

# Software Requirements Specification



CSCE 247: Software Engineering

Team Name: Waterboys

Team Members: Vansh Nagpal, Scott Do, Jagger Tanner, Adam Stewart

January 22, 2023

Version 2.0

## Table of Contents

Introduction	3
Stakeholders	4
Constraints	7
Overall Description	7
Business Use Cases	8
Functional Requirements	9
Non-Functional Requirements	9
Definitions and Acronyms	10
Competitive Analysis	11
References	15
Appendices	15

## Introduction

There are nearly 8 billion people alive around the world today and less than half a percent of those people possess the knowledge to code. With computer and information technology jobs on the rise, being able to code is a skillset with great demand. Whether someone has no coding experience or considers themselves a veteran in the coding industry there are always new skills and techniques that can be learned. That is why we want to create a console application Learning Management System(LMS) that allows people with all skill sets to learn and/or perfect their coding knowledge. This LMS will create suggested learning modules for users based on their knowledge of the coding language they wish to learn.

This document will cover information on the stakeholders that have a vested interest in the project, personas with more detailed information on our significant shareholders, any constraints that have been placed on our project, a description of the environment that our system will operate in, and a competitive analysis of three different products. A solid background in programming can prepare a person for careers in Artificial Intelligence, Software Engineering, Data Science, and many more. These careers define the future of engineering in many regards and yield extremely high paying salaries.

## Stakeholders

### Teacher, Scott Barker



Scott Barker is a middle aged computer science teacher with two kids. His career started after he graduated with his MS in Computer Engineering from Embry-Riddle Aeronautical University. Following this, he started to work at NASA as a junior engineer with many responsibilities. Yet, he somehow felt that he wasn't enjoying himself, fondly remembering that he really enjoyed his computer science classes and being a tutor. He decided to give his two weeks' notice to NASA and interview for an opening at a local high school. The administrators loved him and hired him on the spot as their new computer science teacher. He was tasked with teaching kids Intro to Computer Science. He had the choice to teach the kids Java or Python. He chose to teach them Python because of its dominance as a language in the Data Science field. However, he quickly ran into a problem: he wasn't the best teacher. He had never tutored for an introductory class or had to explain what a function or method was to someone who had never coded before. He was stuck. Here, he had left NASA to work as a high school teacher, but he couldn't even teach an introductory Computer Science class. He was having a crisis. His friend, a member of the new software developing think tank The Waterboys, told him to look into their new product: a premier Learning Management System(LMS) to teach people of all coding experiences how to code.

Scott immediately fell in love: he didn't have to explain the concepts anymore, the LMS took care of that. Instead now all he had to do was answer questions in class to reinforce understanding.

## Student, Brian Scott



Brian is a Sophomore at Georgia State University. He is majoring in computer science, and ultimately wants to be a software developer or data scientist at Google or Apple once he graduates. He did not have his career planned out until late in his senior year of high school, and as a result he doesn't have any experience coding outside of what he's learned in college. So far in his classes, he's learned Java and C++, but he knows that knowing those languages alone won't make him a very marketable employee once he's out of college. He graduated summa cum laude from his high school and has maintained a 4.0 GPA, so he is on track to be a desirable employee as long as he has the technical skills. He knows that Apple products are generally coded with the language Swift, and he generally thinks that the more languages he knows, the better. However, with the workload that being a Computer Science major entails, he doesn't have time to add any more courses to his already packed schedule. He plans to supplement his learning with an LMS that can help him pick up extra languages on his own time. After looking into it for a short period, he decided upon using a new LMS that he can use asynchronously. This way, whenever he's got a particularly packed week, he can put any additional languages on the back burner. However, when he's got a light workload he can put in some personal time to learn any languages that he thinks will be helpful.

**Client , Portia Plante**

Portia Plante is the CEO of a successful business, and has more than enough money as a result. She is married with one son, a 25-year old named Portio who still lives at home. He was fairly spoiled for most of his life, and never went to college or learned any trades because he could just ask his mother for money when he needed it. Recently though, Portia made up her mind that she wants him to be independent. She heard through the pipeline at work how much software engineers and programmers can make, and thinks that her son could pursue that. However, outside of her general knowledge that “programmers make money,” she doesn’t know the first thing about coding or software. She was thinking through the possibilities of what steps her son could take to learn how to code and considering the routes; college would be expensive as her son had already been out of school for nearly 7 years and wouldn’t have any scholarships. While she was thinking, she had an idea: rather than sending him to college, she could have a program developed to teach her son, as well as anyone else who has an interest, how to code. She was willing to pay out of pocket, and decided to get a team to start working on it the next day.

## Constraints

### Time Constraints

- The project's timeframe is limited to the duration of the Spring 2023 semester - or approximately 13 weeks to plan and design with only 2 weeks to create.

### Budget Constraints

- This project has been allocated a budget of \$0.

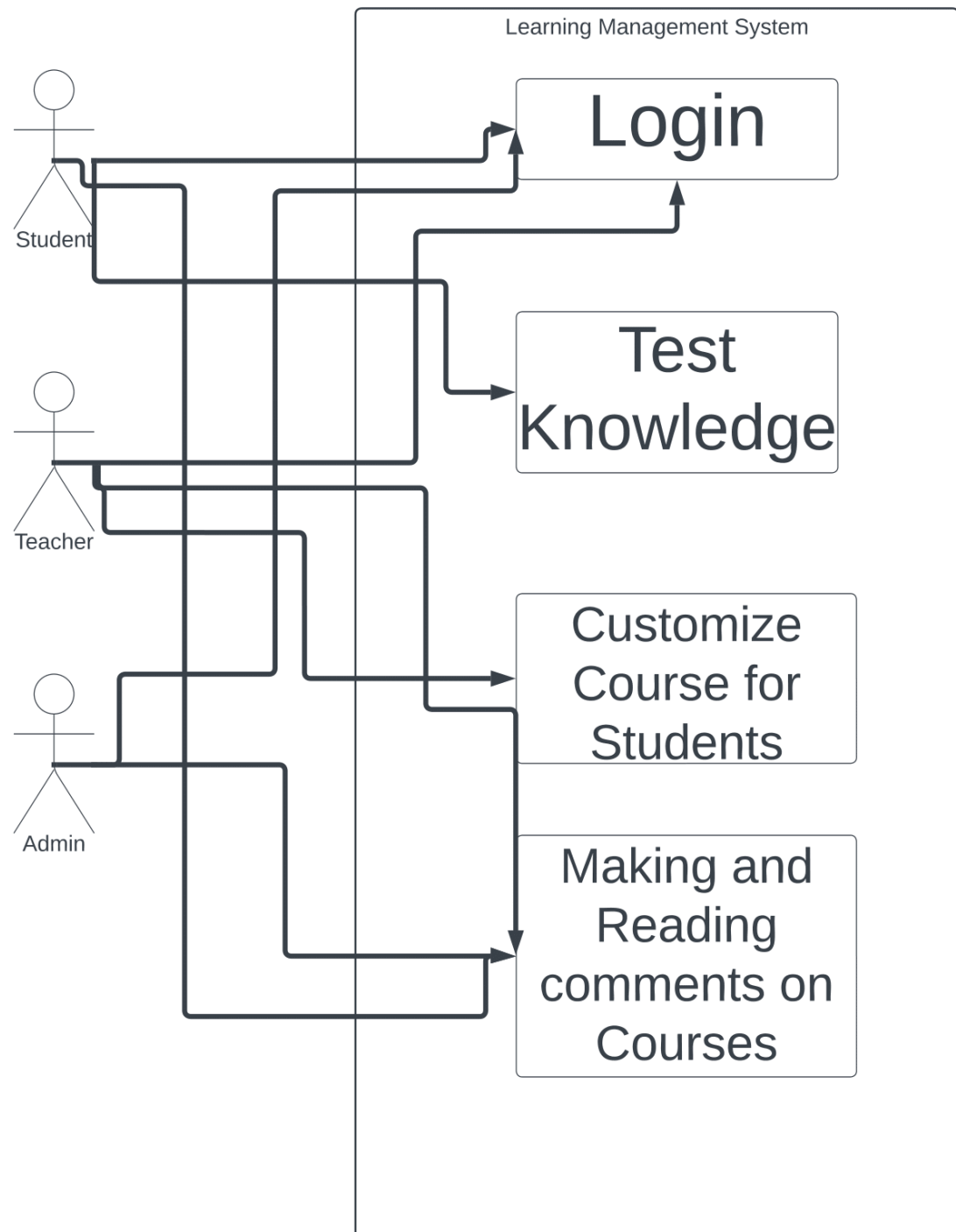
### Technical Constraints

- The product must be functional in a console.
- The product must be developed using the Java language.
- The project group may have varying experiences with developing a learning management system.

## Overall Description

This system must operate within a command line interface(windows cmd, bash, or powershell). The software must run from java source code, and a main driver must be compiled and executed for a command line UI. This code must be portable and must compile with other java compilers. Another main software system that will interface with this system will be github, which will allow for multiple developers to work on code at the same time.

## Business Use Cases





## Functional Requirements

### requirements

## Non-Functional Requirements

### Look and feel requirements

- The interface of the learning management system shall be designed to allow first-time users a clear understanding of what they can do.

### Usability requirements

- Courses shall be easily navigable in the learning management system.

### Performance requirements

- The learning management system shall be able to produce a user's chosen result in at most 2 minutes.

### Maintainability and support requirements

- The learning management system shall operate on Windows and Linux based machines.

### Security requirements

- The credentials of a user shall be secured in a safe manner.

### Legal Requirements

- The learning management system shall follow copyright guidelines.

## Definitions and Acronyms

- Learning Management System (LMS) - a software application for the administration, documentation, tracking, reporting, automation, and delivery of educational courses, training programs, materials or learning and development programs.
- Java - Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible.
- C++ - C++ is a high-level general-purpose programming language created by Danish computer scientist Bjarne Stroustrup as an extension of the C programming language, or "C with Classes".
- Swift - Swift is a high-level general-purpose, multi-paradigm, compiled programming language developed by Apple Inc. and the open-source community.
- Python - Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.
- Github - GitHub, Inc. is an Internet hosting service for software development and version control using Git. It provides the distributed version control of Git plus access control, bug tracking, software feature requests, task management, continuous integration, and wikis for every project.
- Command Line Interface - A command-line interpreter or command-line processor uses a command-line interface to receive commands from a user in the form of lines of text. This provides a means of setting parameters for the environment, invoking executables and providing information to them as to what actions they are to perform.

## Competitive Analysis

	<u><b>CodeHS</b></u>
<b>Strengths</b>	<ul style="list-style-type: none"><li>• It is easily integrable with a school classroom. Instructors are allowed to issue assignments alongside subtopics. They are able to issue coding assignments which will issue a sandbox IDE allowing students to experiment with solutions for their assignments, and instructors are able to keep track of their students' progress.</li><li>• It has a comprehensive curriculum with over 100 customizable courses in 6 languages. Students can get personalized feedback from remote, qualified tutors aside from their own instructor.</li></ul>
<b>Weaknesses</b>	<ul style="list-style-type: none"><li>• While it has a free option, it is limited in choice and its more comprehensive content is paywalled through three subscriptions depending on the scale of teaching.</li><li>• It is primarily oriented towards a school environment, and thus may not be helpful for an individual learner looking to introduce or hone skills with programming languages.</li></ul>
<b>Audience</b>	<ul style="list-style-type: none"><li>• The audience for this product is heavily focused towards high school and middle school students, mostly those who take courses that involve Computer Science.</li><li>• It is structured to provide instructors the means to introduce the topics of Computer Science in a learnable, interactive method.</li><li>• It does not focus on assisting college students, or individual learners.</li></ul>

	<b><u>Coursera</u></b>
<b>Strengths</b>	<ul style="list-style-type: none"><li>• Coursera partners with more than 200 universities to offer more than 3,800 courses to their users.</li><li>• Coursera offers nearly 2,000 courses to users free of charge.</li><li>• Coursera gives users to opportunity to watch lectures and complete tasks from wherever they have an internet connection</li><li>• The courses that Coursera does charge for cost significantly less than they would at a university</li></ul>
<b>Weaknesses</b>	<ul style="list-style-type: none"><li>• Fees can be difficult to calculate depending on your location</li><li>• Even beginner level courses require users to have prior knowledge of the subject. There are no courses that allow users to start without any knowledge</li></ul>
<b>Audience</b>	<ul style="list-style-type: none"><li>• Courser's target audience is quite large at ages 18-55 because they offer so many different products</li><li>• Coursera's target audience is those who did not attend or complete college and are looking to get college level certifications</li><li>• Coursera is made for varying levels of experience and appeals to anyone with basic level knowledge of the subject they wish to learn</li></ul>

	<b><u>Codesters</u></b>
<b>Strengths</b>	<ul style="list-style-type: none"><li>● Easily integrated within a classroom setting</li><li>● Allows teachers to make classes and enroll students</li><li>● Allows for ease of grading assignments and viewing student progress</li><li>● Teaches Python in a fun and easy way</li><li>● Allows for creativity as students are allowed to pick their own projects</li><li>● Graphics oriented approach to programming captivates interest of young minds</li></ul>
<b>Weaknesses</b>	<ul style="list-style-type: none"><li>● Bad for individual learning</li><li>● Requires users' teachers to make an enrollable python course</li><li>● Bad for more mature people who are not as interested in games</li><li>● Only teaches python and not other programming languages</li></ul>
<b>Audience</b>	<ul style="list-style-type: none"><li>● K-8 students who want to learn how to code</li><li>● Python is an easy to understand first language</li><li>● Intuitive platform for kids and teachers</li><li>● Game based teaching makes learning for kids in this age range captivated and interested in coding</li></ul>

## Summary

From the research and analysis of the learning management systems above, we have learnt many great details towards developing such systems. For one, our product will need to accommodate individual learners. Codesters and CodeHS are learning management systems whose primary audience is students who are in grade school, and those looking to learn programming on their own time will find these resources troublesome to use. Branching out in relation to audiences will allow the product to be more approachable and accessible to more people than limiting it to academics. As we look to make our product accessible to a wider audience, it would be of importance to ensure that it is approachable to beginners as well. Coursera is noted to offer a plethora of courses, but functions under the assumption that the user starts the courses with prior knowledge of them. Approaching beginners by providing a method to introduce them, such as a tutorial lesson, would benefit its users. From the analysis of the learning management systems above, our goal is to develop a product that will be approachable to people both inside and outside of academics, and of different backgrounds in regards to familiarity of programming.

## References

- <https://sweetishhill.com/what-percentage-of-the-population-can-code/>
- [https://en.wikipedia.org/wiki/Learning\\_management\\_system](https://en.wikipedia.org/wiki/Learning_management_system)
- [https://en.wikipedia.org/wiki/Java\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))
- <https://en.wikipedia.org/wiki/C%2B%2B>
- [https://en.wikipedia.org/wiki/Swift\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Swift_(programming_language))
- [https://en.wikipedia.org/wiki/Python\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))
- <https://en.wikipedia.org/wiki/GitHub>
- [https://en.wikipedia.org/wiki/Command-line\\_interface](https://en.wikipedia.org/wiki/Command-line_interface)

## Appendices

Any relevant information that can assist in understanding the requirements.