WASHINGTON STATE UNIVERSITY
VANCOUVER

Software Requirements Specification

for

Pro-Crastinator

Version 1.0

Prepared by

Group Name: Once We Were Programmers

Spencer Ross	**11686231**	**s.ross@wsu.edu**
Sarah Mathes	**11065825**	**sarah.robison-mathes@wsu.edu**
Kyle Stennfeld	**10722379**	**k.stennfeld@wsu.edu**

Date: October 25,2019

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Sarah Mathes, Spencer Ross, Kyle Stennfeld	First version of system require specification requirements document. System is in design phase.	10/25/2019

1 Introduction

Students are in need of a concise schedule organizer in order to meet the demands of their busy schedules. The goals of this system, Pro-Crastinator, are:

1. To be a time management system that is unencumbered by unnecessary features or complicated graphic user interface.
2. To improve the lives of students by organizing their classes, homework, and tests.

This system is intended to be a simple interface that the student can customize for their own workload.

This document outlines the high-level requirements for this system. The use-case of the system is also included, along with the overall description, and other non-functional requirements.

1.1 Document Purpose

This document describes the requirements for the web application Pro-Crastinator version 1.0. The first section serves to introduce the scope of the system, its audience, and an overview of the document itself. The second section details overall description of of the system, including the operating environment, system functionality, and user documentation. The third section details the system's hardware and software interfaces, and user interfaces.

The scope of this document encompasses all features of the system, and includes the system log-in, class management, grade calculation, and assignment organization. The system log-in is the process by which the student logs into their schedule. Class management is the sub-system where a student adds a class to their weekly schedule. Grade calculation is the sub-system where a student can edit the breakdown of their class grades and calculate a theoretical final grade based on the syllabus and submitted assignments. Assignment organization is the sub-system by which a student can add an upcoming assignment or test to their schedule.

1.2 Product Scope

The software described in this document is the web application Pro-Crastinator. Pro-Crastinator is a schedule planner intended to be used by students in order to manage their school workload. The software system saves a user's information, primarily their name, classes, and assignments, and recalls them upon the user logging in. Once logged in, the user can view a calendar of their classes and assignments, edit their class schedule, add a new assignment, edit existing assignments, and calculate their class grade.

Although many student planner applications exist, one of the main goals of Pro-Crastinator is to have a simple graphic user interface where the student can easily edit their class schedule and assignments from day to day. Upon logging in, the student is presented with options to either view, edit, or add to their class workload.

One of the core advantages to this system is that the user will not have to navigate through several menus, calendars, or course webpages in order to organize their workload. This lightens the burden of needing to mentally keep track of multiple assignments and tests across a number of classes. The system is a simple way to organize and then visually see the user's school workload.

1.3 Intended Audience and Document Overview

The primary intended audience for this document are developers of the system Pro-Crastinator. The purpose of this document is to formally state the functionality of this software system and its subsystems. Developers can refer to this document during development in order to solidify core goals, and to maintain consistency. The section that is most relevant to developers is section 3.1.3, Software Interfaces, as well as 2.2, Product Functionality.

Similarly, project managers should use this document to drive consistency as the system is developed and updated. Project managers should primarily refer to the section 2.1, Product Perspective, and 2.2, Product Functionality, in order to keep the main goals of Pro-Crastinator in mind. They should also familiarize themselves early in development with the Use Case View in section 3.3.1, which outlines the functionality of the system and the relationships of its subsystems.

The secondary audience for this document is the professor of CS 320 at Washington State University, Fundamentals to Software Engineering, Dr. Xinghui Zhao, who will be reviewing and grading the development of Pro-Crastinator. The section that is most useful for Dr. Zhao is section 3.3.1, the Use Case View, because it is the easiest way to visually understand the relationships between the system, its users, and its subsystems. Additionally, section 2.2, Product Functionality, will be useful for Dr. Zhao to refer to during system development, and when grading the final version of Pro-Crastinator.

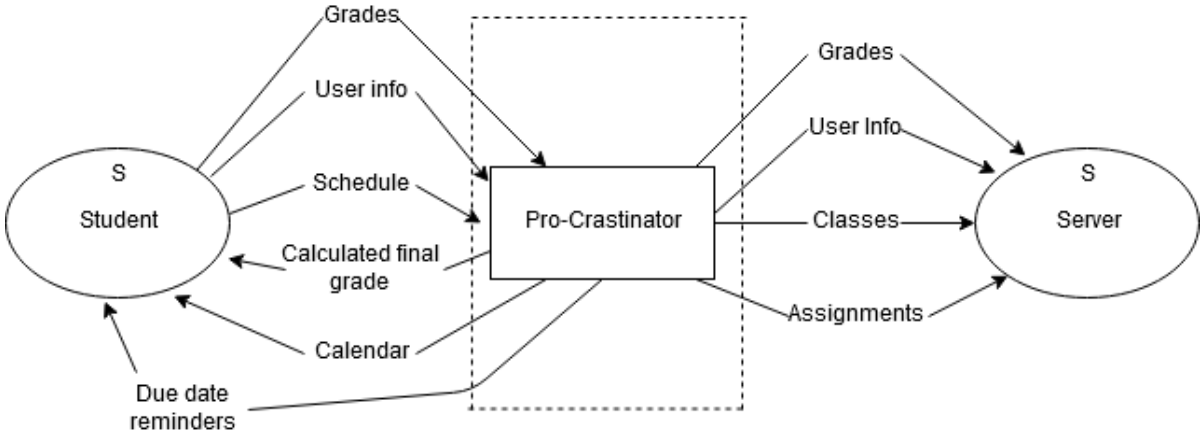
This document is also intended for product testers of Pro-Crastinator. If at any point product testers have questions about the intended functionality of the system, they should refer to this document. Product testers should start with the use case view, section 3.3.1 which will give them an overall view of the system's operation. They should then move on to section 3.1.1, User Interfaces, which provides guides to the user interface and its operation.

1.4 Document Conventions

Formatting coventions in this document follow the standard IEEE formatting requirements. Standard font for this document is Arial, size 11, with one inch margins. Lines are single spaced, with double spaces between paragraphs. New paragraphs are not indented. Comments are italicized. All headings are bold, size 14, and flush left. Sub-section headers are bold, size 12, and flush left.

2 Overall Description

2.1 Product Perspective



The Pro-Crastinator is a new, self contained product that utilizes existing Javascript libraries. The user is able to update their personal info, including their username, email address, and password. They can then add a class to their schedule, which Pro-Crastinator saves in the server. After a class has been added to their schedule, the user can create an assignment, of type homework or test, under that class. The assignment includes the class it belongs to and the date that its due. Pro-Crastinator also saves this information to the server. The system uses the class and assignment information to generate a calendar of class times and due dates. It also generates reminders for assignment due in the next twenty-four to forty-eight hours, which is delivered back to the user. The user also has the option to enter the grade breakdown for their class, so that the system can calculate a final grade based on theoretical assignment grades.

2.2 Product Functionality

The first function that the user can perform upon creating a profile and logging in is setting up a class. When a new class is created, the user can save:

- The name of the class
- The name of the professor
- Contact information for the professor, such as an email address
- If it's an in-person class or an online class
- If it's an in-person class, they can add the start and end times and days that class is held
- Optionally, they can edit the final grade breakdown by percentages. For example, they can enter that attendance is worth 10%, exams are worth 30%, homework is worth 30%, and the final is worth 30%.

Once a class is created, the student can add an assignment. When they choose to add an assignment, the user edits:

- The class that the assignment belongs to
- The type of assignment: homework, or test
- Due date of the assignment (including date and time of day)

- Optional: notes about the assignment

After classes and assignments have been added, the system populates a calendar with visual representations of the classes and assignments. In the calendar view, the student can filter by:

- Assignment type
- Due date
- Class

The student can also click an assignment or class in the calendar view to see more information.

If the student has edited the grade breakdown for a class, the student can calculate their final grade based on theoretical grades of an assignment. This grade calculator is one of the options on the main landing page.

2.3 Users and Characteristics

The users that Pro-Crastinator is designed for are middle schoolers, high schoolers, undergrad students, and grad students. The system's features are applicable to students of all levels. The users that Pro-crastinator can help the most are undergrad and graduate students, because their classes schedules vary the most. These two groups of students also have the most demanding workload, and need to organize it around family and work life. The format of this application might also be useful to educators, but they are not an intended user for Pro-Crastinater.

This system may be used by a younger audience with limited technical skill, but the application will feature a brief tutorial to facilitate ease-of-use. Pro-Crastinator will also use non-technical vocabulary in order to make it accessible to younger students.

The feature that is most helpful for students is the calendar view, which displays their class schedule and assignment due dates. This allows for a high-level view of their workload in an interface that is visually simple. It is also a way to consolidate assignments across multiple classes and institutions. This way, the student does not have to navigate through multiple menus, class websites, or third-party platforms.

2.4 Operating Environment

Pro-Crastinator is a web application. It is intended to be run on a desktop or laptop computer. The computer must have internet connection, either wired or wifi, in order to access Pro-Crastinator. No downloads will be required to use the system, but the internet access should be reliable. The machine must also have an internet browser installed, with up-to-date Java plug-ins. Pro-Crastinator aims to be compatible across all internet browsers. It will not be optimized for mobile use on smart phones or on tablets.

2.5 Design and Implementation Constraints

This will be the first web application designed by the programmers in this team and given the time to design and implement the application is six weeks, timing could be a restraint to produce a final product. This is due to a large amount of time being spent on research.

The Pro-Crastinator will have to use an open-source server for the purpose of the web application's ability to save user accounts and their data. This server will have to have some level of encryption to protect user's login information.

The user interface will also be under some constraints since it will only be optimized for use on a desktop computer. There are no plans to implement a user interface for mobile devices. However, there will be implementation in the program to allow it to be used on different display screens

2.6 User Documentation

The user documentation will be a separate page of the website. This page will have to include sequential steps on how to create a user's account, along with troubleshooting instructions if the user has problems creating an account. Furthermore, the help page will have to include tutorials on how to operate the account once the user has logged in to their account. This will include tutorials in the following:

- Create a class
- Assignment pop-up page
- Calendar
- Grade calculator page

Other documentation will include a Frequently Asked Questions page. This page will include common problems of users, and the solutions to those problems. This can also be included on the help page.

2.7 Assumptions and Dependencies

Dependencies for the design would be in the form of third-party software for the following:

- JavaScript packages for creating a web application
- Server for storage of user data
- Encryption package for protecting user's data

Assumptions that could affect the design:

- User has dual enrollment and wants to have separate classes for each institution
- User wants to include non-school events on their calendar

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The user interface will have multiple pages to help the user organize their school schedule. One page will have a list of the user's classes. Each class will then list all assignments along with the grade for the assignment. This page will give the user the option to add an assignment for each class. When the add assignment icon is initiated, a pop-up screen will be produced allowing the user to enter the name of the assignment, the score they received, and the grade weight of the assignment. The assignment screen will also give the user the option to add a due date instead of a grade.

Another page will produce a What If calculation for future assignments. This page will allow the user to add arbitrary grades to each ungraded assignment to see what their potential grade could be in the class.

Final page will be a calendar view of the user's classes and assignments. The user will be able to get an overhead view of when they have class and when their assignments are due.
Other user interface details will be a side bar navigation that list each class and an option to add a new class.

3.1.2 Hardware Interfaces

Since this is a web application, hardware interfaces do not have a substantial role with the Pro-Crastinator design. Only requirements would be a desktop running a recent operating system with a web browser. The desktop will also need ethernet connection or wireless card.

3.1.3 Software Interfaces

The Pro-Crastinator will be operated on a web-browser.

3.1.4 Communications Interfaces

The Communication interface shall use FTP for uploading any data created/changed by the user to the server. The server shall store accounts and accounts' data. The Pro-Crastinator system shall use HTTP to allow user to view the planner app in a web browser on their device by downloading the website from the server.

The data in the server will be encrypted. The Pro-Crastinator system should use the AES-256 encryption standard.

3.2 Functional Requirements

By use case:

Make Class:

- Mandatory fields for the name of the class, first day of class, last day of class, the day of the week that class is held, and hours of the day that class is held
- Optional fields for professor's name, professor's contact information, and breakdown of the final class grade
- Method to save the class and store it on the server
- Error if the class time or day overlap with an existing class
- Once today's date has moved beyond the final date, the class is removed automatically.

Make Assignment:

- Mandatory field for what class the assignment belongs to (of classes that have been created by Make Class), what day and time the assignment is due, and what type of assignment it is (either test or homework).
- Optional field for notes about the homework, limited to 300 characters.
- Assignment is saved on the server. Student has the option to mark it as completed to remove it from the server, and therefor from the calendar.
- Error if the due date is past the last day of class.

View Calendar:

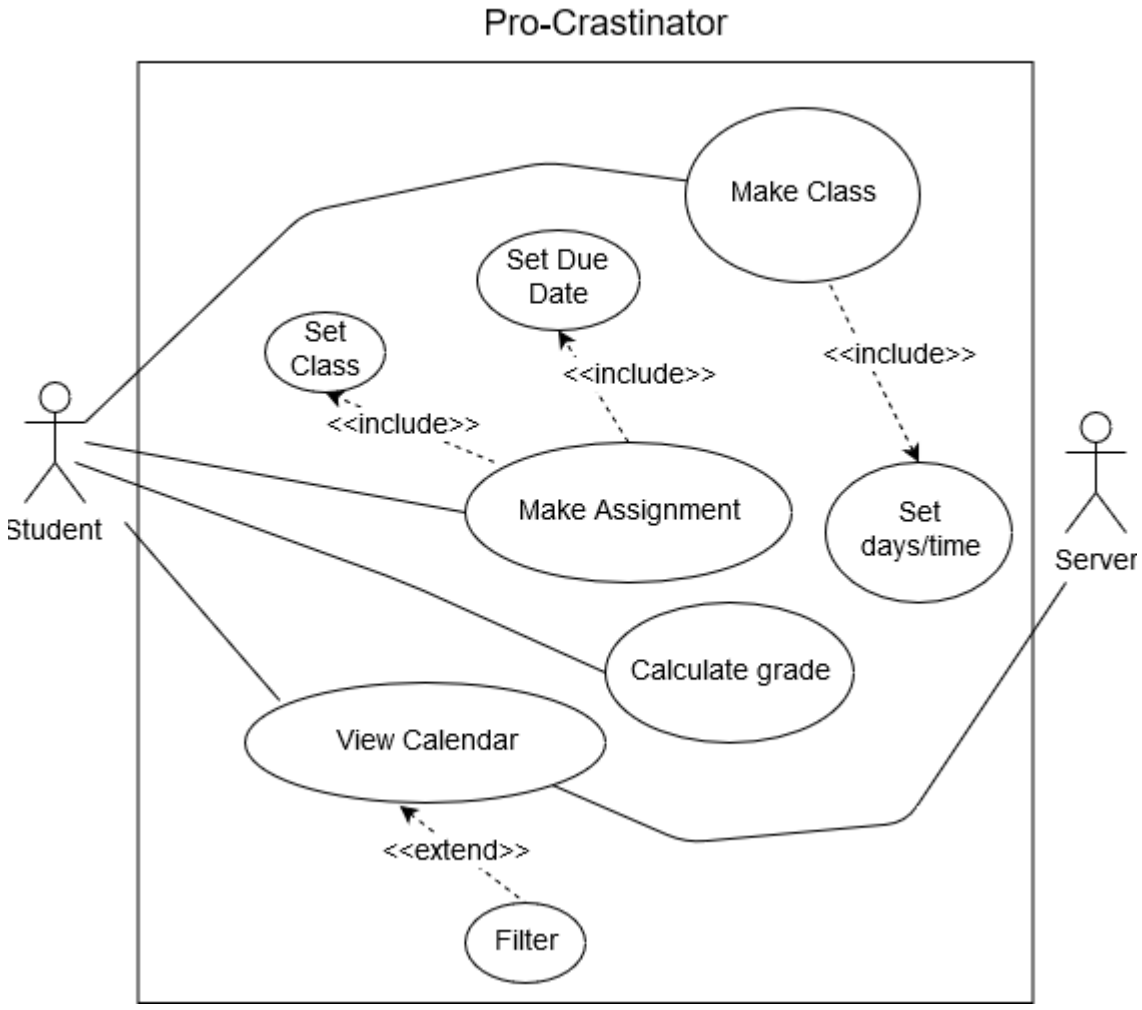
- Visual, grid representation of a calendar, with abbreviated names of assignments and classes placed by date.
- Method to use the classes and assignments saved to the server in order to populate the calendar.
- Method to implement an optional filter by class, or by assignment type.
- Assignments and classes can be selected to go to a detailed view. In this view, the student can mark it as complete and it will be removed from the server.

Calculate Grade:

- Select a class from an existing class. If the grade breakdown has been added to the class, use those grade percentages. If not, prompt the user to add to the following:
- Mandatory fields for the percentage of the final grade that the assignment is responsible for, the student's current grade, and the theoretical grade of the assignment.
- Method that returns the studen't theoretical final grade based on calculations using the mandatory fields.
- Nothing from this use case is saved to the student's account on the server.

3.3 Behaviour Requirements

3.3.1 Use Case View



- Log In: Using credentials created during account creation, user logs in to their account. Upon logging in, they are reminded of upcoming assignment due dates.
- Make Class: Student creates a class in their schedule. Required fields are class days, start and end times. Optional fields are class type (online or in-person), professor contact information, and grade breakdowns.
- Make Assignment: Student creates an assignment. Required fields: Due date, due time of day, class. Optional fields: Notes about the assignment.
- View Calendar: Student can view a calendar which organizes their classes and assignments by day. In the calendar view, the student can filter by class, due date, and assignment type.
- Calculate Grade: If the student has entered the total grade breakdown for a class, they can submit a theoretical grade for an assignment and the grade calculator will calculate their final grade. If not, the student can enter their current grade, the percentage of their final grade that the assignment is responsible for, and the theoretical assignment grade, and the grade calculator will return their final grade percentage.

4 Other Non-functional Requirements

4.1 Performance Requirements

1. Sign in will take no more than 10 seconds
2. Website should display on Chrome, Safari, and Explorer browsers
3. Website should adapts to mobile browsers
4. Navigation for the website should be intuitive

4.2 Safety and Security Requirements

- Account information shall not be public.
- Account data shall be abstracted from developer view, useable but not readable.
- User data will be encrypted and given account key.
- Account usernames will be stored in server.
- Account Password shall be encrypted in server and displaced in encrypted form with the account key.

4.3 Software Quality Attributes

4.3.1 Reliability

The website shall be useable from a variety of web browsers specified in 4.1. The planner system will allow students of any major use it. Pro-Crastinator should use the most commonly used versions of html, CSS, and Javascript to ensure maximum compatibilty.

4.3.2 Portability

It should be accessible from Desktop and mobile devices. The website should automatically detect what kind of device is trying to display the student planner. Once Pro-Crastinator has determined whether the device is mobile or desktop, the webpage should automatically format to match.

Eventually, Pro-Crastinator should be a mobile app. Currently, creating mobile applications is out of the teams scope.

Appendix B - Group Log

Group Meetings:

10/14/19:

Sarah, Spencer, Kyle met for ninety minutes to brainstorm potential ideas for their group project.

10/15/19:

Sarah, Spencer, Kyle had a sixty-minute online discussion about their favorite ideas and unanimously chose to design a student planner.

10/18/19:

Sarah, Spencer, Kyle met for 120 minutes and reviewed the SRS document, then broke it up into the following sections:

- Sarah would write sections 1.1 – 2.4
- Kyle would write sections 2.5 – 3.1 (including all subsections)
- Spencer would write 3.2 – 4.1

10/19/19 – 10/23/19:

Each group member worked on their respect sections, daily.
There was a daily group chat to discuss each member's progress and to ask any questions related to the design of the project. These meetings lasted about 30 minutes.

10/24/19:

Sarah, Spencer, Kyle met in person for 120 minutes to go over the final project and make any final changes to the project or SRS document.