

A Preliminary Design of Application for Medicine Taking Notification

Kasikrit Damklaing¹, Parit Maneechot¹ and Phannarai Anachotikul¹ +

¹ Information and Communication Technology Department, Faculty of Science, Prince of Songkla University

Abstract. This paper proposes an architectural and graphical user interface (GUI) design of application for medicine taking notification. The application will run on iOS smartphone. The application is a reminder system that helps parents or care takers remember to give children their medication. The application provides validated and high reliable information about medicine usage for children. There consist of symptom and treatment information of sicknesses. Therefore, the parents can and give medicines to their children in the safest, efficient and effective way.

Keywords: architectural; GUI; design; smartphone; alert; notification; medicine usage; children.

1. Introduction

Small children are learning all the time, they are studious and are surrounded technology which enables them to grow and develop physically and intelligently at a rapid pace. They are always eager to know and are curious. The nature of them might bring a chance to erupt incident and illness in an easy way. Sickness is an important factor that impedes children development. Because of sickness they lose the chance for smooth growth and development.

Parents or caretakers are the key people who take care children. Therefor the parents must have validated and reliable information for taking care of their children. Furthermore, the parents need to be aware if their children are required to take medication. Nowadays, smartphone are part of our daily life. Most of people take their phones with them everywhere they go and utilize it to do many tasks more than any other communication device.

In the past, there are many related health system. These are the large care system that integrated many components including a mobile device such as a mobile phone to interface with the server Nowadays, there are also many healthy or productivity of instances application for smartphone. These products are distributed for free and commercially via online places and market in both Apple and Android platforms [1][2]. The most of them provides particularly in term of general medicine usage. These applications do not provide mechanisms for alerting users' notification. In addition, the applications always give only knowledge to users that have no medicine usage information for small children.

This paper proposes an architectural and graphical user interface (GUI) design of Application for Medicine Taking Notification. The application is an alert medicine taking notification application which will run on Apple's iOS smartphone. The application will notify parents or caretakers to remember to give children their medicine. Furthermore, the application provides reliable information about medicine usage for children, which includes symptoms and treatment information.

+ Corresponding author. Tel.: +6674288694; fax: +6674288697.
E-mail address: kasikrit.d@psu.ac.th

2. Backgrounds

2.1. Medicine Usage for Small Children

The validated and high reliable information about medicine usage for children derives from a book entitled “Pharmacist Takes Care for Small Children”, published by The Association of Hospital Pharmacy (Thailand) [3]. There consists of many interesting subject followings: How to take care children when they have a high temperature, How to take care children when they have fever, When your children cough what should I do, Help me I have a colic, Antibody drug usage for children, Children might suffer from constipation how to solve problems, Let know about Fluoride, How is ORS important when children have diarrhoea, Induction of skin inflammation, Danger from toxics in house, and the last one is When feed some medicines to children. Each subject consists of symptom, treatment and medicine usage.

2.2. Cocoa Touch Framework

In our already published paper [4], we have proposed an architectural software and GUI design of an application for tablet computers. We have classified application development into four frameworks. There are Native, Web Based, Embedded WB and Cross Platforms. Each framework has both advantages and disadvantages depending on several factors such as device hardware utilization or fast network connection requirement, etc [5].

In this paper, we choose Native framework development. The advantages of this framework are latest application programming interface (API), fast software development and management, and device hardware utilization. Whereas, the developer has to engage with the framework specification requirements, for example, programming language must be Objective-C on only Xcode IDE. These conditions are disadvantages for platform specific development.

Cocoa is Apple's native object-oriented API for the Mac OS X operating system. The related API Cocoa Touch includes gesture recognition, animation, and a different user interface library. On the other hand, Cocoa Touch is for applications running the iOS operating system, used on Apple devices such as the iPhone, the iPod Touch, and the iPad Cocoa consists of the Foundation Kit, Application Kit, and Core Data frameworks and the Objective-C runtime itself [6]. Due to Cocoa development environment, the Xcode IDE consists of Xcode, Interface Builder and Simulator which provides Apple iOS SDK already bundled with iOS.

2.3. Model-View-Controller

Model–View–Controller (MVC) is an architectural pattern that splits interactions between users and applications into three roles: the Model classes represent raw data, such as documents, configurations, files, or objects in memory. The Views are visual representations of the data in the model meanwhile the Controller classes contain logic which links the models to their views, and maintains state to keep them synchronized to each other. This separation of concerns facilitates the independent development, testing, and maintenance of each role [5][8].



Fig. 1 The MVC and Interface Builder in iOS development [Ref6]

Cocoa's design is a strict application of MVC principles by providing framework support for all three MVC layers as shown in Fig.1. Apple's goal is to reduce the amount of boilerplate or "glue" code that developers have to write, freeing up resources or drain memory wasted to spend time on application-specific features [8].

3. Application Architecture and Design

3.1. Cocoa Touch Architecture

In feasibility study, we concern to choose hardware target for our application. Then, the application will be running on Apple's iOS smartphone, iPhone. The Cocoa Touch layer and the Core Services layer have their own Objective-C framework. In Fig. 2, Core Cocoa framework of iOS includes UIKit that provides the object and application displays in its user interfaces and defines the structure for application behaviour. The rest is Foundation that defines the basic behavior of objects, establishes mechanisms for their management, and operating-system services.

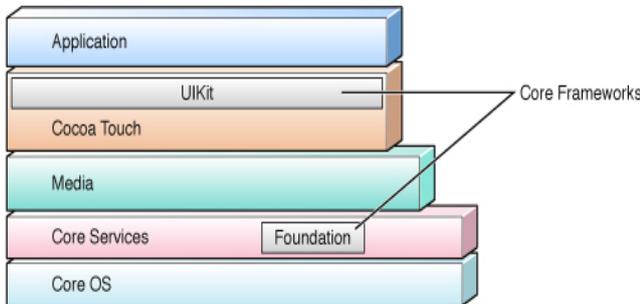


Fig. 2 The Cocoa in the architecture of iOS [6]

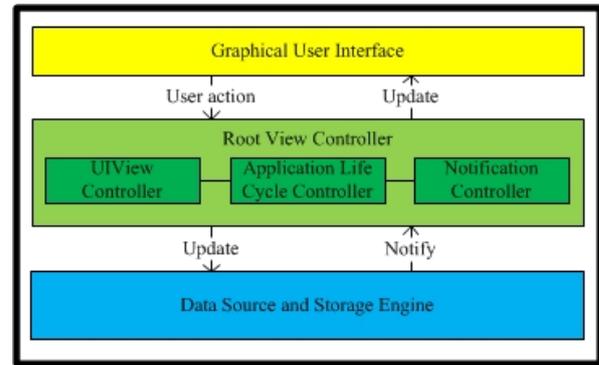


Fig. 3 The application architecture

3.2. Application Architecture

We design the application architecture based on MVC principles and divide it into three components, as shown in Fig. 3. The most important component is Root View Controller which is responsible for application controlling and orchestrating all components to drive for the application goal.

The Root View Controller orchestrates each UIView Controller which responses for each user interface object. In case that that user touches screen on an interface object, the object sends the action to the UIView Controller. The controller then sends the update back to user interface, for example, changing page etc. On the other way, if the user action is an influence against data, for example, the user configures the schedule for taking medicine, the Data Source and Storage Engine will then be in charge.

In case of notification, the Data Source and Storage Engine will notify the Notification Controller to send a message for notify the user the schedule. The management of the application life cycle is the most complex one. In the case that the application is not active, the Notification Controller still has to keep monitoring the schedule. Consequently, the Application Life Cycle Controller and the Notification Controller will be in charge in this scenario even though the user has ended the application from background process. It is essential to utilize *Notifications Center* at the iOS *Core Services* which is rather complex. We design the functional structure using the user interaction. The application activity diagram is shown in Fig.4.

3.3. States of Application Life Cycle

The parents can store and manage medicine schedule designated from the doctor for each child. They can store those medicines expiry date in their household stock. Then, the application will alert the parents of the expiry date. Therefore, the parents can give medicine to their children in the safest, efficient and effective way.

Our application is multi-view application. The application consists of many views or pages; each view contains many objects that co-operate each other to meet the specification requirements. Therefore, we determine the number of views and the interaction amongst them using state diagrams. Furthermore, the state diagram can reveal the big picture of application life cycle and conduct graphical interface design in the next phase. Fig. 5 shows main state diagram of the application. There are six views/pages, consisting of Main Page, Medicine Knowledge Page, Notification Page, Medicine Searching Page and About Us Page. The Main Page is the first page which is initialized into memory and firstly appear to a user. Each page can freely access each other.

Fig. 6 shows Knowledge Page state diagram. The application will navigate user through the flow of the pages. A user can select the symptom that frequently occurred to children. The information for each symptom includes treatment and medicine usage. Users sometimes might need to search and directly access to the medicine usage information. The application provides Searching Medicine Page as shown in Fig. 7

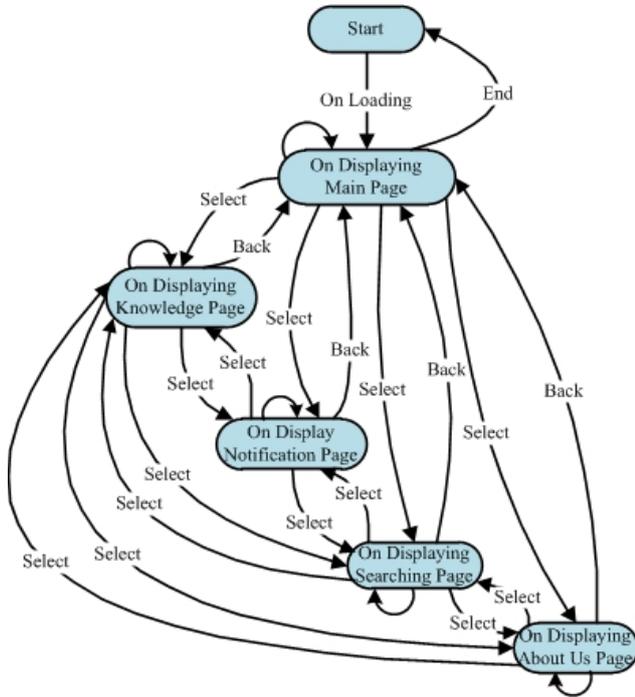


Fig. 5 The Main State Diagram of the application

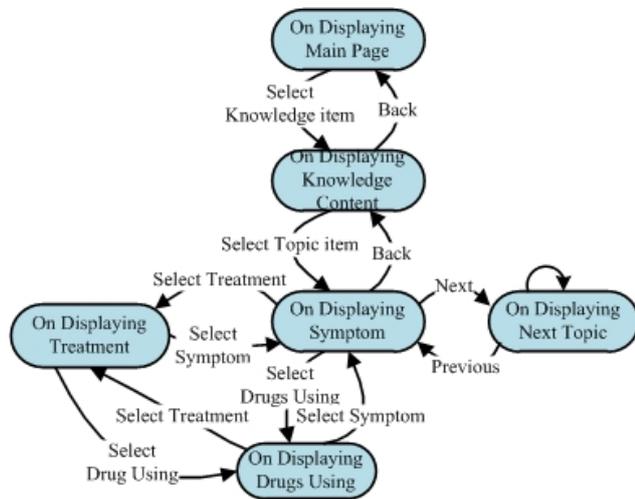


Fig. 6 The Knowledge Page State Diagram of the application

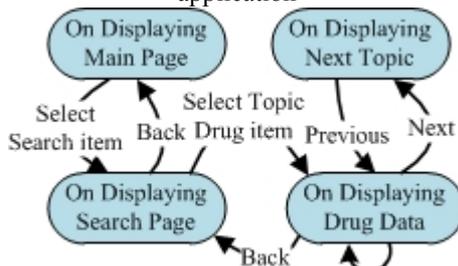


Fig. 7 The Medicine Searching Page State Diagram of the application

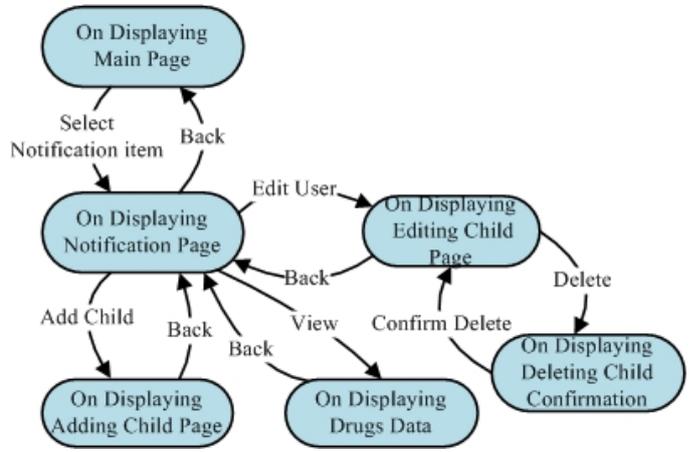


Fig.8 The Notification Page State Diagram of the application

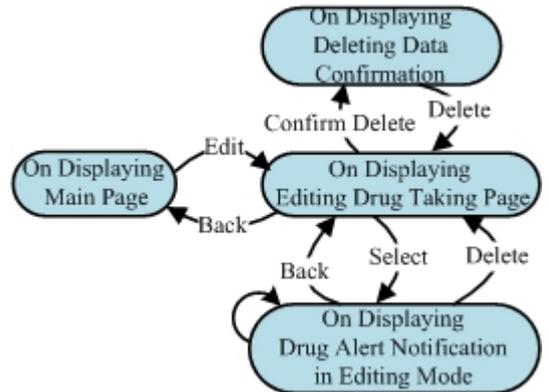


Fig.9 The Notification Editing Page State Diagram of the application

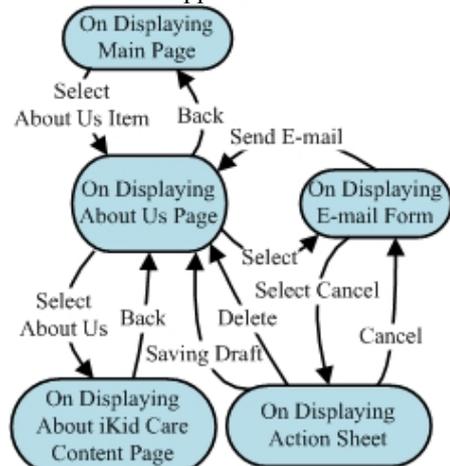


Fig.10 The About Us Page State Diagram of The Application



Fig. 11 The iKid Care Application's Main page



Fig.12 (a) The Knowledge Page and (b) the Notification Page

Fig. 8 shows Notification Page state diagram. In this page, users can view, store, update and delete the schedule. The application provides the store, update and delete the schedule process using editable page, the state diagram as shown in Fig. 9.

The last state diagram of the application is About Us Page as shows in Fig. 10. The main objective of this page is that the user can send feedbacks about the application to developer team via email. In this state, the application assumes that the user has internet connection. In addition, if the internet connection is unavailable due to any condition; the user saves it as a draft for later transmission.

3.4. Graphical User Interface

We design GUI in only Thai language. The application supports only portrait orientation displaying in according to the standard as many Cocoa Touch applications. The main application GUI derives from Tab View, Navigation Bar and Customize Table based on UIKit interface repository. The example of the application GUI is shown in Fig. 11 – Fig. 17. Fig. 11 shows main page which firstly appears after the On Loading state initialized and established the application into main memory.

The user can touch or 'tap' to choose each menu item appearing on the first main page. It consists of two buttons and six tab image items menu. The upper button presents the fundamental information for taking care of children. Another way, to get the information is to tap the second tab image from the left as shown in Fig 12 (a). The lower button is for the schedule management for each child prescribed by the doctor. Another way to access the schedule is that to tap the third tab image from the left as shown in Fig. 12 (b). The rest is the pages for medicine searching and information about the application page respectively.

Fig. 13 shows the Notification Editing Page in children list view. When a user taps \ominus symbol in Fig.13 (a), the application will then prompt the delete button for the user as shown in Fig. 13 (b). When a user taps $>$ symbol, the application will then prompt each child's medicines taking schedule in an editable mode as shown in Fig. 14 (a). We utilize three-column data picker enabling the user to configure the usage for each type of medicines as shown in Fig. 14 (b). In addition, the user can delete the information about the medicine usages using the Action Sheet of UIKit from the Notification Editing Pages in managing mode as shown in Fig. 14 (c).

Fig. 15 (a), (b) and (c) show GUI for the flow of medicine searching pages from the left hand side to the right hand side. A user can gesture up and down to list medicine name items as show in Fig.15 (a). On the other way, the user might tap on search text field for inputting medicine name as shown in Fig.15 (b). Pagej. Finally, the user taps the required item into the list. Then, the application will direct the user to the medicine content as shown in Fig.15(c).

The About Us Page is shown in Fig.16 (a). The application provides an email form for sending feedbacks and comments to the developer team as shown in Fig. 16 (b). In addition, the user saves it as a draft unless the internet connection is available by selecting command from pop-up Action Sheet as shown in Fig. 16 (c).

Fig. 17 (a) shows an example of the schedule for taking medicines consisting of a child name, list of medicines and usage respectively. Fig. 17 (b) shows an example of alert message on the Lock Screen using

Action Dialog from UIKit. When the user chooses OK, the dialog will close whereas the user choose View, the iOS will unlock screen and the application will direct the user to the Notification Page in view mode.

4. Memory Management

The iOS does not have a memory management. Therefore, it is necessary for developers to manipulate dynamically memory allocation and deallocation of many objects in the application by themselves. We concern and plan to implement the application against this obstacle. However, Cocoa Touch framework and the iOS provide customized Objective-C to deal in monitoring and utilizing all object references. Cocoa Touch manages dynamically allocated memory. Objects derived from the NSObject root class responds to a *retain* and a *release* message to keep a retain count which can be queried by sending a *retainCount* message [4][5].

An object is created with *alloc* or *copy* has a retain count of one. The object increments the retain count when it sends a *retain* message whereas it decrements the retain count with a *release* message. Therefore, when an object has a retain count of zero that means no longer any pointer references it, it will be destroyed or *dealloc* from memory.

5. Distribution Platform

The application will support iOS version 5.0 or later. Application distribution platform is a trademark apple smartphone, iPhone 4 or later. It has 3.5 inch (diagonal) widescreen Multi-Touch display and 960-by-640-pixel resolution at 326 ppi [9].

6. Conclusion

We propose an architectural and GUI design of application for medicine taking notification which will run on Apple’s iOS smartphone, iPhone 4 or later. The application is a reminder system that helps parents or care takers remember to give children their medication. We introduce Cocoa Touch framework and the application architecture based on MVC principles. Then, we concern the views of the application using the state diagrams. Finally, we present detailed delicateness GUI design. We will implement and test first version in the next phase.

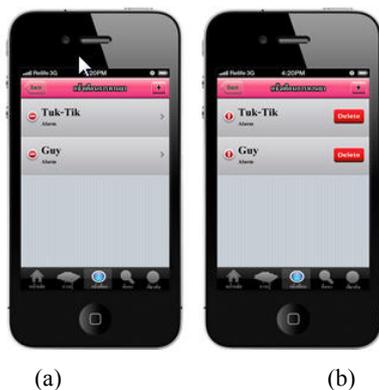


Fig. 13 The Notification Editing Page in children list view

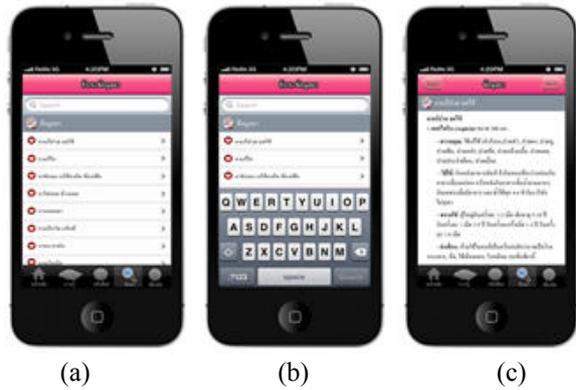


Fig. 15 The Searching Medicine Pages

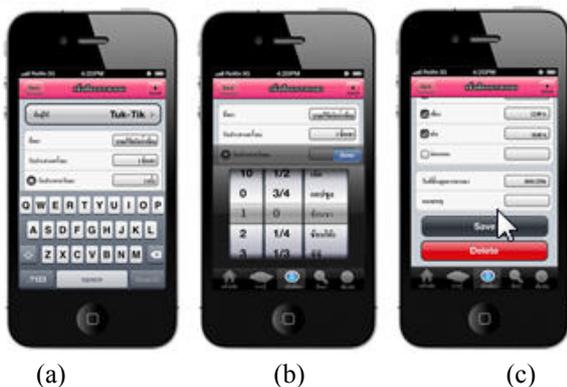


Fig. 14 The Notification Editing Pages in managing mode

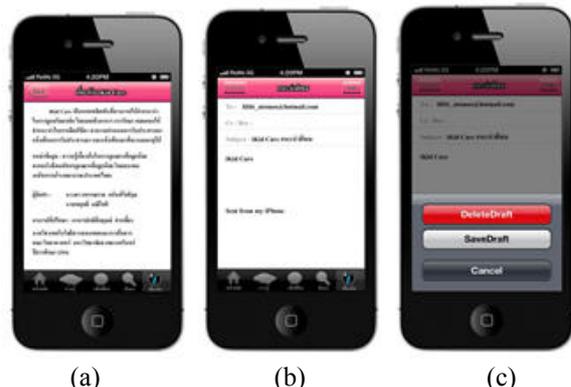


Fig. 16 The About Us Pages



(a) (b)
Fig. 17 The Notification Page

7. Acknowledgement

This research work is a senior student project at Information and Communication Technology Programme, Faculty of Science. This work is supported by Faculty of Science, Prince of Songkla University. The authors are also thankful for The Association of Hospital Pharmacy (Thailand).

8. References

- [1] Talaei-Khoei, A., Ray, P., Parameswaran, N. An Awareness Framework for Agent-Based Mobile Health Monitoring. *Proc of the Next Generation Mobile Applications, Services and Technologies, 2009. NGMAST '09. Third International Conference on*, vol., no., pp.108-113, 15-18 Sept. 2009, doi: 10.1109/NGMAST.2009.39.
- [2] Nkosi, M.T., Mekuria, F. Cloud Computing for Enhanced Mobile Health Applications. *Proc of the Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, vol., no., pp.629-633, Nov. 30 2010-Dec. 3 2010, doi: 10.1109/CloudCom.2010.31.
- [3] The Association of Hospital Pharmacy (Thailand). *The Pharmacist Takes Care Medicine Usage For Children*. Bangkok: Hyodmuek Publishing Ltd. First published 2003. ISBN 9748222861.
- [4] Damkliang, K., Chanlert, S., Thongnuan, A. Traditional Thai Musical Instrument For Tablet Computer– Ranaad EK. *Proc. of the: International Conference on Computer, Communication and Information Sciences, and Engineering (ICCCISE 2012)*, Paris: World Academy of Science, Engineering and Technology 64 2012, pp. 861-866.
- [5] Steven F.Daniel. *Xcode 4 iOS Development Beginner's Guide*. Birmingham: Packt Publishing Ltd. First published 2003. ISBN 978-1-849691-30-7.
- [6] What Is Cocoa. Available online: May 2012. https://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/CocoaFundamentals/CocoaDesignPatterns/CocoaDesignPatterns.html#//apple_ref/doc/uid/TP40002974-CH6-SW6.
- [7] Cocoa (API). Available online: May 2012. [http://en.wikipedia.org/wiki/Cocoa_\(API\)](http://en.wikipedia.org/wiki/Cocoa_(API))
- [8] The Model-View-Controller Design Pattern. Available online: May 2012. https://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/CocoaFundamentals/CocoaDesignPatterns/CocoaDesignPatterns.html#//apple_ref/doc/uid/TP40002974-CH6-SW6.
- [9] iPhone 4 - Technical Specifications. Available online: May 2012. <http://support.apple.com/kb/SP587>.