

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

APPLE INC.,

Petitioner

v.

SMITH INTERFACE TECHNOLOGIES, LLC,

Patent Owner

IPR2024-01088

10,656,758

DECLARATION OF RAJEEV SURATI, Ph.D.

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
II.	PROFESSIONAL BACKGROUND	1
III.	COMPENSATION.....	10
IV.	LEGAL CONSIDERATION	10
A.	Obviousness	10
B.	Claim Interpretation	11
V.	Overview of Task & Basis for Opinions	12
VI.	Level of Skill in the Relevant Art	13
VII.	OVERVIEW OF THE '758 PATENT	14
A.	Independent Claims 1, 63, and 124.....	15
VIII.	OVERVIEW OF THE PRIOR ART	17
A.	U.S. Patent Application Publication No. 2011/0252372 (“Chaudhri”)	17
B.	U.S. Patent Application Publication No. 2004/0051726 (“Martyn”).....	24
C.	U.S. Patent Application Publication No. 2011/0248948 (“Griffin”)	29
D.	Petitioner’s Proposed Combination of Chaudhri, Martyn, and Griffin.....	33

IX. The Chaudhri-Martyn-Griffin Combination Lacks Evidentiary Support.....	35
A. A POSITA would not have combined Chaudhri with Martyn.....	35
1. Petitioner’s motivation would not have led a POSITA from Chaudhri to Martyn.....	37
2. Even if Chaudhri would have led a POSITA to Martyn, A POSITA would not have a reasonably expectation of success because the combination of Chaudhri and Martyn creates a conflict.	43
B. A POSITA would not have then combined Chaudhri-Martyn with Griffin	45
1. A POSITA would not have been motivated to combine Chaudhri and Martyn with Griffin.....	45
2. Even if a POSITA would have been motivated to combine Chaudhri and Martyn with Griffin, a POSITA would not have had a reasonable expectation of success.	47
X. Claims 1, 63, and 124 are non-obvious over Chaudhri-Martyn-Griffin	49
A. [1.h], [1.j], and [1.k]: Action-Option-Display Criteria, Action Option Objects, and Two Time Thresholds	49
B. [1.h]: Criteria Evaluation	52

C.	[1.j]: Action-Option-Display Criteria.....	53
D.	[1.k]: Operation Menu.....	54
XI.	CONCLUSION	56

I. INTRODUCTION

1. My name is Dr. Rajeev Surati, Ph.D. I have been retained by Smith Interfaces Technologies, LLC as an independent expert consultant in this inter partes review (“IPR”) proceeding before the United States Patent and Trademark Office (“PTO”).

II. PROFESSIONAL BACKGROUND

2. All of my opinions stated in this Declaration are based on my own personal knowledge and professional judgment. In forming my opinions, I have relied on my knowledge and experience in designing, developing, and researching the technology referenced in this Declaration.

3. I am over 18 years of age and, if I am called upon to do so, I would be competent to testify as to the matters set forth herein. I understand that a copy of my current curriculum vitae, which details my education and professional and academic experience, is being submitted as EX2002. The following provides a brief overview of some of my experience that is relevant to the matters set forth in this Declaration.

4. I hold a Doctor of Philosophy (Ph.D.) degree in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology (“MIT”) (awarded in 1999) with a Grade Point Average of 5.0/5.0. I obtained a Master of Science degree in Electrical Engineering and Computer Science from MIT (awarded

in 1995) with a Grade Point Average of 5.0/5.0. I have a Bachelor of Science degree in Electrical Engineering from MIT (awarded 1992), graduating with a Grade Point Average of 4.9/5.0.

5. My Ph.D. thesis was entitled “Scalable Self-Calibrating Technology for Large Scale Displays.” My master’s thesis was entitled “Practical Partial Evaluation.” My undergraduate thesis, for which I received MIT EECS’s William A. Martin thesis prize for best undergraduate thesis, was entitled “A Parallelizing Compiler based on Partial Evaluation.” I was also awarded the Department of Energy’s highly-selective Computational Science Fellowship in 1995, which funded my Ph.D. studies.

6. In 1989, I began programming embedded systems and, in 1995, built one of the first publicly available Internet Coke Machines. During that time, I worked on various user interfaces. For example, one to augment the coke machine to dispense coke and another that became one of the early ecommerce mechanisms, using standards-based protocols for tracking user consumption, payments, and drink stock.

7. In 1997, while at MIT pursuing my Ph.D., I started a company called Flash Communications. At Flash, I invented an instant messaging platform focused on enterprise needs. I had the idea while observing the popularity of the Zephyr Instant Messaging Service at MIT beginning in 1988. Attracted by Flash’s

enterprise focus, Microsoft acquired Flash in 1998. With Microsoft, we built both a client and server product. The basic protocol we invented became the foundation for the well-known XMPP protocol that was widely used in the mid-2000s among instant messaging providers. I developed both the client and server products and dealt with many, if not all, of the issues one might face when implementing user interfaces for instant messaging systems. In fact, we helped to build Microsoft's famous MSN product, which adopted various user interface elements we created.

8. As a result of this experience, I became heavily involved with User Interface design on desktop machines, experience that I have continued to develop by building desktop, mobile, and web-based applications used by millions of people.

9. After graduating, I joined Microsoft (as was required by Microsoft in its acquisition of Flash Communications), where I worked on both client and server technologies related to instant messaging. My work spanned both the Microsoft Exchange Instant Messaging product that was released in 2000 and MSN Messenger.

10. From 2000 to 2007, I cofounded, led, and sold a business called photo.net to Namemedia, which is now part of GoDaddy.com. In early 2000, Photo.net was considered a top 1,000 website and it received large amounts of traffic. I built the site from its beginning as a single computer installed in a datacenter to a full rack of computers in that same data center. As a result, I became very familiar with the careful design and programming one must employ to build

and maintain systems that enable both engaging user interfaces and experiences at scale.

11. At photo.net, we prototyped many fundamental Internet community features, such as photo sharing, social networking, and memberships. The photo.net system was written on top of Oracle SQL Database and served up many millions of web pages each day (corresponding to millions of records) with high-volume inserts. The system also incorporated an ecommerce platform used to bill subscriptions, track users, etc. I spent seven years running the site, both writing new features driven by the Oracle SQL database and maintaining tables containing the many millions of records generated by the site. I also served as the database administrator, which provided me with extensive operational experience running such a system. I dealt with everything from database query speedup to the more mundane day-to-day issues, such as the maintenance of relational databases, including backups. In addition to the web interface, we also ran a mobile version of the product based on WAP, which I helped to develop.

12. After developing one of the first photo sharing sites, we further increased user engagement by designing an innovative photo rating interface. Our photo rating system became a huge draw for the site. The rating system increased engagement by four orders of magnitude, ultimately resulting in Sergey Brin, cofounder of Google, asking photo.net to provide imagery to Google for a screen

saver application. Joichi Ito, former head of the MIT Media Lab and a famous internet luminary called it “the best photo sharing site” in 2001.

13. I also innovated user interface technology at Nexaweb, where I was an investor and technical advisor. Nexaweb designed User Interfaces that worked well over the web. At Nexaweb, I spent a great deal of time solving underlying issues related to implementing a user interface framework both in Java and Javascript. We also innovated a mechanism using server push prior to the AJAX becoming widely adopted.

14. In 2004, I started Scalable Display Technologies based on the technology in my Ph.D. thesis and the related patent (U.S. Patent No. 6,456,339) entitled “Super-resolution Display.” As part of my Ph.D. thesis, I designed a special display that allows one to create ultra-high-resolution displays composed of multiple projectors tiled with a slight overlap. The system uses camera-based feedback to create an inverse map that drives the system such that a person can only see a continuous, seamless display. There is no visible bezel or overlap. During this project, I became familiar with how and why to drive content onto the display. Thus, I also became familiar with content encoding/decoding, user interfaces for driving large displays, multimedia content storage, high resolution imagery, networks, recording, GPUs, storage of content, etc.

15. At Scalable Display Technologies, I developed the core algorithms and implemented—with a team—the entire software system, which included the core user interface, machine vision and embedded components, and a multimedia playback system. Our first customer was Pixar. Today, the software is primarily used by the US military and commercial theme parks, such as Disney and Universal. Scalable is well-known in the industry for having some of the easiest and intuitive software to use.

16. In parallel with my work at Scalable, I lectured at MIT on many subjects, including the Android operating system. In 2008, I worked with a group of MIT students on the development of mobile applications for Android with Rich Miner, a cofounder of Android. Since many students created applications involving Google Maps and localization, I became familiar with these technologies. I also served as a lecturer and mentor at MIT Play Labs, which was an incubator for augmented reality (AR) and virtual reality (VR) software for mobile handsets and headset applications. Notably, in addition to mobile handsets, Android runs on the Oculus headset. As part of that program, I worked with several startup companies on mobile applications, including a startup that developed applications for Telegram, which is a cross-platform, cloud-based instant messaging system.

17. Since 2014, I have worked as an independent consultant for several companies including NEC, Hitachi, Hi Marley, and Estee Lauder.

18. In 2018, I became a senior partner at nCent Labs. In that role, I consulted on the development of an incentive-market-based platform for block chains and cryptocurrency. Another part of my work at nCent Labs focused on the development of SMS messaging applications for the nCent platform. In that part, I developed a full stack application around the protocol.

19. From 2019 to 2020, I served as a Technical Lead of the Skunkworks at Hydrow, which is a startup company that develops indoor rowing machines. In this role, I worked on special projects, including the development of a VR experience using Magic Leap and Oculus that immersed users in a team-based rowing crew on a scull using VR cinematography. I leveraged my experience with user interfaces and Human-Computer Interaction (“HCI”) to develop a system that gives people a sense of rowing on the water with others, while in reality, they were rowing on a rowing machine. We accomplished our goal through a clever use of cinematography. The result was quite compelling and increased user engagement with their exercise program.

20. In 2020, I started a company called Skyline Nav AI Inc. that develops technology that uses a visual location (i.e., the skyline) to geo-locate the place a picture was taken. At Skyline, I built and now maintain a full user experience centered around a workflow for people to snap a picture and use the image’s skyline and downloaded 3D mode to localize a person. I have written applications for both

Android and iOS that communicate with a backend server running the machine vision algorithms.

21. Over the past decade, I have served as a technical consultant and expert witness on patent litigation matters. In the course of this work, I have provided consulting services to a wide variety of technology companies, including BritishTelecom, Apple, IBM, Philips, Shopify, Zillow, Polaris Powered Technologies, Amazon, Salesforce, Hitachi, Slack, Harris Teeter, and others.

22. During the course of my career, I was the inventor of the subject matter for approximately 15 U.S. Patents. I have also received additional patents, including: U.S. Patent No. 8,817,111, entitled “System and method of calibrating a display system free of variation in system input resolution,” U.S. Patent No. 8,994,757, “System and method for providing improved display quality by display adjustment and image processing using optical feedback,” U.S. Patent No. 9,215,455, “System and method of calibrating a display system free of variation in system input resolution,” U.S. Patent No. 9,369,683, “System and method for calibrating a display system using manual and semi-manual techniques,” and U.S. Patent No. 9,497,447, “System and method for color and intensity calibrating of a display system for practical usage.”

23. I am also a co-inventor of patented technology related to instant messaging, including U.S. Patent No. 5,943,478, which describes associated

technology that I developed related to pop-up, two-way messaging over the Internet. While at Microsoft, I was an inventor on several patents, including: U.S. Patent No. 6,260,148 relating to methods and systems for message forwarding and property notifications using electronic subscriptions; and U.S. Patent Nos. 6,415,318 and 6,604,133 relating to inter-enterprise messaging systems using bridgehead servers. Aspects of these patents relate specifically to messaging and notification technology in telecommunications systems.

24. I serve on the advisory boards of several technology companies, including Paneve, which develops general purpose ASICs coupled with compiler technology; Nexaweb, which develops real-time web application frameworks using HTTPS; Antix Labs, which develops compiler technology for a universal gaming platform; and Permabit, which develops content-addressable storage.

25. In addition to my above experience, I have published papers on subjects relating to computing systems, computer network communications, databases, and other subjects within the realm of electrical and computer engineering. I have also received several awards for my contributions as an inventor and entrepreneur, including the Global Indus. Technovator Award 2009 and Laureate of 2009 Computer World Honors Program. My publication and award history are listed in more detail on my CV.

III. COMPENSATION

26. My work on this matter is being billed at my customary rate of \$700 an hour. Also, I am being reimbursed for reasonable expenses I incur in relation to my services. I have no pecuniary interest in the outcome of this proceeding. I understand I will be paid regardless of the outcome of any proceeding in which my work is used.

IV. LEGAL CONSIDERATION

27. I am not an attorney. My understanding of the law is based on information provided by counsel for Smith Interface Technologies, LLC.

A. Obviousness

28. I understand that a claimed invention is obvious and, therefore, not patentable if the subject matter claimed would have been obvious to a person of ordinary skill in the art at the time that the invention of the '758 Patent (a "POSITA"), which I have been asked to treat as August 5, 2011.

29. I understand that a claim can be obvious in view of a single prior art reference (e.g., via modification of that prior art reference) or multiple prior art references (e.g., via a combination of two or more prior art references), if such a modification or combination was within the skill of a POSITA. I understand that there must be some articulated reasoning with some rational underpinning to support a conclusion of obviousness. I also understand that to establish a finding of obviousness one must show that one of ordinary skill in the art would have had a

motivation to combine the prior art references to produce the claimed invention and a reasonable expectation that the combination would be successful.

30. I further understand that exemplary rationales that may support a conclusion of obviousness include: (1) simply arranging old elements in a way in which each element performs the same function it was known to perform, and the arrangement yields expected results, (2) merely substituting one element for another known element in the field, if the substitution yields no more than a predictable result, (3) combining elements in a way that was “obvious to try” because of a design need or market pressure, where there was a finite number of identified, predictable solutions, (4) that design incentives or other market forces in a field would have prompted variations in a work that were predictable to a person of ordinary skill in the art, and (5) that there was some teaching, suggestion, or motivation in the prior art that would have led a POSITA to modify or combine prior art references to arrive at the claimed invention.

B. Claim Interpretation

31. I understand that a claim term is interpreted according to its ordinary and customary meaning as a POSITA would have understood the term in light of the surrounding claim language, other claims, the specification, and the patent’s prosecution history, which are referred to as intrinsic evidence. I also understand that prior art references cited in the patent’s prosecution history are considered

intrinsic evidence. I further understand that evidence outside the patent and its prosecution history (e.g., dictionaries and technical articles), may inform the context in which a POSITA would have understood the claims of a patent. I understand this ordinary and customary meaning applies absent unique circumstances, such as where a patent clearly expresses an intent to set forth a special meaning for a term.

V. Overview of Task & Basis for Opinions

32. I have been asked to review the '758 Patent. I have been asked to provide opinions related to certain issues from the perspective of a person of ordinary skill, having knowledge of the relevant art, as of August 5, 2011, and—except where otherwise noted—the opinions stated in this declaration are from that perspective. The qualifications and abilities of such a person are described in paragraph 38 below.

33. More particularly, I have been informed that this proceeding is currently at a preliminary stage during which the Patent Trial and Appeal Board (the Board) will decide whether the Petition establishes a reasonable likelihood of at least one claim of the '758 Patent being invalid. I also understand that the Board may review the Petition under a stricter “compelling merits” standard that requires the Petition to make a stronger showing of invalidity; however, my analysis is not performed using this higher standard. At this preliminary stage, I have been asked to focus my analysis on the Petition’s proposed invalidity theories for the '758 Patent’s independent claims 1, 63, and 124.

34. My opinions are based on my education, training, and experience as well as items that I reviewed to prepare my opinions, including the Petition, the documents listed in the Exhibit List included with the Petition, and any other items that I reference in my below analysis.

35. I understand that other issues may arise that require further explanation, and I will provide that explanation if appropriate. As a result, I respectfully reserve the right to update and supplement this declaration and the information and opinions provided herein.

VI. Level of Skill in the Relevant Art

36. I understand that a patent must be analyzed from the perspective of a person of ordinary skill in the art (“POSITA”) as of the time of invention of the patent, which is typically considered to be the patent’s earliest filing date. While I do not necessarily agree with every aspect of Dr. Cockburn’s explanation of the level of ordinary skill in the art, I do not currently believe the aspects of that explanation I would change or clarify would impact the analysis set forth in this declaration. For purposes of this declaration, my opinions are therefore provided from the perspective of the POSITA described by Dr. Cockburn’s declaration (EX1003) as of August 5, 2011, which is produced below:

“a Bachelor of Science degree in computer science, computer engineering, or a comparable field and at least two years of professional

experience working with graphical user interfaces and applications for computers. Such experience could be obtained through research and study in a graduate program or through comparable exposure through industry employment, and additional years of experience could substitute for the advanced-level degree.” EX1003, ¶39.

VII. OVERVIEW OF THE ’758 PATENT

37. The ’758 Patent discloses “touch screens,” where a “user experience may be altered, utilizing [touch] signal[s].” EX1001, 1:39-40, 17:3-4.

38. The ’758 patent enable a user to access different device functionalities for varying touch sequences. EX1001, 1:44-4:33.

39. When operating a touch-screen device users are frequently presented with a screen to begin their navigation, the ’758 patent describes certain sequences that begin with a user facing an “application launching user interface.” EX1001, 1:49-50, 2:13-15, 2:46-51, 3:1-5, 3:33-42, 3:54-58, 4:20-25. The ’758 patent describes four different single finger touch sequences to reach unique functionalities of the device when faced with this screen.

40. In the first sequence, a user may tap on an application to launch it. EX1001, 97:49-64.

41. In the second sequence, a user may touch-and-hold an application to launch a menu displaying information related to that application. EX1001, 97:65-98:12.

42. In the third sequence, a user may touch-and-hold for a longer duration to access operations associated with the application, such as the option to delete the application. EX1001, 101:33-35.

43. In the fourth sequence, a user may relocate the selected application after touching-and-holding. EX1001, 56:62-67.

A. Independent Claims 1, 63, and 124

44. The '758 Patent has three independent Claims 1, 63, and 124.

45. Claim 1 is directed to an “electronic device.” EX1001, 97:29.

46. Claim 63 is directed to a “method.” EX1001, 108:41.

47. Claim 124 is directed to a “non-transitory computer readable storage medium.” EX1001, 114:60.

48. Claims 1, 63, and 124 are substantially similar.

49. Claims 1, 63, and 124 each recite “an application launching user interface that includes a plurality of application icons for launching corresponding applications.” Limitations [1.f], [63.b], [124.a].

50. Claim 1 recites “while displaying application launching user interface, detecting a first single-finger touch input” “at a location on the touch-sensitive

surface that corresponds to a[n] ... application icon.” Limitations [1.g]. And Claims 63 and 124 then recite “with the application launching user interface displayed, detecting a first single-finger touch input” “at a location on the touch-sensitive surface that corresponds to a[n] ... application icon.” Limitations [63.c], [124.b].

51. Claims 1, 63, and 124 each recite “in response to detecting the first single-finger touch” “determining a response based on evaluating the” “duration or “movement” of the “touch input against” one of four criteria. Limitations [1.h], [63.d], [124.c]. These four criteria are an “application-launch criteria,” an “action-option-display criteria,” an “operation criteria,” or “a movement criteria.” Limitations [1.h], [63.d], [124.c].

52. The “application-launch criteria” is met when the duration of the touch is “less than a first time threshold.” Limitations [1.i], [63.e], [124.d].

53. The “action-option-display criteria” is met when the duration of the touch is “greater than the first time threshold.” Limitations [1.j], [63.f], [124.e].

54. The “operation criteria” is met when the duration of the touch is “greater than a second time threshold that is greater than the first time threshold.” Limitations [1.k], [63.g], [124.f].

55. Claim 1 recites that “in accordance with a determination that the first single-finger touch input meets” one of the four criteria then one of the following sequences happens. Limitations [1.i-l]. Claims 63 and 124 recite that “based on a

determination that the first single-finger touch input meets” one of the four criteria then one of the following sequences happens. Limitations [63.e-h], [124.d-g].

56. If the “application-launch criteria” is met, then the application is “launch[ed].” Limitations [1.i], [63.e], [124.d].

57. If the “action-option-display criteria” is met, then “one or more action option objects associated with the first application without launching the first application” are “display[ed].” Limitations [1.j], [63.f], [124.e].

58. If the “operation criteria” is met, then “an operation in connection with the first application icon” is “perform[ed].” Limitations [1.k], [63.g], [124.f].

59. If the “movement criteria” is met then the “application” is “mov[ed]” “in a foreground virtual display layer so that the first application icon appears to float.” Limitations [1.l], [63.h], [124.g].

VIII. OVERVIEW OF THE PRIOR ART

A. U.S. Patent Application Publication No. 2011/0252372 (“Chaudhri”)

60. U.S. Patent Application Publication No. 2011/0252372 is titled “Device, Method, and Graphical User Interface for Managing Folders” and lists as its inventor Imran Chaudhri. I will refer to this reference as “Chaudhri.”

61. Chaudhri discloses two modes for user input on a touch screen device. The first mode is the “normal operation mode,” which refers to the standard interface that a user would see when unlocking a touch screen device. EX1004, [0217]. The

second mode is the “user interface reconfiguration mode.” EX1004, [0218]. This mode is designed to provide a user enhanced functionalities for rearranging applications on a touch screen device.

62. In the “normal mode, tapping on an application will launch the application. EX1004, [0217]. And when in “user interface reconfiguration mode,” a user is able to rearrange applications and can reach a “pop-up dialogue” by touching a separate “object modification target.” EX1004, [0274].

63. The pop-up dialogue allows a user to confirm actions associated with rearranging application such as creating a folder, naming a folder, or deleting/removing an application from a folder. EX1004, The former of these three are referred to as a “name confirmation dialogue” box. EX1004, [0232]. Figures. 5P-5Q, 5LLL. Figures 5P, 5Q, and 5LL are inserted below.

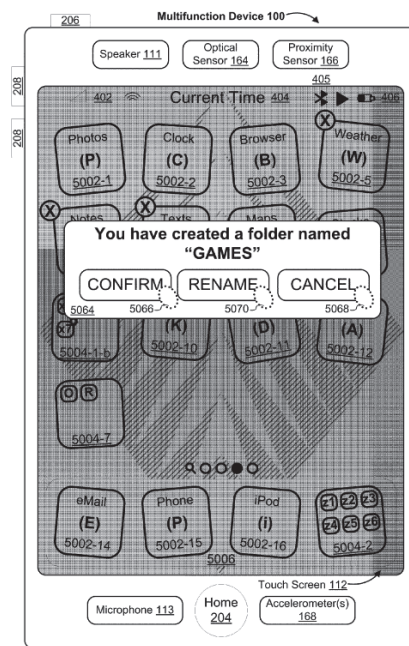


Figure 5P

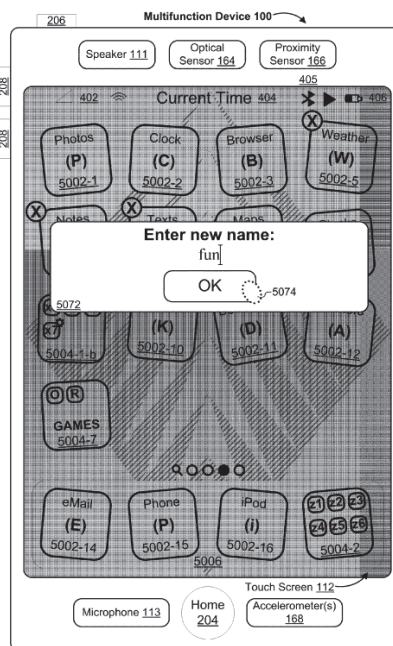


Figure 5Q

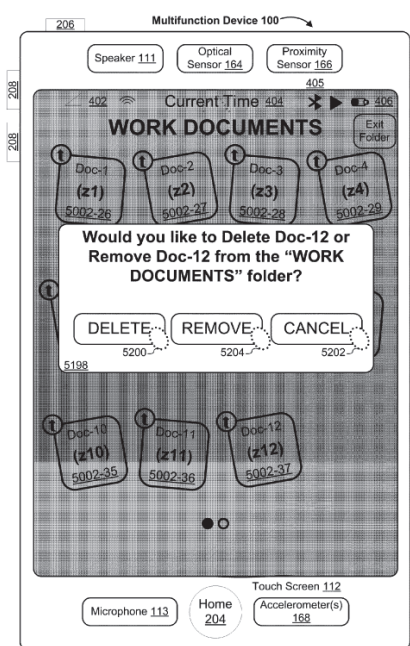


Figure 5LLL

64. Chaudhri is directed to “electronic devices with touch-sensitive surfaces, including but not limited to electronic devices with touch-sensitive surfaces for managing folders.” EX1004, [0003]. But Chaudhri’s disclosures are not limited to touch input. Other input methods disclosed in Chaudhri include “physical buttons (e.g., push buttons, rocker buttons, etc.), dials, slider switches, joysticks, click wheels, and so forth.” EX1004, [0085]. The click wheel may be “physical or virtual.” EX1004, [0093]. The “one or more physical buttons, [may be a] ‘home’ or menu button.” EX1004, [0175].

65. At the time of invention, it was not uncommon for users to have buttons on a touch screen device. The presence of a touch screen on a device did not eliminate the need or desirability for buttons to also be present. Buttons would be familiar to users and allow users to access enhanced functionalities while interacting with the touch screen.

66. Chaudhri also highlights how “touch-sensitive surfaces” “are widely used to manage folders by manipulating selectable user interface objects on a display.” EX1004, [0004].

67. Chaudhri is primarily concerned with manipulating objects for folder management. Chaudhri emphasizes that “[e]xemplary manipulations include creating a folder, displaying a folder view associated with a folder, adding ... icons, document icons, folder icons, etc.[.] to a folder, removing ... objects from a folder,

repositioning ... objects within a folder view of a folder, repositioning a folder icon ... and deleting a folder.” EX1004, [0005].

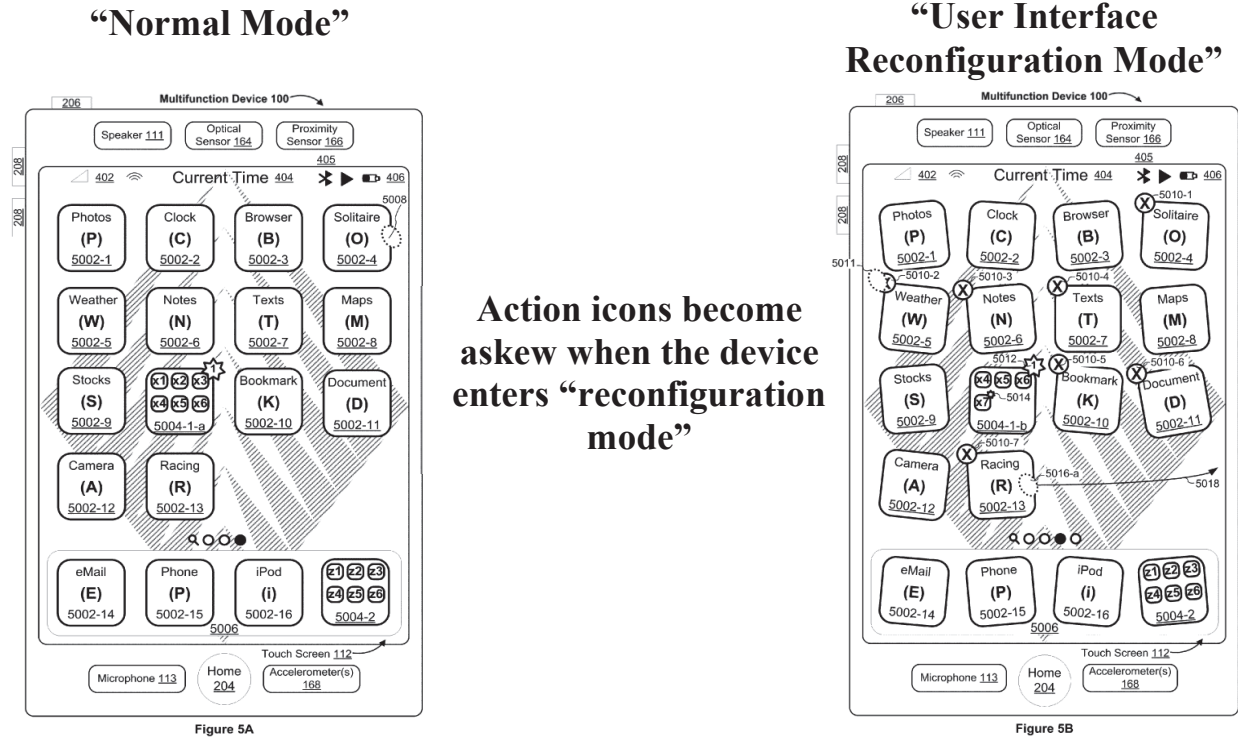
68. In light of this need, Chaudhri notes that “existing methods” “are cumbersome and inefficient.” EX1004, [0006]. Chaudhri gives the example that “a sequence of inputs to create, modify and/or delete folders and content within folders is tedious and creates a significant cognitive burden on a user.” EX1004, [0006].

69. To facilitate “more efficient methods and interfaces for managing folders” EX1004, [0007], Chaudhri discloses “user interfaces for computing devices with touch-sensitive surfaces” EX1004, [0008].

70. As shown below, Figures 5A and 5B illustrates a user interface page where “the device detects a contact 5008 with the solitaire application icon 5002-4 for more than a predetermined period of time (e.g., 2 seconds) and, in response, the device enters a user interface reconfiguration mode.” EX1004, [0217]. Once “the device has entered a user interface reconfiguration mode” “the selectable user interface objects jiggle as though they are floating.” EX10004, [0218]. The device is in “user interface reconfiguration mode” when “each respective selectable user interface object oscillates about a respective average position of the selectable user interface object on the display” EX1004, [0218].

71. A user would see this jiggling when operating the device. But movement is difficult to depict on still images, so to communicate this Chaudhri

depicts the applications as askew when the device is in user interface reconfiguration mode.



72. In some embodiments, while the device is in “user interface reconfiguration mode” “an object removal badge (e.g., 5010 in FIG. 5B)” appears and “when the device detects activation of the object removal badge (e.g., tap gesture 5011 in FIG. 5B), the selectable user interface object (e.g., weather application icon 5002-5 in FIG. 5B) associated with the object removal badge is removed from the user interface (e.g., deleted).” EX1004, [0218].

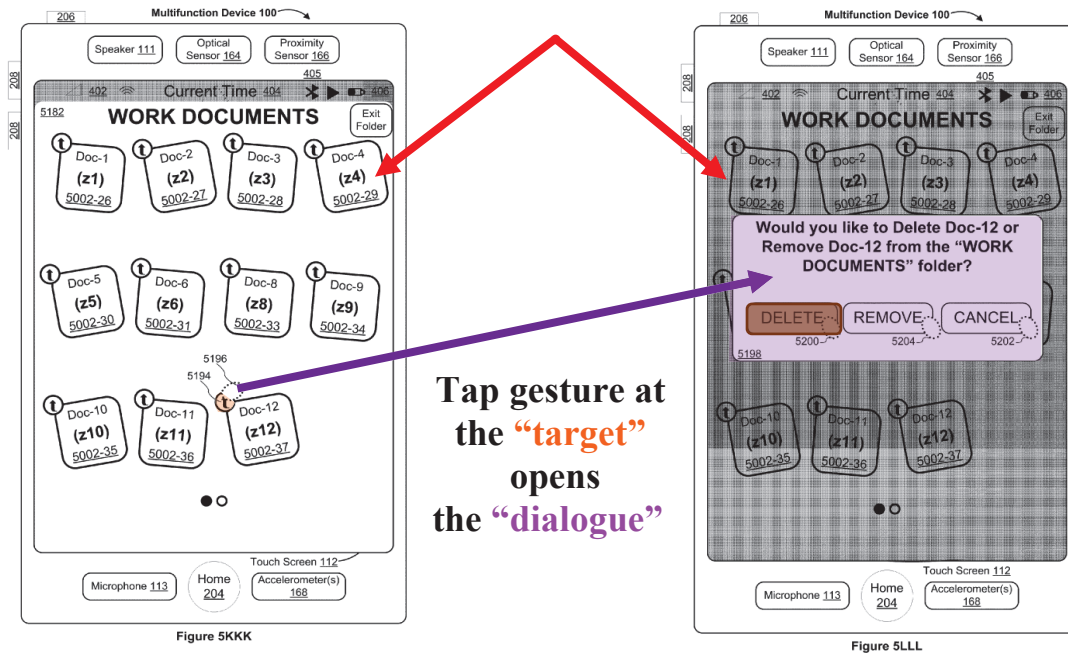
73. This selectable element used in the management of applications is further disclosed “[i]n some embodiments” as “an object modification target associated with one or more of the selectable user interface objects.” EX1004,

[0273]. This object modification target is also only displayed while in “user interface reconfiguration mode.” EX1004, [0274].

74. While in “user interface reconfiguration mode” the device may “detect[] a request to activate the **object modification target [5194]** for a respective selectable user interface object (e.g., tap gesture 5196 at a location on the touch screen 112 that corresponds to a location of an object modification region for action icon 5002-37), the device displays an object modification dialogue. For example in FIG. 5LLL the device displays a **pop-up dialogue 5198** that provides the user with options for modifying the selectable user interface object.” EX1004, [0274]. And then a user is able to delete the “action icon 5002-37” by a “tap gesture **5200 on a delete button.**” EX1004, [0274].

75. The device remains in “user interface reconfiguration mode” throughout this entire process as shown in Figures 5KKK and 5LLL by the askew icons.

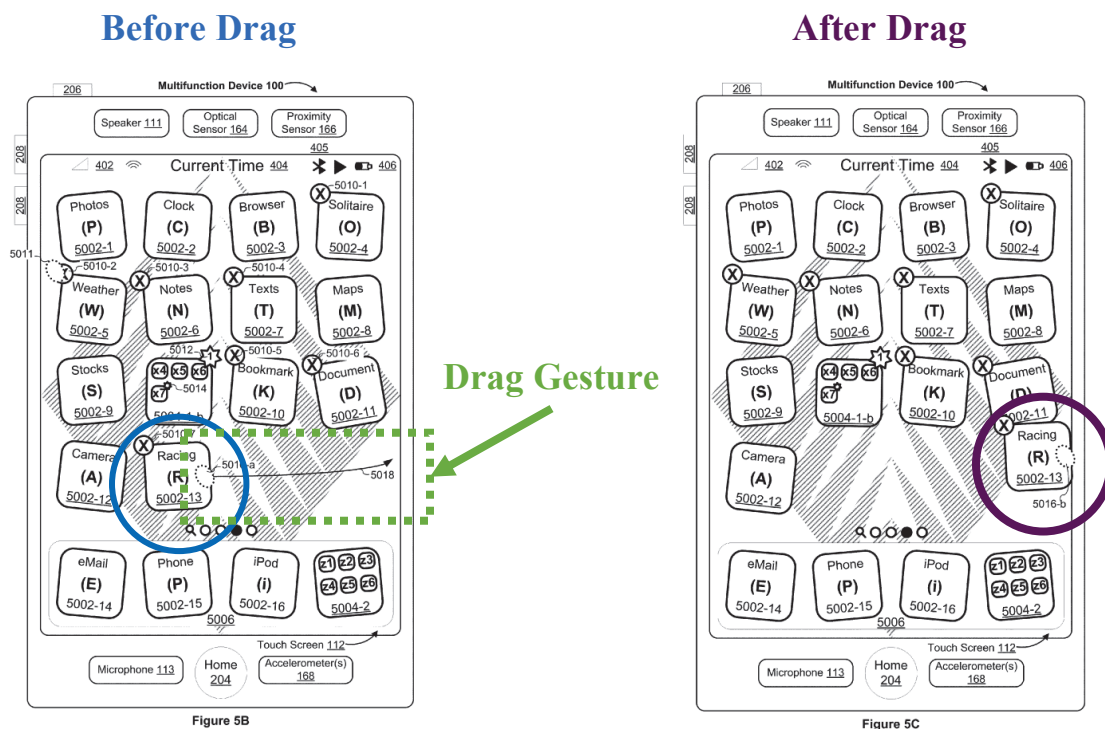
**Device remains in “reconfiguration mode”
Askew actions icons**



76. Chaudhri also discloses rearranging applications using a drag gesture when in “user interface reconfiguration mode.” EX1004, [0101], [0217-0220], [0273], Figures, 5B-C, 9A-9B, 5KK-5MM.

77. The device in Chaudhri can “detect[] a request to move a” “user interface object to an edge of the screen.” EX1004, [0220]. The request is initiated by “a contact 5016-a on the touch-sensitive surface” at the location of the ““racing’ action icon 5002-13” where a “subsequent movement 5018 of the contact to the edge of the touch-sensitive surface” relocates the action icon 5002-13 as shown in Figure 5C. EX1004, [0220].

78. This drag gesture is shown in Figures 5B and 5C, which “illustrate exemplary user interfaces for creating a new folder.” EX1004, [0220].



B. U.S. Patent Application Publication No. 2004/0051726 (“Martyn”)

79. U.S. Patent Application Publication No. 2004/0051726 is titled “Computing device with improved user interface for applications” and lists as its inventor Mathieu Martyn. I will refer to this reference as “Martyn.”

80. Martyn discloses a “‘snap-shot’ view.” Martyn also uses the descriptors “App Snapshot,” “summary window,” and “snapshot view” to describe this view. EX1005, [0009-0010], [0015], [0017-0026], [0028-0031].

81. In 2004, at the time of Martyn’s disclosure, few phones with touch screens were commercially available. Those phones primarily were equipped with

resistive touch and used a stylus. This is different than the capacitive touch of later model phones that enabled users to effectively and efficiently use a finger touch.

82. The word touch is not used in Martyn. Martyn does not disclose a touch-screen device.

83. The “‘snap-shot’ view ... brings together, in one summary window, a limited list of common functions and commonly accessed stored data which itself can be reached directly from the main menu listing some or all applications.” EX1005, [0010]. This summary window addresses the need for “user interface[s] ... to demonstrate or make explicit the changing internal status of [a] mobile telephone as navigation proceeds.” EX1005, [0008]. “For example, to select or initiate a function ... a user has to understand (a) how to navigate to that function in order to select that function and (b) that the status of the telephone is such that the function can be selected or initiated.” EX1005, [0008].

84. For a user to successfully navigate a user interface, the user interface must accurately reflect what operations the device will perform given a certain set of user inputs. For example, if a device is programmed to launch an application when given the option to delete the application, then the user interface is non- functional.

85. Similarly, a user interface will also be non-functional when the user input does not reach a desired result. This failure to function can be because the

device is not programmed or designed properly. For example, a user interface can fail to function when the same input is associated with two different outputs.

86. To reach the summary window, a user starts by looking at device in an “application launcher view.” EX1005, [0016]. “The Application Launcher view is ... presented as a standard scrolling list of application[s].” EX1005, [0016]. “The list is vertical and only one application is presented per line.”

87. Navigating this view is accomplished by “[s]tandard highlight functions.” EX1005, [0016]. A “highlight” indicates which application the user currently has the choice of selecting. *Id.* This is shown below as annotated from Martyn’s Figure 1.

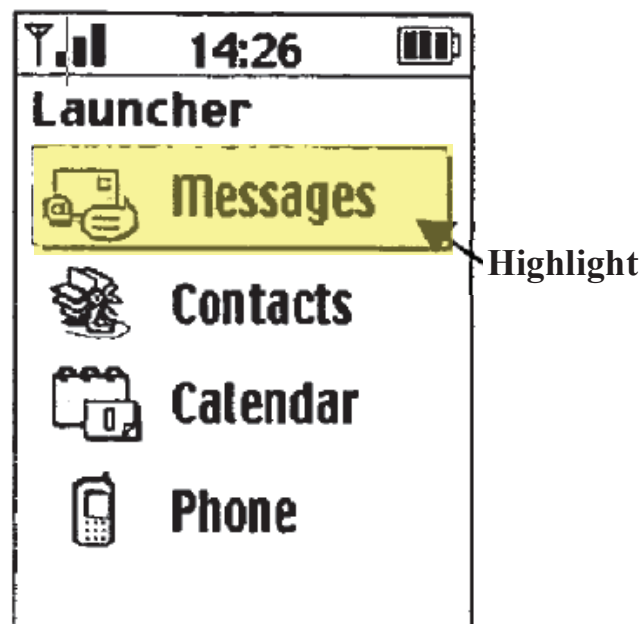


Figure 1

88. A highlight function is a method to communicate to the user what operation the device will perform given the user input. For example, a properly functioning highlight will communicate to the user that the device is programmed allow a user launch an application via further input when a visual indication overlays that application.

89. Martyn discloses that a “user may take conventional navigation steps, such as scrolling, to move the highlight and using the available select function to navigate to the required highlighted application.” EX1005, [0016].

90. Martyn does not disclose a touch screen device. Accordingly, a POSITA would have understood that Martyn was referring to some other input method to accomplish scrolling. Martyn discusses that “...with conventional user interfaces, a user may need to scroll around...” EX1005, [0005]. Additionally, given that the application was filed in 2001 and the images figures depict a user interface similar to a Blackberry device of the time period a POSITA would have understood that this scrolling was likely accomplished via a physical wheel on the device of Martyn.

91. Martyn discloses that the way to access “[t]he innovative summary window” is “as follows: should the highlight rest on the name of an application in the App Launcher for a certain amount of time (say a 1.2 second timeout), the

summary window (the ‘App Snapshot’) drops down from the highlight bar.”

EX1005, [0017]. This is shown below in Martyn’s Figure 2 with annotations added.

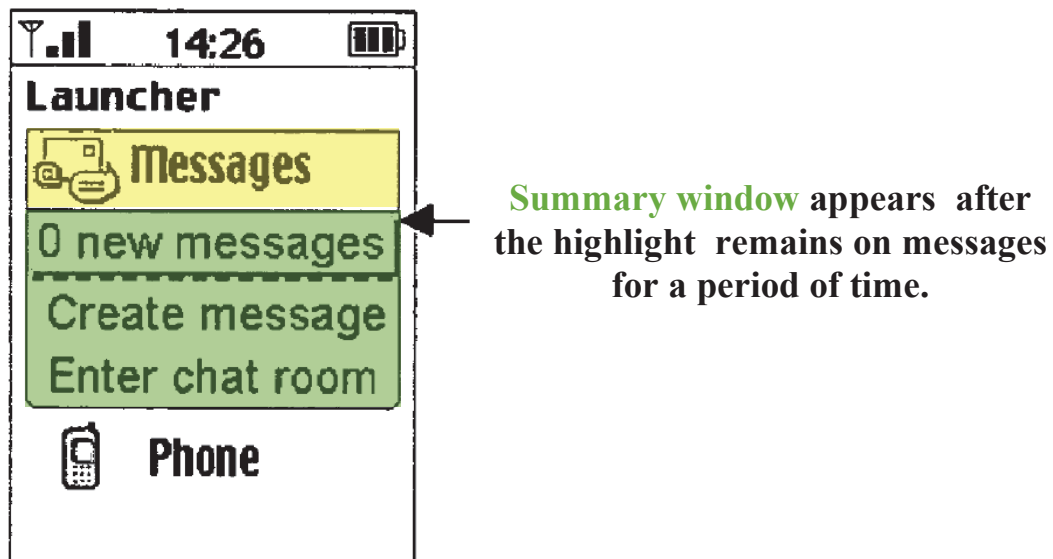


Figure 2

92. A timeout is where user input ceases for a period of time. In this case the user scrolls the highlight to the desired application and then ceases user input in order to access the summary window. *See* EX1004, [0017].

93. Martyn further describes that in addition to a timeout method. “Other selection processes could also be used (e.g. voice activation, softkey selection etc) to access the App Snapshot.” EX1004, [0017]. Additionally, “the App Snapshot may be ‘called’ ... by using a right scroll function ...; a press and hold of the select function when an application name is highlighted; or using a right cursor key on a highlighted application name.” EX1004, [0017].

94. In addition to describing how to call the summary window, Martyn discloses ways to dismiss the summary window. EX1005 [0020]. Martyn connects dismissal method with the calling methods. EX1005, [0021-0022]. “If a timeout is used to call the App Snapshot, scrolling the highlight back to the relevant application name” or “[s]crolling down through the App Snapshot” “removes the App Snapshot.” EX1005, [0021]. “If scroll functions (e.g. right scroll) call the App Snapshot the opposite scroll function removes the snapshot. EX1005, [0022].

C. U.S. Patent Application Publication No. 2011/0248948 (“Griffin”)

95. U.S. Patent Application Publication No. 2004/0051726 is titled “Touch-sensitive device and method of control” and lists as its first inventor Jason Tyler Griffin. I will refer to this reference as “Griffin.”

96. A common problem with touch screen devices is inadvertent user input. A device must discern between intended and inadvertent user input. To make the determination, a device must monitor user input.

97. Griffin primarily addresses the issue of inadvertent user input. Griffin primarily relates to “a method and apparatus that reduces the occurrence of inadvertent touches causing functions to be performed.” EX1006, [0061].

98. Griffin states: “Multiple touch thresholds may be applicable for a touch, and different values of touch thresholds may be associated with different functions or input.” EX1006, [0032].

99. To address the issue of inadvertent input, Griffin “describes an apparatus for and method of modifying a touch threshold.” EX1006, [0014]. Modifying touch thresholds “reduces the occurrence of inadvertent touches causing functions to be performed, such as entering characters, calling a contact, or sending text messages, which may needlessly drain the battery of a portable electronic device.” EX1006, [0061]. To further conserve precious battery, the device “may optionally be suspended until inadvertent touches are no longer detected.” EX 1006, [0043].

100. Griffin chooses to address the problem of suppressing inadvertent user inputs by using “multiple thresholds.” EX1006, [0061]. Having multiple types of threshold “provides a mechanism by which multiple different functions may be triggered by touch alone.” EX1006, [0061]. These multiple types of threshold provide the necessary context a device needs to ascertain whether a touch was inadvertent.

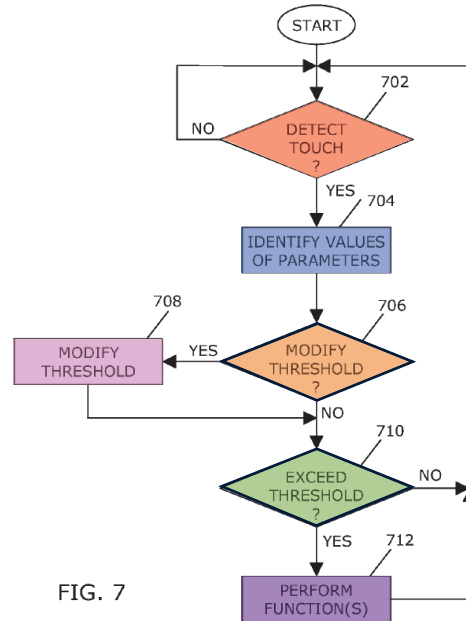
101. Griffin describes “[a] touch threshold [as] a value associated with a touch, such that when a value of a touch meets the touch threshold, a function is performed.” EX1006, [0032]. A touch threshold can be associated with “a force imparted by a touch,” “a displacement distance,” “an area of contact,” “a time duration,” “or any combination thereof.” EX1006, [0032].

102. Griffin also notes that “[b]aseline thresholds may be established, for example, to reduce inadvertent input.” EX1006, [0033]. “For example, a touch threshold related to contact area for a touch may be set to disregard touches individually or cumulatively larger than a given area that may suggest normal operation is not occurring.” EX1006, [0033]. “The device 100 collects data from the touches to establish at least one baseline for one or more touch thresholds.” EX1006, [0033].

103. To discern inadvertent from intended touches Griffin discloses that “touch threshold[s] may be modified at any time based on one or more parameter or parameters.” EX1006, [0059]. “Modifying a touch threshold involves changing the value of one or more characteristics associated with the touch threshold.” EX1006, [0042].

104. Griffin’s Figure 7 discloses a method for “modifying a touch threshold” as shown below. As seen in the “flowchart” several steps occur before a determination is made regarding whether a threshold is exceeded. EX1006, [0039]. First, “touch is detected at [red] 702.” EX1006, [0040]. Second, “values of one or more various parameters are identified at [blue] 704.” EX1006, [0040]. Third, “[a] determination whether to modify one or more touch threshold is made at [orange] 706.” EX1006, [0040]. Only then are “[o]ne or more values of one or more parameters or characteristics of the touch are compared to the touch threshold at

[green] 710” and “one or more functions, ..., are performed [purple] 712” EX1006, [0053]. Relevant touch thresholds are modified at [pink] 708.” EX1006, [0041].



105. Griffin provides “several examples of when and how one or more touch thresholds may be modified.” EX1006, [0042]. For example, “[t]he touch threshold may be modified based on the rate of detected touches” because “when the touch rate is high, the touch threshold may be decreased, to reduce the user's effort,” and “[w]hen the touch rate is low, the touch threshold may be increased, resulting in more deliberate touches to meet the threshold.” EX1006, [0044].

106. The touch threshold may also be modified “based on location of the touch,” “based on a function parameter,” “in response to history information for a function,” “based on one or more ambient parameters,” or “based on user parameters such as user history.” EX1006, [0044-0050]. These “[t]ouch thresholds may be

repeatedly modified in response to change in any number, combination, and hierarchy of parameters.” EX1006, [0052].

107. In relation to modifying *time* thresholds, the only specific example that Griffin gives is for “when a user frequently deletes a ‘K’ after entering an ‘L,’ the touch threshold may be modified to increase the touch threshold, e.g., duration, that triggers entry of a ‘K’ after an ‘L’ is input.” EX1006, [0050].

D. Petitioner’s Proposed Combination of Chaudhri, Martyn, and Griffin

108. Dr. Cockburn proposed combination begins with one of Chaudri’s embodiments for a touch screen user interface, and then adds Martyn’s teachings about providing a contextual shortcut sub-menu of common application functions in response to a press-and-hold user input as well Griffin’s teachings about evaluating a user’s touch input against predetermined threshold criteria. EX1003, ¶¶61,63.

109. The embodiment that Dr. Cockburn choose from Chaudhri includes launching an app based on a tap gesture and entering a “reconfiguration mode” based on a touch-and-hold. EX1003, ¶¶61, 63.

110. Dr. Cockburn chooses from the several options disclosed in Martyn, the already assigned touch-and hold from Chaudhri. EX1003, 63. This requires Dr. Cockburn to analogize the touch-and-hold from Chaudhri with the press-and-hold from Martyn. EX1003, ¶63.

111. Looking at the respective disclosures Dr. Cockburn notes that the touch-and-hold gesture, or the appropriate analog as mentioned above, are assigned in Chaudhri to the method of entering “user interface reconfiguration mode” and in Martyn to calling the summary window. EX1003, ¶75.

112. Dr. Cockburn notes this conflict by describing how a POSITA would have been “[p]resented with three distinct operations invoked by duration-based touch gestures at an application icon—namely, [1] a tap to launch the application, [2] a touch-and-hold to trigger a contextual sub-menu, and [3] another touch-and-hold to transition into the reconfiguration mode.” EX1003, ¶75.

113. To address the fact that the design choices faced by the POSITA involve touch different results assigned to the touch-and-hold gesture, Dr. Cockburn suggests a POSITA would have attempted to distinguish these gestures to resolve the conflict. EX1003, ¶76.

114. Dr. Cockburn thinks that a POSITA would have created two different “duration criteria” from Griffin’s disclosure. EX1003, ¶77.

115. Finally, having created this conflict, Dr. Cockburn makes an arbitrary decision as to how to resolve it by assigning Martyn’s App Snapshot the shorter touch duration because the App Snapshot “is related to selecting and launching the application (i.e., assisting the user with selecting a desired application functionality)” and therefore should come first. EX1003, ¶77. This leaves the longer

touch duration for the “interface transition[]” into the “‘reconfiguration’ mode.” EX1003, ¶77.

IX. The Chaudhri-Martyn-Griffin Combination Lacks Evidentiary Support

A. A POSITA would not have combined Chaudhri with Martyn

116. Dr. Cockburn states that “the predictable result of the Chaudhri-Martyn Combination is a mobile device featuring an application menu user interface that displays a contextual sub-menu, such as Martyn discloses, in response to a touch-and-hold gesture applied to Chaudhri’s touch screen.” EX1003, ¶63.

117. Chaudhri does disclose “launching an application in response to a ‘tap gesture’ at an application icon.” EX1003, ¶75; EX1004 [0217-0218].

118. Chaudhri does not disclose popping up a context menu in response to exceeding a first threshold. Rather, as Dr. Cockburn admits, “Chaudhri’s instruction [are] to trigger specific functionalities based on the duration of the user’s touch contact on an application icon—e.g., a tap gesture to launch an application and a touch-and-hold gesture to trigger the reconfiguration mode.” EX1003, ¶71.

119. Dr. Cockburn points to four reasons to support the Chaudhri-Martyn combination. EX1003, ¶¶64, 66, 68-69.

120. Dr. Cockburn’s first reason is that “providing a sub-menu of selectable options corresponding to commonly used application functions was a ubiquitous feature in computer GUIs by 2011.” EX1003, ¶64.

121. Dr. Cockburn's second reason is that "a POSITA would have recognized that applying Martyn's teaching of a contextual sub-menu would predictably improve Chaudhri's user interface for 'a menu of applications' from a usability standpoint." EX1003, ¶66.

122. Dr. Cockburn's third reason is that "Martyn identifies specific 'advantages' of the 'App Snapshot' contextual sub-menu over other known techniques for accessing a desired functionality on a mobile device." EX1003, ¶68.

123. Dr. Cockburn's fourth reason is that "the POSITA would have been prompted to pursue a combination with Martyn based on Chaudhri's suggestions to do so." EX1003, ¶69. To support this Dr. Cockburn uses the example that "Chaudhri teaches the notion of displaying a contextual 'pop-up dialogue' sub-menu that 'provides the user with options for modifying the selectable user interface object.'" EX1003, ¶69.

124. These reasons do not accurately reflect a POSITA's understanding of Chaudhri and Martyn.

1. *Petitioner's motivation would not have led a POSITA from Chaudhri to Martyn*

a) *Dr. Cockburn's first and second motivations are conclusory and contradictory*

125. Dr. Cockburn's first motivation merely states that "providing a sub-menu of selectable options corresponding to commonly used application functions was a ubiquitous feature in computer GUIs by 2011." EX1003, ¶64.

126. Dr. Cockburn's suggestion is merely that a feature desirable on one device would automatically be desirable if incorporated on another device. This is not true.

127. A POSITA would carefully consider adding popular user interface features to another device and would not merely rest on previous desirability. For example, a stark contrast in success existed between the Nintendo Wii and the Sony PlayStation Move. The Wii's motion-sensing controllers were intuitive and engaging in its application. But when Sony introduced this proven desirable feature on its gaming platform, the PlayStation 3 in the form of the PlayStation Move, it failed. Despite having similar technology, the Move did not achieve the same level of success because of poorly articulated design implementation. Similar examples exist in touch devices such as Apple's iPad and Windows 8 version of Microsoft's surface tablet.

128. Dr. Cockburn's second motivation is that a mobile device with a superior interface from a usability standpoint would be more attractive to consumers so a POSITA would have recognized that applying Martyn's teaching of a contextual sub-menu would predictably improve Chaudhri's user interface. EX1003, 66-67.

129. Dr. Cockburn fails to provide a reason that a user interface would be superior.

130. Dr. Cockburn fails to explain why the proposed combination would be superior in light of considerations of buttons, battery, and navigation.

131. Context matters. Ubiquity alone is not a reason for a POSITA to choose a solution to any given problem. The problems faced by specific design implementations are often unique to device design choices. The functionalities in a user interface are frequently constrained by the design choices of the physical device. For example, if a device has a button considerations will go into what that button will be designed to do in relation to the user interface.

132. Although a sub-menu can sometimes be a desirable feature, the battery and user cognition savings are only achieved when the sub-menu is a solution to a problem that is faced.

133. Additionally, there is no explanation as to battery savings as the menus already exist in Chaudhri. Merely using hindsight bias to rearrange menus is not a reason to look at another reference.

134. In Chaudhri, the existence of a sub-menu is not the problem as underscored by Dr. Cockburn's reference to Chaudhri's disclosure of a "pop-up" dialogue, rather the problem is navigating to this sub-menu on a touch-screen. EX1003, ¶64.

135. Since Chaudhri already discloses a way to reach this menu, it would increase rather than decrease the cognitive burden on a user who is already familiar with reaching this menu.

136. Furthermore, a general principal of good user interface is to let the user do what they want faster not introduce unnecessary steps.

137. In user interface design, less is often more. Introducing complexity that is not towards a stated purpose is not an obvious modification. There must be context for why a POSITA would introduce extra steps.

138. For example, the title of Chaudhri is "DEVICE, METHOD, AND GRAPHICAL USER INTERFACE FOR MANAGING FOLDERS." When the stated goal is folder management, a POSITA would not have looked to extra steps not directed toward achieving that goal.

139. Maryn recognizes the limitation of the time period as well, in acknowledging that a device may or may not have 4-way scrolling capacity. *See* EX1005, [0017]. Consistent with the time period, this evidences Martyn's the limited options for user input for mobile devices.

140. Chaudhri was not limited in the same way Martyn was. Chaudhri describes both a variety of buttons and gestures. EX1004, [0085], [0093], [0151], [0174-0175], [0260], [0292], [0379], [0380], [0380].

b) Dr. Cockburn’s four motivation contradicts, rather than justifies, the proposed combination

141. Dr. Cockburn’s undercuts his first two motivations in his fourth. Dr. Cockburn contends that “Chaudhri teaches the notion of displaying a contextual ‘pop-up dialogue’ sub-menu that ‘provides the user with options for modifying the selectable user interface object.’” EX1003, ¶69 (citing EX1004, [0274], [0317] (discussing “a contextual menu”), Figure 5LLL).

142. Dr. Cockburn still fails to explain why—despite Chaudhri’s disclosure of a pop-up dialogue sub-menu—a POSITA would have turned to Martyn’s App Snapshot. Rather, Dr. Cockburn merely asserts that Chaudhri’s disclosure of the feature “would have prompted a POSITA to seek out and apply the teachings from other references in the same field, like Martyn, that describe analogous implementations of contextual sub-menus to augment a user interface for a multi-functional mobile device.” EX1003, ¶69. But this is simply self-serving and unrelated to the cited references.

143. Dr. Cockburn fails to offer any meaningful rationale, instead concluding merely that because Chaudhri discussed a pop-up sub-menu, a POSITA would have looked to Martyn for a pop-up sub-menu. But that belies the

inconvenient truth that the only “problem” with Chaudhri’s pop-up sub-menu is that Chaudhri doesn’t trigger it in a way that Petitioner could argue met the claims. A POSITA would not have looked further unless they needed to, presented with a solution they would not have created a problem. For example, Chaudhri describes accessing its menu via an object modification target while in reconfiguration mode, Chaudhri teaches that as the way to access this menu. Accordingly, adding an extra object modification target is what Chaudhri teaches. Dr. Cockburn does not offer any explanation to deviate from this teaching. This teaching is shown below with a red circled “m” and a black arrow pointing to the equivalents as an annotation to Chaudhri’s Figure 5KKK.

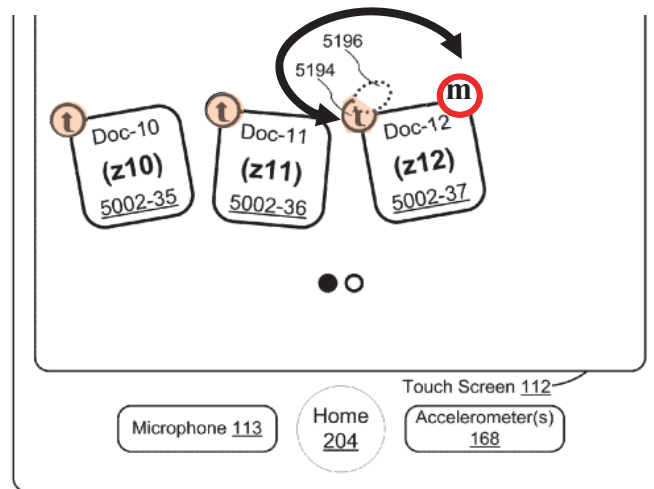


Figure 5KKK

c) Apple's third motivation regarding Martyn's specific "advantages" of its App Snapshot ignores any motivation for *calling* the App Snapshot.

144. Dr. Cockburn's third motivation suggests that "Martyn identifies specific 'advantages' of the 'App Snapshot.'" EX1003, ¶68. These advantages, "ease and speed of navigation," do not match the proposed combination because Dr. Cockburn proposes not just including "App Snapshot," but calling it using a specific input method. EX1003, ¶63. This specific input method of Martyn is an input method already assigned in Chaudhri. Dr. Cockburn purports that despite this conflict that a "POSITA would have sought to predictably improve Chaudhri's interface for accessing application functions in the same way Martyn did." EX1003, ¶68. A POSITA would not have predictability improved Chaudhri by reassigning one of its core functionalities.

145. Further, Martyn discloses multiple alternative methods of triggering the App Snapshot—that do not require a conflict with Chaudhri—and Dr. Cockburn does not provide a motivation for choosing the "press and hold of the select function." *See* Martyn [0017]. These methods include "a 1.2 second timeout, ... voice activation, softkey selection[,] etc[.]" and "using a right scroll function ...; a press and hold of the select function ...; or using a right cursor key ..." EX1005, [0017].

146. Dr. Cockburn provides no motivation for selecting that input method over others.

2. *Even if Chaudhri would have led a POSITA to Martyn, A POSITA would not have a reasonable expectation of success because the combination of Chaudhri and Martyn creates a conflict.*

147. Dr. Cockburn reaches his proposed combination only by, first, unnecessarily creating a conflict where two functions—calling up an App Snapshot and entering reconfiguration mode—are both assigned the same touch and hold input and then unnecessarily modifying the references to resolve this conflict.

148. Dr. Cockburn fails to explain why a POSITA motivated to add an App Snapshot would choose to create this conflict in the first place instead of simply assigning these two functions to different inputs explicitly disclosed in the references.

149. By choosing the press and hold gesture implementation from Chaudhri for Martyn's App SnapShot, Dr. Cockburn creates a conflict with Chaudhri because Chaudhri already uses a press-and-hold to enter reconfiguration mode. EX1003, ¶63.

150. Dr. Cockburn provides no explanation as to why a POSITA motivated to add an App Snapshot would create this conflict instead of using one of Martyn's alternative assignments or additional input selection options from Chaudhri.

151. The suggestion that a POSITA would have created a conflict, only to resolve the conflict without looking to the explicit teachings of the references is puzzling.

152. The core function of Chaudhri is entering a user interface reconfiguration mode to rearrange applications. In the proposed combination, this core function is assigned the touch-and-hold operation.

153. The core function of Martyn is opening a context menu. In Dr. Cockburn's proposed combination, this core function is also assigned to a press-and-hold operation.

154. A POSITA could not have designed an operational device, in line with Dr. Cockburn's proposed combination, as disclosed by Chaudhri and Martyn and maintained their core functions.

155. Dr. Cockburn does not explain why A POSITA would create a conflict when a conflict could have been wholly avoided by looking to the references.

156. Instead of creating this conflict, a POSITA could have retained the assignment to enter user reconfiguration mode by touching-and-holding from Chaudhri and noticed that Martyn discloses several alternative ways of accessing the App Snapshot as discussed above. *See supra* § VIII.B, EX1005, [0017]. This would prevent any conflict and leave both Chaudhri and Martyn unaltered.

B. A POSITA would not have then combined Chaudhri-Martyn with Griffin

157. Dr. Cockburn suggests that a POSITA would have looked for references that “provide additional information on [the] subject” of “tracking touch duration.” EX1003, ¶71. The only reason Petitioner needs Griffin, is because of the unnecessary conflict created by the combination of Chaudhri and Martyn. *Supra* §IX.A.2.

158. Putting aside this conflict, a POSITA still would not have been motivated to combine Chaudhri and Martyn with Griffin. Griffin’s teachings are unhelpful to a POSITA solving the problem of sequential durational touch thresholds as I have explain below. *Infra* §IX.B.2.

1. A POSITA would not have been motivated to combine Chaudhri and Martyn with Griffin.

159. Dr. Cockburn suggest two reasons why a POSITA would combine Chaudhri and Martyn with Griffin. EX1003, ¶70.

160. The first reason is that “Chaudhri’s instruction to trigger specific functionalities based on the duration of the user’s touch contact on an application icon ... would have prompted a POSITA to seek out and apply the teachings from references in the same field, like Griffin, that provide additional information on this subject.” EX1003, ¶71-73.

161. A POSITA would not have looked more into time tracking unless they needed to. Dr. Cockburn provides no explanation why a POSITA would look past all the input methods in Chaudhri and seek out additional information on time durations. *Supra* §IX.A.2. A POSITA would need a reason to look beyond the primary reference. Chaudhri does not supply such a reason.

162. Griffin merely describes durational thresholds, there are no instructions on how to assemble or when to use them. A POSITA would not have understood from reading Griffin that two duration thresholds could be assigned to three different functions. At most Griffin describes a sequence of measuring a threshold and then if the threshold is passed triggering that function. Griffin does not teach how to use the duration of those thresholds to trigger functions.

163. The second reason is that “a POSITA would have appreciated that Griffin’s implementation details were all well-known techniques.” EX1003, ¶74.

164. A POSITA would not have implemented Griffin even if the techniques were well-known.

165. Merely knowing of options does not motivate a POSITA to include those options. Knowing about the durational thresholds from Griffin, would not have motivated a POSITA to radically alter the core functionality of Chaudhri.

166. The time threshold from Griffin is what allows the discernment between a tap and a touch-and-hold. Neither Chaudhri nor Griffin nor Dr. Cockburn explain why a POSITA would bisect the touch-and-hold with another time threshold.

167. A POSITA cannot be motivated to create a conflict because that defeats the purpose of solving a problem.

2. *Even if a POSITA would have been motivated to combine Chaudhri and Martyn with Griffin, a POSITA would not have had a reasonable expectation of success.*

168. Griffin primarily relates to “a method and apparatus that reduces the occurrence of inadvertent touches causing functions to be performed.” EX1006, [0061].

169. “The use of multiple thresholds provides a mechanism by which multiple different functions may be triggered by touch alone” while avoiding inadvertent touches. EX1006, [0061].

170. Griffin states: “Multiple touch thresholds may be applicable for a touch, and different values of touch thresholds may be associated with different functions or input.” EX1006, [0032].

171. This disclosure is bereft of any teaching.

172. It is not enough to simply point to an element in a reference. A POSITA needs a motivation to combine. The fact that Griffin notes that multiple thresholds

are possible does not convey that multiple thresholds are desirable or should be used in a particular application, especially when there is an alternative available.

173. A POSITA does not pick and choose elements from a disclosure and assemble them spontaneously. A POSITA needs instructions for taking those elements and configuring them. A POSITA without a purpose in combining the elements would not have done so.

174. Griffin does not solve the problem created by the Chaudhri-Martyn Combination.

175. Dr. Cockburn's assignment of two functions to the same gesture are not resolved by Griffin's disclosure.

176. Griffin does not teach how to assign those thresholds to different functions. As I have explained above Griffin merely provides that different types of thresholds are modifiable and those modifiable thresholds can prevent inadvertent touch input. *See* ¶¶99-101.

177. The only explanation of Griffin's thresholds is to avoid inadvertent touches by modifying those thresholds.

178. This does not explain why Griffin would lead a POSITA to create two different time thresholds in order to assign already assigned functions.

X. Claims 1, 63, and 124 are non-obvious over Chaudhri-Martyn-Griffin

179. Dr. Cockburn does not argue that on their own, Chaudhri, Martyn, or Griffin disclose limitations [1.h], [1.j], [1.k], or [1.l] from claim 1. *See* EX1003, ¶¶88, 93, 95, 97, 100.

180. Dr. Cockburn asserts that “[t]he Chaudhri-Martyn-Griffin Combination renders obvious” limitations [1.h], [1.j], [1.k], or [1.l]. EX1003, ¶¶88, 93, 95, 97, 100.

181. For the above and below reasons, the Chaudhri-Martyn-Griffin Combination fails to render limitations [1.h], [1.j], [1.k], or [1.l] obvious.

A. [1.h], [1.j], and [1.k]: Action-Option-Display Criteria, Action Option Objects, and Two Time Thresholds

182. Limitation [1.h] recites “evaluating a duration of the first single-finger touch input against at least one of: ... one or more action-option-display criteria.”

183. Limitation [1.j] recites “in accordance with a determination that the first single-finger touch input meets the one or more action-option-display criteria that is met when the duration of the first single-finger touch input is evaluated to be greater than the first time threshold, displaying one or more action option objects associated with the first application without launching the first application.”

184. Limitation [1.k] clarifies that the “first time threshold” must be less than (and therefore different) than “a second time threshold” associated with “performing

an operation,” which Dr. Cockburn maps to entering “reconfiguration mode” in Chaudhri. EX1003, ¶97.

185. Dr. Cockburn does not suggest that Chaudhri discloses “displaying one or more action option objects associated with the first application without launching the first application,” let alone doing so “when the duration of the first single-finger touch input is evaluated to be greater than the first time threshold.” EX1003, ¶¶95-96. Chaudri only discloses performing a different action when the duration exceeds “a predetermined period of time”: “enter[ing] a reconfiguration mode.” EX 1004, [0217].

186. Chaudri actually discloses displaying a “pop-up dialogue” sub-menu, which Petitioner alleges as analogous to the App Snapshot in Martyn, Pet., 18, based on detecting a user interface input that is different than required by the claim language “when the duration of the first single-finger touch input is evaluated to be greater than the first time threshold.” *See* EX1004, [0274] (“the device displays a pop-up dialogue ... in response to an input that corresponds to a request to delete the selectable user interface object (e.g., tap gesture 5200 on a delete button)”).

187. While Dr. Cockburn suggests that “Martyn’s App Snapshot” meets the recitation of “one or more action objection objects,” Dr. Cockburn does not suggest that Martyn discloses displaying the “App Snapshot” “when the duration of the first single-finger touch input is evaluated to be greater than the first time threshold.”

EX1003, ¶¶95-96. Rather, Martyn discloses a number of possible “selection process” through which “the App Snapshot may be ‘called’ or ‘fired,’” but none involve a touch-screen gesture, let alone a “first single-finger touch input [] evaluated to be greater than the first time threshold.” EX1005, [0017].

188. Griffin discloses “modifiable” touch thresholds based on “a force ..., a displacement distance ..., an area of contact of a touch, a time duration of a touch, and so forth, or any combination thereof.” EX1006, [0032-0033]. These touch thresholds are designed as such to “reduce[] the occurrence of inadvertent touches.” EX1006, [0061]. Griffin’s disclosure of numerous, generic, and dynamic touch thresholds is distinct from the two, specific time thresholds required by the claims.

189. Unable to point to disclosure in any of the three cited references, Dr. Cockburn relies on the “Chaudhri-Martyn-Griffin Combination” as purportedly including calling up Martyn’s App Snapshot based on a “single-finger touch” evaluated to exceed a “first time threshold” that is less than and different than a “second time threshold.” EX1003, ¶¶88-96. As discussed in Section IX.A.1, Dr. Cockburn fails to provide evidentiary support demonstrating that a POSITA would have been motivated to incorporate into Chaudhri, Martyn’s App Snapshot using the particular user interface gestures required by Claims 1, 64, and 124.

B. [1.h]: Criteria Evaluation

190. Limitation [1.h] recites “in response to detecting the first single-finger touch input, determining a response to the first single-finger touch input based on evaluating the first single-finger touch input against at least one of a plurality of criteria, including evaluating a duration of the first single-finger touch input against at least one of: one or more application-launch criteria, one or more action-option-display criteria, or one or more operation criteria, and further including evaluating a movement of the first single-finger touch input against one or more movement criteria, for:” EX1001, 97:49-59.

191. Further Dr. Cockburn attempts to “turn[] back to Chaudhri” to provide the “application-launch criteria, an operation criteria, and a movement criteria.” EX1003, ¶91. While using Martyn to find the action-option-display criteria. EX1003, ¶92. It is undisputed that Chaudhri’s operation criteria and Martyn’s action-option-display criteria are both assigned to a touch-and-hold contact. (“Chaudhri ... a longer-duration touch-and-hold contact (e.g., 2 seconds) for initiating a reconfiguration mode (operation criteria[]),” EX1003, ¶91, and “Martyn’s “summary window” is invoked by a touch-and-hold contact on an application icon that meets a predetermined duration threshold (action-option-display criteria,[]),” EX1003, ¶92.)

192. A POSITA would not have been motivated to combine Chaudhri and Martyn to create a conflict and then arbitrarily resolve said conflict when Chaudhri and Martyn suggest alternative ways of assigning the operation criteria and the action-option-display criteria to prevent a conflict from arising in the first instance. *See supra* §IX.A.2.

C. [1.j]: Action-Option-Display Criteria

193. Element [1.j] recites “in accordance with a determination that the first single-finger touch input meets the one or more action-option-display criteria that is met when the duration of the first single-finger touch input is evaluated to be greater than the first time threshold, displaying one or more action option objects associated with the first application without launching the first application.” EX1001, 97:60-64.

194. Dr. Cockburn suggests that “Martyn’s teachings” regarding a contextual sub-menu called the App Snapshot, would have been integrated with “Chaudhri and Griffin such that the App Snapshot summary window is displayed in accordance with a determination that the user’s single-finger touch input meets a corresponding action-option-display criterion where the duration is evaluated to be greater than the first time threshold.” EX1003, ¶96.

195. A POSITA looking at two references disclosing a touch-and-hold gesture and a press-and-hold gesture assigned to different functions, would not have

attempted to arbitrarily resolve this conflict when a POSITA could have completely avoided any conflict by taking suggestions from the references.

196. Dr. Cockburn fails to articulate any type of superiority by failing to articulate the interplay of user interface design factors. Dr. Cockburn ignores many contextual design factors as I discussed above in paragraphs 130 through 138.

D. [1.k]: Operation Menu

197. Element [1.k] recites “in accordance with a determination that the first single-finger touch input meets the one or more operation criteria that is met when the duration of the first single-finger touch input is evaluated to be greater than a second time threshold that is greater than the first time threshold, performing an operation in connection with the first application icon.” EX1001, 98:5-12.

198. Dr. Cockburn suggests that “[a] POSITA would have understood that Chaudhri’s disclosure of detecting a touch lasting ‘more than a predetermined period of time (e.g., 2 seconds)’ teaches or renders obvious making a determination that the user’s single-finger touch input meets an operation criterion where the duration is evaluated to be greater than the second time threshold.” EX1003, 98. This suggestion is inconsistent with what Chaudhri actually discloses.

199. Chaudhri only disclose one time threshold—the threshold to transition into reconfiguration mode. EX1004, [0217]. Martyn likewise only discloses one

time threshold—the threshold giving rise to the App Snapshot. EX1005, [0017], [0020].

200. Dr. Cockburn reasons that Martyn’s App Snapshot is “related to selecting and launching the application” so it would be assigned a shorter time threshold is arbitrary and is only needed because Dr. Cockburn’s proposed combination has created an unnecessary conflict. EX1003, ¶77.

201. By choosing to add a time threshold, Dr. Cockburn directly contravenes his earlier statement, that “Chaudhri’s disclosure would remain intact” through the proposed combination. EX1003, ¶62. A POSITA would not have selectively chosen the teachings from their primary reference.

202. Factors such as user context, the requirements of a specific task, environmental factors, and device form factor are all relevant decisions in picking a user input method. Jumping to the suitability of using a second time threshold ignores all of these design considerations. Without a comparison of an additional time threshold against alternative input methods, Dr. Cockburn suggests a POSITA would make an arbitrary decision.

XI. CONCLUSION

203. All statements made herein of my own knowledge are true, all statements made herein on information and belief are believed to be true, and these statements were made with the knowledge that willful false statement and the like are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001.

Respectfully submitted,

Dated: October 18, 2024



Rajeev Surati, Ph.D.