

# 3XA3 SRS

fayezs

October 2017

## 1 Project Drivers

### 1.1 Purpose

The purpose of this project is to re-implement the open source project K-Touch. K-Touch is a utility that allows users track their speed and accuracy in typing, and results in improved typing skills through practise and repetition. The re-implementation will improve upon the original project by making the it more user friendly and providing more comprehensive documentation.

### 1.2 Stakeholders

The stakeholders in this project are the clients, the developers, and the consumers. The clients are Dr. Bokhari and the TA's, as they are the ones who commissioned the project and it is their expectations that the developers are trying to meet. We, Abrar Attia, Mediha Munim, and Susan Fayez, are the developers of the project. The consumers or users for this project include any person who wishes to improve their typing skills.

### 1.3 Mandated Constraints

One of the main constraints for this project is the time frame in which it must be completed. The final prototype must be completed and demonstrated by November 27, 2017 and the final source code must be submitted by December 6, 2017.

Beyond time constraints, a major limiting factor for this project is that it must have similar functionality to the original K-Touch project. It must offer various text "lessons" for the user to type as quickly and as accurately as they can while offering real-time statistics on their accuracy and speed.

The functionality of the actual program will be constrained by the speed and processing power of the computer on which it is run. This is an issue that is present with the original project, as the program experiences heavy delays between the user input and the interface response.

## 1.4 Naming Conventions and Terminology

- **Lessons:** The selections of text provided by the application for the user to type as quickly and accurately as they can.
- **K-Touch:** The original project. A utility that is designed to help users improve their typing skills

## 1.5 Relevant Facts and Assumptions

The original project comes with very little user instruction and often experiences heavy delays between the user input and the interface's representation of the user's progress. It also has an unfair policy of requiring the user to type with 100% accuracy in order to advance progress in the lesson. These are issues that need to be addressed in the re-implementation.

The project will be implemented in such a way that it is assumed that the user has basic knowledge of computers, enough to open and run the program. Beyond that, the project will be user friendly and provide comprehensive instructions to ensure the user has an optimal experience. It will also be assumed that the program will be used by one user at a time on a desktop or laptop computer with a physical keyboard.

# 2 Project Issues

## 2.1 Open Issues

In the re-implementation of this project, the source code will be converted from C/C++ to Java. This presents several issues, the most critical of which being the use of required libraries. These libraries may not be portable to Java, which could hinder the functionality of our implementation.

Another challenge this project faces is the lack of familiarity the developers have with the language the original project is implemented in. One group member has limited knowledge of C/C++, but it will still be problematic for the developers to understand the original source code enough to re-implement and improve upon it.

Similarly, the developers of this project have never worked on a project of this scope. It will be a challenge to ensure that each component of the application runs smoothly, accurately, and concurrently.

## 2.2 Off-the-Shelf Solutions

Many similar applications to K-Touch exist that are readily available to users. There are competitive online applications that allow users to race others to accurately complete a selection of text, such as TypeRacer, FreeTypingGame, Nitro Type, and Rapid Typing. There are also applications that are more similar to K-Touch, in that they focus singularly on the user and helping them improve their speed and accuracy, such as Typing Master, Key Hero, and 10FastFingers.

## **2.3 New Problems**

The use of this application can potentially cause new problems for the user. One such problem is a potential Repetitive Strain Injury such as Carpal Tunnel Syndrome if the application is used for a prolonged period of time in a non-ergonomic way.

## **2.4 Tasks**

The tasks for this project are essentially to complete the deliverables prescribed by Dr. Bokhari within the time frame he provides. The final code should be completed by November 27, 2017 for the final demonstration. Other deliverables include: the Problem Statement, Development Plan, Requirements Document, Proof of Concept Demonstration, Test Plan, Design Document, Revision 0 Demonstration, Peer Evaluation, Test Plan, Test Report, and User Guide.

## **2.5 Migration to the New Project**

## **2.6 Risks**

The biggest risk for this project is that the developers may be too unfamiliar with aspects of the original project to effectively re-implement it. Given the time constraints on the project, they must learn quickly in order to have a viable product at the end of the project term. Another risk is in the conversion from C/C++ to Java. It is possible the new project won't be able to have the same level of functionality as the original in the transition.

## **2.7 Costs**

The cost for this project will simply be the time and energy of the developers. They will invest their energy in coding, learning elements of C/C++, researching libraries, learning how to develop GUI's, potentially learning how to implement wrappers, writing documentation, and presenting their work.

## **2.8 User Documentation and Training**

The final application shall be very user-friendly and easy to use. Comprehensive instructions will be included to diminish any chance of uncertainty in using the product.

## **2.9 Ideas for Solutions**

For the issue of having incompatible required libraries, the first solution would be to search for libraries in Java that have the same functionality as the original C/C++ libraries. Should this solution fail and no viable Java libraries

are found, The next solution would be to implement to use of a wrapper for the original libraries to make them usable in Java.

For the issues of inexperience and unfamiliarity, the solution is for the developers to commit themselves to researching and learning about the areas in which they are unclear. Online resources and the help of TA's will be implemented when needed.