

CosplayBook

Informacje

- Aplikacja sieciowa na system operacyjny Android. Portal społecznościowy dla cosplayerów.
- DB: MySQL
- BackEnd: Spring (REST-owe API)
- FrontEnd: Android (natywna aplikacja)
- Budowanie aplikacji: Gradle
- Repozytorium: Github

<https://github.com/Antah/Projekt-zespolowy/tree/development>

Architektura aplikacji

- Aplikacja serwerowa w Springu podzielona jest na dwa moduły „CosplayModel” (wszystkie możliwe struktury definiujące REST’owe odpowiedzi) oraz „CosplayServer” (Definicja REST-owych serwisów).
- Serwer jest budowany zautomatyzowanym skryptem w Gradl’u. Tworzony jest **CosplayModel.jar**, który **można ponownie wykorzystać w aplikacji na Androidzie**.

Projekt interfejsu w pidoco

- <https://pidoco.com/rabbit/invitation/TeuUKMfAt3nYdIslSaJetSECCwkpPOJwwtT3INlj>

Można poklikać, jest to tylko szkic, więc nie jest bardzo dokładny. Szkic ten będzie omówiony na następnych slajdach.

Uproszczone przypadki użycia oraz
szkic interfejsu.

Logowanie

The image shows a mobile application interface for a login page. At the top, there's a status bar with signal strength, time, and battery icons. Below it, the title 'Login page' is centered. The main heading 'CosplayBOOK' is prominently displayed. Underneath, there are two input fields labeled 'Login' and 'Pass'. A large, dark grey button with the text 'Sign In!' is positioned below the password field. At the bottom of the screen, there are two smaller buttons: 'Sign up!' on the left and 'Forget the password?' on the right.

P1. Rejestracja (pominięte w projekcie interfejsu)

1. Użytkownik wprowadza dane:

- Nick
- Hasło
- Email
- Podpowiedź hasła (jak w Windowsie)

2. Na email wysyłany jest link aktywacyjny konto

3. Użytkownik klika link i od tego momentu może się już logować.

P2. Przypomnienie hasła (pominięte w projekcie interfejsu)

1. Użytkownik podaje email

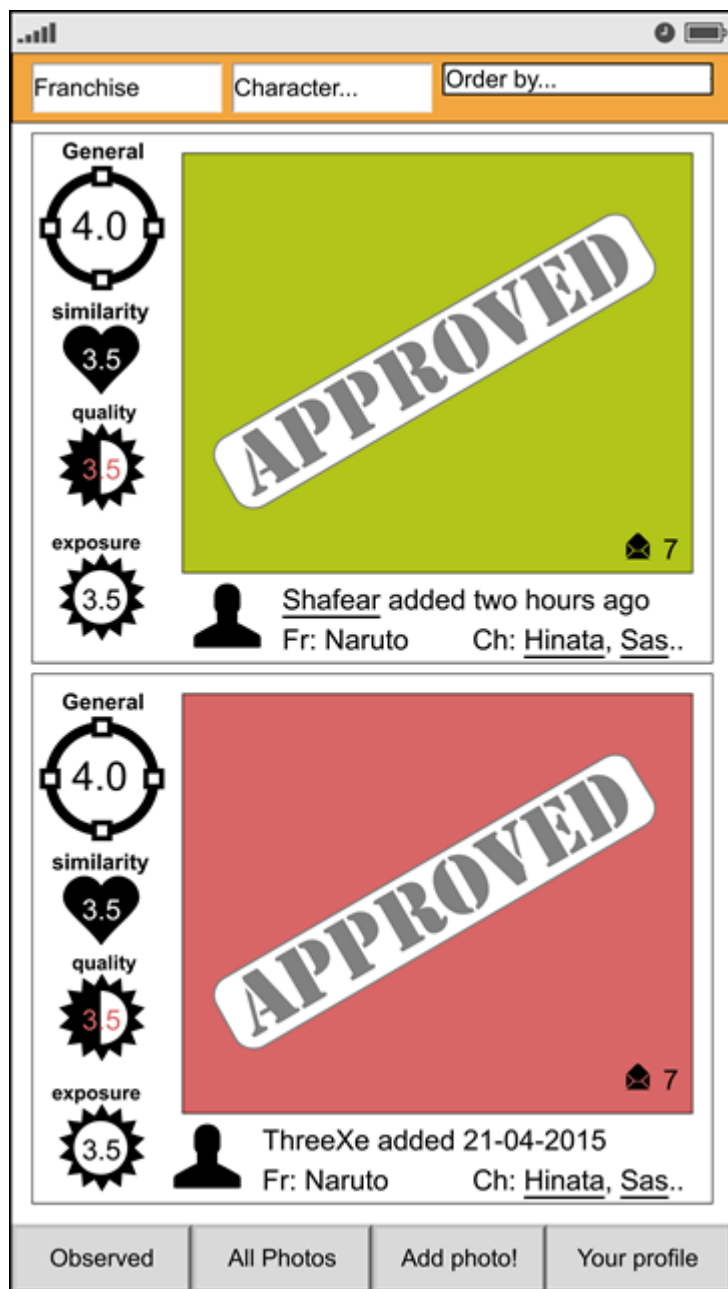
2. System pokazuje podpowiedź hasła

P3. Logowanie (zdjęcie po lewej) // zamiast „Login” będzie „Email”

1. Użytkownik podaje email i hasło

2. U. klika „Sign In!” i loguje się do Menu Głównego

Menu główne (wszystkie zdjęcia)



P4. Przeglądanie zdjęć

To menu będzie dostępne w 3 wariantach

1. All Photos – wszystkie zdjęcia
2. Observed – wszystkie zdjęcia obserwowane przez użytkownika
3. [Someone] Photos – wszystkie zdjęcia danego użytkownika

Jednak będzie to tylko jedno zapytanie GET do serwera z określonymi parametrami.

Wyniki będzie można sortować po:
dacie, ocenie g,s,q lub e, liczbie komentarzy

Wyniki będą zwracane po 2 (aby nie ściągać wszystkich obrazów na raz)

(w czasie gdy użytkownik widzi pierwsze 2 kolejne 2 będą ładowane w tle, gdy się załadują użytkownikowi odblokuje się „swapowanie” w lewo do kolejnych 2 zdjęć)

Możliwe kliknięcia (poza oczywistymi)

- >Menu oceniania po kliknięciu na którąkolwiek ocenę (*chyba, że to zdjęcie zalogowanego użytkownika!*)
- >Menu użytkownika po kliknięciu w awatar
- >Menu główne (wszystkie zdjęcia) przefiltrowane przez klikniętą franczyzę albo postać.

Menu zdjęcia



P5. Przeglądanie zdjęcia

To menu będzie dostępne w 2 wariantach

1. Mine – moje zdjęcie (zalogowanego użytkownika)
2. Others – zdjęcie innego użytkownika

Możliwe kliknięcia (poza oczywistymi)

>[Others] Menu oceniania po kliknięciu na którąkolwiek ocenę

>[Mine] Usunięcie zdjęcia (będzie dodatkowy guzik)

>Menu użytkownika po kliknięciu w awatar

>Menu główne (wszystkie zdjęcia) przefiltrowane przez klikniętą franczyzę albo postać.

Menu zdjęcia - ocenianie

Your vote

General
4.0
★ ★ ★ ★ ★

similarity
3.5
★ ★ ★ ★ ★

quality
3.5
★ ★ ★ ★ ★

exposure
3.5
★ ★ ★ ★ ★

Vote

Comments

ThreeXe commented two hours ago
Looks great, love Hagrid's style!

Input your comment here...

Send

Observed **All Photos** **Add photo!** **Your profile**

P6. Ocenianie zdjęcia

Wybieramy liczbę gwiazdek i oceniamy
Identycznie będzie się zmieniać ocenę

To okienko będzie możliwe do wywołania z menu wszystkich zdjęć
lub z menu jednego zdjęcia lub z menu użytkownika.

Dodanie zdjęcia



The screenshot shows a mobile application interface for adding a photo. At the top, there's a status bar with signal strength, a question mark icon, and a battery level indicator. The main content area features a large square image placeholder with a green background and a white diagonal stamp that reads "APPROVED". Below the image, the section is titled "Photo". It contains a text input field with the value "sdcard/photos/myawesomeimage.jpeg" and a "Choose photo" button. The next section is titled "Description" and contains a text input field with the value "My awesome description...". Below that is a section titled "Franchises" with a text input field containing "Naruto, Harry Potter, Shaman King...". The final section is titled "Characters" with a text input field containing "Sakura, Hagrid, Yo Asakura...". At the bottom of the main content area is an "Add photo!" button. The bottom of the screen features a navigation bar with four buttons: "Observed", "All Photos", "Add photo!", and "Your profile".

Photo

sdcard/photos/myawesomeimage.jpeg Choose photo

Description

My awesome description...

Franchises

Naruto, Harry Potter, Shaman King...

Characters

Sakura, Hagrid, Yo Asakura...

Add photo!

Observed All Photos Add photo! Your profile

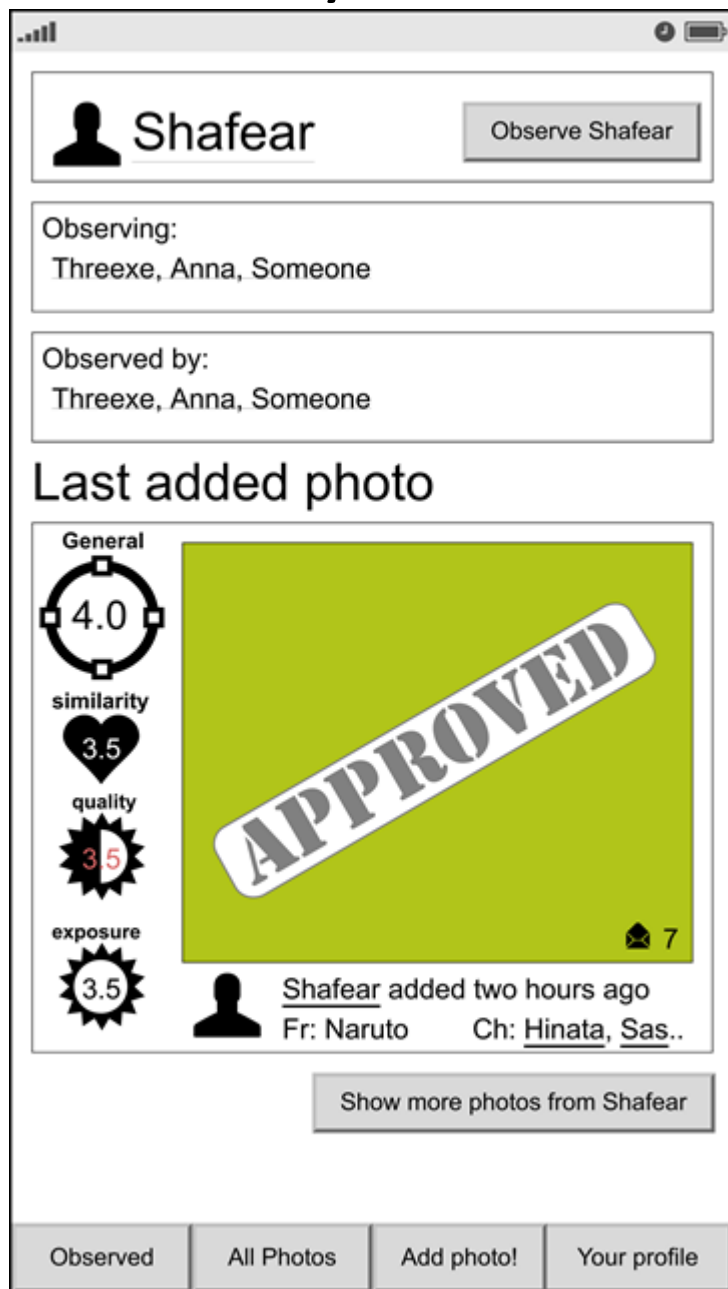
P7. Dodanie zdjęcia

Franczyzy i postać będziemy oddzielać przecinkiem.

Po kliknięciu add photo przejdziemy do menu dodanego właśnie zdjęcia.

! REST będzie musiał zwrócić ID dodanego zdjęcia

Menu użytkownika



P8. Menu użytkownika

Warianty:

1. Mine
2. Others

Możliwości?

1. [Mine] Zmiana awatara (przy rejestracji nie ustawiamy awatara, ta opcja będzie dostępna tylko tutaj) po kliknięciu w awatar.
2. [Others] Kliknięcie „obserwuj”/”odobserwuj”
3. Można kliknąć w użytkownika z Observing albo Observed by i się pojawi jego menu użytkownika.

Last Added Photo?

Zasady identyczne jak w Menu wszystkich zdjęć, tylko tutaj mamy jedno.

Co już zostało zrobione?

- Wizja systemu
- Repozytorium development/production
- Architektura systemu (jeden przykład, przechodzący przez wszystkie warstwy systemu aby każdy mógł się zapoznać z konwencją)
[Czyli kod gotowy do rozszerzania!]
- Skonfigurowany i przygotowany serwer do testów w ramach domeny shafear.space (żeby nie testować tylko na „local-hoscie”)

Co teraz zrobimy?

Warsztaty

1. Spotykamy się razem w ramach warsztatów, gdzie chciałbym aby każdy wcześniej miał już przygotowanego laptopa z potrzebnymi narzędziami i przetestowanym uruchomieniem aktualnego kodu (dostaliście wcześniej instrukcje jak to zrobić <http://pastebin.com/GJiDF36s>) [jeżeli coś się nie uda to na spotkaniu spróbujemy to naprawić].
2. Razem zaprojektujemy i zaimplementujemy całą bazę danych.
3. Razem zaprojektujemy REST API.
4. Podzielimy zadania, o tym niżej.

W zespole mamy 5 osób, praca zostanie podzielona następująco:

1. Osoba 1 – fullstack dla przypadku P1 (rejestracji) i P2 (odzyskiwania hasła) (samodzielne zaprojektowanie i wykonanie 2 funkcjonalności zaczynając od bazy danych, REST-api i kończąc na kliencie Androida). Zadanie wymaga zapoznania się z wszystkimi technologiami, jednak same przypadki są na tyle odseparowane (i proste) aby osoba początkująca sobie spokojnie poradziła.
2. Osoba 2 (Ja – Michał Jereczek) - wszystkie pozostałe przypadki po stronie Androida
3. Osoba 3 + 4 – Implementacja serwisów RESTowych, w których występuje przesyłanie lub odbieranie zdjęć.
4. Osoba 5 – Pozostałe serwisy RESTowe + bycie głównym testerem systemu