

SE 441

Software Requirements Specification

Course Scheduling Application - “Milestone”



California State University

SAN MARCOS

Jordan Mower

Jordan Nienaber

Jonathan Hua

Nick Luckey

1. Introduction

1.1 Purpose:

Contained within this document is the description of the various elements that compose the software system to be produced. This document will also serve as a guideline for the end product and its functionality. This specification is intended for business personnel involved with and contributing to the production of the software system described below, as well as confirmation of functionalities with the customer.

1.2 Scope: This subsection should

The production of Milestone will provide students with the ability to manage, view, and share tasks assigned to them by course instructors. It will also provide them with the ability to track the completion of tasks that have been entered and the overall importance of the task in relation to the course. The production of task information *will not* be automatically generated and must be entered manually by students. The application of this software will provide students with a community based solution for keeping track of assigned tasks within the classroom. This is beneficial in the following two ways: First that only one student has to create a set of tasks and then all of his or her associations can use those to help organize the course. Second, that it allows for increased awareness of the tasks that will be due.

1.3 Definitions, Acronyms, and Abbreviations

User: A student enrolled in a course and using the application to organize tasks.

Task: An assigned homework, test, or another element of work assigned by a course instructor.

Course: A container for tasks to be completed by a student of an instructor.

1.4 References

Compton, Jerry. "Personal Planner" PowerPoint and WordDocument.

IEEE Computer Society. "IEEE Recommended Practice for Software Requirements Specification". 20th October, 1998.

1.5 Overview

Described in detail below, you will find the necessary information regarding the implementation and general requirements for producing the application proposed by Jerry Compton.

Section two will give an overall description which will discuss how the product relates to other products on the market, the major functions of the product, a description of the intended users of the product, limitations on development, assumptions about the audience, any dependencies the product may have, and features that we could see being implemented in future versions of the product.

In section three we will discuss specific requirements which include external interfaces, functional requirements, performance requirements, design constraints, software system attributes, and other requirements. Section 3.1 will discuss in detail all inputs and outputs to the system and the various interfaces (user, hardware, software, and communication). These will include screen formats, page layouts, page content, and optimizations for users of the system, supported devices and how we plan to support them, software packages integrated into the development process, and network protocols and dissemination of data over them. Section 3.2 will address system input processing for generating outputs and describes user classes. Section 3.3 will address verifiable numerical requirements of the system. In section 3.4 any limitations to the product based on

hardware or other limiting factors will be discussed. Section 3.5 will attend to portability, maintainability, reliability, security, and availability. Lastly, section 3.6 will touch on any other requirements that may be deemed necessary.

2. Overall Description

This section will explain the basic functionality of the system while also describing the stakeholders and their functions in the system. It will also cover any constraints, assumptions and dependencies.

2.1 Product Perspective

The “Milestone” system will be a self contained mobile application. It should not need to rely on any other applications running on the mobile device like GPS, contact list, or a web portal for administration. A database will be used to store user information and course information which will also allow for easy sharing between associated users.

2.2 Product Functions

The “APP NAME” system will allow users to generate courses to monitor. Courses will be composed of a course name, meeting days, an instructor name, and a unique color that will be used when presenting the course in the planner view. Within each course a user will create tasks that represent the various assignments that will be due. Tasks will have a title, a due date, a priority level, a percentage value, and an option for additional comments. The home page of an enrolled user will display the date, time and a 6 month calendar view with color coded tasks populating the various dates.

2.3 User Characteristics

There will be only one type of user that interacts with the system which will be the users of the mobile application. Mobile application users will be the ones who create

courses under their account and add tasks to each course. Intended users will be those who are enrolled in any course or set of courses. The average user is likely to be a college student enrolled in an undergraduate program. No technical expertise other than the ability to operate a smartphone will be required.

2.4 Constraints

The most obvious constraint is a stable internet connection. When sharing task/course details amongst associated users the receiving user will be required to connect to the database to fetch course details. A nonfunctioning internet connection will completely limit the ability to share.

Database availability, capacity, and speed could constrain the systems ability to provide adequate service to users.

2.5 Assumptions and Dependencies

As we are creating this software for Android devices only, we are assuming that the majority of users will have the OS that we are programming for installed. However, this is a faulty assumption as according to a table from [an android developer site](#) , there is no OS that the majority of users have installed, requiring us to assume that the various users of this app will have a different version of the Android OS installed.

2.6 Requirements Subsets

In future versions of the application we would like to:

- Implement friends and chat functionality.
- the ability to suggest tasks to a course creator.
- Copying and Storing of courses users have subscribed to.
- Add a seperate user class "Teachers" that allows registered teachers the ability to generate their own courses that students can subscribe to.

-
- A rating system for course creators that allows users to quickly judge whether or not that course they are viewing is from a reputable source.

3. Specific Requirements

This section contains all the functional and qualitative elements of the system. It gives a detailed description of the system and all its features.

3.1 External Interface Requirements

This subsection will give a detailed description of all inputs/outputs from the software system by describing user interfaces, hardware interfaces, software interfaces, and communication interfaces.

3.1.1 User Interfaces

All windows should be configured for the screen size in which the device is operating on. This means that a device screen width will have to be checked upon application start to accommodate various screen sizes. A landscape view will *not* be supported.

A first time user of the application should see a log-in page with the option to register a new account. The log-in page will simply ask for a username and password. The

registration page will ask for a username, password, password confirmation, date of birth, school name (if any) and current year/grade in school if applicable.

<div><div>AppLogo</div><div>Username</div><div></div><div>Password</div><div></div><div>Login</div><div>Not Registered? Click Here</div></div>	<div>Username</div> <div></div> <div>Password</div> <div></div> <div>Confirm Password</div> <div></div> <div>Birthday</div> <div></div> <div>School</div> <div></div> <div>Grade/Year</div> <div></div> <div>Login</div>
--	--

Upon a successful login the user will be taken to the “Home” page. Located at the top will be the current date and time. Below the date/time will be buttons to add a course, add a task, and a search tab. This screen will show a vertically scrollable calendar view of 6 months from the current month. The users created tasks will show up on the due date in the identifying color selected for that course. Each task will also show a priority level if one was selected.

Selecting a day on the calendar will show a view of that days tasks with the appropriate course background color, task percentage if applicable, task title, the course it belongs to, and the priority level. It will also have buttons to delete, edit, and mark as complete.

Search will allow users to search for courses to subscribe to. Sharing courses will be achieved by subscriptions. Subscribing to a course will synchronize your calendar with that of the user who created the course. Any additional tasks added by creator after

subscription will be automatically reflected in the subscribed users course. All created courses will be searchable. Editing a subscribed course will have no effect.

11:57 PM
Sunday, September 22nd, 2019

search

Course

Task

September

			1			
			2			

October

November

--	--	--	--	--	--	--

Tuesday, 9/24/19

SE441, Assignment #1

25% - Priority Level 1

✓

E

✗

BIO302, Lab 4

5% - Priority Level 3

✓

E

✗

Introduction to Computer Networks, Wireshark Lab

10% - No Priority

✓

E

✗

Search Courses...

SE 441 - T/TH

Instructor: Yongjie Zheng

14 Tasks

Created By: Yongjie Zheng

Subscribe

Cybersecurity - 497-6 - M/W

Instructor: Teresa Macklin

12 Tasks

Created By: Teresa Macklin

Subscribe

Pressing the "Course" button will bring up the course menu screen. Users will have the option to add, edit, and delete a course here by selecting a course and pressing the appropriate button. The add course screen will require users to enter a course name, meeting days, an instructor name, and select a color for the course. Edit button will populate the add course screen and allow changes. Delete course will delete the selected course.

Add Course

Course Name:

Instructor Name:

Select Meeting Days

S	M	T	W	Th	F	S
---	---	---	---	----	---	---

Select Course Color:

Add

Course Menu

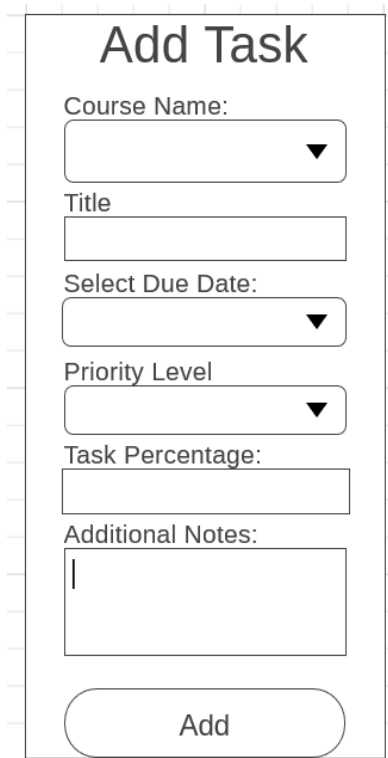
Select Course ▼

Add Course

Edit Course

Delete Course

Clicking the Task button will bring up the Add Task screen. The add task screen will simply require users to select from a list of currently available courses. Then they will enter a title, select a due date in the future, a priority level (default no priority), a course percentage for the assignment, and any additional notes they wish to enter. Using the edit function from the calendar view will move users to a populated version of the add task screen and allow them to change details.



The image shows a mobile application form titled "Add Task". The form is enclosed in a rounded rectangle and contains the following elements from top to bottom: a label "Course Name:" followed by a dropdown menu; a label "Title" followed by a text input field; a label "Select Due Date:" followed by a dropdown menu; a label "Priority Level" followed by a dropdown menu; a label "Task Percentage:" followed by a text input field; a label "Additional Notes:" followed by a larger text input area; and a rounded rectangular button labeled "Add" at the bottom.

3.1.2 Hardware Interfaces

The mobile application does not have any direct hardware interfaces other than the mobile phone that the application runs on. Seeing as the application is being developed specifically for Android mobile devices it will only be supported on that platform.

3.1.3 Software Interfaces

The mobile application will run on Android devices so only the OS on the phone will be required. The application will also interface with a database to maintain courses, tasks, and friends.

3.1.4 Communication Interfaces

All communication protocols will be handled by the underlying logic of the OS.

3.2 Functional Requirements

Fundamental actions that must take place in the software in accepting and processing the inputs and in processing and generating outputs.

3.2.1 User Class 1 - Mobile Application User

3.2.1.1 Functional Requirement 1 - Registration

The system shall allow users to register using a username, password, school name, grade/year and date of birth. A username will be 8-20 alphanumeric characters. A password will be 8-20 alphanumeric characters. A school name will be at most 50 alphabetical characters. Grade/year will be a range from 1-12. Date of birth will be a range from January 1st 1920 to present. All users MUST be registered to use the application.

Use Case Name	3.2.1.1 Registration
Goal	Allow an unregistered user to create an account
Primary Actor	User
Preconditions	The user must have a smartphone and the application downloaded.
Postconditions	The user will be enrolled in the system with the username and information provided.

MSS	<ol style="list-style-type: none"> 1. User enters username, password, password confirmation, DoB, School, Grade/year 2. User presses submit/login 3. User is enrolled in database 4. User is logged in
Extension 1	<ol style="list-style-type: none"> 1a. User incorrectly matches password and password confirmation. <ol style="list-style-type: none"> 1a1. User is alerted to error with password matching and asked to reconfirm. 1b. User uses invalid characters in one of the provided fields. <ol style="list-style-type: none"> 1b1. The user is alerted to the invalid entry and asked to reenter.
Extension 3	<ol style="list-style-type: none"> 2a. User fails to fill in all fields <ol style="list-style-type: none"> 2a. User is alerted to missing fields and asked to fill them in. 2b. Username is taken <ol style="list-style-type: none"> 2b1. User is alerted that username is taken and asked to change to something else.

3.2.1.2 Functional Requirement 2 - Log In

The system shall allow users to log in using their registered credentials.

Use Case Name	3.2.1.2 Log In
Goal	Log a user into the application using their registered credentials.
Primary Actor	User
Preconditions	3.2.1.1
Postconditions	The user will be logged in to the application
MSS	<ol style="list-style-type: none"> 1. User enters username 2. User enters password 3. User presses Log In 4. User is logged in to the application
Extension 1	<ol style="list-style-type: none"> 3a. User enters an incorrect username or password <ol style="list-style-type: none"> 3a1. User is asked to reenter their username or password. 3a2. Start back at MSS 3.

3.2.1.3. Functional Requirement 3 - Add Course

The system shall allow users to add courses that include a course name, instructor name, meeting days, and course color. A course name will be 1-20 alphanumeric characters. An instructor name shall be 1-25 alphabetical characters. Meeting days will consist of only Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, and Saturday. Course colors shall be in the range of 10-30 different color variations. The addition of a course name that already exists within your planner will result in an error.

Use Case Name	3.2.1.3 Add Course
Goal	Allow user to add a new course to their planner.
Primary Actor	User
Preconditions	3.2.1.1, 3.2.1.2
Postconditions	The user has added the course to their planner.
MSS	<ol style="list-style-type: none">1. User enters course name, instructor name, meeting days, and selected a course color2. The add course button is pressed.3. The users course is added to their planner.
Extension 1	<p>1a. User uses invalid characters in the course name or instructors name</p> <p>1a1. The system alerts the user there is incorrect information</p> <p>1a2. The user fixes error</p>
Extension 2	<p>2a. The user already has a course with the same course name.</p> <p>2a1. The user is prompted to change the course name</p> <p>2a2. Begin at MSS 2.</p>

3.2.1.4 Functional Requirement 4 - Delete Course

The system shall allow users the ability to delete a course that is in their planner. Deletion of a course will result in unsubscribing to the selected course and removal of all data from their planner related to that course.

Use Case Name	3.2.1.4 Delete Course
Goal	User deletes selected course
Primary Actor	User
Preconditions	User has navigated to course menu. 3.2.1.3, Or user has subscribed to another users course
Postconditions	The selected course is deleted/unsubscribed from the users planner.
MSS	<ol style="list-style-type: none">1. User selects a course2. User presses delete course button3. System deletes course information from users planner and/or unsubscribes them from other users course.

3.2.1.5 Functional Requirement 5 - Edit Course

The system shall allow users to edit courses that have been added to their planner. This includes changing the instructor name, course name, meeting days, and course color. Editing shall adhere to validity checks mentioned in functional requirements 3.2.1.2. A user will only be able to edit a course that they have created.

Use Case Name	3.2.1.5 Edit Course
Goal	User edits course information
Primary Actor	User
Preconditions	3.2.1.3, and course being edited was created by the user.
Postconditions	Course information is successfully edited and saved.
MSS	<ol style="list-style-type: none">1. User selects a course from the course menu.2. User selects edit course from course menu.3. System populates the course fields

	4. User edits a field from the add course screen. 5. User presses add course 6. The edited course is saved and uploaded.
Extension 1	2a. The selected course from the course menu was not created by the user and can not be edited.

3.2.1.6 Functional Requirement 6 - Add Task

The system shall allow users to add tasks to the selected course. Task titles shall be in the range of 8-20 alphanumeric characters. Due dates shall be in the range of present day to any day in the future. A priority level shall be in the range of 0-3. A task percentage shall be in the range of 0-100%. Additional comments shall be alphanumeric characters in the range of 0-255. A user will only be able to add tasks to courses that they have created.

Use Case Name	3.2.1.6 Add Task
Goal	Add a task to the selected course
Primary Actor	User
Preconditions	3.2.1.3, user is the owner of selected course.
Postconditions	The task is added to the selected course and displayed in planner.
MSS	1. User navigates to Task Menu 2. User selects the course that they wish to add a task to. 3. User fills in task information. 4. Task is added to course and planner.
Extension 1	2a. User selects a course that they are not the creator of. 2a1. User is alerted that they cannot add tasks to courses they didn't create and are asked to select another course. 2a2. Begin at MSS 2
Extension 2	2a. User enters invalid characters into one of the fields.

3.2.1.7 Functional Requirement 7 - Edit Task

The system shall allow users to edit tasks that have been previously added. Editing tasks shall adhere to validity checks proposed in Functional Requirement 3.2.1.5. The system shall not allow users to edit tasks that are past due. A user will only be able to edit tasks for courses that they have created.

Use Case Name	3.2.1.7 Edit Task
Goal	User edits a previously added task
Primary Actors	User
Preconditions	3.2.1.6, Course that task exists in belongs to user
Postconditions	The edited task is saved and updated in planner.
MSS	<ol style="list-style-type: none">1. User navigates to view tasks from calendar view2. User selects 'E' on task they wish to edit.3. System populates add task view4. User edits information5. User presses add task6. Task is saved and planner is updated
Extension 1	2a. User selects a task they do not own 2a1. User is told they cannot edit tasks they do not own
Extension 2	4a. User enters invalid characters into one of the task fields 4a1 User is told to correct the invalid field.

3.2.1.8 Functional Requirement 8 - View Task

The system shall allow users to view a task by selection of the day from the home page by clicking on a date. If a date is selected that contains multiple tasks they will be vertically stacked on the view day screen.

Use Case Name	3.2.1.8 View Task
Goal	User views tasks for a specific date
Primary Actor	User
Preconditions	3.2.1.6 or 3.2.1.10
Postconditions	User is given a view of tasks for the selected date
MSS	<ol style="list-style-type: none">1. User selects a date from the calendar view.2. System provides a view of all tasks for the selected day if any exist.

3.2.1.9 Functional Requirement 9 - Delete Task

The system shall allow users to delete tasks that have been previously added to courses. A user will only be able to delete a task for a course that they created.

Use Case Name	3.2.1.9 Delete Task
Goal	Delete a task from the course
Primary Actor	User
Preconditions	3.2.1.6 or 3.2.1.10, and user is owner of selected course, task is in the future
Postconditions	User successfully deleted a task from their course.
MSS	<ol style="list-style-type: none">1. User selects a date from the calendar view.

	2. System populates task view 3. User selects "X" on the task they wish to delete 4. System deletes the task from their course and updates planner.
Extension 1	2a. No tasks to display for selected date. 2a1. User is shown an empty task view.
Extension 2	3a. User selects a task they do not own. 3a1. System alerts the user that the task cannot be deleted because they don't own it.

3.2.1.10 Functional Requirement 10 - Complete a task

Use Case Name	3.2.1.10 Complete Task
Goal	Mark a task as completed
Primary Actor	User
Preconditions	3.2.1.8, task is in the future, user is the owner of the course that the task belongs to
Postconditions	User successfully marked a task as completed or finished
MSS	1. User selects a date from the calendar view. 2. System populates the task view. 3. User selects the green check on the task he/she wishes to mark as completed. 4. System will mark the specified task as completed and it is greyed out on the calendar view and task view.
Extension 1	2a. No task for the date. 2a1. User is shown an empty view.
Extension 2	3a. User selects a task they do not own. 3a1. System alerts the user that the task cannot be marked with completion because they don't own it.

3.2.1.11 Functional Requirement 11 - Subscribe to Course

The system shall allow users to subscribe to a course created by another user. This will populate their calendar with the selected courses information. Users subscribed to a course cannot alter or append any information to a subscribed course.

Use Case Name	3.2.1.11 Subscribe to Course
Goal	User successfully subscribes to another users course.
Primary Actor	User
Preconditions	User is not already subscribed to this course and 3.2.1.12
Postconditions	User will be subscribed to the new course and the planner will be updated.
MSS	<ol style="list-style-type: none">1. User presses subscribe button on the course that they wish to subscribe to.2. User is asked to confirm subscription3. Subscription is confirmed and users calendar is updated with subscribed courses information.
Extension 1	<p>2a. User already has a course with the subscribed courses name.</p> <p>2a1. System asks user if they wish to overwrite the already existing courses information.</p> <p>2a2. User confirms and data is overwritten or user cancels and subscription is aborted.</p>

3.2.1.12 Functional Requirement 12 - Search for Courses

The system shall allow users to search for courses based on keywords. These will be alphanumeric and no longer than 8-20 characters.

Use Case Name	3.2.1.12 Search for Courses
---------------	-----------------------------

Goal	User is able to search for courses that they specify by keyword.
Primary Actor	User
Preconditions	User has navigated to search function. Other users have added courses.
Postconditions	User will be presented a list of courses to subscribe to.
MSS	1. User navigates to search menu 2. User types in a keyword or phrase to be searched. 3. System populates the window with relevant results
Extension 1	2a. User enters invalid characters into search bar. 2a1. System will return no results.
Extension 2	3a. System finds no results matching search criteria 3a1. System reports that no search results were found.

3.3 Performance Requirements

This section pertains to system attributes and expected load and run times for the user.

3.3.1 Importing Courses

Importing courses after subscribing will not take longer than 3-5 seconds.

3.3.2 Changing screens or navigating the application

There will be at most 1-2s of lag time between changing views from any screen to another screen.

3.3.3 Addition of tasks, subscriptions, and courses

Whenever a new element is added to the application (Course, Task, Subscription) any screen that makes use of that data must reflect the change upon the next viewing.

3.4 Design Constraints

Constraints that can be imposed by other standards, hardware limitations, etc..

3.4.1 Application Memory and Hard Drive usage

The application shall keep all memory usage within the expected bounds of the AWS service provided by the instructor.

3.5 Software System Attributes

The requirements of this section pertaining to the reliability, availability, maintainability, and portability.

3.5.1 Reliability

The system shall produce accurate imports of shared courses 99% of the time.

3.5.2 Availability

The systems database element that controls subscriptions, courses, and tasks shall be available 95% of the time.

3.5.3 Maintainability

The system should be extensible, that is, future development of the application should not require major reworking of the previous systems.

3.5.4 Portability

The system will run on Android devices only, but should run on at least 80% of all devices.

3.5.5 Security

The transmission of passwords between client and host shall be encrypted so as not to allow others to sniff username/password combinations.

3.6 Other Requirements - None