

# My Child : Persuasive Mobile Based Approach to Enhance Children's Health

D. Wellappili

Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka  
dushaniwellappili@gmail.com

**Abstract**—The purpose of this project was to develop a native Android application called My Child, for the user to manage the information of the child's health and development and monitor the growth of the child since birth. The parent could keep on updating the records and track the progress of their child's growth through medication, milestones and vaccination logs. The application was also developed with report generation and vaccination due to reminder functionalities. The approved immunization schedule in Sri Lanka was selected to be the default vaccine schedule of the application because the main target audience of the system was Sri Lankan parents. Since there is no such facility at the moment for Sri Lankan parents to track their child's well-being, this application was expected to fill that gap. The application was developed using Android SDK and various other open libraries. Android API 19 and above versions are supported by the application. This is a major tool for every parent who wants to track continuously updated facts about their child's health and well-being.

**Keywords**—Android; Child Health Development; Vaccination log; Growth log; Milestone log

## I. INTRODUCTION

Current experience suggests that the growth monitoring programs are not always fulfilling the prerequisites and consequently are not achieving the desired impact on child's health. Both slow growth and fast growth may indicate a nutritional or other health problem. Therefore, growth monitoring is an essential part of primary health care in children. But there's more in tracking your child's development than logging height and weight. There are a number of other childhood milestones to keep watch for. Not reaching these milestones, or reaching them much later than other children, could be a sign of a developmental delay. Once become a parent, the amount of all that information you need to handle regarding your child's care can be too overwhelming to manage. Every Sri Lankan parent gets a manual health development report card to keep records of health and growth of their child but that may even get misplaced. Parents may tend to forget the due dates of their child's vaccinations with their busy schedules. If there are multiple children the situation can become even worse.

My Child is a personal health recording Android app to keep track of child's drugs, medications, vaccinations, weight, height and all other feel-good basics. The child's size measurements can be plotted on the growth charts of the system. Those can be used as a comparison to other

children of the same age and to document a pattern of growth for an individual child. These can be often referred to when there are other indications of a possible health problem. The system helps the parents to be well organized about the vaccinations so they won't miss an appointment or forget a due date for a vaccination. It also keeps track of every precious moment of the child through a milestone logger.

The system can track more than one child at a time, send notification reminders and even export data as a report in to a pdf document. Since a mobile phone is something which everyone has with them every 24 hours and Android is a popular mobile operating system used in Sri Lanka, the system was developed as an Android application. It was developed as a rich client application, so that the business and data service layers are located on the device itself.

## II. LITERATURE REVIEW

The objective of developing this application is to include all the essential functionalities needed for a parent to monitor their child's growth and health in an effective single mobile app. When starting the mobile development, few questions may be raised on what differentiates mobile application development and traditional application development, what options and design principles are available and on which platform or rather operating system the project has to be implemented. It was also important to review about the existing systems. These questions were answered as each literature was reviewed.

There are similar mobile applications such babycenter, babydiary etc., already available in both iOS APP Store and Google Play Store [1]. But most of those do not provide the facility to keep track of vaccines of the child. Such apps consist of third party advertisements as well. Many of the vaccines schedules are different from the vaccines recommended and issued by the family health bureau of the Ministry of Health, Sri Lanka. This is due to differences in the epidemiology (patterns and frequency) of the disease in each country, differences in the way that countries make decisions about which vaccines to be offered to everyone. Some apps do not provide the facility to upload different vaccine schedules apart from the default one.

As an introduction going into the development issues, paper on Software Engineering Issues for Mobile Application Development pointed out that various mobile

devices with various screen sizes and different hardware would add to the complexity of testing [2]. The best practice suggested is to follow an agile methodology that can quickly adapt to changing user requirements and to follow development guidelines published by various platforms. Further, the native application languages are generally known to be more complicated than web application languages [3]. In terms of user experience, web applications must be connected to the Internet during the entire time the application is running, but native applications can work offline as well as online. The conclusion was to favor data over decoration in the application. The advantage of a native application is that performance issues are not related to the size of payload and the code is already compiled.

When comparing the two of the leading platforms in the world Android and iOS, it was seen that both have advantages and choice between them that would depend on the target market of the app [4]. Android was the choice for My Child since it is an open source platform which supports multifunction, provides rich tools to make interactive applications and downloading the software required for making the application is absolutely free. A suggestion made by the book on Mobile Interaction Design for capturing user needs was to apply disciplines such as psychology and sociology and to look at past successful designs [5]. Limiting the number of items in the menu to a maximum of seven was one of the best practices of UI designing. The studies presented in the book indicated that people can generally remember plus minus seven items in a menu. Use of icons (symbols) was suggested to replace text whenever possible to free-up space. Generally, pictures are easier to remember than text. Other design guidelines included quick navigation to frequent functions and limiting excessive scrolling or page-to-page navigation. Excessive scrolling would become tiring on a small screen and users could get lost in page-to-page navigation.

### III. SYSTEM MODELS

This section describes the system requirement specification of the system. 'Add Parent', 'Maintain Child Profile', 'Maintain Growth', 'Maintain VaccineLogs', 'Maintain Milestones', 'Maintain Medications', 'Notify Vaccination', 'Generate Reports' were identified as the main functional requirements. The ultimate goal of the functionalities is to help the parent to keep easy track of their child's health and development.

A parent can have multiple child accounts with child's name, gender, birth date, birth measurements, blood type etc and edit them later as well. The user can enter daily entries on child's growth on weight, height and head size. The growth of the child is shown in plotted charts according to the growth entries that the user has entered before. The application provides the functionality to add details about drugs given to the child with drug name, route, dosage etc. Whenever the drug is given to the child, a medication log can be added for that with date and time. Routine vaccinations are already added in the application. The user has the ability to add any other vaccines apart from the default schedule. Whenever the vaccines are due to be given, the user gets a

notification. After the child had the vaccination, the parent can enter a vaccination log for that as well. The parent or the user can add milestones of their child such as rolling over, sitting up, holds head, speak, walk alone etc. The user can generate a report on child's details by giving a period of time and can even ask for that report to be sent to a given email address. That will be saved in phone's storage as well.

For non-functional requirements of the application, UI elements with compatible look and feel requirements were used in order to ensure that the user shall be able to understand what they mean and tend to use them without refusing due to other reasons. The default health related terms used were entered with a note of what they mean and why they are needed. For reliability, it was made sure that the functionalities are always available. The main functionalities of the application are shown in Figure 1.

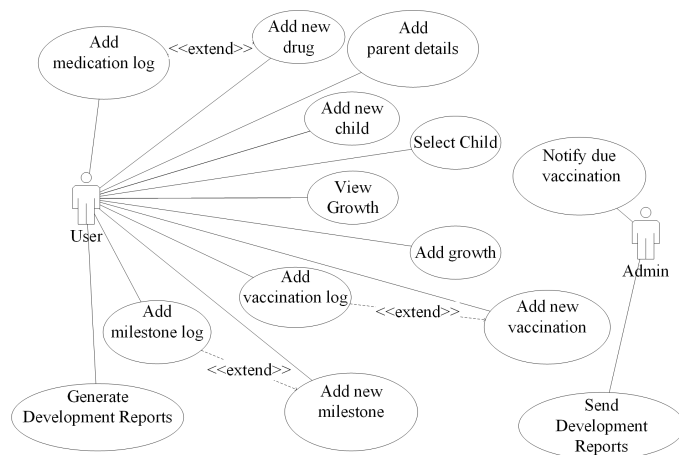


Fig. 1. Use case diagram of the system

### IV. SYSTEM DESIGN

The Model View Controller pattern architecture was the main architecture used for the system. The main core operation of the system is to retrieve data from a data storage and update the user interface with the newly requested information based on the user's inputs. The logical sense is to tie the user interface components with the data store components. However, since the user interface components are updated regularly to accommodate the changing user requirements and new technologies more than the data store components, extra coupling is introduced in the system. The aim of this pattern was to separate the components of the user interface (View); core functionality and data (Model) and the response to user inputs (Controller). The UI components resemble the View component of the pattern. The service and data components represent the Model component of the pattern which monitors the behavior of data and core application domain. It also handles requests from the View and Controller components. The widget components are similar to the Controller where it waits for user inputs (such as a button click) then interprets those inputs and notifies the Model or View of the change.

The domain class diagram is shown in Figure 2. The underlined parameter is the primary key of each entity.

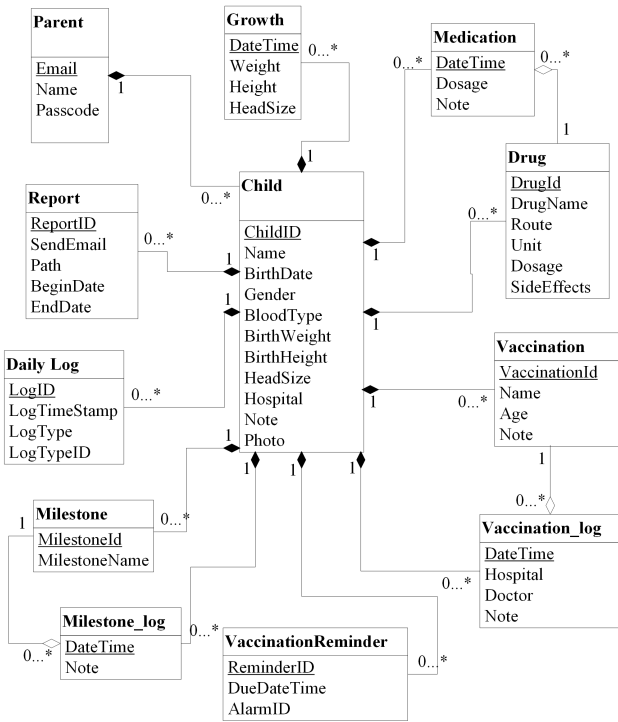


Fig. 2. Domain class diagram of the system

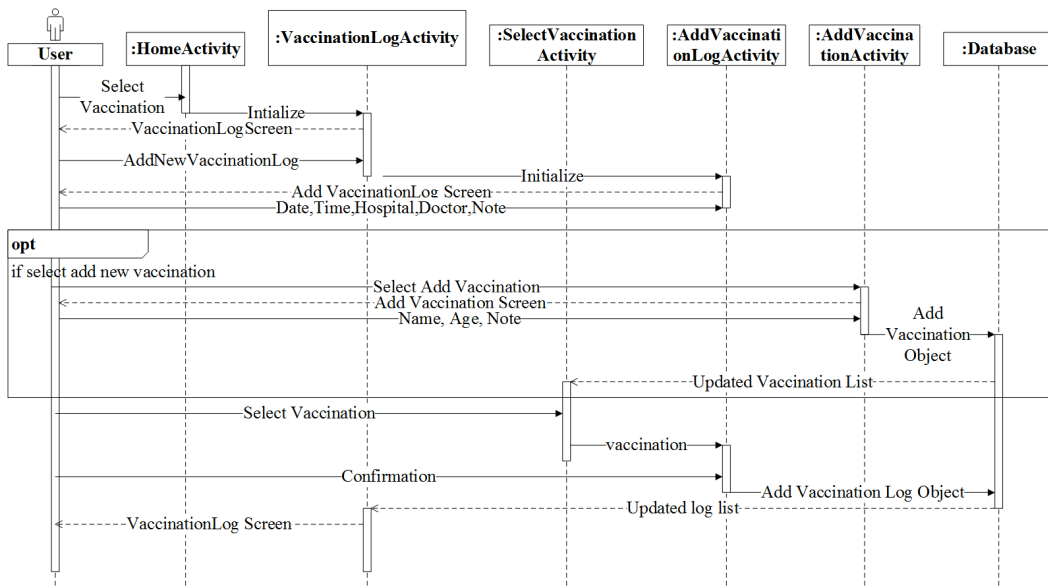


Fig. 3. Sequence diagram of Adding a Vaccination Log

## V. SYSTEM IMPLEMENTATION

### A. Implementation Procedure

The system was implemented using Android SDK which includes development tools, an emulator and the required libraries to build the application [6]. Application was written using the Java programming language and run on Genymotion, a custom virtual machine designed for embedded use which runs on top of a Linux kernel [7]. It enables application's compatibility with android platform by means of API. SQLite open source database is used to store

The main sequence diagram of adding vaccination log is shown in Figure 3. When adding a vaccination log first the user has to select the vaccination menu icon from the HomeActivity then through an intent home activity passes the child id and start the VaccinationLog activity and the VaccinationLog activity initializes the SelectVaccination activity. Then the user has to enter vaccination log details while selecting a vaccination. If the user needs to add a new vaccination apart from the existing list, AddVaccination activity is initialized. So the user can add a new vaccination. Those details will be sending over to the database through a vaccination model object via VaccinationController. When user confirms adding the vaccination log, VaccinationLog objects is created and sends over to the database through VaccinationLog Controller. Finally user can see the updated log list of vaccinations with the one he has already added.

The database was designed in a way that data redundancy was prevented and data integrity, isolation and security aspects were preserved. The domain class diagram in Figure 2 explains the database design used for the system.

data in the application [8]. For plotting graphs for growth charts of the child achartengine:1.2 is used [9]. For generating reports in PDF format, itextpdf library is used [10]. Itext is a popular library for creating and manipulating PDF documents. Since the application heavily depends on date and time functions, joda-time2.9.3 library is used to make date and time calculations easy [11]. JavaMailAPI was used to send reports via email [12].

The database was implemented in a thread safe manner to prevent memory data leakages. The update and delete queries of the SQLite database were written to run as

transactions. So in case of a failure, it will make sure that the entire transaction will rollback and data would get preserved. The default vaccination schedule in the system was according to the National Immunization Schedule for EPI Vaccines – Sri Lanka, approved at the National Advisory Committee on Communicable Diseases on 03rd June 2011 [13]. These vaccinations were added under each age category with an additional note. Ex: Age - 0-4 weeks, Vaccination Name – BCG, Note - Before leaving hospital, preferably within 24 hours of birth. If a scar is not present 2nd dose could be offered after 6months, up to 5 years. Parent can review the plotted growth charts and assess their child’s development with the percentile data. These percentile data added to the system refers to the child growth standards published by World Health Organization [14]. Every child experiences general milestones when they grow up. These milestones are already added to the database prior to adding a child [15]. Few of these milestones are walking alone, standing alone, walking with assistance, crawling, speaking etc.

**B. Algorithm**

The most important algorithm of the source code of the project relates to scheduling, previewing and canceling vaccination notifications. The pseudo code of the algorithm is shown in Figure 4.

Whenever a new vaccination is added, the due date for that particular vaccination is calculated. Then a vaccine reminder is added to the database and an alarm with a unique alarm id is set using an alarm manager. If the due date and time is behind the beginning of current date there is no need of generating notifications. So once the due date and time arrives, the service is invoked to show a notification to the user. Whenever user edits vaccination’s age attribute or deletes the vaccination, the previously set alarms for notifications is canceled off.

```

vaccination ← current vaccination
child ← current child
onVaccineAddSuccess() {
    addVaccinationReminder(vaccination) }
addVaccinationReminder(vaccination) {
    dueDate ← (child → birth date)+
              (vaccination → age)
    dueDate ← change time of dueDate to
              00:00
    if (dueDate <= currentDate){
        alarm id ← generate unique alarm id
                    using system time
        vaccinationReminder ← vaccination

        reminder with dueDate,
                    (child → id), (vaccination →
                    id), alarm id
        pendingIntent(alarm id)
        alarmManager →
        setAlarm(pendingIntent, dueDate) } }
cancelAlarm(vaccineReminder) {
    alarmManager → cancelAlarm(vaccineReminder
    → alarmId) }

```

Fig. 4. Pseudo code for the vaccination notification

**C. Main Interfaces**

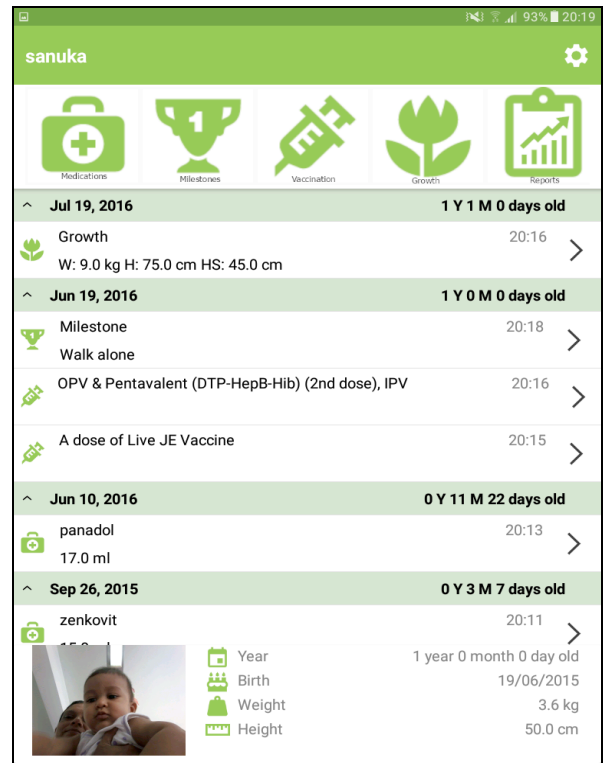


Fig. 5. Home screen of the child

When the application is launched, the user is directed to the home page of the first child added as shown in Figure 5. If not, add child view is prompted. The home screen has an overview of the child with age, birth height and weight along with a photo. Above that there is a summary of the child’s past activities like growth logs, medication logs and vaccination logs with the date and age. Figure 6 shows add child activity screen where the parent can add their child to the application. The user can upload new or an existing photo of the child from the phone with the name, birth date, gender, blood type, height, weight, head size, hospital and some other notes.

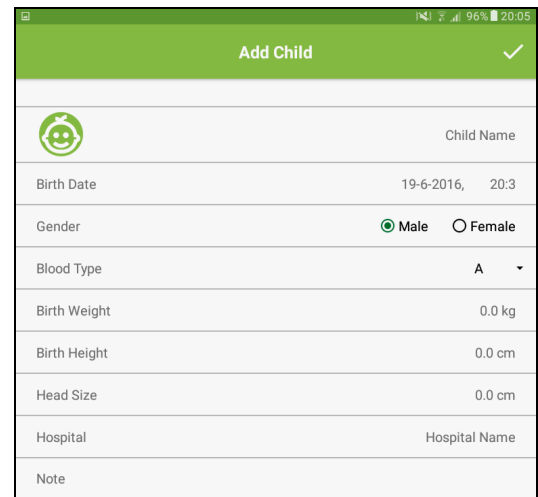


Fig. 6. Add Child Screen

Figure 7 shows the interface of how the growth details are shown for the user. The weight is plotted in a graph based on the added growth details. The percentiles weight curves recommended by World Health Organization are plotted by default. So the parent can assess their child's growth based on that.

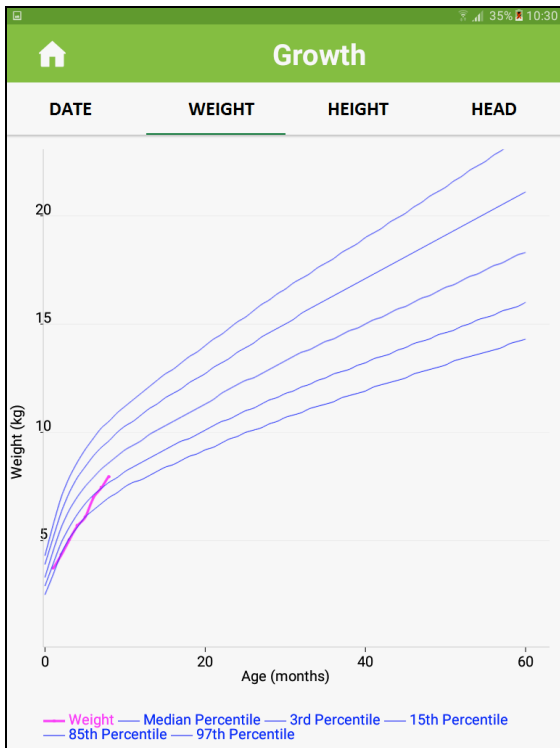


Fig. 7. Weight(kg) vs. Age(months) Growth Graph of the child

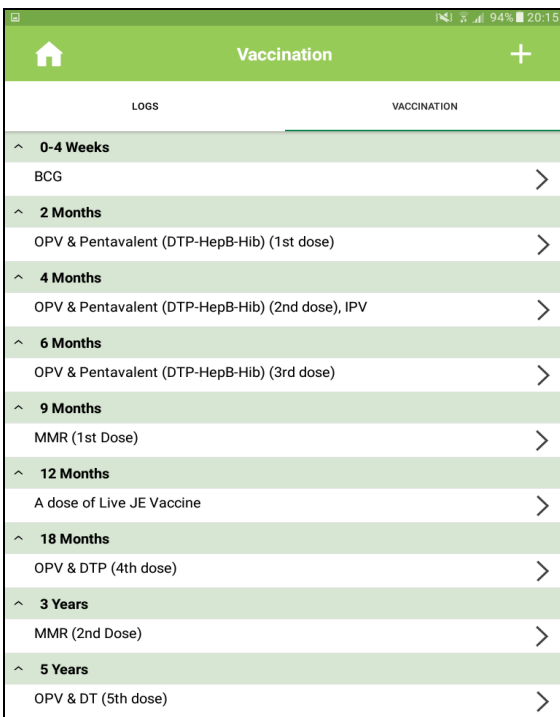


Fig. 8. Vaccination Screen

The vaccination interface is shown in Figure 8. The database is populated with the vaccination details of default vaccinations needed. The user can add any additional vaccination as well. Then when the child gets a vaccine, a vaccination log can be added. Each time a vaccination is entered, a reminder is added and when the due date arrives a notification is shown to the user as shown in Figure 9.

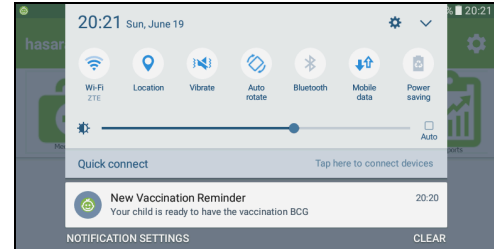


Fig. 9. Vaccination Reminder Notification in notification bar

## VI. SYSTEM TESTING AND ANALYSIS

The test scope included testing of all functional, application performance and use case requirements listed in the software requirements document. By undertaking the test effort defined, it was assured that the system met the full requirements, including quality requirements and fit metrics for each quality requirement. Database and data integrity testing was carried out to exercise database access methods and processes independent of the UI in order to observe and log incorrect functioning target behavior or data corruption. Such technique carried out was invoking each database access method and process, seeding each with valid and invalid data or requests for data.

The user interfaces were tested using the Espresso testing framework which supported automated testing of all the major views of the application [16]. It was tested whether the app behaves as expected in window navigations, the behavior of the app when using various key presses and touch gestures in an input form and menu sizes, position, states and focus. The performance and behavior of the application were assessed under certain conditions such as low battery, bad network coverage, low available memory by several users and other conditions. Behavior and performance of the application were observed for designated use cases under varying workload conditions.

It was ensured that only those users granted access to the system are capable of accessing the application and the user has access only to the set of functions he has been given with. For application-level security, the user type (there is only one user in our case) and the functions or data that each user type has permissions for were identified (Adding a child, maintaining records etc.) and restricted the user to delete default entries. It was verified that only the registered users are enabled to sign in for the application in system level security. Running unit tests using JUnit4 after every build helped to catch and fix software regressions introduced by code changes to the app. Functional testing was carried out by viewing and executing use cases using both valid and invalid data. It was tested whether correct age

is displayed for the given birth date and whether it is restricted for the user to enter future dates.

Analysis of the defects indicates that the majority of the defects found tend to be minor problems classified as low or medium in severity. The other significant finding was that software components comprising the interface to notification and report sending contained the highest number of defects. The area of tests with the highest failure rate was the Notification and Report Sending functionality, which failed under high volume testing. Approximately, 93% of the code was executed during the testing. This coverage exceeded the target of 90%.

## VII. CONCLUSION AND FUTURE WORK

MyChild app was successfully implemented with most of the proposed features. The application can be used by anyone who wants be updated and monitor their child's development on a daily basis. The vaccination and vaccination log functionalities of the system play significant importance. Because this functionality can be rarely seen in similar child health apps. The notification system makes sure that the user parents will not miss their child's vaccines and give the best possible protection for their child. The growth monitoring functionalities implemented gives the opportunity for parents to assess their child's development personally and digitally. Through this many nutritional and other health related problems regarding the child's development can also be identified. MyChild assures that the parent will not miss their child's most important milestones. The report generation functionality also plays a key role in the MyChild functionalities. It gives the flexibility to the user to get reports within any time period he wants. The reports are generated in PDF format, so that can be easily read or shared among any other parties.

To improve the application, following enhancements can be implemented in the future. 1) Additional feature to give growth assessments automatically by analyzing the growth records and WHO percentile data. 2) Expand the application scope and give the chance for midwives to connect with the app and get notifications about the vaccinations and their due dates of the children they are examining. 3) Add more functionality to backup the database of the child to a cloud service. 4) Create a forum for parents who are using the app

to connect and discuss their interests and problems and find solutions among themselves.

## REFERENCES

- [1] "baby care and child development apps - Google Play," 2016. [Online]. Available: <https://play.google.com/store/search?q=baby%20care%20and%20child%20development&c=apps&hl=en>. [Accessed 09 April 2016].
- [2] A. I. Wasserman, "Software engineering issues for mobile application development," in Proceedings of the FSE/SDP workshop on Future of software engineering research, Santa Fe, New Mexico, USA, 2010.
- [3] A. Charland and B. Leroux, "Mobile application development: web vs. native," Communications of the ACM, vol. 54, no. 5, pp. 49-53, May 2011.
- [4] M. H. Goadrich and M. P. Rogers, "Smart smartphone development: iOS versus android," in Proceedings of the 42nd ACM technical symposium on Computer science education, Dallas, TX, USA, 2011.
- [5] M. Jones and G. Marsden, Mobile Interaction Design, John Wiley & Sons, Ltd., 2005.
- [6] "Download Android Studio and SDK Tools | Android Studio," 2016. [Online]. Available: <https://developer.android.com/studio>. [Accessed 3 April 2016].
- [7] "Genymotion - Fast & Easy Android Emulator," 2016. [Online]. Available: <https://www.genymotion.com/>. [Accessed 03 April 2016].
- [8] "SQLite Home Page," 2016. [Online]. Available: <https://sqlite.org>. [Accessed 2 April 2016].
- [9] "achartengine.org - News," 2016. [Online]. Available: <http://www.achartengine.org>. [Accessed 10 May 2016].
- [10] "iText," 2016. [Online]. Available: <http://itextpdf.com>. [Accessed 5 May 2016].
- [11] "Joda-Time - Home," 2016. [Online]. Available: <http://www.joda.org/joda-time>. [Accessed 5 May 2016].
- [12] "JavaMail API Reference Implementation: Wiki: Home - Project Kenai," 2016. [Online]. Available: <https://java.net/projects/javamail>. [Accessed 5 May 2016].
- [13] M. o. H. S. L. Epidemiology Unit, "Immunization Schedule," 2011. [Online]. Available: [http://epid.gov.lk/web/images/pdf/Immunization/immunization\\_schedule-english\\_2015-06-26.pdf](http://epid.gov.lk/web/images/pdf/Immunization/immunization_schedule-english_2015-06-26.pdf). [Accessed 10 May 2016].
- [14] W. H. Organization, "WHO | The WHO Child Growth Standards," 2016. [Online]. Available: <http://www.who.int/childgrowth/standards/en>. [Accessed 10 May 2016].
- [15] "Espresso," 2016. [Online]. Available: <https://google.github.io/android-testing-support-library/docs/espresso>. [Accessed 10 June 2016].
- [16] C. f. D. C. a. Prevention, "Developmental Milestones," [Online]. Available: <http://www.cdc.gov/ncbddd/actearly/milestones/>. [Accessed 10 May 2016].