Knowledge-based Interface transition diagram for SRS(Software Requirements Specification) in mobile application*

Taeghyun Kang
Department of Computer Science
University of Central Missouri
Warrensburg, MO, USA
tkang@ucmo.edu

Hyungbae Park
Department of Computer Science
University of Central Missouri
Warrensburg, MO, USA
park@ucmo.edu

Venkata Inukollu
Department of Computer Science
Purdue University Fort WAYNE
Fort Wayne, IN, USA
inukollv@pfw.edu

***Abstract*—This paper presents a phased development of user requirements for mobile applications. The usage of mobile applications has become increasingly prevalent and indicates the most rapid and exponential growth. Nonetheless, the number of unsatisfactory apps has been growing owing to the miscommunication between stakeholders and developers. Low-fidelity and prototype tools help to define user requirements by visualizing the needs of the users. However, a user who doesn't have any foundation in using prototype tools, will have difficulties in representing the requirements sufficiently. In this paper, the authors have proposed solutions to efficiently develop user requirements based on their understanding of the application and prototype tool. The authors of this paper suggest incremental developments of requirements in accordance with the user's knowledge in expressing the requirements for a mobile application.**

***Index Terms*—Interface transition diagram, UML, Requirement Engineering, mobile application.**

I. Introduction

With the exponential rise in the number of mobile device users, the demand for superior quality apps is also increasing. The number of smartphone users is expected to reach up to 3.5 billion [1] and mobile apps are estimated to generate more than 935 US billion dollars in revenues [2]. To develop high quality mobile applications desired by the end users, the customer requirements should be clearly defined and each element of the application needs to be efficiently presented on a page. Therefore, many software projects use a prototype to record customer's requirements and also ensure not to misrepresent the requirements to the development team. Various prototype software tools are used in the IT industry, however, the tools need to be used differently based on the classification of the software product. There are 2 fundamental types of Software product:generic product and customized product.

Generic software products, such as gaming apps or apps that provide a service to the users are marketed and sold to anyone who need them. Software development team analyzes and gathers the needs of the customers and processes the project according to the requirements. The requirements and layout of a generic product are usually defined by the development team. Developers and Designers are required to invest time in learning to use the prototype tool. Though, each prototype software tool provides different interface and functionalities, usage of the tool is not a big constraint given the diverse experiences of the developers and designers.

In the case of the customized product, a user who does not have the necessary programming skills, places a request with the development company for the desired product. Requirements in a customized product are defined outside of the development team. Also, there are significant limitations for the end user or customer to learn the prototype tool and deliver the desired functions with the screen layout.

UML(Unified Modeling Language) has become the international standard to analyze user's requirements and to define static and dynamic model of an application [3]. Use case diagrams and use case descriptions are used to define customer's requirements. Class diagrams represent static structure of an application and sequence diagrams depict dynamic interaction between objects in an application. These diagrams are used as a guide for software development. However, the survey of UML practitioners shows considerable variation using the above diagrams [4]. Various devices such as mobile devices, self-driving cars, and digital machines, have been developed and many types of software applications that come with it also have varied, but there is no standard guide on which UML diagrams should be used for each type of application to improve the quality of an application. Also, while UML diagram specification is released with time, there is no newly introduced UML diagram which can support new devices and paradigms like agile. In addition, the assumption that UML design is not required in Agile, undermines the advantages of using UML.

This paper describes an approach to define user requirements for mobile applications by considering the characteristics of a user. Characteristics to develop good and efficient requirements for mobile applications are defined, and, class of users are identified based on the user's knowledge in expressing the requirements for mobile application. Different levels of ITD (Interface transition diagram) are designed to enhance communication between customers and developers.

DOI reference number: 10.18293/SEKE2020-065

II. Related Work

The interviews and prototype are the most widely used techniques to elicit requirements in mobile application [5]. the literature review also shows that user participation is the most common issue in the requirement gathering process.

Use Case diagram is introduced by Ivar Jacobson, it is widely used to represent functional requirements [6]. The functionality of an application is described using actors, use cases, an association between actor and use case, and system boundary. Use Case diagram provides an overview of the system and defines key requirements [7]. However, there are limitations in expressing interface layout and finger gestures, which are key elements of mobile application.

The behavior of an application on mobile devices relies on finger gestures, sensors, and location data unlike software running on a desktop. Furthermore, due to the limited screen size, screen layout and design have a significant impact on the success of a mobile application [8], [9]. Therefore when customers define their requirements, the requirements engineering tool should encourage customers to consider the characteristics of a mobile applications.

The software requirements specification describes the capabilities of an application [10]. The SRS includes functional, non-functional requirements and external interface requirements. Also, a separate interface specification document can be developed for a critical aspect of an embedded system. It describes the layout of an application, connections between an application and other software components, an interface between the software components and hardware, and communication functions such as email and network protocols.

R. Hennicker et al. [11] proposed UWE(UML-based Web Engineering Approach) to design user interfaces. UWE consists of three design phases: conceptual, navigation and presentation design. The conceptual design shows internal structure of an application and navigation design identify the instances in an application and shows when they are used. The instances which is identified in navigation space model, are used in the presentation design which shows an abstract interface. The author also proposed the UML profile to build an abstract user interface and storyboarding scenarios from the navigation space model [12].

P. Abrahamsson et al. [13] proposed The Mobile-D approach to overcome technical constraints of mobile environment. The approach is consist of well-known agile practices such as continuous integration, pair programming, and user-centered focus. However it does not provide detail guides how user express their requirements.

III. Characteristics of good requirements for mobile application

A. Agility

The Agile Manifesto was published for lightweight software development and to focus on customer's satisfaction [14]. The result of the study indicates that agile methods are well suited for the development of mobile applications due to the dynamic and incomplete requirements of the mobile application [15]. Thus, quick response to customer request is a key factor for a successful project. Similarly, customers must communicate their requirements to the development team as clearly and quickly as possible. Due to the limitation of the screen size, the interface layout in an application and events based on finger gesture need to be reflected in the requirement. Many prototype applications have been developed and used to express customer's requirements, but customers are often unfamiliar with the use of these prototype tools. Customers therefore need a way of expressing their requirements which is not constrained by a prototype tool.

B. Finger gestures in mobile application

Interacting with mobile devices is greatly different from interacting with desktops or laptops. The desktop applications mainly use keyboard and mouse. However, mobile applications can be controlled by various finger gestures [16]. Therefore, these finger movements should be represented at design phase. The basic finger movements that control mobile applications are explained in table I. Mobile platforms such as android, IOS, and windows phone support more core finger gestures. For example, Android uses two-finger swipe down to immediately display switches for wi-fi, Bluetooth, mobile data, and the like. Thus, clear and diverse finger gesture information should be shared between customers and developers for representing customer's requirements accurately. Furthermore, more finger gestures can be defined according to the advancement of the mobile device's hardware. If the prototype tool does not immediately reflect these new changes, the customer cannot accurately reflect the desired requirements.

Gesture	Description
Tap	Press and release a portion of the screen
Double Tap	Press and release same part of the screen twice within a certain time(1second)
Long Tap	Press and hold the same part of the screen for a certain time (1second)
Drag	Press a portion of the screen, then move holding the finger on the screen and release
Flick	Press a portion of the screen, Quickly move and release.
Pinch in	Pinch inward
Pinch out	Pinch outward
Rotate	Press a portion of the screen with two finger, and rotate left and right
pan	Hold a side of palm on the screen, then Quickly move and release.
Shake	Move a smartphone up and down or side to side with rapid movements
Scroll	Move one finger across the screen without lifting. Drag a list up or down

TABLE I
Basic finger gestures to control mobile application

C. Interface design

In mobile applications development, the UI design is considered as one of important phases in the development process and the user requirements are highly dependent on the interface design [17]. Thus, the layout of an application, the events and detailed design should be considered together in tandem to express user requirements. Text-based requirement specifications are not intuitive and often lead to misinterpretations.

IV. Classification of User Group

User requirements of a customized product are defined outside of the development team. Unambiguous, clear, understandable and complete requirements should be delivered to the development team. However, most end users are unfamiliar with how to develop a good and requirements document. Thus, a developer needs to know the characteristics of a user to be able to distinctly understand the action items and what information should be gathered from the customer. The characteristics of a user in defining requirements is summarized in Table II.

Type of software product	Knowledge of using a prototype tool (Y/N)	Determined layout (Y/N)	Detailed Design (Y/N)
Customized product (C1)	N	N	N
Customized product (C2)	N	Y	N
Customized product (C3)	N	Y	Y
Customized product (C4)	Y	N or Y	N
Customized product (C5)	Y	Y	Y
Generic product (P1)	N/A	N/A	N/A

TABLE II
CHARACTERISTICS OF CUSTOMER

The class of an user can be identified by five groups for customized product and one group for generic product, based on the knowledge of using a prototype tool and the ability to design user interfaces. The first group in the category of customized product, does not know how to use the prototype tool. In addition, a layout and detailed design is not determined. In the scenario of first group, the Customer can only define functionalities of an application. The second User Group also has no experience in using prototype tool. Nevertheless, they have more specific requirements regarding functionalities and layout of the application's. Therefore, detailed design choices are to be suggested by development team. The user in the third group has detailed requirements for design and interface layout. So they need a way of succinctly communicating requirements without using a prototype tool. Lastly, customers can express their requirements using a prototype tool. Customer's requirements can be changed in the process of listening to developers and other design experts. However, base requirements specifications and interface design are determined by the customers and these tasks can be omitted in the development process done by developers. The detailed design cannot be developed without considering the application's layout. thus, such a user group is not classified in the table.

The authority of requirements and interface design for generic products are owned by the development team. They develop an application to compete with other products in the market. In order to understand the needs of customers in the market, the development team can survey or interview a selected group of users to define the required mobile application characteristics. During the market research, the developer requires a communication tool such as use case diagrams to understand the needs of the users. However, use case diagrams are not suitable for identifying the requirements related to mobile application's layout and detailed interface design.

V. Interface transition diagram for requirement specifications of mobile application

Various approaches are used to determine a customer's requirements. Nonetheless, there are still many constraints in delivering accurate requirements. Use case diagrams assist in visualizing user requirements and do not require long training session to learn, but, use case diagrams cannot describe detailed functionalities and user interface. Interview or survey techniques are document based; thus the initial requirements can be misleading over time. Requirements for a customized product should be defined at the user's level. Requirements of mobile applications can be expressed using low, medium, and high fidelity founded on the user's knowledge of prototype and design tools.

A. *Low fidelity diagram*

Low fidelity (LF) diagrams can be used for the C1 group in the Table II. LF diagrams can express basic finger gestures and interface transition according to them. Fig 1 depicts LF diagram that is divided into three sections like class diagrams. The first section identifies user interface using an interface name. Second section is used to briefly describe the main purpose of the page and last section contains element name, type, and finger gesture. Finger gesture makes UI transition. UI transition is represented using solid line with closed and filled arrowhead.

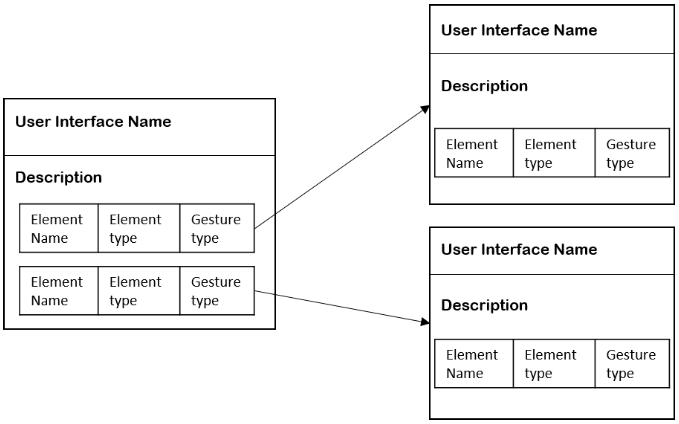


Fig. 1. Low fidelity interface transition diagram for UI transition

The popup menu is indicated by (p) in the page name. If the gesture on the element in the popup page triggers page transition, it is represented by using solid line with closed and filled arrow head as seen in Fig 2. On the other hand, if the gesture closes the popup menu and gets the user to remain on the page that opens the popup menu, the gesture is represented using dashed line with closed and filled arrow head. Relationship between elements is represented by a thick solid line surrounding the elements group.

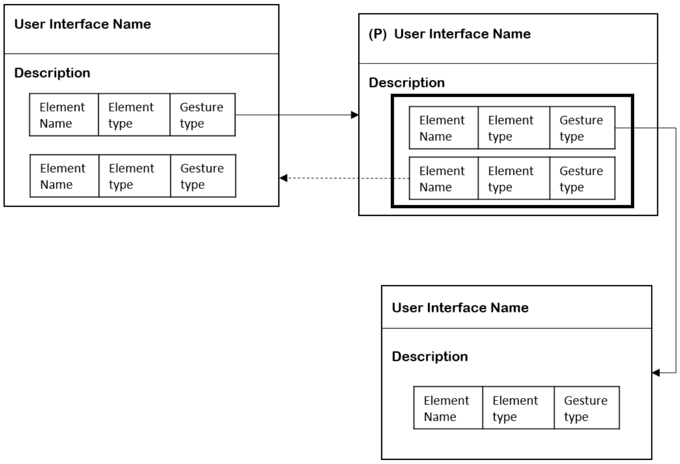


Fig. 2. Low fidelity interface transition diagram for grouping of element and popup menu

If the finger gesture activates an action only when it satisfies a specific condition, it is described by using diamond like flowchart. The condition in the diamond symbol requires a "Yes" or "NO" response and branch to the different user interface accordingly (Fig 3).

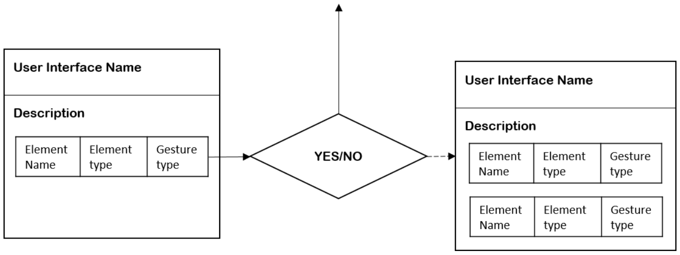


Fig. 3. Conditional Transition of an interface

One finger gesture in a mobile application can trigger two or more events and these actions can be represented with open arrow and a solid line (Fig 4). The purpose of LF diagram is to define elements, action, and finger gestures that trigger the action without considering a layout and detailed design.

B. Medium fidelity diagram

Medium fidelity(MF) transition diagram is used to determine the layout of an interface with information in LF transition diagram. The C2 user group in table II, can use MF diagram to define requirement specifications. Customer places an element without considering the details and style of the design. Events that trigger an interface transition are described on the transition line (Fig 5). Type of transition line and conditional events are expressed in the same way as LF.

If LF is already implemented and additionally customer defines MF, customer defines a layout for each interface and associates the page layout with LF using the page name and the element name (Fig 6).

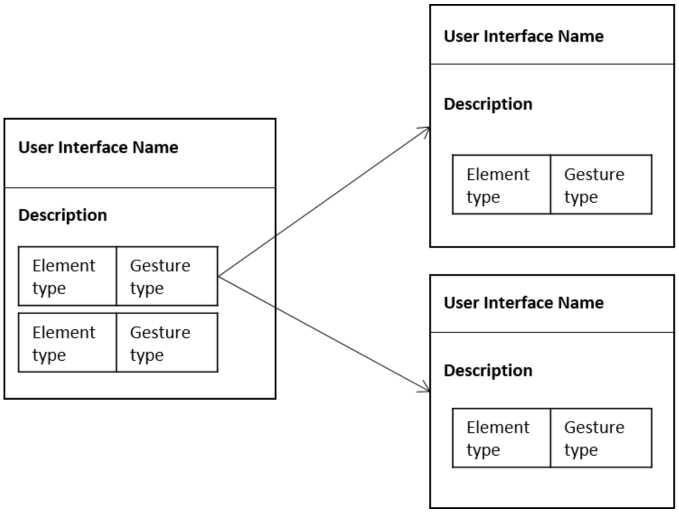


Fig. 4. A finger gesture can activate two different events

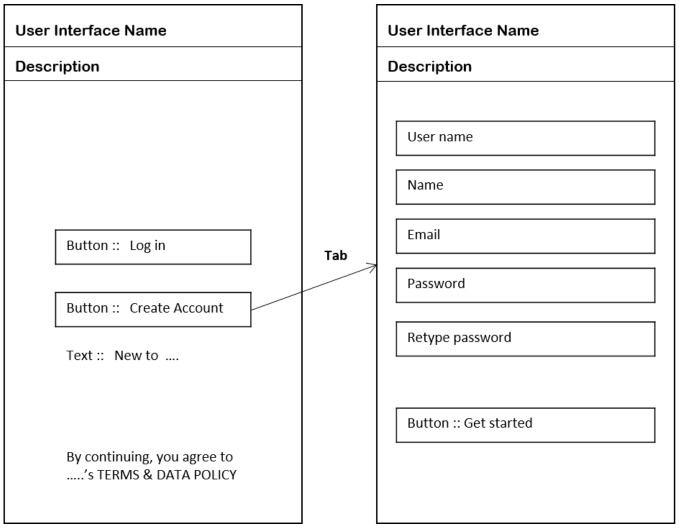


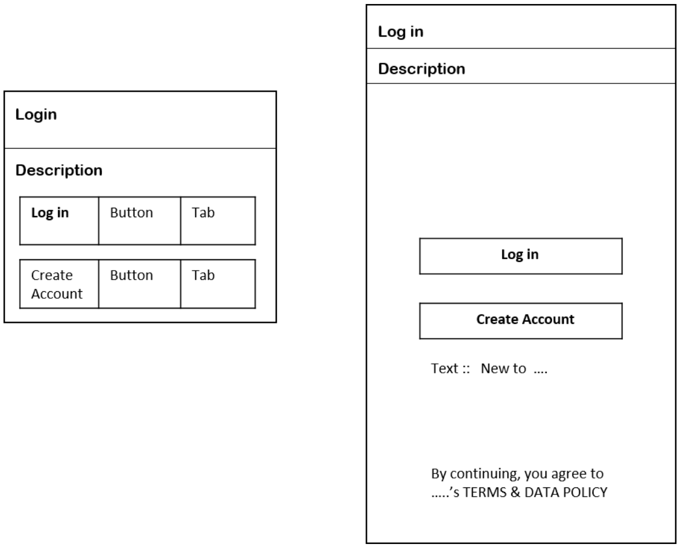
Fig. 5. Medium fidelity interface transition diagram

C. High fidelity diagram

User group C3 in the table II can use High Fidelity (HF) diagram to define their requirements. C3 user group can define and provide a basic design information such as size of an element and color (Fig 7). However, most end users do not have professional designing skills/knowledge. Thus, an interface designer of the development team needs to provide a detailed guide on the MF diagram.

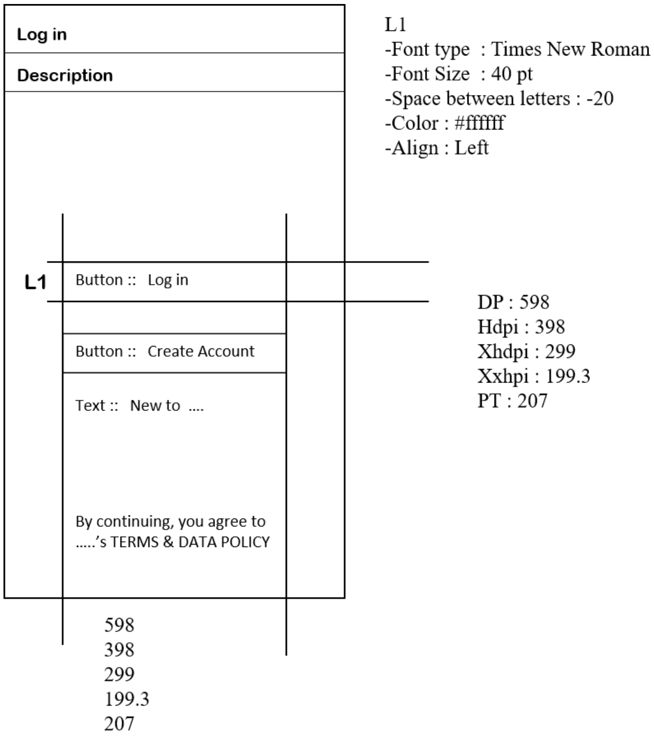
VI. Interface Transition Diagram for Requirement Specifications of Mobile Application

Requirements specification is one of the primary and crucial steps in the software development life cycle. Incorrectly defined or misrepresented requirements are one of the key factors that lead to the failure of the whole project. The following



Log in	Button	Tab
Create Account	Button	Tab

Fig. 6. mapping between LF and MF

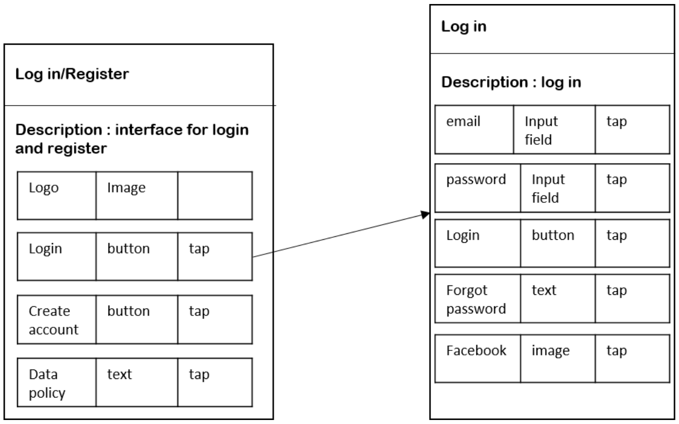


L1
-Font type : Times New Roman
-Font Size : 40 pt
-Space between letters : -20
-Color : #ffffff
-Align : Left

DP : 598
Hdpi : 398
Xhdpi : 299
Xxhpi : 199.3
PT : 207

Fig. 7. A finger gesture can activate two different events

case study illustrates the process of defining user requirements for a "Log In" page of a mobile application using LF, MF and HF diagram. Customers with inadequate knowledge of mobile application development, can define only functional requirements using LF diagram.



Logo	Image	
Login	button	tap
Create account	button	tap
Data policy	text	tap

email	Input field	tap
password	Input field	tap
Login	button	tap
Forgot password	text	tap
Facebook	image	tap

Fig. 8. Low fidelity interface transition diagram for Login page

The functional requirements of the login page, can be represented by defining element's name, type and event (Fig 8). Development team needs to evaluate and clarify the customer's requirement and suggest a layout of the application by using the MF diagram (Fig 9). If a customer has a desired layout for an application, the customer can define functional requirements and layout with the help of the MF diagram. In this scenario, the development team can save time in defining the interface layout. Nevertheless, in order to proceed with the actual mobile application development, the requirements of detailed design need to be specified and defined distinctly.

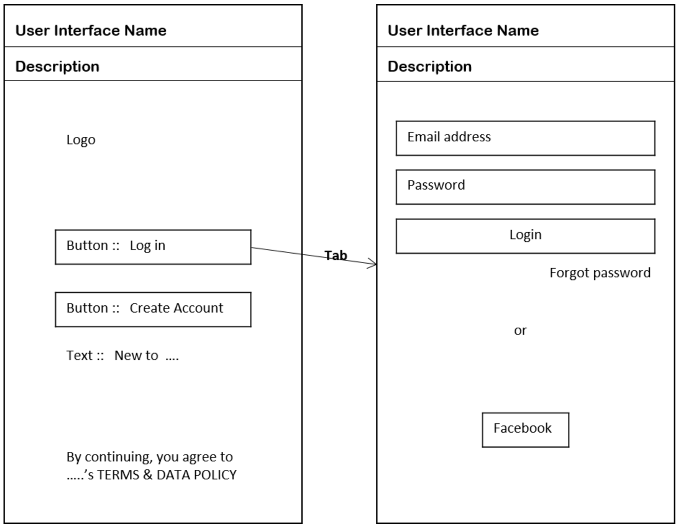


Fig. 9. Medium fidelity interface transition diagram for Login page

If a customer doesn't have specific requirements on the design, the development team needs to elicit design requirements with a prototype or HF diagram. HF diagrams are limited in expressing detailed design, but, they can present basic designs and facilitate collaboration with the design team.

VII. Conclusion

This paper has described an approach to define user requirements for mobile application based on the user's knowledge of using prototype tool. Most users are unfamiliar with the prototype tool and often do not know how to express their requirements.

Users can use Low fidelity, Medium fidelity or High fidelity diagrams. LF diagrams can identify element and event on the page. Users who have requirements of interface layout, can use MF diagrams. Basic design elements such as color, font, alignment, and size of element can be expressed using HF diagram. Furthermore, LF, MF, and diagrams enable incremental definition of requirements. Since the inputs and outputs of each diagram are clearly defined, the developers and users can easily identify their tasks to be done next.

This paragraph describes the future research for the interface transition diagrams. The diagrams need to be validated by further applying to different applications. Also, systematic management of changed requirements need to be developed.

References

[1] https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/

[2] https://www.statista.com/statistics/269025/worldwide-mobile-app-revenue-forecast/

[3] Kobryn, C. (1999). UML 2001: A standardization odyssey. Communications of the ACM, 42(10), 29-37.

[4] Dimensions of UML Diagram Use: A Survey of Practitioners

[5] H. Dar, M. I. Lali, H. Ashraf, M. Ramzan, T. Amjad, B. Shahzad, "A systematic study on software requirements elicitation techniques and its challenges in mobile application development", IEEE Access, vol. 6, pp. 63859-63867, 2018.

[6] I. Jacobson, M. Christerson, P. Jonsson. and G. Overgaad, "Object-Oriented Software Engineering: A Use Case Driven Approach," Addison-Wesley, Wokingham, 1992.

[7] F. Alhumaidan, "A Critical Analysis and Treatment of Important UML Diagrams Enhancing Modeling Power," Intelligent Information Management, Vol. 4, No. 5, pp. 231-237, 2012.

[8] A.Wasserman, Software Engineering Issues for Mobile Application Development, Proc. of the FSE/SDP workshop on Future of software engineering research, FOSER 2010, IEEE Comp.b16 Soc. Press, pp. 397-400

[9] J. Dehlinger and J. Dixon, "Mobile application software engineering: Challenges and research directions," in Proceedings of the Workshop on Mobile Software Engineering. Springer, 2011, pp. 29-32.

[10] Wiegers KE (1999) Software requirements. Microsoft Press, Redmond, WA

[11] Hennicker R., Koch N.: A UML-based Methodology for Hypermedia Design. In Proceedings of UML 2000, Evans, A., Kent, S. (Eds), LNCS, Vol. 1939. SpringerVerlag (2000) 410-424.

[12] HENNICKER R. and KOCH N. 2001. Modeling the User Interface of Web Apllications with UML. In Practical UML-Based Rigorous Development Methods, Workshop of the pUML-Group at the UML´01, Gesselschaft für Informatik, Köllen Druck-Verlag

[13] P. Abrahamsson, A. Hanhineva, H. Hulkko, T. Ihme, J. Jäälinoja, M. Korkala, J. Koskela, P. Kyllönen, and O. Salo, "Mobile-D: An Agile Approach for Mobile Application Development," presented at OOPSLA 2004, Vancouver, Canada. 2004.

[14] Kent Beck, "Manifesto for Agile Software Development". Agile Alliance, http://agilemanifesto.org, 2001.

[15] Flora, H. K., Chande, S. V., Wang, X. (2014). Adopting an agile approach for the development of mobile applications. International Journal of Computer Applications, 94(17)

[16] C. Villamore, D. Willis and L. Wroblewski, Touch Gesture Reference Guide, April 15, (2010)

[17] Kumar, N. A., Krishna, K. H., Manjula, R. (2016). Challenges and best practices in mobile application development. Imperial Journal of Interdisciplinary Research, 2(12).