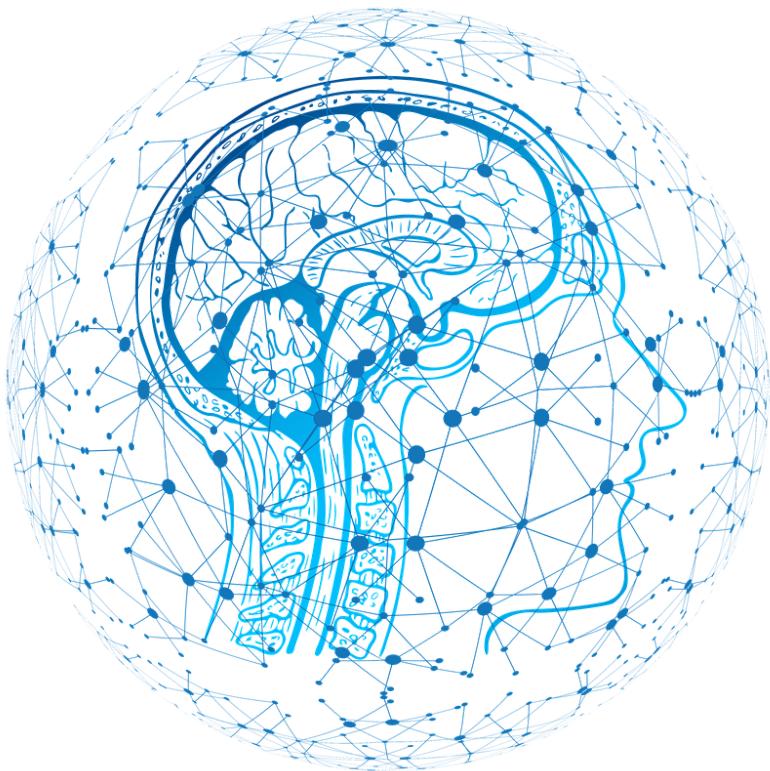


Coder Cruncher -

The Dynamic Programming Language Learning Management System



Software Requirement Specifications

CSCE 247: Software Engineering

Yusiff Cassimjee, Jason Tran, Burt Sumner, Shahraiz Aziz,

Michael Yang

Team Urameshi

January 22nd, 2022

Version 1.00

Table of Contents

1. Introduction	2
2. Stakeholders	2-5
3. Constraints	6-7
4. Overall Description	7
9. Competitive Analysis	7- 13

1. Introduction

Purpose

The U.S. Bureau of Labor Statistics is estimating the job market for software developers will grow 25.0% the next decade, which means many people will be interested in learning how to code. Our product is a learning management system that helps people learn different programming languages so that they can find jobs, or pursue whatever other programming endeavors they would like to undertake. We would like our learning management system to deliver an intuitive, easy to use, and quick to learn system that greatly boosts the end user's skill and proficiency with many programming languages. It will eliminate the perplexing nature and the overwhelming feeling of being thrown into a metaphorical abyss found across many learning management systems out there today.

Source:

<https://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm>

2. Stakeholders

- The Client/Investors
- Students studying Computer Science
 - Undergraduate University Students
 - Graduate University Students
- People interested in coding
 - High School Students
 - Hobbyists programmers
 - People who want to learn

Personas

Lucas Anderson

"I can't wait to start making my own programs!"

Motivation



Since Lucas is soon graduating from high school, Lucas has been looking into different majors that he might be interested in. As he stumbled across Computer Science, he was intrigued by what skills he might learn if he were to choose this major. His cousin Carl, who is studying Computer Science, recommended Coder Cruncher to dip his toes in to see if this is a field he would like to go into.

Goals



- To understand beginner programming techniques
- To try many different types of programming languages
- To understand basic computer science theory

Needs



- Concepts to be shown visually
- Learning material to be beginner friendly



AGE	18
GENDER	Male
OCCUPATION	Southside Senior High School Student
LOCATION	Greenville, SC

BIO

Lucas works part-time at his school's computer lab, so he helps out other students with computer problems. Lucas is a visual learner, and he often asks his teachers to draw out concepts, so that he can understand better. Lucas loves learning, and he usually learns things he is interested in online on his own time.

Jackie Thompson

"I want to learn JavaScript as soon as possible!"

Motivation



Jackie decided to take a website design class this semester, but the projects and homework assignments require Priscilla to use JavaScript, a language Jackie is not familiar with, to complete them. She emailed Professor Anderson, who is teaching her class, for online resources to learn JavaScript. Professor Anderson then referred Jackie to Coder Cruncher, a resource that his past students recommended.

Goals



- To become proficient in JavaScript in a timely manner
- To potentially learn other languages for future endeavors

Needs



- The option to indulge in challenging learning material
- Concepts to be explained thoroughly in words



AGE	21
GENDER	Female
OCCUPATION	Arizona State University Junior Undergraduate Student
LOCATION	Phoenix, AZ

BIO

Alongside being a full-time student, Jackie is an intern at a local IT company, so she does not have a lot of free time. As a junior computer science student, she is proficient in many languages such as C++, Python, etc. Jackie is an avid reader, and she prefers learning by reading text.

Kip Smithers

"I am smarter than I look!"

Motivation



After seeing his former co-worker struggling with learning how to code, Kip decided that a learning module system, that is suitable to those that haven't coded before, was the next billion dollar idea. After selling his family farm and quitting his construction job, Kip set out to create Coder Cruncher and make it the best learning management system on the market so that he could prove to his father that it was the right choice to sell the farm and quit his job.

Goals



- Dreams of investing in and creating the world's most successful learning management system
- He wants to be the next Elon Musk of the world.

Needs



- The user graphical interface to be simple and easy to understand
- The learning management system to be encouraging and supportive



AGE	33
GENDER	Male
OCCUPATION	Entrepreneur/ Former Construction worker

LOCATION Canton, OH

BIO

Kip Smithers is a hard working individual that was born and raised in the classical farm lands of Rittman, Ohio. He sold his family farm and is trying to become generational wealth by using the funds from the farm to hire a team of developers to create a new and exciting learning management system.

3. Constraints

There are several constraints that either we have relegated upon ourselves or we have been relegated to by circumstance.

- **Scheduling:** Due to all of us being full-time university students, we have been relegated to being left at the mercy of sporadic meeting times for meetings in person.
- **Correspondence:** Sporadic scheduling notwithstanding, however, it seems as though we can get a hold of one another a majority of the time, as we have opted to constrain communication to via Discord, which has both a mobile and desktop application, making it the most reliable platform for contact with team members
- **Workplace Environment:** We have decided to constrain any future in-person meetings to a specific location. The tentative agreed-upon location as of writing this is one of the below-ground floors of the Thomas Cooper Library. Nice and quiet and allows for us to have our own room in which we can collaborate in privacy and use the whiteboards provided within the rooms for articulating our thoughts and ideas.
- **Choice of Frontend:** We have chosen to simply relegate ourselves to meeting the base requirements of engineering a console-based application; only after satisfying that requirement can we entertain the idea of creating a graphical user interface for the application (GUI).
- **Choice of GUI API:** If we do make it to the point where we can feasibly upgrade our final product to having its own GUI, we chose to limit ourselves to something that comes out of the box with the Java base standard library. This API is known as Swing. It is the successor to the older AWT graphics

API, and the predecessor to the newer JavaFX graphics API; we would have chosen JavaFX if it weren't for the fact that JavaFX doesn't come prepackaged with the base libraries of the Java Development Kit (JDK) it is its own separate library that you must either explicitly specify the path to the modules when compiling the Java code, or edit your environment variables for Java to recognize those modules implicitly.

- **Money:** Being that the five of us are all college students, it is clear that a considerable constraint will be money. Our project's funding will only be relegated to the revenue provided by any future stakeholders. That is to say, funds gathered from angel investors, benefactors, venture capitalists, pre-orders from anticipative customers, etc. Even if we take into account any funds we might put into the development of the app, it will pale in comparison to any outside funding, which, even once combined, might end up short. This isn't to say this will cripple our project by any means, but it does mean we will have to tentatively temper our expectations.

4. Overall Description

In order for this learning management system to reach a majority of people, this application will be able to be accessed by anyone with a desktop or laptop device so that everyone, who is willing to learn how to program, is able to. The learning management system will have a plethora of courses that will each have educational content on their respective programming language. The user will have the option to indulge in the lesson module of their choice, so that they can forge their own learning path.

9. Competitive analysis

	TalentLMS
Strengths	<p>Implements gamification into the traditional LMS to motivate learners and help keep track of progress via features like achievements. These gamification settings can also be configured by the user.</p> <p>Contains intuitive add-ons and integrations like the iFrame (to add webpages) and wix pages that can be used to create any type of content to enhance the learning process.</p> <p>Features many types of test options, each consisting of simple but appealing features such as drag-and-match, fill-in-the-blanks, and reordering.</p> <p>Administrators can easily switch between roles and can benefit from features specific to each role.</p>
Weaknesses	<p>When compared to competitors, TalentLMS lacks complexity and instead favors simplicity and accessibility instead.</p> <p>The quizzes in this LMS do not allow the user to go back to a previous question for reference or correction, thus undermining the very goal of the product</p> <p>The procedure to set up answers for courses/questions is restrictive and pales in comparison to the more intuitive approach taken by competitors</p>
Focus	<p>There is a relatively heavy emphasis on gamification of the product, and badges, rewards, leaderboards, levels, and points are utilized to encourage the user to learn more</p>

	Open edX
Strengths	<p>Boasts credibility since it is jointly spearheaded by the likes of Harvard and MIT and is a non-profit product, which further highlights the product's genuine intentions.</p> <p>Extremely user friendly which allows for focusing on the learning process instead of focusing on navigating the tools. The pool of features at the disposal of the user is presented in a minimalistic manner which makes the interface appealing.</p> <p>The product is made not only with learners in mind but with teachers as well. Course authors can create simple or very complex courses all from the built-in studio.</p> <p>The product also supports a mobile application which makes the product portable, and it is an open source software which allows for great customization opportunities and allows each user to tailor the product to their liking if need be.</p>
Weaknesses	<p>Documentation for problems is extensive and very specific, and needs expertise for navigating through technical problems, despite the large community behind the product.</p> <p>The administration features are tedious to use and it can be difficult to review or track a student's progress.</p> <p>There is no standard procedure for bulk registering users, and certain features like the wiki section are uninvolved and lack interactive features.</p>

	<p>There is no offline access to content on desktops and the interface lacks drag-and-drop interactions.</p> <p>Scheduled learning is ineffective for users since it is dependent on a constant online connection.</p>
Focus	<p>Open edX dedicates much of its resources to ensuring a user friendly interface and navigation system and focuses on providing the services of a LMS while staying minimalistic. There is also great emphasis on the open source and highly customizable nature of the product.</p>

	Docebo
Strengths	<p>The product is very flexible and users are able to conduct training sessions anytime, anywhere. Docebo is also agile in the sense that it makes tracking training performance seem effortless.</p> <p>The product provides easy integration with business systems (such as CRM) and offers ready-made widgets for users who may not have prior coding experience.</p> <p>It allows users to organize courses with the appropriate customization and is a great platform to create material that needs desktop and mobile distribution. The dashboards also feature the ability to drag and drop widgets for user convenience.</p> <p>Offers multiple language support to cater to international audiences and has added localization opportunities that grant the</p>

	ability to add or modify existing translations
Weaknesses	<p>The product is restricted beyond the initial 14-day trial period and users must pay to use the product's services.</p> <p>Docebo presents the Inability to do item analysis on test questions without API and programming knowledge, thus somewhat restricting new users.</p> <p>The organization of assets inside a course is not up to par with the quality offered by competitors.</p> <p>While Docebo is packed with features and tools, there is a steep learning curve to the software and there are minimal resources built in into the product to help with this issue.</p> <p>Monitoring capabilities for managers are either missing in some respect or not well designed when compared to that of competitors.</p>
Focus	Docebo aims to serve as a LMS utilized mainly by enterprises & corporations to educate, train, and develop customers and partners in large groups as opposed to focusing on individualized and isolated learning. This sets Docebo apart as a LMS that makes its users feel part of a community of people.

Summary

	Strengths	Weaknesses	Focus
TalentLMS	<ul style="list-style-type: none"> +Gamification of the traditional LMS +Add-on features and integrations +Multiple roles for administrators +A variety of test type options 	<ul style="list-style-type: none"> -Features lack in complexity -Restricted control over test types like quizzes -Restrictive procedure for setting up tests when compared to competitors 	<ul style="list-style-type: none"> -To use gamification of the traditional LMS to appeal to larger audiences
edX	<ul style="list-style-type: none"> +Friendly and minimalistic user interface +Built in studio that facilitates creation of easy and complex courses +Open source software code +Supports a mobile application 	<ul style="list-style-type: none"> -Needs expertise for navigating through technical problems -Tedious administration features -No way to bulk register users -No offline access to content 	<ul style="list-style-type: none"> -To create a highly customizable software with a welcoming UI for newcomers
Docebo	<ul style="list-style-type: none"> +Offers ready made widgets +Desktop and mobile distribution +Flexible session scheduling +Easy-to-use progress tracking 	<ul style="list-style-type: none"> -Difficult item analysis on test questions -Sub-par organization of content in courses -Steeper learning curve than competitors -Limited monitoring capabilities 	<ul style="list-style-type: none"> -To serve as a tool utilized by corporations and enterprises to develop and train employees

Having learned of the different kinds of products currently available on the market, all of which are trying to accomplish a goal similar to that of ours, we can outline a clearer and more intuitive approach to our own product. It is clear from the evidence presented above that it would be very difficult to create a product that receives a perfect score on all fronts, but that is not to say that certain features cannot be given priority over others. While Open edX spent resources on ensuring a welcoming UI, it lacked the quality of life features that Docebo had. Similarly, TalentLMS focused far too much on gamification of the product to attend to the need of complexity in its features. With our product, we will strive to find a balance between some of these essential features, and only after we have laid the foundations for our product will we move forward to ensuring that quality of life features are not neglected. It is evident that the heart of a LMS lies in its capacity for enabling learning for its users and so, in the early stages of development, we will avoid favoring a pretty surface over a functioning backend.