THINK INDIA JOURNAL

ISSN: 0971-1260
Vol-22-Issue-16-August-2019

A Milky Android Based Mobile Application

Maanvendra Prajapati
Lovely Professional University
Phagwara, Punjab, India.
manvendrag.143@gmail.com

Ruchika
Lovely Professional University
Phagwara, Punjab, India.
ruchika.24905@lpu.co.in

Bhupinder Kaur
Lovely Professional University
Phagwara, Punjab, India.
bhupinder.23611@lpu.co.in

***Abstract*-** TheMilky is an android based mobile application that helps university students in getting fresh milk and provide hostel door delivery service. The main objective of this application is to provide door services to working people and for those individuals staying outside home. Based on feasibility user can choose plan and proceed for payment and already fixed. All registered users' complete details and transections regarding payments are stored in firebase data base. Our purposed system provides a technical solution for the above scenario. Every user is uniquely identified by his or her identification number. Unique ID number and password will be credentials for Login after successful Signup.

Keywords – Android, SDK, API, APK, Application

I. Introduction

Android is an open source working framework dependent on Linux. It was for the most part intended for versatile touchscreen gadgets as cell phones and tablets. Android can likewise be found in a wide scope of gadgets and equipment, for example, TVs, smartwatches and vehicles. Android enables clients to make various applications by exploiting every one of the gadgets qualities.

Android is created by a consortium of designers known as the Open Handset Alliance, with the fundamental patron and business advertiser being Google.

II. SDK

Android is the most popular mobile operating system in the world. Consequently, Google has created instruments for the advancement of new applications to make it as simple as could reasonably be expected. Inside these instruments we can discover Android Software Development Kit (SDK). With Android SDK any software engineer can code, test and troubleshoot Android applications.

Android SDK versions and API levels:

Android SDK versions are identified by a unique number known as the API level.

- Android 1.0 Apple Pie – API Level 1 (September 2008)
  First android commercial version.
- Android 1.1 Banana Bread – API Level 2 (February 2009)
  Added some new functionality and some previous version bugs fixed.
- Android 1.5 Cupcake – API Level 3 (April 2009)
  Added support for widgets (very useful now), keyboard in screen, predictive text, transitional animations, auto-rotation and other features.
- Android 1.6 Donut – API Level 4 (September 2009)
  This version adds the ability to add personal applications results in search results, first text to speech implementation with Synthesis of speech, support for gestures.
- Android 2.0 – 2.1 Éclair – API Level 5-7 (October 2009, December 2009, January 2010)

ISSN: 0971-1260
Vol-22-Issue-16-August-2019

This version support multiple accounts, adds camera, multi-touch events, animated wallpapers, Bluetooth 2.1, new GUI, hardware optimization and more supports for different screen sizes.

- Android 2.2 Froyo –API Level 8 (May 2010)
  Ability to install apps in external memory, push notifications, wifi-hotspot, adobe flash support, support for numeric and alphanumeric passwords.
- Android 2.3 Gingerbread – API Level 9-10 (December 2010)
  It supports large screens dimensions and resolution, implementation of "cut, copy, paste" tools, NFC support, multiple camera and video chat through google talk.
- Android 3.0 Honeycomb – API Level 11-13 (February 2011)
  First version to support multi core processors, an action bar. It was launched and optimized for tablets.
- Android 4.0 Ice Cream Sandwich – API Level 14-15 (October 2011)
  It included software buttons on screen, screen-shot integrated functions, screen lock application shortcuts, facial unlock, Android Beam, Wifi-Direct support, customized launcher
- Android 4.1 Jelly Bean – API Level 16-18 (July 2012)
  This version brought Expandable notifications, multichannel audio, One-finger gestures, Day dream screensaver, new download notification and some bug fixes.
- Android 4.4 KitKat – API Level 19-20 (October 2013)
  Version was initially named "Key Lime Pie", came with immersive mode (keep navigation and status bar hidden). It was initial release of Android Wear Platform for smartwatches.
- Android 5.0 Lollipop – API Level 21-22 (November 2014)
  This version features a redesign user interface built around a responsive design called "Material Design", in this version Dalvik was replaced by ART, Vector drawable. Lock screen provide shortcuts to application and notification settings and device protection.
- Android 6.0 Marshmallow – API Level 23 (August 2015)
  Features like Contextual search from keywords within apps, App Standby feature, Native finger print support, USB Type-C support, adoptable External Storage to behave like internal stor
  age, Experimental multi-windows feature.
- Android 7.0 Nougat – API 24 (August 2016)
  This is the first Android update to bring support for VR, screen zoom, quick switch of apps, add a "Clear All" button, Multi window support, Night light, new emoji's and other improvement.
- Android 8.0 Oreo – API Level 26 (August 2017)
  It introduces faster, smooth loading times and faster execution, new safety features such as blocking unknown apps by source. Adding snoozing and Notification Channels, to customize notification you want to see.

ISSN: 0971-1260
Vol-22-Issue-16-August-2019

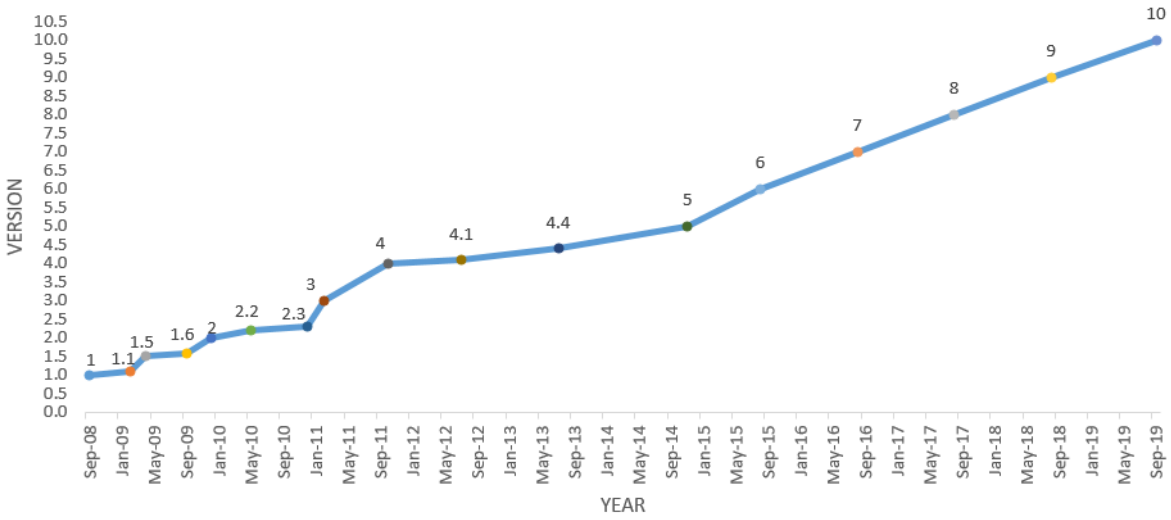


Figure1. Versions of Androids

III. What is an App?

App is an abbreviation for application. Applications are software programs that are executed in a PC, cell phone or other gadget. The most widely recognized use for the term application is to name software programs that sudden spike in demand for cell phones or cell phones.

Applications structure some portion of Android's design. In Android, Apps contain compiled Java code, information documents and assets records utilized by each application. The entirety of this comes stuffed up in a document named Android Application Package (APK), with extension .apk. APKs are a variation of Java's JAR design.

Figure 2. APK

Applications are bundled by utilizing the tool AAPT (Android Asset Packaging Tool) which is included in Android SDK. To package an application, Java code is compiled, at that point, incorporated libraries are likewise included and all the code is optimized so it very well may be executed in Android RunTime. Additionally, the packages incorporates records of information and assets, for example, pictures, media documents, xml, and so forth. A significant file that is additionally included is the AndroidManifest.xml

ISSN: 0971-1260
Vol-22-Issue-16-August-2019

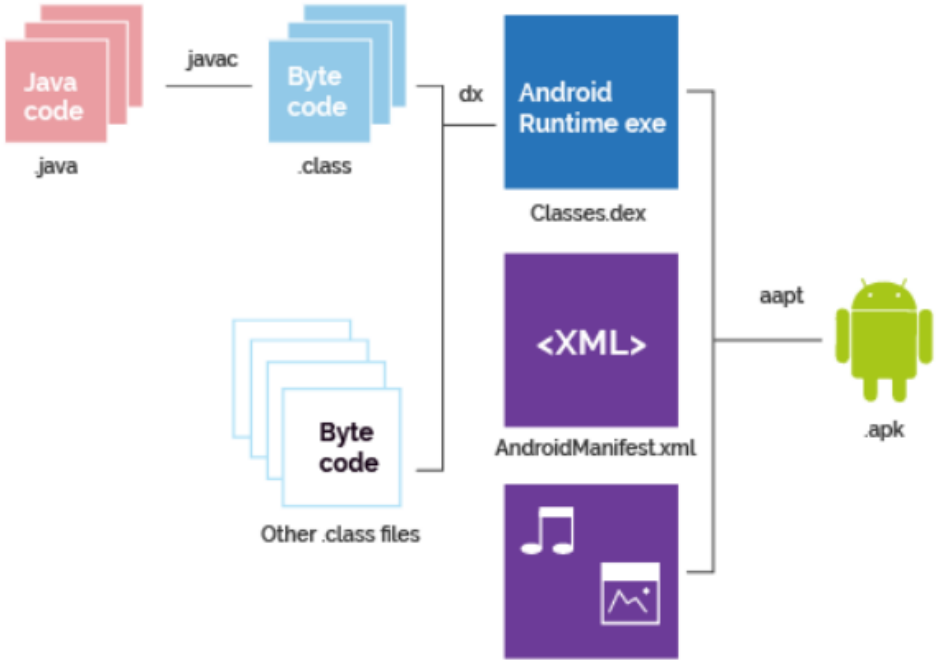


Figure3.AAPT(Android Asset Packaging Tool)

Apps can be differentiated on the basis of their behaviour.

1. **Foreground Applications**: These are the most widely recognized application. These applications have a graphical interface that is executed while the client is utilizing it. When the application shuts, these stop their execution. At the point when a client leaves the application and this isn't shut, the application is suspended and utilizes less assets. Suspended applications can be cleaned up by Android's resource management and this occurs as a last asset when the gadget is coming up short on memory.
2. **Background Application:** Applications that are executed out of sight in a silent manner. They have practically little or zero interaction with the client. They usually waits for any hardware generated event or any message to take some action.
3. **Intermittent Applications:** Intermittent applications are a mix of the two past types. They are applications that have a graphical interface where the client can interact with the application, additionally listens in the background for events. Example includes chat applications, where the client can get messages despite the fact that the application is shut and can likewise open the application and read the entirety of the messages got.
4. **Widgets:**They are applications that have a little realistic interface inside the client's home screen and are otherwise called mini-apps. The client can interact with the gadget without the need of opening the application. Gadgets are regularly used to show data to the client. A widget can likewise perform activities to control hardware peripherals. A client can move widget through the launcher's home pages or change their size.

IV. App Components

Android applications are developed with a combination of following components: Activities, Services, Content Providers and Broadcast Receiver.

ISSN: 0971-1260
Vol-22-Issue-16-August-2019

a) Activities- They are used to represent a (single) screen with a UI. Activities are a central piece of an application, where the client connects with the application. Every activity is assigned a window. Main activity is first activity that is executed when an application is launched. They are independent of each other.
b) Services- Services are utilized to perform delayed undertakings or assignments for remote processes in the background. Services don't have a graphical interface. Services are significant on the grounds that they can execute processes while the client is performing different errands.
c) Broadcast receivers- Broadcast Receivers are utilized to perform explicit errands when a system wide broadcast is identified. A portion of these broadcasts are begun by the system. For example: incoming calls and sms, low battery, network state changed, image captured, etc. Broadcast Receivers additionally absence of a graphical interface and are commonly used to show notifications or warnings to the clients.
d) Content Providers- Content Providers oversee access to an organized arrangement of information. This information can be put away on the file system, in SQLite database or another industrious storage area. The principle normal for these is that they can share this information between applications. Content providers provide an encapsulated access to data.
e) Intents- Intents are asynchronous messages that send activities, services and broadcast receivers. The most common use of intents is to move from one activity to another. They can also open other application.

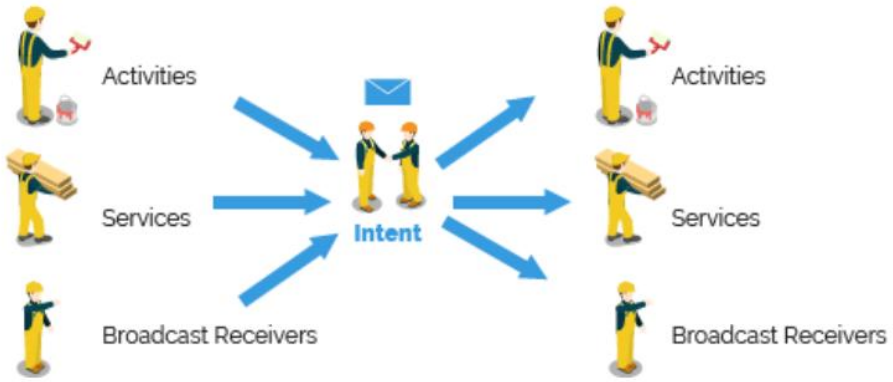
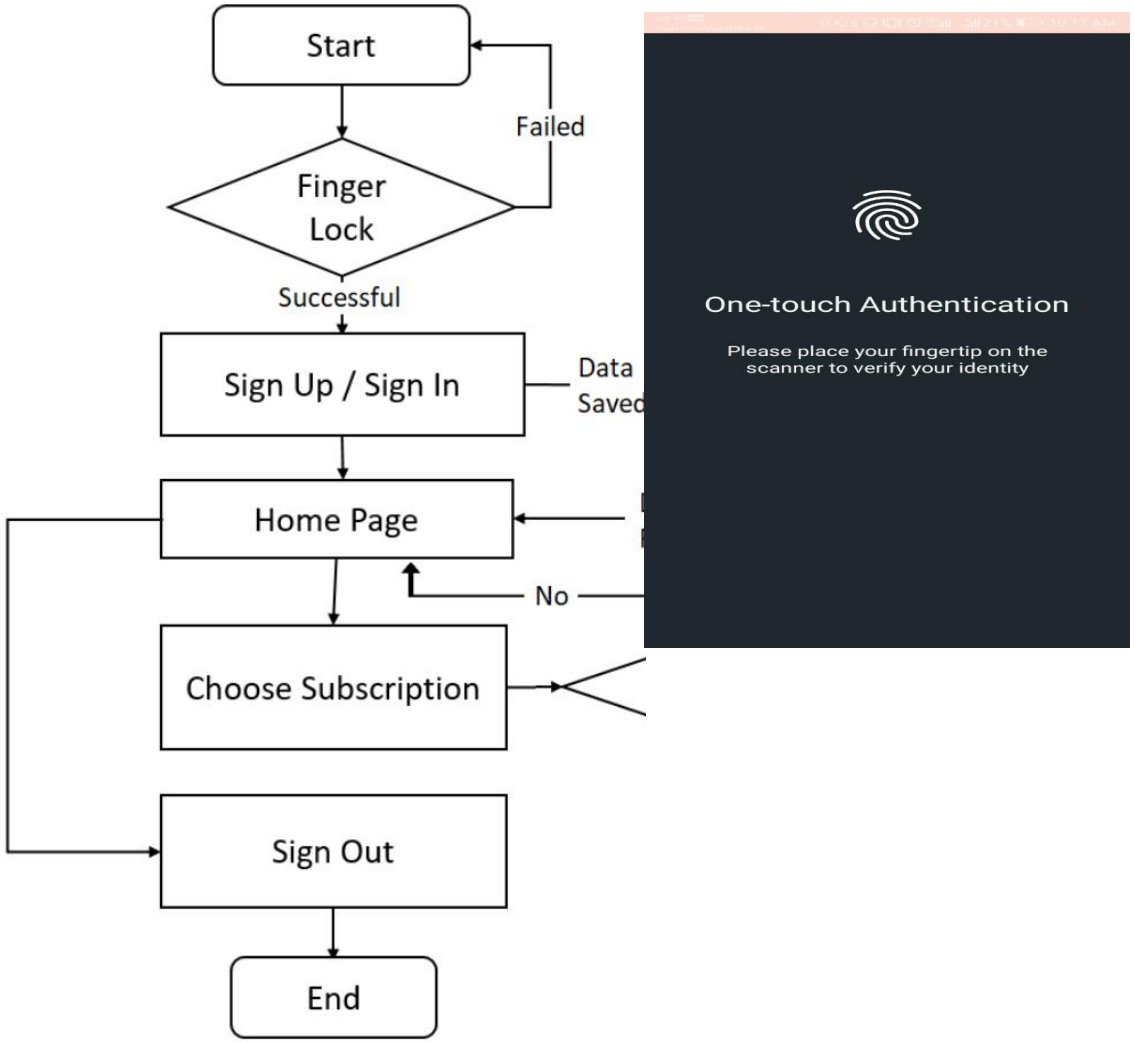


Figure4. Application content

V. Proposed Work

The proposed framework is an Android Application which helps University students in getting fresh Milk and hostel doors. Application can be used by all students staying in hostels whether he or she. Application has option of choosing one plan out of two as proposed by start-up owner. Based on feasibility user can choose plan and proceed for payment and already fixed. All the user credentials and transaction number obtained from successful payment is stored in Firebase database. Every user is uniquely identified by his or her University Registration number. Registration number and password will be credentials for Login after successful Signup.

ISSN: 0971-1260
Vol-22-Issue-16-August-2019



VI. Algorithms

❖ Finger Unlock and Login

Input: Finger Unlock followed by login credentials

Output: Home Page loaded after successful login

i. Place saved finger over finger sensor
ii. Enter login credentials for login
iii. Choose subscription plan from home page
    a. By clicking subscription button
iv. Directed to UPI supporting app available
v. After successful payment number of days left
    a. Updated.

ISSN: 0971-1260
Vol-22-Issue-16-August-2019

❖ Sign Up

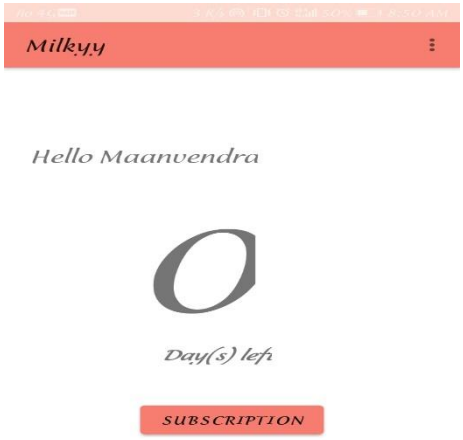
Input: Fill correct details in sign up page

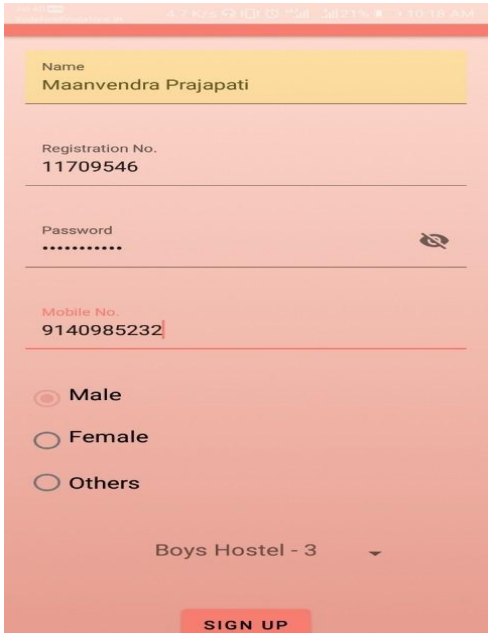
Output: Successful Sign up

i. Enter required details
ii. Details will be saved in Firebase database
iii. Home Page will be loaded
iv. Number of days left will be 0
v. Choose your subscription plan

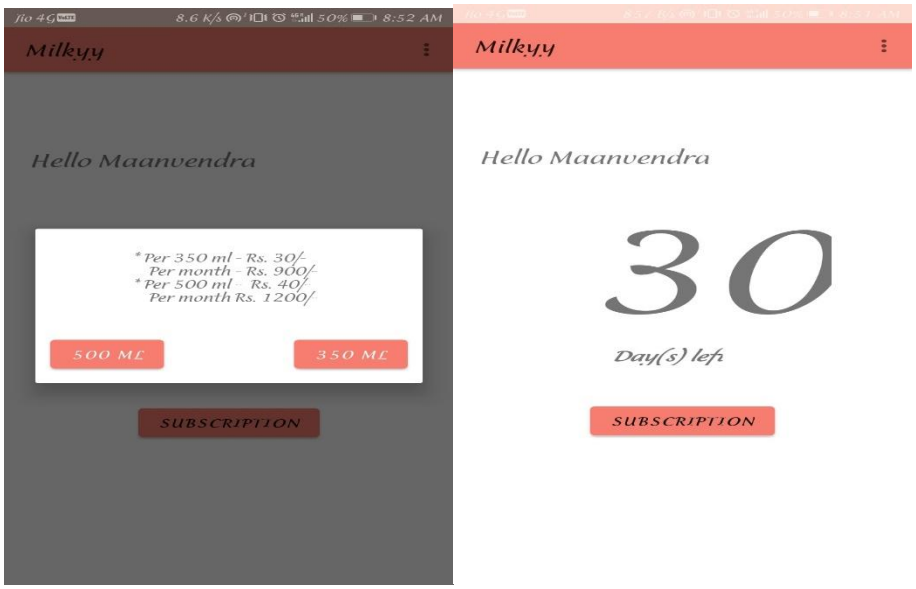
❖ Home Page

i. User name and number of days left will Be loaded from Firebase Database
ii. Subscription button helps in choosing your Desired plan.
iii. Number of days left will be updated after successful Payment.





ISSN: 0971-1260
Vol-22-Issue-16-August-2019



There is separate application for distributor, who will be delivering Milk to students having active subscription. When he delivers Milk to some student number of days left for his/her will get decreased by 1, may be more based on number of Milk units taken. It will be done by him manually. Application will display list of all the students having their Name, registration Number, Hostel and Quantity of Milk to be given.

Distributor needs to careful when clicking on delete button as it cannot be reversed back and once balance gets reduced in Database, it cannot be increased again by Distributor.

❖ Main Page Algorithm

- List of all the students active subscription Will be displayed
- Click on Delete button to decrease number of Days left
- Long click on Delete Button to remove student Detail from list

ISSN: 0971-1260
Vol-22-Issue-16-August-2019

VII. Conclusion

I developed an Android application for a small start-up of delivering Milk to university students. It was developed by using Java language, Material Design and using platforms like Firebase. This application will help university student in getting Milk at their hostel doors of their desired quantity. With this, they can save their valuable 25-30 minutes and use them in some other work. App included feedback form as well to collect feedback from users and improve application and service of start-up.

VIII. References

[1] L. Apostol and C. Dobre, "Android Fingerprint Sensor: Pitfalls and Challenges," Proc. - 16th Int. Conf. Embed. Ubiquitous Comput.EUC 2018, no. 978, pp. 133–137, 2018.

[2] P. K. Binu and V. S. Viswaraj, "Android based application for efficient carpooling with user tracking facility," 2016 IEEE Int. Conf. Comput. Intell.Comput.Res. ICCIC 2016, 2017.

[3] H. Darvish and M. Husain, "Security Analysis of Mobile Money Applications on Android," Proc. - 2018 IEEE Int. Conf. Big Data, Big Data 2018, pp. 3072–3078, 2019.

[4] R. C. Jisha, M. P. Mathews, S. P. Kini, V. Kumar, U. V. Harisankar, and M. Shilpa, "An Android Application for School Bus Tracking and Student Monitoring System," 2018 IEEE Int. Conf. Comput. Intell.Comput.Res. ICCIC 2018, pp. 1–4, 2018.

[5] S. Karthick and S. Binu, "Android security issues and solutions," IEEE Int. Conf. Innov.Mech. Ind. Appl. ICIMIA 2017 - Proc., no.Icimia, pp. 686–689, 2017.

[6] L. Li, J. Gao, T. F. Bissyandé, L. Ma, X. Xia, and J. Klein, "Characterising deprecated Android APIs," Proc. - Int. Conf. Softw. Eng., pp. 254–264, 2018.

[7] W. Yuan, H. H. Nguyen, L. Jiang, and Y. Chen, "LibraryGuru: API recommendation for Android developers," Proc. - Int. Conf. Softw. Eng., pp. 364–365, 2018.

[8]Z. Fang, W. Han, and Y. Li, "Permission based Androidsecurity: Issues and countermeasures," Computers &Security, vol. 43, pp. 205–218, Jun. 2014.

[9] J.-K. Park and S.-Y.Choi, "Studying security weaknesses ofAndroid system," International Journal of Security and ItsApplications," vol. 9, no. 3, pp. 7–12, Mar. 2015.

[10]Matsumoto T., Matsumoto H., Yamada K. & Hoshino S. "Impactof artificial" gummy" fingers on fingerprint systems Optical Securityand Counterfeit Deterrence Techniques," IV (Vol. 4677, pp. 275-290).International Society for Optics and Photonics.