アクセル娘 Project Development Reference

Author : 謝侑驊

Team Name : 碼農們 **(TEAM NO.19)**
Team member : 謝侑驊 **,**王祥至

1.Introduction

This is an **android application** which has functions to make user's cellphone much more safer in security, and brings conveniences to manage one's cellphone, i.e., memory clean, app management.

In order to make this app much more intriguing, we also use **Live2D Inc. SDK** in the app, which make user be able to have **interactive action with anime character** and clean memory.

Besides, we add an **market** to make users be able to **use coins to download** new Live 2D's theme to add into their app. Coins can be collected by downloading recommended app or In-app billing (Google play).

2.Client Functional requirement

2.1 Clean Memory

By touching **Live 2D** anime character, app can kill all other **processes** to release memory. Also, need an **list** to let user to choose which processes currently running need to be killed.

2.2 Application Lock

This is an **Security** function. User can set an **password** and choose which app they want to lock. Once this app is locked, the user need to enter the password to open.

2.3 Application manager

List all the apps installed in this cellphone and manage them (start, uninstall, share)

2.4 Phone security (Anti theft)

This is an **Security** function. Use one "Service" as an listener to monitor the **SIM card**, once the SIM has been changed, phone secretly **send SMS** containing GPS location and thief number to pre-set number.

Also, user can send specific SMS to lost cellphone to **wipe out their data remotely.**

2.5 Coin market (Connection to server)

User can **register** new account, **login**, and use **coin** to buy more anime character in the market, and use **In-app billing** services to buy more coin. Use **Retrofit** - a **type-safe REST client** to communicate with server.

3.Server Functional requirement

3.1 Database

Manage user account, coins and their purchases by **MongoDB**.
(Why not SQL ? just for learning NoSQL...XDDD)

3.2 Connection

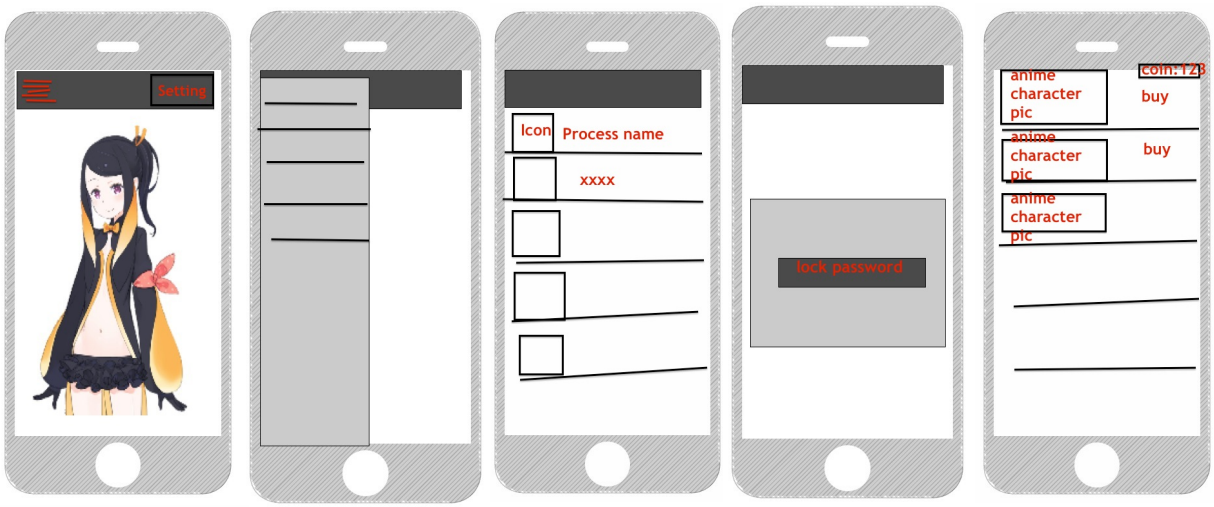
Use and **Apache Tomcat Server** and **Java EE Spring framework** to manage all the **https** connections and route the requests to the specific servlet and implement each servlet to process corresponding data.

4.Design and UI

4.1 Gamification

we add some gaming element in this app such as interactive Live 2D anime character which makes cleaning memory much more fun, and use **coin system** to attract users to download our promoting app or to pay.

4.2 User Interface (UI)



From left to right, from top to down :
Main UI (Anime character)
Sliding Menu, Process Manager ListView
App Lock Enter Password Dialog
Coin Market ListView
App lock ListView
App Manager ListView

5.Software Environment requirement

Client Development Environment : Android Development Toolkit
Server Development Environment : Eclipse Java EE, Gradle 2.2
Client Runtime : Android 2.23 (or higher version.)
Server Runtime : Apache Server, J2EE Spring Framework , MongoDB

6. Communication Interface with server

6.1 Communication interface address.

https://localhost:8443/accel

6.2 Communication methods.

All communications are under **Https**, the upload parameters are all **encoded in URL query**, and the return value are encoded as **Json.**

e.g :
https://localhost:8443/accel/login
{“status”:1,”userInfo”:{“name”:”qw^^GH@HJDSD”,”pass”:”123”}}

6.3 Params

6.3.1 Client TokenLogin

address: https://localhost:8443/aceel/login

Params		Return	
String phoneNum	phone_sha1	int status	0 = token expired 1 = success
String token	token the client keep		

6.3.2 Client logout

address: https://localhost:8443/accel/logout
@GET(value="/logout")
no return value.

6.3.3 Client get account status

Params		Return	
int status	233	Purchase pr	an Object Purchase which contains all the user purchase and coins.

6.3.4 Client promote App (This will be defined when developing)
6.3.5 Client Buy character (This will be defined when developing)
6.3.6 Client Delete character (This will be defined when developing)
,and so on.

7.Software Development Model

We use **Scrum** as our agile development model, which means before developing, everything has not been defined clearly. We list our **to-do list** (see 2. and 3.) and use **sprint** to divide the developing processes. Before each sprint, we choose task from our to-do list, and decide what to do in this sprint.

Why we choose Scrum rather than Waterfall model ? Because we have very few android developing experience, we might need to change our design, thoughts and time limitation, which means Waterfall model is a bad choice for us, we want to use an agile method as our model.

8.Division of Work

謝侑驊	Android SDK research Server Servlet (Spring) MongoDB management integrating Live 2D into app(SDK research) Client connection to server (Retrofit) App function - App Lock App function - Coin Market
王祥至	Android SDK research Android UI design (xml) App function - Clean Memory App function - App Manager App function - Phone Security App - SlidingMenu research. Beautify UI (Use open source library,i.e.,SmoothTransition)

9.Reference

1. http://projects.spring.io/spring-framework/
2. http://square.github.io/retrofit/
3. http://www.live2d.com
4. http://developer.android.com/index.html
5. https://github.com/jfeinstein10/SlidingMenu
6. https://github.com/dkmeteor/SmoothTransition