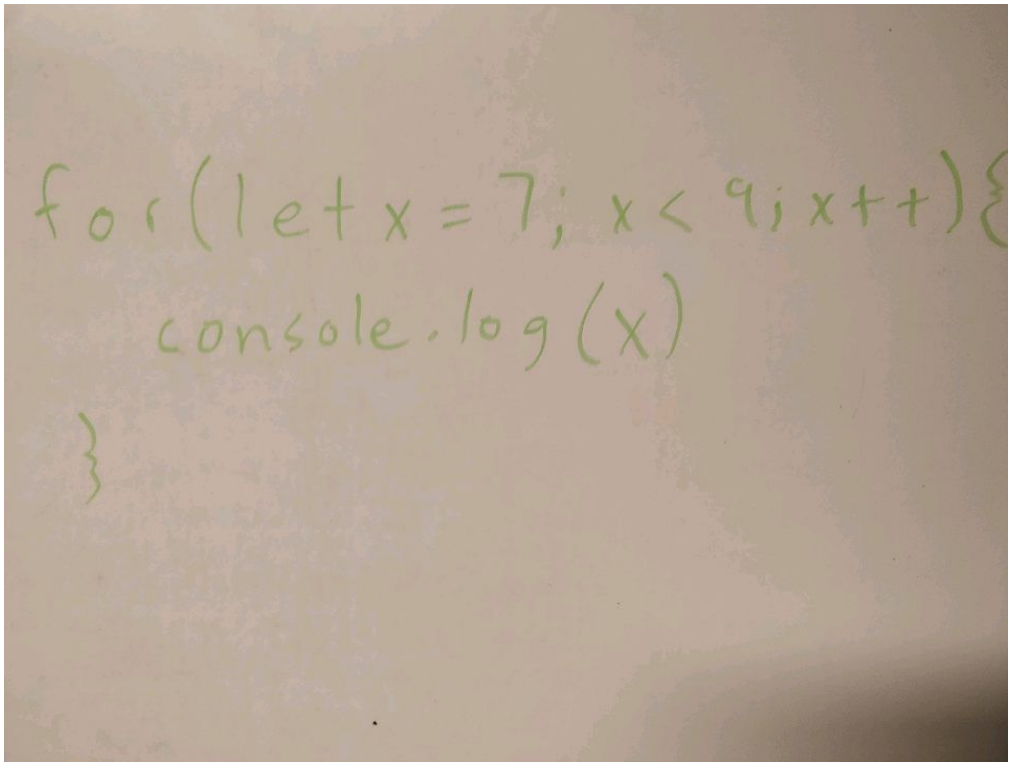
Black Panthas - Eric Higgins, Zak Kulphongpatana, Josh High

Deployment Plan

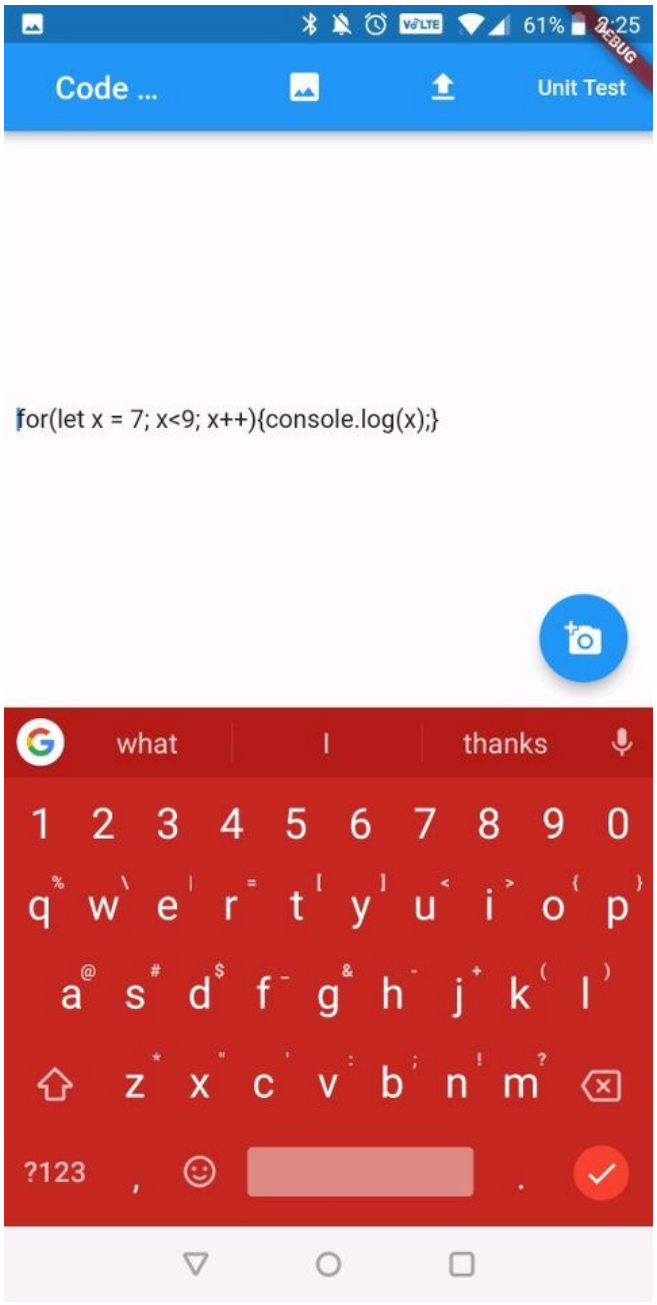
Product Description:

Our product is an application that allows for handwritten code to be edited and executed from a mobile device. This entails that a user is able to take a picture of code or upload an image of code that is handwritten on any surface and they would be able to run that code on their mobile device or debug their code. In the current iteration of the product, the only programming language that can be executed is Javascript. An example of an image that a user may try to run from their mobile device is below:



This would execute a for loop that prints out the values “7” and “8”. The user’s mobile device acts like a terminal and displays that print statements of the code. Currently, the image is passed to a server from the mobile app and Object Character Recognition is performed on the image to detect characters and identify which characters are within the image. Because Object Character Recognition is inaccurate with great enough frequency to have concern, we pass the results of Object Character Recognition back to the user to correct, as shown below:

Black Panthas - Eric Higgins, Zak Kulphongpatana, Josh High



After the code is corrected than the user can upload the code to a server where it is executed. This returns the outputs of print statements and displays it to a seperate page as shown below:

Future Expansion of Product:

Within the future there are a number of features that we believe are necessary to be implemented before bringing the product to market.

Firstly, we need to significantly improve the Object Character Recognition algorithm to be usable in all lighting situations. The current algorithm is very finicky when it comes to lighting and shadows. Additionally, the context detection component of the algorithm needs to be significantly improved as well.

Secondly, another feature that is imperative would be the capacity to execute multiple programming languages. While Javascript is useful, many other languages would be desirable in the final product.

Thirdly, we believe that it is necessary to build out the app as a platform where precreated problems can be downloaded to the application and allow the user to test their coding skills by answering such problems.

Fourthly, because we were so focused on the backend components of the project. The front end leaves much to be desired and there are a number of bugs that need to be fixed on the front end.

Market Opportunity:

We believe that there are two primary market opportunities.

The first is directed towards companies that hire software developers and other coding professionals. Our application could be used by companies in the process of coding interviewers as they make hires. Essentially, within an interview, while the interviewee is working through a particular coding problem and writing their solution on either paper or a whiteboard, the interviewer could test their code solution with their mobile device, while adjusting for minor syntax errors, in order to evaluate that individual's coding prowess. It would be integral for us to implement the feature to test multiple languages as many interviews are language agonist. Additionally, the feature that allows for a created question bank would be useful in further streamlining the interview process. We believe that large corporations that recruit a high volume of software developers would find our product useful to streamlining the process and allowing for them to compare results between applications as well.

The second market opportunity, which is our original inspiration, relates to its application in education. We were inspired by the boardworks that we completed in Programming 1 and Programming II and found it interesting the potential to be able to actually test the code we had written on the whiteboard immediately. This would have been very useful while working through practice problems, especially if a teaching assistant or professor is not immediately available to review the code. With this application, we would partner with universities and professors to implement our application into their classes as a way to improve students' learning experiences. Additionally, this could have strong

application in K-12 education. Considering the same use of having students learning to code on a whiteboard, for K-12 schools it is very difficult to download software needed to code onto computers due to district regulations and firewalls, as a result a mobile application that allows students to run their code would be of great benefit.

A secondary market opportunity is the general public as well, selling the application through the Android and iOS app stores. Because we developed the application through Flutter our app is deployable across Android and Apple devices. As a result, any individual could have access to our application through those public app stores if they have any device that has access to those app stores.

Cost of Deployment to Companies and Schools:

The cost of deploying to Companies and Schools has multiple components. These costs include marketing and travel to conferences.

Marketing the product to companies would have associated costs related to material and hiring people to promote the product. In total for a year, this would likely cost approximately $100,000 when including costs for staff in addition to travel costs, including travel to conferences and associated configuring costs. A breakdown for this would be $80,000 to hire two employees for a year to travel to companies and schools. The remaining $20,000 would be used for travel costs as well as travel to conferences for the year.

Cost of Deployment on App Store:

The cost of deploying the app to both the Android and iOS through the Google Play Store and Apple App Store, respectively, is approximately $250 over 2 years according to https://www.quora.com/How-much-does-it-cost-to-publish-a-mobile-app. The cost to deploy an iOS app is $99 per year, while deploying onto Android is a $25 one time fee to deploy.

Open Source Data:

A large component of our application included the collection of handwritten samples from students at the University of Kansas through working with Professor Dr. John Gibbons and Professor Dr. David Johnson. Because there was a limited amount of

public handwritten training samples for ASCII characters outside of alphanumeric characters, we went to the classes of the aforementioned professors and from their Programming I and Digital Logic classes sourced all of the non-alphanumeric characters we were missing. We have anonymized all of our samples and they can be found in the repository here: https://github.com/ehiggins98/HASCII.