

## **Project Iron - CRC and Software Architecture diagram**

## Table of Contents

1. Crc Cards	page 2
2. Software Architecture diagram	page 3
3. Software Architecture explanation	page 4

## Section 1 - Crc cards

Class: LoginActivity

Parent Class: AppCompatActivity

Responsibilities:

1. onCreate - Sets up the login form
2. attemptLogin - Attempts to login to account
3. isEmailValid - Validate Email
4. isPasswordValid - Validate Password
5. showProgress - Shows the progress UI/Hides login form
6. addEmailsToAutoComplete - adds emails to database
7. ProfileQuery - Stores User's profile
8. \*Lets you login with/register a user profile

Collaborators:

1. UserLoginTask

Class: UserLoginTask

Parent Class: AsyncTask

Responsibilities:

1. doInBackground
2. onPostExecute
3. onCancelled
4. Represents a login/registration task to authenticate user

Collaborators:

- 1.

Class: RegisterActivity

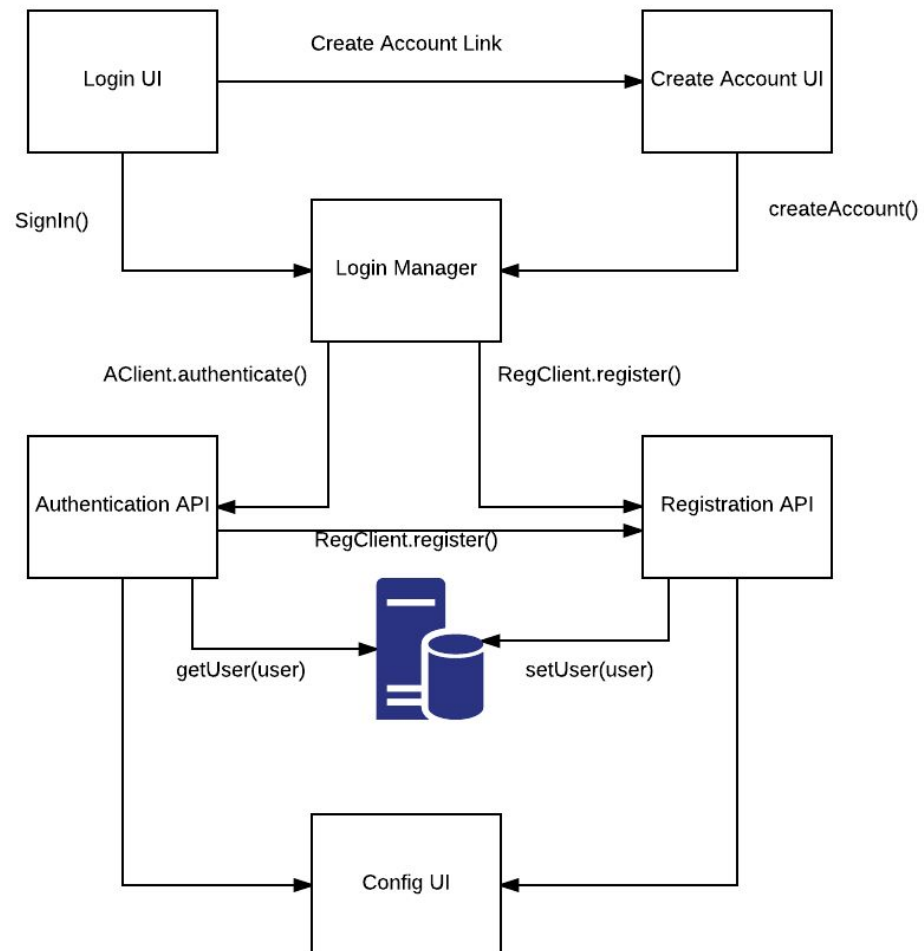
Parent Class: AppCompatActivity

Responsibilities:

1. onCreate

Collaborators

## Section 2 : Systems Architecture design



### Section 3: System Design explanation

LoginManager:

// this holds the different options the user can choose, from both login pages. We pass this through APIs to do condition checks

ENUM = {FB, GOOGLE, CUSTOM}

Objects:

AuthenticationClient

RegisterClient

UserObj = new User(getEmail(), getPassword())

Methods:

// from the login page when the user clicks FB or Google login. Assume they are existing users now, if not will prompt to register later

signIn()

// from the create account page, goes straight to register api

createAccount()

from signIn() do AuthenticationClient.authenticate(ENUM\_CHOSEN, userObj)

from createAccount() do RegisterClient.register(ENUM\_CHOSEN, userObj)

try catch all this or throw, we may catch exceptions from other apis like "user already exists" from registration api. This gets displayed on the screen, maybe similar to how validation is

AuthenticationAPI:

Objects:

RegisterClient

methods:

authenticate(ENUM\_CHOSEN, userObj):

- read firebase check if exists read(userObj.getEmail())
- if exists then move on, change view
- if not RegisterClient.register(ENUM\_CHOSEN, userObj)

RegistrationAPI:

Objects:

methods:

register(ENUM\_CHOSEN, userObj)

- if fb enum go to fb create page to confirm
- if google enum go to google create page to confirm
- custom login goes straight to register

- writes user to database
- changes view to config

fbRegisterPage()

googleRegisterPage()

errors:

user already exists