## 

## The Programming Education System (TPES)

## Interactive Design Requirements Specification

Version: 0.1.0

Date: 01/25/2018

Presented by: Tide Pods

|  |  |
| --- | --- |
| Team Member 1’s Name | Christopher Lim |
| Team Member 2’s Name | Justin Liu |
| Team Member 3’s Name | Kyle Kaniecki |

# Revision History:

<Record all document updates by date and document revision number.>

## Remove any instructions and notes provided in the < >‘s

# System Concept

The programming education system (TPES) will teach middle school students the basics of programming. This service will offer comprehensive lessons and activities allowing students to acquire basic programming skills through an interactive medium as opposed to many current static-based lessons offered. TPES aims to peek interest in programming at a younger age and replace the non-interactive forms of self-education programming lessons, by presenting lessons in as easy-to-understand, immersive form factor that will allow any student to pick up and begin learning.

# Interview/observation Notes

<Document the interview and observation process - who, what questions, what notes.>

# Work Roles

<Identify the various work roles that you anticipate will use this product for the **primary stakeholders only** (no administrators or other secondary users unless the primary stakeholder’s tasks lack the breadth required in the subsequent sections). Work roles are characterized with profile information such as personal goals, frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, domain knowledge, abilities and disabilities, or skills and experience. Describe the pertinent characteristics of each work role, in its own table such as the one below… quantify whenever possible >

| **Work Role: Student** |  |
| --- | --- |
| Context of use |  |
| Goals | To use the product and obtain knowledge in programming. Since the student has not been exposed to programming before, they will have to put in practice hours to grasp the programming subject. |
| Frequency of use | A couple of times a week. Around 1 hour a day for a complete daily lesson. |
| Work responsibilities | To focus on the lesson and not get distracted. In order to learn effectively, the product requires the user to be committed to learning for the duration of the lesson. |
| Work environment | At home/school with a computer that has a keyboard and mouse that also has access to the internet. |
|  |  |
| Abilities | No abilities or skills in programming. Complete Beginner. |
| Personal | Grades 6-12. Any gender, any culture |

| **Work Role: Parent** |  |
| --- | --- |
| Context of use |  |
| Goals | To make sure that their child is learning how to code from the lessons they paid for |
| Frequency of use | A couple of times a week. Whenever they want to check up on their child’s progress. |
| Work responsibilities | To assess the students’ progress when learning to code and make sure that their child is not slacking. To also pay for the lessons because there is a price to use The Programming Education System. |
| Work environment | At home with a computer, keyboard and mouse with access to the internet. |
|  |  |
| Abilities | Beginner to Expert Knowledge in programming. |
| Personal | Any gender, Any age, Any Culture. |

| **Work Role: Teacher** |  |
| --- | --- |
| Context of use |  |
| Goals | To ensure that the student is learning properly. To assess student progress and provide any supplemental assistance if needed. |
| Frequency of use | A couple of times a week. Around 1 hour for every class they are teaching programming in. |
| Work responsibilities | To assess the students’ progress in learning, and grade them based on their assessment. Also, they will provide additional support for the students in case they need additional resources. |
| Work environment | In a classroom with multiple computers that have a keyboard and mouse that also have access to the internet. |
|  |  |
| Abilities | Expert knowledge in programming. |
| Personal | Has a programming license to teach, Any gender, Any age. |

# Work Flow Diagram

# 

# WAAD

<Synthesize work activity notes from the raw interview notes. Then create a work activity affinity diagram (WAAD) using the process discussed in class. Embed a (high resolution) photo of the completed WAAD and a short **synopsis** of the team’s reflection on the process experience>

# Interactive Design Requirements

<Specify requirements in **formal requirements statements**. Reference the relevant primary and as necessary secondary feature or category traceable to the WAAD ID. Document any rationale and notes that may be useful. A table format might be helpful to organize the requirements.>

# Usability Requirements

*<Specify a list of usability requirements for the product that will be important to either the customers or the developers. Write these to be specific, quantitative, and verifiable when possible. You will use the success criteria as part of your usability tests. Add any other related Software Quality Attributes that have usability implications.>*

# Design Modeling - Social Model Diagram

< Provide a brief overview of your system social model diagram. Embed or reference the diagram >

# Design Modeling - Task Analysis

< Provide the (HTA) task analysis (descriptive only) for **five non-trivial features** for the system. Organize the tasks by system features, the major services provided by the product....>

## Feature 1 <Don’t really say “Feature 1.” State the feature name in just a few words followed by HTA task analysis>

## Feature 2 (and 3 and so on each in their own sub-sections)

# Design Modeling - Usage Scenarios

< Write **usage scenarios**, as task interaction models, for **five** features supported by the system. Describe key usage situations happening over time>

## Usage Scenario 1

## Usage Scenario 2 (and 3 and so on each in their own sub-sections)