A System Perspective to Privacy, Security and Resilience in Mobile Applications

A Thesis Submitted to the
College of Graduate and Postdoctoral Studies
In Partial Fulfillment of the Requirements
For the Degree of Master of Science
In the Division of Biomedical Engineering
University of Saskatchewan Saskatoon

By

Ming Xu

PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis/dissertation in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other uses of materials in this thesis in whole or part should be addressed to:

Head of the Division of Biomedical Engineering
57 Campus Drive
University of Saskatchewan
Saskatoon, Saskatchewan S7N 5A9 Canada

OR

Dean
College of Graduate and Postdoctoral Studies
University of Saskatchewan
116 Thorvaldson Building, 110 Science Place
Saskatoon, Saskatchewan S7N 5C9 Canada

ABSTRACT

Mobile applications have changed our life so much, but they also create problems related to privacy which is one of basic human rights. Protection (or security) of privacy is an important issue in mobile applications owing to the high likelihood of privacy violation nowadays. This thesis is devoted to a fundamental study on the privacy issue in mobile applications. The overall objective of the thesis is to advance our understanding of privacy and its relevant concepts in the context of mobile applications. There are three specific objectives with this thesis.

Objective 1 is to have a more comprehensive understanding of the concepts of privacy, security and resilience (PSR for short) along with their relationship in the context of mobile applications.

Objective 2 is to develop the principles of design of a mobile application system with a satisfactory PSR.

Objective 3 is to develop a demonstration system (PSR demo for short) to illustrate how the principles of design can be applied.

A salient approach was taken in this thesis, that is based on a general knowledge architecture called FCBPSS (F: function, C: context, B: behavior, P: principle. SS: state and structure). An analysis of literature was conducted first, resulting in a classification of various privacies against the FCPBSS architecture, followed by developing a theory of privacy, protection of privacy (security), and resilience of the system that performs protection of privacy, PSR theory for short. The principles of design of a mobile application system based on the PSR theory were then developed, which are expected to guide the practice of developing a mobile application for satisfactory privacy protection. Finally, a demonstration system, regarding the doctor booking for minimum waiting time and energy consumption, was developed to issue how the PSR theory and design principles work.

The main contribution of this thesis is the development of the concept of PSR, especially the relationship among privacy (P), security (S), and resilience (R), and a set of design rules to develop a mobile application based on the PSR theory.

ACKNOWLEDGEMENTS

I will never forget the nights and days devoting myself into the intensive study, research and thinking at the delicate idea of this thesis, the passion and inspiration for my life.

People around me sacrificed so much to eventually witness the accomplishment of this job. Among them I would extend my special gratitude to Professor Chris Zhang who is not only a supervisor but also my most respective friend in guiding me, suggesting me and encouraging me throughout this long journey seeking my higher academic performance and better living senses. My wife, Rong, always chooses to stand behind me and sticks together. Her encouragement and awakened suggestions are still echoing around me and I will treasure this accompany by my side forever. Appreciations should be also presented to Dong, my elder brother, who encouraged me to make the decision to study in the University for a wide spectrum of choices, which definitely benefits me a lot. There should be many thanks to my daughter, Fairy, who has been fulfilling my life with happiness and tenderness. She is the reason why I have come alone so far and I will be her beloved knight regardless aging and distance.

I feel so grateful to get along with so many friends, Mrs. Meng, Dong He, Daniel Chen, Hong Wang and Joseph Luis. I will remember each moment we stuck together. I wish to express my appreciation to all of those who have given me a hand when I needed during this period of time at the University of Saskatchewan.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

4G LTE	4th Generation Long-Term Evolution
5G	5th Generation
Ad-hoc	Advanced Data Handler For On-line Control
ADT	Android Development Tool
APT	Advanced Persistent Threats
BSC	Base Station Controller
BSS	Base Station Subsystem
BTS	Base Transceiver Station
CPU	Central Processing Unit
DB	Data Base
DDoS	Distributed Denial of Service
EMEI	International Mobile Equipment Identity
FBS	Function-Behavior-Structure
FCBPSS	Function-Context-Behavior-Principle-State-Structure
GL	Geographical Location
GMSC	Gateway Mobile Services Switching Centre
GSM	Global System for Mobile Communications
HA	Human Attributes
HCI	Human Computer Interaction
HGL	Hospital Geographical Location
HN	Healthcare Number
HTTPS	Hyper Text Transfer Protocol over Secure Socket Layer
IA	Interaction Attributes
IARH	Information Attributes Reference to Human
IDE	Integrated Development Environment

IEEE	Institute of Electrical and Electronics Engineers
IN	Identity Number
IP	Internet Protocol
IS	Infrastructure and Substance
IW	Interact With
IW	State of Interaction With
JMV	Java Virtual Machine
MANET	Mobile Ad hoc Network
MME	Mobility Management Entity
MNS	Mobile Network System
MSC	Mobile Services Switching Centre
MT	Mobile Terminal
MT	Mobile Terminal
NDK	Native Development Kit
NSS	Network Switching Subsystem
OE	Outside Entities
OS	Operating System
PGL	Personal Geographical Location
PHP	Hypertext Preprocessor
PKI	Public Key Infrastructure
PSR	Privacy, Security and Resilience
RGL	Registration Geographical Location
SDK	Software Development kit
SIM	Subscriber Identity Module
SYN	Synchronization
TCP	Transmission Control Protocol

TLS	Transport Layer Security
TRAU	Trans-coding Rate and Adaptation Unit
UDP	User Data Protocol
UI	User Interface
UWSN	Untended Wireless Sensor Network
VNF	Virtualized Network Functions
WIFI	A technology for radio wireless local area networking of devices based on the IEEE 802.11 standards

CHAPTER 1 INTRODUCTION

1.1 Background

With the fast development of communication technologies such as 5G mobile network, our way of communication in daily life has been changed greatly. For instance, we rely on mobile devices more than ever before. According to Osseiran et al. (2014), telecommunication technology refers to infrastructures, software and regulations (or protocols) that support massive device-to-device communications with high reliability. In this connection, the 4th Generation (4G) or 5th Generation (5G) telecommunication technology stands for different types of mobile telecommunication technologies. The 5G mobile network can provide a much higher data transmission speed, such as up to 20 Giga Byte per second, than that of 4G.

Telecommunication technology is further divided into wired and wireless. The Internet-based wireless communication technology is closely related to the mobile system, as users are free to move while carrying them. According to (http://www.cac.gov.cn/cnnic39/index.htm), above 95% of Chinese users visit the internet via mobile devices in 2016 and this situation is believed to be a global trend.

The internet-based wireless communication technology (IWCT) is nearly used in every corner of our life, including the healthcare sector, which has a high potential to solve many problems that otherwise present to us without the IWCT. For instance, in the healthcare sector of Canada, one of the most challenging problems is long waiting with patients, especially to outpatients who do not have family doctors or who do not want to go with family doctors. In fact, this problem occurs in many places around the world, including China especially in big city.

According to Commonwealthfund (2017), "the median waiting time for diagnostic services such as MRI and CAT scans is about two weeks with 86.4% of the patients waiting fewer than 90 days. 56% of all Canadians need to wait for over four weeks for specialist appointment". Let us call this kind of outpatient a long-term attention (LTA) patient. Over 29% of Canadians report that they waited for up to four hours last time they visited the outpatient departments according to Commonwealth Fund Survey 2016 (Commonwealthfund, 2017). Let us call this kind of patient short-term attention (STA) patients. The STA patients also include those who visit the emergency department. According to Chen (2018), over 52% of the LTA patients need to wait for more than two hours, excluding the waiting times for drugs and examination. In Shanghai, China, the STA patients usually need to prepare for a whole day to visit a walk-in clinic or hospital. By the way, in China, the notion of LTA patients does not exist; the family doctor system is not available.

This thesis focused on the STA patients. Long waiting time to STA patients apparently creates many problems. Efforts have been made from a wide variety of aspects to cope with them by the Canadian healthcare system, including：

- Increase of investments on healthcare resources, e.g., increasing the number of clinics and hospitals, doctors, specialists and facilities.
- A focused investment on a certain type of diseases, e.g. cancer care, hip and knee replacement, etc.
- Change of the culture on the use of medical resources, e.g., the cultural shift from over use of medical examinations and treatments to a philosophy called lean use of medical resources (Vogel, 2015).
- Optimization of hospital resource management (Barlow, 2002).
- Use of the information and communication technology on improving the efficiency of interactions between patients and hospitals or clinics (Dai, 2016).

Mobile devices especially smart phones are indispensable devices in our daily life today. Higher performance of these devices in terms of computational capability and storage capacity enables many real-time applications in the interactions of patients and hospitals.

In the past decades, the health service system takes advantage of information technology, including the wire and wireless communication technology to enhance the efficiency of healthcare information management practice and to improve the quality of patients' service as well.

However, to be convenient is not the entire story, concerns such as security, privacy, usability and sustainability etc. have been discussed intensively. Many studies have been done on security and privacy as more and more users begin to be aware of the importance of privacy (Casaló, Flavián, & Guinalíu, 2007; Hackett, Kazemi, & Sellen, 2018).

According to Zhang and Lin (2010), to be resilient is an ability to stand against the negative impact and recover from the hazard situation. Resilience is an ability of a system to recover from damage of the system (Zhang and Luttervelt, 2011). Zhang and Lin (2010) outlined some ways to design a resilient enterprise information system, such as resource redundancy, anticipation and monitoring mechanism, learning and actuation method.

1.2 Motivation

This thesis is motivated by addressing two problems. The first problem is related to utilization of mobile technology to solve the long waiting time (LWT) problem, in particular the mobile technology for improving the booking of doctor appointment (Dai, 2016). The second problem is related to how to develop mobile technology with the salient feature of resilient protection of privacy. To the second problem, for example to avoid forgetting username and password for different online accounts, some users would record the account information, which is typically privacy information, with plain words in mobile devices. When devices are lost or this information is sneaked by the malware, the violation of the user's online account is in high likelihood. It is worth to mention that security makes sense to a particular concern related to human and society; in this thesis, security refers to the concern of the protection of privacy information. Further, security for privacy information can be regarded as a specific function of a system within the context of the wireless mobile environment. As such, the resilience refers to the protection of privacy information (i.e., security in this thesis). In fact, this thesis focuses on addressing the second problem but takes the first problem as a vehicle to facilitate the research on the second problem. In this way, both problems are tackled.

A general **question** to this thesis is: *how to make a mobile application system to be resilient to the function of security or to the protection of privacy information?*

The answer to the aforementioned question is seldom seen in literature. Further, based on the author's preliminary observation, the author believed that this question should be better addressed in a system perspective. By a system perspective, the author believes that development of a mobile application with a capability of the resilient protection of privacy information should begin with the analysis of work domains of the application (e.g. work domain analysis for the health service system for patients to visit doctors) and classification of privacy information upon the domain model. Definition of work domain analysis is directed to the literature (Wang et al., 2016).

1.3 Objectives and scope of the thesis

The overall objective of this thesis was to advance our understanding of the concepts of privacy, security, resilience and their relationship as well as to advance the technology to build a mobile application system in which an individual's privacy is highly protected with a high degree of resilience. The thesis took a particular mobile application – doctor appointment booking (Dai, 2016) as a study vehicle. Clearly, the outcome of the research will answer the aforementioned question. The specific objectives were:

- **Objective 1**: to have a more comprehensive understanding of the concepts of privacy, security and resilience (PSR for short) along with their relationship in the context of mobile applications.
- **Objective 2**: to develop the principles of design of a mobile application system with a satisfactory PSR.
- **Objective 3**: to develop a demonstration system (PSR demo for short) to illustrate how the principles of design in Objective 2 can be applied.

It is noted that in this research, the specific context, i.e., booking of doctor visiting, was considered only. As such, privacy and security make sense in this context. However, the theoretical development, i.e., theory and methodology, is applicable to other contexts, where privacy and security make sense.

1.4 Thesis organization

This thesis is composed by six chapters. Chapter 1 makes a brief introduction to the background and motivation of the thesis, and subsequently identifies the research question and specific research objectives. This chapter also gives an overview of the organization of the thesis.

Chapter 2 provides the preliminaries of knowledge for this research, including design and programming for mobile network system (MNS). Then, privacy, privacy protection (i.e., security) and resilience for privacy protection in the current literature are reviewed, which will give a basis for justification of the need of this research in the last section of this chapter.

Chapter 3 discusses the concepts of privacy, security and resilience from a system perspective (MNS in particular). Specifically, Section 3.1 presents a general framework of system. Section 3.2 discusses the concept of privacy on this framework, followed by a discussion of the concept of security (Section 3.3) and resilience (Section 3.4).

Chapter 4 proposes the principles of design of a mobile application to achieve a high degree of resilience in the protection of privacy or high PSR.

Chapter 5 presents a demonstration mobile application to illustrate the principles of design, as proposed in Chapter 4 and to show how a high PSR is achieved.

Chapter 6 presents the conclusion of the research along with the contribution of the research as well as the future work.

CHAPTER 2 BACKGROUND AND LITERATURE REVIEW

The development of computer software, particularly program, is coincided with the birth of the modern computer. Software is a bunch of operational codes, which occasionally binds with data and instructions together, under the guide of an algorithm that would be executed in a certain order to realize specific functions. Due to different goals, software is categorized into two groups which are system software and application software. System software, including Operating Software (OS), Database (DB), Device driver and other applications such as C# and JAVA compilers, acts as a mediator to bridge hardware and user applications. Application software is closer to end users. Microsoft Office, Chrome browser and Facebook app are kinds of application software, which facilitate the services provided by system software such as OS and DB to realize specific functions. This section would primarily focus on the mobile network environment.

2.1 Analysis and design of MNS

2.1.1 Requirement analysis

The first step to develop a software product, e.g., mobile applications, is to define the requirements from customers' voice of needs. The requirement has two types: function requirement and non-functional requirement. The function requirement is the specification of the information regarding what, how and why a software product is expected to do. For example, to provide a digital account report periodically or upon request at a time is the primary functional requirement of an accounting software product. The non-functional requirement is related to the operational environment, e.g., the response time of a system, storage limit and user interface. In hardware product development, the non-functional requirement is also called constraint and performance requirement (Fan et al., 2015; Dai, 2019). Laplante and Phil (2009) argued that defining a system's requirements accurately, clearly and correctly plays the most critical role for success of the development of the system. Any change of the requirements at the moment when the development has already entered the later process，such as prototyping or testing, would draw a significant negative impact on the development of the system, soaring of work hours and budget in particular. It is crucial for the designer and customer to have an access to each other's concerns and communicates effectively in order to make an agreement on a system's requirements. The graphical representation such as unified modelling language (UML) provides a bridge to the gap between the designer and customer. After the requirement is clarified, the possibility of a software solution needs to be determined, at which the cost effectiveness will also be discussed.

2.1.2 Design

Design is a cognitive process from the system requirement to the technical solution, which also meets various constraints coming from both social and environmental sustainability. In software product design, the technical solution is represented in form of design patterns or architecture (see Appendix A for more details).

Mobile applications are operated under an OS such as IOS and Android. They are different in the areas of the API interface, components, application design framework and programming language. Therefore, the foremost thing in design of a mobile application is to decide the type of OS. The most important design of a mobile application is the design of user interface (UI for short), because the mobile application is running on a small and light terminal which is limited in storage, computational capability, battery power capacity, display screen size, and demand on the direct touch screen operation. Before moving to more detailed discussions of the design of mobile applications, it is necessary to discuss the app style such as Native, Web or Hybrid app style (Gollotti, 2017; Appel, 2014), as they affect the subsequent design process.

Native App

Native app or native application is usually developed by the programming languages such as Java and C#. Its feature in UI is finger-touch at the specific icon on a smart phone desktop, and it has advantages including better user experience (UE) through smooth operation, rapid operation response and attractive UI. Native app also has the advantage of accessing the local resource such as camera, microphone and diverse sensors. However, negative aspects such as reluctant download request for new app version, which usually has considerable downsize, for both performance and security reasons hinder the wide use of the Native apps.

Web App

HTML5 refers to the fifth generation HTML framework. It provides cross-platform compatibility, considerable cost and effort in both software development and maintenance compared with native apps. Web apps are rooted in the core of Web technique (i.e., the HTML5 framework) and offer their functions through web browsers. As it is well known that the web system is vulnerable with security, web apps suffer from the vulnerability to "attacks". Web apps have also difficulty to access the local resource such as camera, and so on.

Hybrid App

Hybrid apps combine native and web styles in its framework and development technology. However, a specific combination (i.e., what from native app and what from web app) depends on particular application problems and further to particular application developers and users. Though the development of any UI needs to understand the mental model of developers and users (Zhang, 1994), hybrid apps need to decide how to combine, which is an additional challenge in developing any hybrid apps. Note that a general theory for hybrid design of engineering systems can be found from the work of Zhang et al. (2010) which pointed that the compatibility of two hybrid elements is quite crucial in responsible for the trade-offs between the system performance and compatibility. At present, the popular product of hybrid apps is such that the framework is based on native style while the operation is extended with the browser technique, e.g., Android's app store. The previous version of Facebook application had given up hybrid style and turn to native style because of the low efficiency with the web style in human-machine interactions (McComb, 2015).

The selection of design patterns

Once the app style is chosen, the next step of design is to find a design pattern to make the software system met the requirements. According to Raviteja (2007), a design pattern is a complete solution to the past problems, which is something like modules in the modular product system. A design pattern is thus reusable and adaptable to new problems. The next step of design is programming.

2.2 Programming of MNS

In the field of mobile programming, many mature computer languages are available, and some of them have undergone through a continuous evolution, e.g., C, C++, C#, Object_C and Swift. C is a classical structural language, while C++ adds the attribute of object to make it more efficient in building system with less coding due to the object inheritance mechanism, polymorphism and encapsulation. C# begins the concept of automatic resource management, e.g., garbage collection for memory management. Object_C and its enhancement edition Swift, build upon C# by providing the function such as access to the resources, UI and language function modules by inheritance; in addition, the operation mechanism such as dynamic type, dynamic binding and dynamic loading, which makes the task of programming with more flexibility and lean in terms of consumption of system resources.

Software development kit (SDK or Devkit for short) is often available to mobile programming. According to Sandoval (2016), SDK is "typically a set of software development tools that allow for the creation of applications for a certain software package, software framework, hardware platform, computer system, video game console, operating system, or similar development platform." By a simple inheritance, SDK improves the efficiency in software development dramatically. For instance, the development of an Android app on Java platform requires a Java SDK (JDK); for IOS app requires the IOS SDK; and for Universal Windows Platform requires the .NET Framework SDK. Different SDKs are with different focuses with respect to the different design and programming requirements. According to Shoavi and Orly (2015), the most popular SDK categories for Android mobile apps are analytics and advertising.

As mentioned before, to program a mobile application needs to determine the OS. At present, there are two OSs dominating the market (Gs.statcounter.com, 2018), which are Android OS (74.69% market share) and IOS (22.34% market share). Accordingly, there are two separate development environments, including programming languages, SDK, API and various component construction tools, corresponding to the two OSs, respectively. In the following, a more detailed discussion on these two environments is presented.

Mobile System Programming Environment for Android

Android is an open source operating system based on Linux and is primarily adopted in mobile devices such as smart phone and tablet. It is occasionally taken on desktop or tablet for benefits of good visibility and convenience. Figure 2.1 shows the architecture of Android software development environment. Android applications need to be developed under a particular compiling environment which is similar to Java virtual machine (JMV). As a result, developers tend to choose Eclipse as integrated development software. In Figure 2.1, ADT stands for Android development tool which acts as a middle-ware to enable the Eclipse to create the Android project and to facilitate Android software development. ADT is an indispensable plug-in component to develop Android apps on the Windows OS platform.

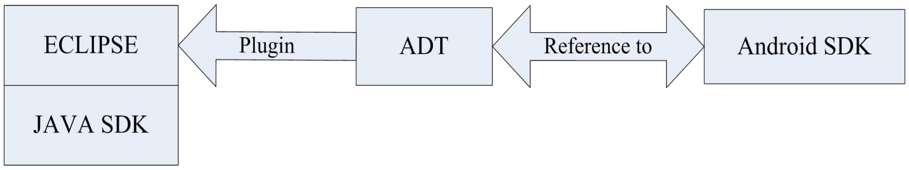


Figure 2.1 Software environments for Android programming

Android is operated under the Dalvik virtual machine (DVM) mechanism, which is a modified JVM, and is further believed to be the reason why Java is accepted as the most popular language for Android application programming. Besides Java, languages such as C/C++, Python, Ruby and so on are also used mobile application programming under Android. For instance, the programming language fraction for a project posted by Google (Luis et al., 2007) shows that Python and C/C++ take 40% and 40%, respectively, in the Android applications, by contrast, only 20% of the project code is written by Java. Different languages might be chosen due to specific aims. For instance, Java script might be adopted when programmers want to create a web frame for interaction in order to support high cross-platform requirement. A native app might choose C++, which is excellent to manipulate and communicate with system hardware, to guarantee an intensive interaction with database on the server side for better efficiency compared with the use of Java.

Mobile System Programming environment for IOS

Compared to the Android development environment, that of IOS is simpler. A big fraction of programmers would choose Xcode, which is issued and maintained by Apple Inc., as the integrated development environment (IDE) for the development of IOS applications. According to ("Xcode 10", 2009), Xcode (a type of specific IDE) "includes everything you need to create apps for all Apple platforms". Usually, an IOS mobile application can be created, programmed, debugged and tested in Xcode. However, when it comes to a team work or deals with a complex development project, some other tools maybe involved. For instance, Bugzilla is a project of management software for programmers to keep tracking crucial bugs, problems or issues involved with the application development from an initial state to the present situation in order to provide workflow management or bug visibility control for security of the project. Edition management issues including coding version control, resource distribution and updating of patches need to be handled as well. Cocoa Pods is an application helping to deal with issues related to resource dependency, such as class, third party user interface and kit and so on, to provide a standard format for managing external libraries by automatic integration into Xcode projects. GIT is also an equivalent tool for tracking changes in computer files and coordinating work on those files for team job.

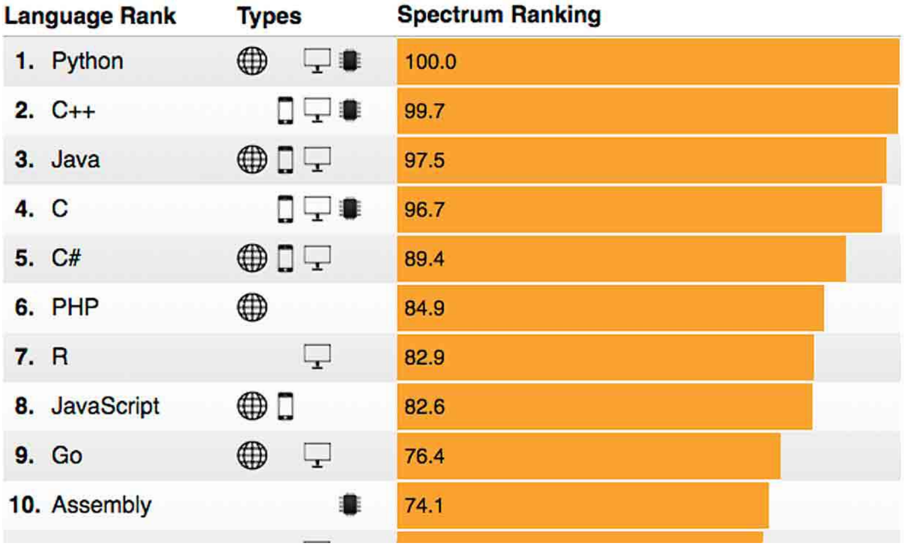
2.3 State of the art of design of related aspects

User interface (UI) design is an important task in mobile devices (Lin and Zhang, 2004; Lin et al., 2003; Lin et al., 2006; Xue et al., 2015). Punchoojit and Hongwarittorrn (2017) pointed out that the content of UI for mobile systems includes menus, list pattern, navigation, button, icon, filter and input area, which were also called widgets or gadgets (Zhang, 1994).

For menus, Park et al. (2017) examined conventional adaptive and adaptable menus to conclude that adaptable menus with highlights were in favor by most users, as the highlights could reduce memory load for adaptation. For button, Conradi, Busch, & Alexander (2015) studied the optimal size of virtual buttons for use while walking and suggest that larger buttons were verified to be less error and more efficient compared to that of small size buttons in the occasion of a moving context. For icon, Luo and Zhou (2012) investigated the effects of the icon-background shape and the Figure-background ratio on findability to conclude that the unified background yielded better findability. In terms of the input area, Kim and Jo (2015) argued that the input pattern for users includes the thumb input pattern and finger input pattern. The thumb-based interaction on virtual keyboards showed a 30% drop in throughput, as well as a significant drop in speed and accuracy; at the same time, the thumb-based interaction had lower stability in hand gripping (Kim and Jo, 2015). Besides, according to Akiki, Bandara & Yu (2014), design methods for UIs, such as an Adaptive Model-Driven UI Development methodology, are proposed based on the Adaptive UIs design method and Model-Driven engineering.

Design of the framework for mobile systems would be adaptive to specific domains. Unhelkar and Murugesan (2010) presented a mobile enterprise application development framework, which consists of six layers including communication, information, middle-ware and binding, applications, presentation, and security, dealing with challenges in the development and implementation of enterprise mobile applications. Schnall et al. (2016) proposed a framework consists of five typical stages including the problem recognition, information search, evaluation of alternatives, purchase and post-purchase evaluation. According to Schnall et al. (2016), "benefits for the framework include clearly illustrating the construction of the application as well as the cycling of the information, such as the source of the information and the place where the information is stored or shared."

The development of programming language is not so active compared with the software development strategy and techniques. Figure 2.2 illustrates the top ten programming languages from IEEE Spectrum (Gs.statcounter.com, 2018).



Language Rank	Types	Spectrum Ranking
1. Python	🌐 🖥 ■	100.0
2. C++	📱 🖥 ■	99.7
3. Java	🌐 📱 🖥	97.5
4. C	📱 🖥 ■	96.7
5. C#	🌐 📱 🖥	89.4
6. PHP	🌐	84.9
7. R	🖥	82.9
8. JavaScript	🌐 📱	82.6
9. Go	🌐 🖥	76.4
10. Assembly	■	74.1

Figure 2.2 Top programming languages of 2018 (Gs.statcounter.com, 2018)

The trend is very obvious that different languages are often combined together as a package to handle specific requirements on the basis of their different feature and capability. A single language can hardly handle more and more challenging requirements from users regarding the function, performance and UI efficiency. For instance, TaoBao is a popular e-commerce app for Chinese consumers and the application is a typical hybrid application, which is developed by the language package including Java, PHP, JavaScript and HTML5.

2.4 Privacy, privacy protection, and resilience: a summary of the literature

Privacy

There are several definitions of privacy in literature. Allen (2003) described privacy as the way for information control. Solove (2002) regards privacy as a multi-dimensional concept. National Research Council (2010, p.3) stated that information privacy concerns the protection of information about individuals and other entities. Malmir (2015) stated that privacy determines the right to control the information about people. Garner (2004) said "a private person has a right to choose to determine whether, how, and to what extent information about oneself is communicated to others, especially sensitive and confidential information". It seems that the definition of privacy in literature has not changed with the change of the medium to store the privacy information: paper medium and digital medium, and in the digital medium, there are further three types of data, namely structured, semi-structured, and unstructured. In fact, different medium to store the privacy information may matter with the definition of privacy information.

Classification of privacy has been studied in literature. Reiter and Rubin (1998) categorized anonymity into a six-point continuum: absolute privacy, beyond suspicion, probable innocence, possible innocence, exposed, and provably exposed. These are actually the description of six levels of the situation the privacy settlement might face. Kavakli et al. (2014) proposed to identify the privacy related properties that might be the potential threats toward a wider cloud computing use. These properties are isolation, provenanceability, traceability, interveneability, accountability and platform sensitivity. They provide the underlying mechanism for threats in a cloud computing environment related to privacy. The mechanism also explains why the information in a specific context would suffer from vulnerability and threats. Solove (2002) proposed six patterns of privacy data, which are right to be alone, limited access to the self, secrecy, control of personal information, personhood and intimacy, respectively. Solove (2002) also pointed out that the evolving technology has greatly changed the face of privacy.

In summary, the aforementioned work on privacy suffers from a serious shortcoming, that is the disconnection of the content of data[1], the restriction of access to a particular piece of data, the context in which a piece of data gets semantics, and the principle that governs the restriction of

[1] Data carries both information and knowledge (Zhang, 1994). To be general, this thesis uses ‘data’ and ‘information’ interchangeably.

access. Indeed, the point Solove (2002) made echoes well with the point that in characterizing privacy of a piece of data, the context is absent. In fact, privacy falls to the category of behavior according to the general knowledge architecture of FCBPSS (Zhang et al., 2005; Zhang and Wang, 2016). Details of FCBPSS can be seen in Appendix B.

Privacy protection (i.e., privacy security or security)

Zhang et al. (2009) proposed seven aspects of risks of privacy infringe in the clouds computing environment, including (1) privileged user access, (2) regulatory compliance, (3) data location, (4) data segregation, (5) data Recovery, (6) investigative support and (7) long-term viability. They considered reducing these risks from the server side but not the client side. There was a study on both techniques for reducing the risks of privacy infringe and application data systems (their structures) for reducing the potential exposure of privacy data of clients. Huang et al. (2014) proposed the concept of Mobile Cloud computing environment (Mobicloud for short) which provided the privacy protection service (e.g., the first being the user-centered inter-media authentication and the second being the information protection by means of security isolation). Indeed, a great effort has been taken on study of the technique from the server side to protect privacy data when it is in storage and transmission. Wang and Huang (2018) proposed an algorithm for privacy data protection in the process of data exchange between the third-party and service provider with the goal to eliminate the exposure of service users' privacy data.

Mukherjia and Sougata (2017) proposed a method called the role-based access control and identity certification, which was further based on a profiling algorithm. Alpár, Hoepman & Siljee (2011) proposed a model for identity management to ensure the control of privacy data at the hand of the privacy data provider. Encryption and steganography are the technique to enhance the protection of privacy data. Another technique introduced a third party to have privacy data between the client and service provider. Imran and Bhagyashree (2014) proposed a method to facilitate the third party auditor (TPA for short) to manage the log of the incoming request and outgoing response with a certain indicator (e.g. message digest). Chen, Beaudoin & Hong (2017) addressed the issue of privacy protection from three new directions: the extended parallel process model, self-control theory and routine activity theory. They suggested that the Internet users need to understand how the Internet scams work and resist the desire for immediate monetary benefits to avoid the risk of privacy infringe. The nature of calling some attention from the user's side is to view the privacy protection not only on the service provider side but also the client side. Li et al. (2018) addressed the issue of the location privacy protection based on the idea of shielding the location data through fuzzification of the location data. Damiani and Cuijers (2013) proposed a framework of the request management server (a reliable third party) between a mobile device user and a location provider as an inter-connection. Bai, Li & He (2018) made a survey on Chinese college students and found that over 98% of them got in touch with online social media for daily activity, e.g. on-line shopping, while 99.1% of those who had shared or provided their personal information for registration or contract, etc. were worried about the privacy violation. However, only 66% of them took some measures intensively to prevent privacy violation, such as providing make-up identity information, giving the least amount of information if possible.

In summary, the state of knowledge in privacy (data) protection or privacy security or security (without confusion of the context) appears to lack of a system view of the privacy protection issue. This seems to be related to the similar situation in the understanding of privacy data in the first instance (see the above discussion).

Resilience

The concept of resilience was first observed on human's body when tissue recovers from certain damages. According to Holling (1973), the same phenomenon also observed in field of the nature ecosystem. When a specific ecosystem encounters perturbations, such as drought or massive decease, an adaptation process is automatically triggered to make the system rebound from the disaster and eventually reach a new balance. This phenomenon is called resilience (Holling, 1973). Resilience is also well understood in material science, which is much related to the inherent property of material called elasticity.

Resilience is also well studied in **human centered system** or sociology system. According to Berkes and Ross (2013), “a resilient sociology system needs to embrace wider forms of structural change and it usually keeps significant bond with surrounding ecosystems which would draw influence on the sociology system.” Hegney et al. (2007) recognized the inevitability of change and embraces transformation and adaptation to address and manage change. Almedom (2015) argued that individual resilience can be browsed from not only psychological and behavioral aspects but also from the interaction with the outside environment of different levels including micro, meso, macro, and cross scale.

However, the foregoing understanding of resilience is upon one element instead of a network of elements, a view made by Zhang and Lin (2010). As such, changes are assumed to happen on one element only, absent from a possible cascade effect of privacy violation. Another important absence in all the above work is the so-called infrastructure-substance (I-S) framework of a system especially a network system. This framework says that a system has an infrastructure sub-system and a substance sub-system and the latter “flows” over the former. Therefore, in the I-S system, a change can be made on the infrastructure sub-system, or substance sub-system or their interaction or all of them. More discussions on the I-S framework will be given in Chapter 3.

Attention to resilience in man-made systems, i.e., engineering systems, was taken in the early 2000. Except Zhang and Lin (2010), Zhang and Luttervelt (2011) considered the resilience as a property that covers robustness. Figure 2.3 illustrates this school of understanding of resilience, where one can see that the resilience is spanned by three attributes that are robustness, recoverability and adaptation (Woods, 2015).

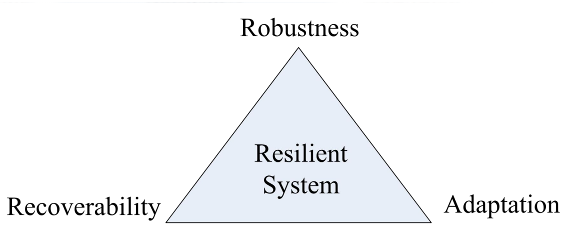


Figure 2.3 Attributes for a resilient engineering system

The earliest study of resilience to the data system may go to Zhang and Lin (2010), in which a mechanism called derived data was described for data recovery by a data system. This mechanism is in fact borrowed from the notion called derived attribute in data management, and in general systems especially **mechanical systems**, it is also called functional redundancy (Zhang et al., 2012). Efforts were taken to establish the resilient digital and information system; particularly a method called forward secrecy was developed to enhance the network from intrusion for untended wireless sensor network (UWSN) in forms of protocol (Roberto et al., 2013). Also, the method (Roberto et al., 2013) was developed to have the resource or function redundancy, namely period broad casting attribute that ensures the security feature for the entire UWSN. Tarik, Adlen, & Bruno (2016) studied the resilience for the next generation mobile system with the basic idea being to relocate and restore the lost state information by use of the existing resource such as visualized network function (VNF) and mobility management entity (MME) rather than costly physical redundancy equipment. In fact, their work is in line with the work of Zhang and Lin (2010). It is also noted that the privacy data protection system can be viewed as a **human-machine system**. The protection of privacy data is not only about the computer system including mobile devices and algorithms but also the clients or users and service providers. The protection is a collaborative effort by both machines and humans. Therefore, the approach in the resilience of human-machine systems, see the work of

Kiswendsida et al. (2013), may well be borrowed to enhance the resilience of a privacy data protection system.

In summary, the literature review shows that the state of knowledge on the resilience of privacy data protection system is in infancy. More efforts need to be taken on this problem.

2.5 Conclusion

Definition of privacy in literature does not take it to a system or privacy is viewed as an attribute to a human only. Privacy does not seem to be viewed as a relative concept, i.e., relative to a system that could be another human. Besides, the protection of privacy is thus lack of a systematic treatment, which is therefore unable to have a concept of resilience in protection of privacy. This assessment of literature supports the need of the objectives defined in Chapter 1 of this thesis.

CHEPTER 3 PRIVACY, SECURITY AND RESILIENCE IN MNS

The philosophy behind this thesis regarding privacy, privacy protection and resilience of the privacy protection system is a system view to privacy. There are two axioms in this thesis. The first axiom **(PSR-Axiom I**) is that privacy is of relativity, that is, Entity A has an attribute X, which may be anonymous to B but may not be anonymous to C. The second axiom (**PSR-Axiom II**) is that privacy is an interacting behaviour in a human-machine system, in particular the behaviour of how a machine or computer system protects the information of the human or human organization that should be anonymous or shielded to other humans or organizations or machines. According to PSR-Axiom II, a piece of data alone does not make sense to privacy.

In this chapter, a conceptual model of PSR is presented. In particular, Section 3.1 outlines a system concept called infrastructure-substance (I-S) framework. Section 3.2 presents the conceptual model of privacy, followed by Section 3.3 which presents the conceptual model for security, and Section 3.4 presents the conceptual model for resilience. A summary is given in Section 3.5.

3.1 The I-S model of the mobile network system

A mobile network system is an integrated system, which includes users, mobile terminals, wireless back-ends, and the Internet and operation management (Figure 3.1). In Figure 3.1, the relationships among these components are displayed. Users request services through mobile terminal (MT). MT has two functions. The first function is to communicate with users, similar to the function of graphic user interface (GUI), and the second function is to encode the user request into codes to be transmitted. The first function is achieved by a device such as cell phone or tablet, and the second function is achieved by a device (card) called subscriber identity module (SIM). The encoding of the user request has several purposes, including compression of the request signals to minimize the bandwidth, protection of the encoded signal in a robust (eavesdropping of data) and resilient manner (detecting errors and correcting them during the signal transmission process in the open air). The wireless back-end plays a role to manage all the user requests to various servers over the Internet.

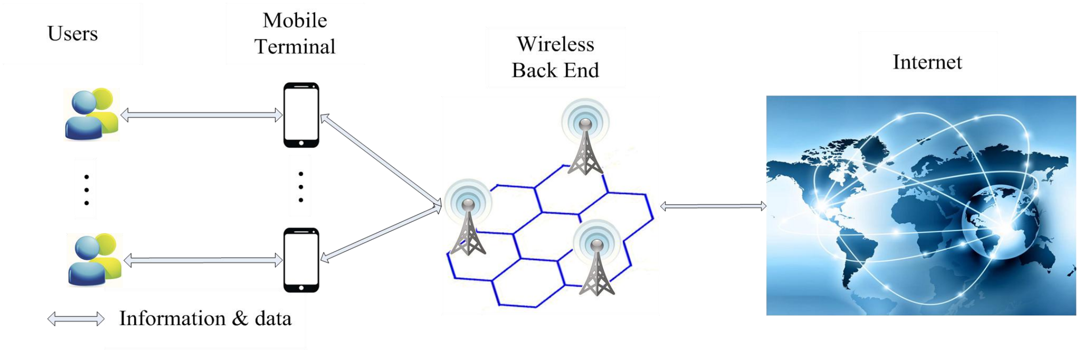


Figure 3.1 Layout of the Mobile Network System

Zhang and Luttervelt (2011) proposed a general framework for the service system, that is, a service system has two layers: infrastructure and substance (I-S). To MNSs, the infrastructure refers to the terminals and back-end systems, and the substance refers to the signal. It is noted that the general overall function of the MNS is to meet the user request for data (information and knowledge) (Zhang, 1994), which can be perceived by the user sensory organs (video, audio, etc.), and this function is achieved by the infrastructure that processes the substance to be brought to the user (Figure 3.2). For instance, when a real time voice call is made between two cell phone users, the service is in the form of the continuous voice message flow, measured by such as flow rate, integrity of the flow, and so on. This service is made possible by various infrastructures including the terminal, and so on.

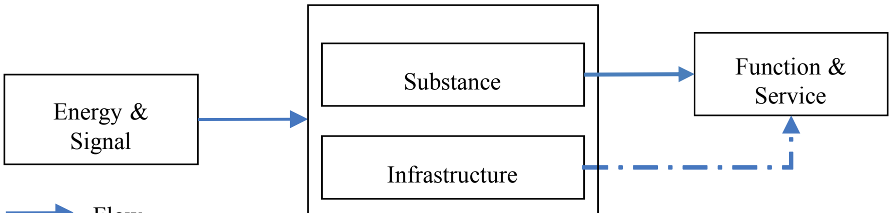


Figure 3.2 The I-S frameworks for MNS

3.2 Privacy

3.2.1 Conceptual model for privacy

By following the aforementioned two axioms (at the beginning of Chapter 3) regarding privacy, this thesis assumed that privacy makes sense within a particular system. Further, this thesis concerns data privacy, such privacy is one of the behaviors of a particular system, where privacy data can be defined. As such, a privacy system[2] follows the I-S framework, that is, it has two layers: infrastructure (human and equipment) and substance (data). The basic concept or ontology of privacy can be represented by the general knowledge architecture FCBPSS (see Appendix B for details), see Figure 3.3.

The structure of the privacy system is that it consists of the main entity say A (human or organization) and its privacy data are associated with other entities say B and C (Figure 3.3). The relationship of A to B and C is a one-to-many relation. Note that in Figure 3.3, A is labeled as the human and B and C are labeled as the outside entity (OE) to A for simplicity in discussion but without loss of generality.

Remark 3.1: The conceptual model as presented in Figure 3.3 is the so-called strong type data

[2] More precisely speaking, it is a system on which privacy data is defined.

model (Zhang, 1994), that is, the whole piece of information is represented by the type and instance mechanism. For instance, a person with ‘name’ of ‘John’ is first represented by the definition of a type ‘Person’ which the attribute ‘name’,

Person: = Name, Age, …..

‘Person’ here is a type. To this type, we can further define classes, e.g., Employee (a class), which means that the class ‘Employee’ has the attributes of ‘Name’, ‘Age’. An individual employee is then represented by the instantiation to the class ‘Employee’ such that

{EM 1 :=‘John’, ‘58’, …..}, where EM 1 gives the identity to this employee.

Remark 3.2: The structure may change with respect to time, event and /or space. This means that data may be added, deleted, or updated in the structure part of the model in Figure 3.3. Privacy may change, and changes may be due to the structure of privacy.

In Figure 3.3, the states of the structure include the attributes associated with the entities and their interactions. For instance, to the entity[3] ‘Person’, the attributes such as ‘Name’, ‘Age’ and so on define the states of the entity ‘Person’. To the entity ‘Interaction’, the attributes such as ‘Type of relationship’ and so on define the states of the Interaction, and the type of relationship further defines the degree of anonymity of one entity say A with respect to the other entity say B. For instance, the salary of one employee is of privacy to another employee in Canadian organizations.

[3] This thesis used word ‘entity’ for type, class or instance.

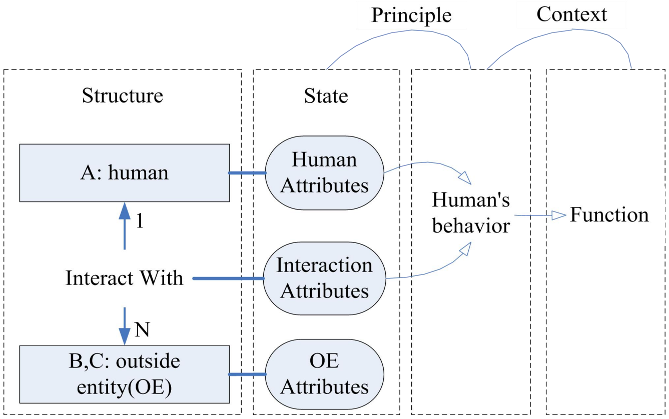


Figure 3.3 Conceptual model for privacy based on FCBPSS

In Figure 3.3, behavior is the relationship among the states of the structures. Instances of classes are created in conformity with their types. For instance, a person ‘John’ is created to the class ‘Employee’ that is the type ‘Person’. Accordingly, an instance <’John’, ‘Hospital’,> is created for the class ‘Interaction’, and further the instance ‘high’ to the attribute ‘degree of anonymity’. It is noted that why the degree of anonymity is ‘high’ is governed by the privacy rule of a particular judiciary body.

In Figure 3.3, the principle is the knowledge that governs the behavior. For instance, in the context of Canadian organizations, one of the judiciary rules for privacy protection is that among employees, they are not supposed to know each other’s salary. In the conceptual model, the principle knowledge is proposed to be represented as a semi-structure approach, such as XML format. This is because this kind of knowledge is between the document and structural data.

In Figure 3.3, function of a system is to fulfill the requirement of a human individual. For an individual who wants to accomplish a certain aim, the individual needs to interact with outside entities (Figure 3.3). For example, an individual needs to buy something from Walmart through the Internet. Here, the outside entity is the Walmart and the service system that assists the individual to fulfill this need. There will be a privacy issue between the individual and Walmart, and the internet-based service system has a sense of privacy protection, which is the function shown in Figure 3.3.

In case of Figure 3.3, it is also shown that function of privacy protection is context sensitive. Context refers to the pre-condition, the environment, and post-condition, which are related to the structure of a system according to FCBPSS (Lin and Zhang, 2004). Changing context will change the function of the system. For instance, Amazon provides not only online services but also brick and mortar stores. Sam decides to buy the T-shirt from the online store due to the lower price after acquiring the needed information about the size and style from a physical store. Here, one can see the change from offline shopping to online shopping. In this event to context change, though the structure of the system remains the same, which is still Sam (or Human) and Amazon (or Outside Entity) and the content of their interaction also remains, i.e., 'purchase'. The function of the system is now changing from the traditional commercial interaction to the e-commercial interaction.

3.2.2 Two general classes of privacy

Privacy can be classified into two categories: entity and relationship. The entity category of privacy refers to the attributes that define the entity. For instance, the attributes such as legal name, date of birth, sex orientation, finger print and genetic map and so on are this category of privacy. In addition, the information that exhibits the biological or social details of an individual also belongs to this category, for example visual image, figure, voice, hobbies, political trend, etc. Further, the property information, e.g., the information related to biological privacy, geographic privacy and intellectual privacy etc., also falls into this category.

The relationship category of privacy information refers to a pair of information related to one entity with another entity or other entities. Figure 3.4 illustrates an example that an individual, Sam, prepares to register to see a doctor. Sam first contacts the front desk of the clinic and provides necessary personal information (Figure 3.4) for the clinic registration. Healthcare number (HN) is the identity attribute of the Sam's healthcare information. In the meantime, the clinic also has the identity number (ID), and both HN and IN are associated to make Sam's registration happened. Register, which is shown in the middle of Figure 3.4, includes the attributes such as scheduled time, symptoms. The information as illustrated in Figure 3.4 is clearly related to two entities: an individual patient and the health service organization (i.e., clinic in this case). In this example, privacy issue may be raised in the third party who should be anonymous to the patient's identity information but get this information. Clearly, whether the privacy issue arises depends on the process of the interaction between the patient and registration unit as well as the system that enables this process.

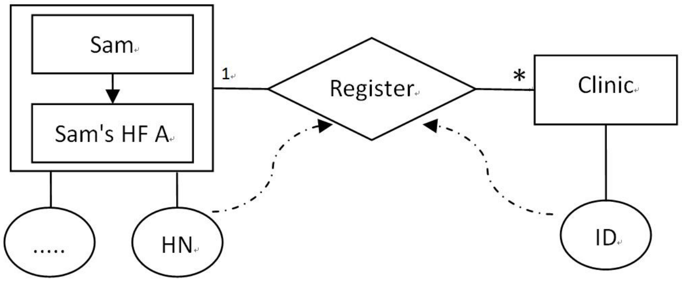


Figure 3.4 Interaction information for Sam's registration events

Privacy within the context of MNS

Privacy in the context of MNS is a specialized type of privacy as discussed above. At the state level (Figure 3.3), the location of a person is known to most of the operating systems in MNSs. Also, a person's biological characteristics, such as face identification (Phillips et al. 2018) or finger print, which may be used to unlock the person's cell phone, are the information that should be anonymous to many other entities. Moreover, the infrastructure identity may turn out to be a critical privacy issue with MNS. Mobile devices, especially smart phone, have been regarded as an indispensable utensil for humans. Therefore, the device identity becomes an attribute of the human user who carries the device. Consequently, the device identities such as international mobile equipment identity (EMEI), service identity module (SIM) as well as the phone number are all relevant to the privacy of the human carrier within the context of MNS.

In the context of MNS, privacy at the behaviour level (Figure 3.3) is changing and evolving fast due to the rapid development of technologies as well as usage patterns. The coming 5G technology greatly increases the bandwidth of data transfer from several KB per second to over 20 GB per second. This increase of the speed implies the increase of interactions of a user with his or her outside world, which further implies a potential increase of the cases of privacy violation. Another concern related to the privacy issue is that the development of technologies of big data mining increases the accuracy of inference of sensitive personal information both in cognition and emotion, and therefore increases the cases of privacy violation as well (Li, 2018).

3.3 Security

Security in this thesis refers to the protection of privacy. Since privacy in this thesis is about personal or organization information, security in this thesis is thus about information security. In MNSs, security is always one of the most important concerns by the user and widely discussed from various points of view. The security in MNS can be examined in the I-S framework (Figure 3.3). Figure 3.5 presents a model of security, which has three layers. The top layer is the one as illustrated in Figure 3.3, which is the privacy model. The middle layer in Figure 3.5 illustrates more details of infrastructure from the top layer. The bottom layer in Figure 3.5 is to extend the middle layer to the whole MNS with the information including mobile terminals and wireless back-ends, details of which can be found in Appendix C.

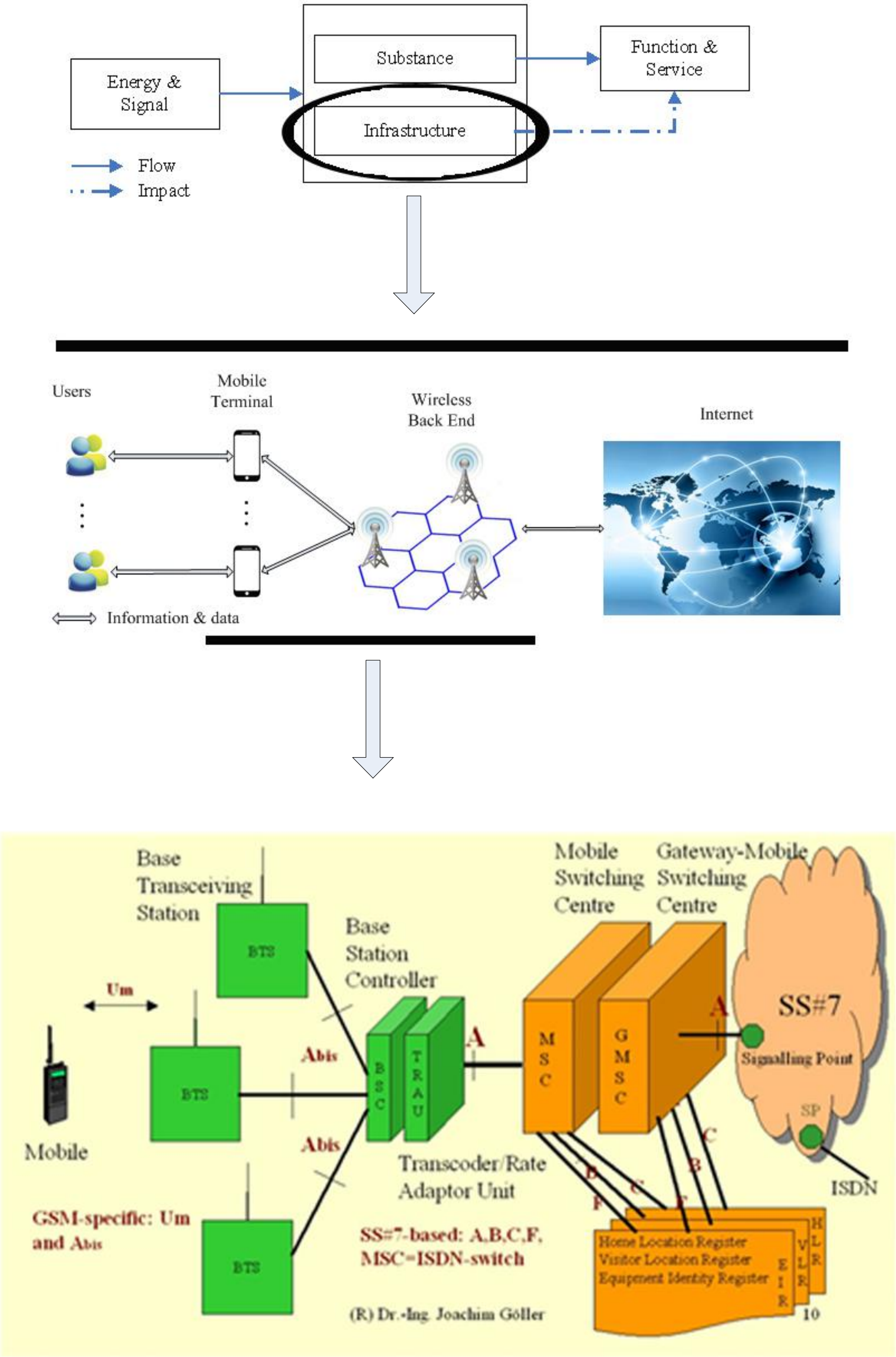


Figure 3.5 Security model in MNSs
(The bottom figure is taken from Dreamincode.net (2018))

Based on the above discussion, the first step to security enhancement is to understand various risks, i.e., various scenarios where an entity's privacy could be violated, i.e., made known to the parties that should be anonymous. To be systematic, naturally, the privacy model was taken, which means that risks are examined on the infrastructure, substance, and power[4] to run a MNS.

3.3.1 Risks associated with the infrastructure system in MNS

The infrastructure system of MNS may be physically damaged. For instance, the signal receiver module of the BBS is invalid; hence, the entire BSS can no longer provide services. When BSS is down, a local mobile network within a certain area is out of function. It is noted that a particular BSS is only one of the nodes of the whole system and each node may dysfunction. Therefore, the more number of nodes would imply the high likelihood of failures of the system.

Another example is that a user lost his smart phone or the processor chip is out of function. In such situations, the user may lose important personal information that might further result in privacy violation due to the reason that the lost phone may be accessed by other entities to which the entity's information should be anonymous. The damage of the infrastructure system of MNS may further be caused by incidents or accidents, e.g., natural disasters that may destroy the mobile terminals or the base stations.

The infrastructure of MNS may also dysfunction due to improper operations of it. The operation includes planning, scheduling, and executing with a particular MNS in receiving the clients' requests. These operation tasks need to ensure the entire system to run in consistence with the principles of the system at various levels. The principle is divided into three levels.

[4] In this thesis, power and energy are used interchangeable.

- Level 1: The principle of the lowest level component system, e.g., the principle that governs how the signal is transferred, how the circuit is connected.
- Level 2: The protocol related to communications of different component systems in a particular area of domain. For instance, the TCP/IP protocol is the one that facilitates the wired signal transfer.
- Level 3: The principle across different domains. For instance, the mobile roaming strategy is the one across different domains and aggregated from the protocols that govern the operation in different domains, including the authentication, home registration, and channel distribution.

Security challenges are related to all the three levels in operation in MNS. To the first level of operations, security challenges may occur in the design stage for component systems, at which a wrong principle may be chosen for a particular component system. Another example is the reluctant updating patch for the smart phone OS could be vulnerable.

Security challenges at the second level of operation are widely recognized, e.g., communication protocols (http:// www.opennetworking.org), in which the concept of the distributed denial of service (DDoS) is discussed by utilizing the weakness of synchronization sequence number (SYN) pattern, which is for TCP protocol to establish shake hands process for further process of data transfer, to overload the system by flooding massive fake SYN request to exhaust both computational and storage capacity of the target system.

Ad hoc is a Latin phrase which indicates a signified or special solution for certain commissions. According to Lee and Ra (2015), a wireless ad-hoc network refers to the wireless network which is continuously self-configuring depending on mobile terminals to maintain the communication of a network. The major task of the wireless ad-hoc network is to establish a network without any central control. The routing protocol is a crucial part to establish and maintain the Ad hoc network. Risks related to the routing protocol for the wireless ad-hoc network were discussed by Geetha and Sreenath (2015).

One example of the security challenges related to the third level is the operation of global system for mobile communications (GSM). GSM is a world wide mobile network communication standard which describes the protocols for second-generation (2G) of digital cellular networks. GSM is regarded as a cross domain-level principle which had passed strict test and security evaluation. However, with the fast technical evolution, GSM gradually exposes the flaws such as the lack of strong encryption and authentication methodology. Suspects facilitate the flaws to commit crime, for example as Liu (2016) introduced that the criminals using fake BTS instrument to deliver trapping messages.

Another case of the third level operation is advanced persistent threats (APT). ATP refers to a sophisticated invading technique using malware to exploit vulnerability in target information system. Moreover, ATP tends to continuously monitor and extract data from the target system posing more severe consequence and threats toward the counterpart which is different from conventional malware technique (https://en.wikipedia.org/wiki/Advanced_persistent_threat). According to Chen et al. (2018), APT not only facilitates a limited vulnerability of hardware or software for attacking but involves with wider scale and persistent intrusion by means of detecting, information gathering in both technical and managerial fields.

3.3.2 Risks associated with the substance of MNS

Data transfer happens intensively in the mobile communication environment, such as location sharing, navigation and critical data inquiry, and so on. The substance security, namely data and information security, in MNS, is the most critical source of the security issue. The back-end of the mobile network shares the analogous structure and technology as opposed to that of the conventional network; as the result, the substance or data security issues happen at the back-end of the system are quite similar to those in the conventional network, which are widely discussed in literature. For example, Mukherjea and Sougata (2017) discussed data encryption, role-based access control and identity certificate by use of the trust profiling in the context of mobile privacy data inquiry and transmission.

Within the context of mobile networks, data are stored or transmitted in and between mobile devices and relevant systems. Security issues are related to the actions such as unauthenticated visiting, storage, manipulation or data abusing. In addition, the risks from the providers of the mobile clouds service are also rising due to the typical events such as what happened with US T-Mobile users to lose their personal data because of the failure of the service provider in 2009 (Maggie, 2018).

3.3.3 Risks associated with the power

Power is needed to run the system. The example of the power system is battery. When the power system is damaged, data including privacy data are lost and disclose of privacy may happen. The example of the power system damage is battery explosion. There is also a case that an improper usage pattern causes an improper usage of mobile terminals, which further causes the power supply dysfunction. Moyers et al. (2010) noticed the battery exhaustion attack which would trigger rapid energy shrinkage.

3.4 Resilience

In the context of MNS (App in this case), the **resilience** of a system refers to the system's ability to **recover** the function of the system at an acceptable level subject to perturbations or mishaps from both the inside and outside of the system, adapted from (Zhang and Luttervelt, 2011). By saying ‘recover’ it is implied that the App system is partially damaged in its structure and the system then performs changes in its remaining structure in a time duration to make the lost function available to the user, adapted from (Zhang and Luttervelt, 2011). It is noted that the resilience as defined above is different from robustness in that the **robustness** of a system is the system’s ability to **keep** the function of the system at an acceptable level subject to perturbations or mishaps from both the inside and outside of the system, adapted from (Zhang and Luttervelt, 2011). By saying ‘keep’, it is implied that the system has not damage in its structure, adapted from (Zhang and Luttervelt, 2011).

To the protection of privacy information (or security in this thesis), the resilience refers to the ability of the system to protect individual’s privacy information from being acquired by outside entities to which the information should be anonymous; particularly in the case that privacy information has been lost but the system recovers the lost privacy information. There are two situations regarding the protection: (1) to defend hacking attacks and (2) to alert individuals for correct decisions and actions when facing attacks.

There are two situations regarding the loss of privacy information. The first situation is that a piece of privacy information has been hacked. The second situation is that a piece of infrastructure, over which a piece of privacy information runs, has been damaged. The damaged infrastructure may carry privacy information, and there are two further cases: Case (i) the damaged infrastructure is left to the outside entity that should be anonymous to this information, and this case returns to the first situation above, and Case (ii) the damaged infrastructure is lost and the owner of this information wants to get back this information. Case (ii) has been discussed in literature (Zhang and Lin, 2010); specifically, the problem with this case can be addressed by the design principle for information systems called “derived information” mechanism, which may also be called redundancy data (Zhang and Lin, 2010).

3.5 Relationship among PSR

According to the previous discussion, privacy is related to data (both information and knowledge), so it is data privacy. The nature of data privacy is that data of an individual entity (one person, a community or a technology) should not be known to another entity or other entities. Therefore, privacy is related to the substance system from a point of view of the I-S framework (Figure 3.6). Security is related to both infrastructure and substance systems, in particular it is about how the infrastructure system can protect privacy from being lost. To MNS, security (i.e., protection of privacy) is one of the functions or services (Figure 3.6). Therefore, the resilience makes sense to the security. A resilient security system is such that if the security of a system is partially damaged, the system can recover this lost security in an allowable time and with an allowable cost by itself. Finally, to make the infrastructure and substance systems run, the system must be supplied with energy or power, as well as proper format to represent private data or data that makes sense of privacy and needs of protection; see Figure 3.6. It is noted that in this thesis, we always assumed the availability of a proper format for data representation; in other words, this thesis does not study the data representation for the purpose of privacy protection.

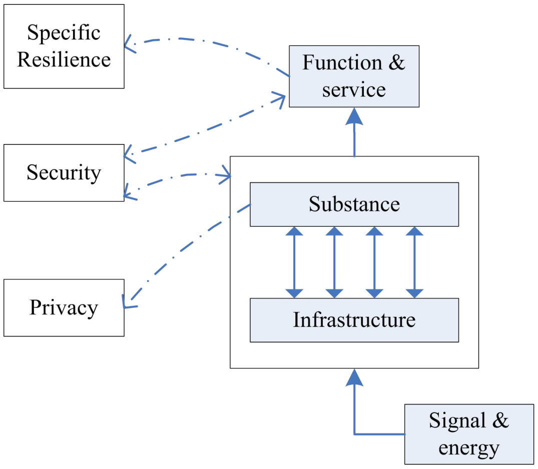


Figure 3.6 The I-S view of the relationship among PSR

3.6 Conclusion

In this chapter, privacy, security and resilience were discussed from a system perspective and a relativity perspective. The salient point in this thesis is to have displayed the relationship among privacy, security and resilience. It is noted that in the contemporary literature, nobody seems to relate the resilience to privacy and privacy protection (i.e., security). A novel generic model for these three concepts has been developed.

CHAPTER 4 DESIGN PRINCIPLES

In this chapter, the design principles for a resilient privacy protection system are proposed. According to the discussion in Section 3.5, we will first discuss the design principle for security in terms of privacy protection (or security for simplicity with confusion) in Section 4.1. After that, we will discuss the design principle for making the security be resilient or for a resilient security system in Section 4.2. It is noted that the system here refers to MNS. Section 4.3 is a summary of the discussions.

4.1 Design principles for security in terms of privacy protection

Based on the preceding discussion, referring to Figure 3.6 in particular, we divide the design principles for security from three aspects: the infrastructure, substance and energy[5].

4.1.1 Infrastructure aspect

- Rule I-1: To choose well evaluated, matured and widely accepted options of the infrastructure to fulfill the functional and constraint requirements.
- Rule I-2: Any new algorithm should be evaluated and tested before it is used.
- Rule I-3: Choosing an adaptive application style based on the function and security requirements.
- Rule I-4: A platform, in particular clouds and OS, must be evaluated before it is used for developing applications for the requirements of the security and function. At present, the trend of mobile applications for both service and data storage is moving toward the clouds system. Different Cloud platforms provide different services with respect to

[5] Energy and power are different but this thesis uses them interchangeably.

different service requirements. This trend transfers the security obligation from the end devices along with traditional servers to the Clouds system.

4.1.2 Substance aspect

Substances for the mobile system are data. In the following, the design rules for information privacy are discussed, followed by the design rules for the protection of the privacy information.

- Rule S-1: Addressing the privacy information. Privacy information needs to be identified, classified according to the discussion in Chapter 3 especially the model of Figure 3.3
- Rule S-2: To specify the responsibility of the user side and system side precisely to determine forms to enhance the responsibility. There are two forms: legal binding and non-legal or informal binding. The former approach involves more steps to establish, which often makes the user shy away of it or do it with poor compliance in understanding the terms and condition. Therefore, a trade-off is needed by the system manager, which is a part of the considerations with the system designer.
- Rule S-3: To gather the least amount of information from the user. More information has a higher risk to lose privacy information.

The following design rules are applicable to protection of privacy information.

- Rule S-4: To determine appropriate techniques and algorithms for the security in terms of privacy information (or just say security) or protection of privacy information. The examples of the techniques are the ones for encryption, certification and PKI.
- Rule S-5: To plan the strategies for security at the design stage in the areas of data storage and data processing. To store data locally has the advantage of high degree of security but may be poor in terms of data processing efficiency. While to store data remotely in cloud

has the advantage of data processing efficiency but may be with a low degree of security of data. Therefore, a trade-off is needed in determining the best way to store data and to process data.

- Rule S-6: To trade-off among authentication, authorization, encryption, storage strategy, usability, and computational capability for an acceptable security expectation. This actually proves the importance of requirement analysis in the area of privacy and privacy protection, as discussed in Chapter 3.
- Rule S-7: To develop a management strategy against the abuse of privacy throughout the whole data life time. The management strategies and policies need to consider human and cultural factors besides the techniques.

4.1.3 Energy aspect

Lack of sufficient battery power is a common problem in mobile systems. This will not only affect the task fulfillment but also create threats to the security (the risk to lose privacy information; see the previous discussion in Section 3.3.3). The following rules are applicable to the security concerning this aspect.

- Rule E-1: To keep the matter of energy in mind before running any critical process. Particularly, before performing a critical process such as heavy encryption, heavy algorithm calculation and mass data transfer, it is important to check the energy level (for example below 5% without plug in for recharge). The system interface should be designed to alert the user of the low battery situation.
- Rule S-2: To closely monitor the status of the battery considering the environment factors such as battery temperature to facilitate the managing system to more pro-active to the power shortage problem.

4.2 Design principles for a resilient security system

Axiom 1: Design redundant resources for critical security functions

From a system point of view, resources for mobile system include function, capacity and infrastructure. Redundancy of the system is focused on the function, capacity and infrastructure. The following are some design rules for redundancy.

- Rule R-1: To identify critical functions by means of resource duplication. Function redundancy does not mean to duplicate components. For instance, data transportation is a crucial function for mobile application. The tunnel for data transportation can be either 4G LTE or 2G GSM, and this forms a redundancy, i.e., data transportation either on 4G LTE or 2G GSM whichever is available.
- Rule R-2: To arrange the redundant capacity to meet the needs of the function. Capacity refers to the availability of electricity, computation, storage and bandwidth. Redundancy among the capacity is important to ensure the resilience of a mobile system. Taking the redundancy of electricity as an example, heavy loaded computation such as encryption for a block of data would occasionally consume much electric energy. For a resilient system, the availability of electricity redundancy, for instance, a recharger plugs in or the battery capacity should be above a certain level before the encryption process. Another example is the storage of critical data. Redundancy of storage capacity would be applied by establishing not only online database access process or function, but also having an access to local database as a duplicate safe guard to avoid the loss of the data when online service is unreachable.
- Rule R-3: Design the redundant infrastructure regarding the critical function. Mobile applications would involve not only the end user but also the back-end service parties such as Clouds, Web server, Data Base server and BSS, BTS etc. The infrastructure

redundancy is often necessary for the back-end to ensure the reliable and robust service existence; also, it makes the system possible to find critical materials against the failure of hardware.

Axiom 2: Effective management of redundancy

Mobile applications can be a simple app for fun or a sophisticated application integrated with many complex functions including user interface. Management of resources with redundancy is a sensible issue. For instance, a small app for users to order dishes in a specific restaurant merely needs the hardware and software of the hand-sets as well as OS. In contrast, Google Map as a sophisticated application needs not only local resources but also back-end resources including Data Base service for geographical map data, customer uploaded pictures and relevant comments information. A resilient mobile application requires a mechanism to manipulate the redundant resources; in other words, the system needs to decide when and how to reconfigure the system for the lost function by replacement of the mal-functioned sectors or training to work with a new configuration with the redundant resources.

- Rule R-4: The designer should first list the type, attribute and situation of the available redundant resources for the app.
- Rule R-5: The creation of clear instructions, which might be in form of algorithm or management rules, about how and when to establish the replacement as well as the way for training while the application has a failure.

Axiom 3: Monitoring of system performance

As the application is developed and operated under the mobile OS, it would be possible to keep sensing the key status of the system concerning the function of the security by sensors through a specific Application Program Interface (API). For example, Developer.mozilla.org (2018) introduces to call the method, e.g. navigator.battery, to enquire how many seconds before the end of battery recharging by visiting 'chargingTime' attribute. In this way, it is possible to decide whether or how to run a program regarding the battery status. Actually, the principle of monitoring makes the system have the idea about what is going on; further, it records the necessary status information in form of data for the later learning stage to forecast system errors.

Axiom 4: Error forecasting and handling mechanism

Errors for mobile applications come from both inside such as design, coding or app operation and outside including utilization of resources. At the design stage, the following rules may be followed:

- Rule R-6: To forecast and locate the possible vulnerability of algorithms and logical procedure as well as the relationship between different components of the applications.
- Rule R-7: To predefine the counter-measure for an error situation.

For instance, suppose that an application needs to access a remote database and the likelihood of success is low due to uncontrollable factors. In this case, to login to the remote database should be treated as a separate procedure in order to reduce the coupling of the login procedure with other functions of the remote database, which is conducive to improving the robustness of the main program.

Axiom 5: Needs software version control for evolution of the mobile app.

Mobile applications are relatively autonomous in that not many other programs rely on them. That gives an opportunity for them to make use of updating to counter-measure the external attacks. That is, by updating, the old diseased program is totally replaced by a new healthy one. To effectively manage the updating, the version control mechanism needs to be implemented in the application.

CHAPTER 5 CASE STUDY

5.1 Introduction

In order to illustrate how the design principles, proposed in Chapter 4, work, a demonstration mobile application (DEMO for short) was developed. DEMO is about the outpatient registration to visit doctor. For the rest of this chapter, Section 5.2 presents the requirement DEMO is supposed to meet. Section 5.3 illustrates the conceptual design of DEMO. Section 5.4 presents the scenario DEMO supports.

5.2 Requirement analysis

A systematic design procedure, as advanced very recently by Dai (2019), was followed. The step is to understand the requirement DEMO was supposed to fulfill based on the requirement model (Dai, 2019). Table 5.1 gives key requirements for DEMO, including the requirement for protection of privacy. The first column shows the type of requirements: functional and non-functional (or constraint). It is noted that in this column, a particular type of requirement, called resilience, is listed separately from the non-functional requirement. The second column gives a description to each requirement, and the third column indicates the ways and patterns to satisfy the specific requirements. It is noted that DEMO was built upon the program developed by Dai (2016), where there are two main functions: shortest waiting time and shortest distance between the location of patients and the location of hospitals. In DEMO, the function related to the resilient protection of privacy is added.

Table 5.1 Requirements analysis along with devices

Type	Content	Application
Functional requirement	Shortest outpatient waiting time	Algorithm
	Select one of the clinics among the recommends which have approximate location, waiting time or distance.	UI design, function module
	Navigation designated	UI design, function module
Non-functional requirement	Simple and clearly instructed operation	UI design, function arrangement
	Gather user information for registration and illness symptom for patient arrangement	UI design, database access, design principles for privacy
	Protection of privacy	principles for privacy
	Security concerns	principles for security
System requirement	Resilient security of privacy info.	principles for resilience

Specification of the app development environment

A laptop with Windows 10 OS is taken to develop the DEMO. Android SDK is chosen. Eclipse acts as the primary development platform integrated with JDK. The app is expected to operate within Android 5.0 or above. FURTHER, the app is also expected to operate in 3G/4G or WIFI wireless environment with the Google Map being preloaded.

5.3 Conceptual design of DEMO

Figure 5.1 is a brief description of the operational procedure of DEMO. A patient should first register in the DEMO app with a provincial health card. Registration or login information is then transferred via the wireless connection to the Internet and authenticated by a remote Data Base (DB), owned by a corresponding authority for the management of healthcare information. Once the patient's input information matches those stored in the Public Healthcare DB, the patient is required to give a brief description of the symptom. After that, DEMO will generate several candidate hospitals to the patient, e.g., three, and let the patient make a final choice. Once the decision is made, the program will again access the Public Healthcare DB to double check the authentication as well as to record the information in the Public Healthcare DB. At the same time, there will be a navigation option which is initiated by OS through Google Map navigation service.

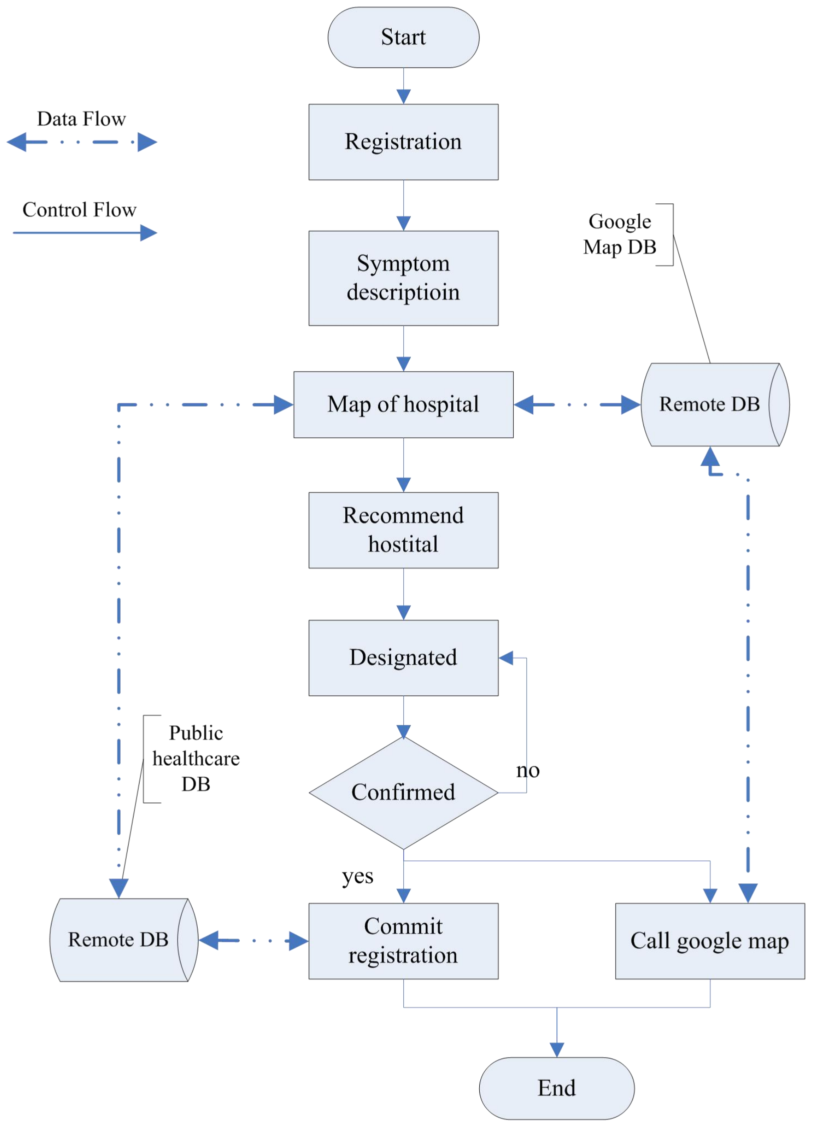


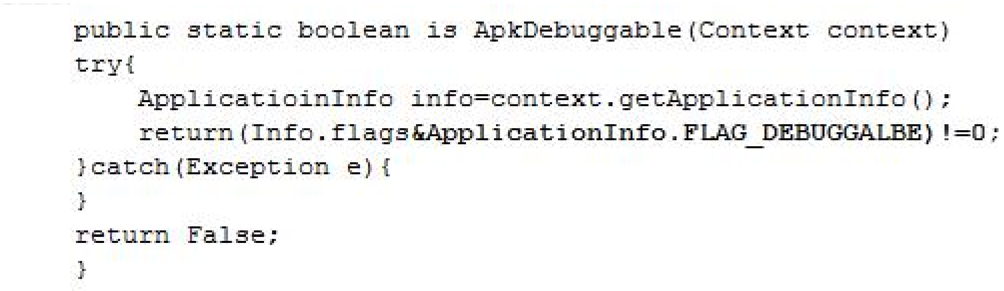
Figure 5.1 Program flow of the application

5.4 Resilient protection of privacy information

5.4.1 Application of the design principles

The infrastructure point of view

Application of Rule I-1 discussed in Section 4.1.1. AndroidManifest.xml is an entrance file or a framework to define the operational status and references for the application. As the application is developed for Android 6.0 OS and later versions, it is necessary to make an additional setup for the AndroidManifest.xml. First, set the value of allowBackup to ‘false’ in order to avoid an unauthorized copy of the application data by evil intention by enabling the USB debugging option. Second, set the value of Deguggable to ‘false’ to reduce the likelihood of being acquired and falsified for sensitive information and even program logistic process by use of Java kits, such as JDB, to steal the user password or access data bypassing authentication process. Figure 5.2 is a program fragment intended to get the value of the Debuggable parameter.



```
public static boolean is ApkDebuggable(Context context)
try{
    ApplicatioinInfo info=context.getApplicationInfo();
    return(Info.flags&ApplicationInfo.FLAG_DEBUGGALBE)!=0;
}catch(Exception e){
}
return False;
}
```

Figure 5.2 Code for the value of the Debuggable parameter

Application of Rule I-3 discussed in Section 4.1.1. Native development kit (NDK) is a toolset that lets the developer implement parts of the app in native code. Further, C++ and C languages have an excellent capability against decompilation, which may reduce the potential risk of decompiling of source codes as well as the crack of the application. As a result, C++ was chosen as one of the primary programming languages together with native development kit (NDK) to realize the crucial function, for example the login module illustrated in Figure 5.3 and user registration module.

Application of Rule I-2 discussed in Section 4.1.1. The user-defined soft keyboard provides safe inputs for sensitive data in the process of login and registration. Users are required to input the password during those processes. There may be a hazard to encounter stealing and recording of the password when using the third party input method or default soft keyboard. As a result, the application chooses to develop a user-defined keyboard, as illustrated at image in Figure 5.3 with random distributed key words to ensure the security of password input.

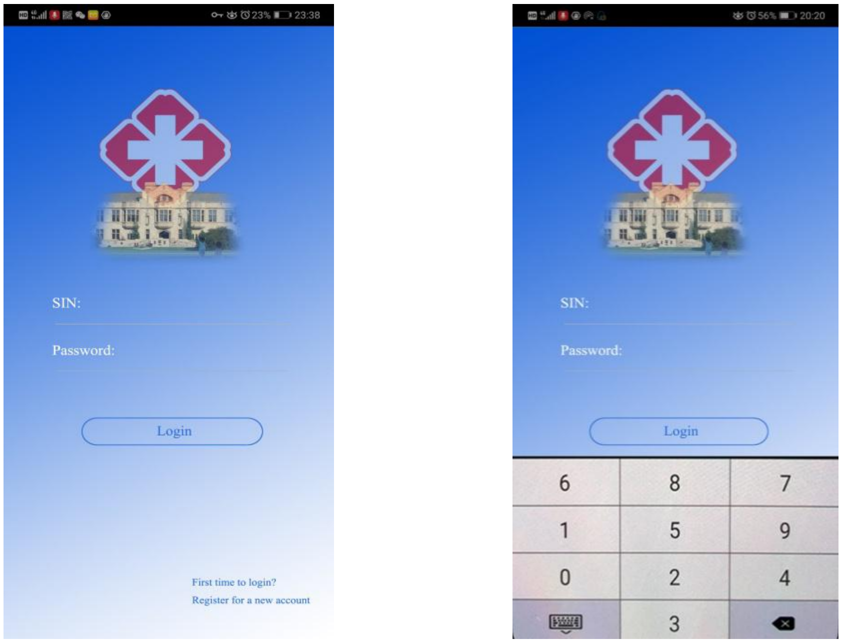


Figure 5.3 Screenshot of the app at login and after login

The substance point of view

Identification of the privacy related information: application of Rule S-1 discussed in Section 4.1.2. Due to the method introduced in Section 3.2, Figure 5.4 shows an entity-relationship model for the patient and hospital in the context of the patient registration. In Figure 5.4, the following attributes, such as the patient’s telephone Number (Tel.), name, personal health No. and personal geographical location (**PGL**) are captured. The attributes of the action (Register) include patient's symptom; besides, both the time that the patient lodges registration and the location are marked as the time and registration geographical location (**RGL** for short) in Figure 5.4. For hospital, the attributes such as the hospital ID, hospital geographical location (**HGL** for short) and address are identified.

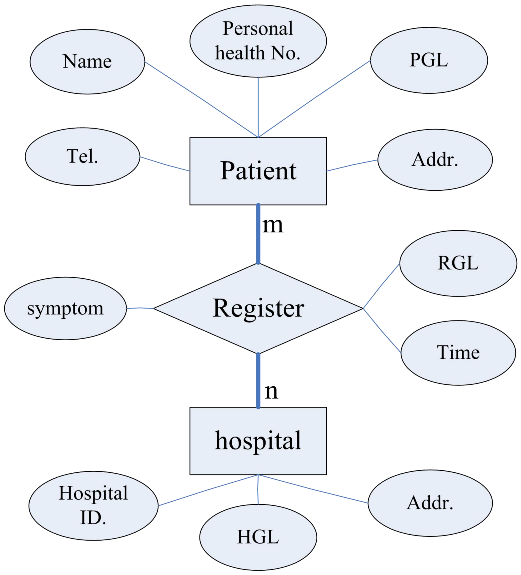


Figure 5.4 Entity Relationship between patient and hospital

The least information: application of Rule S-3 discussed in Section 4.1.2. It is noted that the PGL in Figure 5.4 is being monitored by the app during run time. One of the most critical issues is what part of the location related data should be transferred and stored to avoid compromise of the personal privacy. According to the least information principle discussed in Section 4.1.2, PGL should not be stored or transferred as it is closely related to the personal identity. By contrast, the personal location information regarding registration, i.e., RGL in Figure 5.4, can be stored either in the local storage or transferred to a remote server and database, as the evidence of registration.

Management for data storage, use and transfer: application of Rule S-5 discussed in Section 4.1.2. Trade-off is made among security, store efficiency and cost effective, and eventually the local storage was chosen as a media to store processing data. For instance, patient's personal health No.

and symptom description are stored in SQLite, which is a light weight and default system database with Android OS for smart phone, to avoid the risk of being hacked compared with the approach to store them on the portable device such as SD card.

After that, two aspects of the management rules are established. First, for the data stored in the local storage, it is crucial to ensure that only the data owner or device owner will have an access to the relevant data by authentication strategy and technique; furthermore, data access should not be permitted without the owner's formal consent.

Second, for the data that have been uploaded to the remote database, the tools such as identity shielding technique are applied based on the least data provision principle. By the way, this technique can also make some difficulty in an attempt to infer the individual’s personal information through the data mining technique.

Authentication and certification: application of Rule S-4 discussed in Section 4.1.2.- Authentication is established with the patient's personal health No. together with password or finger print. Besides, when the app proceeds to transfer the privacy data of the patients, such as personal health No. and symptom description to the server, the security of the communication channel is ensured by certification. HTTPS is a transfer protocol for secured communication over a computer network and it is widely used on the Internet. The certification is performed at the occasion when a mobile terminal is prepared to transfer the encrypted data, such as login and symptom description content. Keytool, which enables to manage users’ public/private key pairs and certificates, integrated with JDK was employed in DEMO to generate a free certification function for HTTPS, and at the same time, Tomcat Servlet, as a service provider, was established to support the use of HTTPS.

Convey of the relevant information and consent: application of Rule S-2 discussed in Section

4.1.2. Figure 5.5 shows the two sites: the app side (left side) and the remote database (right side). From Figure 5.5 it can be seen that the information content transferred to the remote database is less than the information content on the left side, which is the result based on the least data provision principle. After that, a legal consent form is presented to the user for later action.

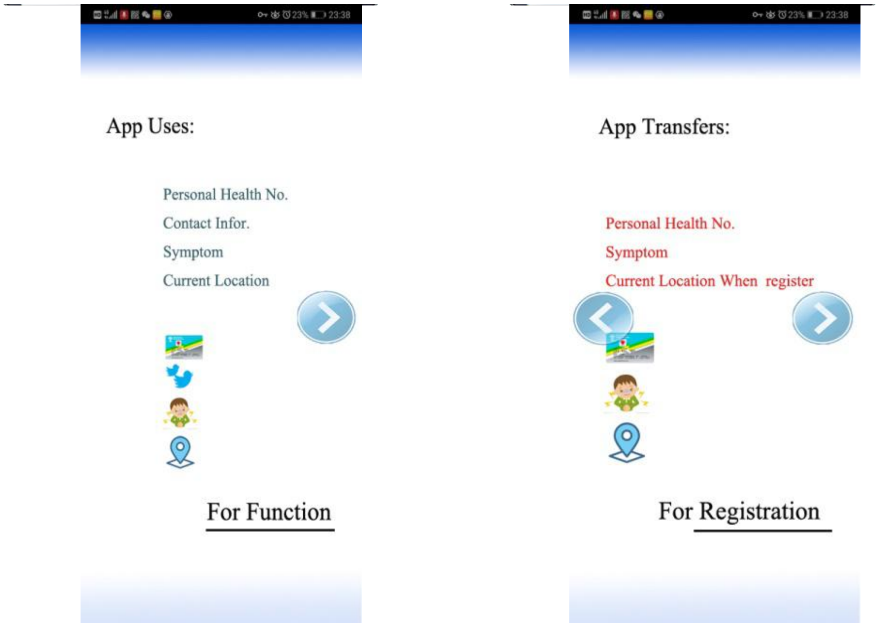


Figure 5.5 Patterns for convey of privacy information to the users

The energy point of view

Energy monitoring: application of E-1 and E-2 discussed in Section 4.1.3. To create an object, whose name is BroadcastReceiver to monitor the state of system event, in particular ACTION_BATTERY_CHANGED, to get the real time information of battery by acquiring the value of the Intent to calculate the quantity level of the battery capacity. In this way, it could be possible for the user to decide whether to allow the service of heavy energy consumption to operate or not, e.g., the navigation by Google Map.

5.4.2 The resilient protection of the privacy information

Design of the necessary redundant resource

Application of Rule R-1 discussed in Section 4.2. For MNS, Mobile Outpatient Booking app was expected to run on the OS platform. Majority of redundancy resources were possessed and administrated by the OS. For instance, computation and storage facility were taken as the capacity redundancy, wireless connection and transfer methodology as the function redundancy, users with two or more smart phones in pocket as the infrastructure redundancy.

In the case of design of the PSR mobile application, the algorithm for recommendation of candidate hospitals, such as 'recommend hospital' and 'map of hospital' in Figure 5.1, relies on the data from the remote database. A pre-loaded map whose data was stored locally, as opposed to the locally stored information of hospitals was of resource redundancy. Also, the critical information such as the current geographic data, symptom description and information of designated hospital were made dual-backup, i.e., one copy at the local database and one copy at the remote database.

Monitoring the state of the system function

This is related to application of Axiom 3 proposed in Section 4.2. The redundancy resource includes the pre-loaded map and hospital information, real time geographic data, symptom description and information of hospitals. The pre-loaded map and hospital information should be loaded by the program at the moment when the wireless connection is out of service. This thus requires that the system should have a monitoring function in place, for example requesting the state of ActionListener port after performing the enableNetwork method (a tool available at Android) to monitor whether the Internet connection is established or not. Decision on triggering the system to use a redundant resource was based on the result of the monitoring function.

Error forecasting and handling mechanism

This is related to application of Axiom 4 discussed in Section 4.2. In the design process, potential errors were estimated, which are for example: (1) the error occurs while performing the database connection request; (2) users forget the password or personal health number when performing the login into the system; (3) users are unable to unlock the mobile devices; (4) power is off while performing a critical process.

Software version control

This is related to application of Axiom 5 discussed in Section 4.2. For DEMO, only the first version of the system was available. However, the continuous improvement was in the mind of the developer, which means that updated versions of DEMO were expected. A data model for the version control was available in the system, which is surrounding the data model of the system and describes the update taken and rational behind the update in generating each new version of the software.

5.5 Summary

This chapter presented the development of a DEMO to illustrate how the design principles described in Chapter 4 can be applied to provide a resilient secured application regarding privacy. The DEMO is about the doctor appointment booking. Codes for some important parts of the DEMO system, e.g., database connection, call Google map service for navigation and the definition of customized keyboard, are listed in Appendix D for reader’s convenience.

CHAPTER 6 CONCLUSION

6.1 Overview

The research in this thesis was motivated by the recent attention to the privacy issue in mobile applications. In general, there lacks the knowledge for how a mobile application is developed by the systematic consideration of the privacy issue. In particular, the concept such as resilience in the context of privacy protection is not clear in the literature of general privacy protection and mobile applications. The overall objective of the research in this thesis (or this thesis) was then to advance the understanding of privacy, privacy protection, and resilience in terms of privacy protection. There are three specific objectives defined for this study, namely:

- **Objective 1** is to have a more comprehensive understanding of the concepts of privacy, security and resilience (**PSR** for short) along with their relationship in the context of mobile applications.
- **Objective 2** is to develop the principles of design of a mobile application system with a satisfactory PSR.
- **Objective 3** is to develop a demonstration system (PSR demo for short) to illustrate how the principles of design can be applied.

These objectives have been achieved. In particular, Chapter 2 presented a comprehensive review and analysis of the literature to further justify the need of researching the proposed three objectives. A new understanding of privacy, security in terms of privacy protection, and resilience in terms of security has been proposed based on the idea that the privacy issue should be viewed around the general knowledge architecture, FCBPSS in particular in Chapter 3. This part is related to Objective 1. Chapter 4 presented the design principles based on the theory of

PSR developed in Chapter 3 for developing a mobile application system with an attention to resilient privacy protection. This part is related to Objective 2. In Chapter 5, a DEMO system regarding a mobile system for patients to book the appointment for doctor visiting was developed by applying design principles presented in Chapter 4. This part is related to Objective 3.

The main conclusions drawn from this study are: (1) the PSR theory, as developed in this thesis, is useful in that it builds the relationship between the resilience and privacy protection and it gives a complete picture of privacy and its protection to any particular application, owing to the system perspective to privacy; (2) the design principles for a mobile applications considering privacy protection are effective, according to the experience gained in constructing a DEMO.

6.2 Contribution

The main contribution of this thesis is the development of the concept of PSR, especially the relationship among privacy (P), security (S), and resilience (R), which is called in short PSR theory, and a set of design rules to develop a mobile application for systematic and complete privacy protection. In the current literature, the concept of resilience is not connected with privacy protection at all. The understanding of privacy, privacy protection and resilience in terms of privacy protection was made first in this thesis. The other main contribution is the concept of privacy, defined from a system perspective and data relativity principle, which allows classifying privacies in a complete and systematic manner.

6.3 Future Work

There is an important limitation with the present work. The DEMO system is too simple and has not demonstrated all the features of the design principles, and therefore the future work is directed to the construction of a more comprehensive DEMO. Another future work is to study the relationship of privacy with other system performance attributes, e.g., usability. Resilience in the aspect of privacy protection is in its nature resilience of an information system, as privacy is about data. In enhancing the resilience of an information system, the data redundancy is a common approach. However, data redundancy has an inherent problem, i.e., data inconsistence, which eventually causes some problem in data integrity. How to manage data redundancy in light of the enhanced resilience of privacy protection while keeping data integrity is a problem worthy of study in future. Finally, how to measure the quality of protection of privacy and how to test a mobile application system regarding its capability of privacy protection.

REFERENCES

Akiki, P., Bandara, A., & Yu, Y. (2014). Adaptive model-driven user interface development systems. *ACM Computing Surveys, 47*(1), article 9.

Allen, & Anita, L. (2003). Privacy isn't everything: Accountability as a personal and social good. *Alabama Law Review, 54*(4), 1375-1391.

Almedom, A. (2015). Understanding human resilience in the context of interconnected health and social systems, Whose understanding matters most. *Ecology and Society, 20*(4), 40.

Alpár, G., Hoepman, J., & Siljee, J. (2011). The Identity Crisis: Security, Privacy and Usability Issues in Identity Management. *Arxiv.Org, 1101.0427*, 1-15.

Bai, W., Li, Z., & He, X. (2018). Study on Privacy Protection of College Students' Mobile Social Network. *Value Engineering, 37*(27), 215-216.

Berkes F., & Ross H. (2013). Community Resilience, Towards and Integrated Approach. *Society & Natural Resources, 26*(1), 5–20.

Bernard, G. (2017). 'Security Apps Know The Difference Between Native, Web And Hybrid Apps'. [Online]. Available at: https://www.linkedin.com/pulse/security-apps-know-difference-between-native-web-golllotti-cpp. [Accessed 2 Oct. 2018].

Chen, H., Beaudoin, E., & Hong, T. (2017). Securing online privacy: An empirical test on Internet scam victimization, online privacy concerns, and privacy protection behaviors. *Computers in Human Behavior,* (70), 291-302.

Chen, J., Su C., Yeh, K., & Yung, M. (2018). Special Issue on Advanced Persistent Threat. *Future Generation Computer Systems*, (79), 243-246.

Chen, S. (2018). 'A survey of patients for state owned hospitals in Shanghai shows: long outpatient waiting time stands at the lowest satisfaction rate'. [Online]. Available at: https://www.thepaper.cn/newsDetail_forward_1931409. [Accessed 2 Feb. 2019].

Commonwealthfund. (2017). 'Commonwealth Fund Survey 2016'. [Online]. Available at: https://www.commonwealthfund.org/interactives-and-data/surveys/biennial-health-insurance-surveys/2017/2016-biennial-health-insurance-survey. [Accessed 12 Dec. 2018].

Conradi, J., Busch,O., & Alexander, T. (2015). Optimal touch button size for the use of mobile devices while walking. *Procedia Manufacturing*, (3), 387–394.

Dai, F. (2016). *On Development of a Green Web-based System for Reducing Waiting Times of Outpatients.* MSc Thesis, Division of Biomedical Engineering, University of Saskatchewan, Canada.

Damiani, M., & Cuijers, C. (2013). Privacy Challenges in Third-party Location Services. *Process of the 14th IEEE International conference on Mobile Data Management*．Piscataway，NJ：IEEE.

Developer.apple.com. (2009). 'Xcode 10'. [Online]. Available at: https://developer.apple.com/xcode/. [Accessed 28 Oct. 2018].

Developer.mozilla.org. (2018). 'Navigator.battery'. [Online]. Available at: https://developer.mozilla.org/en-US/docs/DOM/window.navigator.battery. [Accessed 3 Oct. 2018].

Dreamincode.net. (2018). 'TCP/IP Protocol For Cellular Communication'. [Online]. Available at: http://www.dreamincode.net/forums/topic/39492-tcpip-protocol-for-cellular-communication/. [Accessed 6 Oct. 2018].

Elmasri, & Ramez. (2016). Fundamentals of Database Systems. *Reading,* 73–74.

En.wikipedia.org. (2018). ARPANET. [Online]. Available at:

https://en.wikipedia.org/wiki/ARPANET . [Accessed 6 Jan. 2019].

En.wikipedia.org. (2018). 'C++'. [Online]. Available at: https://en.wikipedia.org/wiki/C%2B%2B. [Accessed 18 Oct. 2018].

Evangelia, K., Christos, K., Haralambos, M., & Stefanos, G. (2014). Privacy as an Integral Part of the Implementation of Cloud Solutions. *Security in Computer Systems and Networks The Computer Journal, 58*(10), 2214-2224.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Elements of Reusable Object-Oriented Software. Addison-Wesley Professional.

Geetha, A., & Sreenath, N. (2015). Review of Security Threats and its Countermeasures. *Advances in Natural and Applied Sciences, 9*(6), 421-425.

Gerald, L., & Barlow. (2002). Auditing hospital queuing. *Managerial Auditing Journal, 17*(7), 397-403. doi:10.1108/02686900210437507.

Gs.statcounter.com. (2018). 'Mobile Operating System Market Share Worldwide - October 2018'. [Online]. Available at: http://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-201709-201810. [Accessed 13 Sep. 2018].

Hegney, D., Buikstra, E., Baker, P., Rogers-Clark, C., Pearce S., Ross H., & Watson-Luke, A. (2007). Individual resilience in rural people, a Queensland study. *Rural and Remote Health, 7*(4), 1–13.

Holling, C. (1973). Resilience and stability of ecological systems. *Annu Rev EcolSyst*, (4), 1–23.

Howard, J. (2003). 'Common Design Patterns for Android with Kotlin'. [Online]. Available at: https://www.raywenderlich.com/470-common-design-patterns-for-android-with-kotlin. [Accessed 10 Oct. 2018].

Huang, D., Zhang, X., Kang, M., & Luo, J. (2014). Mobicloud, building secure cloud framework for mobile computing and communication. *Proceedings of the Fifth IEEE International Symposium on Service Oriented System Engineering*, SOSE, pp. 27–34.

Imran, H., & Bhagyashree B. (2014). Cloud Information Security Using Third Party Auditor and Cryptographic Concepts. *International Journal of Application, 3*(11), 1124-1129.

Ja, A., & Chaudhari. (2015). English Security Aware Adhoc on Demand Distance Vector Routing Protocol in Vehicular Adhoc Network Dhao. *International Journal of Innovative Research in Computer and Communication Engineering, 2*(12), 7258-7263.

Kim, I., & Jo, J. (2015). Performance comparisons between thumb based and finger-based input on a small touch-screen under realistic variability. *International Journal of Human-Computer Interaction, 31*(11), 746–760.

Kiswendsida, A., Simon, E., & Fre´de´ric, V. (2013). *Engineering Applications of Artificial Intelligence*, (26), 24–34.

Laplante, P. (2009). *Requirements Engineering for Software and Systems (1st ed.)*. Redmond, WA: CRC Press.

Lauren, V. (2015). More hospitals Choosing Wisely. *CMAJ, cmaj.*109-5078. doi:10.1503/cmaj.109-5078.

Lee, A., & Ra, I. (2015). Network resource efficient routing in mobile ad hoc wireless networks. *Telecommunication Systems, 60*(2), 215-223.

Li, B., Gao, Z., & Cui, S. (2018). Over View of Privacy Disclosure and Privacy Protection Methods Based on Mobile Terminal Location. *Journal of Information Security Research, 4*(8), 698-703.

Li, Y., Han, F., Huang, S., & Luo, Y. (2018). A content-based goods image recommendation system. *Multimedia Tools and Applications, 77*(4), 4155-4169.

Lin, Y., & Zhang, W. (2004). Towards a novel interface design framework, function-behavior-state paradigm. *International Journal of Human Computer Studies, 61*(3), 259-97.

Lin, Y., Zhang, W., & Watson, G. (2003). Using Eye Movement Parameters for Evaluating Human-Machine Interface Frameworks under Normal Control Operation and Fault Detection Situations. *International Journal of Human Computer Studies, 59*(6), 837-873.

Lin, Y., Zhang, W., Koubek, R., & Mourant RR. (2005). On integration of interface design methods: Can debates be resolved. *Interacting with computers, 18* (4), 709-722.

Liu, H. (2016). Investigation and Forensics against Fake BTS based Tele-fraud Crime. *Police Technology*, (02), 8-10.

Liyanage, M., Ahmad, I., Ylianttila, M., Santos, J., Kantola, R., Perez O., Itzazelaia, M., de Oca, E., Valtierra, A., & Jimenez, C. (2015). Security for future software defined mobile networks. *Proceeding of the 9th International Conference on Next Generation Mobile Applications Services and Technologies (NGMAST)*. IEEE, pp. 1–9.

Luis, V., Carlos, F., & Miguel, G. (2007). The role of security, privacy, usability and reputation in the development of online banking. *Online Information Review, 31*(5), 583-603.

Lundberg, J., & Johansson, B. (2015). Systemic resilience model. *Reliability Engineering and System Safety,* (141), 22–32.

Luo, S., & Zhou, Y. (2015). Effects of smart phone icon background shapes and Figure/background area ratios on visual search performance and user preferences. *Frontiers of Computer Science, 9*(5), 751–764.

Malmir, A., & Malmir, M. (2015). Government’s civil liability towards individuals’ privacy in cyberspace. *International Journal of Law and Management, 57*(2), 98-106.

McComb, M. (2015). 'Facebook Enables Native App Development in JavaScript with React Native'. [Online]. Available at: https://www.infoq.com/news/2015/02/facebook-announces-react-native. [Accessed 21 Nov. 2018].

Moyers, B., Dunning, J., Marchany, R., & Tront, J. (2010). Effects of Wi-Fi and bluetooth battery exhaustion attacks on mobile devices. P*roceedings of the 43rd Hawaii International Conference on System Sciences*, HICSS, IEEE, pp. 1–9.

Mukherjea, & Sougata. (2017). Mobile Application Development, Usability, and Security. *Heyshey, IGI Global, 95*(112), 117-136.

Mukherjea, Sougata. (2017). Mobile Application Development, Usability, and Security. *Heyshey: IGI Global, 95*(112), 117-136.

National Research Council. (2010). Steering Committee on the Usability, Security, Privacy of Computer Systems. *Toward better usability, security, and privacy of information technology : Report of a workshop*. Washington, D.C.: National Academies Press.

Osseiran et al. (2014). Scenarios for 5G mobile and wireless communications, the vision of the METIS project. *IEEE Communications Magazine*. *52* (5), 26–35. doi:10.1109/MCOM.2014.6815890.

Park, J., Han, S., & Park, Y. (2017). Human complementary menu design for mobile phones. *IFAC Proceedings Volumes, 40*(16), 67–72.

Phillips P., Yates, A., Hu, Y., Hahn, C., et al. (2018). Face recognition accuracy of forensic examiners, super recognizers, and face recognition algorithms. *Proceedings of the National Academy of Sciences of the United States of America, 115*(24), 6171-6176.

Punchoojit, L., & Hongwarittorrn, N. (2017). Usability Studies on Mobile User Interface Design Patterns, A Systematic Literature Review. *Advances in Human-Computer Interaction,* (2017), 1- 22.

Rachel Appel. (2014). 'Modern Apps : Mobile Web Sites vs. Native Apps vs. Hybrid. Apps'. [Online]. Available at: https://msdn.microsoft.com/en-us/magazine/dn818502.aspx. [Accessed 2 Jan. 2018] .

Raviteja. (2007). 'iOS Design Patterns'. [Online]. Available at: https://www.e-consystems.com/Articles/iOS/iOS-Design-Patterns.asp. [Accessed 10 Oct. 2018].

Rebecca, S. (2016). A user-centered model for designing consumer mobile health (mHealth) applications (apps). *Journal of Biomedical Informatics,* (60), 243-251.

Reiter, M., & Rubin, A. (1998). Crowds: Anonymity for Web transactions. *ACM Transactions on Information and System Security (TISSEC), 1*(1), 66-92.

Reiter, M., & Rubin, A. (1998). Crowds: anonymity for Web transactions. *ACM Transactions on Information and System Security (TISSEC), 1*(1), 66-92.

Roberto, D., Gabriele, O., Claudio, S., & Gene, T. (2013). United We Stand, Intrusion Resilience in Mobile Unattended WSNs. *IEEE Transactions on Mobile Computing, 12*(7), 1456-1468.

Sandoval, K. (2016). 'What is the Difference Between an API and an SDK?'. [Online]. Available at: Nordic APIs Blog. Nordic APIs AB. [Accessed 7 Oct. 2018].

Shiels, M. (2018). 'Phone sales hit by Sidekick loss'. [Online]. Available at: http://news.bbc.co.uk/2/hi/technology/8303952.stm. [Accessed 6 Sep. 2018].

Shoavi, O. (2015). 'The All-Star Winners of Mobile App Tools (SDKs)'. [Online]. Available at: http://code.google.com/p/gaeproxy/. [Accessed 21 Oct. 2018].

Solove, D. (2002). Conceptualizing privacy. *California law review, 90*, 1087-1132.

Sun.com. (2003). 'JAVASOFT SHIPS JAVA 1.0'. [Online]. Available at:

http://www.sun.com/smi/Press/sunflash/1996-01/sunflash.960123.10561.xml. [Accessed 16 Sep. 2018] .

Sun, Z., Zhang, B., Cheng, L. & Zhang, W. (2011). Application of the redundant servomotor approach to design of path generator with dynamic performance improvement. *Mechanism and Machine Theory, 46*(11), 1784-1795.

Tarik, T., Adlen, K., & Bruno, S. (2016). On Service Resilience in Cloud-Native 5G Mobile Systems. *IEEE Journal on Selected Areas in Communications, 34*(3), 483-496.

The 39th report of current Chinese Internet usage and statistics. (2017). [Online]. Available at: http://www.cac.gov.cn/cnnic39/index.htm. [Accessed 16 Oct. 2018].

Thomas Fox-Brewster. (2018). 'Facebook Is Playing Games With Your Privacy And There's Nothing You Can Do About It'. [Online]. Available at: https://www.forbes.com/sites/thomasbrewster/2016/06/29/facebook-location-tracking-friend-games/#1b58eb6d35f9. [Accessed 12 Aug. 2017].

Unhelkar, B., & Murugesan, S. (2010). The enterprise mobile applications development framework. *IT Professional, 12*(3), 33–39.

Wang, Y., & Huang, G. (2018). Research on Privacy Protection Algorithm for Mobile Users of Road Network. *Application Research of Computers, 35*(10), 3078-3081.

Wang, J., Wang, H., Ding, J., Furuta, K., Kanno, T., Ip, W. & Zhang, W. (2016). On domain modeling of the service system with its application to enterprise information systems. *Enterprise Information Systems, 1*(10), 1-16. doi:10.1080/17517575.2013.810784.

Woods, D. (2015). Four concepts for resilience and the implications for the future of resilience engineering. *Reliability Engineering & System Safety*, (141), 5–9.

Xu, M., Cheng, Y., & Yin, Q. (2018). Toward a Resilient System: The Inspiration from Information Security Management Appliance. *2018 13th Asia Joint Conference on*

Information Security (AsiaJCIS). August 8-9, Guilin, China. pp. 42-46. doi: 10.1109/AsiaJCIS.2018.00016

Xue, L., Liu, C., Lin, Y., & Zhang, W. (2015). On Redundant Interface: Concept and Design Principle. *Proc. 2015 IEEE/ASME International Conference on Advanced Intelligent Mechatronics.* July 8-11. Busan, South Korea.

Zhang, W., & Van, L. (2012). Toward a resilient manufacturing system. *CIRP Annals - Manufacturing Technology*, (60), 469–472.

Zhang, W., & Lin, Y. (2010). On the principle of design of resilient systems-application to enterprise information systems. *Enterprise Information Systems, 4*(2), 99-110. doi:10.1080/17517571003763380.

Zhang, W. (1994). An Integrated Environment for CAD/CAM of Mechanical Systems, Ph.D. thesis, printed by Delft University of Technology, The Netherlands, ISBN 90-370-0113-0, pp. 1-263.

Zhang, X., Schiffman, J., Gibbs, S., Kunjithapatham, A., & Jeong, S. (2009). Securing elastic applications on mobile devices for cloud computing. *Proceedings of the 2009 ACM Workshop on Cloud Computing Security,* CCSW’09, ACM, New York, NY, USA, pp. 127–134.

APPENDIX A

DESIGN PATTERNS

According to Gamma et al. (1994), "a design pattern is a description of communicating objects and classes that are customized to solve a general design problem in a particular context." The reason why we discuss design pattern here is to find an existing pattern solution that fits the current design requirements and to enhance the development efficiency of the app. Design patterns for mobile system are categorized by Howard (2003) into three groups which are creational, structural and behavior catalogue. Among which, creational catalogue includes builder, dependency injection and singleton patterns etc. Structural catalogue includes adapter and facade patterns etc. According to (https://sourcemaking.com/design-patterns-ebook), behavioral catalogue includes Command, Observer, Model View Control (MVC for short), Model-View-View-Model (MVVM) and Clean Architecture patterns etc., respectively. The following content would discuss some of the design patterns which is more close to the development of the App concerned with this thesis.

Singleton pattern, a member of creational catalogue, is simple for application design. It ensures that a class has only one instance and provides a global point of access to that instance. By means of Private Construction method and initiating Static Instance, it reduces the use of memory resource as well as frequent creation and destruction of the relevant Instance, for example frequent reloading of web page cache. Besides, Singleton Pattern is benefit to avoid multiple consumption for both computing and storage resource such as interruption for file writing or reading.

For design of IOS app, the Cocoa (Touch) version of MVC comprises the Composite, Command, Mediator, Strategy and Observer patterns which base on the theory of layered function distribution. As a result it is necessary to make a brief discussion of MVC pattern which belongs to the Behavior catalogue. MVC was first initiated in desktop applications. For this pattern, model (M) refers to the organization of application data, while view (V) stands for User Interface and control (C) represents the control of model and view objects. In case of mobile app, control object, which bridges M and V, usually takes in charge of the acquirement and preconditioning of application data before sending the data to V object to display on user interface; besides, when data updated, C object needs to refresh the display of the latest content regarding the current status of the data. During the above process, because M and V are isolated entirely, it would be possible to update to another View pattern when application need to be evolved. Briefly, M objects describe and maintain the app data by means of defining strategy, logic and algorithm. The aim of M objects is to isolate explicit association among user interface used for presenting and editing it. In contrast, V objects primarily focus on the interaction with app user and how to show the corresponding information to the user as well. In this way, V objects appear to have low coupling with M objects which enables high reusability for MVC pattern.

For Structural pattern group, Facade act as a cover, which is similar to the interface, for a complex system or class in order to be simplified to get through the entry point, the cover, without aware of detailed structure underneath the facade objects. The aim of Facade pattern is not to provide new interface for subsystem, but to reduce the unnecessary interaction between outside users and the subsystem which lower the system coupling. This pattern actually simplifies the way to facilitate existing functions of the subsystem or modules. In other words, the function of Facade objects is well encapsulated to provide service for outside users.

APPENDIX B

FCBPSS

According to Lin and Zhang (2003), FCBPSS(Function-Context-Behavior-Principle-State-Structure) which is the involvement of FBS methodology provides us a more detailed and thorough description about the way to understand the process of engineering design for specific function requirements. In addition, Zhang and Lin (2010) apply the ontology of FCBPSS as a general knowledge on design of enterprise system which takes performance, resilience and sustainability as the key factors. The methodology had been testified to be an effective guidance to address the problem domain for supply chain from a system's point of view. As had cited above, FCBPSS appears to be an efficient method to help designer comprehending and drawing the big picture of the whole system in order to address the specific issues.

APPENDIX C

MOBILE TERNINAL AND WIRELESS BACKEND

Mobile terminal (MT) is constructed by two separated components, the first one is mobile station or simply cell phone, and another is mobile equipment or subscriber identification module (SIM). SIM is an integrated circuit card to identify and authenticate subscribers on mobile terminals. According to the description in Dreamincode.net (2018), wireless backend is composed by The base station subsystem (BSS) (colored green in Figure 3.5) and the network switching Subsystem (NSS) (colored yellow in Figure 3.5).The BSS is constructed by the following three components:

The base transceiver station (BTS). By use of necessary hardware and infrastructure, BTS acts as the terminal executor both receives and transfers data between mobile terminal and base station controller. In addition, BTS also report the current status such as real-time operation resource, equipment status etc.

Base station controller (BSC). This component plays the key role in the management on forward BTS. It is in charge of several BTSs, and maintains the real-time load distribution data bank in order to balance the resource among the BTSs which are controlled by it.

The network switching subsystem is illustrated with dark yellow color in the Figure 3.5 stands behind the forward structure elements, such as mobile terminals and BSSs. The function of NSS and the corresponding components are responsible for all the call processing, controlling and data bank referring functions which are necessary to examine the authentication, to make set-up the call, to encrypt the data and to control roaming. The components include mobile services switching centre (MSC for short), gateway mobile services switching centre (GMSC), home location register, visitor location register, equipment identity register etc. States for the first two which are MSC and GMSC are a little different. MSC provides reliable data link and channels switch; while GMSC ensures the reliable data switching with regard to different type of networks and patterns, for instance, data switches from fixed network to mobile network or mobile network to mobile network. The rest three different registers belong to kind of dynamic information stored in database which can be visited and shared by means of different visiting methodology and authentication.

Briefly, above three instruments are only fractional parts of the whole mobile network structure. The real picture of the mobile network is consisted by abundant of those elementary units which connected to each other to provide the desired service.

APPENDIX D

CODES FOR CRUCIAL PROCEDURES OF THE APP

```
/*Database connection and interaction*/
package com.mtjsoft.www.myapplication.data;
import android.util.Log;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

/* @author ming 2018-10-9 10:00:48
  *                Created by Administrator on 2018/10/9.
  */                android access sqlserver database



public class MtjServerDatabaseTools {
        private static String user = "database username";
        private static String password = "access password";
        private static String DatabaseName = "database name";
        private static String IP = "Ip address of database server";

        private static String connectDB = "jdbc:jtds:sqlserver://" + IP + ":1433/" + DatabaseName +
";useunicode=true;characterEncoding=UTF-8";

        private static Connection conn = null;
        private static Statement stmt = null;

        private static Connection getSQLConnection() {
                Connection con = null;
                try {
                        Class.forName("net.sourceforge.jtds.jdbc.Driver");
                        /*Database connection object*/
                        con = DriverManager.getConnection(connectDB, user,
                    password);
        } catch (Exception e) {
        }
        return con;
    }

    public static int insertIntoData(String tabName, String tabTopName, String values) {
        int i = 0;
        try {
            if (conn == null) {
                conn = getSQLConnection();
                stmt = conn.createStatement();
            }
            if (conn == null || stmt == null) {
                return i;
            }
            String sql = "insert into " + tabName + tabTopName + " values " + values;
            i = stmt.executeUpdate(sql);
        } catch (SQLException e) {
            e.printStackTrace();
            Log.i("mtj", "Synchronize database [" + tabName + "] FAILD !");
        }
        return i;
    }
    /* Insert row into database and return primary ID of the operation */

    public static int insertIntoDataReturnId(String tabName, String tabTopName, String values, CallBackImp
callBackImp) {
        int id = 0;
        try {
            if (conn == null) {
                conn = getSQLConnection();
                stmt = conn.createStatement();
            }
            if (conn == null || stmt == null) {
                return id;
            }
            String sql = "insert into " + tabName + tabTopName + " values " + values;
            int i = stmt.executeUpdate(sql);
            if (i > 0) {
                ResultSet resultSet = stmt.executeQuery("select SCOPE_IDENTITY() as id;");
                while (resultSet.next()) {
                    id = resultSet.getInt(1);
                }
            } else {
                Log.i("mtj", "Synchronize [" + tabName + "] ---->FAILD !");
            }
        } catch (SQLException e) {
            e.printStackTrace();
            Log.i("mtj", " Synchronize table of the database [" + tabName + "] ---->FAILD !"");
        }
        return id;
    }
    /**
     * Close the connection with database
     */
    public static void closeConnect() {
        if (stmt != null) {
            try {
                stmt.close();
                stmt = null;
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
        if (conn != null) {
            try {
                conn.close();
                conn = null;
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
}

// Call the Google map service for navigation
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.GoogleMap.OnMapLongClickListener;
import com.google.android.gms.maps.LocationSource;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;

import android.location.Location;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;

function computeDistance(startCoords, destCoords) {
var startLatRads = degreesToRadians(startCoords.latitude);
var startLongRads = degreesToRadians(startCoords.longitude);
var destLatRads = degreesToRadians(destCoords.latitude);
var destLongRads = degreesToRadians(destCoords.longitude);


var Radius = 6371; // radius of the Earth in km
var distance = Math.acos(Math.sin(startLatRads) * Math.sin(destLatRads) +
Math.cos(startLatRads) * Math.cos(destLatRads) *
Math.cos(startLongRads - destLongRads)) * Radius;


return distance;
}


function degreesToRadians(degrees) {
radians = (degrees * Math.PI)/180;
return radians;
```

```
}
// END

var map = null;
var watchId = null;
var prevCoords = null;    //previous location for rout appearance.

var options =
{
enableHighAccuracy:true,
timeout:100,
maximumAge:0
}

function getMyLocation()
{
if (navigator.geolocation)
{
navigator.geolocation.getCurrentPosition(displayLocation, displayError, options);
var watchButton = document.getElementById("watch");
watchButton.onclick = watchLocation;    //runtime locating function
var clearButton = document.getElementById("clearWatch");
clearButton.onclick = clearWatch;    //cancel the runtime location funtion

}
else
{
alert ("oops, no geolocation support");
}
}


function watchLocation()
{
watchId = navigator.geolocation.watchPosition(displayLocation, displayError);
}
```

```
function clearWatch()
{
navigator.geolocation.clearWatch(watchId);
watchId = null;
}

function scrollMapToPosition(coords)
{
var latitude = coords.latitude;
var longitude = coords.longitude;
var latlong = new google.maps.LatLng(latitude, longitude);
map.panTo(latlong);
addMarker(map, latlong, "Your new location", "You moved to:" + latitude + "," + longitude);
}

function displayError(error)
{
var errorTypes =      //Four attribute for error message object
{
0:"Unknow error",
1:"Permission denied by user",
2:"Position is not available",
3:"Request timed out"
};
var errorMessage = errorTypes[error.code];
if (error.code == 0 || error.code == 2)
{
errorMessage = errorMessage + "" + error.message;
}
var div = document.getElementById("location");
div.innerHTML = errorMessage;

options.timeout += 100;
navigator.geolocation.getCurrentPosition(displayLocation, displayError, options);
div.innerHTML += ".....checking again with timeout=" + options.timeout;
}

function displayLocation(position)
{
var latitude = position.coords.latitude;
var longitude = position.coords.longitude;
var div = document.getElementById("location");
var distance = document.getElementById("distance");
var km = computeDistance(position.coords, ourCoords);     //Distance between start point and end point
div.innerHTML = "You are at Latitude" + latitude + ", longitude" + longitude;
distance.innerHTML = "You are " + km + "km from the wickedlySmart HQ"
div.innerHTML += "(with" + position.coords.accuracy + "meters accuracy)";
div.innerHTML += "(found in " + options.timeout + "milliseconds)";
if (map == null)
{
showMap(position.coords);
prevCoords = position.coords;
}
else
{
var meters = computeDistance(position.coords, prevCoords) * 1000;
if (meters > 20)
{
scrollMapToPosition(position.coords);
prevCoords = position.coords;
}

}
}

@Override
        public void activate(OnLocationChangedListener listener) {
            mListener = listener;
        }

        @Override
```

```
    public void deactivate() {
        mListener = null;
    }

    @Override
    public void onMapLongClick(LatLng point) {
        if (mListener != null && !mPaused) {
            Location location = new Location("LongPressLocationProvider");
            location.setLatitude(point.latitude);
            location.setLongitude(point.longitude);
            location.setAccuracy(100);
            mListener.onLocationChanged(location);
        }
    }

    public void onPause() {
        mPaused = true;
    }

    public void onResume() {
        mPaused = false;
    }
}

private LongPressLocationSource mLocationSource;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.basic_demo);

    mLocationSource = new LongPressLocationSource();

    SupportMapFragment mapFragment =
            (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.map);
    mapFragment.getMapAsync(this);
}

@Override
protected void onResume() {
    super.onResume();
    mLocationSource.onResume();
}

@Override
protected void onPause() {
    super.onPause();
    mLocationSource.onPause();
}

@Override
public void onMapReady(GoogleMap map) {
    map.setLocationSource(mLocationSource);
    map.setOnMapLongClickListener(mLocationSource);
    map.setMyLocationEnabled(true);
}
```

```
function addMarker (map, latlong, title, content)
{
var markerOptions =
{
position:latlong,
map:map,
title:title,
clickable:true
};
var marker = new google.maps.Marker(markerOptions);

var infoWindowOptions =
{
content:content,
position:latlong
};
var infoWindow = new google.maps.InfoWindow(infoWindowOptions);

google.maps.event.addListener(marker, "click", function ()    //google.maps.event.addListener
{
infoWindow.open(map);
}
};
}

function showMap(coords)
{
var googleLatAndLong = new google.maps.LatLng(coords.latitude, coords.longitude);
var mapOptions =
{
zoom:10,
center:googleLatAndLong,
mapTypeId:google.maps.MapTypeId.ROADMAP
};
var mapDiv = document.getElementById("map");
map = new google.maps.Map(mapDiv, mapOptions);

var title = "Your Location";
var content = "You are here" + coords.latitude + coords.longitude;
addMarker(map, googleLatAndLong, title, content);
}
```

//The following codes realizes the customized keyboard

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:orientation="vertical">

  <View
    android:layout_width="match_parent"
    android:layout_height="2px"
    android:background="@color/btn_gray"/>

  <RelativeLayout
    android:id="@+id/rl_back"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/iv_back_bg"
    android:padding="10dp">

  <ImageView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_centerInParent="true"
      android:src="@mipmap/keyboard_back"/>
  </RelativeLayout>

  <View
    android:layout_width="match_parent"
    android:layout_height="1px"
    android:background="@color/btn_gray"/>

  <android.support.v7.widget.RecyclerView
    android:id="@+id/recycler_view"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/keyboard_bg"
    android:overScrollMode="never"></android.support.v7.widget.RecyclerView>

</LinearLayout>


public class KeyboardView extends RelativeLayout {

  private RelativeLayout rlBack;
  private RecyclerView recyclerView;
  private List<String> datas;
  private KeyboardAdapter adapter;
  private Animation animationIn;
  private Animation animationOut;


  public KeyboardView(Context context) {
    this(context, null);
  }

  public KeyboardView(Context context, AttributeSet attrs) {
    this(context, attrs, 0);
  }

  public KeyboardView(Context context, AttributeSet attrs, int defStyleAttr) {
    super(context, attrs, defStyleAttr);
    init(context, attrs, defStyleAttr);
  }

  private void init(Context context, AttributeSet attrs, int defStyleAttr) {
    LayoutInflater.from(context).inflate(R.layout.layout_key_board, this);
    rlBack = findViewById(R.id.rl_back);
    rlBack.setOnClickListener(new OnClickListener() {
      @Override
      public void onClick(View view) { // Shut down the keyboard
        dismiss();
      }
    });
    recyclerView = findViewById(R.id.recycler_view);

    initData();
    initView();
    initAnimation();
  }

  // Filling the data
  private void initData() {
    datas = new ArrayList<>();
    for (int i = 0; i < 12; i++) {
      if (i < 9) {
        datas.add(String.valueOf(i + 1));
      } else if (i == 9) {
        datas.add(".");
      } else if (i == 10) {
        datas.add("0");
      } else {
        datas.add("");
      }
    }
  }

  // Initiate the adaptor
  private void initView() {
    recyclerView.setLayoutManager(new GridLayoutManager(getContext(), 3));
    adapter = new KeyboardAdapter(getContext(), datas);
    recyclerView.setAdapter(adapter);
  }

  // Initiate the screen scene
  private void initAnimation() {
    animationIn = AnimationUtils.loadAnimation(getContext(), R.anim.keyboard_in);
    animationOut = AnimationUtils.loadAnimation(getContext(), R.anim.keyboard_out);
  }

  // Popup the keyboard
  public void show() {
    startAnimation(animationIn);
    setVisibility(VISIBLE);
  }

  // Shut down the keyboard
  public void dismiss() {
    if (isVisible()) {
      startAnimation(animationOut);
      setVisibility(GONE);
    }
  }

  // Ensure the state of the soft keyboard
  public boolean isVisible() {
    if (getVisibility() == VISIBLE) {
      return true;
    }
    return false;
  }

  public void setOnKeyBoardClickListener(KeyboardAdapter.OnKeyboardClickListener listener) {
    adapter.setOnKeyboardClickListener(listener);
  }

  public List<String> getDatas() {
    return datas;
  }

  public RelativeLayout getRlBack() {
    return rlBack;
  }
}
```

```
//disable the default keyboard from been active
if (Build.VERSION.SDK_INT <= 10) {
  etInput.setInputType(InputType.TYPE_NULL);
} else {
   getWindow().setSoftInputMode(WindowManager.LayoutParams.SOFT_INPUT_STATE_ALWAYS_HIDD
EN);
  try {
    Class<EditText> cls = EditText.class;
    Method setShowSoftInputOnFocus = cls.getMethod("setShowSoftInputOnFocus", boolean.class);
    setShowSoftInputOnFocus.setAccessible(true);
    setShowSoftInputOnFocus.invoke(etInput, false);
  } catch (Exception e) {
    e.printStackTrace();
  }
}

//Define button events
@Override
public void onKeyClick(View view, RecyclerView.ViewHolder holder, int position) {
     etInput.setText(etInput.getText().toString().trim() + datas.get(position));
     etInput.setSelection(etInput.getText().length());
     break;
}

@Override
public void onDeleteClick(View view, RecyclerView.ViewHolder holder, int position) {
 // Delete button is down
 String num = etInput.getText().toString().trim();
 if (num.length() > 0) {
   etInput.setText(num.substring(0, num.length() - 1));
   etInput.setSelection(etInput.getText().length());
 }
}
```