uC[≡]de

uCode - Run Handwritten Code Anywhere

Dr. Chen: Senior Project I

Instructor Comments/Evaluation

Table of Contents

Abstract

This project's main purpose is to recognize handwritten characters and interpret them as code using a smartphone. The code is then compiled and its output displayed on the phone's screen. This project aims to be a solution geared towards students and teachers in low-income schools. By having this program recognize and compile handwritten code, this eliminates the need to buy a classroom set of computers to teach a computer science course. The only hardware required is a smartphone. With the recent improvements in machine learning technology, software that can recognize handwritten characters is now possible.

The program will run as an application on a phone that takes a picture and sends it to a server. The contents of the picture will be interpreted as text using machine learning techniques. Afterwards it will be compiled and the output will be sent back to the phone. This application will enable the end user to quickly and conveniently compile code live. It will also serve a purpose for an instructor, allowing them to validate and run code in an educational setting.

The purpose of this document is to define the requirements of the software that is being made. The product should satisfy the requests made in this document. Anyone who uses this product should expect every feature to be explained here in detail, including the team details, workflow, product background, and specifications. The use cases of this project will also be shown.

Introduction

Background/Overview: Teaching a computer science course has never been an inexpensive investment as "schools across the country are spending billions on various kinds of technology for the classroom" (NPR Staff, 2013). Setting up a classroom with the appropriate equipment (e.g. wifi/internet, cyber security, classroom computers) can cost upwards of hundreds to thousands of dollars. Low income schools can't spare the expense to include a computer science class into their curriculum. This denies students an education in an innovative and continually rising field. To be more cost effective, cell phones will be incorporated into the classroom, which in NPR's (2013) study conveyed 73 percent of teachers said that "cell phones are now either part of their classroom experience, or their students' classroom experience". This project is designed to eliminate the need for a classroom set of computers while still allowing for an effective way to teach computer programming.

Objective of Project: The objective for this project is to take a picture of handwritten code, compile it, and then return the response to the phone. This can be done by using the phone's camera to send a picture of the handwritten code to a remote server. The server uses machine learning to interpret the handwritten letters into a text document. The document would then be compiled and report any errors to the user. The user can manually edit the recognized code or interact with the program as it is executed.

Team details & dynamics:

The team assignments are shown in the table below:

Team Member Name	Team Role
	Specs & Implementation
	Analysis
	Design
	Presentation

Application Domain

Project Context:

The main domain of this project is an educational setting, either personal or in the classroom. It can also be used by professors to grade their exams, or whiteboard interviews. They can scan or take pictures of their student's written code, compile, and execute it. This could also be used in whiteboard interviews to get instant feedback on a potential employee.

For additional information on any terms conveyed in this document, please refer to the Technical Glossary in Appendix I.

Initial Business Model

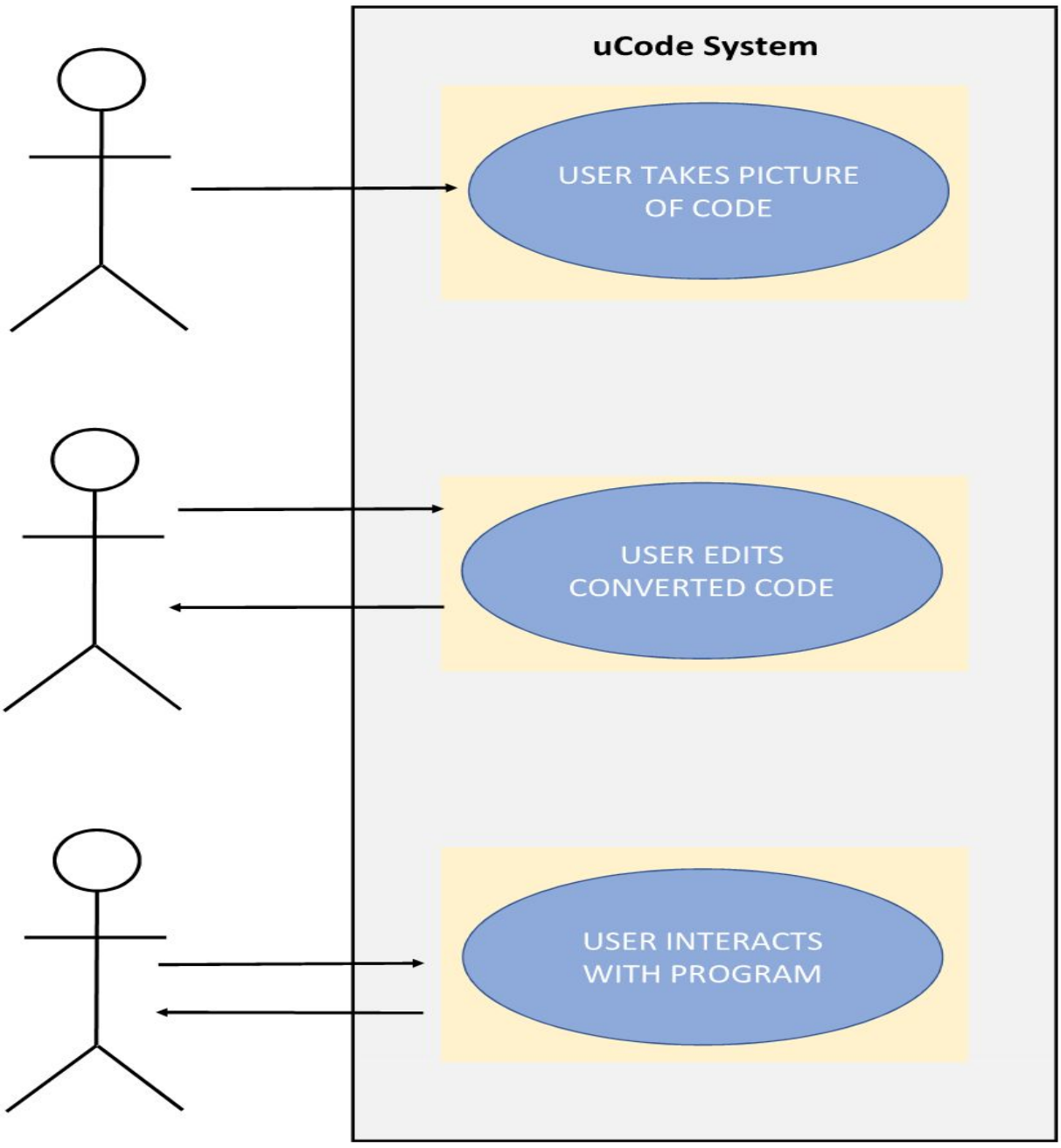
Operating Environment:

This project is specifically geared towards educating students in third world or low income communities. These communities may not have the budget to provide adequate hardware. Students will only need to have access to a mobile device, running a common operating system like Android or iOS. Revenue would be earned from advertisements inside the application which would be free version on the Google Play or Apple stores; an in-app purchase would disable these advertisements. For classroom use, revenue would come from the schools. The release of this software could be free with potential governmental programs. In the future, the company could evolve into a non-profit, making government funding available for the projects to offset the costs of development and maintenance.

Description of Data Sources:

The source of data for this project will be the image taken by the user. The image will then be sent off the phone to a backend server, which will determine the text on the image. An additional feature may allow the user to manually edit the recognized code, in case it is incorrect.

Use Case UML Diagram: (Figure 1)



Description:

The user scans an image of the code using a mobile device camera. Once executed, uCode will return either the compiled code or error descriptions that will be open for the user to edit. After user is satisfied with edits, it may be passed back to backend server for compulation and execution. The user may interact with the program when it's executed.

Initial Requirements

This project requires the development of a mobile application, preferably running on both iOS and Android. The application will need to take pictures and send them to a backend server to convert the image to text. This text will then be either sent back to the phone or compiled and ran on the server. Either way, the output will be sent back to the phone. A later feature would allow the user to handle user input via a socket connection.

Functional:

1. Take a picture with a smart mobile device and send it to the backend server.
2. Convert the picture to source code using a machine learning algorithm.
3. Run the source code on the backend server.
4. Send the results back to the mobile device.

Nonfunctional:

1. The front end will run on iOS and Android platforms.
2. The ackend will be a Linux/Unix operating system operating on the cloud.
3. It will be reliable in that the backend server will be in the cloud and can be scaled based on demand and is always running.
4. Response times should be lower than 10 seconds.
    a. Five seconds devoted to the machine learning and translation.
    b. Five seconds for the compilation and execution time.
    c. However, response times may vary heavily on the task being executed.

Documentation:

Documentation will be provided to the customer at the time of release. The documentation will show the inner workings of the application (what API's / libraries were used). An API is an "Application Programming Interface" that companies have "built dedicated URLs that return pure data responses" (Gazarov, 2016). It will also include a user manual, so users understand how to properly use the application for the best results.

Testing / Revisions

Testing will be completed by running through many different images of source code. This can be easily obtained from Google Images. It will be functionally tested by taking actual images of source code written on paper. The hardest and most vulnerable part of the project will be the machine learning. When the code is recognized incorrectly by the machine learning algorithm, an inevitable problem, the manual editor and possible suggestion engine should be tested as well. A system would need to be implemented to prevent asailents from writing and running malicious code on the backend server. With virtualization or containerization of each user's code, the access and resources each user can use on the backend could be limited.

Revisions will be managed through Git, a software versioning system to manage code changes in team based environments. Code will be checked in and checked out of the devstream by individual developers. Then with one other developer's approval, the code can be moved into the mainstream build.

References

Beal, V. (n.d.). API - application program interface. Retrieved October, 2018, from https://www.webopedia.com/TERM/A/API.html

Beal, V. (n.d.). TCP - Transmission Control Protocol. Retrieved October, 2018, from https://www.webopedia.com/TERM/T/TCP.html

Graphical user interface. (2018, October 19). Retrieved from https://en.wikipedia.org/wiki/Graphical_user_interface

Gazarov, P. (2016, August 13). What is an API? In English, please. – freeCodeCamp.org. Retrieved from https://medium.freecodecamp.org/what-is-an-api-in-english-please-b880a3214a82

NPR Staff (2013, March 03). Teaching 2.0: Is Tech In The Classroom Worth The Cost? Retrieved from http://www.npr.org/2013/03/03/173372736/teaching-2-0-is-tech-in-the-classroom-worth-the-cost

Oracle. (n.d.). What Is a Socket? Retrieved from https://docs.oracle.com/javase/tutorial/networking/sockets/definition.html

Vision API - Image Content Analysis | Cloud Vision API | Google Cloud. (n.d.). Retrieved October, 2018, from https://cloud.google.com/vision/

Appendix I: Technical Glossary

API (Application Program Interface): a set of routines, protocols, and tools for building software applications (Beal, 2018).

Android: a mobile operating system developed by Google.

Backend server: cloud-based Linux server.

Compiler: a program that converts instructions into a machine-code or lower-level form so that they can be read and executed by a computer.

Functional requirements: basic functions completed by the application.

Google Vision: API used to access Google's image recognition capabilities. "Cloud Vision offers both pretrained models via an API and the ability to build custom models using AutoML Vision to provide flexibility depending on your use case" (Google Vision, 2018)

GUI (Graphical User Interface): a form of user interface that allows users to interact with electronic devices through graphical icons and visual indicators (Graphical User Interface, 2018)

iOS: a mobile operating system created and developed by Apple.

Machine Learning: a branch of artificial intelligence where systems can learn from data, identify patterns and make decisions with minimal human intervention

Networking: a process that cultivates information and data on a digital telecommunications network and disperses it over data connections to different nodes

Node: a point on a computer network (e.g. cell phone, server, computer, printer, etc.)

Nonfunctional requirements: tools used to complete the basic functions within the application.

Sockets: endpoint of a two-way communication link between two programs running on the network (Oracle, 2018)

TCP connection: one of the main protocols in TCP/IP networks. Whereas the IP protocol deals only with packets, TCP enables two hosts to establish a connection and exchange streams of data. TCP guarantees delivery of data and also guarantees that packets will be delivered in the same order in which they were sent (Beal, 2018).

Appendix II: Team Details

The requirement document was developed by the following individuals, all of whom contributed the following:

xxxxx acted as the workflow leader for this section of the project. He divided up the sections to the other team members. xxx also scheduled the Writing Center meeting to ensure the document xxxx was grammatically and contextually correct. He acted as the main editor for the document, making slight changes to each section.

was responsible for the document's basic outline, along with the rough draft for the abstract, introduction, and project objective.

xxxx was responsible for creating the Table of Contents and Appendices, infusing a uniform format, collecting reference information and conducting final edits to the requirements document.

xxxx created the UML chart and completed the *Initial Business Model* Section. He also contributed to other various aspects of the requirement document including formatting, proofreading and grammar checking. Hard copy presentation was also completed by x.

Appendix III: Workflow Authentication

I, xxxxx, attest that I executed the functions listed within the team details section of the document. Also, I agree with all aforementioned information stated within the requirements document.

_____	_____	_____
Printed Name	Signature	Date

I, xxxx, attest that I executed the functions listed within the team details section of the document. Also, I agree with all aforementioned information stated within the requirements document.

_____	_____	_____
Printed Name	Signature	Date

I, xxxxx, attest that I executed the functions listed within the team details section of the document. Also, I agree with all aforementioned information stated within the requirements document.

_____	_____	_____
Printed Name	Signature	Date

I, xxxxxx attest that I executed the functions listed within the team details section of the document. Also, I agree with all aforementioned information stated within the requirements document.

_____	_____	_____
Printed Name	Signature	Date

Appendix IV: Report from the Writing Center

Cal U Writing Center Report

Client: xxxx
Staff or Resource: Brittany K.
Date: October 22, 2018, 4:00pm - 5:00pm

Did the student request that the instructor receive a visit report?: Yes

What course was serviced by this visit?: CSC 490

What goals were established for this tutoring session?: Editing;

How did the process of this consulting session address the established goals?: The client visited with his group to work on their group project. We compared the paper to the guidelines posted on D2L to make sure these guidelines were adequately met. I read through the sections that included paragraphs and full sentences to check for grammar and style issues. Overall, aside from a few issues with the paper's structure and grammar, the group seemed to have a strong grasp on the assignment.

Please provide any additional comments relevant to this session.: