Personalized Mobile Navigation: Assessing the Benefits of Custom Gestures Over Default Methods

Faraz Akbarzadeh, Ammar Rafiqui, Omer Faruk Celik, Melika Sherafatt
Dept. of Electrical Engineering and Computer Science
York University
Toronto, Ontario, Canada M3J 1P3
{Fafaki80, Omer49, Ammarr58, Meddeta}@my.yorku.ca

ABSTRACT

A viable substitute for conventional mobile interactions is gesture-based navigation, particularly for repetitive tasks where accessibility and speed are key considerations. Using a controlled experiment with twelve people, this study investigated the performance and usability of customized gesture controls in comparison to the default system navigation techniques. With the help of a specially designed Android software, users were able to link movements to commonplace actions like opening apps. According to the results, customized gestures regularly obtained faster average task completion times and greater user satisfaction ratings, even if they did not statistically substantially exceed default approaches. Custom gestures had a little greater battery consumption, although this improved after several tests. In general, participants favoured gesture-based interactions, showcasing the potential for customized gesture navigation to improve the user experience on mobile devices.

Keywords

Gesture-Based Navigation, Mobile Interface Usability, Custom Gesture Controls, User Experience Optimization, HCI.

INTRODUCTION

Smartphone usage has surged over the past decade, enabling users to perform diverse tasks and install apps from a vast selection. A study by Ceci [1] reported that global mobile app downloads from the Google Play Store reached 110 billion in 2022, with the number of available apps hitting 1.68 million in 2024 [2]. As mobile technology continues to evolve at an astonishing pace, the way users interact with their devices critically impacts digital communication, efficiency, and productivity. The relationship between productivity and ergonomic factors becomes apparent because extended use of mobile devices leads to digital eyestrain and user fatigue while also increasing the risk of repetitive strain injuries (RSI) and conditions like carpal tunnel syndrome. Gesture inputs can minimize ergonomic risks like repetitive strain by lowering the need for constant tapping and unnatural finger motions while improving comfort in user interactions. The emergence of new ergonomic risks from poorly designed gestures necessitates a thorough evaluation of their impact.

One of the rising trends giving shape to user interaction today is gesture controls. Gesture controls offer a spontaneous, efficient, and customizable alternative to traditional touch-based interactions by enabling users to perform actions through predefined or personalized gestures. This alternative to traditional touch-based interaction reduces interaction on complex menus and eliminates unnecessary taps. Given the increased number of repeated tasks in many smartphones, such as launching a news app daily, navigating to hotspot in settings, gesture-based controls offer a more efficient and ergonomic solution. Unlike standard touch interfaces, which rely on taps and swipes, gesture-based user interfaces (GBUI) allow smoother interactions, often reducing cognitive load, enhancing user engagement by adapting the device experience to individual preferences and improving accessibility [3]. Gesture-based interfaces require ongoing gesture detection, which could lead to higher CPU demands and battery consumption than standard touch interactions. A thorough assessment of their power efficiency becomes essential for practical implementation decisions.

Advancements in mobile technology and gesture-based user interfaces have further advanced gesture recognition, increasing accuracy and responsiveness. Studies suggest that custom gestures can significantly improve task completion speed, reduce cognitive load, and enhance overall user satisfaction [4]. However, despite these advantages, challenges remain in recognition accuracy, user adaptability, and implementation across different applications. This paper explores these challenges by examining custom gestures in contrast to traditional touch interfaces, focusing on speed, accuracy, and usability.

Power consumption becomes a vital factor for gesture-based interfaces because ongoing gesture recognition requires increased processing power, which could reduce battery life. Maintaining user satisfaction relies heavily on efficient resource management in devices where battery life plays a major role in user experience.

The paper is organized as follows: Section 2 provides an overview of related works on gesture interaction and its applications across various areas. Section 3 presents the methodology used in this study. Section 4 discusses the results of our user study. Finally, Section 5 presents conclusions and suggestions for future work.

RELATED WORK

Sun et al. [5] investigated the use of dynamic shortcuts to streamline app launching. They developed an application called *AppRush*, which predicts the apps a user is most likely to launch based on their app usage history. The system features a frontend widget that dynamically displays frequently used applications, enabling users to access them more efficiently. Their study demonstrated significant potential for reducing the time needed to open apps. However, the results indicated that further refinements were necessary to enhance prediction accuracy and user experience. The research recommended enhancements to the prediction model through the addition of user behaviour features along with the use of advanced machine learning algorithms to more accurately forecast user requirements.

Li [4] introduced *Gesture Search*, a mobile application that facilitates quick access to mobile data by integrating gesture-based interaction with search functionality. The system allows users to draw gestures on a touchscreen to retrieve contacts, applications, and other data efficiently. A user study revealed that participants primarily used the application for contact retrieval. Moreover, the application received highly positive feedback, with an average user rating of 4.5 out of 5 based on over 5,000 reviews. However, there is still room for improvement in app-based gesture interactions, particularly in enhancing recognition accuracy and expanding support for more complex gestures to improve usability.

Modern mobile interface technology developments emphasize touch-based interactions, although their intuitive nature does not offset the issues related to speed and accessibility. Research by Pomboza-Junez et al. [3] and Kim, Nam and Park [8] demonstrates a movement towards refining gesture-based controls to enhance natural user interactions on mobile and wearable devices.

Developing gesture-based interfaces presents various challenges, particularly in making these systems accessible while integrating them into daily-use technology. Loch [7] discovered that hierarchical gestures for touchscreen devices create new opportunities for user interaction customization that improve mobile device usability. The research from Li [4] alongside Arsan et al. [6] demonstrates that gesture-based shortcuts contribute to faster data access and improved performance of routine tasks on mobile devices. Arsan et al. [6] found that gesture-based shortcuts result in faster data access and increased efficiency when performing common tasks on mobile devices.

Even though advancements show potential, comparative analysis of customizable gesture efficiency versus default system navigation remains an unsolved issue. This research seeks to fill the current research gap through the creation of an application that enables users to establish and store personalized gestures for routine activities to discover how these tailored gestures can optimize processes that are currently limited by slower and fixed interaction methods.

METHOD

This comparative user study employed an experimental data collection method to investigate the efficiency and usability of customizable gesture controls compared to default system navigation for performing repetitive tasks on mobile phones or tablets. Participants completed a series of tasks using both default system navigation and custom gestures. Before testing, participants completed a questionnaire to gather additional data on their familiarity with gesture-based interactions and smartphone usage habits. The combined data was analyzed to determine whether a significant difference exists between the two interaction methods regarding task completion time and user satisfaction.

Participants

Twelve participants, six male and six female, aged between eighteen and twenty-four, were recruited from among family members, friends, and classmates on campus. Participation was voluntary, and no monetary compensation was provided.. They were selected based on their mobile phone or tablet usage, which was at least two hours per day, and their familiarity with common mobile tasks and navigation gestures. This ensured a diverse representation of individuals with varying levels of technical proficiency. For the study, participants were volunteers who were recruited by the research team.

Hypothesis Statement

This study evaluated the efficiency of customizable gesture controls compared to default system navigation for performing repetitive tasks on mobile phones or tablets. The following hypotheses were tested:

- Performance Hypothesis:

H_0: There is no difference in task completion time between custom gestures and default system navigation.

H_a: Custom gestures result in faster task completion times compared to default system navigation.

- Statistical Test:

A paired t-test will compare the mean task completion times for the two methods. This test is suitable for comparing the means of two related groups when the data is continuous and normally distributed.

- Battery Consumption Hypothesis:

H_0: There is no difference in battery consumption between custom gestures and default system navigation.

H_a: Custom gestures result in lower system resource consumption compared to default system navigation.

- Statistical Test:

A paired t-test was used to compare the mean battery consumption between the two methods. This test is suitable for comparing the means of two related groups when the data is continuous and normally distributed.

- User Satisfaction Hypothesis:

H_0: There is no difference in user satisfaction between custom gestures and default system navigation.

H_a: Custom gestures result in higher user satisfaction compared to default system navigation.

- Statistical Test:

A Wilcoxon signed-rank test was used to compare the median satisfaction scores for the two methods. This test is appropriate because the data is ordinal and not assumed to be normally distributed.

Apparatus

The study was conducted using a custom-built application developed using Android Studio (Version Meerkat 2024.3.1 Patch) [9]. The study used the examiner's Android smartphone (Google Pixel XL), which ensured consistency in the testing apparatus across trials. The application allowed users to define and save custom gestures for common tasks, such as launching a specific application. Only four letters were implemented for user testing. These were selected as representative samples, with the assumption that if the system performed effectively for them, it would likely generalize well to the full alphabet and support for future custom gestures. Task completion times were logged automatically by the application. Android Studio profiling tools monitored additional metrics, such as battery consumption. The experiment conditions remained consistent throughout the study. The testing area was a relatively quiet and comfortable private room inside Scott Library, and it remained the same throughout the study, allowing the participants to focus on the task. After the study was completed, the research team used *GoStats* to perform the statistical analysis to extract the data and examine the results to see if there was a significant difference [10].

Procedure

Participants will first be informed about the nature of the study, after which they will be asked to sign a consent form and will be given the first portion of the questionnaire. Then, the experimental instruction will be read aloud, detailing the operation of the experimental interface, the process of the experiment, and the experimental task. Participants will be informed that they will perform a series of tasks using two interaction methods, custom gestures and default system navigation. The tasks will involve common actions, such as opening frequently used apps. Participants used a standardized device (Google Pixel XL provided by the researchers) to minimize variations in hardware sensors, software performance, and interface responsiveness that might influence study outcomes. Participants uncomfortable or unfamiliar with the Android OS were provided additional training sessions before beginning the experiment to ensure comfort and competency, thereby maintaining internal validity.

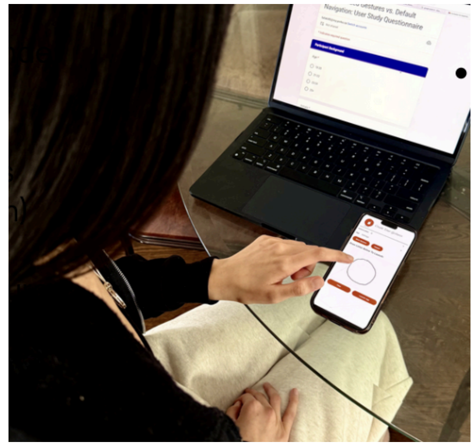

Figure 1. Participant's interaction with GestureIT.

Participants will open a commonly used app, such as Chrome. By default, they will open the app by navigating through the app drawer or home screen. To access the same app using the custom gesture technique, they will make and execute a gesture, such as drawing an "I" on the screen. The goal is to check which method is faster and the most preferred option by participants. Time to complete the work, user effort (evaluated on a Likert scale from 1 to 5), and participant satisfaction (measured by a post-task survey) are some of the metrics that will be used to assess how well these approaches operate.

Every task will have a timer. Participants will rate user effort and satisfaction following each task. To assess how well the default navigation methods and customized gestures perform on all tasks, the gathered data will be statistically analyzed using paired t-tests.

Design

The user study employs a 2 x 3 within-subjects study design. The independent variables and levels are as follows:

- Interaction Method: Custom Gestures, Default Navigation
- Task: Opening an App

For every interaction mode (default navigation and custom gestures), each participant will perform a task: opening an app. The experiment will therefore have a total of 72 trials (12 individuals x 2 interaction methods x 3 attempts x 1 task = 72 trials).

Participants will be split into two groups to balance the testing order and avoid any learning effects. In one group, the default technique will be used first, followed by custom gestures; in the other group, custom gestures will be used first, followed by the default approach. To guarantee that there is no bias in the order, participants will be assigned at random to one of the groups. Task completion time, user effort, and satisfaction will all be considered dependent variables. The amount of time it takes a participant to do all the tasks using the chosen method will be measured by task completion time. A questionnaire adapted from established usability and user experience instruments, such as the System Usability Scale (SUS) by Brooke (1996), will be used to collect qualitative data regarding user preference and usability for both interaction techniques. Additionally, attention-check measures, including inverted Likert-scale questions, will be incorporated to ensure participants maintain focus and respond consistently throughout the study.

RESULTS AND DISCUSSION

Within the 72 sets of data, there were no major outliers that significantly deviated from the general trend. All participant performances were included in the calculations, as any variations observed were representative of natural differences in user familiarity and adaptability. Out of the data collected, three key metrics were used to assess the effectiveness of the interaction methods: performance, resource efficiency, and user satisfaction. These variables were selected to provide a well-rounded understanding of the potential long-term viability for this new mobile interaction.

ANOVA tests were conducted on each of these datasets to determine statistical significance. The averages across all participants and multiple trials were used to evaluate the hypotheses presented earlier in this study. This analysis helps to determine whether GestureIT offers measurable advantages over traditional touch-based interaction.

Performance

The performance was measured by users' task completion time in seconds. The overall mean completion time across both interaction types was 10.64 seconds. As seen in Figure 2, the GestureIT interaction had a mean completion time of 9.85 seconds (SD = $\pm\ 0.92$), while the traditional interaction method had a mean time of 11.43 seconds (SD = $\pm\ 2.21$). GestureIT was 16% faster compared to the traditional interaction type. However, the effect of interaction type (GestureIT vs. Traditional) on task completion time was not statistically significant ($F_{1,11} = 3.027$, $p > .05$), likely due to the small sample size and high variability in the Traditional interaction group.

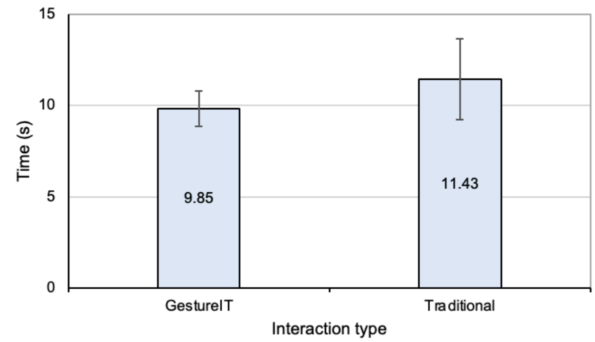

Figure 2. Task completion time in seconds by interaction type.

As seen in Figure 3, users became more familiar with the GestureIT app, task completion times steadily improved, showing a pattern similar to the traditional method. By the third trial, the average completion time for GestureIT was 8.76 seconds, compared to 9.60 seconds for the traditional method. Initially, both methods started with higher times: 10.63 seconds for GestureIT and 13.88 seconds for the traditional method. From Trial 1 to Trial 3, GestureIT showed an 18% improvement in completion time, indicating that users became faster with practice. In comparison, the traditional method showed a 31% improvement. The effect of repeated trials on completion time was statistically significant ($F_{2,22} = 5.887$, $p < .01$). However, the effect of interaction type across the trials was not statistically significant ($F_{2,22} = 2.891$, $p > .05$), possibly due to overlapping performance trends between the two methods as users became more proficient, reducing the distinction in effectiveness between interaction types over time.

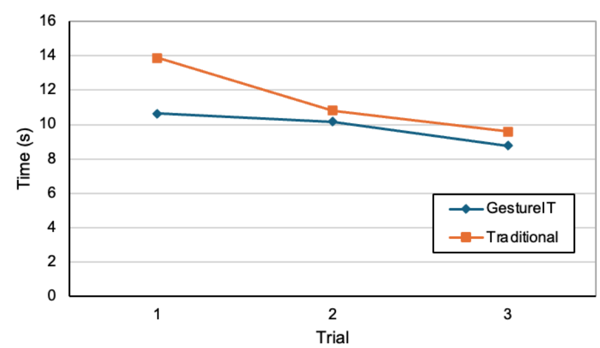

Figure 3. Task completion time in seconds by interaction type and trials.

Hence, these findings suggest that while GestureIT did not significantly outperform the traditional method statistically, its consistent performance, faster average completion times, and improvement with practice indicate promising potential for becoming a more efficient and user-friendly interaction method with further refinement and user adaptation.

Resource Efficiency

The battery usage was measured by the amount that was drained in the total time it took the user to complete the task. The overall mean current drain for GestureIT was -645.47 µA (SD = ± 54.53), while Traditional interaction consumed an average of -534.13 µA (SD = ± 43.43). As shown in Figure 4, GestureIT consumed 21% more battery power than the traditional interaction method. However, the difference in battery consumption between interaction types was not statistically significant ($F_{1,11} = 2.224$, $p > .05$), suggesting that any observed difference may be due to random variation rather than a consistent effect of interaction method.

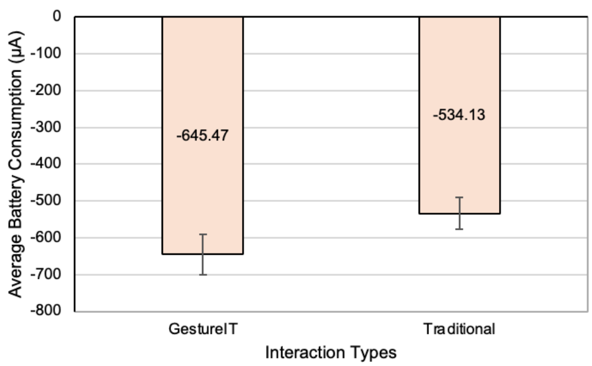

Figure 4. Average battery consumption (µA) by interaction type.

As shown in Figure 5, battery consumption trends varied over three trials. GestureIT began with the highest battery usage in Trial 1 at -707.21 µA, but improved steadily to -603.88 µA by Trial 3, resulting in a 15% improvement in power efficiency. Conversely, Traditional interaction showed a slight increase in battery drain over time, starting at -529.57 µA in Trial 1 and increasing to -549.64 µA in Trial 3. Although neither method showed statistically significant differences in consumption trends over time ($F_{2,22} = 0.776$, *ns*), the data suggests that GestureIT may become more resource-efficient with continued use, while the Traditional method may show signs of stagnation. This means that while both methods exhibit changes in energy consumption, the observed variations over the three trials were not large or consistent enough across participants to be confidently attributed to the interaction method rather than random fluctuation. Similarly, the effect of interaction type across the trials was not statistically significant ($F_{2,22} = 2.714$, $p > .05$), indicating no strong evidence that one method consistently performs better or worse than the other across all trials.

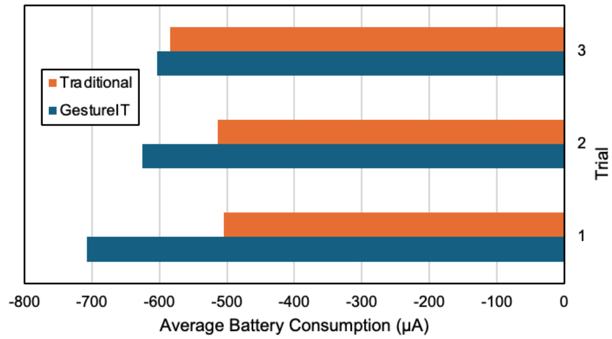

Figure 5. Average battery consumption (µA) by interaction type and trias.

Thus, these findings suggest that while GestureIT may not yet be a definitive replacement for traditional navigation in terms of resource efficiency, with further battery optimization and an increase in user familiarity, it has the potential to evolve into a more viable and power-conscious alternative in future implementations.

User Satisfaction

User satisfaction was assessed through two main metrics: intuitiveness of gesture-based interactions and overall preference compared to default navigation methods.

Figure 6 illustrates participants' ratings of intuitiveness for the gesture-based interaction method. Participants rated intuitiveness on a scale from 1 (not intuitive at all) to 5 (extremely intuitive). Among the 12 participants, most rated the gesture-based interactions as moderately intuitive or better, with approximately half of them assigning ratings of 4 or 5. This indicates that while users generally perceived the gestures as straightforward and easy to understand, there is still room to enhance their naturalness and intuitiveness further.

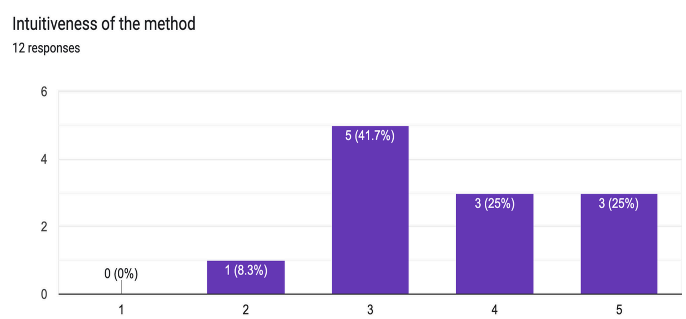

Figure 6. Participants' intuitiveness ratings for gesture-based interactions (1 = not intuitive, 5 = extremely intuitive.

Figure 7 shows participants' overall preference between default navigation methods and custom gesture interactions. A significant majority, approximately 75% of participants, preferred custom gestures, while only 25% favoured default navigation methods. This strong preference underscores users' inclination towards custom gestures, highlighting their perceived superiority in enhancing interaction satisfaction. Users found custom gestures to be a more satisfying and efficient method for engaging with their devices, reinforcing the value of personalized gesture interactions in improving overall user experience.

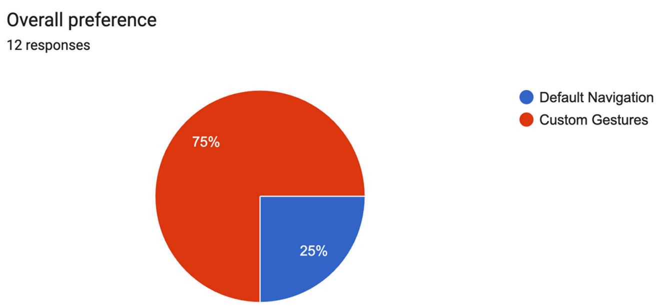

Figure 7. Participants' overall preference between default navigation and custom gestures

Overall, these findings indicate positive user satisfaction with custom gesture-based navigation, though further refinement can improve intuitiveness for broader acceptance.

CONCLUSION

In conclusion, the majority of users found gesture-based interactions to be more intuitive, faster, and more user-satisfying, however, these differences were not statistically significant. Participants valued the opportunity to customize their experience and felt that gestures were natural, particularly when they fit with their routines or mental models. Additionally, ergonomic advantages surfaced since motions eliminated the need for repetitive tapping, which may lessen strain and fatigue. In addition, the GestureIT application was faster compared to the traditional type, which also suggests that with future improvement, it can replace the traditional navigation. Moreover, custom gestures initially used more battery, but after repeated use, the battery consumption decreased, indicating potential for improvement.

The study did have many drawbacks, though, chief among them being its small sample size, brief length, and single-device testing. Some users voiced worries regarding gesture overlap and unintentional triggers. Recognition systems, and machine learning based adaptive gesture recommendations should all be part of future studies. In addition, user feedback suggested that future implementation should include adding a widget to further enhance the speed and accessibility of the GestureIT application. Custom gestures have the potential to greatly improve mobile interaction's comfort and utility with additional development.

REFERENCES

1. Ceci, L. (2024a). Google Play annual app downloads 2020. Statista. https://www.statista.com/statistics/734332/google-play-app-installs-per-year/
2. Ceci, L. (2024b). *Google Play Store: number of apps 2019 | Statista*. Statista. https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/
3. Pomboza-Junez, G., Holgado-Terriza, J.A., & Medina-Medina, N. (2019). Toward the gestural interface: comparative analysis between touch user interfaces versus gesture-based user interfaces on mobile devices. *Universal Access in the Information Society, 18*, 107–126. https://doi.org/10.1007/s10209-017-0580-6
4. Yang Li. (2010). Gesture search: a tool for fast mobile data access. In *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology* (UIST '10). Association for Computing Machinery, New York, NY, USA, 87–96. https://doi.org/10.1145/1866029.1866044
5. Chen Sun, Jun Zheng, Huiping Yao, Yang Wang, D. Frank Hsu. (2013). AppRush: Using Dynamic Shortcuts to Facilitate Application Launching on Mobile Devices. *Procedia Computer Science, 19*, 445-452. https://doi.org/10.1016/j.procs.2013.06.060
6. Deniz Arsan, Ali Zaidi, Aravind Sagar, and Ranjitha Kumar. (2021). App-Based Task Shortcuts for Virtual Assistants. In The 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21). Association for Computing Machinery, New York, NY, USA, 1089–1099. https://doi.org/10.1145/3472749.3474808
7. Loch, F. (2012). *Hierarchical gestures: Gestural shortcuts for touchscreen devices* (Master's thesis, University of Twente). https://purl.utwente.nl/essays/61931
8. Kim, J. W., Nam, T. J., & Park, T. (2017). CompositeGesture: Creating custom gesture interfaces with multiple mobile or wearable devices. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 11, 77-82. https://doi.org/10.1007/s12008-014-0208-5
9. The GestureIT application was developed by Faraz Akbarzadeh, Ammar Rafiqui, Omer Faruk Celik, Melika Sherafatt, https://github.com/AmmarR58/EECS-4443-Group-Project
10. The GoStats application was developed by I. Scott MacKenzie, the API data is at https://www.yorku.ca/mack/GoStats/GoStats.html