Software Requirements Specification for the `ConMan`

Contributors:

Angel De Castro Luis Retana
Nicholas Otto Christopher Yip

Version 0.1
10/18/2013

Contents

1 Introduction **2**

1.1 Purpose . . . 2

1.2 Intended Audience . . . 2

1.3 Product Scope . . . 2

2 Overall Description **3**

2.1 Product Perspective . . . 3

2.2 User Classes and Characteristics . . . 3

2.3 Product Functions . . . 3

2.4 Operating Environment . . . 4

2.5 User Documentation . . . 4

3 External Interface Requirements **5**

3.1 User Interfaces . . . 5

3.2 Hardware Interfaces . . . 6

3.3 Software Interfaces . . . 6

3.4 Communications Interfaces . . . 6

4 System Features **7**

4.1 Calendar View . . . 7

4.1.1 Description and Priority . . . 7

4.1.2 Stimulus/Response Sequences . . . 7

4.1.3 Functional Requirements . . . 7

4.2 Adding, Editing, Deleting Tasks . . . 7

4.2.1 Description and Priority . . . 7

4.2.2 Stimulus/Response Sequences . . . 8

4.2.3 Functional Requirements . . . 8

4.3 Adding, Editing, Deleting Checklists . . . 8

4.3.1 Description and Priority 8
4.3.2 Stimulus/Response Sequences 8
4.3.3 Functional Requirements 8
4.4 User Profiles 9
4.4.1 Description and Priority 9
4.4.2 Stimulus/Response Sequences 9
4.4.3 Functional Requirements 9
4.5 Notification System 9
4.5.1 Description and Priority 9
4.5.2 Stimulus/Response Sequences 9
4.5.3 Functional Requirements 9

5 Other Nonfunctional Requirements **10**

5.1 Performance Requirements 10
5.2 Safety Requirements 10
5.3 Security Requirements 10
5.4 Software Quality Attributes 10
5.5 Business Rules 11

1 Introduction

1.1 Purpose

The Calendar, Organization, and Notification Manager - `ConMan` - is a web-suite that gives project managers the tools needed to plan, assign, and follow-up on tasking given to project team members. This SRS document covers all aspects of `ConMan`.

1.2 Intended Audience

This document is intended primarily for developers and reviewers of the `ConMan` suite. Advanced users of `ConMan` may find reading this SRS document interesting reading as well.

1.3 Product Scope

`ConMan`'s main objective is to give project managers and project team members a simple, but effective, tool for planning their tasking and keeping to a schedule. `ConMan` accomplishes its purpose with these mechanisms:

- Tight integration of scheduled tasking with a calendar view. `ConMan` will allow a user to see at a glance the tasks they are assigned, how much progress has already been made, and the effort remaining to complete the task.
- Meaningful notifications and updates. Users can choose to select a notification schedule themselves or recieve automatic notifications of changes to their tasks and upcoming deadlines.

2 Overall Description

2.1 Product Perspective

`ConMan` is inspired by other attempts to create a useful calendar, checklist, and task management software suite such as Trello, SourceForge, and some basic calendar applications for iPhone and Android. `ConMan's` approach to creating a task-management suite is emphasizing simplicity and ease of use as the primary development and design philosophies.

2.2 User Classes and Characteristics

The two user classes for `ConMan` are managers and members. Managers have privileged commands available which effect an entire team. Members are assigned to teams and can modify the tasks that are assigned to them. When a member switches to the individual context, each member becomes his/her own manager.

2.3 Product Functions

`ConMan's` functions can be divided into two categories: application functions and user functions. Application functions are functions that describe the basic functionality of `ConMan`. They include:

1. Creating a user profile.
2. Switching to calendar view.
3. Switching to task view.
4. Switching to checklist view.
5. Receiving notifications and updates.

User functions are further divided into manager and member functions. Manager functions are only available to users designated as project managers. Users are designated as project managers during the creation of their team. Manager functions include:

1. Creating and deleting teams.
2. Adding and modifying tasks for a team.
3. Creating, modifying, and deleting checklists for the team.
4. Creating group notifications.
5. Performing member verification functions.

Finally, the member functions apply to every user of `ConMan`. These functions include:

1. Writing task progress notes.

2. Writing checklist progress notes.
3. Switching between team and individual views.
4. Marking checklist items as complete (if the checklist is assigned to the member).

2.4 Operating Environment

`ConMan` deploys on an ASP.NET server using a SQL Server Express database. The target browsers in which `ConMan` will run are Firefox and Chrome. The application's client-side code (JavaScript, jQuery, etc.) will run within these browsers.

2.5 User Documentation

The documents available describing `ConMan` include this software requirements specification, a user manual, design document, testing report document and an in-app help menu.

All of the documentation will be available in PDF.

3 External Interface Requirements

3.1 User Interfaces

User Interface characteristics for the `ConMan` web application are as follows:

- The application will use standard HTML, CSS, JavaScript, and possibly ASP.NET controls to structure and create the user interface.
- The web application will provide a consistent layout through the use of consistent coloring, formatting, etc.
- The web application will provide the user with a "Help" button so that the user can quickly and easily learn how to use each component of the application.
- The web user interface will handle errors in different ways:
  - If a web form is not completed in a valid manner, then the UI will present an error message in red text to the right of each form field that was not valid. Each of these error messages will display a message that is appropriate for the error. For example, if a field accepts non-negative numbers and the user enters -1.5, the UI will display an error message indicating that input must not be negative.
  - All other errors will be presented to the user via a dialog/message box that cannot be ignored without acknowledgment (i.e. pressing "OK"). This message must present the user with helpful messages that describe the error. For example, an error message for a database transaction may state "ERROR: Communication with the server's database has been lost".
- The web application will allow the user to quickly switch between calendar, checklist, and calendar/checklist views.
  - Calendar View
    - The calendar view shows the current month's calendar and for each day, it will display the name of each task that has a checklist due on that day. By clicking on the name of a task, the web application will redirect the user to the task view (discussed below under the "Task View" bullet).
  - Task View
    - The task view will show all information about the selected task. The task name, description, checklist(s), and notes will be available to the user. The user will also be able to edit (add a checklist or modify other task information) and delete a task. In addition, this view will enable the user to sign up for notifications concerning the current task. Clicking on a checklist from this view will redirect the user to the checklist view (discussed below under the "Checklist View" bullet).
  - Checklist View
    - The checklist view will show all information about the selected checklist. The checklist name, description, parent task, item(s), and notes will be available to the user. The user will also be able to edit (add an item to the checklist or modify other checklist information) and delete a checklist. In addition, this view will enable the user to sign up for notifications concerning the current checklist. Clicking on a checklist item will display the details associated with that checklist item.

- The web application will have log-in information available in the top-right corner of the page.
  - If the user is not logged in, the top-right corner of the page will prompt the user for log-in credentials
  - If the user is logged in, the top-right corner of the page will display the users log-in name and will display a link/button for the user to sign out
- The web application will alert the user that one or more checklists are due for the current day by presenting the user with dialog/message box notifications stating which tasks have checklists due. Notifications will also be presented for checklists with due dates in the near future (e.g., due dates within the next 3 days).
- The web application will also alert the user via notifications if any tasks or checklists assigned to them have been modified.

3.2 Hardware Interfaces

Not applicable for `ConMan`.

3.3 Software Interfaces

As mentioned in section 3.1, the web application will be built using HTML, CSS, JavaScript, and ASP.NET. The application will also need to interface with the database located on the server through the use of SQL and database libraries such as those provided by the .NET framework. This database will store all data pertaining to user information (e.g., ID, name, log-in credentials, age, employer, team name, email address, etc.), company information (e.g., ID, name, employees, teams, etc.), task information (e.g., ID, due date, creator team ID, employees, checklists, etc.), and checklist information (e.g., ID, creator ID, due date, task ID, employees IDs, list, etc.).

3.4 Communications Interfaces

The web application will rely on the following technologies for communication:

- SMTP for sending out email notifications to users concerning checklists with upcoming due dates
- SQL for performing transactions with the application's database

4 System Features

The system will have a simple layout. The main screen will show a simple month-view calendar with the current day highlighted. The name of an assigned task or checklist will appear on the day of the month when it is due. The calendar then offers the possibility of clicking on any day or task. If a task is clicked, a list will appear with the task details along with options for adding, deleting or editing checklists associated with the task. If the day of the month was clicked, the only option will be to add a new task. The interface will also have a menu to select user context, change view and any other secondary features helpful to the system (e.g. help-section links).

4.1 Calendar View

4.1.1 Description and Priority

Creating a calendar view will be the main priority. It should allow the user to click on any day to see or create new tasks. The system should also implement the possibility of changing months to see past and future tasks. This will be of the highest priority, 9, since all future features are based on the user being able to choose a day from the calendar.

4.1.2 Stimulus/Response Sequences

After logging into `ConMan` the initial screen will be the calendar view. When the user clicks on a day it should transition into a to-do list interface. Here the user can see, add, edit or delete tasks. The view feature will show a form with the tasks information.

4.1.3 Functional Requirements

The interface shall be intuitive, with minimal training required to learn how it is used. To minimize user error, all action buttons will be large and simple to understand.

4.2 Adding, Editing, Deleting Tasks

4.2.1 Description and Priority

Modifying a task is the primary user action in `ConMan`. Adding a task will create a new entry for a specific day inside the calendar view. Deleting a task completely removes a task, while editing a task will allow the user to change any of a task fields. This will be of the highest priority, 9, since all of the system functionality depends on the user being able to assign tasks to a day.

4.2.2 Stimulus/Response Sequences

Once the day from the calendar view is selected, the user will be able to choose between adding, deleting, editing and viewing the task. Adding a task will take the user to a form where he can enter the initial task values and then create the task. Editing will take the user to the same form with the fields already completed and deleting will remove the task entry.

4.2.3 Functional Requirements

The task form will require the usual fields in a calendar. It should include date, time, description, and alarm. Once filled the system adds the task to the day specified. The form will check for errors in the information supplied before accepting the task. If the user modifies the deadline date, the change should also be reflected on the calendar and a notification should be sent to all employees associated with the task.

4.3 Adding, Editing, Deleting Checklists

4.3.1 Description and Priority

Adding a checklist to a task will update the task by placing the new checklist in the task's checklist list. Deleting a checklist completely removes a checklist, while editing a checklist will allow the user to change any of a checklist fields. This will be of priority, 7, since the usefulness of tasks are greatly diminished if checklists cannot be associated with the tasks.

4.3.2 Stimulus/Response Sequences

Once a checklist is selected, the user will be able to choose between adding, deleting, editing and viewing the checklist. Adding a checklist will take the user to a form were he can enter the initial checklist values and then create the checklist. Editing will take you to the same form as adding a checklist, but with the fields completed with existing information and deleting will remove the checklist and all references to it.

4.3.3 Functional Requirements

The checklist form will require the usual fields in a calendar. It should include date, time, description, and alarm. Once the form is filled, the system then adds the checklist to the specified task. The form will check for errors in the information supplied before accepting the checklist. If the user changes the date this should also be reflected on the calendar and a notification should be sent out to all employees associated with the task.

4.4 User Profiles

4.4.1 Description and Priority

This feature allows a user to log into a profile, loading all the tasks previously created onto the calendar view. A simple authentication process is required to ensure privacy; a user will also be able to create an account on startup. Since this feature is important but not critical to system use, User Profiles have a priority of 7.

4.4.2 Stimulus/Response Sequences

Users will see an authentication screen each time the log into `ConMan`. The authentication screen will offer the possibility of creating a new user or entering existing credentials. If a new user is created, the system will transition into the user creation interface where she can complete the fields. If the user already has an account, she will be showed her calendar with all the previously created tasks.

4.4.3 Functional Requirements

The new user form will require the usual fields: name, surname, password, email, etc. After the fields are supplied, the system will take the user to the calendar. The authentication process must be secure. Therefore, it must control who accesses each profile since data is confidential. If incorrect user credentials are entered or if the information for the new user is incorrect, the user will be prompted with a error message.

4.5 Notification System

4.5.1 Description and Priority

This feature can be activated at the task creation form. If activated, this feature will notify the user with an email and with a message/dialog box (via the web application) when a task has an upcoming due date. The notification alerts will have priority 5.

4.5.2 Stimulus/Response Sequences

This option will be made available to the user once she has entered the task creation form. It will give the user the option to choose a time to remind her and possibly other users of the event.

4.5.3 Functional Requirements

The system will have to validate the time entered and ensure that it is plausible and formatted correctly. If an invalid notification time is set, the user will be informed of the error.

5 Other Nonfunctional Requirements

5.1 Performance Requirements

As a general requirement, the user should be able to visually see some type of response from the program within 0.2 seconds after a valid user action. If, for instance, a user performs an action that requires validation from the server (such as authenticating the user upon login), this requirement does not stipulate that the server will finish processing the request within 0.2 seconds, but rather the program will give at least a loading indicator to signal the user that the action was been received.

All e-mails sent by `ConMan` will be sent within 5 minutes of the indicated alarm time. `ConMan`, of course, has no control over the performance of the user's email network which will receive the mail.

5.2 Safety Requirements

There are no known physical safety requirements at this time. Caution should be exercised when using any unfamiliar device, however.

5.3 Security Requirements

A user should only be able to log onto `ConMan` with valid user credentials. Furthermore, a user should only be able to modify tasks or checklists which they are assigned to. Also, member level users can only view tasks assigned to the team to which they belong.

Manager level users, however, are the only user class that can add or assign tasks to teams. Also, Manager users are solely responsible for marking tasks finished or checklists as complete. Basic users will be allowed to perform these functions.

5.4 Software Quality Attributes

The source code will adhere to the following attributes:

- The source code will be reasonably documented. All non-trivial functions or methods will have header documentation describing the following:
  1. Name
  2. General description, including pre-conditions
  3. List and description of input parameters
  4. List and description of returned values
  5. Information about state change during method execution (post-condition)
  6. Functions which are entirely described by the function name or private functions specific to a small code block need not satisfy all of these requirements.

- The source code will adhere to a single programming style, namely the Stroustrup variant of K&R style.
- All code blocks will be enclosed in curly brackets, even if not explicitly required.

Above all else, the system should be at least minimally usable upon delivery. If time allows, features which increase user flexibility will be added, but attempts to increase user options must not delay the project delivery.

5.5 Business Rules

This product will ship by the end of the fall semester.