Business Card Wallet

An Android application to promote yourself

Table of contents

01

Overview

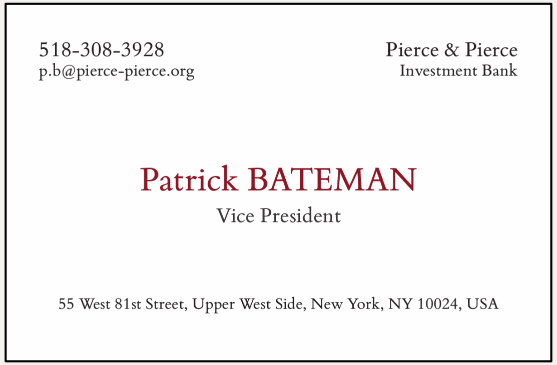
Goals

The project aims to create a digital business card wallet that:

- It is easy to maintain
- Can share business cards
- Keep secure your business card

In a simple and efficient way!

...in other words

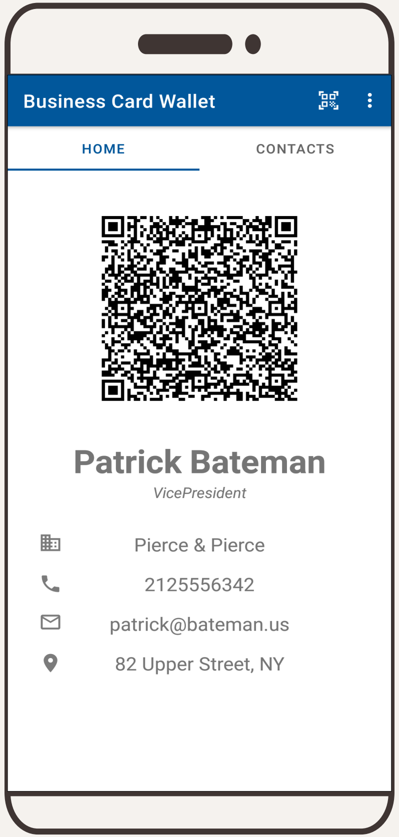


518-308-3928
p.b@pierce-pierce.org

Pierce & Pierce
Investment Bank

Patrick BATEMAN
Vice President

55 West 81st Street, Upper West Side, New York, NY 10024, USA



Business Card Wallet

HOME CONTACTS

Patrick Bateman
VicePresident

Pierce & Pierce
2125556342
patrick@bateman.us
82 Upper Street, NY

Functionalities

- Log in with your account or make a new one
- Show the QR code associated with your business card
- Real-time sync accross different devices
- Decode QR codes encoded business cards with a dedicated scanner
- Interact with a business card by tapping on the displayed info
- Share or Print a business card as a QR code
- Import a business card from an image containing its QR code
- Manage a contact list and search through them
- Update your business card and your account info

02

Arhitecture

Structure

AuthenticationActivity

LoginFragment
RegisterFragment

MainActivity

BusinessCardFragment
ContactListFragment

ContactInfoActivity

BusinessCardFragment

SettingsActivity

SettingsFragment
EditAccountFragemnt
EditUserFragment

ScannerActivity

CustomBarcodeScanner

Navigation

- The AuthenticationActivity allows you to log in through LoginFragment or create a new account using RegisterFragment
- Once authenticated the user is redirected to MainActivity which will pull user data and contacts from the database and will display them using a tab layout. From here he can check his business card, through BusinessCardFragment or his contacts, through ContactListFragment.
  - The BusinessCardFragment allows the user to share or print through wifi the displayed QR code
  - Selecting a contact in ContactListFragment, will start ContactInfoActivitywhich will reuse BusinessCardFragment. The displayed information are interactive (es. phone dealer when tapping on phone number)
- The SettingsActivity will sync the local preferences with the one stored in cloud and will display the preferences using SettingsFragment. The user can modify his business card through EditUserFragment or can modify his account through EditAccountFragment. The updated data are automatically sinc with the cloud database.

Navigation 2

- From MainActivity the user can start the ScannerActivity which will start the camera and manage the permissions. If a valid QR code will be detected the app will add the decoded business card to the contacts list and will start a ContactInfoActivity
- A QR code can be decoded from an image stored in the device, by using Import from MainActivity

03

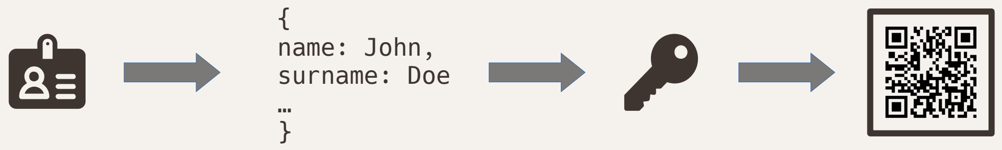
Implementation

Storage and Authentication

- Authentication is done using Firebase Authentication (email)
- The data are stored in Firebase Firestore, a NoSQL database
- Firestore consists in Documents, stored in a Collection. A Document contains pairs of attributes and values and can have Collections as well
- In the app database each user is a Document consisting in user’s business card information and a Collection, Contacts, containing the contacts that the user has scanned
- Whenever a user changes a sync data (es. Delete a user from his contact list) an async funcion will update the remote data as well
- To improve modularity the database and authentication functions are implemented passing through a DatabaseUtils class

QR code and encryption

- The QR code is generated “on the flight”
- The Contact is parsed into a Json string using Google Gson API
- The Json string is encrypted using a AES, a symmetrical encryption algorithm, to make the string “proprietary”
- The encrypted string is converted into a QR code using Zxiang API
- The Zxiang API provides also the scanner and retriving the object follows the same steps but in reverse



Share, Print and Import

- Print is implemented using Google API
- Share is implemented by saving the bitmap qr code in a cache file and then starting an intented on this file, passing through a content provider (check issues)
- Import is implemented by using Zxiang API, by cropping and prelaborate the qr code image into a bitmap and then decoding it, similarly how it is done with the scanner

04

Faced issues

Faced issues

- ViewPager2 and TabLayout are broken with dynamic fragment
- From SDK 24 you need a Content provider to access a file that is not in your app storage
- Async database functions
- Asymetric encryption latency (RSA)

Thanks