Table 1: Revision History

Date	**Developer(s)**	**Change**
Sept 27	Mediha Munim	Updated template with project/team details
Sept 28	Mediha Munim	Completed intro - Git worflow plan
Sept 28	Abrar Attia	Completed POC - Coding Style
Oct 14	Mediha Munim	Updated GanttProject link
Oct 15	Abrar, Mediha	Updated POC
Dec 6	Mediha Munim	Linked Google Java Style Guide, added project review

SE 3XA3: Development Plan
ReTouch

Team #7, ReTouchers
Abrar Attia - attiaa1
Susan Fayez - fayezs
Mediha Munim - munimm

December 6, 2017

KTouch is an open source program that was designed to improve the typing abilities of its users. Reimplementing the program as ReTouch will present several challenges, and this document provides an efficient plan to produce the software with ease.

1 Team Meeting Plan

The ReTouchers team has agreed upon a weekly meeting schedule to ensure the project stays on schedule and all milestones are completed on time. On top of meeting twice a week during the labs on Mondays and Thursdays, the team plans to meet from 11:30 a.m. - 12:20 p.m. on Wednesdays in a study room in Thode Library. If the allotted time is insufficient, the group will decide on another convenient time to meet.

The notes from each meeting should follow an organized format. They will consist of an *Agenda* section (describing what needs to be discussed in the meeting), a *Minutes* section, and end with a *Homework* section (which will detail what needs to be done).

2 Team Communication Plan

The team plans to communicate primarily through Facebook Messenger since it is easy to use and is checked regularly by all members of ReTouchers. Discussions on Facebook will be about general questions and for decision making. Git issues will also be used to further ensure proper communication by allowing another way for team members to clarify actions/items on Gitlab.

3 Team Member Roles

The team roles were assigned early on and are subject to change in the future. Currently, Abrar is the team leader, the front end developer, and the Git expert. Susan is the C++/C expert and one of the main back-end Java coders. Mediha is the second main back-end Java coder, and is also the Latex/documentation expert. Development of the actual implementation will therefore be split equally amongst team members, with each member focusing on their main area of expertise.

In meetings, the meeting chair will also be the scribe. This role will rotate between the team members every week.

4 Git Workflow Plan

The Git workflow plan that the ReTouchers team has decided to follow is the *Feature Branch Workflow*. This workflow takes advantage of branching - every time a team member wants to add a certain feature, they can create a *feature branch* and commit their updated files there. They can push their changes to the central repository without affecting the master branch. Then, a pull request needs to be filed so that the other team members can verify the changes and merge the feature branch with the master branch. Labels will be useful since they can be added to merge requests and issues, as well as when a bug has been discovered. It has also been agreed upon that milestones will be used to track the project's main deliverables.

5 Proof of Concept Demonstration Plan

The most significant risks that exist with developing the ReTouch application begin with the expected difficulties for the implementation. The original KTouch application is written in C/C++, and there may be some difficulties with transferring the code structure and functionality into Java. In addition, many of the required libraries in the original application may not be portable and therefore may not work with Java code. In order to overcome this risk, similar Java libraries that resemble and function like the C/C++ libraries may be found and used for the ReTouch application. If no libraries are found, a wrapper can be used around the C/C++ libraries to make them usable in Java. Another risk with developing the application is that implementing a full GUI may be difficult and time consuming. Additionally, testing all the functionalities of the application may also cause some trouble. To overcome these risks, tasks will be divided amongst the developers to make sure that testing and GUI implementation are being done successfully and on schedule. Furthermore, another prevalent risk is running code concurrently. The timer, for example, will need to run in parallel with the main program (keyboard input). In the proof of concept demonstration, we will demonstrate that we can run two concurrent components of the program by implementing a timer which will continue to

update as user input is verified. We will also demonstrate that we can use the C/C++ libraries sufficiently and that we can make the program portable across both Linux and Windows.

6 Technology

The technology that will be used for the implementation, testing, and documentation of the ReTouch project is as follows. The main programming language that will be used to implement the project is Java. The IDE's that will be used are Eclipse or Net Beans depending on the developer's preference. In order to test the application, JUnit will be used for standard unit testing. Lastly, the project documentation will be created in Latex and JavaDocs will be used to create the module interface specification.

7 Coding Style

The coding style that best suits the ReTouch project implementation follows the Google Java Style Guide. The source code to be written will follow all of the structure, syntax rules, naming, and alignment that is provided in the Google Java Guide.

8 Project Schedule

The Gantt Project, which illustrates the project schedule, can be found here or in the ProjectSchedule directory.

9 Project Review

The software development process for ReTouch was smooth and the number of issues that were encountered was minimal. An aspect of the project that went well was work distribution; the work was split up very evenly between team members, and all team members made valuable contributions to the project. Team meetings were also very effective and the format used in the meeting minutes allowed the team to work efficiently.

An aspect of the project that did not go as well as planned was following the Gantt project schedule and staying on top of the project. For example, a lot of the documents were completed last minute, and it was difficult to follow the plan due to unforeseen deadlines outside of the course. In a future project, it would be important to take external factors into account when creating the Gantt project. Furthermore, a lot of the document sections were dependent on other parts of the document, and there was a linear order that certain sections needed to be completed in (e.g., if section 4 of a document depended on section 1). It would be helpful to take that into account when splitting the work and setting

separate deadlines for each team member to ensure everyone has sufficient time to complete their sections.