1. INTRODUCTION

The design of secure authentication protocols is quite challenging, considering that various kinds of root kits reside in PCs (Personal Computers) to observe user’s behavior and to make PCs untrusted devices. Involving human in authentication protocols, while promising, is not easy because of their limited capability of computation and memorization. Therefore, relying on users to enhance security necessarily degrades the usability. To that end, we propose two visual authentication protocols: one is a one-time-password protocol, and the other is a password-based authentication protocol.

Our approach to solving the problem is to introduce an intermediate device that bridges a human user and a terminal. Then, instead of the user directly invoking the regular authentication protocol, she invokes a more sophisticated but user-friendly protocol via the intermediate helping device. Every interaction between the user and an intermediate helping device is visualized using a Quick Response (QR) code. The visual involvement of users in a security protocol boosts both the security of the protocol and is re-assuring to the user because she feels that she plays a role in the process. To securely implement visual security protocols, a smartphone with a camera is used. Instead of executing the entire security protocol on the personal computer, part of security protocol is moved to the smartphone. This visualization of some part of security protocols enhances security greatly and offers protection against hard-to-defend against attacks such as malware and keylogging attack, while not degrading the usability. However, we note that our goal is not securing the authentication process against the shoulder surfing attacker who can see or compromise simultaneously both devices over the shoulder, but rather to make it hard for the adversary to launch the attack.

1. OBJECTIVES

Sequoro aims at providing an economical way to provide secure login to websites .It is secure against common password stealing attacks including key loggers. We aim at realizations of protocols that not only improve the user experience but also resist challenging attacks, such as the key logger and malware attacks. Our protocols utilize simple technologies available in most out-of-the-box smartphone devices. Our work indeed opens the door for several other directions that we would like to investigate as a future work. In future, we plan to implement our protocol on the smart glasses such as the Google glass, and conduct the user study.

1. REQUIREMENT SPECIFICATION

**3.1 Software Requirements**

Operating system : Android

Platform : Android Studio

Client Side : Android

Server Side : JSP

**3.2 Hardware Requirements**

Processor : 1.6 GHz or faster processor.

Memory size : 1GB RAM.

Storage : 10 GB Hard Disk.

Keyboard : Virtual keyboard with 102 keys.

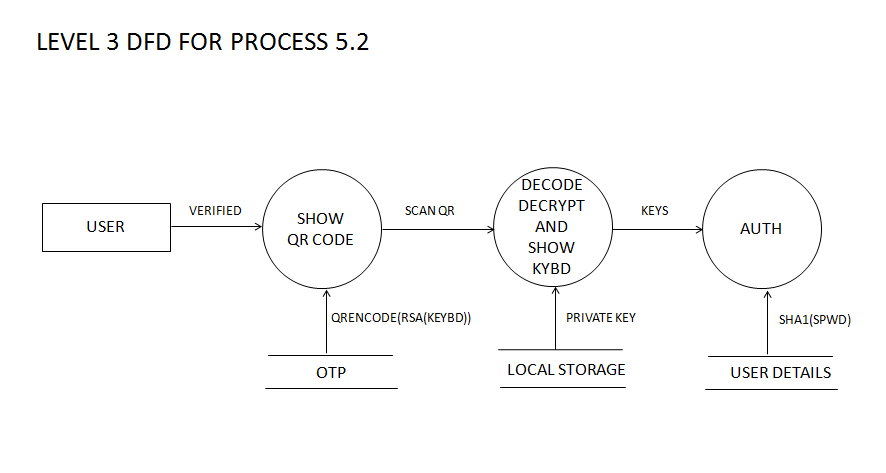
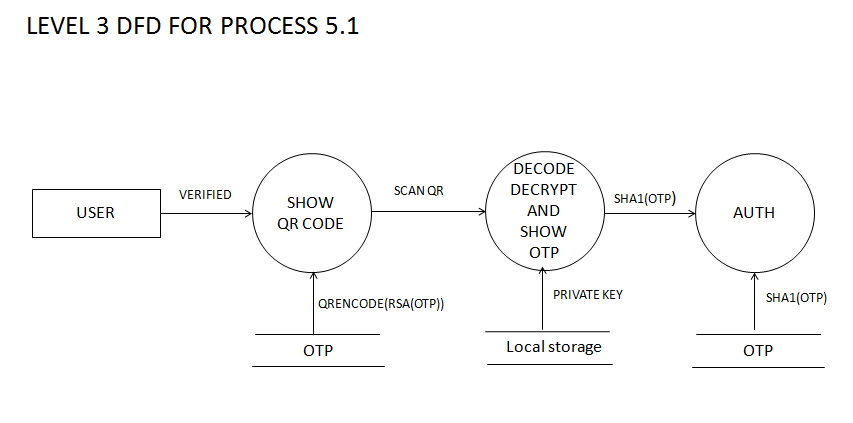
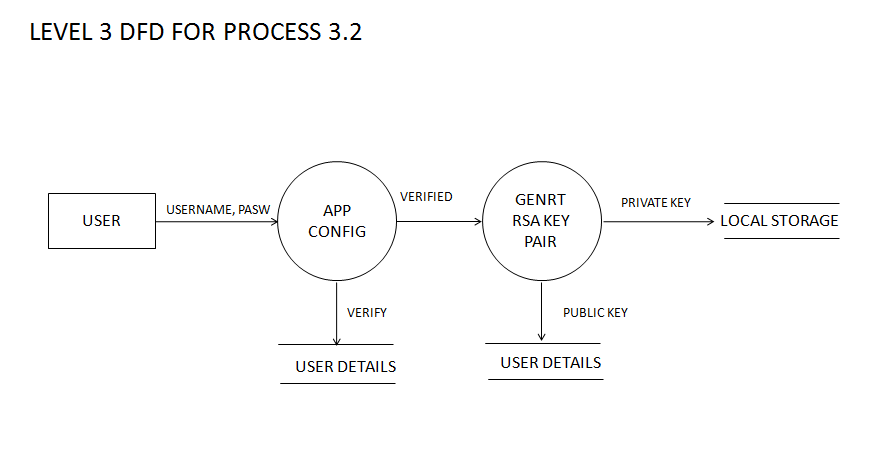
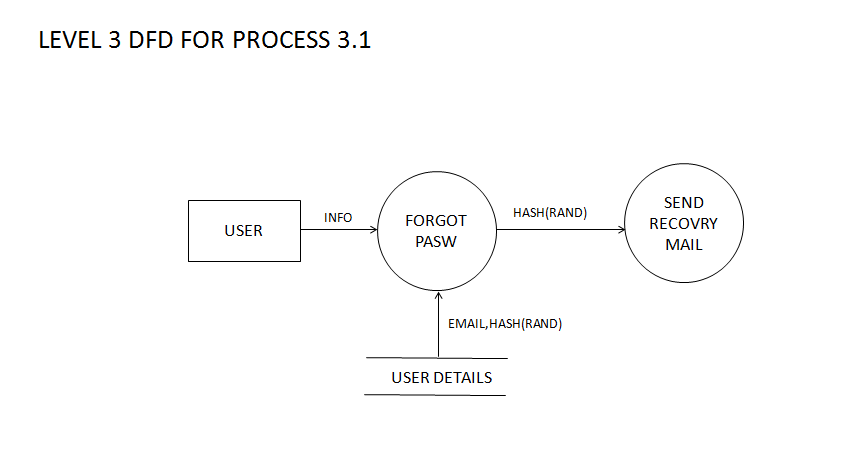
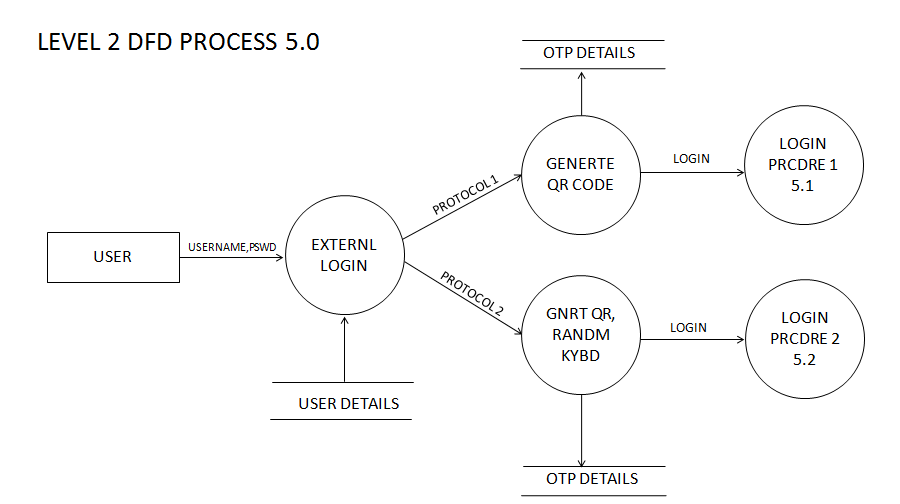
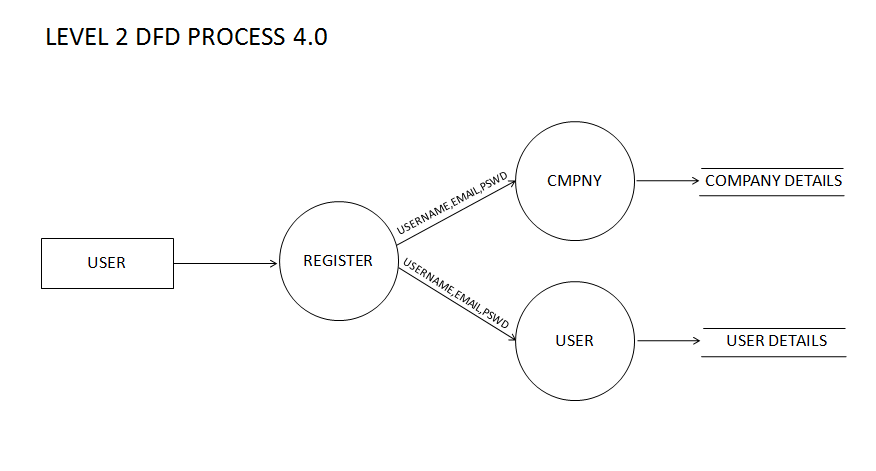
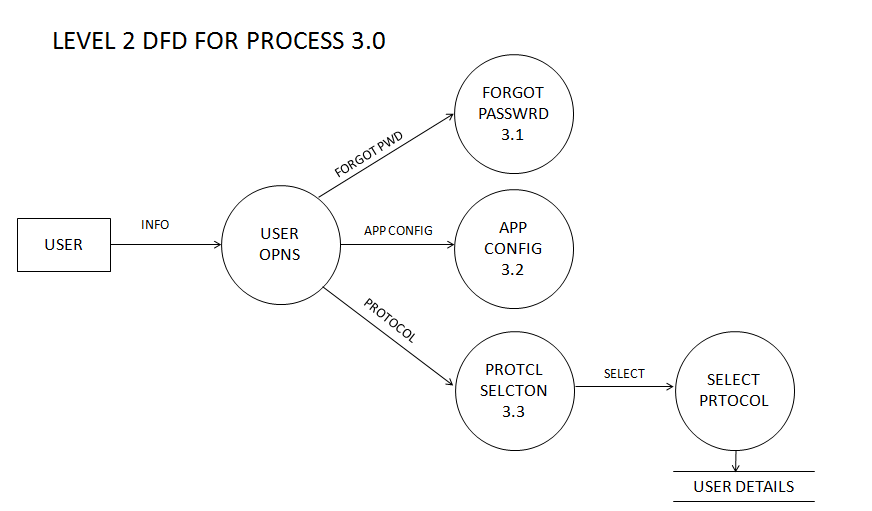
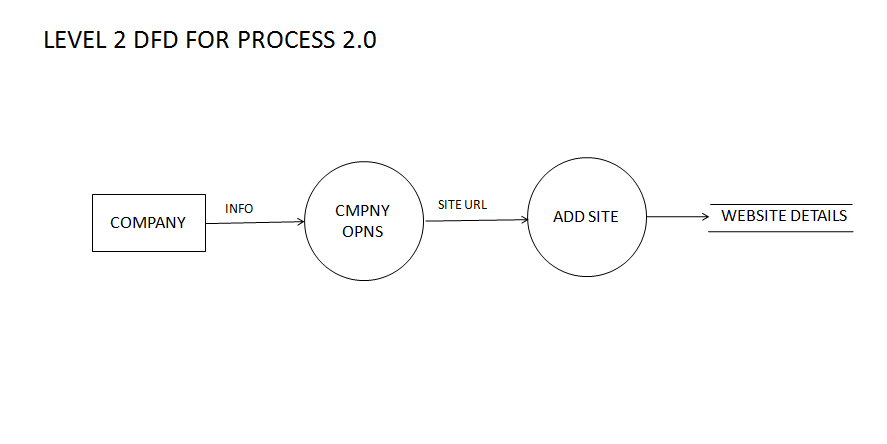
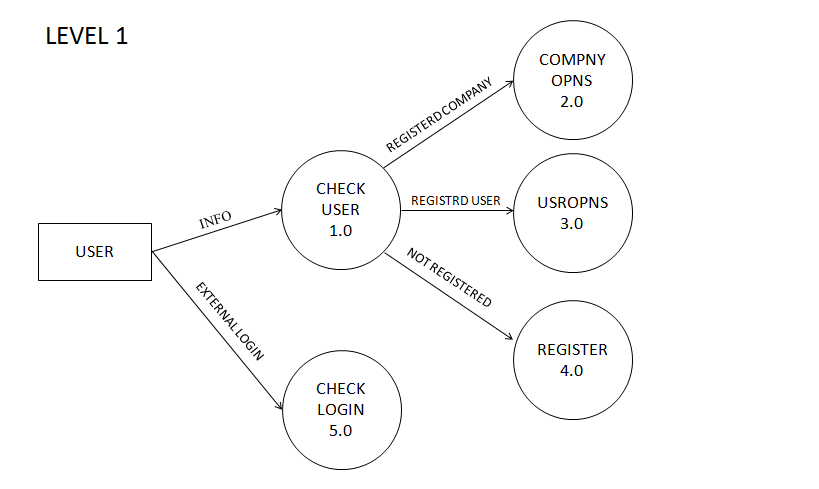
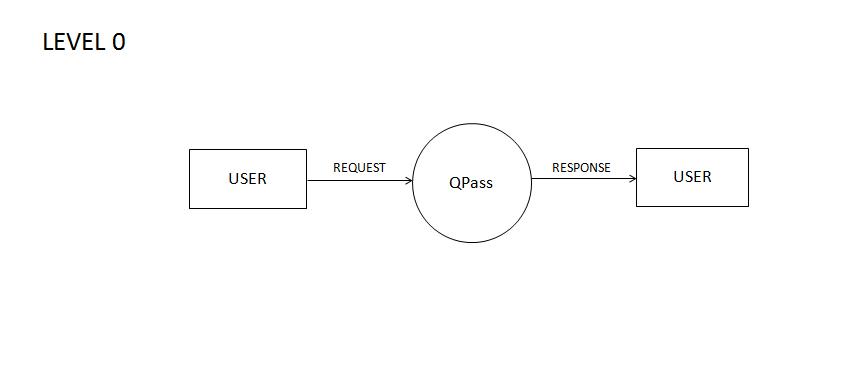
Device : Android Mobile with 2MP or higher camera.

1. MODULES

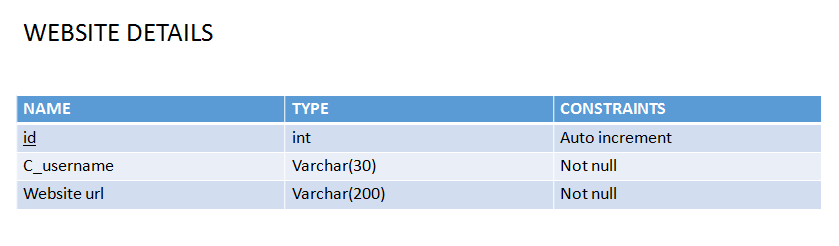
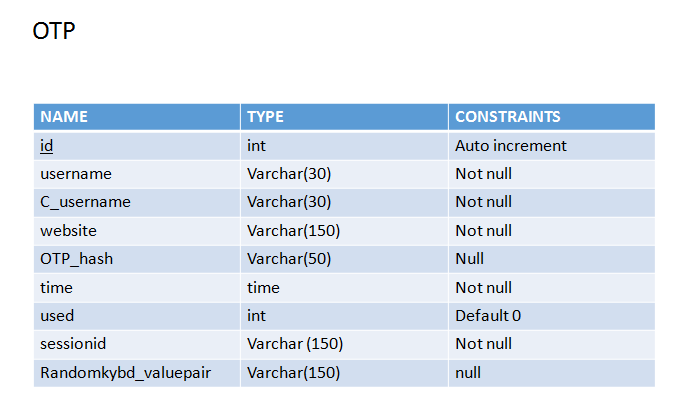
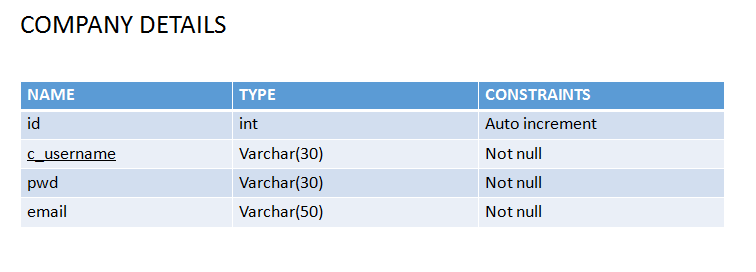
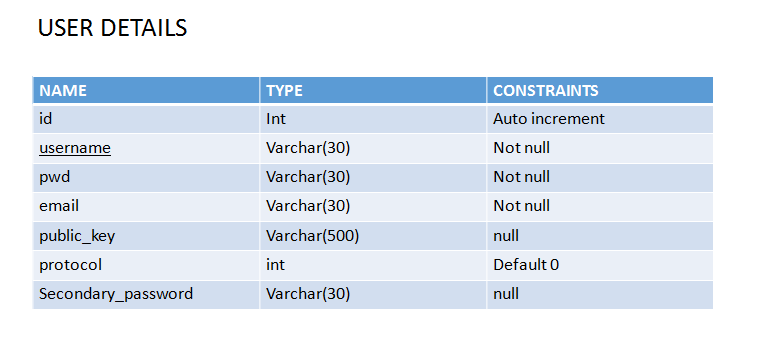
There are 4 modules in this project.

* REGISTRATION
  + Username, password and email address are collected from users and companies
  + Companies can enter their website details
  + User can select protocol and specify secondary password
* APPLICATION CONFIGURATION
  + Verified user can only use the system
  + RSA 2048 bit key pair generation
  + Public key is uploaded to server
  + Private key is saved to local storage
* QR CODE GENERATION
  + User is verified and protocol selected
  + For protocol 1 OTP is generated, encrypted and encoded to QR code
  + For protocol 2 random keyboard layout is created ,encrypted and encoded to QR code
* LOGIN PROCEDURE
  + User is verified , QR code is generated and displayed
  + User scans the QR Code
  + For protocol 1 OTP is decoded from QR code, decrypted and displayed
  + For protocol 2 Keyboard layout is decoded from QR code ,decrypted and displayed

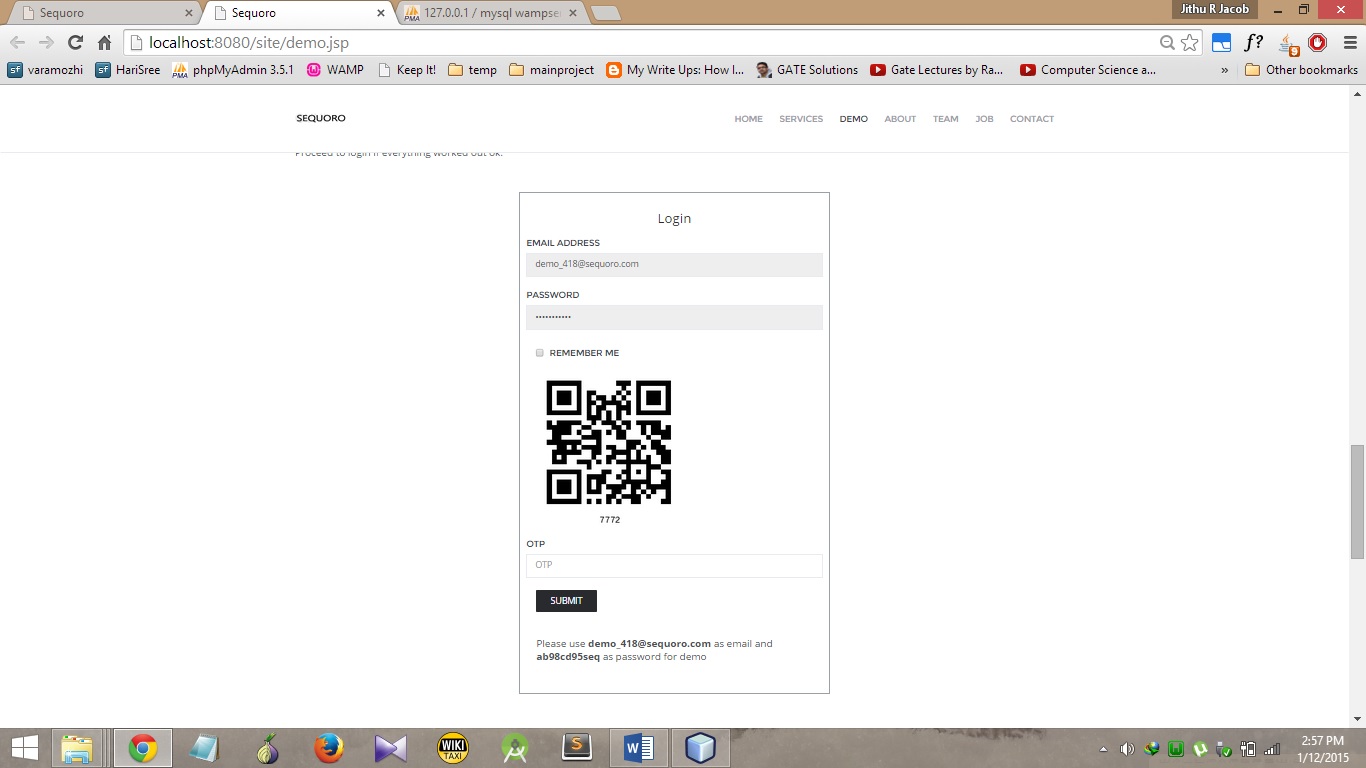
1. DATA FLOW DIAGRAMS



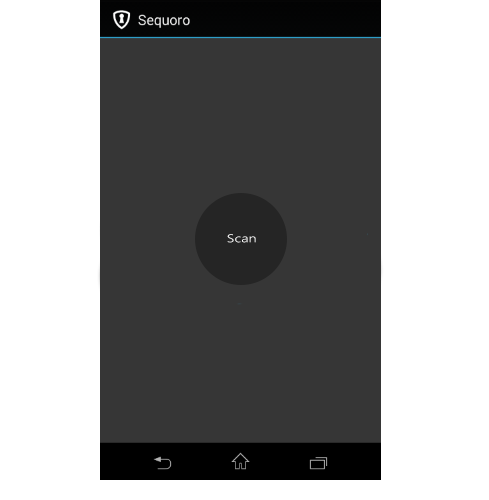
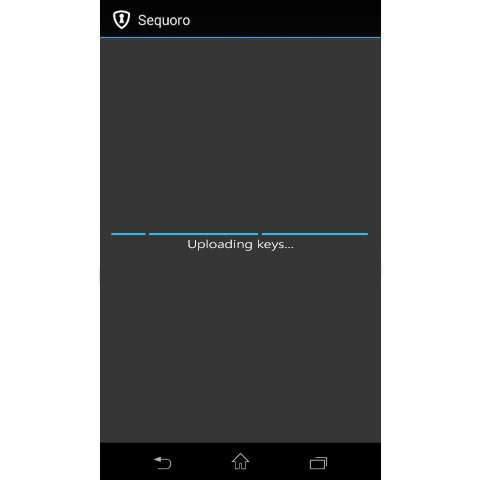
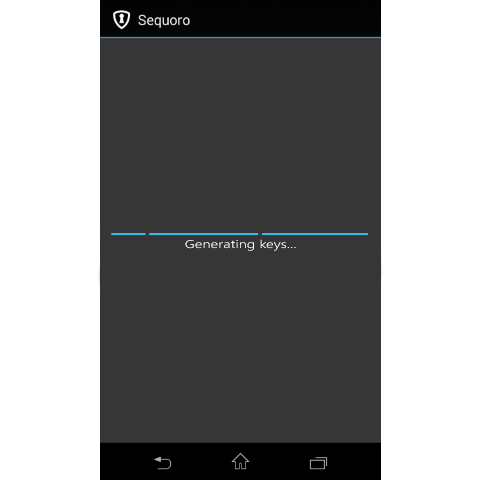
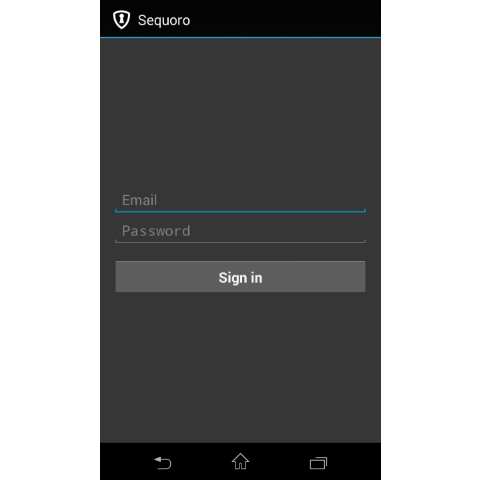
1. TABLE DESIGN



1. SCREEN SHOTS



ANDROID APP INTERFACE



1. CONCLUSION

The proposed system thus helps us to provide a universal economic secure solution. Proposed system not only help to find who took our phone, but also help as to keep our mobile phone secure from unknown persons. It help us to keep our personal data away from unknown person. Our approach visualizes the security process of authentication using a smartphone aided augmented reality.

Sequoro aims at providing an economical way to provide secure login to websites It is secure against common password stealing attacks including key loggers. We aim at realizations of protocols that not only improve the user experience but also resist challenging attacks, such as the keylogger and malware attacks. Our protocols utilize simple technologies available in most out-of-the-box smartphone devices. Our work indeed opens the door for several other directions that we would like to investigate as a future work. In future, we plan to implement our protocol on the smart glasses such as the google glass, and conduct the user study.

1. REFERENCES

[1] —. Google authenticator. http://code.google.com/p/google-authenticator/

[2] —. Rsa securid. http://www.emc.com/security/rsa-securid.htm.

[3] Cronto. http://www.cronto.com/.

[4] —. BS ISO/IEC 18004:2006. information technology. automatic identiﬁcation and data capture techniques. ISO/IEC, 2006.

[5] —. ZXing. http://code.google.com/p/zxing/, 2011.

[6] D. Boneh and X. Boyen. Short signatures without random oracles. In Proc. of EUROCRYPT, pages 56–73, 2004.

[7] J. Bonneau, C. Herley, P. C. Van Oorschot, and F. Stajano. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In Security and Privacy (SP), 2012 IEEE Symposium on, pages 553–567. IEEE, 2012.

[8] J. Brown. Zbar bar code reader, zbar android sdk 0.2.http://zbar.sourceforge.net/, April 2012.

[9] C.-H. O. Chen, C.-W. Chen, C. Kuo, Y.-H. Lai, J. M. McCune, A. Studer, A. Perrig, B.-Y. Yang, and T.-C. Wu. Gangs: gather, authenticate’n group securely. In J. J. Garcia-Luna-Aceves, R. Sivakumar, andP. Steenkiste, editors, MOBICOM, pages 92–103. ACM, 2008.

[10] S. Chiasson, P. van Oorschot, and R. Biddle. Graphical password authentication using cued click points. In Proc. of ESORICS, 2008.

[11] D. Crockford. The application/json media type for javascript object notation (json). http://www.ietf.org/rfc/rfc4627.txt?number=4627, July 2006.