI. Software architecture

A. ER Diagram
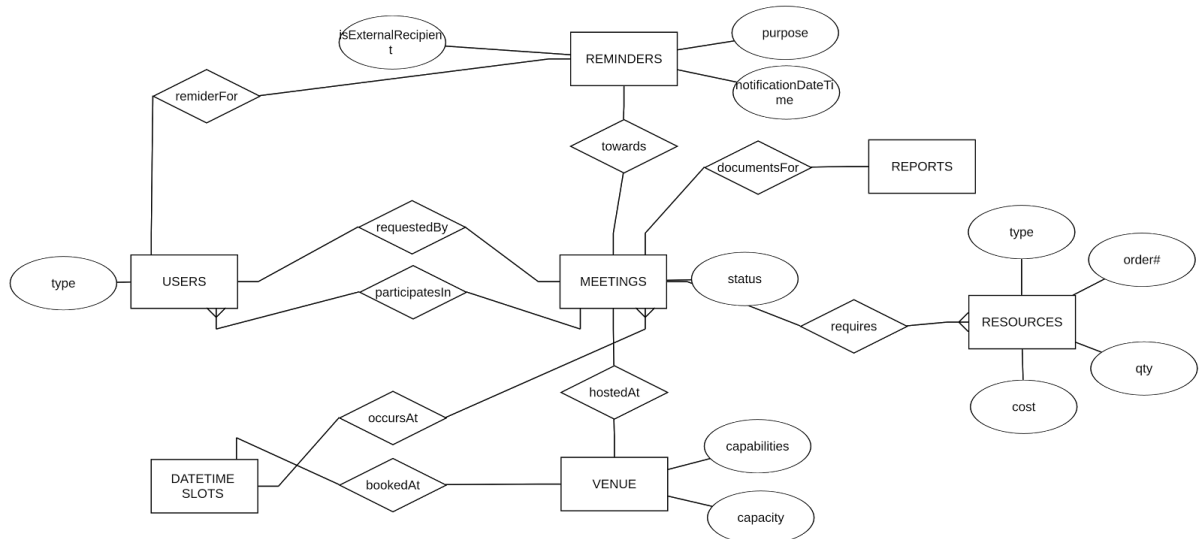
The software is centered around the MEETING entity with relationships to USERS, REMINDERS, REPORTS, VENUE, RESOURCES and DATETIME SLOTS.

Further elaboration of the diagram will be done post client meeting.

B. State Transition Diagram for Meeting Status

This diagram shows the general lifecycle of a meeting. The states include:

1. SCHEDULING: State of meeting immediately after the request is made.
2. SCHEDULED: State of meeting after approval but before meeting requirements are considered.
3. PREPPING: State of meeting after requirements gathering begins.
4. PREPPED: State of meeting after conclusion of requirements gathering.
5. RUNNING: State of meeting when it's on-going.
6. POST_MEETING: State of meeting after completion of actual meeting but pending completion of formalities (gathering signature forms, reports etc.).
7. FINISHED: Post completion state of entire meeting lifecycle.
8. CANCELLING: State of meeting immediately after cancellation request but pending cancellations of allocated resources.
9. CANCELLED: State of meeting post allotted resource cancellation.