**CS Outreach Application**

Kyle Blauer

Tristan Currens

Justin McClain

Ajeet Yadav

CS5363 – Software Project Management

September 15th 2015

**Project Statement and Justification**

The project being proposed is a CS Outreach App in hopes of helping students with programming. The idea will be more like interactive CS-101 tutorial. This application will teach interactive programming to prospective CS students.

The justification for this application is to improve the learning scale for programming languages. If a student feels the need to learn a language they do not know this app will be easy access and allow them to interactively work out hands out problems programming on their mobile devices.

**Project Background**

Interactive CS outreach applications are nothing new, in fact, there are plenty of them out there. We hope to take good ideas from the applications that have come before ours in order to offer the best experience to our consumers.

*Code School (*[*https://www.codeschool.com*](https://www.codeschool.com)*)*

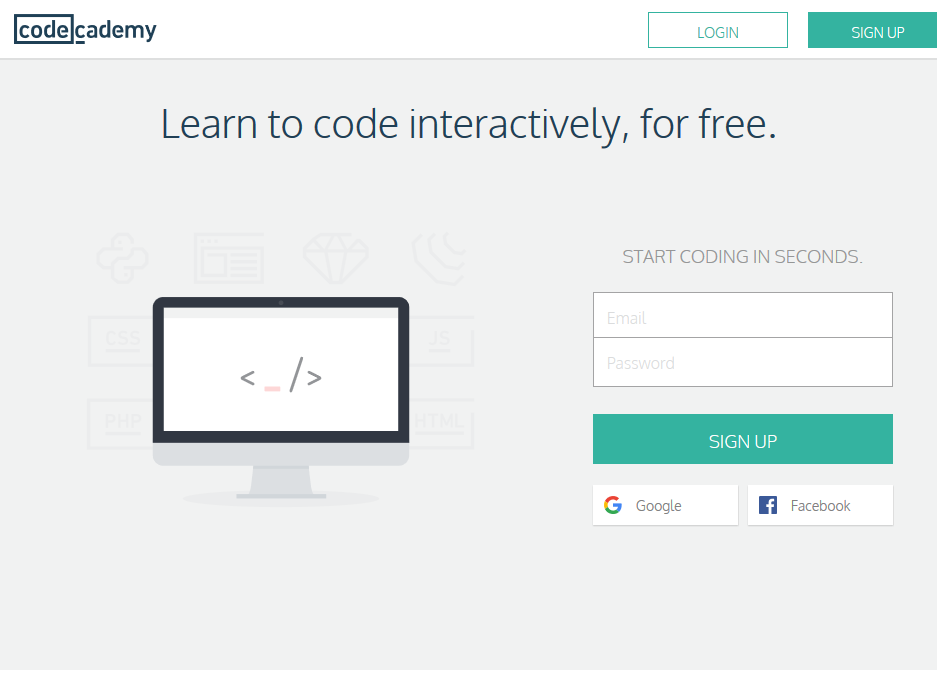
Code School is the de facto flagship of interactive coding webpages. The website offers “paths” that a user can take that are synonymous with learning curriculum. Some of their major paths include Ruby, JavaScript, HTML/CSS, iOS, Git and other similar technologies. For the sake of maintaining an acceptable scope in our application, I think we could implement a single “path” in our application that will take the user through some interactive courses that will teach them basic information that would be taught in a CS101 type class.



Code School uses a paid course model that makes money through selling access to these paths to its users. This model, while profitable, is obviously targeted at users who are already established programmers and not the average person off the street who wants to just try computer science.

*Codecademy (*[*https://www.codecademy.com*](https://www.codecademy.com)*)*

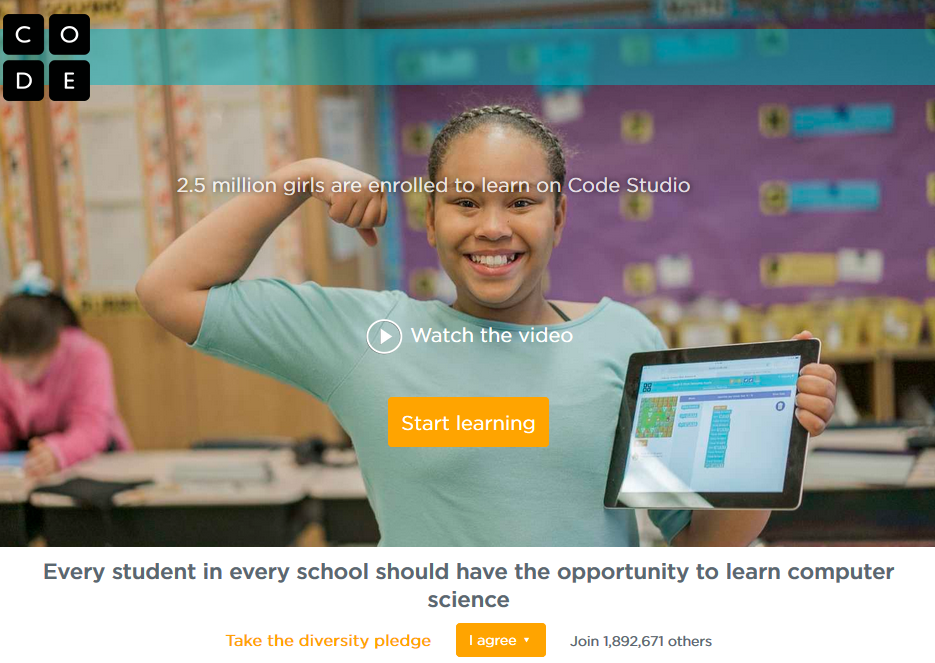
Codecademy, is less robust in terms of number of languages supported and offers a similar service for interactive programming. Like Code School, Codecademy gives the user a quick tutorial about Javascript they are trying to learn and then asks the user to “fill in the blanks” to teach them the basic programming examples. As the user progresses through the curriculum, the examples become harder and harder. Through this scaling difficulty curriculum, the user eventually learns the material, all of which is taught “by doing”.



Unlike Code School, users can learn to program on Codecademy absolutely free of charge. This is definitely the model that should be taken with an application like our CS outreach as we are trying to attract less experienced users who are new to STEM.

*Code.org (*[*https://www.code.org*](https://www.code.org)*)*

Code.org is a non-profit organization founded by two brothers, Hadi and Ali Partovi. Their goal was to bring Computer Science classes to K-12 programs across the United States. Although they don’t have an interactive application like we are proposing, they captivate the goals of our application. The ultimate goal in our application and in their organization is to get people involved in STEM and Computer Science.  
  
The organization has received backing from major figures in the technology industry, including Bill Gates of Microsoft and Facebook’s Mark Zuckerberg. The site, although not comparable to what we are proposing, allows users to “learn” to program by using a proprietary building block visual programming language. By chaining blocks together, the user creates a basic program flow that accomplishes small tasks. This type of learning is a no-barriers entry to programming that doesn’t offer a deep or immersive learning experience, but instead aims to show users that they can make something out of nothing.



**Preliminary Design, Implementation, and Evaluation**

3.1 Design of the Project

3.1.1. User Characteristics

3.1.1.1 Users

This is an android-application that requires no specialized knowledge or technical expertise of any kind. Primarily, a beginner, without any background and knowledge in computer science shall be able to use this application. The later part of tutorials will be building upon the initial part of the tutorials, which shall require the user to follow the tutorials sequentially.

3.1.1.2 User's Environment

This system shall be designed using the Client-Server architecture. The web application will be designed and built on the server side and it will be accessed by the user using internet connection.

3.1.2. Functional Requirements

3.1.2.1 User Interface

This section provides a description of the logical characteristics of each interface between the software product and its users.

a. User interface shall have a homepage.

b. User shall be able to register for the application.

c. User shall be able to login to the application after registration is completed.

d. User shall be able to update their own profile information.

e. User shall be able to stream/download the tutorial videos.

f. User shall be able to track the progress of their learning accomplishments.

3.1.2.2 Development platform

Android Studio shall be used for the application development.

3.1.2.3 Hard ware Requirements

Smartphone with high speed Wi-Fi/data connection. As the application will be data intensive, Wi-Fi connection will be heavily recommended.

3.1.2.4 Software Requirements

Primarily focused on the operating system Android Lollipop 5.1 (Market share – 20%) and Android Marshmallow 6.0 (Forthcoming release). Also, we would be looking at backward compatibility of the Android KitKat 4.4.2 (Current market share – 40%).

3.1.3. Non-Functional Requirements

3.1.4. Deliverables

3.1.3.1 Iteration 1

3.1.3.1.1 Proposal document

3.1.3.1.2 CS Tutorial 1.0

3.1.3.2 Iteration 2

3.1.3.2.1 Management Plan

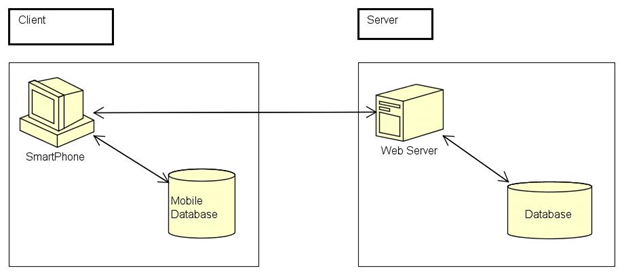
3.1.3.2.2 CS Tutorial 2.0

3.1.3.3 Iteration 3

3.1.3.3.1 Android application on the play store

3.1.3.3.2 CS Tutorial 3.0

3.2 Software Architecture



3.3 Deployment

3.4 Quality Assurance

Extensive debugging and testing will be done by using the Android Studio. Also, a testing framework (most probably robotium) will be used to automate the test cases.

3.5 Measures of Success

Various metrics will be used to measure the success of the application. Some of the metrics will be including but not limited to:

a. The android application is always available on the server.

b. The application should not be memory extensive.

c. The application should not drain the battery at an alarming rate.

**Risk Analysis**

As with any application development, there are numerous risks that we may encounter as we go.

*Communication* – Because our group is not all local to Lubbock, we are going to have to rely heavily on email and other forms of asynchronous communication. This is going to force us to need to constantly communicate with each other to make sure we deliver our application on time.   
  
*Scope* – With an application as open-ended as ours, it can be easy to lose track of what an acceptable scope is. We will need to keep in mind throughout our development that we are on a very strict schedule and won’t have time to put all of the bells and whistles that we may want in the application. We will need to be diligent about assessing what items are suitable for a minimum viable product and not over promise the functionality we can deliver.

*Skill Gap* – Android applications are structured differently than standard object oriented applications and will require a good amount of learning from all members of the team. Especially with the product that we want to deliver, it is going to be important that we learn the material quickly in order to deliver on schedule.

*Presentation* – Similar to the communication risk, we will be hindered in our ability to present our progress to the class. Because a number of us are distance students, we will have limited question and answer time with the class. It is therefore important that we are thorough in preparing our slides and make sure we remain detailed throughout.

**Schedule**

The schedule for this project follows a university course, with the main milestone dates being presentations and the resulting submissions. Below there is a detailed Gantt chart showing the schedule and main tasks, however first here are the main milestone dates:

Project Proposal - 9/17/15

Round 1 Submission - 10/8/15

Round 2 Submission - 11/5/15

Round 3 Sumbission - 11/24/15

Gantt Chart -

<https://drive.google.com/file/d/0B4L5olVYzUKmUE42S1JXcnJFTzg/view?usp=sharing>

**Budget**

The budget for this project is fairly simple as it is for a university course. However, we will still account for server costs, rent, and personal needs.

Employees : $4750/month

* 10 hours/week \* 4 Programmers \* $25/hour = $1000 \* 4 months == $4000
* Apartment for working - $550/month
* $200/month for computer maintenance

Server/database - Renting from Amazon : $427.70/month

* ec2 server = $229.98/month
  + M3.xlarge
* DNS Service: Amazon Route 53 = $0.50/month
* Amazon RDS (Relational Database Service) = $200.24/month
  + Db.m3.large
  + 500gb

Total Costs = $5177.70/month

**Team Expertise**

**Ajeet Yadav** is a Graduate Software Engineering student at Texas Tech University. He completed his B.S. Computer Science degree from Texas Tech University. Currently he works as a Graduate Assistant at the High Performance Computing Center (HPCC). His research interests include cyber-security, big-data, bio-informatics and software engineering. His fluent programming languages include Java, C++, PHP, and JavaScript. Also he has programming experiences with C#, C and Python. The IDE he has worked with are Visual Studio, XAMPP and NetBeans. Some of his recent projects are as follows:

1. Project Name: TTU Computer Science Outreach Website
   1. Position: Team Lead
   2. Date: 01/2015 – 05/2015
   3. Technologies used: PHP, Cascade Style Sheet (CSS), JavaScript (JS), Apache Server, MySQL Database.
   4. Comments: Built and designed a web site for the computer science department, directed towards K-12 students.
2. Project Name: Fantasy Sports Games
   1. Position: Software Developer
   2. Date: 01/2015 – 05/2015
   3. Technologies used: PHP, Cascade Style Sheet (CSS), JavaScript (JS), Apache Server,
   4. Comments: Built and designed a web application for the users to play March Madness game. Used Selenium tool for the test driven development.
3. Project Name: SMARTx
   1. Position: Research Assistant
   2. Date: 09/2014 – 12/2014
   3. Technologies used: C#, Microsoft SQL Database, NHibernate.
   4. Comments: Designed the database in Microsoft SQL, developed the web application in C# and combined them together. Maintained database which was used as a platform for the project.
4. Project Name: Hitori Puzzle Solver
   1. Position: Software Developer
   2. Date: 06/2014 – 07/2014
   3. Technologies used: Java, Clingo.
   4. Comments: Designed a puzzle solver written in two different languages and combining them together. The front end (GUI) was written in Java and the back end (logic) was written in Clingo, a language used in Artificial Intelligence.

**Justin McClain** is a Software Developer for GMIT (General Motors IT) at the GMIT Austin Innovation Center in Austin, TX and currently a graduate student for his Masters of Science in Software Engineering. He Graduated Texas Tech University in May of 2014 with his Bachelors of Science in Computer Science with a minor in Mathematics. He well rounded in graphic designs and is good with UI/UX development. He knows java, c++, c#, html and css/bootstrap. He’s capable of learning new languages of programming when needed. His job right now is getting him to learn AngularJS, Shell Scripting and SQL. He was involved in 3 major projects in his college career to date.

1. TTU Physical Plant Employee Training Website
   1. Team Project for Concepts of Database course in undergrad in Spring 2013
   2. Developed a website for the Physical Plant Employees that employees registered for scheduled training classes and for admins to edit what was needed when needed to.
   3. Technologies used: PHP, MySQL and Apache
2. Security Failure-Tolerant Online Banking System- •
   1. Team Project for Senior Project Design course in Spring of 2014
   2. Developed a prototype online banking website that verified activity via SMS and Email.
   3. Technologies used: ASP.NET, C#, MySQL, a little PHP used
3. Society of Women’s Engineering Prototype Outreach Website
   1. Team Project for Software Modeling Architecture course in Spring 2015
   2. Developed an outreach mobile friendly website that works with survey monkey and stores survey data from all of their outreach events throughout the world.
   3. Technologies used: HTML5, CSS Bootstrap, SQL, JQuery and C#.

**Kyle Blauer** is currently a student at Texas Tech University earning his Master of Science in Software Engineering. He earned his Bachelor of Science in Computer Science from Texas Tech University. He’s currently a Software Engineering Intern for International Business Machines Corporation and a Student Research Assistant and Web Developer for Texas Tech University DISCL.

His works involve the following:

1. Software Engineering Intern for International Business Machines Corporation
   1. Collaborated frequently with a team of engineers providing early programs teams a service to create new sites for marketing their new products, documentation, and demos.
   2. Designed, developed, tested, and deployed a new site for internal use allowing queries across multiple Trac ticket databases
   3. Altered an existing IBM WebSphere survey tool to allow for advanced conditional question management, enabling users to configure when a question should be required or not based on the state of the survey
   4. Configured Windows and Linux server stacks, both local and production, as well as developed server and client code using PHP, Java (WebSphere), IBM DB2, HTML, CSS, and JavaScript
2. Student Research Assistant & Web Developer for Texas Tech University DISCL
   1. Undergraduate research work mentored by Postdoctoral student
   2. Developed a simulation for testing I/O schedulers in a large scale supercomputing cluster using open-source tools such as ROSS and CODES
   3. Communicated with professors in order to provide websites for TTU use
   4. Designed, developed, tested, and deployed websites including an inclusive site for managing a conference run by DISCL
   5. Utilized several different open-source frameworks for providing a quick solution that remains easy to edit in browser
3. Software Engineering Intern for Ultra Electronics Advanced Tactical Systems
   1. Worked alongside a team of engineers developing a web based mapping application that provides distributed tactical operations awareness to authorized users
   2. Implemented a proximity alert system that provides notifications on entry and/or exit of an overlay boundary. System was implemented for both OpenLayers and Google Maps mapping engines
   3. Developed both server and client code using JavaScript, Python, PostgreSQL, HTML and CSS using a Linux based Apache Stack
   4. Contributed to meetings and presentations within the company and led individual code reviews
   5. Project utilized the Agile software development methodology using SCRUM

**Tristan Currens** is currently a student earning his Master of Science in Software Engineering at Texas Tech University and currently a Full Stack Developer on an internal site at AT&T. He’s mainly worked on android project on in school projects and he most of his experience is in JAVA application and API development. He also has a lot of web development and database administration experience.