CECS 453 - Android

Final Project : NFC Transfer App

App Name : NFC Transfer
Version : 1.0
Platform : Android Api 25
Minimum : SDK : Android Api 19
Contact : gottingoscar@gmail.com
Developers : Oscar Gotting, Maxime Boguta, Nicolas Charvoz, Luca Demmel

App Summary

This app acts like a 2.0 business card, using NFC (Near field communication). It allows the user to create its own profile, by adding personal data to it such as email or cellphone number, but also to attach social media accounts such as Twitter, Facebook etc… Once the profile is ready, the user sticks his phone with one other phone, and both profiles are exchanged thanks to NFC. He can then import all the data into his contacts, and follow the person he just met, on the social medias.

Back-end

This application includes networking. Two reasons explains that : First NFC is unidirectional, that would mean that both users would have to stick their devices two times. It's not convenient.
Our NFC procedure only transmits the user id. Using an API, the user can notify the other user that he has to pull data related to the profile. Secondly, we wanted to provide an account system for our users, so they can have their profile stored in the cloud, and they can retrieve it on every device they log-in. For information, we have developed an operational backend using Node.js, Express, MongoDB, Twilio, and NGINX.

External Libraries

- android-recyclerview (The famous Android recyclerview which is a more advanced listview)
- sdp-android (A cool library that handle smart measurements for views)
- kyleduo-switchbutton (A fork of the Android SwitchButton with more style and functionalities)
- markushi-circlebutton (A circle button with a nice effect when clicked)
- sdp-android (A powerful library that recalculates all the dimensions)
- squareup-retrofit2 (Networking library)
- socketio-socket.IO (WebSockets library)
- googlecode-libphonenumber (Utils for the phone numbers)
- ogaclejapan-smarttablayout (A nice library that provides animations and a TabLayout for the view Pager)

Application

Project Structure : Everybody 4 days

- Defining the project structure
- Defining the needs, and research about external libraries
- Creation of the App from scratch, adding external libraries
- Initialization of the project structure
- Initializing the Manifest file, Adding the permissions, activities

UserPreferences & Session : [Oscar Gotting] ½ day

- Creation of a Session singleton that holds data related to the session (userid, accesstoken etc)
- Encapsulation of the SharedPreferences, with a singleton
- Retrieving, accessing and committing changes to the SharedPreferences

Database : [Luca Demmel] ½ day

- Implementing the SQLite database
- Creating columns and functions initializing DB
- Creating method for the CRUD

Launcher Activity : [Maxime Boguta]

- Initializing the launcher, register it as the actual launcher
- Initializing all the Social app SDKs (Fb, LinkedIn, etc)
- Initializing all our Singletons and important components
- Retrieving/Saving data in the UserPreferences to determine if it's first launch, if an access token is saved

WebSockets : [Luca Demmel + Maxime Boguta] ½ day

- Implementiating Socket.io
- Creation of a Websocket watcher/manager/interfaces that notifies registered callbacks
- when the socket is connected/disconnected
- Implementing all the websocket events

Models : [Oscar Gotting + Luca Demmel] ½ day

- Creation of the Profile model
- Creation of the Field model (AProfileField, CellphoneField, EmailField, FacebookField etc)
- Creating all the models that will be serialized/deserialized to/from the API
- Defining all the properties for each model
- Defining enums for the model properties (EntityType, DeletableType etc)
- Creation of a factory function that creates empty or pre-filled models

Networking [Oscar Gotting] 1 day

- Encapsulation for the serialization, with interface IAPIResponse, AuthResponse etc
- Initialising and Implementing the networking library
- Creation of the NFCTransferService (contains all the routes)
- Creation of a Singleton used for networking (NFCTransferAPI)
- Creation interfaces and callbacks
- Linking all the routes
- Defining an enum with all the HTTP codes

Social SDKs [Oscar Gotting + Luca Demmel] 2 days

- Creating an encapsulation that provides an abstraction of connecting
- any kind of SDK, synchronizing with the user account, and retrieving data
- such as the user name and userid (ASocialAPI)
- For all the SDKs the tasks are pretty much the same
- Registering the app on the developer console
- Generating key hashes,
- Integrating the SDK to the app

- Facebook SDK implementation : **[Luca Demmel]**
- Twitter SDK implementation : **[Oscar Gotting]**
- LinkedinSDK implementation : **[Luca Demmel]**

Social SDK Viewer : [Maxime Boguta] 1 day

- Controller that provides abstraction to visualize a user social account (SyncProfileViewer)
- Facebook viewer to open facebook or browser
- Twitter viewer to open twitter or browser
- Linkedin viewer to open twitter or browser

NFC and BeamActivity : [Oscar Gotting + Maxime Boguta] 2 days

- Research about NFC and Android beam
- Implementing a library that helps creating NDEF tags
- Implementing all the functions for NFC tag exchange
- Connecting to the API to pull the user profile
- Sending the user profile to the ProfileDisplayerActivity

UserLogin Activity : [Nicolas Charvoz + Oscar Gotting] 1 day

- Creation of all the UI
- Creation of all the UI resources, such as drawables, icons, colors
- Implementing the IntlPhoneInput library with all the flags for every country
- Implementing keyboard listeners
- Implementing custom animations (fades, eases)
- Implementing all the listeners, and actions, callbacks
- Creating all the context menus
- Linking the networking part

ConfirmAccount Activity : [Maxime Boguta] ½ day

- Creation of UI, animations
- Handle networking
- Error handling, error messages with dialogs

MainActivity [Nicolas Charvoz + Luca Demmel] 2 days

- Creation of the UI
- Creation of the app Style and theme, chosing colors
- Creation of the menus, context menus, options menus
- Creation of the toolbar
- Creation of a ViewPager, spent time on the design
- Initialize the ViewPagerAdapter that contains 3 fragments
- Registering for WebSocket events

AddFieldActivity : [Luca Demmel] less than ½ day

- Creation the UI, layouts
- Creation of a ListView
- Creation of the ListView adapter
- Retrieving data from intent
- Sending back data to parent intent

ProfileFragment [Oscar Gotting + Luca Demmel + Nicolas Charvoz] 1 day

- Initialize the fragment, creating the UI
- Implementing the RecyclerView
- Creating the rows for the RecyclerView
- Implementing an adapter for the RecyclerView, with ViewHolder that implements listeners
- Opening the good context menu regarding the type of the item in the recyclerview

- Creating the CRUD functions, Add/Edit/Delete field
- Implementing a rollback method
- Opening the AddFieldActivity, sending data to this activity regarding what it has to display
- Error handling, error dialogs
- Controls handling, hit back button to go to previous screen

ActionShareFragment : [Oscar Gotting] less than ½ day

- Creation of the UI, layouts, adding the CircleButton

MatchesFragment : [Nicolas Charvoz] less than ½ day

- Retrieving data from the DB
- Creation of Listview
- Creation of adaptper
- Creation of listeners, send data to profileDisplayer

ProfileDisplayer : [Nicolas Charvoz] less than ½ day

- Creation of the UI, layouts
- Creation of two listviews, displaying the target User Data
- Creation of two Adapters, for the two listviews
- Deserialise received data and generate the view
- Adding custom button to the toolbar

ContactAdder : [Maxime Boguta] less than ½ day

- Methods to create a contact entry in the phone, and add data to it

AlertDialogTools : [Oscar Gotting]

- Tools that we use to populate the AlertDialogs
- Generate proper keyboard, title and messages
- Bind callbacks, with interfaces

FieldEntryParser : [Nicolas Charvoz]

- A parser class that parses the inputs regarding their type
- Implementing regexes

Custom Views (code from stackoverflow) [Nicolas Charvoz]

- TintableImageView
- TabLayoutIconResizer