Towards an Improved Shortcuts App through Siri Shortcuts

Evangelos Kassos
ekassos@college.harvard.edu

ABSTRACT

The Shortcuts app is an application distributed by Apple across its platforms and enables users to extend the functionality of their devices with personalized shortcuts. User evaluation showed that the current version of the application lacked adequate tools to discover actions, the building blocks of shortcuts, and debug user-created shortcuts. This paper introduces extra functionality to aid in action discovery and shortcut troubleshooting, built on top of the Shortcuts app through Siri shortcuts. A user study empirically verified that the proposed interventions enhanced the Shortcuts experience for both novice and advanced users. What remains to be seen is the impact on the user experience after the interventions' integration in the Shortcuts application.

Author Keywords

Shortcuts; actions; Siri; feature discovery; debugging; natural language processing (NLP).

CCS Concepts

•**Human-centered computing** → **User interface programming; User interface toolkits; Graphical user interfaces;** *Natural language interfaces; User studies; Usability testing;*

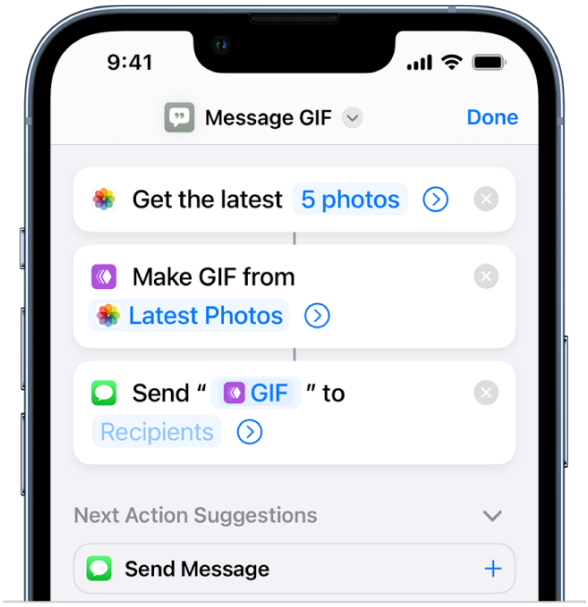

Figure 1. Building a shortcut with 3 actions already placed [2].

INTRODUCTION

Shortcuts is a visual scripting application developed by Apple and provided on its iOS, iPadOS, macOS, and watchOS operating systems. The app lets users create macros for executing specific tasks on their devices [2].

An *action*, the building block of a shortcut, is a single step in a Shortcuts *task*. These task sequences can be created by the user using action "blocks" and shared online for other users to use (Figure 1). A number of curated shortcuts can also be downloaded from the integrated Gallery section of the Shortcuts app.

Shortcuts can be manually activated through the Shortcuts app, Shortcut widgets placed on the user's home screen, the operating system share sheet, and Siri. They can also be automated to trigger after an event (called an "Automation"), such as the time of day, leaving a set location, a user activating a Focus mode or opening an app.

Shortcuts have seen a great boost in development from users (see: r/shortcuts [4], RoutineHub [5]) after the expansion of support for all Apple platforms and operating systems. However, literature on the application is extremely limited. The single published paper in the topic explored specific uses of Shortcuts for user groups, exploring how Shortcuts could help create an efficient surgical logbook tool [6]. Most media coverage or research on Shortcuts simply addresses potential applications, with even those guides and approached remaining very technical.

I hypothesized that the increased technicality of Shortcuts actions would limit the perceived utility of the Shortcuts app from novice users. I conducted a user-evaluation study of the current Shortcuts app and validated the claim that beginner users of the Shortcuts app find it hard to identify its utility. Moreover, I discovered that experienced users reported missing desired debugging and testing tools for their shortcuts. I hypothesized that the implementation of said missing features would improve the perceived utility of the Shortcuts app for all types of users.

In this paper, I present four modifications of the Shortcuts app to address issues that came up through the user evaluation study. To do so, I implemented an extra layer of user interaction using Siri Shortcuts, which were triggered when the user said a specific phrase, since the functionality could not be directly integrated into the app interface. First, a suggestion panel presents the user with actions that they might want to add based on the latest action they selected. Second, a search function allows the user to describe their desired functionality and get suggestions on actions. Third, a debugging panel presents the user with a flowchart of how their actions are connected with each other. Finally, a debugging panel allows the user to input specific initial values for testing and see the values "flow" through the Shortcut flowchart.

To test the usability and utility of the presented modifications, I performed a user evaluation study with a group of experienced and novice Shortcuts users. The results empirically support the claim that the modifications implemented enabled novice users to discover more relevant actions and allowed more experienced users to debug their shortcuts effectively. Ultimately, the true effect of the interventions presented in this paper cannot be fully evaluated without their full integration into the Shortcuts app, but the results illustrate the need for more action-selection and debugging methods within the Shortcuts app. The results presented also need to be extended to other Apple platforms, including watchOS and macOS.

RELATED WORK

Research on the Shortcuts app has been limited, with the only recorded Shortcuts application being that in the work of Daniel Thompson [6]. The paper presented an "Operation Note" shortcut, which increased accuracy and efficiency in surgical logbook data collection, but proposed no modifications to the actual Shortcuts app. One of the main issues with expanding and iterating on the Shortcuts app is that its codebase is proprietary and only distributed by Apple through its platforms. However, the Shortcuts app has not seen a huge redesign or significant feature expansion since its first release.

DISCOVERING USER NEEDS

When introduced, the Shortcuts app was marketed as an app to run shortcuts provided by apps installed on the user's device and for users to create create their own shortcuts [1]. Ever since, both functionalities have seen immense development, with more apps providing app-specific shortcuts and more actions being available for user-created shortcuts. Today, iOS, iPadOS, watchOS and macOS users, use Shortcuts to automate some part of their device workflow or perform complex actions that they weren't able to perform by simply using apps on their phone. as described below.

Usability Evaluation Design

To understand how useful and efficient the current iteration of the Shortcuts app is for users, I invited a diverse group of iOS users to participate in a usability evaluation study. The group contained three different subgroups:

- 5 users that had never used the Shortcuts app before (*novice group*),
- 5 users that had only used shortcuts provided by installed apps (*intermediate group*), and
- 5 users that had used the Shortcuts app to create their own shortcuts (*advanced group*).

The study was limited to current iOS users that had been using an iOS device (iPhone, iPod, iPad) for at least six months. This was done to ensure that the participants were familiar with the iOS paradigm and mental models, which every iOS app, including Shortcuts, is based on.

Usability evaluation tasks

The purpose of the user evaluation was to understand the user's attitude towards the Shortcuts app, specifically when creating and using user-created shortcuts. Each participant was asked to perform the following tasks:

1. Open the Shortcuts app and create an empty shortcut.
2. Create a Shortcut that calculates the right tipping amount. The shortcut should give users an input field for the total amount price (without tax), and a list of tipping percentages. It should display the total pay amount and the tipping amount.
3. Create a Shortcut that sends a message to a specific contact. The shortcut should ask for a specific message and add a greeting at the beginning of the message based on the time of day. For example, it should add "Good morning!" at the beginning of the message if the message is sent in the morning hours.

Note that the first shortcut creation task contains specific action hints that users could use, while the second one is open-ended, to evaluate the action-discovery functionality. Users were instructed to test and debug both of their created shortcuts to evaluate the debugging functionality.

Results

Overall, participants from all groups were able to perform or complete some or all the tasks. Table 1 presents a concise review of which users completed each task. As expected, most users found the creation of the messaging shortcut the hardest, while many managed to create a working version of the tipping shortcut. Additionally, users in the novice group had the hardest time creating correct shortcuts, while all users in the advanced group managed to create a correct tipping shortcut. Finally, even though all users in the advanced group were able to create the messaging shortcut, two users were not able to fully debug the shortcut they created.

Based on the user comments that the users provided in their off-boarding interview the following four main conclusions were extracted:

1. There is no easy way to discover and explore all functionality and available actions. Specifically, it is hard to discover actions that may help expand the current functionality the shortcut provides.
2. Searching for actions that will perform a specific task is currently very challenging, as the user needs to remember or know part of the name of the action.

	Number of users completing...		
Group	*Task 1*	*Task 2*	*Task 3*
Novice group	5 (100%)	2 (40%)	0 (0%)
Intermediate group	5 (100%)	4 (80%)	1 (20%)
Advanced group	5 (100%)	5 (100%)	3 (60%)

Table 1. Number (and percentage) of users in the original user evaluation study that correctly completed each task. Task numbers correspond to the numbered list in Usability Evaluation Tasks. Task 1: Create empty shortcut; Task 2: Create tipping shortcut; Task 3: Create messaging shortcut.

3. Debugging a long or complicated shortcut seems often almost impossible. To do so, many users resort to interjecting "Quick Look" actions as a way to check the value of variables, in an attempt to simulate breakpoint debugging. A time-consuming process of unraveling actions, executing them step by step and looking at the output often ensues.

4. There is no way to test the user-created shortcut by providing pre-determined input and seeing how the inputs are modified until the shortcut terminates, especially without triggering Shortcut actions like sending messages.

INTERVENTION IMPLEMENTATION

The main take-aways from the initial user evaluation study revolved around two themes: discovering actions and debugging. In this paper, I present four interventions aimed at addressing those concerns and elevating the Shortcuts app experience.

Preliminaries

To implement the interventions described below, I created Siri shortcuts that were triggered when the user said a specific phrase. This is in replacement to the function of having the buttons directly integrated into the Shortcuts app interface, as the codebase is proprietary and not accessible. In some cases, when the user interacted with the screen of the intervention, they were redirected back to the Shortcuts app and some actions were performed using AssistiveTouch, simulating the actions the app would support if the implementation was integrated into the Shortcuts app. Specific implementation details are included in each intervention description below.

Data collection

Some of the following interventions required data regarding the popularity of actions within user-created shortcuts, as well as information about the relationships between actions. For instance, many users added a "Get contents of URL" right after passing on a link through the "URL" action. This section describes how the data was collected.

Reddit is a network of communities where people can dive into their interests, hobbies and passions. Some users create subgroups (subreddits) with a focus on specific interests, which other users can join and create posts for. The *r/Shortcuts* subreddit currently has over 277,000 members and provides a space for users to discuss, troubleshoot and share their shortcut creations. As discussed in the Introduction, user-created shortcuts are sharable via links hosted on iCloud, Apple's cloud storing service. Using these sharable links other users can download, save, and run the shortcuts locally.

To gather the data described above, I performed a subreddit-wide search for all user-created Shortcuts links posted by users. After downloading all corresponding Shortcuts, a Shortcuts parser created relationship graphs between action blocks, counting how many times each block appeared, and how many time an action combination appeared (e.g., "URL" action → "Get contents of URL" action, as described above). Specific implementation details about how this data was utilized are included in each intervention description below.

Interventions to aid action discovery

Action autosuggestions

This intervention aims to augment the current "Next Action Suggestions" panel currently implemented in the Shortcuts app. The panel presents three or four action suggestions that the user might find useful to integrate into their app. As participants noted in the initial user evaluation study, while building their shortcuts, the actions presented were not relevant and rather distracting instead.

The action autosuggestions intervention makes use of the action relationship dataset created from the r/Shortcuts subreddit. The dataset provided a model on how to suggest future actions based on the actions the user had already placed. Since the universe of possible actions and action combinations is almost infinite, I focused on the most-used actions for this implementation.

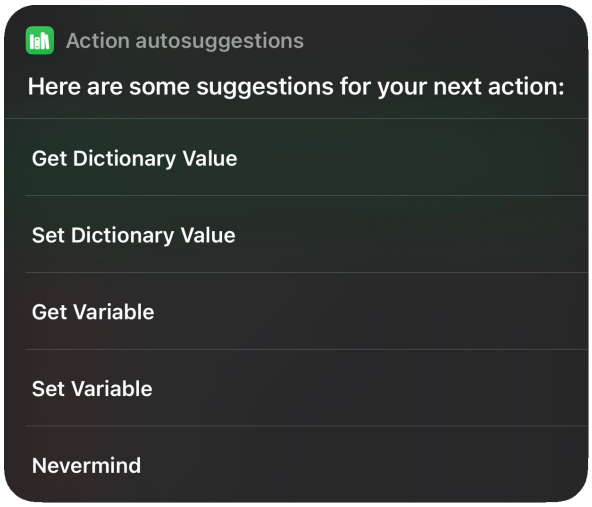

Figure 2. Example of an autosuggestion prompt.

Since it was not possible to integrate this functionality into the Shortcuts app and replace the "Next Action Suggestions" panel, a Siri shortcut was implemented. Users bring up the shortcut by saying *"Hey Siri, show me suggestions for [action name]."* The shortcut shows a list of clickable suggested actions (Figure 2). Once the user selects an action from the list, they are redirected back to the Shortcuts app and the selected action is searched for and added to the shortcut using AssistiveTouch. The suggestion is also stored in the Siri shortcut

to enable suggested action list ranking based on the chain of actions already placed. Finally, a "nevermind" option is included in the action suggestions list in case the user does not find any of the presented suggestions relevant.

Natural language action suggestions

This intervention aims to augment the current search functionality currently implemented in the Shortcuts app. The current search function requires users to search using keywords included in the name or short action description. This creates for a sub-optimal user experience, especially for novice or intermediate users who are not familiar with exact action names. For instance, if a user searched for "insert value" hoping to find the action to create a new dictionary value ("Set Dictionary Value"), the search would return no results. As participants noted in the initial user evaluation study, some users reported thinking some functionality was not available, because it would not show up in their search queries.

The natural language action suggestions intervention makes use of the Stanford Named Entity Recognizer (CRFClassifier) [3] and the action name and descriptions provided in the Shortcuts app. The action name and descriptions provided a model on which the CRFClassifier was able to identify the relevant action based on the user's search query. Since the universe of possible actions is extremely large, I focused on the most-used actions for this implementation.

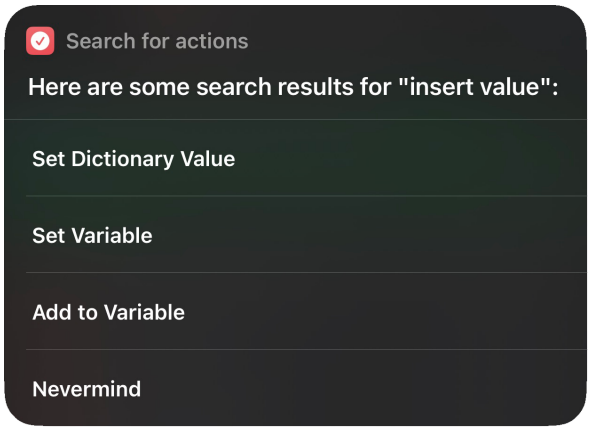

Figure 3. Example of a search results action prompt for the search query "insert value."

Since it was not possible to integrate this functionality into the Shortcuts app and replace the search functionality, a Siri shortcut was implemented. Users bring up the shortcut by saying *"Hey Siri, search for action."* The shortcut shows an input filed, where the user types the search query they would have typed in the actual shortcut search field. The shortcut then presents the user with a list of clickable suggested actions (Figure 3). Once the user selects an action from the list, they are redirected back to the Shortcuts app and the selected action is searched for and added to the shortcut using AssistiveTouch. Finally, a "nevermind" option is included in the action suggestions list in case the user does not find any of the presented suggestions relevant.

Interventions to aid debugging

Flowchart debugger

This intervention aims to help users understand how the information flows throughout the shortcut actions. As participants noted in the initial user evaluation study, while building their shortcuts, understanding the flow of a complicated shortcut seemed almost impossible for users.

The flowchart debugger intervention makes use of the same Shortcuts parser that was used to parse the data from the r/Shortcuts subreddit. The data is then served to the intervention shortcut, which creates an easy to follow diagram that highlights relationships between data. For instance, if-statement blocks are prominently displayed, and variable relationships are depicted with arrows. Since the universe of possible actions and action combinations is almost infinite, I focused on the most-used actions for this implementation.

Since it was not possible to integrate this functionality into the Shortcuts app, a Siri shortcut was implemented. Users bring up the shortcut by saying *"Hey Siri, show me the flowchart."* The shortcut shows a flowchart picture generated by the flowchart debugger. The user can examine the flowchart and click "Done" to return to the Shortcuts app.

Test case debugger

This intervention aims to help users test their shortcut using specific initial values, in a similar way that software developers test their code by providing some initial values through unit tests. As participants noted in the initial user evaluation study, it is extremely difficult in the current Shortcuts app version to test the user-created shortcut, especially without triggering outwards-facing actions, like sending a message. Many users resort to building the shortcut piece-by-piece and testing each action independently, which increases the creation and testing time.

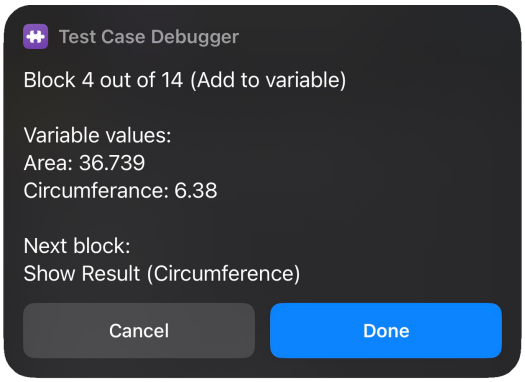

Figure 4. Example of a test case debugger prompt for the circumference example shortcut used in the intervention user evaluation study.

The test-case debugger intervention makes use of the same Shortcuts parser that was used to parse the data from the r/Shortcuts subreddit. The data is then served to the intervention shortcut, which uses the initial values and implements a "Quick Look" step between each action for step-by-step debugging (similar to the breakpoint method). Since the universe of

possible actions and action combinations is almost infinite, I focused on the most-used actions for this implementation.

Since it was not possible to integrate this functionality into the Shortcuts app, a Siri shortcut was implemented. Users bring up the shortcut by saying *"Hey Siri, test my shortcut."* The shortcut shows a menu for input variables. Once the user inputs the initial variable conditions, they are presented with every "Quick Look" result, which shows the current condition of each variable after each action (Figure 4). More importantly, the shortcut skips all outwards-facing actions are skipped, which allows for repeated test-case debugging.

EVALUATING THE INTERVENTION

User Evaluation Study Design

To maintain consistency between the two studies, I invited the same group of 15 participants that participated in the initial usability evaluation study. The purpose of this user evaluation study was to understand the user's attitude towards the interventions and compare their stance to the original design, specifically when creating and using user-created shortcuts. Each participant was asked to perform the following tasks:

1. Create a Shortcut that calculates the circumference and area of a circle. The shortcut should give users an input field for the circle radius, and a list of the two available calculations. It should display the radius and the calculation result, along with the calculation time. (Note that participants were give the following hint: Circumference $= 2\pi r$, Area $= \pi r^2$, to eliminate any discrepancy between the participants' skills based on their maths background.) Use the action autosuggestions and natural language action suggestions in your process.

2. Create a Shortcut that sends an email to a specific contact. The shortcut should ask for a specific message and add a greeting at the beginning of the message based on the contact's name. For example, it should add "Hey Nathan!" at the beginning of the message if the message is sent to a contact named Nathan. Use the flowchart and test case debugger in your process.

These tasks are equivalent in difficulty but not identical to the tasks in the original user evaluation study, in order to evaluate whether users experienced an improvement in their ability to create correct shortcuts. Note that the first shortcut creation task contains specific action hints that users could use, while the second one is open-ended, to evaluate the action-discovery functionality. Users were instructed to test and debug both of their created shortcuts to evaluate the debugging functionality.

Results

Participants from all groups showed an improved ability to perform or complete some or all the tasks. Table 2 presents a concise review of which users completed each task. Compared to the initial user evaluation study, almost all participants from all groups were able to complete the first task. Most users found the creation of the emailing shortcut easier with the addition of the debugging interventions. All participants in the intermediate and advanced groups were able to complete the second task, and only one participant in the intermediate group failed to debug their shortcut.

	Number of users completing...	
Group	*Task 1*	*Task 2*
Novice group	4 (80%)	3 (60%)
Intermediate group	5 (100%)	4 (80%)
Advanced group	5 (100%)	5 (100%)

Table 2. Number (and percentage) of users in the intervention user evaluation study that correctly completed each task. Task numbers correspond to the numbered list in Usability Evaluation Tasks. Task 1: Create circle circumference/area shortcut; Task 2: Create emailing shortcut.

The following four main points came up consistently in the off-boarding interviews with participants:

1. The participants rated the action autosuggestions as significantly more relevant compared to the organic suggestions within the Shortcuts app. Four participants noted that the suggested actions helped them discover new actions.

2. The search function particularly interested the novice and intermediate participants, who were able to discover what functions were available to help them create the requested shortcuts. Advanced participants welcomed the functionality and mentioned that it would be helpful when trying to discover functionality from other apps (some apps provide additional actions besides the stock actions provided by the Shortcuts app).

3. The flowchart debugger was extremely useful for all participants, especially when their initial construction was not successful. Experienced participants in the advanced group particularly responded to this intervention, as they had direct experience with the lacking debugging functionality in the stock Shortcuts app.

4. Similar reactions were observed regarding the test case debugger. Advanced participants made better and more frequent use of the functionality, as they were already familiar with this kind of debugging strategy. Novice group participants noted the usefulness of the test case debugger, but remarked that it would make more sense to use it when they were more experienced with the app.

5. Participants in the novice group reported that they would be two times more likely to use the app in the future to augment the functionality of their devices beyond the installed applications. Intermediate and advanced group participants noted a 1.5x increase in their attitude to use the Shortcuts app or recommend it to other friends with iOS devices.

DISCUSSION

The study of the Shortcuts app is a novel endeavor that has not been attempted before and might make some wonder about its significance. However, as an app found in billions of devices but utilized by a few millions, any study to increase the user base of the app is impactful. The interventions presented in this paper achieve just that.

The results of the small intervention user evaluation study point to a general design principle: a product is as good as its discoverability. As reported in the initial user study, participants found it hard to discover all the functionality that shortcut actions provided through the current "Next Action Suggestions" panel and the search function. The discoverability interventions aid in bridging the gap, especially for novice and intermediate Shortcuts users.

At the same time, the intervention user study also highlights a tension in design. It is clear that Apple designers were aware of additional functionality they could implement, similar to the proposed debugging interventions. After all, other professional software shipped by the company feature advanced capabilities that satisfy even the most demanding users. However, the Shortcuts app is not an app for professionals. It's a stock app that comes pre-installed with every iPhone. Therefore, the app should be approachable and useful for every user. There comes, then, the hard decision to filter in enough features that make the app powerful and useful, but filter out other features that would overwhelm most users. More exploration is due here (see the "Limitations" section), but a way to satisfy both sides is to hide complexity and let advanced users opt in for it.

LIMITATIONS, FUTURE WORK AND CONCLUSION

The results of this paper confirm our research hypothesis: the Shortcuts app's utility will increase with more advanced action discovery and debugging capabilities. In companionship to the stock Shortcuts app, our intervention provide a data-driven action autosuggestion, a powerful actions search function, and a flowchart and test-case debugger. The results provide empirical evidence that these interventions increase the perceived utility of the Shortcuts app and enable more users to achieve their automation goals with the app. Implementing these changing directly into the Shortcuts app can amplify the effect of these interventions and enable more users to take full advantage of the device in their pocket.

Our implementation of the interventions was an empirically powerful way to investigate whether such additions to the Shortcut app would increase its utility, but it is obvious that future work needs to address several shortcomings. Specifically, as noted many times in the intervention design section, the intervention was not directly applied in the Shortcuts app user interface. Therefore, the effects of directly embedding and replacing features in the stock Shortcuts app needs to be investigated. Future work also needs to clarify whether the addition of these interventions adds to the cognitive load experienced, which might deter some novice users from using the Shortcuts app altogether. It is important to note that such a sentiment did not emerge from the intervention user evaluation, but a study among a larger user group should offer more insight. Last but not least, a further step in the exploration of these interventions needs to include their implementation in the Shortcuts app across Apple's platforms.

ACKNOWLEDGMENTS

I would like to personally thank Prof. Elena Glassman, who wrote and provided helpful comments throughout the ideation process of this project.

REFERENCES

[1] Apple. 2018. WWDC 2018 Keynote. Video. (2018). https://youtu.be/UThGcWBIMpU

[2] Apple. 2022. Shortcuts User Guide for iPhone and iPad. (2022). Retrieved December 20, 2022 from https://support.apple.com/guide/shortcuts/welcome/ios.

[3] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)* (2005), 363–370. http://nlp.stanford.edu/~manning/papers/gibbscrf3.pdf

[4] Reddit. 2022. r/shortcuts: Shortcuts. (2022). Retrieved December 20, 2022 from https://www.reddit.com/r/shortcuts/.

[5] RoutineHub. 2022. RoutineHub: A community for Siri Shortcuts. (2022). Retrieved December 20, 2022 from https://routinehub.co/.

[6] Daniel Thompson. 2021. Utilization of the iOS Shortcuts App to Generate a Surgical Logbook Tool: Feasibility Study. *JMIR Perioperative Medicine* 4, 1 (Jan-Jun 2021). DOI:http://dx.doi.org/10.2196/24644