APPLICATION BASED SMART OPTIMIZED KEYBOARD FOR MOBILE APPS

Sandeep S Machiraju, Ashok K Athukuri, Soumya Gampa, Narendra B Makela, & Venkata N Inukollu

College of Science and Engineering, University of Houston-Clear Lake, USA
Inukollu@uhcl.edu

ABSTRACT

Mobile applications are becoming an emerging property in today's world. Delivering high quality mobile applications by improving the quality of the user interface is a definitive guide in building successful mobile applications and will have a significant increase in the market for that application. This paper addresses some issues related to mobile user interface, current input system and user dictionary. After analysing these issues we have proposed a new system by making some improvements to the current input system and user dictionary. In order to improve the current system, we have taken some issues related to one of the most widely used data entry methods 'Keyboard' and proposed some improvements by introducing Smart Optimized Keyboard (STOKE). We have provided the results, which include performance, reduction in number of keystrokes, and screen space. This describes the significant change after adopting the proposed keyboard. In the end, we have proposed an optimized way of building up the user dictionary, which assists the user in inserting the input very easily, rapidly and effectively. In a combine, these two will help in providing the best user experience to the users.

KEYWORDS

Mobile App, Smart keyboard, keystroke, Predictive Text, Usability, Human Computer Interaction.

1. INTRODUCTION

Mobile applications have become an emerging property in today's world. Mobile application commonly known as a mobile app, is a software application that is designed and developed for wireless devices such as tablets and smart phones. Mobile applications are served for easy access to users compared to personal computers. Now-a-days mobile applications are vastly in use for various activities, including shopping, gaming, texting and many more. Users are capable of using the mobile applications for personal access, business needs and for day-to-day life tasks. For example, Google map has greatly simplified the navigation with GPS and is extensively applied to identify different places and also assists in knowing about traffic jams among various other services, Google notes, aids the users to save their personal information and banking apps help users to execute their banking needs including money transfers, balance inquiries, among many others.

David C. Wyld et al. (Eds) : CSITA, ISPR, ARIN, DMAP, CCSIT, AISC, SIPP, PDCTA, SOEN - 2017
pp. 175– 186, 2017. © CS & IT-CSCP 2017 DOI : 10.5121/csit.2017.70117

Many research papers have been published with respect to user interface and user interface design principles. The primary factor for any mobile app is: usability: described by: the effective functioning of the app, the purpose desired by the user and served by the app, the easy access to the users the user thoughts and feelings about the application [1]. Developing an application in accordance with the user demands and requirements along with the ease to use the app increases the lifespan of an application [2] [3]. In this research effort we discuss about the user interface, problems that a user faces while interacting with the mobiles and how a user can interact efficiently with the introduction of small improvements to the soft keyboard and Optimization of the User Dictionary. The remainder of this paper is organized as follows: Section 2 and 3 discuss about the mobile application user interface and third party smart keyboards in detail, Section 4 presents the proposed methodology in the current mobile framework and discusses the impact on usability after adopting the proposed methodology.

2. MOBILE APP USER INTERFACE AND SMART KEYBOARDS

Businesses for the mobile applications has increased rapidly and significantly all around the globe. In 2012, a study was conducted examining the growth of the market for mobile applications, according to which, by 2016, the market for mobile applications in the United States will reach $55 billion [4, 22]. Mobile user interface is one of the major aspects that needs to be considered while developing smart mobile applications and this will allow customers to decide and act immediately in their moments of need [4, 22].

The two important components that contribute significantly to mobile interfaces are Navigating and Data Entry [5] [6]. As the mobile devices are constrained in screen space, the application developers have to display the required information and need to receive the required information from the user as input by using the limited space. The system needs to provide an interface that should be clear, consistent, simple, user controlled, familiar, responsive and attractive and should be capable of efficiently taking an input from the user and providing the desired output. Such an efficient interface will enable the user to enter the input very easily, rapidly and effectively.

The widely used data entry method in mobiles is the virtual keyboard, because, it provides flexibility to the users, i.e., the Virtual Keyboard provides different modes to different text fields. In that respect are several alternative third party virtual keyboards, which are available in the marketplace. Each of the third party keyboards provides substantial benefits to the users. Despite the benefits, there are many issues that the users are currently confronting, which include: performance issues, security issues and usability issues. When the third party keyboards grant network access, there is a possibility of leakage of keystrokes and personal data of the users [7] leading to security issues. The performance issues of the third party keyboards include the unresponsive keyboard after it has been loaded [8], the unavailability and inaccessibility of the keyboard for a few seconds, etc. With respect to the usability of the third party keyboards, the users might not prefer the third party keyboards, as the users have an option to use the default keyboard that is readily available and is very good in performance, security and satisfies the basic requirements of the user. Therefore, the default keyboard provided by the system is the principal choice of the several users.

Several users prefer the default keyboard despite some drawbacks associated with the default keyboard. However, the implementation of the suggested significant improvements (discussed further in this paper) to the default keyboard from the perspective of the system and the

developer, will enable more and more users to reliably and jubilantly employ the default standard keyboard.

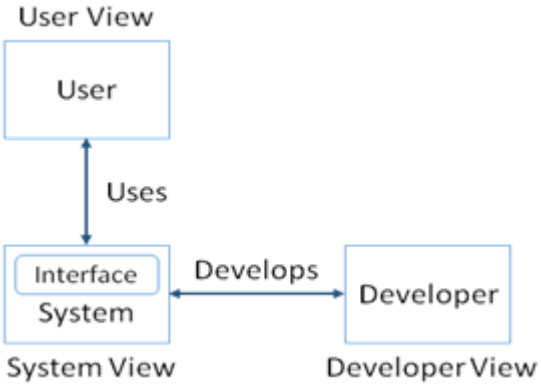

Figure1. Block Diagram that depicts different views (System, Developer & User)

- **Developer View:** This view describes about the mobile operating system guidelines and provided system functionalities that the developer needs to use in order to provide the best user interface to the users. For example, Guidelines such as offering informative feedback, design dialogs to yield closure and many more [9]. At the same time mobile manufacturers like Android and IOS enforce their own usability constraints[10] [11]

  From the developer's point of view, the developer needs to follow the standards of the mobile operating system and use the provided functionalities of the system. For example, displaying the necessary type of the keyboard (the current system provides various input functions shown in below table 1) based on the input to be entered by the user. Having unnecessary keys or navigation (displaying characters instead of numbers) on the keypad will have a critical increment in number of keystrokes and errors. However, most of the software developers are not using the default input method functions provided by the system for the applications.

Table. Android Major Standard Input Functions (For more [12])

Text Field	**input function (Property value)**
Person Name	textPersonName
Password	textPassword
Email	textEmailAddress
Numbers	number

- **System View:** This view describes about the system and the functionalities dealt by the system, the libraries provided by the system and how the system helps the user in providing the input. For example, Libraries such as material design, material design icons, material dialogs, material menu and many more [13].

  From the system's point of view, the system needs to provide an interface that will allow the users to accomplish specified goals with ease, efficiency and satisfaction in a specified context of use [9]. The interface needs to provide the necessary input method

based on the type of input, and, necessary help to the users to enter the input as efficiently and as easily as possible.

- **User View:** This view describes about how effortlessly and how effectively the users will be using the application that aids in fulfilling their requirements.

  From the user's point of view, the users desire to utilize the provided system with ease and to accomplish their objectives. Keeping in view the requirements of the users, the developer ought to build an application by following the guidelines and functionality provided by the system. The proposed system will help the developer to provide the best user interface to the user. The proposed system and its components will be explained in further sections in detail.

3. RELATED WORK

The fundamental factor in any mobile application is the user interface which depicts the effectiveness of the application, the ease with which the users are able to use the app and even measures the user thoughts and feelings about the app. The vast majority of the leading applications has been successful, essentially, by providing the best user interface and best user experience. Aside from fulfilling the business objectives, satisfying the commercial goals can be accomplished by considering the user interface part of an application.

In order to provide the best user interface to the users, the following issues associated with user interface have to be effectively addressed: Utilizing the screen space and Interaction mechanism [14]. Where utilization of screen space incorporates having different layout problems with small screens, Interaction mechanism incorporates difficulty in entering the text with more probability of entering the correct text by the user.

Among the various text entry methods for mobiles, entering the text by using soft keyboard is one of the most widely used method by users in mobiles. In 1867, Christopher L. Sholes and colleagues designed a new layout for keyboard named QWERTY and has become the de facto standard layout for both the virtual and physical keyboards [15]. The users were habituated to this kind of layout as it has been effective for several years and they tend not to see any new kind of layout as it generally consumes lot of time to learn and implement a new layout. QWERTY has a considerable effect of input speed compared to other keyboard layouts [16]. A study on various keyboard layouts in mobiles [17] demonstrates the familiarity and input speed with the QWERTY keyboard layout. This also depicts that users are able to type faster with the QWERTY keyboard as opposed to other optimized keyboards.

In 2000, in light of Fitts law, Silfverberg, et al. proposed a new mathematical model for predicting the text for the users while typing [18]. This predictive model helps users to enter the text very quickly, which was intended to diminish the number of keystrokes. The existing predictive system has a user database (Learning System/Personal dictionary) where each word entered by the users will be saved into the database at the time of typing. However, most of the users do not know how to use the personal dictionary effectively.

4. PROPOSED FRAMEWORK

The proposed framework includes four levels : Application Level, Application Framework Level, System Libraries and Kernel Level. Each level is briefly discussed below:

Application Level: All kinds of mobile applications will reside at this level and each application will have a user input that includes Input View and Input Field that allows the user to enter text.

- **Input View:** It is the user interface of all the input fields where the user inputs the text. The user selects one particular input field at a time.

- **Input Field:** Where the user can enter the data and it can vary in many ways depending on the data type. Input Field is connected to the input method editor to get the necessary keypad user interface (UI).

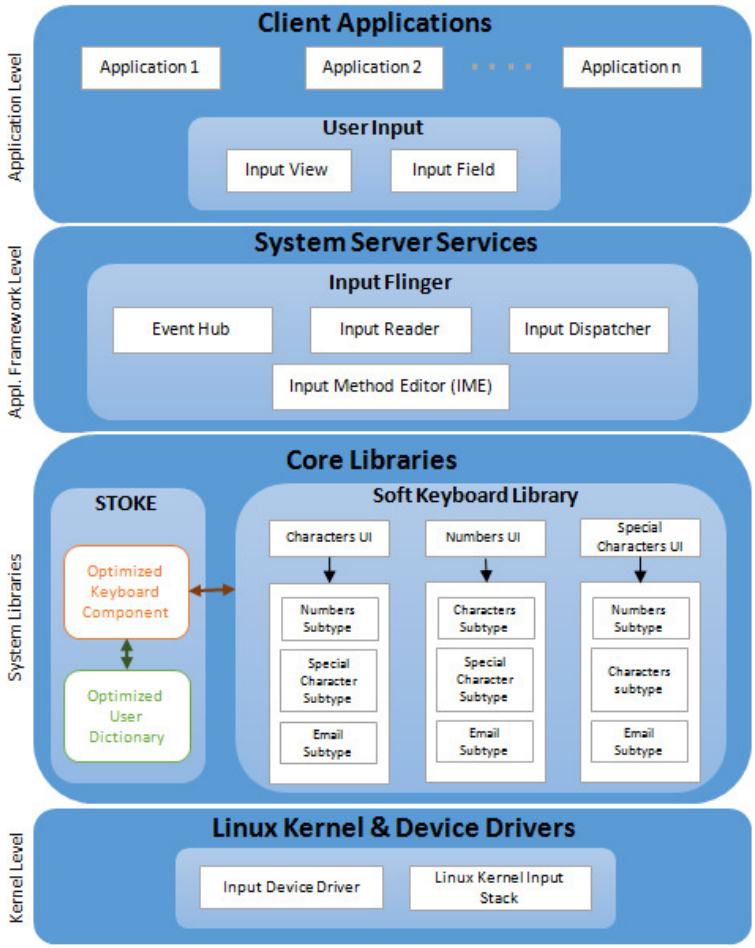

Figure2. Proposed Method in Android Framework
*STOKE = Smart Optimized Keyboard

Application Framework Level: This level has various system server services. Input Flinger, is the service which takes the input from the user. Input Flinger comprises of four components: Event Hub, Input Reader, Input Dispatcher and Input Method Editor. [20]

- **Event Hub:** The primary function of this component is to convert the raw events to operating system events. It also helps in adding and removing of devices with notifications.

- **Input Reader:** It acts like a client for the component event hub. It just reads the converted events and processes them. This processing is done with the help of input mappers. Once the processing is done, the input dispatcher is notified.

- **Input dispatcher:** This will get the processed events from input reader and identifies the target window. After identification, it dispatches to the target application. It uses the socket pair communication in order to know the status of delivery.

- **Input method editor:** This allows the user to enter the text, i.e., it provides an input method framework that helps the user to select various input methods.

Kernel Level: This level has two components: Input device driver, Kernel input stack. Where input device driver helps in registering the input_device. Input device driver also identifies the user input as events. For example, when user enters the text, device driver will create the file that has all interruptions from the user as events and these events are mapped with the default event code. This will be helpful in identifying the input from the user. [20]

System Libraries:

In this level, there are three major components: Core libraries, STOKE Keyboard component and Optimized user dictionary.

- **Core Libraries:** The following libraries are the proposed libraries which are inherited from the default libraries provided by the system. This helps STOKE keyboard to provide the respective keyboard based on the user input.

    - **Characters User Interface (UI):** This UI consists of three subtypes, namely, Numbers subtype, Special character subtype and Email subtype. These subtypes are helpful in displaying the respective keyboard on mobile screen. For example, when the input field is 'Name', then it is not required to display all the subtypes, thus increasing the screen size, decreasing the number of keystrokes and reducing human errors.

    - **Numbers UI:** This UI consists of three subtypes, namely, Character subtype, Special character subtype and Email subtype. These subtypes are helpful in displaying the respective keyboard on mobile screen. For example, when the input field is' zip code' then it is not required to display all the subtypes, thus increasing the screen size, decreasing the number of keystrokes and reducing human errors.

- **Special Characters UI:** This UI consists of three subtypes, namely, Character subtype, Numbers subtype and Email subtype. For Email the Characters UI displays first and then it navigates for selecting the appropriate emails through Email subtype.

- **STOKE Keyboard:** This is the proposed keyboard, which helps in getting the right keyboard based on the type of user input. This keyboard is an enhancement to the default keyboard and was designed by making some significant improvements from both system perspective and developer's perspective. As stated above, we have made some improvements to the existing library and likewise proposed a new library. For example, if Email is the input to be entered by the user, then STOKE keyboard will ask the Characters UI library to display the characters along with some extra features that include: '@' as a menu option. When a user selects that option the user will be navigated to a menu, where the user will be able to select the email types from the list, which includes @gmail.com, @yahoo.com, @outlook.com and many more (as shown in below Figure 3). Here the users will also be able to add their own email type that can be useful in future.

Figure3. STOKE keyboard for Email Type

This will be useful both for the users and the developers. From the users perspective, this will be useful as the number of keystrokes to enter an email, as an input and the screen size that the keyboard utilizes will be significantly decreased (Shown in below table 1). From the developer's perspective, there is no requirement to display the entire keyboard while the user is selecting the email types. Thus, developers can use the space for different purposes such as for better advertising and displaying other important aspects related to business.

Table 1. Keystrokes and Screen Size Comparison

Name of the Field	Number of keystrokes (Using existing system)	Number of keystrokes reduced (After using STOKE)	Screen Size utilization in % (Before adapting STOKE)	Screen Size reduced (After adapting STOKE)
EMAIL	1+8+1+4+1+9 = **24** keystrokes (eg: Seronika$1995@gmail.com (estimated keystroke count including characters, special characters and numbers)	8+1+4+1 = **14**	**53**	**51**

The above table clearly demonstrates that the number of keystrokes and screen size of the keyboard has been greatly reduced after STOKE keyboard was adopted. We have studied different emails of various users and calculated the average number of keystrokes required to enter an email (as shown in table 1). We have also calculated the number of keystrokes required for the user to enter an email address after adopting STOKE keyboard. There is a tremendous change in user experience as the users do not need to enter the complete email extension (ex: @gmail.com) with the STOKE keyboard. Therefore, on an average ten keystrokes will be reduced.

This research effort has also calculated the number of keystrokes and screen size for different input fields of various applications for both the default keyboard and proposed STOKE keyboard. The results demonstrate a significant difference between the keyboards, which clearly depicts an advantage in using the STOKE keyboard. The enhanced keyboard (STOKE) of our research effort does not compromise on the performance and security aspects. In addition to the above performance tests, a test was performed by calculating the time taken for both the Soft keyboard (Current) and STOKE keyboard (Proposed) to get displayed in IOS and Android. The results of both ANDROID and IOS are shown below in table 2 and figures 4&5 show the test results of both the standard keyboard and proposed keyboard in the IOS environment. Likewise, the testing process was conducted in Android environment:

Table 2. Performance Analysis

	ANDROID	**IOS**
Default Keyboard	0.46609 ms	0.10736 ms
STOKE	0.51201 ms	0.15782 ms

*Note that the results may vary based upon the system and the device that you use for testing. The displayed results are a resultant average of all test cases.

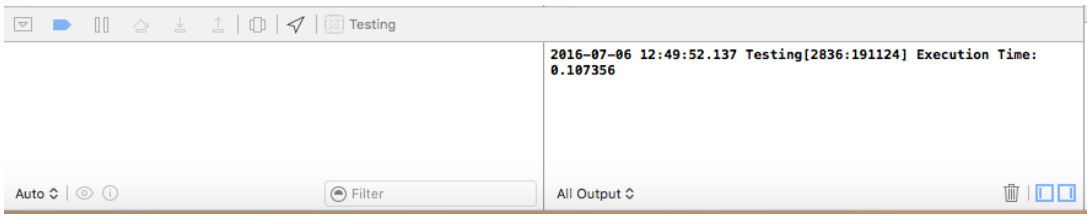

Figure 4. Time taken for the default IOS standard keyboard to get displayed.

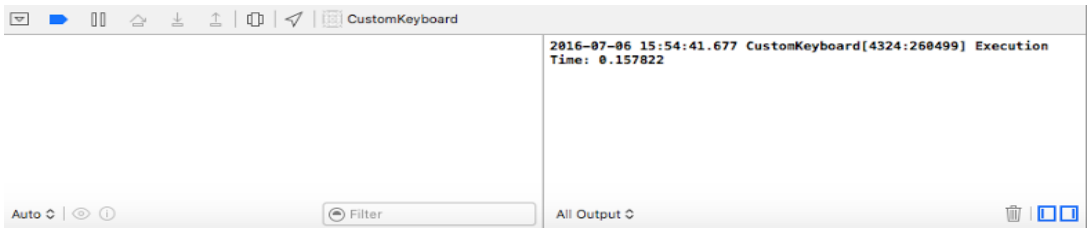

Figure 5. Time taken for the proposed(STOKE) keyboard to get displayed in IOS.

The above shown performance table clearly depicts that the STOKE keyboard has approximately consumed the same amount time in comparison to the default keyboard. Hence, this research effort claims that the users will not experience performance issues with respect to the proposed STOKE keyboard. Thus, this technique helps in providing the necessary keyboard, and also helps in reducing the unnecessary navigation by providing shortcut keys to accomplish the intended task (for example short cut keys while giving an email as input), which also significantly enhances the input speed for the user.

- **Optimized User Dictionary:** Predictive texting helps in saving a lot of time by providing some predictive words related to the entered character or word by the user. This will enable the user to enter the complete word/input in a single stroke [19]. The predictive texting, thus, reduces the number of keystrokes and also increases the accuracy, i.e., the probability of entering the wrong input is significantly diminished, thus, helping the user in finding the correct word in a shorter time. The predictive technique uses both language based dictionary and user dictionary (Personal dictionary) in order to provide predictions to the word/character entered by the user. The language based dictionary will be updated based on the language chosen by the user and the users do not have the permission to add or remove the words from the language based dictionary. In order to accomplish the addition or removal of the frequently used words by the user, the user dictionary has some into existence. This enables the users to add their own frequently used words in the dictionary which can be used in the future. Apart from this, there is a facility of text replacement that will help the users to create shortcuts for the complete phrase. This will allow user to enter the complete phrase (max length) by simply entering the corresponding shortcut word that was created.

  All of the above mentioned methods have significant advantages and greatly help in improving the interaction of the users with the mobiles; iff used appropriately. But, the current system doesn't provide the users with the necessary tools to use the above facilities. With the current system, users need to enter a single word/character at a time into the dictionary or users need to build the dictionary or text replacement dictionary

before using them. These two options will consume a significant amount of time, thereby compromising the user experience. So, in order to improve the usability, we have proposed a new method named Optimized User Dictionary, which will allow the users to enter the complete set of text (of any length) or previously sent messages simply by selecting the complete content or message. The system will consider the complete text and will separate each word from the sentence using various delimiters (space and symbols). The system also provides an option for the user to enter a shortcut for the complete sentence or message, which will be added to the text replacement dictionary. Also, the system also provides an easier means to delete the set of words from the dictionary. This provides an easier, faster and effective means for managing the user dictionary.

This system will be useful for the users while interacting with the mobile applications such as texting. One of the survey states that [21], US users spend 4.8 percent of their smart phone minutes in instant messaging applications. This will be higher if considered globally. As known to all of us, for chatting and messaging, 'English keyboard type' is a standout amongst the most broadly used keyboards around the globe. Users from different parts of the world send messages in their own language by using the English keyboard and the current system provides predictions of only English words. By adopting the proposed method of keyboard, the users of different languages can build their own dictionary in English and can get the complete sentences by simply entering the shortcut created for that particular phrase/sentence.

5. FUTURE WORK

We have proposed a theoretical approach and recommendations that will significantly help to ameliorate the current system. The user has many advantages with the acceptance and implementation of the STOKE keyboard and Optimized User Dictionary. In the future, we are going to provide the usability results i.e., how user has an ease in providing the input while uisng the proposed Smart Optimized Keyboard. For this, we are developing a sample working prototype for both Android and IOS mobile users that will be helpful in getting the usability results.

REFERENCES

[1] L. Picurelli, "The usability of mobile applications is the key element for their success," [Online]. Available: https://en.yeeply.com/blog/the-usability-of-mobile-applications-is-one-of-the-key-elements-for-their-success/. [Accessed 21 august 2013].

[2] J. NIELSEN, "Assessing the Usability of a User Interface Standard," Nielsen norman group, [Online]. Available: https://www.nngroup.com/articles/assessing-usability-user-interface-standard/. [Accessed 28 april 1991].

[3] J. Gould and C. Lewis, "Designing for usability: key principles and what designers think," 1985.

[4] Schadler, et al. "Mobile is the new face of engagement."Retrieved from Forrester Research: http://cdn.blog-sap. com/innovation/files/2012/08/SAP_Mobile_Is_The_New_Face_Of_Engagement. pdf (2012).

[5] S. Hoober, "Mobile Input Methods," [Online]. Available: http://www.uxmatters.com/mt/archives/2012/11/mobile-input-methods.php. [Accessed 1 november 2012].

[6] S. Mohamed, "Challenges facing mobile app usability testing," [Online]. Available: https://www.linkedin.com/pulse/challenges-facing-mobile-app-usability-testing-sally-mohamed?trkSplashRedir=true&forceNoSplash=true. [Accessed 21 october 2015].

[7] Zelster, "Security of third party keyboards on mobile apps" [Online]. Available: https://zeltser.com/third-party-keyboards-security/

[8] Reddit, "Performance Issues of third party keyboards", [Online]. Available: https://www.reddit.com/r/ios9/comments/3hn51m/does_ios_9_improve_3rd_party_keyboard_performance/?

[9] Gong, et al. "Guidelines for handheld mobile device interface design." Proceedings of DSI 2004 Annual Meeting. 2004.

[10] "IOS Human Interface Guidelines - Developer.apple.com." IOS Humar Interface Guidelines. Apple, n.d. Web. 22 Aug. 2016. [Online]. Available: https://developer.apple.com/ios/human-interface-guidelines/

[11] "The State of the Art of Mobile Application Usability" Android User Interface Guidelines. Google N.p., n.d. Web. 22 Aug. 2016. [Online]. Available: http://developer.android.com/guide/practices/ui_guidelines/index.html.

[12] "Various Input Functions". Google n.d. Retrieved August 22, 2016. [Online]. Available: https://developer.android.com/guide/topics/ui/controls/text.html

[13] Wasbeef. "Android User Interface Libraries." Google N.p., n.d. Web. 22 Aug. 2016. [Online]. Available: https://github.com/wasabeef/awesome-android-ui

[14] Heinecke, Andreas M., et al. "What Every Software Developer Should Know about Human-Computer Interaction–A Curriculum for a Basic Module in HCI in Informatics Education." ACM-IFIP.

[15] Nilsson, G Erik. "Design guidelines for mobile applications." SINTEF ICT, Jun (2008).

[16] Butts, Lee, and Andy Cockburn. "An evaluation of mobile phone text input methods." Australian Computer Science Communications. Vol. 24. No. 4. Australian Computer Society, Inc., 2002.

[17] Gkoumas, Apostolos, Andreas Komninos, and John Garofalakis. "Usability of Visibly Adaptive Smartphone keyboard layouts." (2016).

[18] MacKenzie, I. Scott, Shawn X. Zhang, and R. William Soukoreff. "Text entry using soft keyboards." Behaviour & information technology 18.4 (1999): 235-244.

[19] Wikipedia contributors. "Predictive text." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 30 Sep. 2016. Web. 30 Sep. 2016.

[20] "Android Input Architecture: Android Internals"[Online]. Available: http://newandroidbook.com/files/AndroidInput.pdf.

[21] Perez, Sarah. "A Survey on Smart Phone Usage." A Survey on Smart Phone Usage. N.p., 22 June 2015. Web. 22 Aug. 2016. [Online]. Available: https://techcrunch.com/2015/06/22/consumers-spend-85-of-time-on-smartphones-in-apps-but-only-5-apps-see-heavy-use/.

[22] Inukollu, Venkata N., et al. "Factors influencing quality of mobile apps: Role of mobile app development life cycle." arXiv preprint arXiv:1410.4537 (2014).