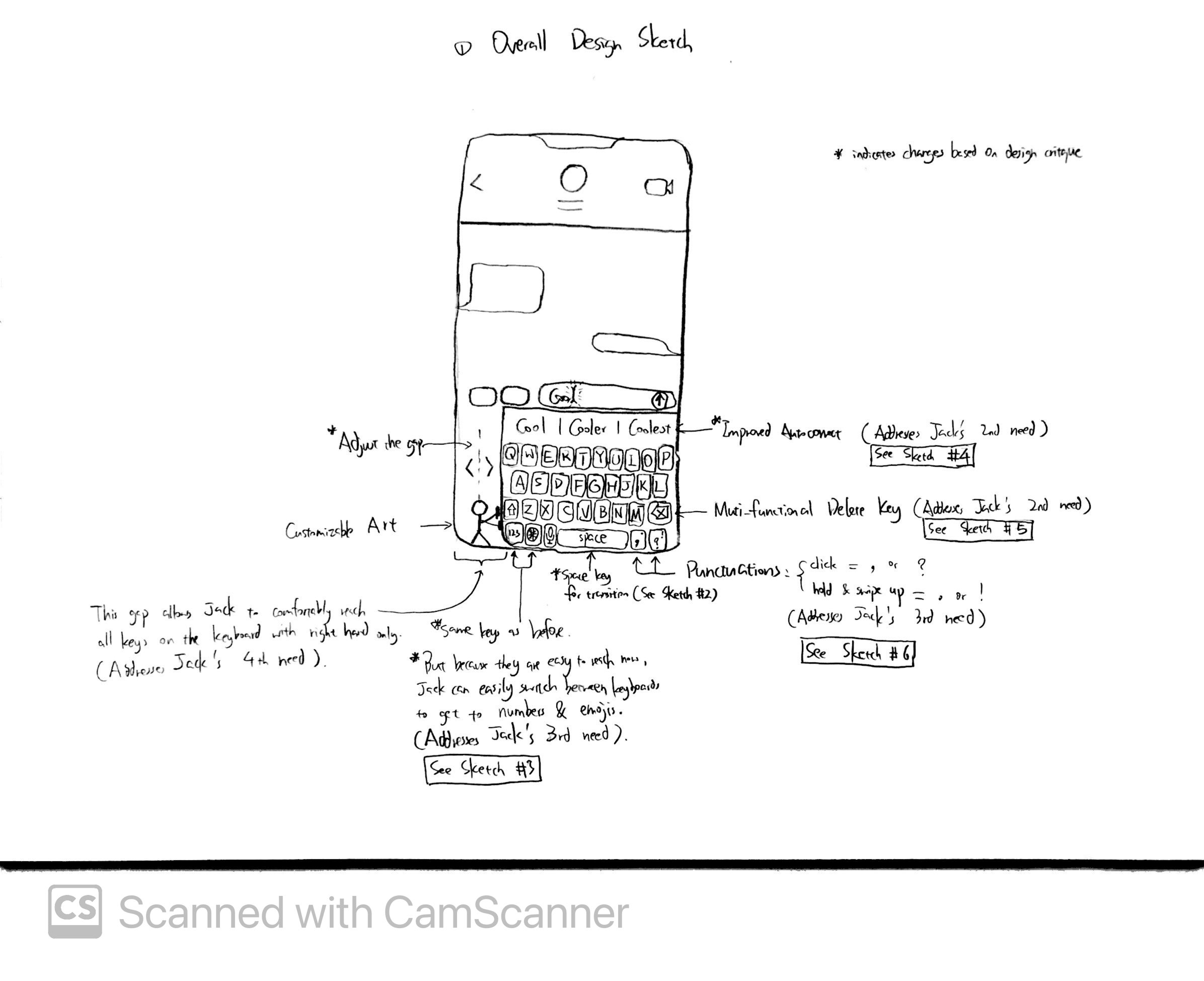
# Final Design

The final design (sketches & storyboards) were created based on the feedback received from design critique. Each team member actually had very different designs, meaning that our user requirements were well-written as they do not imply a specific solution. First, we tried our best to merge the different features of our designs that attempted to resolve the same need of the persona (i.e., addresses the same user requirement(s)), but if we struggled to do so, we collectively voted to keep the best one and discard the others. For example, some of Jerry’s, Yichen’s, Daniel’s, Franklin’s features all addressed the same need of being able to reach all elements on the keyboard (Jack’s 4th need, User Requirements 4, 6), and they were vastly different designs. After careful consideration, we decided to keep the one that is the most feasible (for the paper-prototype and wizard-of-oz) and adequately (in our opinion) resolves the persona’s needs. Next, for each of the features we decided to keep in the final design, we redrew/modified the sketches and also tried to improve on them based on the critique (see each of the following sketches to see the things we improved based on the design critique). Lastly, the storyboards were modified to illustrate the final primary persona and the context of use for our final design.

# Final Sketches

Sketch 1 presents the new one-handed keyboard as a whole, which attempts to address all of Jack’s needs. Sketch 2 shows the process of transitioning between the existing two-handed keyboard (current design) and the new one-handed keyboard (my design). Each of the remaining sketches (3-6) shows a specific part/feature of the overall keyboard, explaining how it attempts to address one (or more) of Jack’s needs.

## Sketch 1



This sketch illustrates the overall one-handed keyboard layout and general description of its features/keys. One major difference compared to the two-handed keyboard is the shift of the keyboard to the right side of the screen. By doing so, Jack should be able to easily reach all keys, including those that are on the left edge of the keyboard, thus accounting for Jack’s 4th need, where he would like to comfortably reach all elements of the keyboard while holding his phone securely. I did not make any significant changes to the layout of the letters because Jack is already so familiar with typing in the “QWERTY” layout, consequently, switching to a drastically different layout not only takes a long time to learn, but also may not even make any meaningful difference in meeting the user requirements. Thus, I just went with making gradual improvements to the existing layout instead of coming up with a breakthrough. One additional feature we added to this was the ability to adjust how much the keyboard is shifted or squished towards the right (see Sketch 3 for more details).

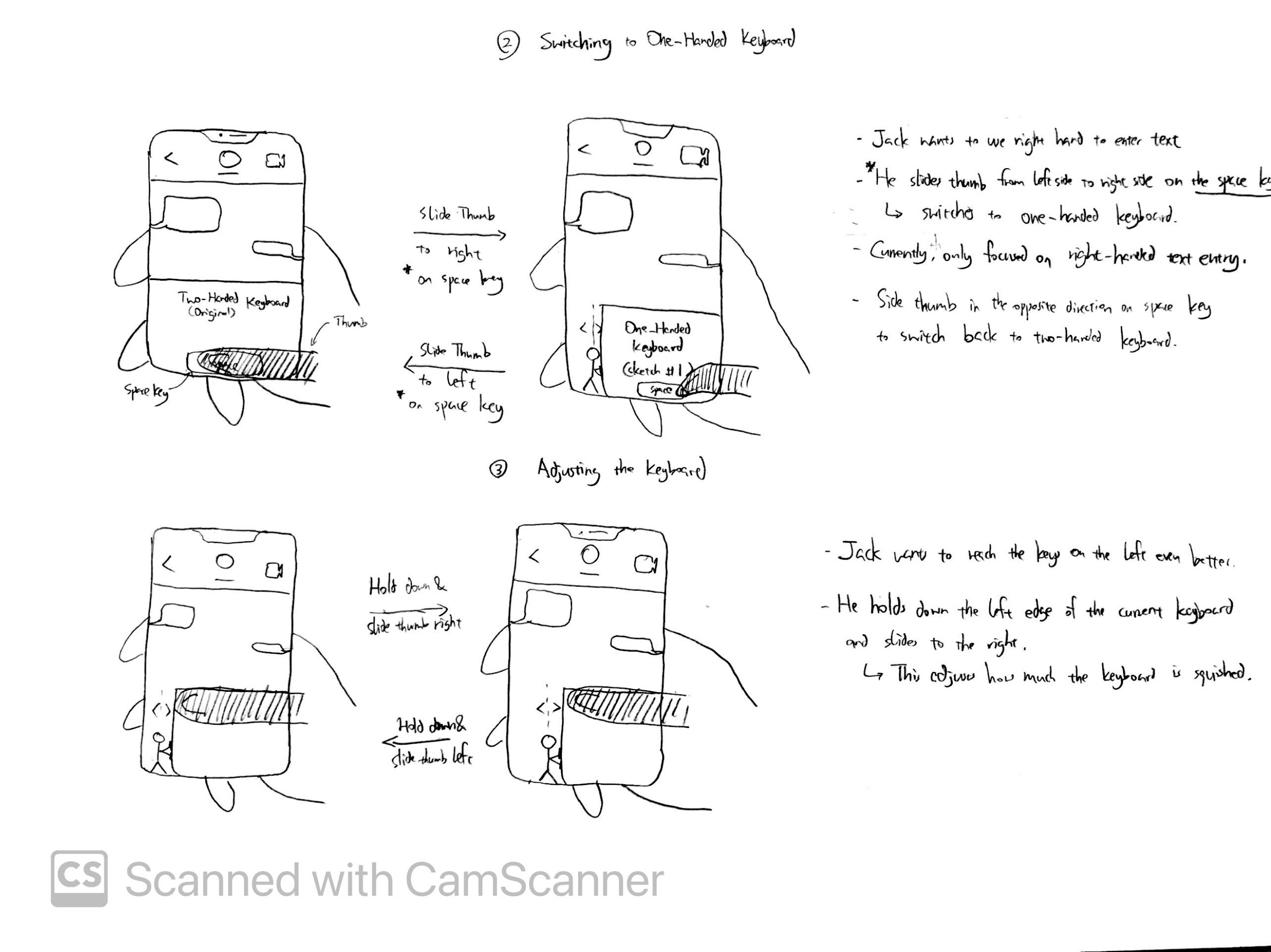
Another noticeable difference is that the “return” key at the bottom right corner is replaced with two separate keys. Before discussing what those keys are, I would like to elaborate on why the return key is removed. Jack is usually in informal situations when using his phone with one hand, such as texting, Google searching, or browsing social media. This means that the “return” key does not function as creating a “newline”; instead, it serves the purpose of “send” (sending his message, his search query, or his comment). However, these apps which Jack uses to text, search, or browse social media, already have a separate “send” key apart from the keyboard, so the “return” key in these situations are actually redundant. Therefore, it is removed and I believe that it does not cause any reduction in functionality. As for the two new keys, they attempt to address Jack’s 3rd need which is to conveniently enter certain characters; specifically, these two keys are for punctuations (see Sketch 6 for details).

Originally, this keyboard also had 2 additional special keys that attempted to address the remainder of Jack’s 3rd need, which was to easily enter *numbers* and *emojis*. However, based on the design critique, I realized that these keys not only would make the keyboard’s bottom row too squished, but they were also a little redundant since the fact that the keyboard being shifted towards the right already implicitly addressed the need to easily enter numbers and emojis, because Jack would now be able to easily reach the emoji/number-switch key on the left side of the keyboard, which was the main breakdown we identified from our contextual inquiry results. Thus, these two keys were removed in the final design.

Despite not having any salient differences in terms of their appearances in this sketch, the autocorrect feature (see Sketch 4) and delete key (see Sketch 5) both have improved functionality, in order to resolve Jack’s 2nd need, which is to easily fix any typos.

Lastly, although Jack’s 1st need is not explicitly addressed by any particular key/feature, I believe that when considering all of the features together as a whole, this new one-handed keyboard would be able to reduce the frequency of typos, thus resolving Jack’s 1st need.

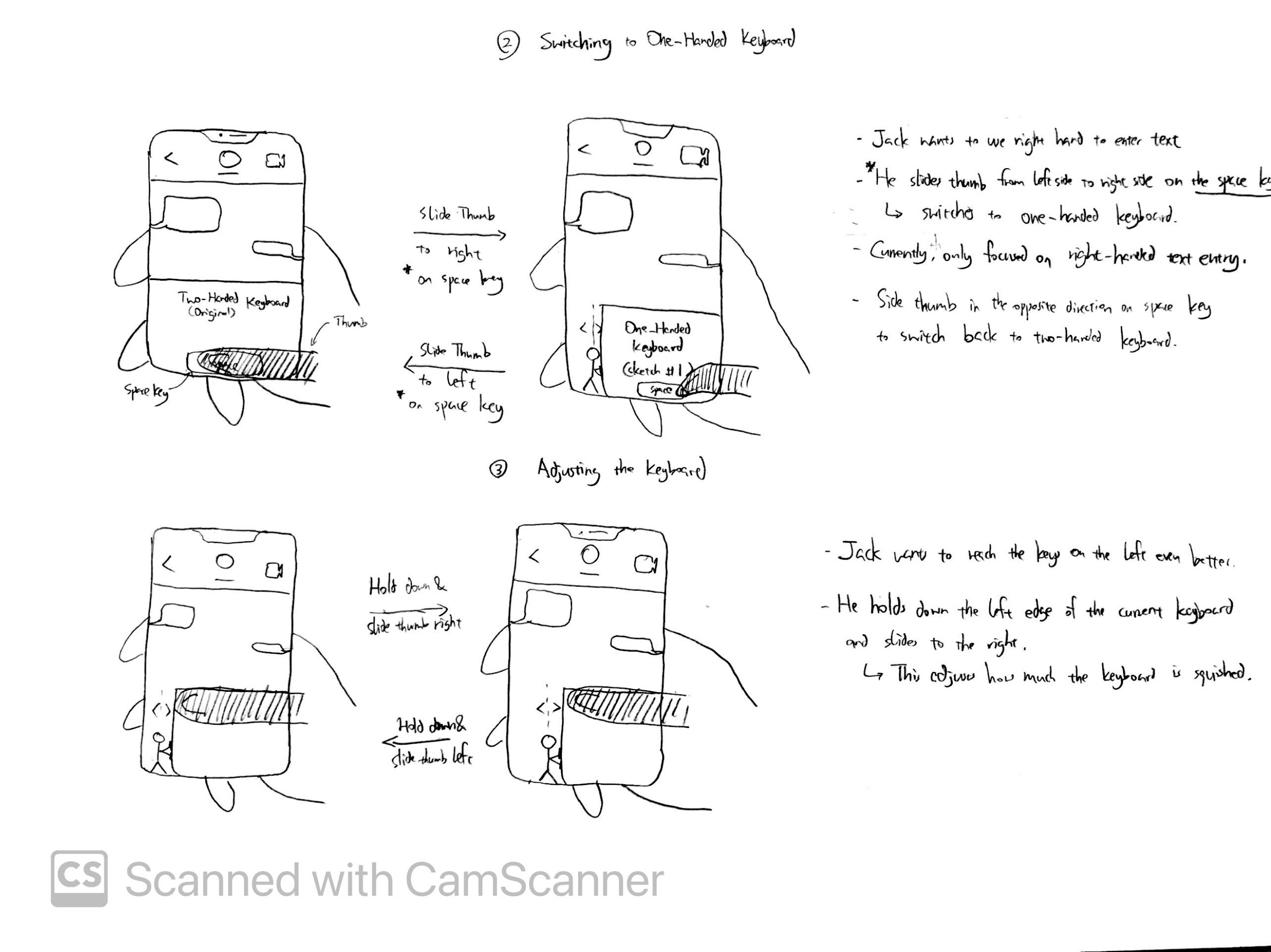
## Sketch 2



This sketch shows the process of switching from the existing two-handed keyboard to the new one-handed keyboard (my design), and vice versa. To do so, Jack simply needs to slide his (right) thumb from the left side on the keyboard’s space key to the right side (as shown in the sketch). To switch back, Jack simply needs to swipe in the reverse direction (from right to left). Note that, since Jack is right-handed, the one-handed keyboard and this transition motion are designed for people who type with the right hand (thumb). In the future, we plan to design for people who mainly type with their left hand as well.

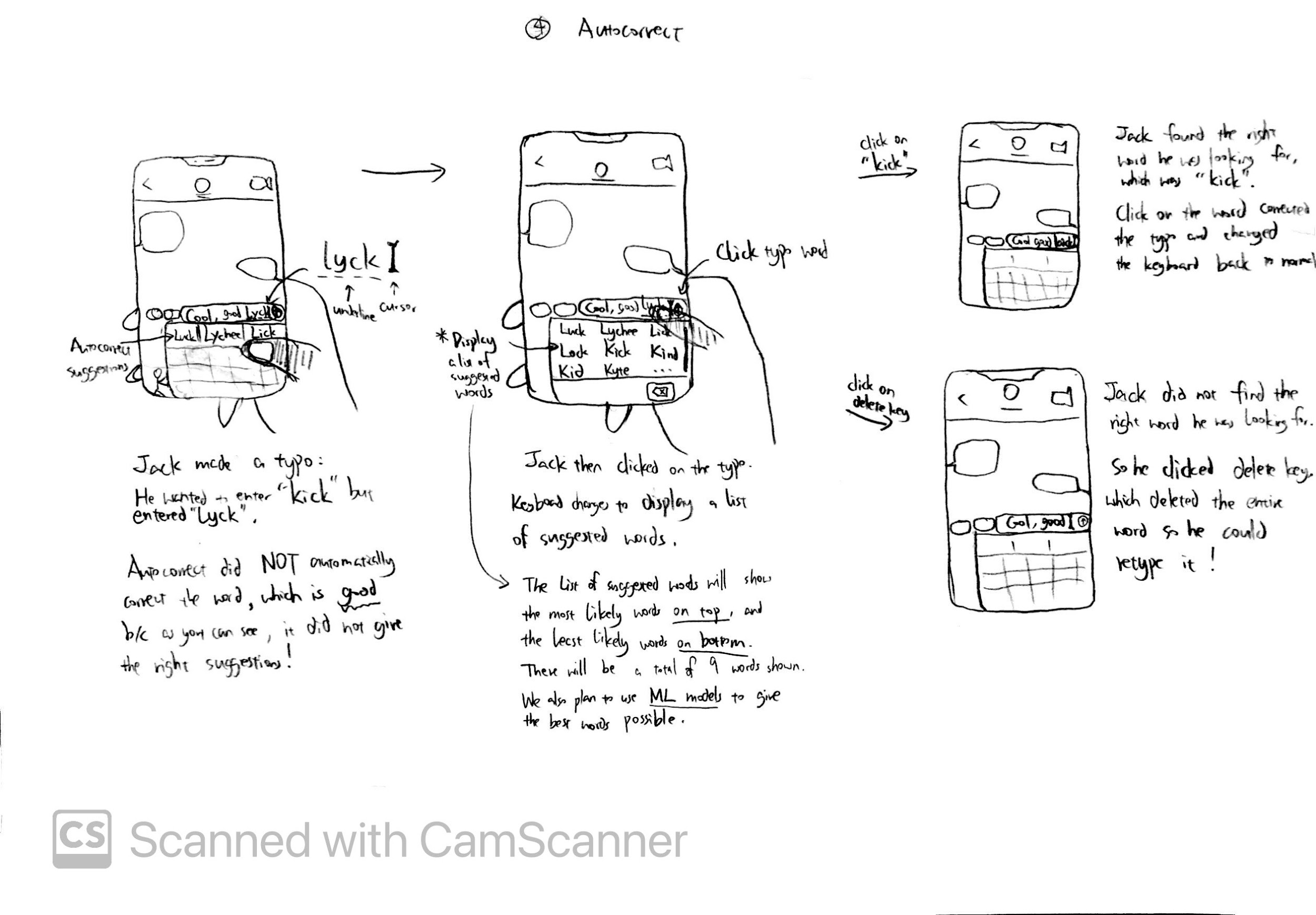
This feature was improved based on the design critique. Originally, this feature allowed the user to swipe anywhere on the keyboard to switch between two-handed keyboard and one-handed keyboard. However, during the design critique, we pointed out that although this allowed a convenient transition, swiping anywhere on the keyboard might interfere with other features on the keyboard and the screen/app, for instance, the swipe-to-text feature and the “back” feature (which are existing features on the iPhone). Thus, we decided to change the area which can pick up this interaction to just the space key, which does not have any existing features when the user swipes on it.

## Sketch 3



This was a newly added feature based on the design critique, which allows Jack to easily adjust the width of the new keyboard, or how much the keyboard is “squished”. This gives him the flexibility since Jack might be holding the phone with different hand positions from time to time, so the need of how well he needs to reach all keys on the keyboard may change. Thus, the addition of this feature addresses Jack’s 4th need more optimally.

## Sketch 4



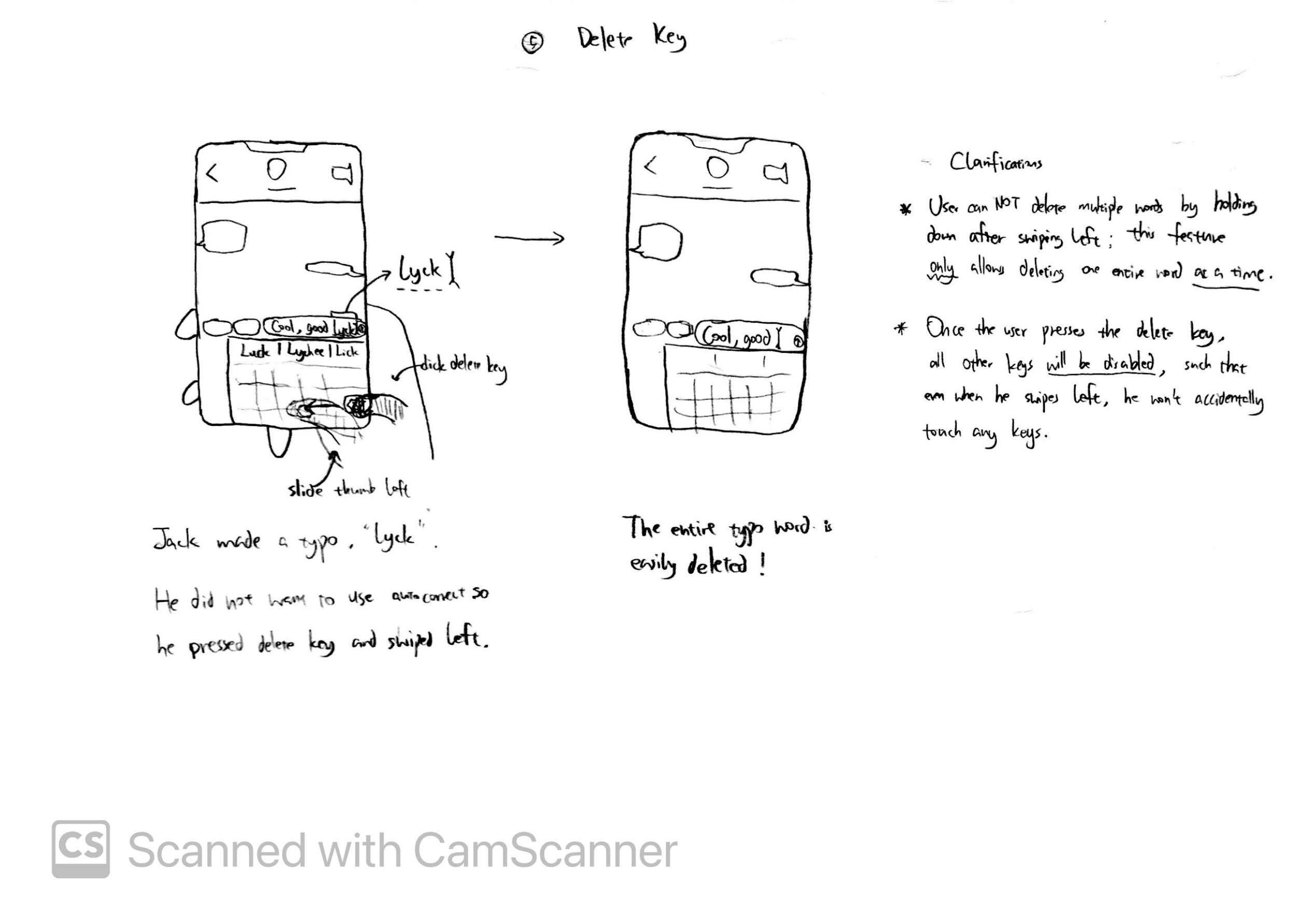
The autocorrect feature in the original two-handed keyboard can be counterproductive, specifically when it automatically changes a typo to a correctly-spelled word but is not what Jack intended, which happens frequently. In order for autocorrect to not automatically change the typo, Jack has to manually select one of the three suggested words displayed at the top of the keyboard, which he tends to forget especially when he is multitasking or distracted, resulting in autocorrect being counterproductive when Jack is entering text with one hand.

In our new one-handed keyboard, autocorrect no longer automatically changes the typo; it only underlines the typo and Jack can choose to manually change it or not. If Jack would like to change the typo, he simply clicks on the word and autocorrect will display a list of 9 words on the keyboard for him to select from; he can also scroll down the list of words to see more suggestions. If he finds the right word, he simply clicks on the word and Autocorrect will replace the typo with the word clicked. If cannot easily find the right word, he simply clicks on the “delete” key which deletes the entire typo, and he can then just retype it.

Based on the design critique, we also decided to incorporate Machine Learning models that learn from the patterns which Jack types to produce a better list of words that have a really high chance of giving the right corrections (merging with another feature by a different team member). We also changed the order in which the words appear to display the most likely words (the algorithm believed to be correct) at the top, and the less likely words at the bottom. The list would also dynamically update based on Machine Learning.

This improved functionality takes care of Jack’s 2nd need (specifically part a), because he no longer needs to fix any mistakes made by autocorrect, therefore, the autocorrect feature is not counterproductive.

## Sketch 5

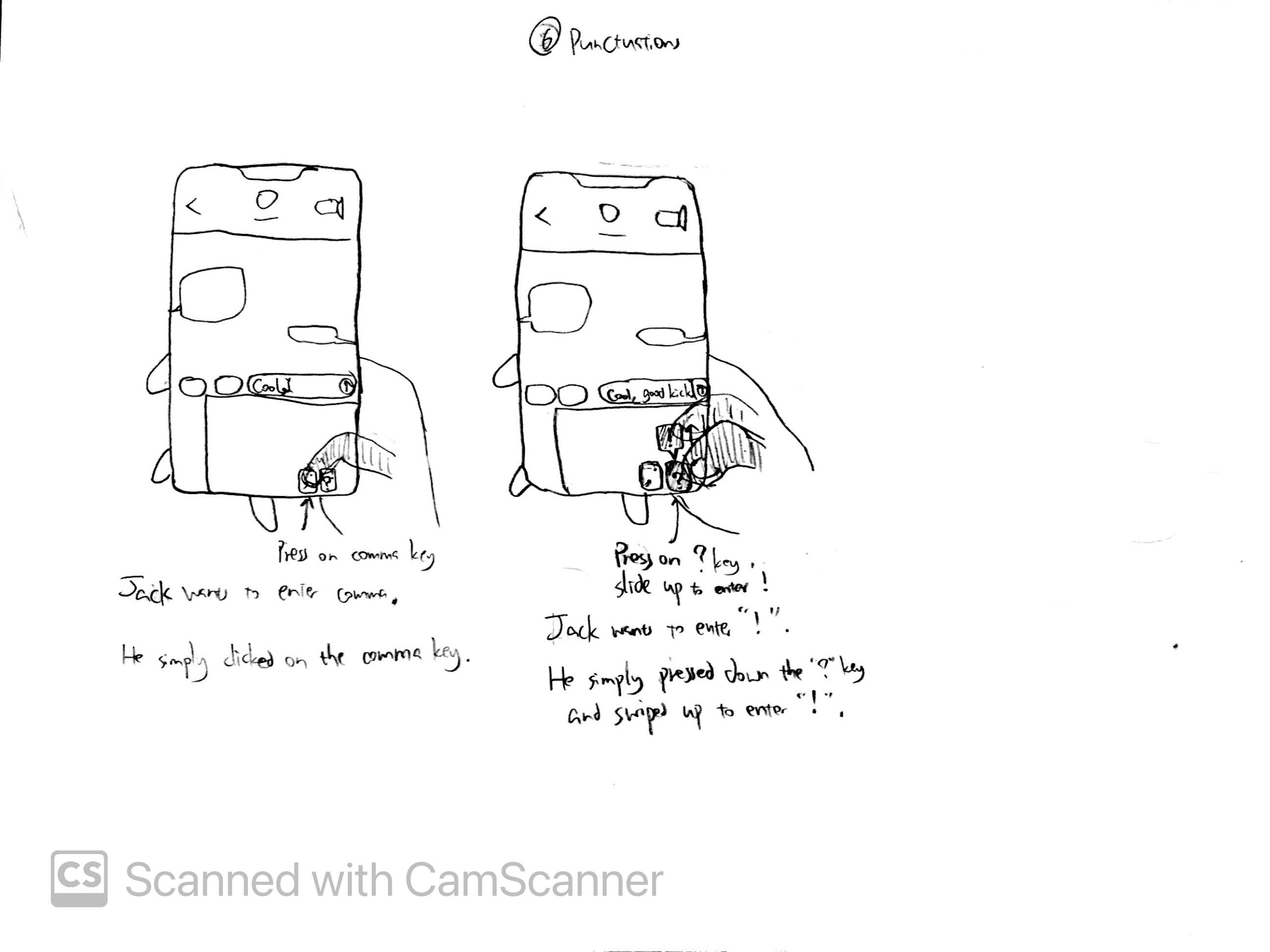


In the original two-handed keyboard, holding down the “delete” key will continuously delete one character at a time at a fast pace. This functionality is very difficult to control, as reflected in Jack often deleting not just the typo, but also the correctly-spelled word(s) in front of the typo. Thus, the delete key is originally counterproductive.

In the new one-handed keyboard, the “delete” key keeps the original functionalities, but also includes a new functionality: Pressing the delete key followed by sliding left will delete an entire word. This allows Jack to quickly and accurately delete the typo itself and nothing else, thus eliminating any counterproductivity caused by delete, resolving Jack’s 2nd need (part b).

This feature was also improved based on the design critique, which we realized that the swiping left interaction may result in the user accidentally clicking on the keys next to the delete key. Thus, to fix this, whenever the user clicks on the delete key, all of the remaining keys automatically are disabled (i.e., clicking them would not result in any text entry / functionality). A visual would also be shown that the keys cannot be clicked. As such, the delete key would accurately delete the words without causing more slips, better resolving Jack’s 2nd need.

## Sketch 6



In the original keyboard, whenever Jack has to enter a punctuation, including the frequently used ones like comma and question mark, he has to switch to the numbers/punctuations keyboard by pressing the keyboard-switch key at the bottom left, locate/enter the punctuation, switch back to the letters keyboard. He has to perform these three steps whenever he wants to enter any punctuation, which is complicated and difficult, especially when operating the phone with just one hand.

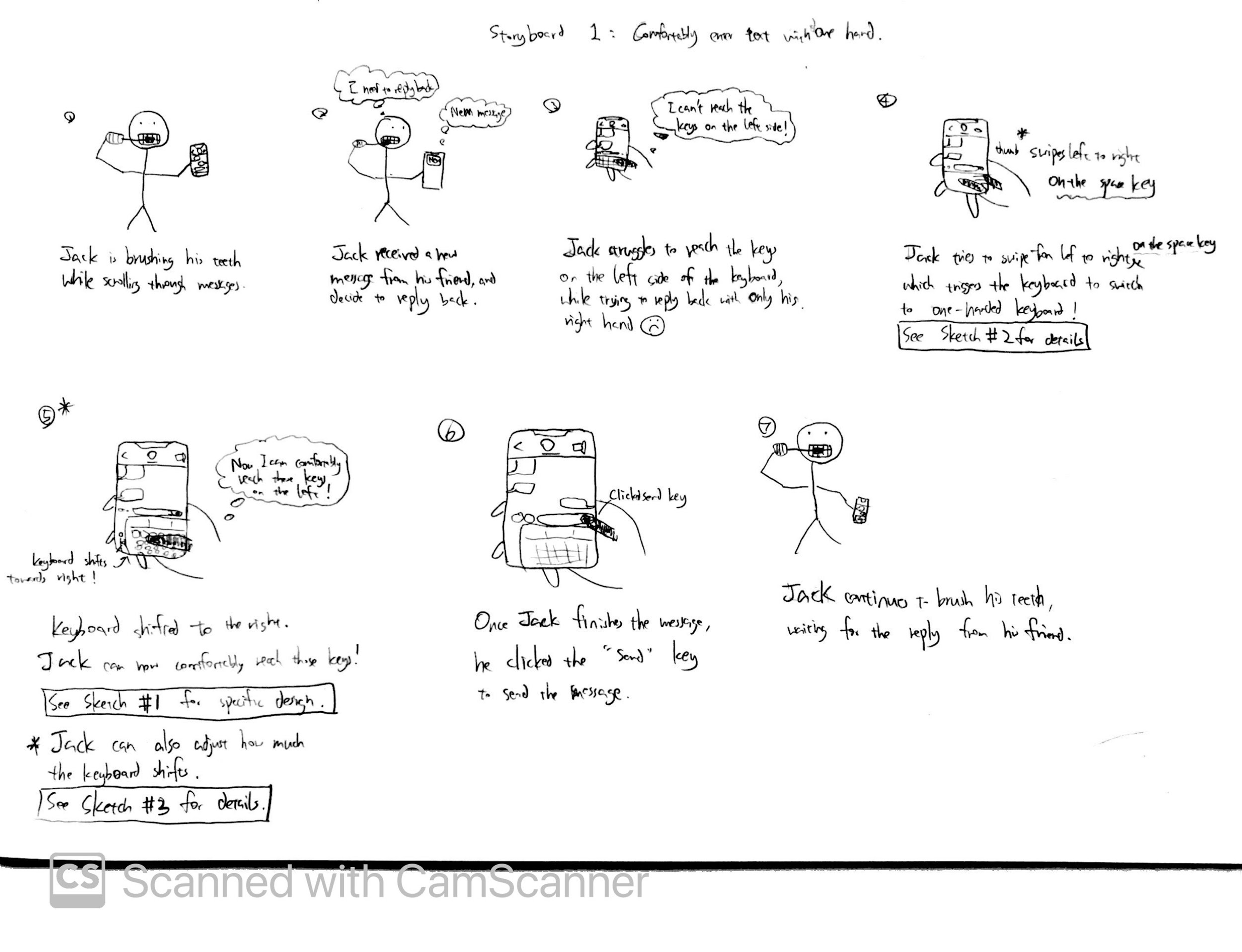
In the new one-handed keyboard, the punctuations key at the bottom right corner allows Jack to conveniently enter four frequently-used punctuation marks with just one step, which is pressing on the corresponding punctuation key, eliminating the excessive steps of keyboard switching. These two keys have default values of ‘,’ and ‘?’ respectively, which can be entered by a simple click. They also offer two alternative values of ‘.’ (period) and ‘!’ respectively, which can be entered by clicking and sliding up, which despite requiring one extra motion, is still much simpler than switching keyboards back and forth. Thus, this new addition of punctuation keys will account for part of Jack’s 3rd need, which is to enter non-alphabetical characters, in this case, the punctuations, conveniently.

Like mentioned previously under Sketch 1, this keyboard also had 2 additional special keys that attempted to address the remainder of Jack’s 3rd need, which was to easily enter numbers and emojis. But based on the design critique, we removed those two keys as they were redundant as the remainder of Jack’s 3rd need would be resolved by the shifted keyboard already (Sketch 1, 3). Thus, this was changed after the design critique for this feature.

# Final Storyboards

Storyboard 1 extends on Sketch 1, 2, 3. Storyboard 2 extends on Sketch 4, 5. Storyboard 3 extends on Sketch 6.

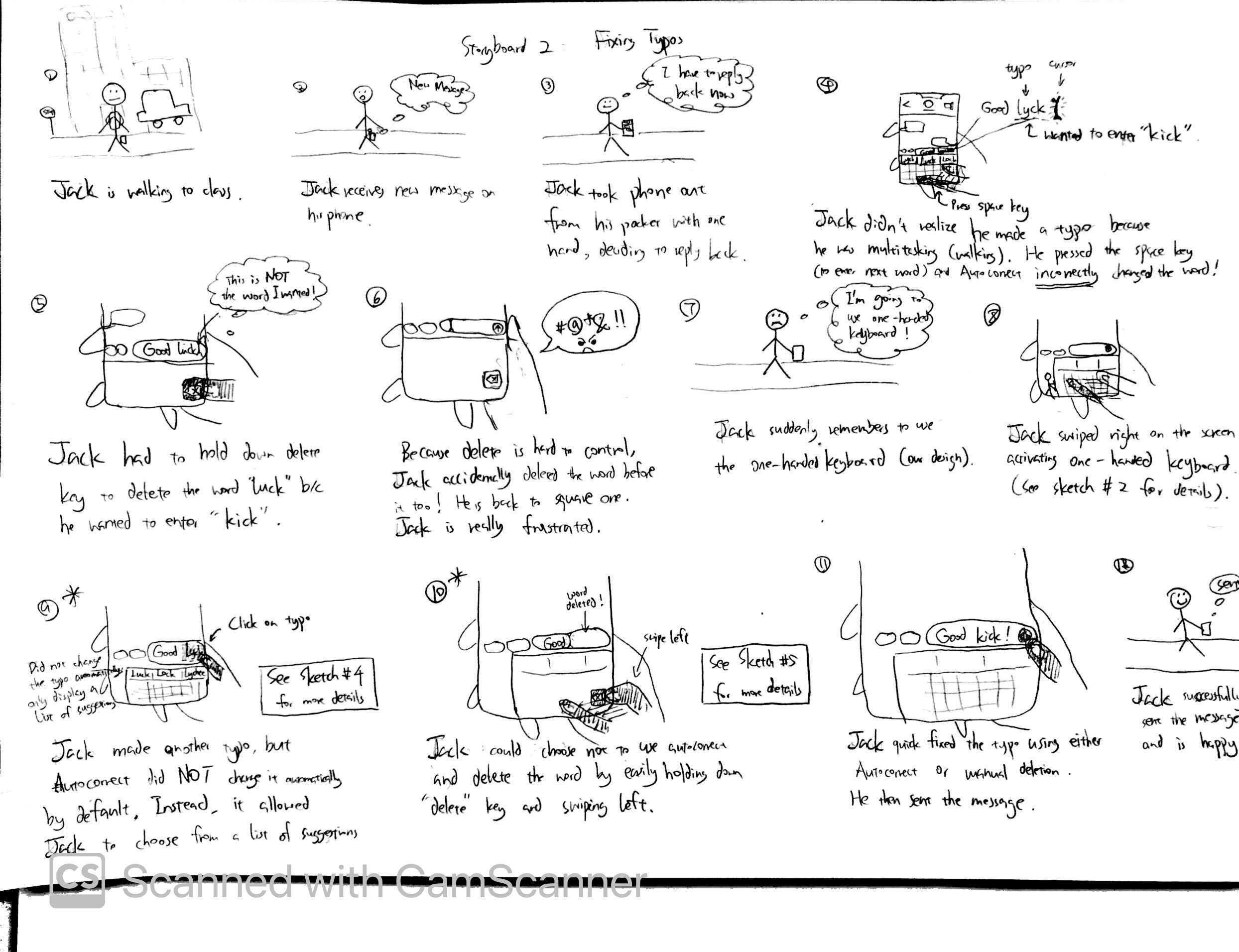
## Storyboard 1



This storyboard depicts how Jack, who is trying to reply to a message to his friend with one hand (specifically, his right hand) while brushing his teeth with his other hand, initially struggles with entering text on the current keyboard on his phone, because his thumb could not reach the keys on the left edge of the keyboard.

However, by changing (Sketch 2) to my new one-handed keyboard design (shown in Sketch 1 in detail), he is able to comfortably reach all keys on the keyboard without readjusting his hand position thus securely holding the phone. Consequently, he is able to effectively complete his message and send it to his friend. Thus, this successfully addresses Jack’s 4th need (and also part of Jack’s 3rd need).

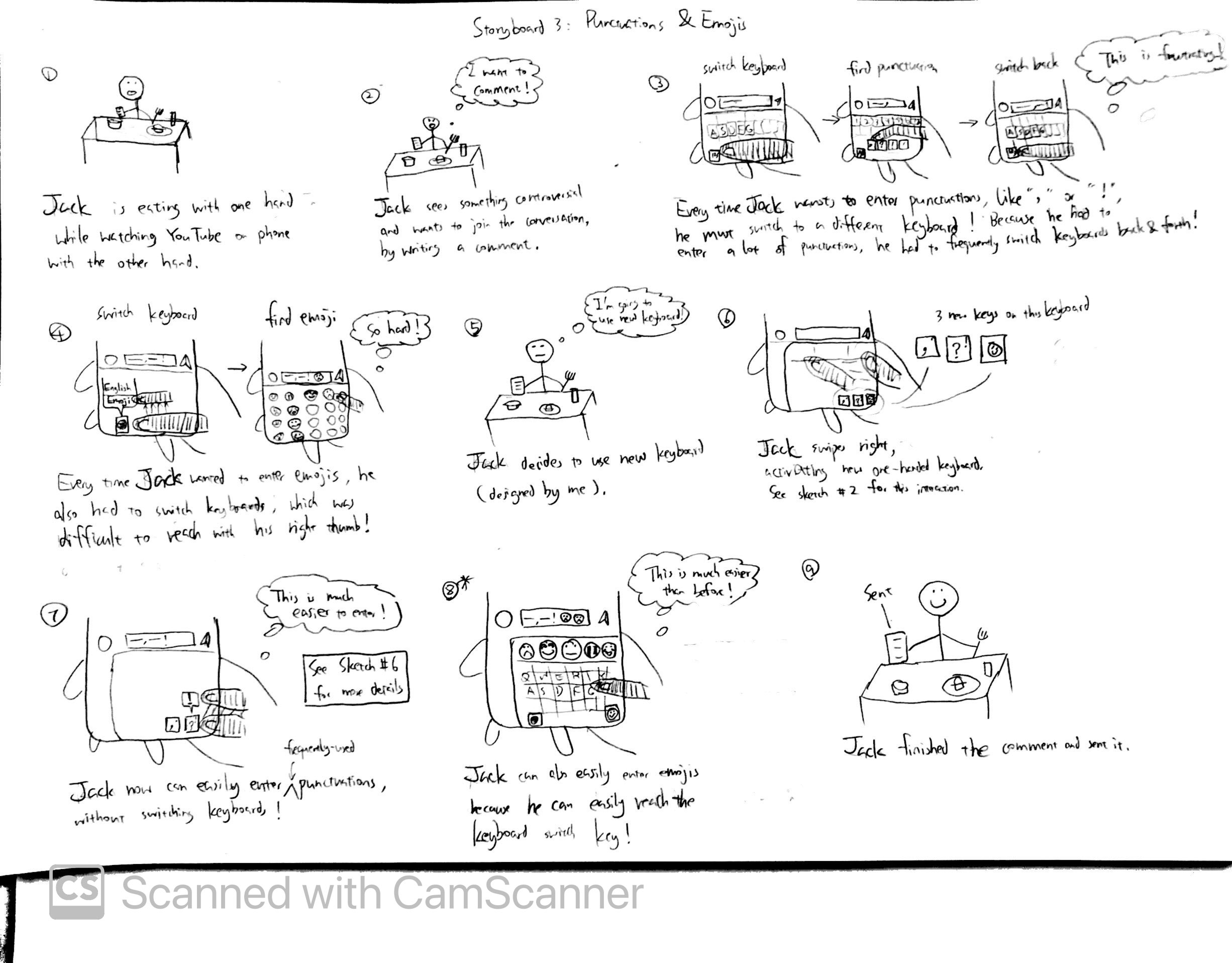
## Storyboard 2



This storyboard depicts how Jack, who is trying to reply to a message from his family member with one hand while walking to class, struggles to fix typos on the current keyboard design because the autocorrect incorrectly changes his typo to a word he did not intend to type, and while deleting this word, he accidentally over-deletes the previous correctly-spelled word too.

However, by changing to the new one-handed keyboard design, Jack is able to easily fix typos because the autocorrect and the delete key are no longer counterproductive, as the former allows Jack to manually pick from a listed of suggested word (Sketch 4), whereas the latter allows Jack to accurately and efficiently delete a word (Sketch 5). Consequently, he is able to efficiently finish the message and send it to his family member. Thus, this successfully addresses Jack’s 2nd need.

## Storyboard 3



This storyboard depicts how Jack, who is trying to comment on YouTube with one hand while eating food with the other hand, struggles to enter punctuations marks and emojis, because everytime he wants to enter one of them, he has to switch to a different keyboard, find the desired element, and switch back to the letters keyboard to enter the next word.

However, by changing to the new one-handed keyboard design, Jack is able to conveniently enter frequently-used punctuation marks, including comma, period, question mark, and exclamation mark (Sketch 6), and also easily enter desired emojis (and numbers) because he can now easily reach all elements on the keyboard. Consequently, he is able to efficiently finish the comment and publish it on YouTube. Thus, this successfully addresses part of Jack’s 3rd need.