

# Koncepcja wykonania systemu PhotoCHAD

Łukasz Klasieński

Marcin Witkowski

26 listopada 2018

## 1. Na poziomie ‘logicznym’ dokładne scenariusze dialogu człowieka z komputerem

### Scenariusz 1.

Przeglądanie posiadanych zdjęć, dostępne zawsze na głównym oknie aplikacji. Funkcjonalność: Galeria zdjęć Opis interfejsu z # 2

### Scenariusz 2.

Przeglądanie posiadanych katalogów podzielonych posiadających zdjęcia przypisywane do poszczególnych kategorii. Funkcjonalność: Katalogi zdjęć Opis interfejsu z # 2

### Scenariusz 3.

Wykrycie oraz weryfikacja duplikatów przez użytkownika Funkcjonalność: Wykrywanie duplikatów Opis interfejsu z # 2

## 2. Zdjęcia ekranów dla ‘scenariuszy’

(Marcin Witkowski)

## 3a. Model konceptualny rzeczywistości (identyfikacja encji i powiązań między nimi)

(to samo co w #4?)

- Photo
  - przechowuje podstawowe informacje o pojedynczym zdjęciu
  - przechowuje jeden lub więcej ‘wskaźników’ na albumy w których może się znajdować (many-to-many)
  - przechowuje jeden lub więcej ‘wskaźników’ na tagi którymi są opisane (many-to-many)
- Album
  - przechowuje informacje o kategorii (ale nie o zdjęciach się w nim znajdujących)
- Tag
  - przechowuje informacje o tagu (ale nie zdjęciach nim opisanych)
  - może, ale nie musi przechowywać dokładnie jeden wskaźnik na twarz (one-to-one)
  - może, ale nie musi przechowywać jeden lub więcej wskaźników na tagi (metatag; many-to-many)
- Face
  - przechowuje informacje o twarzy

- Accounts
  - przechowuje dane logowania do kont w usługach firm trzecich (Facebook, Flickr, etc.)

### **3b. Wymienienie oraz przedstawienie graficzne elementów aplikacji oraz powiązań:**

#### **Sprzęt**

Sprytel, Komp and stuff

#### **Oprogramowanie systemowe, bazy danych, narzędzia programistyczne, oprogramowanie do testowania**

Rust and stuff (Marcin Witkowski)

#### **struktury podziału obiektowego kodu**

Chad++

### **4. Schemat bazy danych (diagram z tabelami/kluczami itp)**

### **5. Przedstawienie zasad kodowania(?)**

Githuby, gałęzie, grafik prac, Rust Style Guide<sup>1</sup>

### **6. Identyfikacja ryzyka i opracowanie zasad zarządzania ryzykiem(?)**

Ryzyko że api serwisów są be więc kupa Ryzyko że nie zdążymy na czas ha!

### **7. Ocena zgodności pracy z wizją z tablicy koncepcyjnej(lab0) i specyfikacją wymagań**

Oczywiście wszystko na 100% 9/11

---

<sup>1</sup><https://github.com/rust-lang-nursery/fmt-rfcs/blob/master/guide/guide.md>