



## ZADÁNÍ BAKALÁ SKÉ PRÁCE

<b>Název:</b>	Colorbis – mobilní aplikace na platform Android pro zvýšení kreativity uživatel
<b>Student:</b>	Oleksandr Balan
<b>Vedoucí:</b>	Ing. Ji í Hunka
<b>Studijní program:</b>	Informatika
<b>Studijní obor:</b>	Softwarové inženýrství
<b>Katedra:</b>	Katedra softwarového inženýrství
<b>Platnost zadání:</b>	Do konce letního semestru 2016/17

### Pokyny pro vypracování

Cílem této bakalá ské práce je vytvo it mobilní aplikaci na platform Android, která umožní vid t ve sv t barvy a rozvíjet kreativitu uživatel .

Navrhnete a implementujete aplikaci, která by analyzovala složení barev ve fotografiích, umožnila jejich vytvá ení, sdílení a filtrování podle barevnosti.

Vytvo ite analýzu mobilních aplikací (minimáln p ti), které umož ůjí práci s fotografiemi a zabývají se jejich sdílením.

Navrhnete vhodnou funkcionalitu aplikace pomocí metod softwarového inženýrství.

Navrhnete vhodné uživatelské rozhraní.

Vytvo ite wireframy aplikace.

Implementujte prototyp.

Otestujte prototyp vhodnou sadou test .

Implementujte finální aplikaci, kterou vystavíte na Google Play.

### Seznam odborné literatury

Dodá vedoucí práce.

L.S.

Ing. Michal Valenta, Ph.D.  
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.  
d kan

V Praze dne 10. listopadu 2015



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Bakalářská práce

## **Colorbis – mobilní aplikace na platformě Android pro zvýšení kreativity uživatelů**

*Oleksandr Balan*

Vedoucí práce: Ing. Jiří Hunka

8. května 2016



---

## Poděkování

Rád bych poděkoval vedoucímu práce Ing. Jiřímu Hunkovi za odborné vedení, poskytnutí praktických rad a znalostí a pomoc při zpracování práce.



---

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Praze dne 8. května 2016

.....

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2016 Oleksandr Balan. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.*

### **Odkaz na tuto práci**

Balan, Oleksandr. *Colorbis – mobilní aplikace na platformě Android pro zvýšení kreativity uživatelů*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.



---

# Abstrakt

Tato bakalářská práce se zabývá vývojem mobilní aplikace pro operační systém Android, která má za cíl rozvíjet kreativitu uživatelů. Aplikace umožní pořízení a sdílení fotografií se zadaným barevným složením, prohlížení fotografií jiných uživatelů a jejich filtrování podle barevnosti. Součástí bakalářské práce je analýza konkurenčních aplikací, návrh, implementace a testování aplikace. Výsledkem je plně funkční aplikace dostupná na Google Play.

**Klíčová slova** mobilní aplikace, fotografování, Android, dominantní barvy, kreativita

---

# Abstract

This bachelor thesis is about development of mobile application for Android operating system, which aims at increasing creativity. The application will allow users to take and share photographs with the specified color composition, view other users' photos and filter them by colors. The bachelor thesis contains an analysis of the competitive applications, design, implementation and testing of the application. The thesis result is a fully functional application available on the Google Play Store.

**Keywords** mobile application, photography, Android, dominant colors, creativity

---

# Obsah

Úvod	1
<b>1 Současná řešení</b>	<b>3</b>
1.1 Aplikace na pořízení, sdílení a ukládání fotografií . . . . .	3
1.2 Shrnutí . . . . .	10
<b>2 Analýza</b>	<b>13</b>
2.1 Dotazník . . . . .	13
2.2 Volba platformy . . . . .	18
2.3 Scénáře použití aplikace . . . . .	19
2.4 Definice požadavků . . . . .	21
2.5 Model případů užití . . . . .	23
2.6 Algoritmy pro nalezení dominantní barvy . . . . .	27
2.7 Algoritmy pro nalezení vzdálenosti mezi barvami . . . . .	30
<b>3 Návrh</b>	<b>33</b>
3.1 Doménový model . . . . .	33
3.2 Diagramy aktivit . . . . .	35
3.3 Návrh serverové části aplikace . . . . .	36
3.4 Databázový model . . . . .	39
3.5 Návrh barev pro úlohy a filtrování . . . . .	43
3.6 Návrh uživatelského rozhraní . . . . .	44
<b>4 Implementace prototypu</b>	<b>53</b>
4.1 Klient . . . . .	53
4.2 Server . . . . .	58
<b>5 Testování prototypu</b>	<b>61</b>
5.1 Testování programátorem . . . . .	61
5.2 Testování použitelnosti . . . . .	62

<b>6 Implementace výsledného řešení</b>	<b>65</b>
6.1 Kamera . . . . .	65
<b>Závěr</b>	<b>67</b>
<b>Literatura</b>	<b>69</b>
<b>A Seznam použitých zkratk</b>	<b>75</b>
<b>B Obrázky a schémata</b>	<b>77</b>
<b>C Formule</b>	<b>85</b>
<b>D Seznam úkolů aplikace</b>	<b>87</b>
<b>E Výsledky testů použitelnosti</b>	<b>89</b>
E.1 Pre-test dotazník . . . . .	89
E.2 Post-test dotazník . . . . .	89
<b>F Ukázka výsledné aplikace</b>	<b>91</b>
<b>G Obsah příloženého CD</b>	<b>95</b>

---

## Seznam obrázků

1.1	Náhled aplikace Instagram . . . . .	4
1.2	Náhled aplikace Flickr . . . . .	6
1.3	Ukázka soutěže z aplikace EyeEm . . . . .	7
1.4	Ukázka obchodu z aplikace Photobucket . . . . .	8
1.5	Ukázka špatného uživatelského rozhraní z aplikace Snapr . . . . .	10
2.1	Výsledky otázky, zda respondent někdy používal aplikaci pro sdílení fotografií . . . . .	16
2.2	Popularita jednotlivých aplikací pro sdílení fotografií . . . . .	17
2.3	Častost používání aplikací respondenty pro sdílení fotografií . . . . .	17
2.4	Rozložení zájmu uživatelů o funkčnosti v navržené aplikaci . . . . .	18
2.5	Diagram případů užití . . . . .	24
3.1	Doménový model . . . . .	33
3.2	Databázový model serveru . . . . .	40
3.3	Databázový model klienta . . . . .	42
3.4	Paleta obsahující 27 barev . . . . .	43
3.5	Výsledná paleta obsahující 15 barev . . . . .	44
3.6	Návrh jednotlivých obrazovek (první verze) . . . . .	45
3.7	Návrh filtrovacího menu (první verze) . . . . .	46
3.8	Druhý návrh jednotlivých obrazovek . . . . .	49
4.1	Porovnání výsledků algoritmů k-means a k-means++ . . . . .	56
B.1	Ukázka běhu algoritmu k-means . . . . .	78
B.2	Diagram aktivit pro filtrování fotografií . . . . .	79
B.3	Diagram aktivit pro splnění úkolu . . . . .	80
B.4	Diagram aktivit pro registraci na serveru . . . . .	81
B.5	Diagram aktivit pro přihlášení do účtu . . . . .	82
B.6	Diagram toku obrazovek . . . . .	83

C.1	Formule pro počítání Delta E94 . . . . .	85
C.2	Formule pro počítání Delta E00 . . . . .	86
F.1	První část ukázky výsledné aplikace . . . . .	92
F.2	Druhá část ukázky výsledné aplikace . . . . .	93

---

## Seznam tabulek

1.1	Porovnání funkcí a vlastností . . . . .	11
2.1	Zastoupení verzí operačního systému Android . . . . .	19
2.2	Pokrytí funkčních požadavků . . . . .	28
2.3	Význam hodnot, kterých může nabývat Delta E . . . . .	30
3.1	Výhody programování serveru v Javě a v node.js . . . . .	37
3.2	Testování první verze uživatelského rozhraní pomocí Nielsenových heuristik . . . . .	47
3.3	Testování druhé verze uživatelského rozhraní pomocí Nielsenových heuristik . . . . .	51
4.1	Vzdálenosti mezi barvami podle formule Delta E 2000 . . . . .	58





---

# Úvod

V dnešní době většina lidí žije ve velkých městech, pracuje v kancelářích a každý den v jejich životě je stejný jako ten předchozí. Lidé se věnují práci a domácnosti, a proto převážně ani nestíhají vidět zajímavé věci, které se kolem nich dějí. Kromě toho na vytváření svého kreativního díla člověk většinou nemá ani čas, ani chuť, a proto ztrácí svůj kreativní potenciál a jako výsledek se může dostat do deprese.

Jedním z možných řešení tohoto problému je mobilní aplikace, která bude pomáhat uživatelům vidět ve světě barvy a dávat možnost ukázat svoji kreativitu jiným. Musí to být aplikace, která nezabírá člověku moc času, ale dává mu možnost k sebevyjádření. Abych lidem „zpestřil“ život, rozhodl jsem se vyvinout aplikaci, která bude denně zadávat úkoly pořídit fotografii předmětu nebo jevu tak, aby výsledná fotografie obsahovala dané složení barev. Uživatel by tak mohl cestou domů například hledat napůl červený a napůl žlutý snímek místo toho, aby myslel na problémy. Výslednou fotografii uživatel může buď uložit do telefonu, nebo sdílet ostatním uživatelům aplikace. Pokud uživatel nedokáže vymyslet, jakou fotografii může udělat, tak má možnost vyfiltrovat fotografie podle barevnosti a inspirovat se tím. Navíc by aplikace měla poskytovat možnost použití speciální kamery, která bude v reálném čase zobrazovat barevné složení aktuálního snímku, čímž je možné mnohonásobně ulehčit splnění úkolu.

Cílem této bakalářské práce je navrhnout a vyvinout aplikaci, která umožní pořízení a sdílení fotografií se zadaným barevným složením a jejich filtrování podle barevnosti. Kromě toho aplikace musí mít přehledné uživatelské rozhraní a být snadně použitelná bez uživatelské příručky.

V rámci bakalářské práce budu zkoumat existující aplikace, které umožňují práci s fotografiemi a zabývají se jejich sdílením. Dalším krokem je zvolení technologií a návrh funkcí aplikace pomocí metod softwarového inženýrství. Poté následuje závěrečná část, která obsahuje vytvoření prototypu klienta a serveru, následné jeho důkladné testování a implementaci výsledného řešení.



---

## Současná řešení

### 1.1 Aplikace na pořízení, sdílení a ukládání fotografií

V době provedení analýzy podobných řešení nebyla nalezena žádná aplikace, která by podněcovala a rozvíjela kreativitu uživatelů pomocí pořízení fotografií obsahujících určité dominantní barvy. Ale existuje velké množství aplikací pro mobilní operační systémy, které se zabývají pořízením a sdílením fotografií s možností filtrování podle klíčových slov nebo barev. Možnosti analyzovaných aplikací, i když nepokrývají množinu požadovaných vlastností, jsou primární v navržené aplikaci. V následujících podkapitolách budu analyzovat nejzajímavější z nich. Všechny analyzované aplikace jsou dostupné pro uživatele mobilních zařízení se systémem Android a iOS.

Hlavními kritérii, podle kterých budu analyzovat aplikace, jsou přehlednost uživatelského rozhraní a snaha rozvíjet kreativitu uživatelů. Tato kritéria byla zvolena tak, aby bylo možné přiblížit analyzované aplikace k myšlence navržené aplikace.

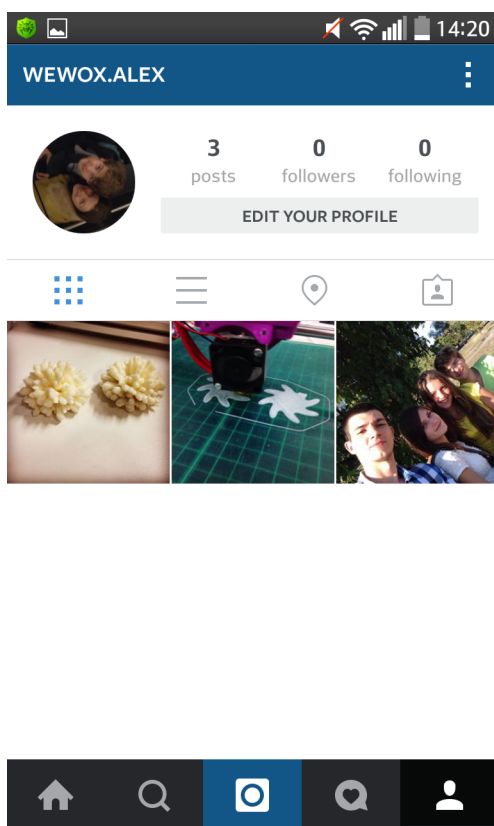
#### 1.1.1 Instagram

Instagram [1] je nejzajímavější a bezpodmínečně nejznámější aplikací určenou pro sdílení fotografií, která je dostupná pro všechny populární mobilní operační systémy, jako jsou Android, iOS a Windows Phone. Instagram je první sociální síť, která byla založena na myšlence sdílení fotografií. Už za dva měsíce po uvedení této aplikace na trh počet registrovaných uživatelů přesáhl hranici jednoho milionu [2]. Právě proto jsem zvolil Instagram jako první aplikaci ke zkoumání.

První, co uvidí uživatel po nainstalování aplikace, je stránka s přihlášením nebo registrací. Bez tohoto kroku není možné prohlížet fotografie jiných uživatelů nebo samozřejmě přidávat svoje. Je to obrovská nevýhoda, protože může se stát, že člověk nechce nebo nemůže nahrávat svoje fotografie na in-

## 1. SOUČASNÁ ŘEŠENÍ

---



Obrázek 1.1: Náhled aplikace Instagram

ternet, a jako důsledek je zbaven možnosti prohlédnout účty svých kamarádů nebo známých osobností. Registrace vyžaduje telefonní číslo nebo elektronickou poštu pro verifikaci uživatele. Jedna z možností, jak se rychle zaregistrovat, je využít účet sociální sítě Facebook<sup>1</sup>. Je to jeden z důvodů, proč je aplikace známá po celém světě a počet uživatelů překonal 400 milionů lidí a každý den je nahráno přibližně 80 milionů nových fotografií [3]. Blízké propojení s velmi známou sociální sítí jako je Facebook, umožňuje snadnou registraci a rychlé nalezení osob, které uživatel bude rád sledovat. Po registraci aplikace nabízí velkou řadu možností, jak využívat svůj účet, a to buď spravovat svoje fotografie anebo sledovat fotografie jiných uživatelů. Ukázkou hlavní stránky účtu uživatele můžete vidět na obrázku 1.1.

Hlavní funkčnost této aplikace je sdílení fotografií, a proto hlavní tlačítko vyzývá uživatele k nahrání svých snímků a přidání popisků k nim. Důležitou výhodou je možnost přidávat nejenom fotografie, ale i videa. Na fotografiích uživatel může označit lidi, kteří jsou na nich zobrazení. Jednou z nevýhod této aplikace je nejasnost, jakou skupinu uživatelů je možné označit na foto-

---

<sup>1</sup>[www.facebook.com](http://www.facebook.com)

grafii. Neexistuje možnost označit libovolného člověka, i když uživatel ví, jaké má jméno nebo přezdívku. Další funkčnost, kterou uživatel potká při přidávání fotografií, je označení místa, kde fotografie byla pořízena. Jiná důležitá možnost, kterou si oblíbily miliony uživatelů, je úprava fotografií a přidávání různých efektů, které zlepšují jejich vzhled.

Prohlížení fotografií jiných uživatelů anebo jejich sledování je druhou hlavní činností, kterou může podniknout uživatel. Vyhledávat lidi je možné pomocí jména, příjmení nebo loginu. Vyhledávání fotografií se provádí podle hashtagu nebo geografického umístění. Díky tomu uživatel může snadno vyhledat fotografii na téma, které ho zajímá. Každou fotografii je možné otevřít a zanechat komentář nebo označit jako „to se mi líbí“. Pokud uživatel je ochoten monitorovat nové fotografie jiného člověka, tak může využít funkci sledování. Ovšem hlavní nevýhodou tohoto je nemožnost zůstat anonymní, protože každý uživatel si může prohlédnout jak seznam těch, které sleduje, tak i seznam těch, kým je sledován.

Velké množství různých možností, i když je možné považovat za výhodu, pro začínajícího uživatele nejsou jasné, a proto ovládání aplikací není tolik intuitivní. Další nevýhoda je to, že cílem aplikace není sdílet fotografie, které jsou uměleckým dílem nebo mají estetickou hodnotu, a proto je dost náročné najít krásnou fotografii přírody, jevu nebo budovy. Cílem aplikace je ukázat život člověka, jak a kde tráví volný čas.

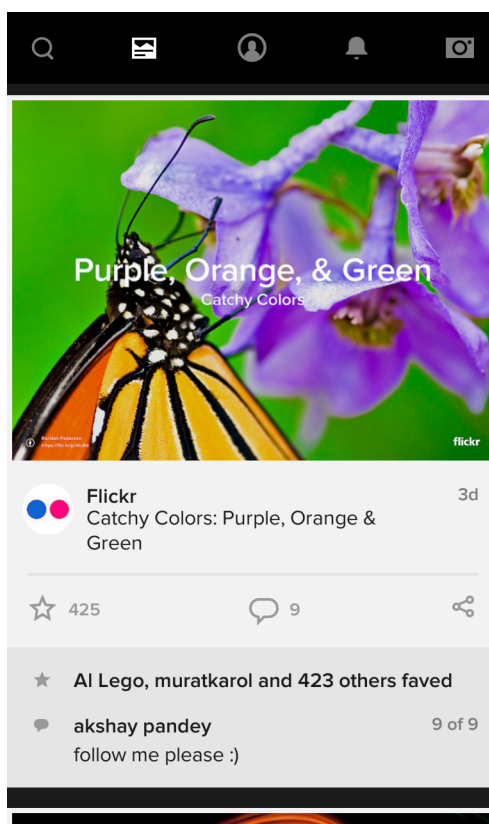
Analýza byla provedena na mobilním zařízení LG Optimus G s verzí operačního systému Android 4.4.2. Verze analyzované aplikace byla 7.16.0.

### 1.1.2 Flickr

Flickr [4] je druhá z nejpoužívanějších aplikací pro ukládání a sdílení fotografií. Hlavní nedostatek této aplikace, stejně jako i u Instagramu, je povinná registrace a přihlášení. I když je logické, že není možné nahrávat fotografie na server bez přihlášení, ale na druhou stranu je to příliš omezující, protože uživatel si ani nemůže prohlédnout nebo vyfiltrovat fotografie, které ho zajímají. Náhled do této aplikace můžete uvidět na obrázku 1.2.

První hláška, kterou uživatel uvidí po registraci, je to, že on má možnost nahrát na server až 1 000 GB fotografií. Na první pohled to vypadá jako obrovská výhoda, avšak při bližším rozboru Instagram neměl žádné viditelné omezení. Fotografie je možné členit do sbírek a nahrávat do různých skupin podle témat. Ještě jednou výhodou je vhodná rychlost odezvy. Uživatel nikdy nemusí čekat, pokud má rychlé spojení s internetem. Další užitečná funkčnost, kterou by bylo vhodné zmínit, je možnost přidávat fotografie do oblíbených položek a zanechávat komentáře. Jediná chybějící funkčnost, co připadá v úvahu pro aplikace tohoto typu, je prohlížení fotografií, které uživatel přidal do oblíbených.

Jednou z překvapujících možností této aplikace je automatické nahrávání fotografií v privátním režimu. Toto pomáhá šetřit místo v telefonu. Pokud



Obrázek 1.2: Náhled aplikace Flickr

uživatel přímo nedovolí ukazovat fotografii ve svém účtu, tak budou skryté.

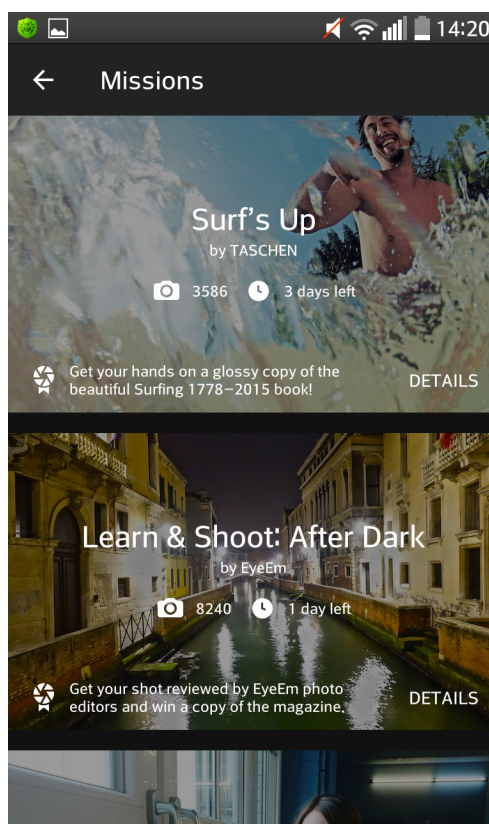
Tato aplikace slouží lidem, kteří nechtějí vidět fotografie s provázáním ke konkrétnímu člověku, ale chtějí uvidět kouzelné fotografie nějakého místa nebo odpovídající nějakému tématu. Flickr zvolilo hodně kvalitních fotografů, kteří chtějí ukázat svoje snímky světu. Aplikace má intuitivní a jednoduché uživatelské rozhraní, kde každá funkčnost je na svém místě.

Analýza byla provedena na mobilním zařízení LG Optimus G s verzí operačního systému Android 4.4.2. Verze analyzované aplikace byla 4.0.7.

### 1.1.3 EyeEm

EyeEm [5] je aplikace, která se věnuje krásným fotografiím. Zvolil jsem ji pro svou analýzu, protože je to jedna z mnoha aplikací, která se snaží rozvíjet kreativitu uživatelů. I když hlavní funkčnost zůstává sdílení a ukládání fotografií, kde každý uživatel má povinnost se zaregistrovat, tvůrci aplikace se zaměřili na různé soutěže. Administrace aplikace často vyhláší zajímavé soutěže o různé ceny, jako jsou knihy nebo časopisy. Témata soutěže mohou být libovolná, například nejlepší fotografie surfaře nebo nejhezčí fotografie ve tmě.

## 1.1. Aplikace na pořízení, sdílení a ukládání fotografií



Obrázek 1.3: Ukázka soutěže z aplikace EyeEm

Ukázku soutěže je možné uvidět na obrázku 1.3.

Další zajímavá možnost je vstoupení na trh, kde každý buď může prodat svou fotografii nebo zakoupit fotografii jiného uživatele. V dnešní době, kde je důležité respektovat autorská práva, je tato funkčnost velmi populární. Vstoupení na trh vyžaduje poplatek.

Ještě jednou funkcí, kterou nelze nezmínit, jsou výukové články pomáhající fotografovi udělat vynikající snímek. Číst tyto články může libovolný uživatel a jsou úplně zdarma. Každý uživatel může napsat svůj vlastní článek a podělit se o zkušenosti, které pomůžou jiným.

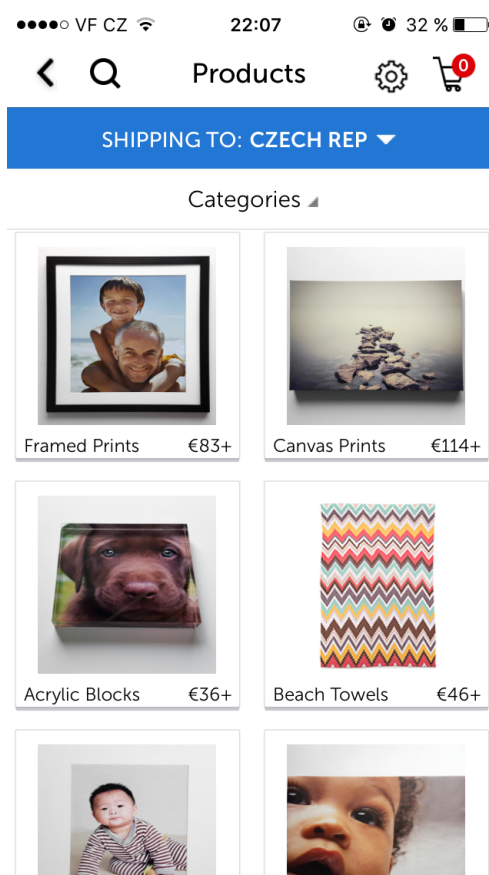
Vyhledávání v aplikaci se provádí podle tágů nebo jména uživatele. Dalšími možnostmi jsou sledování uživatele, zanechání komentáře nebo přidání fotografie do oblíbených položek, a také nabídka fotografií, které mohou být zajímavé pro uživatele.

Hlavní výhodou této aplikace je lakonické uživatelské rozhraní. V aplikaci převládají bílá a černá barva, což je stylově vyvážené a působí to velmi dobře. Aplikace se snaží rozvíjet kreativitu uživatelů pomocí soutěží, což vysoko oceňují kolem 10 milionů uživatelů.

Analýza byla provedena na mobilním zařízení LG Optimus G s verzí ope-

## 1. SOUČASNÁ ŘEŠENÍ

---



Obrázek 1.4: Ukázka obchodu z aplikace Photobucket

račního systému Android 4.4.2. Verze analyzované aplikace byla 5.8.1.

### 1.1.4 Photobucket

Photobucket [6] je aplikace zaměřena ne tolik na ukládání a sdílení fotografií, jak na komerční účely. Aplikace nabízí možnost objednání výtisku dárku s využitím uložených fotografií. Toto znázorňuje obrázek 1.4. Na rozdíl od ostatních aplikací Photobucket neobsahuje uživatelské účty a není možné se dozvědět, kdo nahrál fotografii na server. Tuto aplikaci jsem zvolil pro analýzu, protože se několikrát vyskytla v odpovědích v dotazníku, který je možné si prohlédnout v sekci 2.1. Neposledním důvodem byl zcela odlišný koncept této aplikace.

Photobucket obsahuje mnoho užitečných funkcí od možnosti sdílení fotografií přes známé sociální sítě, jako je Facebook, vyhledávání obrázků podle klíčových slov, také aplikace nabízí nahrávání a prohlížení nejen fotografií, ale i animovaných obrázků. Nejzajímavější na této aplikaci je nástroj na upravování fotografií, který umožňuje přidat různé efekty nebo rámečky. Po spuštění této události aplikace nabídne stažení balíčku obsahujícího obrázky a další do-



plňující věci, které uživatel může přidat ke své fotografii. Jednou z největších výhod, kterou neměla žádná z analyzovaných aplikací, je prohlížení a vyhledávání fotografií bez registrace.

V porovnání s jinými aplikacemi má Photobucket hodně nedostatků, které vadí normálnímu používání. Aplikace je pomalejší, má omezenou funkčnost a nepřehledné uživatelské rozhraní, například není jasné, jak zanechat komentář nebo přidat novou fotografii. Při obvyklých akcích jako je zobrazení fotografie, se občas může ukázat programátorská hláška, která není určena pro uživatele. Ještě jednou nevýhodou Photobucketu je reklama v dolní části obrazovky a oznámení o slevách, které uživatel může získat, pokud si objedná výtisk fotografie.

Analýza byla provedena na mobilním zařízení iPhone 5 s verzí operačního systému iOS 8.3. Verze analyzované aplikace byla 3.3.5.

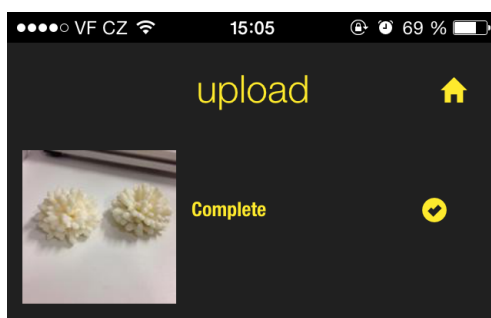
### 1.1.5 Snapr

Snapr [7] je mobilní aplikace, která ještě možná nenašla svého uživatele. Tuto aplikaci jsem zvolil pro analýzu kvůli malému počtu uživatelů, kteří stáhli tuto aplikaci pro operační systém Android, abych se dozvěděl, proč tomu tak je. Dalším důvodem bylo to, že se tato aplikace jednou vyskytla v dotazníku, který je možné si prohlédnout v sekci 2.1.

Na začátku jsem nainstaloval tuto aplikaci jenom na mobilní zařízení s operačním systémem iOS. Při prvním spuštění Snapru se uživatel musí zaregistrovat, udělat to může buď pomocí Facebooku, nebo pomocí Twitteru. Možnost registrace pomocí elektronické pošty je schovaná díky špatnému designu a běžný uživatel ji ani nenajde. Tlačítko, které založí nový účet, bylo napsané bílým písmem na velmi jasném žlutém pozadí. Celá aplikace má řadu nevýhod, které spojené s neúspěšnou volbou barvy, požadované funkčnosti je těžko najít a použít. Tento problém je vidět na obrázku 1.5, kde po nahrávání fotografie není možné pokračovat nebo vrátit se zpět. Občas aplikace obsahuje menu, kde je jen jedna položka, což je zbytečné.

Protože aplikace na mě neudělala žádný dojem, vyřešil jsem ji nainstalovat na zařízení s operačním systémem Android. První překvapení čekalo při spuštění, uviděl jsem úplně odlišný design. Na rozdíl od verze pro operační systém iOS, uživatelské rozhraní pro Android je přehledné, uživatel může snadně využít funkčnost typickou pro tento druh aplikací, jako jsou sledování uživatele a přidávání komentářů. Hlavní nedostatek je, že pro různé operační systémy aplikace nevypadá stejně. Pokud uživatel bude používat Snapr pro Android a pak přejde na operační systém iOS, tak nesmí cítit nepohodlí.

Ještě jedna velká nevýhoda, kterou uživatel potká, pokud používá tuto aplikaci na mobilním zařízení s operačním systémem Android, je to, že nefunguje nahrávání fotografií na svůj účet, tudíž není možné je sdílet. Jelikož je to velmi podstatná funkčnost, aplikace se stává skoro nepoužitelná. O tom



Latest

no photos :(



Obrázek 1.5: Ukázka špatného uživatelského rozhraní z aplikace Snapr

psalo mnoho uživatelů v komentářích na stránce tohoto produktu na Google Play [8], ale tvůrci této aplikace na to nezareagovali.

Obě verze aplikace jak pro Android, tak i pro iOS mají svoje nedostatky, které přesahují výhody, a to jsou nepřehledné uživatelské rozhraní se špatně zvolenými barvami, velké množství chyb a absence příležitostí pro rozvíjení kreativity uživatele. O tom svědčí i malá skupina uživatelů a nepatrný počet nahraných fotografií.

Analýza byla provedena na mobilních zařízeních LG Optimus G s verzí operačního systému Android 4.4.2 a iPhone 5 s verzí operačního systému iOS 8.3. Verze analyzované aplikace pro Android byla 2.9 a pro iOS 4.0.

## 1.2 Shrnutí

V tabulce 1.1 můžete vidět porovnání vybraných funkcí a vlastností jednotlivých analyzovaných aplikací.

V průběhu analýzy podobných řešení jsem se snažil zjistit, jaké možnosti

Tabulka 1.1: Porovnání funkcí a vlastností

Funkce a vlastnosti	Instagram	Flickr	Photobucket	EyeEm	Snapr
Přehledné GUI	Ano	Ano	Ne	Ano	Ne
Úprava fotografií	Ano	Ano	Ano (po nainstalování balíčku)	Ano	Ne
Doporučené novinky	Ano	Ne	Ne	Ano	Ne
Nutné přihlášení pro prohlížení fotografií	Ano	Ano	Ne	Ano	Ne
Přihlášení pomocí Facebooku	Ano	Ano	Ano	Ano	Ano
Vlastní kamera (jen pro Android)	Ano	Ano	Ne	Ne	Ne
Rozvíjení kreativity uživatelů	Ne	Částečně	Ne	Ano (soutěže, výukové články)	Ne
Rozdělení fotografií do skupin	Ne	Ano	Ano	Ano	Ne
Reklama	Ne	Ne	Ano	Ne	Ne
Nakupování obsahu	Ne	Ne	Ano	Ano	Ne
Úprava informace o fotografii	Ano	Ne	Ne (fotografie neobsahuje název)	Ano	Ne

jsou zajímavé pro uživatele, jak tvůrci těchto aplikací se snaží rozvíjet kreativitu uživatelů a jaké uživatelské rozhraní pro tento typ aplikaci je přehledné a snadno použitelné. Z analýzy jsem se dozvěděl, že existují dva druhy aplikací pro sdílení fotografií: ty, které se týkají života uživatelů (jako jsou Instagram a Snapr) a aplikace, které se snaží ukázat fotografie jako umělecká díla (Flickr a EyeEm). Protože mým cílem je vyvinout aplikaci, která by rozvíjela kreativitu uživatelů a pomáhala vidět barvy ve světě, tak budu hlavně zohledňovat nedostatky nalezené ve druhé skupině.

Hlavním nedostatkem těchto dvou aplikací je nutnost přihlášení pro prohlížení fotografií jiných uživatelů, čemu se vyhnu v návrhu funkcí své aplikace. Další důležité pozorování je, že aplikace mají mnoho možností, které nikdo nevyužívá. Ve většině případů mezi těmito možnostmi bylo zanechání komentářů, přidávání efektů a rámečků, což považuji za úplně zbytečné. Závěrem je to, že návrhu uživatelského rozhraní je potřeba věnovat co nejvíce času, protože ve většině aplikací to byl nejdůležitější problém.

---

# Analýza

## 2.1 Dotazník

V dnešní době chytrý telefon s vlastní kamerou je neoddělitelná část života většiny lidí. Žádný člověk už nevystoupí z domu, pokud nebude mít s sebou toto zařízení, protože vždy potřebuje zavolat nebo vyhledat něco na internetu. Toto využívají programátory, kteří se snaží usnadnit život lidem a vymýšlejí nové a nové aplikace. Jelikož potřebuji vytvořit takovou aplikaci, aby vyhovovala lidem a plnila jejich požadavky, tak jsem se rozhodl udělat dotazník, který by mi s tímto cílem pomohl.

Otázky ankety jsou zaměřené na to, abych se dozvěděl, kolik lidí v dnešní době používá aplikace pro sdílení fotografií a jak často se tomu věnují. Hlavním cílem dotazníku bylo zjistit, jestli se uživatelé budou zajímat o šíření kreativních a barevných fotografií nejenom mezi kamarády, ale mezi všemi uživateli.

Dotazník jsem vytvořil pomocí Google Forms<sup>2</sup> v angličtině. Anglický jazyk byl zvolen hlavně kvůli tomu, abych získal odpovědi nejen tuzemských milovníků fotografování, ale i zahraničních. Anketa byla rozeslána prostřednictvím sociálních sítí a fór, které se věnují fotografování.

### 2.1.1 Struktura dotazníku

Celkem anketa obsahuje devět otázek, pět z nich není povinných. Existují dva možné průchody dotazníkem. První následuje po kladném zodpovězení otázky, zda respondent někdy používal aplikaci pro sdílení fotografií a obsahuje dotazy zaměřené na zjištění, jakou aplikaci používal a jak často. Druhý průchod je kratší a následuje po záporné odpovědi na stejnou otázku. Dotazy číslo 3 a 6 mají více možností odpovědí.

---

<sup>2</sup>[www.google.com/intl/cs\\_cz/forms/about/](http://www.google.com/intl/cs_cz/forms/about/)

## 2. ANALÝZA

---

1. Používal jste někdy aplikaci pro sdílení fotografií (například Instagram)?

- Ano
- Ne

Poznámka: pokud vaše odpověď je záporná, pokračujte, prosím, otázkou číslo 7.

2. Jak dlouho používáte (případně jste používal nebo používala) tuto aplikaci?

- Den
- Týden
- Měsíc
- Několik měsíců
- Více než rok

3. Jaký má název tato aplikace?

- Instagram
- Flickr
- EyeEm
- Jiná odpověď

4. Používáte tuto aplikaci doposud?

- Ano
- Ne

5. Jak často používáte tuto aplikaci?

- Několikrát denně
- Jednou denně
- Jednou za několik dní
- Jednou týdně
- Jednou měsíčně
- Skoro nikdy

6. Co se vám líbí na tomto druhu aplikace nejvíce?
- Sdílení fotografií s kamarády
  - Sdílení kreativních fotografií se všemi uživateli
  - Prohlížení fotografií ostatních uživatelů
  - Zanechání komentářů a označení fotografií jako oblíbené
  - Hledání inspirace
  - Sledování někoho
  - Jiná odpověď
7. Líbí se vám fotografování?
- Ano
  - Ne
8. Chcete sdílet svoje kreativní a barevné fotografie s celým světem?
- Ano
  - Ne
9. Zdá se vám, že svět je šedý a ostatních barev je málo (šedá kancelář, šedá ulice, šedá škola)?
- Ano
  - Ne

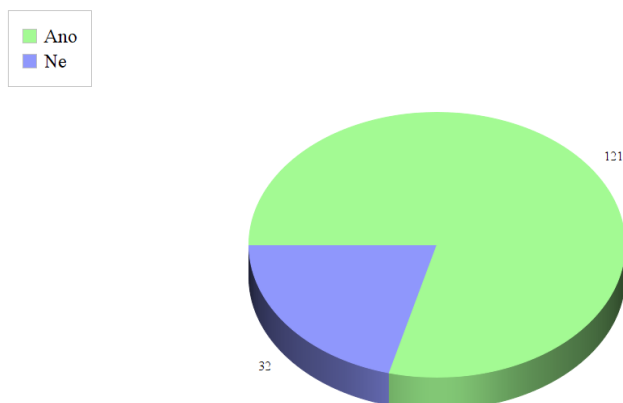
### 2.1.2 Vyhodnocení výsledků

Celkem jsem dostal odpovědi od 153 respondentů, kteří byli požádáni o vyplnění ankety prostřednictvím fór a sociálních sítí. Ve výsledku této ankety celkem 121 respondentů odpovědělo, že někdy používali aplikaci pro pořízení a sdílení fotografií. I když jsem očekával, že hodně lidí se někdy potkalo s aplikacemi tohoto druhu, nemyslel jsem, že se tento počet bude rovnat přibližně 79 %. Výsledky této otázky jsou znázorněny na obrázku 2.1. Z 121 respondentů, kteří se někdy setkali s aplikací tohoto druhu, 93 odpovědělo, že používají tuto aplikaci doposud. Z toho plyne, že pokud člověk začal používat aplikaci pro pořízení a sdílení fotografií, tak už to s největší pravděpodobností nezahodí.

Dalším nepřekvapujícím faktem bylo to, že většina respondentů používala nebo dosud používá aplikaci Instagram. Méně populární byly nabízeny v odpovědích aplikace Flickr a EyeEm. Jen několik respondentů si vzpomenu na jiné aplikace, které mají název Snapr a Photobucket. Na obrázku 2.2 je

## 2. ANALÝZA

---



Obrázek 2.1: Výsledky otázky, zda respondent někdy používal aplikaci pro sdílení fotografií

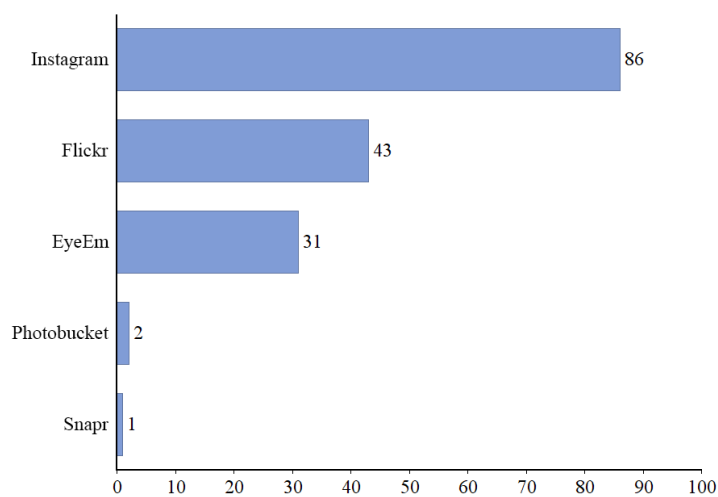
možné vidět počet respondentů a název odpovídající použité aplikaci.

Ještě jedna důležitá otázka spočívala v tom, jak často respondenti používají tuto aplikaci. Cílem této otázky bylo se dozvědět, zda sdílení a prohlížení fotografií je výjimečná činnost nebo jsou lidé na to zvyklí. Výsledky této otázky můžete vidět na obrázku 2.3. Je důležité si poznamenat, že většina respondentů používá aplikaci aspoň jednou za několik dní.

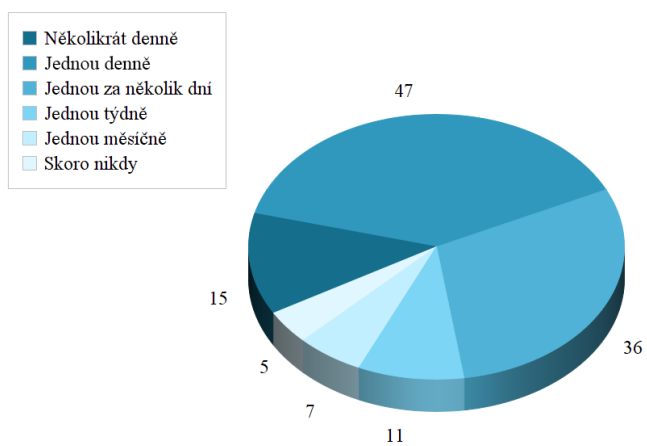
Hlavní otázka, výsledky které mne zajímaly nejvíce, byla o tom, jaké možnosti respondenti oceňují na aplikacích pro sdílení fotografií. Z odpovědi plyne, že se většině uživatelů líbí sdílení kreativních fotografií s celým světem a hledání inspirace ve fotografiích jiných lidí. Překvapující bylo to, že dost málo respondentů oceňuje možnost sdílení snímků jen mezi svými kamarády. Stejně tak málo lidí odpovědělo, že se jim na těchto aplikacích líbí zanechání komentářů. Hlavním pozorováním pro mě bylo, že respondenti nemají zájem o ještě jednu sociální síť, kde budou jen fotografie spojené s životem konkrétního člověka. Výsledky této otázky jsou znázorněné na obrázku 2.4.

Poslední otázka byla obecná a zaměřovala se na osobní pocit respondenta, jestli si někdy nemyslel, že svět, ve kterém žije, je příliš šedý. Převážná většina respondentů odpověděla kladně na tuto otázku. To znamená, že lidé potřebují nějaký důvod k tomu, aby uviděli, že kolem nich jsou barevné věci a život není šedý. Tento výsledek hlavně znamenal pro mne, že nedělám zbytečnou věc a může to pomoci lidem.





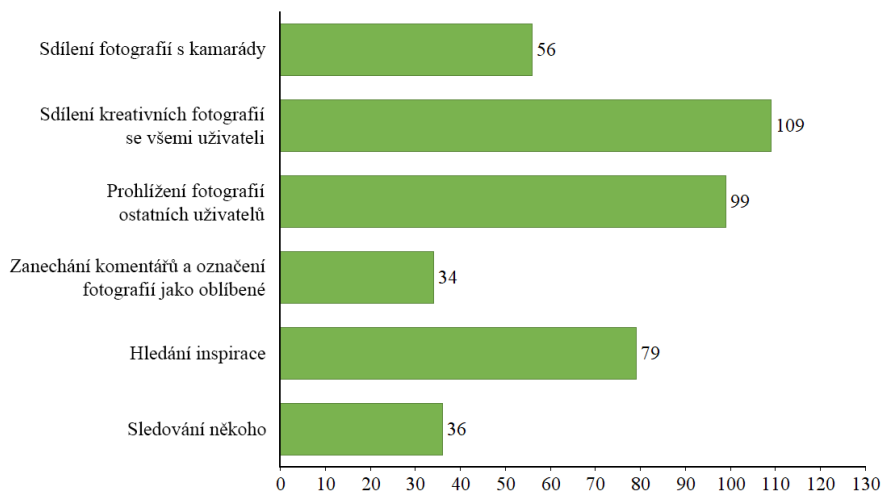
Obrázek 2.2: Popularita jednotlivých aplikací pro sdílení fotografií



Obrázek 2.3: Častost používání aplikací respondenty pro sdílení fotografií

## 2. ANALÝZA

---



Obrázek 2.4: Rozložení zájmu uživatelů o funkčnosti v navržené aplikaci

Přehled všech odpovědí naleznete v příloženém souboru `VysledkyAnkety.csv`.

## 2.2 Volba platformy

V dnešní době většina lidí vlastní mobilní zařízení se zabudovanou kamerou. Jelikož myšlenka aplikace spočívá v nalezení barev kolem sebe a sdílení kreativních fotografií, výběr mobilních zařízení jako cílových zařízení je dost logický. Dalším důležitým krokem bylo zvolit cílovou platformu.

Existují dvě hlavní platformy, pro které v dnešní době má smysl vyvíjet mobilní aplikace: Android a iOS. Podle [9] jednou z výhod vyvíjení pro operační systém iOS je to, že zařízení od americké firmy Apple Inc. mají menší počet možných rozlišení, a proto je snadnější je všechny podporovat. Ještě jednou důležitou výhodou operačního systému iOS je to, že Apple Inc. vždy bezproblémově a rychle aktualizuje operační systém na všech svých zařízeních, a proto skoro nedochází k nekompatibilitě. Na druhou stranu zařízení s operačním systémem Android jsou populárnější, používá je více lidí než zařízení s operačním systémem iOS. Podle [10] podíl Androidu na trhu je 78 % a počet zařízení s tímto operačním systémem přesahuje 1,6 bilionu. Ještě jednou velkou výhodou je to, že testování na Android zařízeních nevyžaduje programátorskou licenci. Obě platformy mají vhodné vývojové prostředí, jako je Xcode pro operační systém iOS a Android Studio pro operační systém Android.

Podle provedené analýzy jsem se rozhodl vyvíjet pro platformu Android.

Tabulka 2.1: Zastoupení verzí operačního systému Android, převzato z [11]

Verze	Označení	API	Podíl
2.2	Froyo	8	0,1 %
2.3.3 - 2.3.7	Gingerbread	10	2,7 %
4.0.3 - 4.0.4	Ice Cream Sandwich	15	2,5 %
4.1.x	Jelly Bean	16	8,8 %
4.2.x		17	11,7 %
4.3		18	3,4 %
4.4	KitKat	19	35,5 %
5.0	Lollipop	21	17,0 %
5.1		22	17,1 %
6.0	Marshmallow	23	1,2 %

Hlavní důraz v navržené aplikaci je kladen na sdílení a prohlížení fotografií, a proto budu potřebovat co nejvíce uživatelů, k čemuž přispívá rozšířenost tohoto operačního systému. Dalším důvodem zvolit platformu Android bylo to, že vlastním dvě zařízení s tímto operačním systémem, což bude určitě vhodné pro testování.

Posledním problémem, který se týká platformy, je volba minimální podporované verze operačního systému Android. Každá verze má svoje vlastní API a částečně se liší od předchozí. Jelikož většinou neexistuje žádná centrální aktualizace pro zařízení s operačním systémem Android, je potřeba podporovat více verzí tohoto systému. V tabulce 2.1 je možné vidět rozdělení jednotlivých verzí na trhu.

Podle [12] je dobrým zvykem podporovat aspoň 90 % aktivních zařízení. Jelikož zastoupení verzí 2.x je dost nízké, rozhodl jsem se tyto starší verze nepodporovat. Dalším důvodem k takovému rozhodnutí bylo to, že aplikace často musí používat kameru, která by v tak starých zařízeních nemusela být kvalitní.

Nepodporovat ostatní verze není žádný důvod, i když zastoupení verzí 4.0.3 – 4.0.4 je také dost nízké. Pro většinu moderních prvků je zpřístupněná zpětná kompatibilita, a proto podpora těch verzí by neměla dělat žádné potíže.

Jako výsledek bude navržená aplikace dostupná pro zařízení s operačním systémem Android verze 4.0.3 a výše.

## 2.3 Scénáře použití aplikace

V následujícím odstavci čerpám z [13].

Základní myšlenka scénářů použití aplikace se skrývá v popisu činnosti uživatele, který se snaží dosáhnout nějakého cíle s využitím systému nebo aplikace. Účelem těchto scénářů je zachytit podstatu interakce uživatele se systémem bez použití vědomostí, jak to funguje uvnitř. Scénář nesmí záležet

na platformě, uživatelském rozhraní nebo implementaci. Scénáře se využívají pro lepší pochopení systému a diskutování o jeho nevýhodách. Rovněž můžou pomoci naplnit stanovené cíle a přispět k dosažení požadovaného výsledku.

### 2.3.1 První scénář

Alice bydlí v obyčejném městě a pracuje v kanceláři. Každý den jede do práce a myslí si, že ve světě není žádná jiná barva než šedá. Alice vždy chtěla fotografovat, i když má jen mobilní zařízení s kamerou a operačním systémem Android. Také se jí líbí prohlížet fotografie jiných uživatelů a hledat inspiraci v nich.

Problém je, že Alice nemá ráda sdělovat fotografie na sociálních sítích a není registrována v žádné z nich. Také problémem je, že Alice neví, co fotografovat v tak šedém městě. Ale jelikož chce fotografovat a vytvářet kreativní fotografie, vyřešila použít novou aplikaci pro zvýšení kreativity uživatelů.

1. Alice zadáním jména nebo přezdívky založí lokální účet.
2. Alice si zvolí barevný úkol, který chce splnit.
3. Alice nebude vědět, jak splnit úkol a jakou fotografii pořídit, tak vyhledá inspiraci mezi snímky ostatních uživatelů.
4. Uzavře aplikaci do toho času, až najde kolem sebe něco, co bude obsahovat požadované barvy. Alice pomocí kamery zanalyzuje barvy.
5. Alice udělá fotografii a splní úkol.
6. Alice výslednou fotografii najde v galerii a může ji buď ukázat kamarádům nebo použít jako pozadí.

### 2.3.2 Druhý scénář

Jan pracuje jako fotograf a rád sděluje fotografie ostatním. Líbí se mu být registrovaným uživatelem různých aplikací, kde může předvést svoje umělecká díla. Také je velmi sociální člověk, a proto rád přidává fotografie do oblíbených a prohlíží fotografie jiných uživatelů.

Hlavní problém Jana je, že potřebuje zvětšit svoje portfolio, a proto chce pořídit více kreativních fotografií.

Jan slyšel o nové aplikaci pro zvýšení kreativity uživatelů a rozhodl se, že tato aplikace plně vyhovuje a splňuje jeho potřeby.

1. Jan se zaregistruje, aby mohl sdílet fotografie.
2. Jan si nahraje svoje kreativní fotografie na server.

3. Jan bude každý den prohlížet seznam posledních nahraných fotografií a hledat v nich inspiraci.
4. Jan vyfiltruje fotografie podle barev, které se mu líbí a vyřeší, co chce fotografovat.
5. Nejlepší fotografie bude Jan přidávat do oblíbených.
6. Jelikož má Jan rád výzvy, tak bude plnit každou úlohu a nahrávat výslednou fotografii na server, čímž se zvětší portfolio.

## 2.4 Definice požadavků

„Analýza požadavků je nezbytnou součástí procesu vývoje, realizace a následné implementace většiny softwarových řešení.“ [14]. Na základě předchozí analýzy již mohu definovat požadavky kladené na systém.

### 2.4.1 Funkční požadavky

Hlavním účelem funkčních požadavků je vymezení funkčnosti systému a přesný popis toho, co by měl systém poskytnout uživatelům. Mezi ně patří popis požadovaných funkčností a vymezení dat, které se budou udržovat v systému [15].

- **FR1 Evidence barevných úkolů**

V aplikaci budou evidované denně zadávané úkoly pořídit fotografii, která obsahuje jednu až tři dominantní barvy. Úkol bude možné splnit pořízením snímku s pomocí zabudované kamery nebo nahrát fotografii z galerie.

- **FR2 Práce s kamerou**

Aplikace umožní použití speciální kamery, která bude v reálném čase analyzovat dominantní barvy na snímku.

- **FR3 Analýza barev z fotografie**

Aplikace bude analyzovat fotografii a určovat jednu až pět dominantních barev. Také aplikace bude umět určovat vzdálenost mezi dvěma barvami pro zjištění, jestli fotografie obsahuje zadané barvy podle úkolu.

- **FR4 Správa lokálních uživatelů**

Aplikace umožní založení lokálních účtů. Toto umožní dvou a více lidem používat aplikaci bez registrace na serveru. Z lokálního účtu se bude možné odhlásit nebo přihlásit. K tomuto účtu nebude požadováno heslo. Zároveň v každém lokálním účtu se bude evidovat počet získaných bodů za splnění úkolů.

- **FR5 Správa registrovaných uživatelů**

Aplikace umožní registraci uživatele na serveru pro nahrávání vlastních fotografií. Účet registrovaného uživatele se bude vytvářet na základě lokálního účtu uživatele.

- **FR6 Výměna fotografií se serverem**

Aplikace umožní nahrávání fotografií na server a mazání již nahraných fotografií ze svého účtu.

- **FR7 Interakce s fotografií**

Aplikace umožní označit fotografii jako oblíbenou.

- **FR8 Evidence fotografií**

Aplikace bude evidovat fotografie, jejich složení barev, vlastníka, počet přidání do oblíbených a datum nahrání na server. Také aplikace musí umožňovat zobrazení seznamu naposled přidanych fotografií.

- **FR9 Filtrování fotografií podle barevnosti**

Aplikace musí umožňovat filtrování všech fotografií podle jedné až tří předdefinovaných barev.

### 2.4.2 Nefunkční požadavky

Nefunkční požadavky určují omezení, cíle a prostředky pro kontrolování systému. Mezi nefunkční požadavky patří technická a jiná omezení, bezpečnost, výkonnost a přístupové požadavky [15].

- **NFR1 Funkčnost na systému Android**

Aplikace musí běžet na operačním systému Android od verze 4.0.3 a výše.

- **NFR2 Stabilita**

Aplikace musí být stabilní vůči nestandardním situacím.

- **NFR3 Internetové připojení**

Funkční požadavky FR1-FR4 nesmí vyžadovat internetové připojení. Pro využití ostatních funkcí aplikace je nutné mít internetové připojení. Aplikace musí správně reagovat na výpadky internetového spojení.

- **NFR4 Bezpečnost přístupu k účtu**

Přístup k uživatelskému účtu na serveru bude chráněn heslem. Přenos hesla se bude provádět v šifrované formě.

- **NFR5 Autorizace a autentizace**

Aplikace musí zajišťovat proces autentizace a autorizace.

- **NFR6 RESTful API**

Komunikace mezi klientem a serverem se bude provádět pomocí RESTful API.

## 2.5 Model případů užití

V modelování systému je nejdůležitější zachytit jeho dynamické chování. Pro tyto účely se používá model případů užití, který se skládá z popisu účastníků, případů užití a jejich vztahů. Diagram případů užití slouží pro modelování chování aplikace a vysvětlení funkčních požadavků. Jedna buňka v diagramu případů užití zachycuje jednu funkčnost aplikace [16]. Diagram případu užití je zobrazený na obrázku 2.5.

### 2.5.1 Účastníci

Tato kapitola obsahuje popis rolí lidí, kteří budou aplikace používat. Každá další role rozšiřuje možnosti předchozí.

#### 2.5.1.1 Uživatel

Člověk se stane uživatelem v okamžiku, kdy stáhne aplikaci. Obyčejný uživatel bude moci si jen založit lokální účet.

#### 2.5.1.2 Uživatel s lokálním účtem

Obyčejný uživatel se stane uživatelem s lokálním účtem v okamžiku, kdy si založí lokální účet. Uživatelé s touto rolí mohou používat hlavní funkčnost aplikace, což je rozvíjení kreativity pomocí pořízení fotografií s určitými dominantními barvami. S internetovým připojením tento druh uživatele může prohlížet a filtrovat fotografie podle barevnosti. Další možností je registrace na serveru.

#### 2.5.1.3 Registrovaný uživatel

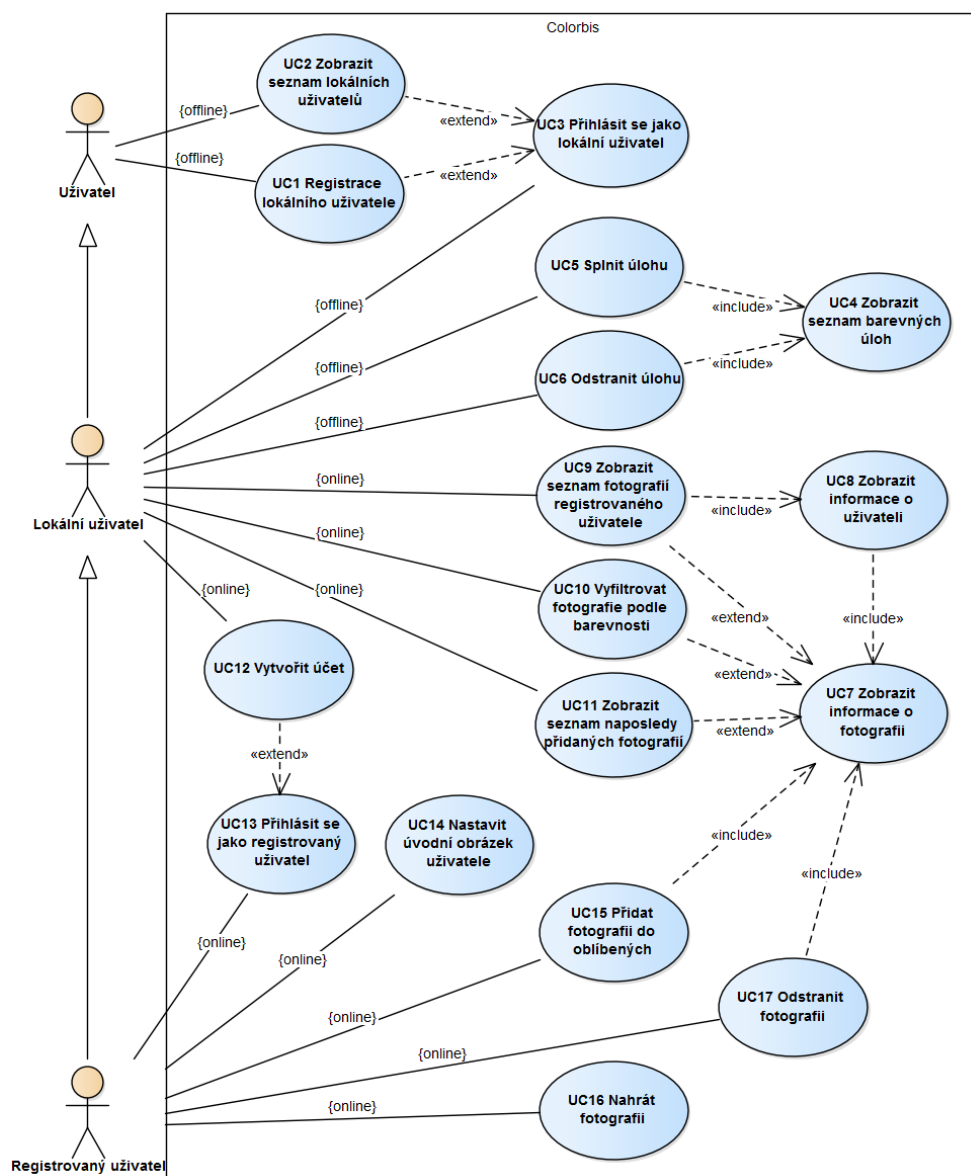
Lokální uživatel se stane registrovaným uživatelem v okamžiku, kdy se zaregistruje na serveru pomocí svého lokálního účtu. Registrovaný uživatel bude moci nahrávat fotografie na server. Další možností je přidání fotografií do oblíbených.

### 2.5.2 Případy užití

- **UC1 Registrace lokálního uživatele**

Aplikace umožní přidávání lokálního účtu uživatele.

## 2. ANALÝZA



Obrázek 2.5: Diagram případů užití



Základní cesta:

1. Příklad užití se začíná po stažení a instalaci aplikace. Uživatel chce používat aplikaci pro zvýšení své kreativity a prohlížení barevných fotografií.
2. Aplikace nabídne vyplnit jméno pro použití aplikace. Jelikož to není přihlášení pro použití serveru, aplikace se nebude ptát na elektronickou adresu a heslo.
3. Uživatel vyplní potřebné informace.
4. Aplikace zkontroluje, jestli neexistuje uživatel se stejným jménem. Pokud kontrola proběhne v pořádku, lokální účet je vytvořen. Pokud uživatel se stejným jménem již existuje, aplikace na to upozorní uživatele.

- **UC2 Zobrazit seznam lokálních uživatelů**

Aplikace umožní zobrazit seznam všech lokálních účtů na mobilním zařízení.

- **UC3 Přihlásit se jako lokální uživatel**

Aplikace umožní přihlášení uživatele do svého lokálního účtu.

- **UC4 Zobrazit seznam barevných úloh**

Aplikace bude denně generovat a zobrazovat úlohy pro zvýšení kreativity s určitými dominantními barvami.

- **UC5 Splnit úlohu**

Aplikace umožní splnit úlohu ze seznamu dostupných.

Základní cesta:

1. Příklad užití se začíná, kdy uživatel chce pořídit fotografii, která splňuje jeden z dostupných úkolů.
2. Aplikace nabídne způsob splnění úkolu, je možné buď nahrát již existující fotografii nebo pořídit novou s pomocí zabudované kamery.
3. Uživatel zvolí způsob splnění úlohy.
  - a) Uživatel zvolí nahrání existujícího snímku a zvolí vyhovující fotografii.
  - b) Uživatel zvolí pořízení fotografie pomocí kamery. S použitím analýzy barev v reálném čase uživatel pořídí vyhovující snímek.
4. Aplikace zanalyzuje barvy na fotografii a vyhodnotí, jestli splňuje úkol. Pokud fotografie nesplňuje zadání, případ užití se skončí. Pokud fotografie je vhodná, aplikace zvětší bodové hodnocení a uvědomí o tom uživatele.

- **UC6 Odstranit úlohu**

Aplikace umožní smazat jeden nebo více nabízených úkolů.

- **UC7 Zobrazit informace o fotografii**

Aplikace umožní zobrazení informací o fotografii, jako jsou její vlastník a barevné složení. Také aplikace bude zobrazovat počet přidání tohoto snímku do oblíbených.

- **UC8 Zobrazit informace o uživateli**

Aplikace umožní zobrazení informací o uživateli, včetně jeho bodového hodnocení.

- **UC9 Zobrazit seznam fotografií registrovaného uživatele**

Aplikace umožní lokálním a registrovaným uživatelům prohlížet fotografie jiných uživatelů.

- **UC10 Vyfiltrovat fotografie podle barevnosti**

Aplikace umožní filtrování fotografií podle barevnosti.

Základní cesta:

1. Příklad užití se začíná, kdy uživatel chce vyhledat fotografie podle určitých barev.
2. Aplikace nabídne zvolit jednu až tři barvy ze seznamu předpřipravených barev, podle kterých uživatel chce filtrovat.
3. Uživatel zvolí jednu až tři barvy.
4. Aplikace vyhledá fotografie, které splňují požadovaná kritéria. Pokud žádná fotografie nebude nalezena, aplikace o tom uvědomí uživatele.

Alternativní cesta:

1. Příklad užití se začíná, kdy uživatel chce vyhledat fotografie podle barev, které byly zadané v úkolu.
2. Uživatel zvolí úkol, podle kterého chce vyfiltrovat fotografie.
3. Aplikace ukáže fotografie, které obsahují požadované dominantní barvy.

- **UC11 Zobrazit seznam naposledy přidanych fotografií**

Aplikace umožní lokálním a registrovaným uživatelům prohlížet seznam naposledy přidanych fotografií na server.

- **UC12 Vytvořit účet**

Aplikace umožní lokálnímu uživateli vytvořit účet na serveru. Pro vytvoření účtu je potřeba mít lokální účet a uvést elektronickou adresu a heslo. Aplikace umožní registraci, pokud na serveru nebude registrovaný uživatel se stejnou elektronickou poštou.

- **UC13 Přihlásit se jako registrovaný uživatel**

Aplikace umožní přihlášení registrovaného uživatele. Pro tyto účely nebude potřeba mít lokální účet.

- **UC14 Nastavit úvodní obrázek uživatele**

Aplikace umožní nastavit fotografii jako úvodní obrázek účtu. Tento obrázek je možné změnit.

- **UC15 Přidat fotografii do oblíbených**

Aplikace umožní přidat fotografii do oblíbených.

- **UC16 Nahrát fotografii**

Registrovaný uživatel má možnost nahrát fotografii na server, tím zveřejní svou fotografii pro všechny registrované a neregistrované uživatele. Uživatel může nahrát fotografii buď přímo do účtu nebo po splnění úkolu.

- **UC17 Odstranit fotografii**

Aplikace umožní odstranit dříve nahranou fotografii.

### 2.5.3 Pokrytí případů užití

I když případy užití jsou popsány z hlediska uživatele a funkční požadavky jsou definované ze strany samotné aplikace, tyto přístupy by se měly navzájem úplně pokrývat, protože se jedná o stejný systém. Toto pokrytí je znázorněné v tabulce 2.2.

## 2.6 Algoritmy pro nalezení dominantní barvy

Jedna z nejdůležitějších záležitostí, kterou budu potřebovat řešit ve své aplikaci je nalezení dominantních barev na obrázku. Obtížnost tohoto problému spočívá v tom, že počítač řeší ten problém matematicky přesně, což ne vždy odpovídá tomu, co by člověk očekával uvidět. Každému se může zdát, že dominantní barva na nějakém zadaném obrázku je odlišná od výsledku, který vypočítá počítač.

První a naivní řešení spočívá ve spočítání průměrné barvy. Tento algoritmus je rychlý, ale skoro pro všechny obrázky výsledkem bude šedá barva [17]. Proto to potřebuje přesnější a složitější způsob řešení.

Tento problém je možné rozdělit na dva následující výroky.

Tabulka 2.2: Pokrytí funkčních požadavků

	FR1	FR2	FR3	FR4	FR5	FR6	FR7	FR8	FR9
UC1				✓					
UC2				✓					
UC3				✓					
UC4	✓								
UC5	✓	✓	✓						
UC6	✓								
UC7								✓	
UC8					✓				
UC9								✓	
UC10								✓	✓
UC11								✓	
UC12					✓				
UC13					✓				
UC14					✓	✓			
UC15							✓		
UC16						✓			
UC17						✓			

1. Dominantnost barvy je určena tím, kolik obrazových bodů stejné barvy obsahuje obrázek.
2. Dvě barvy jsou různé, pokud Euklidovská vzdálenost mezi nimi je velká.

První výrok vede do problému, že většina obrázků neobsahuje mnoho obrazových bodů přesně stejné barvy. Úplně modrý obrázek může zahrnovat stovky odstínů modré barvy. Druhý výrok vede k tomu, jak zjistit, že dva obrazové body jsou „stejně“ barvy. Pro řešení tohoto problému existuje mnoho formulí a algoritmů, které budou probrané v následující kapitole. Ale nejpochoptelnější a nejpoužívanější způsob je nalezení Euklidovské vzdálenosti mezi body ve tří-dimenzionálním prostoru, který reprezentuje RGB prostor [18].

Dále dostáváme problém, kde máme množinu bodů a chceme je rozdělit do skupin podle podobnosti a pak najít nejlepšího reprezentanta skupiny. Toto je popis problému klasterizace, který je velice podobný problému nalezení dominantních barev.

### 2.6.1 Klasterizace pomocí k-means

V této sekci čerpám z [19] a [20].

Jedním ze základních algoritmů klasterizace je k-means clustering. Tento algoritmus řeší jeden z nejstarších problémů výpočetní geometrie. Pro dané

celé číslo  $k$  a množinu dat v prostoru  $\mathcal{X} \subset R^d$ , je potřeba spočítat  $k$  středů tak, aby minimalizovat  $\phi$ ,

$$\phi = \sum_{x \in \mathcal{X}} \min \|x - c\|^2.$$

Přesné řešení tohoto problému je NP-těžké, ale před dvaceti pěti lety Stuart P. Lloyd nabídl optimální řešení problému. Algoritmus Lloyda se často nazývá k-means clustering a je populární dodnes.

Algoritmus k-means je rychlý a velmi používaný způsob vyřešení tohoto problému, i když neposkytuje žádnou jistotu ve výsledku. Tento algoritmus se skládá ze čtyř následujících kroků.

1. Libovolně zvolit počáteční  $k$  středů  $\mathcal{C} = \{c_1, c_2, \dots, c_k\}$ .
2. Pro každé  $i \in \{1, \dots, k\}$ , vymezit shluk  $C_i$  tak, aby obsahoval množinu těch bodů z  $\mathcal{X}$ , které jsou bližší k  $c_i$  než k  $c_j$ , pro každé  $i \neq j$ .
3. Pro každé  $i \in \{1, \dots, k\}$ , nastavit  $c_i$  tak, aby byl těžištěm všech bodů v  $C_i$ :  $c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$ .
4. Opakovat kroky 2 a 3, až pokud  $\mathcal{C}$  se nepřestane měnit.

Algoritmus k-means konverguje k bodu, i když to není minimální součet čtverců. Je to založeno na tom, že algoritmus je heuristický a konverguje jen k lokálnímu minimu. Výsledek tohoto algoritmu je velmi závislý na volbě počátečních středů.

Znázornění tohoto algoritmu pro dvojrozměrný prostor je si možné prohlédnout v obrázku B.1.

### 2.6.2 Klasterizace pomocí k-means plus plus

V této kapitole čerpám z [19].

Algoritmus k-means plus plus je stejný algoritmus k-means s vylepšenou inicializací středů. Označíme  $D(x)$  jako funkci, která určuje nejkratší vzdálenost mezi nějakým bodem a nejbližším již zvoleným středem. Pak můžeme uvést následující kroky.

- 1a. Zvolit uniformně náhodně jeden střed  $c_1$  ze všech bodů z  $\mathcal{X}$ .
- 1b. Zvolit nový střed  $c_i$  s pravděpodobností  $\frac{D(x)^2}{\sum_{x \in \mathcal{X}} D(x)^2}$ .
- 1c. Opakovat krok 1b, až pokud nebudeme mít k dispozici  $k$  středů.
2. Pokračovat stejně jako v obyčejném k-means.

Tabulka 2.3: Význam hodnot, kterých může nabývat Delta E, převzato z [21]

Interval	Význam
$\leq 1,0$	Nerozlišitelné pro lidské oko
1-2	Rozlišitelné při pozorném prozkoumání
2-10	Rozlišitelné na první pohled
11-49	Barvy jsou spíš podobné než opačné
100	Barvy jsou opačné

Algoritmus k-means plus plus je rychlejší a díky pečlivé volbě počátečních středů dává mnohem lepší výsledky. Největší problém k-means je v tom, že pokud náhodně zvolené středy budou na stejném místě a spojí se, algoritmus je již nikdy nerozdělí znovu. Vylepšený k-means plus plus umí zamezit vzniku tohoto problému. Navíc při použití stejného semínka v náhodném generátoru, k-means plus plus bude dávat stejné výsledky na stejných vstupních datech.

## 2.7 Algoritmy pro nalezení vzdálenosti mezi barvami

V následující kapitole čerpám z [21], [22] a [23].

Lidé se zamýšleli nad problémem nalezení vzdálenosti mezi barvami ještě na začátku 20. století. A proto v roce 1913 byla založena organizace International Commission on Illumination (zkráceně CIE, pochází od francouzského názvu Commission Internationale de l'Éclairage), účelem které bylo standardizovat a zkoumat všechno, co se týká světla, osvětlení a barev. Za velký průlom se považuje práce publikovaná v roce 1976, kdy se poprvé zavedl pojem Delta E.

Delta E ( $dE$  nebo  $\Delta E$ ) udává vzdálenost mezi barvami z hlediska vnímání lidským okem. Tabulka 2.3 uvádí význam každého intervalu.

V dnešní době existují tři hlavní algoritmy definovaných organizací CIE pro nalezení vzdálenosti mezi barvami, a to jsou Delta E 76, Delta E 94 a Delta E 2000. Jelikož všechny tyto algoritmy používají různé vzorce a nejsou konzistentní, přesný význam Delta E se může mírně měnit, stejně jako i tabulka. Také není vyloučeno, že nalezené Delta E může být menší než 1, ale barvy budou rozeznatelné pro lidské oko.

Algoritmus Delta E 76 používá CIE LAB prostor, který je založen na teorii opačných barev. V tomto prostoru L vyjadřuje jas barvy, A reprezentuje červené a zelené odstíny, a B reprezentuje žluté a modré. Tento prostor byl vynalezen pro přiblížení lidského vnímání barev, na rozdíl od RGB a CMYK, které pro tyto účely nejsou použitelné. Proto před zpracováním algoritmy pro nalezení vzdálenosti mezi barvami je potřeba převést barvy z libovolného jiného formátu do formátu LAB.

Jiné výše zmíněné algoritmy používají CIE LCH prostor, který je velmi podobný CIE LAB, ale místo pravoúhlých souřadnic pro určení bodu v barevném prostoru používá souřadnice polární. V tomto prostoru L má stejný význam jako v CIE LAB, C reprezentuje nasycení (saturation) barvy, a H je hodnota, která odpovídá odstínu (hue) barvy.

### 2.7.1 Delta E 76

Delta E76 je nejsnadnější a nejstarší algoritmus, který pochází od Euklidovské vzdálenosti. LAB prostor je tři-dimenzionální prostor barevných bodů, a proto přímé měření vzdáleností funguje.

$$\Delta E_{ab}^* = \sqrt{(L_2^* - L_1^*)^2 + (a_2^* - a_1^*)^2 + (b_2^* - b_1^*)^2}$$

V dnešní době se tento algoritmus používá jen tehdy, když není potřeba vypočítávat přesné výsledky a musí to být okamžitě zpracované. Tato formule se používá například pro zpracování obrázku v reálném čase.

### 2.7.2 Delta E 96

V roce 1996 původní formule Delta E 76 byla zlepšena. Nový algoritmus bere v potaz jas a barevný tón. Také tato formule poprvé zavádí konstanty pro různé účely použití, jako jsou textilní průmysl nebo grafika. Jediným problémem tohoto algoritmu zůstává nepřesnost v počítání rozdílu mezi jasy jednotlivých barev.

$$\Delta E_{94}^* = \sqrt{\left(\frac{\Delta L^*}{k_L S_L}\right)^2 + \left(\frac{\Delta C_{ab}^*}{k_C S_C}\right)^2 + \left(\frac{\Delta H_{ab}^*}{k_H S_H}\right)^2}$$

Formule pro počítání jednotlivých proměnných je možné vidět na obrázku C.1.

### 2.7.3 Delta E 2000

V roce 2000 organizace CIE předvedla algoritmus, který má název Delta E00. V tomto algoritmu byly opravené všechny chyby v počítání jasu barev. V dnešní době to je nejpřesnější algoritmus pro nalezení vzdálenosti mezi barvami. Ale i přesto zůstávají situace, kdy subjektivní názor člověka se liší od výsledku tohoto algoritmu.

$$\Delta E_{00}^* = \sqrt{\left(\frac{\Delta L'}{k_L S_L}\right)^2 + \left(\frac{\Delta C'}{k_C S_C}\right)^2 + \left(\frac{\Delta H'}{k_H S_H}\right)^2 + R_T \frac{\Delta C'}{k_C S_C} \frac{\Delta H'}{k_H S_H}}$$

Formule pro počítání jednotlivých proměnných je možné vidět na obrázku C.2.

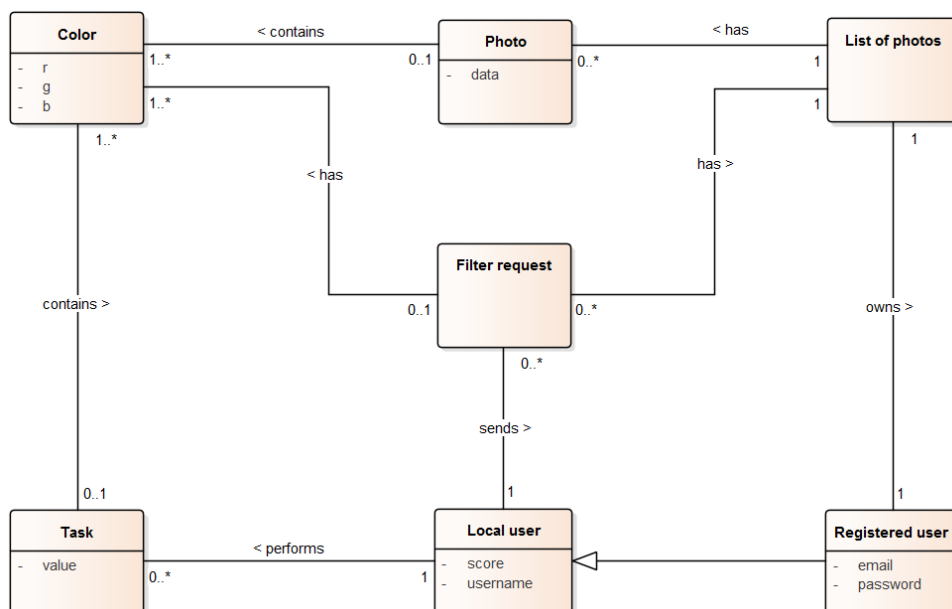




# Návrh

## 3.1 Doménový model

Modelování domény je jedna z cest, jak přejít od vymezení chování systému, což popisují výše uvedené scénáře a případy užití, do návrhu samotné struktury aplikace. Doménový model definuje objekty z reálného světa a vztahy mezi nimi, které v souhrnu musí kompletně popisovat problémovou doménu navržené aplikace. Jeho hlavním cílem není přesně určit softwarové komponenty, ale pomoci pochopit problémovou oblast a poskytnout základ pro další navrhování aplikace [24]. Doménový model je možné vidět na obrázku 3.1.



Obrázek 3.1: Doménový model

### 3.1.1 Lokální uživatel

Lokální uživatel je entita reprezentující člověka, který z vlastních důvodů nechce nebo nemůže být registrovaný na serveru. Tento typ uživatelů obsahuje přezdívku pro zobrazení v lokálním účtu a bodové hodnocení, které je vypočítáno na základě splněných úkolů. Od lokálního uživatele se dědí entita registrovaného uživatele.

### 3.1.2 Registrovaný uživatel

Registrovaný uživatel je entita uchovávající veškerou informaci o uživateli, jako jsou elektronická pošta a heslo. Předávání informací na server musí být zabezpečené, aplikace musí podporovat proces autentifikace a autorizace. Každý registrovaný uživatel může, ale nemusí mít seznam nahraných fotografií.

### 3.1.3 Fotografie

Fotografie je nejdůležitější entita v navržené aplikaci uchovávající samotný obsah obrázku a časovou značku udávající čas nahrání na server. Každá fotografie musí obsahovat aspoň jednu dominantní barvu, která se považuje za její hlavní součást a vlastníka, který nahrál fotografii na server.

### 3.1.4 Seznam fotografií

Seznam fotografií je entita, která může obsahovat nekonečný počet obrázků. Používá se pro reprezentaci všech fotografií uživatele, přehled vyfiltrovaných fotografií a pro zobrazení naposledy nahraných obrázků na server.

### 3.1.5 Barva

Barva je jedna ze základních entit. Barva se skládá ze tří základních částí, a to jsou červená, zelená a modrá složky. Každý obrázek obsahuje aspoň jednu dominantní barvu. Podobně úkol vždy má barvy, které musí být dominantní v pořizované fotografii pro akceptaci této úlohy.

### 3.1.6 Úkol

Úkol je následující ze základních entit v navržené aplikaci a reprezentuje barevnou úlohu, kterou uživatel může splnit a přidat tím další body ke svému hodnocení. Součástí každého úkolu je bodová hodnota, která se po splnění přičte k výslednému hodnocení uživatele. Také každý úkol obsahuje minimálně jednu dominantní barvu, kterou je potřeba mít v obrázku pro splnění úlohy.

### 3.1.7 Požadavek na filtrování

Tato entita reprezentuje požadavek na filtrování, který musí obsahovat aspoň jednu barvu. Po vyřízení se vrátí seznam fotografií vyhovujících danému kritériu, kde každý obrázek bude obsahovat zároveň všechny barvy z požadavku.

## 3.2 Diagramy aktivit

V následující sekci čerpám z [25] a [26].

Diagram aktivit je jeden ze základních UML diagramů znázorňující tok řízení z jedné aktivity do druhé. Tento typ diagramů je považovaný za jeden ze stavových. Hlavním jeho účelem je popsat paralelní, sekvenční a podmíněné aktivity na podrobné úrovni. Většinou diagramy aktivity se kreslí jen pro nejdůležitější byznys procesy v aplikaci.

Občas definování průběhu jednotlivých aktivit v aplikaci, která je navržena jako klient-server může být dost komplikované, a proto je vhodné popsat nejsložitější toky aktivit pomocí jejich diagramů.

### 3.2.1 Filtrování fotografií

Diagram B.2 popisuje filtrování fotografií podle barevnosti. Existují dvě cesty, jak vyfiltrovat fotografie, které obsahují dominantní barvy: zvolit ve filtrovacím menu nebo inspirovat se konkrétním úkolem uživatele. Na diagramu je znázorněné, že filtrování se provádí na serveru, který vrátí seznam vyhovujících fotografií. Podle toho, jestli seznam obsahuje aspoň jeden obrázek, klient rozhodne vypsat hlášku o absenci potřebných fotografií nebo ukázat získaný seznam.

### 3.2.2 Splnění úkolu

Diagram aktivit tohoto procesu je znázorněný na obrázku B.3. Splnění úkolu je nejzákladnější proces v navržené aplikaci. Existují dva způsoby splnění úlohy, a to jsou pořízení fotografie pomocí kamery aplikace nebo výběr z galerií. Pokud zvolený nebo pořízený obrázek splňuje zadání, bodové hodnocení uživatele se zvětší. Pak existuje možnost sdílení fotografie za podmínky, že uživatel má přístup k internetu a je zaregistrovaný. Zabezpečení tohoto procesu bude popsáno v kapitole 4.2.2.

### 3.2.3 Registrace na serveru

Diagram znázorňující registraci na serveru je možné vidět na obrázku B.4. Registrace uživatele na serveru je těžší tím, že existuje možnost používání aplikace bez registrace, a proto první část diagramu se věnuje větvení na přihlášení a vytváření lokálního účtu. Další část je logická a je věnována vyplnění

potřebných údajů a její kontrole na serveru. Zabezpečení tohoto procesu bude popsáno v kapitole 4.2.2.

#### 3.2.4 Přihlášení do účtu

Diagram aktivit na obrázku B.5 popisuje přihlášení registrovaného uživatele do svého účtu. Na tomto diagramu je vidět průběh vyplnění potřebných údajů a následného ověření na serveru. Veškeré zabezpečení tohoto procesu bude popsáno v kapitole 4.2.2.

### 3.3 Návrh serverové části aplikace

Jelikož jednou z hlavních funkcí v navržené aplikaci je sdílení obrázků a poskytování přístupu přes internet k fotografiím uživatelů, je potřeba navrhnout správnou strukturu aplikace. Přímočarou volbou pro tyto účely je dvouvrstvá architektura klient-server. Toto je síťová architektura, kde jeden program (klient) žádá o služby nebo prostředky jiný program (server). Je důležité, že více klientů může být obsluhováno jedním serverem, což je základní požadavek na navrženou aplikaci. V závislosti na rozdělení tří základních vrstev (prezentační, byznys a datová) klient může být tenký a tlustý [27].

Klient, který v sobě obsahuje prezentační a byznys vrstvy, se nazývá tlustý. Naopak klient je tenký, pokud server plní funkce byznys vrstvy, což znamená, že provádí veškeré výpočty a obsahuje rozhodovací logiku. Příkladem tenkého klienta může být webový prohlížeč [28].

Veškerá logika v navržené aplikaci se bude provádět na straně klienta kvůli možnosti splnění úkolů bez internetového připojení, což znamená, že aplikace bude používat model tlustého klienta.

Hlavním účelem serveru bude získávání dat od klientů, uchování v databázi, filtrování podle požadavku a posílání zpátky. Také cílem serveru bude provádění zabezpečení přihlášení, registrace, nahrávání a mazání fotografií, přidávání do oblíbených a zvětšení bodového hodnocení uživatele. Jelikož zakoupený pro tyto účely server není moc výkonný, není možné udělat aplikaci podobnou populárním aplikacím pro sdílení fotografií, a proto se zaměřím na udržovatelnost a snadnost vytváření serverové části.

Existují dvě základní možnosti vytvoření serveru, a to jsou programování v Javě a v node.js, což je framework založený na JavaScriptu. V dnešní době existuje hodně příznivců jak psaní serveru v Javě, tak i v JavaScriptu, protože každý z nich má svoje výhody na nevýhody. V tabulce 3.1 jsou popsány výhody programování serveru v Javě a v node.js [29].

Protože node.js je přímo určený pro tvorbu internetových aplikací, napsání serveru v tomto frameworku je snadnější a výkonnost lepší, což je to, co potřebuji. Pro komunikaci mezi serverem a klientem, budu používat RESTfull API přes HTTP protokol. Toto řešení je logické a velmi často používané. Jako formát posílaných dat byl zvolený JSON, který je přímo součástí JavaScriptu.

Tabulka 3.1: Výhody programování serveru v Javě a v node.js

Java	node.js (JavaScript)
lepší IDE	rychlost
používání více vláken	škálovatelnost
velké množství knihoven	pohodlnější práce s JSON
	snadnější použití

Následujícím krokem je návrh HTTP požadavků, které budou pokrývat všechny potřeby aplikace.

- **GET /photo\_data/:id**

Požádá o data fotografie se zadaným *id*. Vrací 200 (OK), pokud všechno proběhlo v pořádku a 404 (Not Found), pokud fotografie se zadaným *id* nebyla nalezena.

- **GET /bg\_photo\_data/:id**

Požádá o data fotografie pozadí se zadaným *id*. Vrací 200 (OK), pokud všechno proběhlo v pořádku a 404 (Not Found), pokud fotografie pozadí se zadaným *id* nebyla nalezena.

- **POST /photo\_add**

Přidá fotografii na účet uživatele. Vrací 201 (Created), pokud fotografie byla úspěšně vytvořena, 403 (Forbidden), pokud byl zaznamenán neoprávněný přístup a 400 (Bad Request), pokud požadavek nemá správné parametry.

- **POST /bg\_photo\_add**

Přidá fotografii pozadí na účet uživatele. Vrací 201 (Created), pokud fotografie pozadí byla úspěšně vytvořena, 403 (Forbidden), pokud byl zaznamenán neoprávněný přístup a 400 (Bad Request), pokud požadavek nemá správné parametry.

- **DELETE /photo\_delete/:id**

Odstraní fotografii se zadaným *id*. Vrací 200 (OK), pokud fotografie byla úspěšně smazána, 403 (Forbidden), pokud byl zaznamenán neoprávněný přístup, 400 (Bad Request), pokud požadavek nemá správné parametry a 404 (Not Found), pokud fotografie se zadaným *id* nebyla nalezena.

- **POST /register**

Provádí registraci uživatele. Vrací 201 (Created), pokud uživatel byl úspěšně zaregistrován, 409 (Conflict), pokud elektronická pošta je již obsazena a 400 (Bad Request), požadavek nemá správné parametry.

### 3. NÁVRH

---

- **POST /login**

Provádí přihlášení uživatele. Vrací 200 (OK), pokud přihlášení uživatele se podařilo, 403 (Forbidden), pokud údaje nejsou správné a 400 (Bad Request), pokud požadavek nemá správné parametry.

- **GET /photo\_info/:id**

Požádá o informace o fotografii se zadaným *id*, která obsahuje *id* uživatele, přezdívku, časovou značku, seznam barev a seznam s *id* uživatelů, kteří přidali danou fotografii do oblíbených. Vrací 200 (OK), pokud všechno proběhlo v pořádku, a 404 (Not Found), pokud fotografie se zadaným *id* nebyla nalezena.

- **GET /photos\_list/:last\_item**

Požádá o seznam *id* dalších osmi fotografií od *id = last\_item*, které byly naposledy přidány na server. Vrací 200 (OK), pokud všechno proběhlo v pořádku.

- **GET /photos\_list\_by\_user/:user\_id/:last\_item**

Požádá o seznam *id* dalších osmi fotografií od *id = last\_item* uživatelů s *user\_id*. Vrací 200 (OK), pokud všechno proběhlo v pořádku.

- **GET /photos\_list\_by\_colors/:last\_item**

Požádá o seznam *id* dalších osmi fotografií od *id = last\_item*, které vyhovují zadaným barvám (barvy se zadávají jako parametry). Vrací 200 (OK), pokud všechno proběhlo v pořádku.

- **GET /user/:id**

Požádá o data uživatele s odpovídajícím *id*. Vrací 200 (OK), pokud všechno proběhlo v pořádku, a 404 (Not Found), pokud uživatel se zadaným *id* nebyl nalezen.

- **PUT /user\_score**

Nastaví bodové hodnocení uživatele (počet bodů, včetně *id* uživatele a hashi hesla se nachází v těle PUT požadavku). Vrací 200 (OK), pokud všechno proběhlo v pořádku, 404 (Not Found), pokud uživatel nebyl nalezen a 403 (Forbidden), pokud byl zaznamenán neoprávněný přístup.

- **POST /like**

Přidá fotografii do oblíbených (*id* fotografie, včetně *id* uživatele a hashi hesla se nachází v těle POST požadavku). Vrací 200 (OK), pokud všechno proběhlo v pořádku, a 404 (Not Found), pokud uživatel anebo fotografie nebyli nalezeni a 403 (Forbidden), pokud byl zaznamenán neoprávněný přístup.

- **POST /dislike**

Vymaže fotografii z oblíbených (*id* fotografie, včetně *id* uživatele a *hashi* hesla se nachází v těle POST požadavku). Vrací 200 (OK), pokud všechno proběhlo v pořádku, a 404 (Not Found), pokud uživatel anebo fotografie nebyli nalezeni a 403 (Forbidden), pokud byl zaznamenán neoprávněný přístup.

- **GET /key**

Požádá o veřejný klíč (public key) serveru. Vrací 200 (OK), pokud všechno proběhlo v pořádku.

- **GET /nonce**

Požádá o unikátní číslo, které je možné použít jen jednou (zkratka nonce, anglicky number used only once). Vrací 200 (OK), pokud všechno proběhlo v pořádku.

## 3.4 Databázový model

V předchozích kapitolách byla provedena podrobná analýza podobných řešení, byly definované funkční požadavky, případy užití a scénáře, také byl vytvořen doménový model a byla navržena architektura aplikace, tudíž všechno je připravené pro návrh konkrétní struktury aplikace. V následujících sekcích budou popsány databázové modely serveru a klienta.

### 3.4.1 Server

Databázový model serveru je možné vidět na obrázku 3.2.

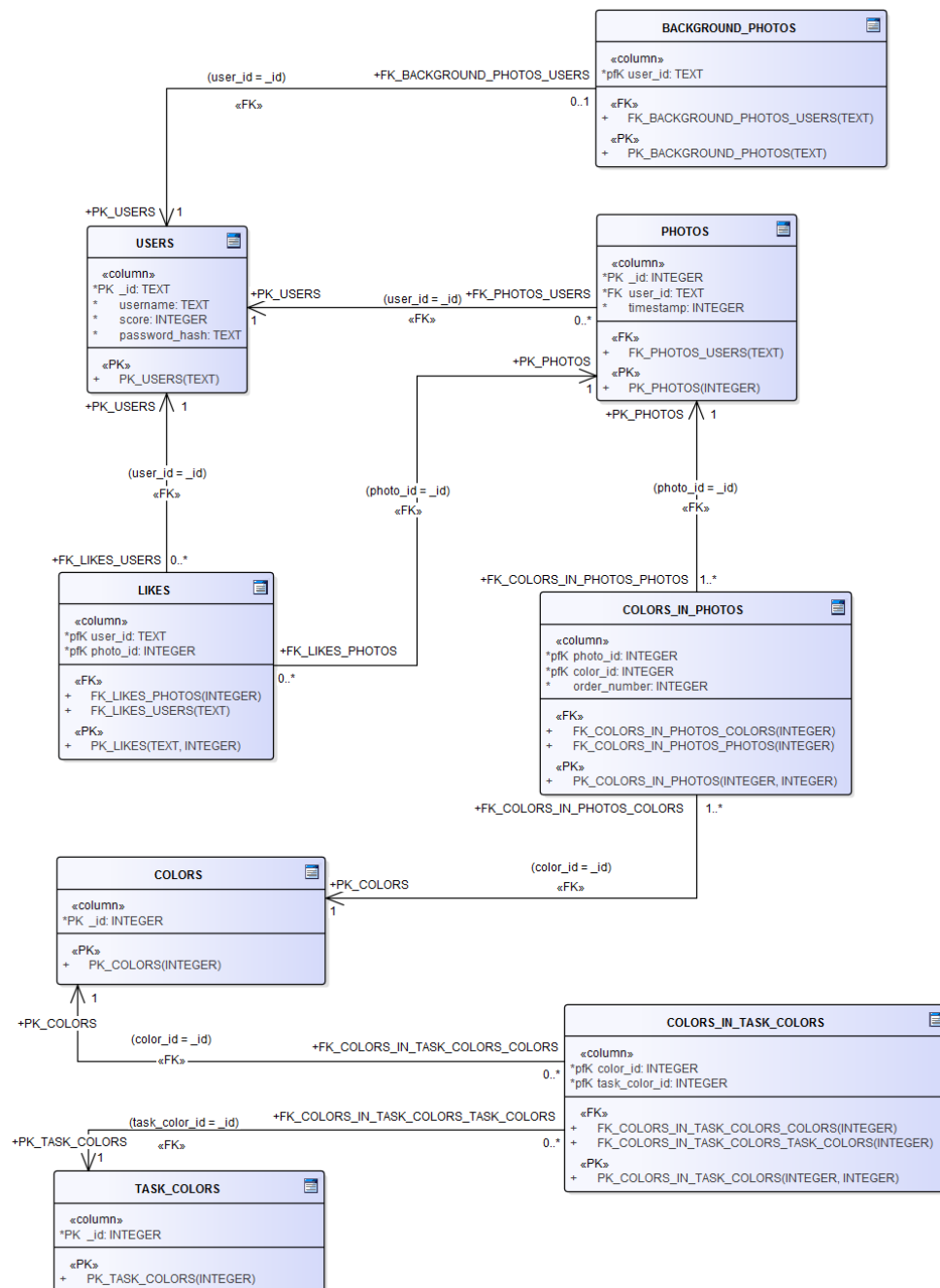
#### 3.4.1.1 Users

Uživatel je jednou z nejdůležitějších tabulek v databázovém modelu serveru. Primárním klíčem této tabulky je elektronická pošta uživatele. Dalšími položkami jsou přezdívka, hodnocení uživatele a hashe hesla. Požaduje se unikátnost jen elektronické pošty, to znamená, že může existovat několik uživatelů se stejnou přezdívkou. Jelikož v databázi klienta stejnojmenná tabulka také má hodnocení, uživatel může splňovat úkoly bez internetového připojení, pak se při opětovném navázání spojení hodnocení synchronizuje se serverem.

#### 3.4.1.2 Photos

Fotografie je další základní entita v tomto databázovém modelu. Jako primární klíč bylo zvolené identifikační číslo (*id*), které se zvětšuje automaticky po nahrávání nové fotografie. Kromě toho tato tabulka obsahuje cizí klíč, což je elektronická pošta uživatele a časovou značku, kdy fotografie byla nahrána na server. Samotný obsah fotografie je uložen ve veřejné složce na serveru.

### 3. NÁVRH



Obrázek 3.2: Databázový model serveru



#### 3.4.1.3 Likes

Entita, která reprezentuje přidání fotografie do oblíbených a je asociační tabulkou mezi fotografií a uživatelem.

#### 3.4.1.4 Colors

Entita, která reprezentuje barvu, kde primárním klíčem je její hodnota v ARGB<sup>3</sup> prostoru.

#### 3.4.1.5 Colors\_in\_photos

Asociační tabulka mezi barvami a fotografiemi, která uchovává informaci o dominantních barvách v obrázcích. Tato tabulka je rozšířena o položku *order\_number*, která udává pořadí barvy v seznamu vypočítané klientem, který je seřazený podle dominantnosti. Řazení barev je potřeba, protože v informaci o fotografii se barvy vykreslují právě v tomto pořadí.

#### 3.4.1.6 Task\_colors

Entita uchovávající přesně definované barvy, které se používají pro úlohy. Podle těchto barev se bude provádět filtrování.

#### 3.4.1.7 Colors\_in\_task\_colors

Asociační tabulka, která ke každé obyčejné barvě udává nějaký počet nejvíce podobných barev definovaných pro úlohy. Barva může nemít podobnou barvu ze zadání, pokud je to například černá, nebo výjimečně může mít několik. Tato tabulka je základem pro filtrování podle barevnosti.

#### 3.4.1.8 Background\_photos

Entita reprezentující úvodní obrázek uživatele. Obsahuje identifikační číslo, které je primárním klíčem. Úvodní obrázek uživatele nesouvisí ani s barvami ani s přidáváním fotografie do oblíbených.

### 3.4.2 Klient

Na obrázku 3.3 je si možné prohlédnout databázový model klienta.

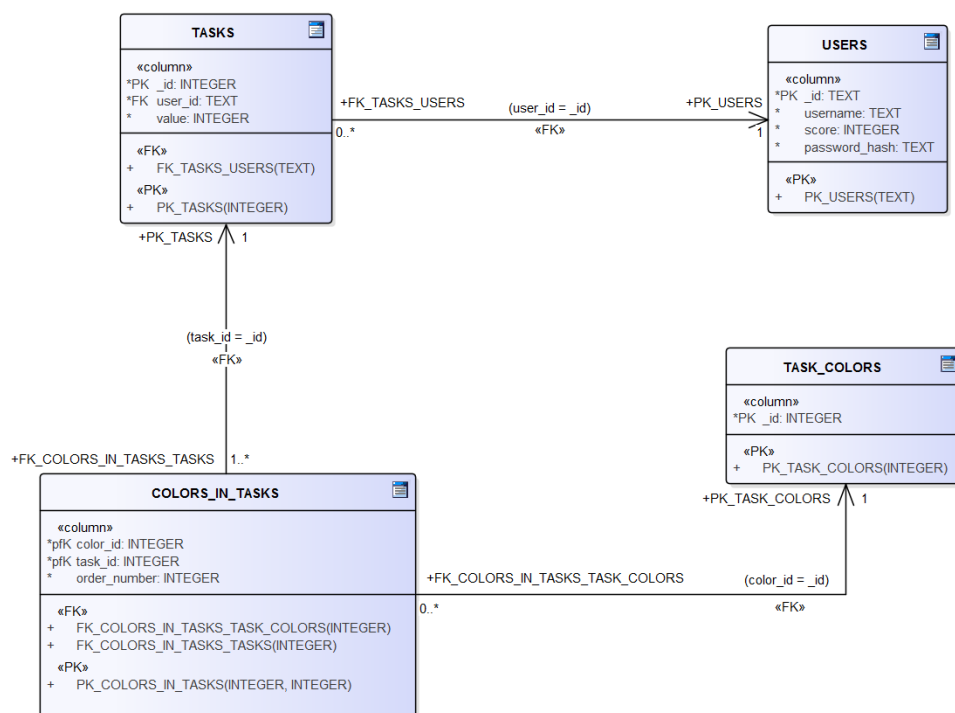
#### 3.4.2.1 Users

Hlavní rozdíl mezi tabulkou na serveru a klientu je v tom, že se na straně klienta ještě musí ukládat lokální uživatelé. Registrování lokálního uživatele

---

<sup>3</sup>ARGB prostor je klasický RGB prostor rozšířený o položku, která uchovává informaci o průhlednosti.

### 3. NÁVRH



Obrázek 3.3: Databázový model klienta




























se provádí tak, že se do položky username zapíše přezdívka uživatele. Primární klíč se vytvoří tak, že se přezdívka překopíruje do položky elektronické pošty. Toto zamezí vytváření více lokálních uživatelů se stejným jménem. Pokud lokální uživatel se rozhodne zaregistrovat, tak vyplní elektronickou poštu a heslo. Tím pádem, po registraci lokálního uživatele na serveru bude možné založit nový lokální účet se stejnou přezdívkou.

#### 3.4.2.2 Tasks

Tabulka s úlohami je uložena jen v databázi klienta, aby splnění zadání bylo dostupné bez internetového připojení. Tabulka obsahuje primární klíč (identifikační číslo) a bodovou hodnotu, která se přičte k hodnocení uživatele po splnění zadání.

#### 3.4.2.3 Colors\_in\_tasks

Asociační tabulka, která spojuje barvy a úlohy. Navíc obsahuje pořadí barvy v seznamu barev ze zadání pro stejné zobrazení barevnosti fotografie.

#2A2A2A 	#2A2A80 	#2A2AD5 	#2A802A 	#2A8080 
#2A80D5 	#2AD52A 	#2AD580 	#2AD5D5 	#802A2A 
#802A80 	#802AD5 	#80802A 	#808080 	#8080D5 
#80D52A 	#80D580 	#80D5D5 	#D52A2A 	#D52A80 
#D52AD5 	#D5802A 	#D58080 	#D580D5 	#D5D52A 
#D5D580 		#D5D5D5 		

Obrázek 3.4: Paleta obsahující 27 barev

#### 3.4.2.4 Task\_colors













Stejná tabulka jako na serveru, kterou je si možné prohlédnout v sekci 3.4.1.6.

## 3.5 Návrh barev pro úlohy a filtrování

Před přechodem do návrhu uživatelského rozhraní je potřeba ještě navrhnout seznam barev, které se budou používat v úkolech. Výsledná paleta musí obsahovat všechny možné hlavní barvy a jejich odstíny, které jsou dostatečně odlišné pro lidské oko. Jelikož výsledná aplikace musí být zajímavá pro uživatele a rozvíjet kreativitu, výsledná paleta by měla obsahovat aspoň 10 barev.

První a nejlogičtější způsob generování palety je rovnoměrně rozdělit RGB prostor a najít středy jednotlivých krychlí. Tím postupem je možné získat 27 barev, které je si možné prohlédnout na obrázku 3.4. Jelikož navržená aplikace je o barvách ve světě, nebylo by dobře mít v zadáních černou (#2A2A2A), bílou (#D5D5D5) nebo šedou (#808080) barvu, a proto zůstane 24 barev.

Jedním z nejdůležitějších problémů této palety je, že obsahuje barvy nerozlišitelné pro lidské oko, jako jsou například barvy #80D5D5 a #2AD5D5. Podobnost barev v paletě může způsobit problémy při filtrování obrázků a splnění úloh. Navíc 24 barev je příliš velké číslo, tento počet barev bude těžké znázornit v uživatelském rozhraní.

#480000 	#E73030 	#FF8A00 	#F7E129 	#471364 
#15649C 	#787EB9 	#9ECAE1 	#824A2B 	#10500C 
#737047 	#6DBE51 	#A21F97 	#F768A1 	#FFC6AC 

Obrázek 3.5: Výsledná paleta obsahující 15 barev

Jelikož mnou nebyl nalezen jiný přesný způsob vypočítání palety s použitím algoritmů Delta E 96 nebo Delta E 2000, rozhodl jsem se udělat paletu z 15 barev. Hlavním požadavkem pro generování bylo to, že rozdíl mezi všemi barvami podle algoritmu Delta E 2000 by měl být větší než 20. Výslednou paletu je možné vidět na obrázku 3.5.

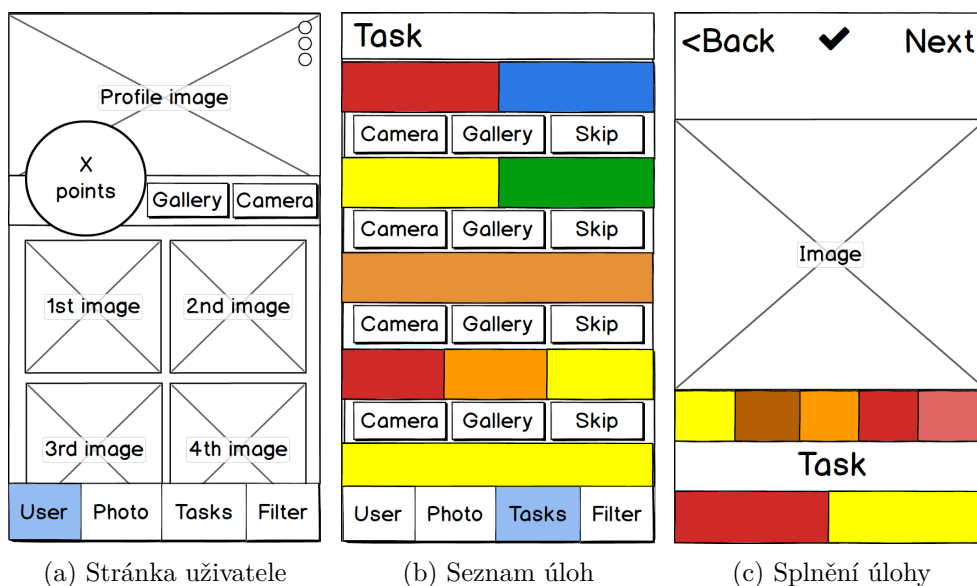
### 3.6 Návrh uživatelského rozhraní

Návrh uživatelského rozhraní probíhal ve dvou etapách, a to byl návrh lo-fi prototypu a hi-fi prototypu. Low-fidelity prototyp se charakterizuje rychlou realizací bez ohledu na design. Většinou se pro vytváření lo-fi prototypu používají tužka a papír nebo speciální programy pro hrubý náčrt uživatelského rozhraní. Naopak high-fidelity prototyp musí reprezentovat finální vzhled aplikace [30]. Obě verze uživatelského rozhraní byly otestovány pomocí deseti heuristických pravidel pro uživatelské rozhraní od Jakoba Nielsena [31].

Následující návrhy byly vytvořeny hlavně pomocí Android guidelines a přednášek magisterského předmětu „Návrh uživatelských rozhraní“ [33] přednášených na Fakultě informačních technologií Českého vysokého učení technického v Praze.

#### 3.6.1 První návrh uživatelského rozhraní

Aplikace bude obsahovat čtyři hlavní obrazovky, které budou odpovídat za hlavní funkčnosti aplikace: stránka uživatele, seznam úkolů, filtrovací menu a seznam naposledy přidávaných fotografií. Přejít mezi těmito obrazovkami se bude provádět pomocí čtyř záložek dolů na obrazovce (prvek Tab Layout v Androidu). Pro vytvoření prvního prototypu aplikace je zapotřebí provést analýzu úkolů aplikace, čímž se zabývá následující sekce.



Obrázek 3.6: Návrh jednotlivých obrazovek (první verze)

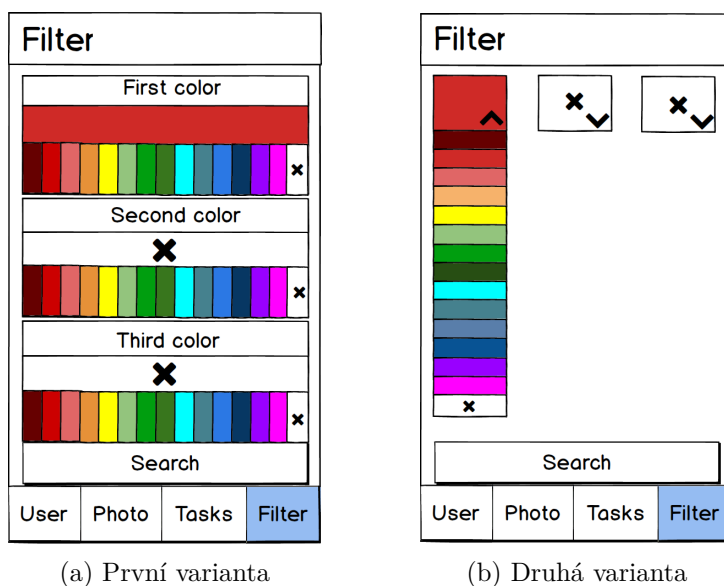
### 3.6.1.1 Úkoly aplikace

Seznam úkolů aplikace (anglicky task list) popisuje všechny možné funkčnosti z pohledu uživatele. Úkoly mohou mít různou úroveň abstrakce, ale v tomto kroku to nehraje žádnou roli. Návrh těchto úloh vychází z případů užití a celkové analýzy aplikace. Kompletní seznam úkolů rozdělených podle funkčnosti je si možné prohlédnout v příloze D.

### 3.6.1.2 Wireframy

Jednou z nejdůležitějších obrazovek v aplikaci je stránka uživatele. Podle analýzy úkolů aplikace tato obrazovka by měla obsahovat profilový obrázek, základní informaci o uživateli, seznam přidávaných fotografií, dvě tlačítka pro nahrání fotografií (pomocí galerie a kamery) a menu pro odhlášení a jiné nastavení. První návrh této stránky je možné vidět na obrázku 3.6a. Profilový obrázek se nachází v horní části obrazovky, přezdívka je napsána bílou barvou uprostřed. Body za splněné úkoly jsou umístěné v kroužku, tlačítka pro přidání fotografií na server jsou napravo od bodového hodnocení. Seznam obrázků zabírá přibližně polovinu prostoru na obrazovce. Důležitou poznámkou je, že při přechodu na stránku jiného uživatele obrazovka musí vypadat podobně. Tuto stránku je možné získat odebráním tlačítek pro přidávání fotografií a menu pro nastavení.

Podle prvotního návrhu obrazovka se seznamem úkolů by měla vypadat jako na obrázku 3.6b. Každá položka seznamu obsahuje barevný pruh, který se může skládat z jedné až tří barev. Pod pruhem jsou tři tlačítka odpovídající



(a) První varianta

(b) Druhá varianta

Obrázek 3.7: Návrh filtrovacího menu (první verze)

za splnění zadání pomocí kamery, splnění zadání pomocí galerie a přeskočení úkolu.

Ještě jednou důležitou obrazovkou je počítání barevnosti obrázku, kde bude zobrazena informace, jestli uživatel splnil úkol nebo ne. Náhled této obrazovky je možné vidět na obrázku 3.6c.

Nejsložitější pro zvládnutí bylo navrhnout obrazovku, která umožňuje filtrování podle jedné až tří barev. Složitost se spočívá v tom, že je možné až třikrát zvolit jednu z patnácti barev, což musí být pochopitelné a snadné pro uživatele. Navrhl jsem dvě varianty uživatelského rozhraní pro tuto obrazovku, které je možné vidět na obrázku 3.7.

### 3.6.1.3 Nedostatky prvního návrhu

Při detailnější analýze vytvořených wireframů je možné vidět, že některé jeho části nesplňují Android guidelines a nevyužívají nový materiální přístup, což bude opravené v následujících verzích wireframu.

Nejdůležitější nedostatky je možné odhalit v obou návrzích filtrovacího menu, které je skoro nepoužitelné. Zvolit potřebnou barvu ani ve vodorovném ani ve svislém filtrovacím menu není možné kvůli velikosti tlačítek reprezentujících barvy. Řešením tohoto problému může být vykreslování barevné palety jen jednou místo tří stejných komponent pro filtrování barev.

Další nedostatky je vidět v návrhu obrazovky pro uživatele. První nevýhoda je to, že na větších obrazovkách rozmístění tlačítek nebude vypadat stejně jako na malých. Další nedostatek se objeví po přechodu do stránky

Tabulka 3.2: Testování první verze uživatelského rozhraní pomocí Nielsenových heuristik

Viditelnost stavu systému	Nelze v této verzi určit
Propojení systému a reálného světa	Splňuje
Uživatelská kontrola a svoboda	Splňuje
Standardizace a konzistence	Nesplňuje (nedodržuje materiální design a obsahuje nestandardní prvky)
Prevence chyb	Nelze v této verzi určit
Rozpoznání namísto vzpomínání	Nesplňuje (chybí možnost filtrování podle barev z úkolu)
Flexibilní a efektivní použití	Splňuje (rozdělení lokálních a registrovaných uživatelů)
Estetický a minimalistický design	Splňuje (neobsahuje zbytečné věci)
Pomoc uživatelů poznat, pochopit a vzpamatovat se z chyb	Nelze v této verzi určit
Nápověda a návody	Nesplňuje (aplikace tohoto druhu musí být pochopitelná bez návodu)

jiného uživatele, kde zmizí tlačítka, a proto kroužek nebude vypadat harmonicky. Navíc ani tlačítka, ani kroužek nedodržují Android guidelines.

Na obrazovce se seznamem barevných úkolů první nedostatek, který je vidět na první pohled, je, že se příliš často opakují stejná tlačítka. Navíc úkol neobsahuje žádné informace o své bodové hodnotě a pořadovém čísle. Ještě jedním důležitým nedostatkem je to, že pokud uživatel chce vyfiltrovat obrázky podle barev z úkolu, tak si potřebuje zapamatovat barvy, přejít na jinou záložku a vyplnit filtrovací požadavek, což je příliš složité. Takže je potřeba přidat tlačítko pro filtrování přímo ze seznamu úkolů.

Společnou chybou pro všechny wireframy bylo to, že záložky pro přepínání mezi hlavními obrazovkami se nachází v dolní části obrazovky, což silně porušuje Android guidelines. Ve většině Android zařízení se dole nachází systémová tlačítka, což může způsobit to, že místo přechodu na jinou záložku uživatel zmáčkne jedno ze systémových tlačítek.

V tabulce 3.2 je uvedeno testování navrženého uživatelského rozhraní pomocí Nielsenových heuristik [31].

### 3.6.2 Druhý návrh uživatelského rozhraní

V druhém návrhu jsem se snažil opravit všechny chyby z předchozí verze a použít základy materiálního přístupu popsáno v následující sekci.

#### 3.6.2.1 Materiální design

V této sekci čerpám z [32].

Materiální design (anglicky material design) je nový předpis od Google ohledně toho, jak musí vypadat aplikace pro Android zařízení. Poprvé tento přístup byl představen v roce 2014 a od té doby většina populárních aplikací, jako jsou například YouTube, Gmail a Google Play přešly na tento nový vzhled. Tento přístup byl inspirován reálnými materiály, jako jsou inkoust a papír. Hlavní myšlenka tohoto přístupu se skrývá v použití stínů a animací pro dosažení efektu realistických objektů a znázornění jejich souvztažnosti. Jedním ze standardních zástupců materiálního designu může být seznam, kde každá položka je reprezentovaná samostatnou kartou se stínem, která se nachází na bílém pozadí. Pokud se zdá, že pozadí a položky seznamu jsou na různých úrovních, cíl je dosažen.

Součástí materiálního designu jsou doporučení ohledně animací jednotlivých prvků, vzhledu ikon a tlačítek, velikosti a stylu písma, barevné palety aplikace a dokonce délky a významu hlášek a dialogů.

#### 3.6.2.2 Tok obrazovek

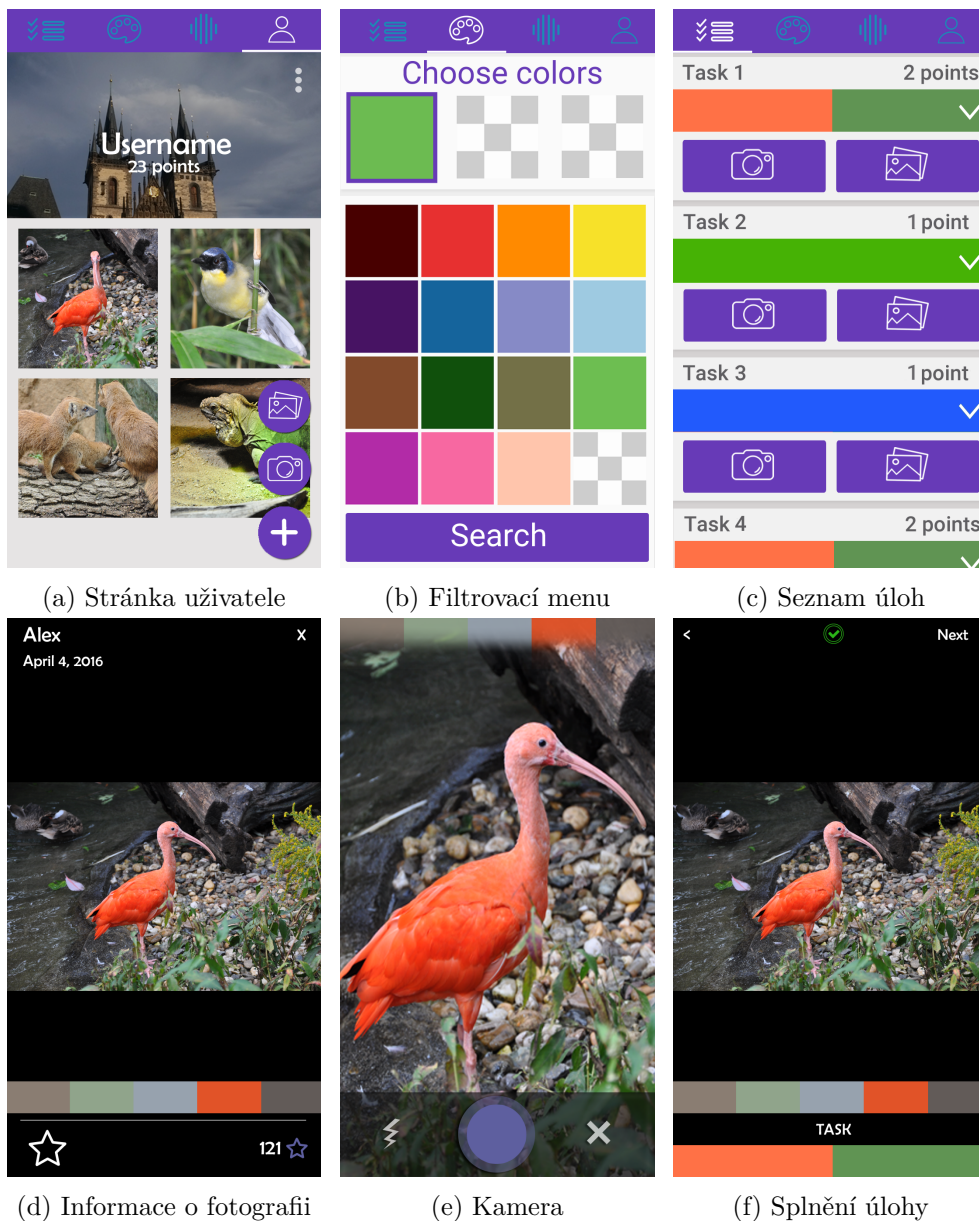
Hlavním účelem diagramu B.6 je znázornit přechody mezi jednotlivými obrazovkami. Počáteční obrazovky jsou zvýrazněny žlutou barvou. Aplikace zobrazí úvodní obrazovku pokud uživatel není přihlášen, jinak se zobrazí obrazovka se seznamem úloh. Ve středu diagramu se nachází Tab menu, které je klíčovým elementem v uživatelském rozhraní. Toto menu obsahuje čtyři nejdůležitější obrazovky, které byly navrženy v minulé verzi uživatelského rozhraní. Po prvním spuštění aplikace je zobrazena úvodní stránka, která obsahuje tlačítka pro registraci a přihlášení. Dalšími obrazovkami jsou stránka s informací o fotografii, stránka s vyfiltrovanými fotografiemi podle úkolů a podle požadavků.

#### 3.6.2.3 Vzhled druhého návrhu uživatelského rozhraní

Zásadní změny se dotkly stránky registrovaného uživatele, kterou je možné vidět na obrázku 3.8a. Tlačítka pro přidání fotografie jsou schovaná v prvku pod názvem Float Action Button, který je jedním z hlavních představitelů materiálního designu. Při zmáčknutí na toto plovoucí tlačítko se zobrazí dvě nová tlačítka, které reprezentují způsoby přidání fotografie. Bodové hodnocení uživatele se píše bílou barvou pod přezdívkou na ztmaveném úvodním obrázku, což zajistí viditelnost nápisu na libovolných fotografiích. Tento design zůstává harmonickým i po převedení do stránky lokálního uživatele. Jediné, co se přidá k této stránce, je tlačítko v dolní části obrazovky pro registraci na serveru, což může pomoci zvětšit počet registrovaných uživatelů.

Obrazovka s filtrováním 3.8b byla zcela předělána podle myšlenky se neopakující palety. Hlavním cílem bylo udělat rychlé a pohodlné filtrování podle





Obrázek 3.8: Druhý návrh jednotlivých obrazovek

jedné až tří barev. Paleta je zobrazena v mřížce o velikosti čtyři na čtyři čtverce, kde patnáct z nich jsou barvy z palety a jeden reprezentuje nezvolenou barvu. Nahoře jsou zobrazena tři tlačítka, která uživatel může obarvit. Po zvolení barvy přechod na následující tlačítko se provede automaticky, což dovolí zmenšit počet kliknutí pro vyplnění všech barev. Pro opravování chybné volby barvy je také dostupné manuální přepínání mezi tlačítky.

Obrazovka se seznamem úkolů 3.8c byla mírně upravena a převedena

### 3. NÁVRH

---

do vzhledu karet podle materiálního designu. Ke každé položce seznamu byla přidána informace o bodové hodnotě a pořadovém čísle. Také ke každému úkolu byla přidána ikona šipky dolů označující, že úkol je možné rozbalit. Po rozbalení úkolu se zobrazí seznam vyhovujících fotografií a v horní části pro pohodlnost je zafixována položka s úkolem. Toto dovolí uživateli se nevracet zpět a rovnou splnit úkol.

Ještě jednou důležitou obrazovkou je zobrazení informací o fotografii, kterou je možné vidět na obrázku 3.8d. Jelikož v aplikaci neexistuje vyhledávání podle přezdívky, přechod na stránku jiného uživatele je možné provést jen z této obrazovky. Další povinnou informací je zobrazení dominantních barev a data nahrání fotografie na server. V dolní části obrazovky se nachází tlačítko pro přidávání fotografie do oblíbených a informace o počtu přidání. Pro fotografie ze svého účtu je dostupné ještě jedno tlačítko pro odstranění fotografie.

Další neobvyklou obrazovkou je kamera, která je zobrazena na obrázku 3.8e. Kamera aplikace se liší od standardní kamery tím, že nahoře má barevný pruh, který ukazuje v reálném čase barevné složení fotografie. Kromě zapínání blesku, přiblížení a fotografování kamera nebude obsahovat žádné jiné možnosti.

Jedinou obrazovkou z minulé verze uživatelského rozhraní, které se nedotkly změny, je obrazovka splnění úkolu. Její novou podobu je možné vidět na obrázku 3.8f.

#### 3.6.2.4 Výhody a nevýhody druhého návrhu uživatelského rozhraní

V tabulce 3.3 je si možné prohlédnout výsledky testování druhého návrhu uživatelského rozhraní pomocí Nielsenových heuristik.

Výsledky testování se zlepšily hlavně díky opravě chyb z minulé verze a detailnějšímu návrhu. V průběhu návrhu bylo vyřešeno, že každá kritická operace, jako je nahrávání a odstranění fotografií, bude zkontrolována potvrzujícím dialogem. Dalším rozhodnutím bylo označit standardním prvkem Progress Bar všechny dlouhotrvající akce, což je například vypočítání barevnosti fotografie nebo libovolné operace se serverem.

Jedinou nevýhodou uživatelského rozhraní, která byla odhalena v průběhu testování na zařízení, se stalo rozmístění šipky pro rozbalení úkolu. Lepším řešením by bylo rozmístit ji pod úkolem, aby bylo jasnější, že úlohu je možné rozbalit.

Tabulka 3.3: Testování druhé verze uživatelského rozhraní pomocí Nielsenových heuristik

Viditelnost stavu systému	Splňuje (pomocí prvku Progress Bar)
Propojení systému a reálného světa	Splňuje
Uživatelská kontrola a svoboda	Splňuje
Standardizace a konzistence	Splňuje (všude, kde je to možné, jsou dodrženy standardy)
Prevence chyb	Splňuje (dialogy pro nahrávání a odstranění fotografií)
Rozpoznání namísto vzpomínání	Splňuje
Flexibilní a efektivní použití	Splňuje (rozdělení na lokální a registrované uživatele)
Estetický a minimalistický design	Splňuje (neobsahuje zbytečné věci)
Pomoc uživatelů poznat, pochopit a vzpamatovat se z chyb	Nelze v této verzi určit
Nápověda a návody	Nesplňuje (aplikace tohoto druhu musí být pochopitelná bez návodu)



---

# Implementace prototypu

## 4.1 Klient

### 4.1.1 Android Studio a Android SDK

Android Studio je vývojové prostředí od Google, které je přímo určené pro vytváření aplikací pro operační systém Android. Toto prostředí je založené na IntelliJ IDEA, což je výkonný nástroj pro programování v jazyce Java [35]. Na rozdíl od Eclipse Android Studio neobsahuje zbytečné funkčnosti, jeho instalace je rychlejší a snadnější a samo prostředí se považuje za nejlepší pro vyvíjení mobilních aplikací pro Android.

Vytváření aplikací v Javě není jediná možnost vyvíjení pro Android. Existují nástroje a prostředí pro multiplatformní vývoj, což znamená, že vytvořený produkt bude dostupný nejen pro platformu Android, ale i pro iOS, Windows Phone a i dokonce jako webová stránka. Jedním z příkladů může být prostředí Xamarin, ve kterém se píše v jazyce C# nebo Ruby. Hlavním nedostatkem multiplatformního vývoje je nejistota v kvalitě výsledku. Jelikož při multiplatformním vývoji se nezabývají konkrétní platformou, aplikace může vypadat jako cizí pro tento operační systém. Navíc práce se zabudovanými prostředky jako kamera je složitější. Proto volba nativního vývoje pro operační systém Android v Android Studiu je logická a promyšlená [34].

Hlavní součástí instalace Android Studia je Android SDK, což je sada nástrojů, které jsou nezbytné pro vývoj aplikace pro operační systém Android. Dále následuje popis nástrojů, které jsem používal nejčastěji při vývoji aplikace „Colorbis“ [36].

- **Android Virtual Device Manager (zkráceně AVD Manager)**

Android Virtual Device Manager je nástroj pro vytváření virtuálních Android zařízení, které jsou spustitelné v Android Emulatoru.

- **Android Emulator**

Emulátor založený na QEMU<sup>4</sup>, který je možné používat pro spuštění a testování svých aplikací na požadovaném zařízení.

- **mksdcard**

Nástroj umožňující simulaci externího úložiště v emulátoru. Používá se pro práci s obrázky nebo jinými daty v emulátoru.

- **SDK Manager**

Umožňuje spravování SDK balíčků, jako jsou například nainstalované platformy.

- **Android Monitor**

Nástroj umožňující monitorování využití paměti, CPU, GPU pro účely testování a optimalizace.

- **Android Debug Bridge (zkráceně adb)**

Nástroj, který pomocí příkazové řádky umožňuje komunikaci s emulátorem nebo připojeným zařízením.

### 4.1.2 Android Support Libraries

V této části čerpám z [37] a [38].

Sada podpůrných knihoven je určena pro poskytování zpětné kompatibility nových verzí API pro starší verze Androidu. To znamená, že aplikace může používat novinky zavedené od poslední verze API, ale stále bude kompatibilní se zařízeními se starší verzí operačního systému Android.

Používání podpůrných knihoven je považováno za doporučený postup při programování pro platformu Android. Toto může pomoci zvýšit výkon aplikace, zlepšit vzhled a samozřejmě zpřístupní aplikaci více uživatelům.

Každá podpůrná knihovna je zaměřena na určité Android API a obsahuje jinou sadu prvků. Proto je velmi důležité si zvážit, kterou funkci je potřeba podporovat v aplikaci a zvolit potřebné knihovny.

- **v4 Support Library**

Podpůrná knihovna v4 je největší ze všech a byla vytvořena pro použití od Androidu verze 1.6 a výše. Její součástí jsou nejpopulárnější API včetně podpory nových komponent, prvků uživatelského rozhraní, zpra-

---

<sup>4</sup>QEMU je generický emulátor s otevřeným zdrojovým kódem.

cování dat a jiných. Příklady použitých mnou prvků knihovny mohou být: Fragment<sup>5</sup>, Loader<sup>6</sup>, ViewPager<sup>7</sup> a jiné.

- **v7 Support Libraries**

Tato knihovna byla vytvořena pro použití od Androidu verze 2.1 a poskytuje sadu prvků, které mohou být nezávisle zasazeny do projektu. Příklady těchto prvků použitých v aplikaci mohou být:

1. **v7 appcompat library**

Přidá třídu implementující Activity, kterou je možné používat jako základní třídu pro vytváření aktivit. Používá se pro lepší práci s prvkem Action Bar.

2. **v7 cardview library**

Přidá podporu pro zobrazování CardView, což je jeden ze základních prvků materiálního designu.

- **Design Support Library**

Balíček těchto knihoven poskytuje podporu pro všechny základní prvky uživatelského rozhraní v materiálním designu, jako jsou Floating Action Buttons, Tabs a jiné.

### 4.1.3 Jiné použité knihovny

- **Android Universal Image Loader**

Android Universal Image Loader poskytuje velmi flexibilní nástroj pro načítání, ukládání do mezipaměti a zobrazování obrázku a jeho součástí je velké množství nastavení a konfigurací pro zajištění kontroly nad těmito procesy [39].

- **Android Parallax Recycle View**

Adaptér, který je možné použít pro docílení parallax efektu na RecyclerView [40].

- **Android NDK**

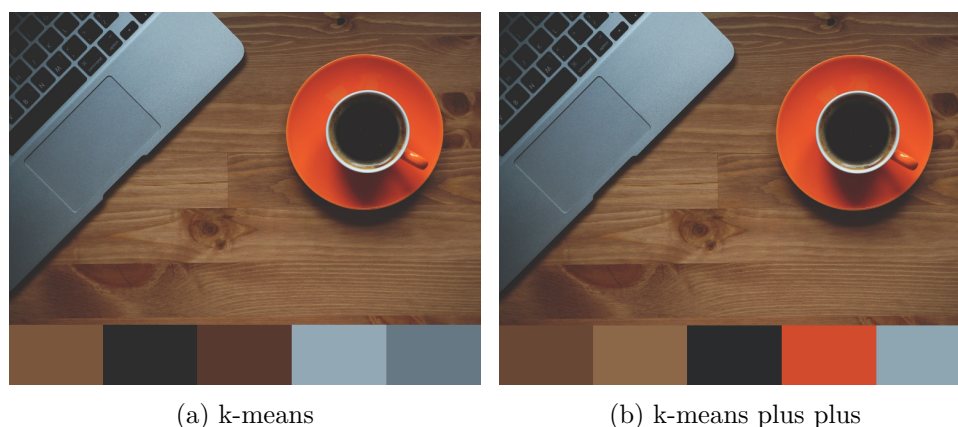
Android NDK je nástroj, který umožňuje implementovat nějaké části aplikace v C++ nebo v C. Toto dovolí zrychlit obtížné výpočty a provádění složitých algoritmů. V aplikaci „Colorbis“ Android NDK se používá pro vypočítání barevnosti fotografie a nalezení vzdálenosti mezi barvami [41].

---

<sup>5</sup>Fragment z podpůrné knihovny přidává podporu zapouzdření uživatelského rozhraní a funkčnosti ve fragmenty.

<sup>6</sup>Loader z podpůrné knihovny přidává podporu asynchronního načítání dat.

<sup>7</sup>ViewPager z podpůrné knihovny přidává podporu pro zobrazování přechodu mezi jednotlivými pohledy.



Obrázek 4.1: Porovnání výsledků algoritmů k-means a k-means++

#### 4.1.4 Implementace algoritmů pro nalezení dominantních barev

Jelikož hlavní činností aplikace „Colorbis“ je nalezení dominantních barev, je důležité správně implementovat algoritmy pro analýzu barevnosti fotografie, které byly popsány v kapitole 2.6.

První z algoritmů byl implementován k-means, který podle analýzy nemusí dávat správné výsledky kvůli náhodné inicializaci středů. Po implementaci jsem se přesvědčil, že pro některé obrázky algoritmus nemůže najít barvy, které by měly patřit do palety dominantních barev. Tento nedostatek je vidět na obrázku 4.1a, kde mezi dominantními barvami není oranžová barva podšálku. Navíc při několikanásobném použití tohoto algoritmu pro stejný obrázek vždy dostanu různé výsledky, což není pro navrženou aplikaci přípustné.

Implementace k-means++ byla snadnější, protože tento algoritmus je založen na obyčejném, již implementovaném k-means. Výsledky dávané tímto algoritmem jsou znatelně lepší než výsledky předchozího algoritmu. Například na obrázku 4.1b je vidět, že k-means++ našel oranžovou barvu šálku oproti k-means, který měl místo této barvy ještě jednu šedou.

Jelikož původní implementace těchto algoritmů v Javě byla velmi pomalá a zpracování některých obrázků mohlo trvat až do minuty, rozhodl jsem použít nástroj Android NDK, který umožňuje implementaci algoritmu v jazyce C (více v kapitole 4.1.3). To pomohlo zrychlit nalezení dominantních barev v obrázku a velmi zkrátilo čas zpracování.

Další problém, který jsem potřeboval vyřešit, bylo to, že v některých případech k-means++ ve výsledku dával několik velmi podobných barev a vynechával z obrázku barvy důležité pro lidské oko. To se stávalo, protože ražení barev se provádí od barvy, která je v obrázku nejdominantnější, a vypisuje se jen pět prvních barev. Například v obrázku, který obsahuje pět odstínů zelené a jen trochu jiné barvy, se mohlo stát, že tato důležitá jiná barva se



ve výsledku nevypíše. Pro odstranění tohoto problému jsem se rozhodl smíchat podobné barvy, kde Euklidova vzdálenost mezi nimi je menší než 40. Použití právě Euklidovy vzdálenosti je v tomto kroku logické, protože využití složitějšího algoritmu (například Delta E 96 a Delta E 2000) by způsobil zpomalení zpracování obrázku. Navíc nalezení této vzdálenosti není tak důležité jako při splnění úkolu a není viditelné pro uživatele. Toto rozhodnutí mělo za následek nutnost hledání více jak pěti barev, protože většina z nich mohla být smíchána. Ve výsledném řešení se hledá deset barev, což zvýšilo řádově kvalitu nalezených výsledků, ale jako následek nepatrně zpomalilo zpracování obrázku.

#### 4.1.5 Volba optimální vzdálenosti mezi barvami

Výpočet vzdálenosti mezi barvami se provádí pomocí formule Delta E 2000 (více v kapitole 2.7), kde ale nastává problém s volbou správné hranice. Pokud bude vzdálenost menší než zvolená hodnota, aplikace bude považovat barvy za podobné a úkol bude splněn. Experimentálně byla zvolena hodnota 17. Jelikož tato formule udává, že barvy jsou pro lidské oko podobné při hodnotě menší než 1, bylo velmi těžké najít vyhovující řešení. Se zvětšením hranice by splnění úkolů bylo příliš snadné a výsledky filtrování podle barev by nebyly správné, a naopak se zmenšením dané hodnoty by splnění úkolu vyžadovalo velké úsilí. V tabulce 4.1 je znázorněno, že optimální řešení tohoto problému neexistuje a pro některé dvojice barev je tato hodnota vhodná, zatímco pro některé není.

#### 4.1.6 Přístup k databázi














Pro implementaci klientské databáze, která obsahuje informace o lokálních uživatelích a jejich úlohách, byla zvolena databáze SQLite.

Pro přístup do databáze byly použity třídy `ContentProvider` a `CursorLoader`. První se stará o centralizaci obsahu a slouží především k poskytování obsahu jiným aplikacím. V případě aplikace „Colorbis“ poskytovatel obsahu nebyl exportován, což znamená, že není viditelný pro jiné aplikace. Poskytovatel obsahu je implementován jako třída zděděná od třídy `ContentProvider`, která má metody výběru, vkládání, mazání a aktualizace dat v databázi.

Přístup k datům je umožněn pomocí URI (anglicky Uniform Resource Identifier, což je jednotný identifikátor zdroje). Například seznam všech uživatelů se nachází v URI `content://authority/users` a informace o uživateli s identifikačním číslem 23 v URI `content://authority/users/23`.

Pomocí třídy `ContentResolver` je pak umožněn přístup k obsahu, který poskytuje implementovaný `ContentProvider`. `CursorLoader` se dotazuje poskytovatele obsahu a vrací výsledky pro Aktivitu nebo Fragmenty. Tato třída má výhodu v tom, že dotazy probíhají asynchronně a jsou automaticky obnovené v případě změny dat.

Tabulka 4.1: Vzdálenosti mezi barvami podle formule Delta E 2000

První barva	Druhá barva	Vzdálenost
		13,5399
		14,3846
		15,2483
		16,0394
		16,7043
		16,8445
		17,3026
		18,2636
		20,8812

## 4.2 Server

### 4.2.1 Nástroje a knihovny

Serverová část byla napsána v JavaScriptu pomocí node.js a několika dalších pomocných nástrojů a frameworků.

- **Express**

Pro snadnou práci s HTTP požadavky byl zvolen Express framework. Tento framework umožňuje snadné nastavení middleware pro zpracování HTTP požadavků [45].

- **sqlite3**

Pro řešení perzistentního ukládání dat byl zvolen framework sqlite3. Tento nástroj umožňuje asynchronní a neblokující přístup k SQLite databázi [46].

- **Multer**

Pro nahrání a uložení fotografií byl vybrán middleware Multer. Tento middleware slouží pro obsluhu multipart/form-data, což se používá pro nahrávání souboru na server [47].

- **imagemagick**

Pro efektivní zobrazení seznamu fotografií se na serveru vytváří zmenšené kopie všech nahraných fotografií. Modul imagemagick se stará o zmenšení původní fotografie a vytvoření její náhledu [48].

- **ursa**

Modul ursa nabízí jednoduché použití asynchronního šifrování pomocí veřejného a privátního klíče z knihovny OpenSSL. Pomocí tohoto modulu se na serveru provádí šifrování a dešifrování hesla uživatele [49].

- **crypto**

Pro vypočítání kódu HMAC na straně serveru byl zvolen modul crypto [50].

### 4.2.2 Bezpečnost

V této části jsou popsány bezpečnostní prvky, které byly implementovány na serveru pro zajištění bezpečného přístupu k účtu uživatele a bezpečné komunikace se serverem. Tato sekce byla napsána pomocí [42], [43] a [44].

Pro registraci na serveru a přihlášení musí uživatel uvést elektronickou poštu jako identifikátor a heslo pro autentizaci a autorizaci. V databázi je heslo uživatele uloženo nikoliv v otevřené podobě, ale jako osolené jeho hashe. Metoda solení zajistí, že ani na serveru nebude uloženo skutečné heslo, což je dobrým zvykem. Jelikož heslo uživatele je nejcitlivější část ze všech dat, je potřeba ho při registraci a přihlášení před odesláním zašifrovat pomocí veřejného klíče serveru. Veřejné a privátní klíče se používají pro bezpečnou komunikaci mezi subjekty. Šifrování dat veřejným klíčem serveru znamená, že jen server, který vlastní privátní klíč, může data dešifrovat. Při registraci server uloží dešifrované hashe hesla do databáze pro porovnání s přijatým hashem při přihlášení.

Dalším nebezpečím bylo to, že se pro komunikaci používají HTTP požadavky, které mohou být v průběhu přečtené nebo změněné. Tento problém se nazývá „člověk uprostřed“ (anglicky man in the middle attack). Jako příklad je možné uvést to, že nahrání fotografie se provádí pomocí POST požadavku, což znamená, že v těle požadavku se nachází data fotografie. Pokud útočník

odposlouchává komunikaci mezi aplikací a serverem, tak může jednoduše změnit tělo požadavku a tím pádem odeslat na server zcela jinou fotografii. Tento problém se dá vyřešit pomocí přidání hlavičky s autentizačním kódem zprávy do POST požadavku.

Pro generování hlavičky je možné použít HMAC, který slouží k ověření datové integrity a autentizaci. Generování HMAC kódu jen z těla požadavku není přínosné, a proto je k tělu požadavku potřeba přidat hashe hesla uživatele, protože tato informace je známá pouze aplikaci a serveru. Na serveru se také vypočítá HMAC kód a porovná se s přijatou hlavičkou. Pokud jsou stejné, tak přenos nebyl narušen a tudíž je platný. Kromě toho požadavek musí mít hlavičku s identifikátorem uživatele, ale nesmí obsahovat hashe hesla.

Existuje ještě problém, který je částečně spojený s problémem „člověka uprostřed“. Útočník může jednoduše přeposlat požadavek, který odposlouchal v komunikaci. Tento problém má název „útok přehráním“ (anglicky replay attack). Tudíž se může stát, že uživatel bude mít stovky stejných fotografií na účtu místo jedné. Zároveň budou duplikované fotografie zabírat místo na serveru. Existuje několik přístupů k řešení tohoto problému. První možností je použít časovou značku, což znamená, že server nebude přijímat relativně staré požadavky. Tento způsob ale vyžaduje přesnou časovou synchronizaci mezi aplikací a serverem. Druhá možnost je použít takzvaný „number used only once“ (zkráceně nonce), což je klíč, který je možné použít jen jednou. Aplikace požádá server o získání tohoto klíče a přidá ho do těla s heslem. Po provedení všech operací bude HMAC kód vypadat následujícím způsobem:

$$\text{HMAC kód} = \text{tělo} + \text{heslo} + \text{nonce}.$$

Aplikace také přidá nonce jako jednu z hlaviček, a to proto, aby server věděl, jak rekonstruovat HMAC kód. Server uchovává vygenerované nonce v seznamu a po zpracování požadavku se známým nonce ho odstraní.

Požadavky na získání dat, jako jsou obdržení seznamu fotografií, informací o uživateli nebo fotografii, nejsou zabezpečené. Naopak požadavky, které provádí změny v databázi, registraci, přihlášení, přidání a odebrání fotografie, přidání do oblíbených a odebrání z oblíbených položek jsou zabezpečené výše popsaným způsobem.

---

# Testování prototypu

## 5.1 Testování programátorem

Implementovaný prototyp aplikace byl pečlivě otestován na dvou testovacích zařízeních:

- LG Optimus G E975, Android 4.4.2, 2 GB operační paměti, 4 jádra procesoru, rozlišení displeje 1280 x 768 obrazových bodů.
- Asus EEE Pad Transformer TF700T, Android 4.2.1, 2 GB operační paměti, 4 jádra procesoru, rozlišení displeje 1920 x 1200 obrazových bodů.

V průběhu testování programátorem byly odhaleny některé kritické chyby jak v serverové části, tak i v klientské. Popis nejzajímavějších z nich je možné si přečíst v následujících podkapitolách.

### 5.1.1 Obsluha více klientů serverem

Tato chyba se vyskytovala, pokud se ve stejný okamžik několik uživatelů rozhodlo nahrát fotografie na server. Chyba mohla nastat jak v případě nahrání fotografie po splnění úkolu, tak v případě přidání fotografie na vlastní účet. Podle návrhu serverové databáze má každá fotografie vlastní identifikátor, což je celé číslo, které se postupně zvětšuje. Data fotografie nejsou uložena do databáze, ale do veřejné složky na serveru a soubor má stejné jméno, jako klíč odpovídající fotografii. Server zpracovává požadavky asynchronně pro umožňování obsluhy několika uživatelů. Na začátku je potřeba uložit fotografii do složky, která má jméno největšího klíče v tabulce zvětšené o jedničku. Dále server uloží náhled fotografie se stejným jménem do složky s náhledy. Posledním krokem je přidání záznamu do databáze včetně barevnosti fotografie.

Každá akce se provádí po ukončení té předchozí. Výše popsáný způsob funguje správně, pokud jiný uživatel nebude nahrávat svoji fotografii. Například tato chyba nastávala, pokud v průběhu zpracování první akce server dostal

ještě jeden požadavek na nahrání fotografie a začal ukládat jinou fotografii se stejným názvem jako předchozí. Jako důsledek mohly být fotografie přiřazeny stejnému uživateli nebo mohly mít špatné barvy. Řešením bylo inkrementovat klíč tabulky s fotografiemi ještě před první akcí a nadále používat tento klíč pro vkládání záznamu do tabulky a uložení náhledu. Tato chyba byla těžko odhalitelná, protože se obvykle testuje činnost jednoho uživatele.

### 5.1.2 Nedostatek paměti

Jednalo se o běžnou chybu, která byla způsobena nesprávnou prací s pamětí na zařízeních s operačním systémem Android. U mobilních zařízení je to složitější než u počítačů, protože nemají velké množství operační paměti a systém může proces ukončit, pokud začíná vyžadovat více, než systém dovoluje. Po nastudování potřebných materiálů [51] byl způsob zobrazení fotografií byl předělán a tím pádem už nedocházelo k překročení paměťového limitu. Tato metoda spočívá v načítání menších verzí fotografií do paměti, například načítání fotografie s rozlišením 2048 krát 1536 obrazových bodů je paměťové nevhodné, pokud bude zobrazena na obrazovce 1280 krát 720. Pro načítání menší fotografie slouží proměnná `inSampleSize`. Také pro zobrazení v seznamu se na serveru vytváří náhledy fotografií, které pomůžou ušetřit datový přenos.

## 5.2 Testování použitelnosti

Testování použitelnosti (anglicky usability testing) se používá pro vyhodnocení použitelnosti aplikace nebo programu. Testování se provádí několika uživateli reprezentujícími cílovou skupinu, kteří mají za úkol provést testovací scénáře. Většinou je průběh testování pozorován člověkem, který popisuje všechny vzniklé složitosti. Cílem tohoto testování je identifikovat problémy spojené s použitelností a navrženým uživatelským rozhraním a určit spokojenost uživatelů s testovaným produktem.

### 5.2.1 Průběh testování použitelnosti

Průběh testování použitelnosti se skládá z vyplnění pre-test dotazníku respondentem, splnění testovacích úkolů a zodpovězení otázek z post-test dotazníku. Testování probíhalo v reálném prostředí (kancelář nebo byt), nikoliv v laboratoři. Testování se zúčastnilo pět potenciálních uživatelů aplikace, kdy každý měl vlastní mobilní zařízení s operačním systémem Android.

#### 5.2.1.1 Pre-test dotazník

1. Jaká je Vaše věková kategorie?

Možné odpovědi: 11-18, 18-25, 25-35, 35-50.

2. Jak byste ohodnotil své zkušenosti s použitím sociálních sítí na mobilních zařízeních?

Možné odpovědi: používám každý den, používám několikrát týdně, někdy jsem to použil/použila, nikdy jsem to nepoužil/nepoužila.

3. Je fotografování Vaším koníčkem?

Možné odpovědi: ano, trochu, ne.

4. Jak byste ohodnotil své zkušenosti s operačním systémem Android?

Možné odpovědi: pokročilý uživatel, běžný uživatel, vlastním telefon jen pro volání.

### 5.2.1.2 Úkoly

- **Vytvoření lokálního účtu.** Představte si, že nejste milovník používání sociálních sítí, ale chce se Vám použít novou aplikaci „Colorbis“, prohlížet fotografie jiných uživatelů a rozvíjet kreativitu bez registrace na serveru. Vytvořte účet lokálního uživatele zadáním přezdívky.
- **Splnění úkolu.** Chcete rozvíjet svou kreativitu a hledat ve světě barvy. Splňte libovolný z dostupných úkolů pomocí svých fotografií z galerie.
- **Filtrování fotografií podle dvou barev.** Představte si, že jste malíř a potřebujete na zakázku namalovat barevný obraz, ale chybí Vám inspirace. Vyfiltrujte fotografie podle dvou libovolných barev.
- **Zobrazení seznamu naposledy přidáných fotografií a přechod na stránku uživatele.** Zobrazte seznam naposledy přidáných fotografií a otevřete libovolný obrázek na celou obrazovku. Pak přejděte na stránku uživatele, který nahrál tento obrázek.
- **Registrace na serveru.** Představte si, že teď chcete sdílet fotografie a proto potřebujete založit účet. Zaregistrujte se na serveru.
- **Přidání fotografie na účet.** Představte si, že se chcete podělit o své fotografie bez splnění úkolů. Přidejte libovolnou fotografii z galerie na svůj účet

### 5.2.1.3 Post-test dotazník

1. Bylo používání aplikace bez registrace jednoduché a pochopitelné?
2. Byl průběh registrace na serveru pochopitelný?
3. Bylo jednoduché splnit barevný úkol?
4. Je filtrování obrázků podle jedné až tří barev pochopitelné?

5. Bylo jednoduché přidat fotografii na svůj účet po registraci?
6. Jaký je Váš celkový dojem z používání aplikace? (ohodnoťte prosím od 1 do 10)
7. Jaký je Váš dojem z myšlenky aplikace? (ohodnoťte prosím od 1 do 10)

### 5.2.2 Výsledky testování

Celkem testování proběhlo bez závažných problémů, ale objevily se drobné chyby v návrhu uživatelského rozhraní a návrhu funkčnosti.

První problém, se kterým se setkali někteří uživatelé, bylo nepochopení toho, jak je možné používat aplikaci bez registrace na serveru. Jeden z uživatelů si nevyšiml tlačítka „Přeskočit registraci“ a při vytváření lokálního účtu rovnou vyplnil elektronickou postu a heslo, čímž se zaregistroval na serveru. Tento problém byl zřejmý ještě na začátku návrhu, ale jelikož jednou z myšlenek aplikace je možnost použití hlavních funkcí bez registrace, nedalo se tomu vyhnout. Také si jeden uživatel všiml, že je velmi obtížné trefit šípku pod úkolem, která otevře fotografie vyfiltrované podle barev zadání.

Dalším důležitým testovacím scénářem bylo splnění úkolů, což je hlavní funkčnost této aplikace. Jen jeden ze všech uživatelů zvolil splnění dvoubarevného úkolu, ostatní se rozhodli splnit lehčí jednobarevný úkol. Tři uživatelé z pěti splnili barevný úkol bez problémů na první pokus. Jeden uživatel měl problém s tím, že si myslel, že žlutá barva a světle šedá budou podobné a aplikace označí úkol jako splněný. Ovšem tento uživatel splnil tento barevný úkol na druhý pokus bez problémů. Uživatel, který zvolil dvoubarevný úkol, splnil zadání až na třetí pokus, přičemž v prvním pokusu se jednalo o omyl uživatele a ve druhém algoritmus určil, že barvy z obrázku a požadované barvy jsou různé, i když se mohlo zdát, že jsou podobné (tento problém je popsán v kapitole 4.1.5). Na třetí pokus uživatel zvládl úkol bez problémů.

Filtrování fotografií proběhlo bez problémů a uživatelé si hráli s různými barvami. Jeden uživatel si všiml, že se občas ve vyfiltrovaných fotografiích vyskytují méně vhodné obrázky, kde je velmi málo požadované barvy. To je způsobené tím, že vyfiltrované obrázky vždy musí vyhovovat zadání s odpovídajícími barvami.

Registrace na serveru také většinou proběhla dobře. Čtyři z pěti uživatelů se rychle zaregistrovali pomocí tlačítka v dolní části obrazovky s lokálním účtem. Jeden uživatel dal přednost tomu, že se odhlásil a pak se zaregistroval s novou přezdívkou. Ovšem pak jsem se dozvěděl, že to udělal speciálně proto, že chtěl změnit přezdívku.

Přidání fotografie na účet po registraci zvládl každý uživatel a většina uživatelů označila tento úkol jako nejsnadnější ze všech.

Přehled všech odpovědí na pre-test a post-test dotazník je možné si prohlédnout v příloze E.



## Implementace výsledného řešení

### 6.1 Kamera

Nejtěžším úkolem v navržené aplikaci bylo implementovat vlastní modul pro práci s kamerou. Problém spočíval v tom, že každé zařízení má vlastní kameru, která se liší od ostatních, a proto nebylo snadné ošetřit všechny podmínky. Vlastní modul pro práci s kamerou musí umožňovat zobrazení barevného složení každého snímku z kamery v reálném čase, což udělá splnění barevných úkolů pohodlnějším a zlepší celkový dojem z aplikace. Velkým překvapením bylo to, že při implementaci nebyl problém s rychlostí zpracování obrázků v reálném čase, ale vyskytl se velký problém s kompatibilitou se všemi zařízeními.

#### 6.1.1 Vlastní implementace

Nejprve byl implementován vlastní modul pro práci s kamerou bez možnosti přiblížení, který fungoval bezchybně na testovacích zařízeních. Také byla implementována analýza barevnosti snímku v reálném čase.

Pro implementaci modulu pro práci s kamerou je nutné přidat do manifest souboru aplikace povolení k zápisu do externí paměti a záznam o tom, že aplikace vyžaduje kameru. Dále byla vytvořena třída `CameraPreview`, která je zděděná od `SurfaceView` a slouží pro veškerou práci s kamerou. `SurfaceView` se používá v případě, že je potřeba častěji aktualizovat uživatelské rozhraní, například pro vykreslení snímku z kamery. Instance této třídy byla umístěna do `CameraActivity`, která byla spuštěna při požadavku na zapnutí kamery. Ve třídě `CameraPreview` byly implementované následující metody zděděné třídy:

- **surfaceCreated()** V dané metodě probíhá otevření kamery a nastavení příslušných callbacků.
- **surfaceChanged()** V dané metodě se nastavují parametry kamery a správné rozměry náhledu zobrazení snímku.

- `surfaceDestroyed()` Ukončení zobrazení náhledu a zastavení kamery.

Při implementaci kamery je potřeba nastavit rozměry náhledu a také šířku a výšku pořízené fotografie. Každá kamera má vlastní seznam podporovaných velikostí náhledu a fotografií. Získat tyto seznamy je možné pomocí metod parametrů kamery `getSupportedPreviewSizes()` a `getSupportedPictureSizes()`. Potom je potřeba ze seznamu podporovaných velikostí náhledu zvolit velikost s nejmenší odchylkou od poměru stran `SurfaceView`, do kterého se budou vykreslovat snímky. Stejným způsobem se zvolí nejlepší rozměr fotografie ze seznamu podporovaných velikostí fotografií. Zvolené velikosti je možné nastavit pomocí metod `setPreviewSize()` a `setPictureSize()`. Parametry kamery ještě mají užitečné metody `setFocusMode()` a `setFlashMode()`, které nastavují způsob zaostření a režim blesku. Po zvolení potřebných parametrů je potřeba je nastavit pomocí metody kamery `setParameters()`, protože jinak se změny neprojeví. Také je nutné nastavit orientaci kamery pomocí metody `setDisplayOrientation()`, aby výsledná fotografie byla správně otočená.

Pořízení fotografie se provádí pomocí zavolání metody `takePicture()` obsahující parametr `PictureCallback` s metodou `onPictureTaken()`, která bude zavolána po pořízení snímku. Tato metoda slouží pro zvolení cesty pro ukládání fotografie, nastavení názvu a samotného uložení do paměti zařízení.

Nevýhodou ale bylo to, že implementovaný modul byl použitelný jen na několika zařízeních, po spuštění na jiných zařízeních kamera nemusela fungovat nebo fotografie měly špatnou kvalitu. Proto bylo zvoleno řešení implementovat modul pro práci s kamerou pomocí knihovny.

### 6.1.2 Implementace pomocí knihovny

Z několika různých knihoven byla vybrána nejpoužívanější knihovna CWAC-Cam2 [52]. Zvolená knihovna obsahuje rozsáhlý demonstrační projekt, který zahrnuje všechny možnosti této knihovny. Kamera se spustí pomocí intentu třídy `CameraActivity`, který se vytvoří podle vzoru `IntentBuilder`. Tento přístup dovolí jednoduchou úpravu implicitních parametrů kamery, například metoda `facing()` indikuje, jakou kameru je potřeba zapnout, metoda `quality()` nastaví kvalitu výsledné fotografie, nastavení režimu zaostření, blesku a přiblížení umožňují metody `focusMode()`, `flashMode()` a `zoomStyle()`. Po pořízení fotografie je aplikace zodpovědná za aktualizace `MediaStore` a za to, aby se výsledná fotografie ukázala v galerii. V knihovně je implementována daná funkčnost, ale implicitně je vypnuta a musí být zapnuta pomocí metody `updateMediaStore()` při vytváření intentu. Jednou z nejdůležitějších metod `IntentBuilder` je metoda `to()`, která má jako parametr objekt `File`, do kterého bude uložena pořízená fotografie. Také v použité verzi knihovny chyběla funkčnost dynamické změny režimu blesku, i když je dostatečně běžná a populární pro většinu aplikací. Proto tato funkčnost byla přidána do zdrojových kódů knihovny.

---

## Závěr

Cílem této bakalářské práce bylo navrhnout a vytvořit aplikaci pro operační systém Android, která pomůže rozvíjet kreativitu uživatelů pomocí pořízení a sdílení barevných fotografií. Výsledkem je plně funkční a stabilní aplikace, která byla umístěna na oficiální obchod Google Play a splňuje požadavky na ni kladené.

Za prvé byla provedena analýza podobných aplikací, která pomohla se vyvarovat zbytečným chybám v návrhu. Další důležitou částí byl průzkum cílové skupiny uživatelů, na základě čehož byly definovány funkční a nefunkční požadavky. Následně byl vytvořen návrh, který kompletně popisuje funkčnost a vzhled aplikace. Další část bakalářské práce byla věnována implementaci prototypu aplikace, která se skládala z vývoje klienta a serveru. Vytvořený prototyp byl podroben důkladnému testování a nalezené chyby byly už opraveny v implementaci výsledného řešení, která navíc ještě zahrnuje vytvoření vlastního modulu pro práci s kamerou.

Výsledná aplikace v současné době nevykazuje v provozu žádné kritické chyby a je pozitivně oceněna uživateli, kteří jsou s ní velmi spokojeni. Tato aplikace skutečně pomáhá vidět ve světě barvy a přináší lidem mnoho zábavy. Jediným nedostatkem v dané chvíli je to, že tento typ aplikací, který je podobný sociální síti, vyžaduje velké množství uživatelů a nahraných fotografií pro pohodlnější používání.



---

## Literatura

- [1] INSTAGRAM. *Instagram* [software]. [přístup 6. prosince 2015]. Dostupné z: <https://play.google.com/store/apps/details?id=com.instagram.android&hl=cs>
- [2] WeRSM. *The Complete History of Instagram* [online]. 2014 [cit. 2015-12-08]. Dostupné z: <http://wersm.com/the-complete-history-of-instagram/>
- [3] BŘEZINA, Petr. 5 nejpopulárnějších fotografií na Instagramu. *DIGIarena* [online]. 10. října 2015 [cit. 2015-12-15]. Dostupné z: <http://digiarena.e15.cz/5-nejpopularnejsich-fotografii-na-instagramu>
- [4] YAHOO. *Flickr* [software]. [přístup 6. prosince 2015]. Dostupné z: <https://play.google.com/store/apps/details?id=com.yahoo.mobile.client.android.flickr&hl=cs>
- [5] EYEEM MOBILE. *EyeEm - Camera & Photo Filter* [software]. [přístup 6. prosince 2015]. Dostupné z: <https://play.google.com/store/apps/details?id=com.baseapp.eyeem&hl=cs>
- [6] PHOTOBUCKET CORPORATION. *Photobucket* [software]. [přístup 6. prosince 2015]. Dostupné z: <https://itunes.apple.com/us/app/photobucket-backup-print-shop/id314439840?mt=8>
- [7] SNAPR LTD. *Snapr* [software]. [přístup 8. prosince 2015]. Dostupné z: <https://itunes.apple.com/us/app/snapr/id359528116?mt=8>
- [8] SNAPR LTD. *Snapr. Google Play* [online]. [cit. 2015-12-20]. Dostupné z: <https://play.google.com/store/apps/details?id=pr.sna.snapr>
- [9] BOLTON, David. A Scoring Comparison of Android and iOS Development. *The New Stack* [online]. 2015 [cit. 2015-12-06]. Dostupné z: <http://thenewstack.io/scoring-comparison-android-ios-development/>

- [10] OGBO, Obaro. 7 reasons why you should develop apps for Android rather than iOS. *Android Authority* [online]. 2015 [cit. 2015-11-30]. Dostupné z: <http://www.androidauthority.com/develop-apps-for-android-rather-than-ios-607219/>
- [11] Google Inc. Android Developers: *Dashboard* [online]. [cit. 2016-02-11]. Dostupné z: <http://developer.android.com/about/dashboards/index.html>
- [12] Google Inc. Android Developers: *Supporting Different Platform Versions* [online]. [cit. 2016-02-19]. Dostupné z: <http://developer.android.com/training/basics/supporting-devices/platforms.html>
- [13] Usage Scenarios for Product Requirements. *Product Arts* [online]. 2015 [cit. 2016-02-24]. Dostupné z: <http://www.product-arts.com/articlelink/917-usage-scenarios-for-product-requirements>
- [14] Analýza požadavku. *SOVA NET* [online]. [cit. 2016-02-25]. Dostupné z: <https://www.sovanet.cz/analyza-pozadavku/>
- [15] Requirements Specification: Functional and Non-Functional Requirements. *Scottish Qualifications Authority: HN Computing* [online]. 2007 [cit. 2016-02-25]. Dostupné z: [http://www.sqa.org.uk/e-learning/SDM03CD/page\\_02.htm](http://www.sqa.org.uk/e-learning/SDM03CD/page_02.htm)
- [16] UML Tutorial: UML - Use Case Diagrams. *Tutorials Point* [online]. [cit. 2016-03-05]. Dostupné z: [http://www.tutorialspoint.com/uml/uml\\_use\\_case\\_diagram.htm](http://www.tutorialspoint.com/uml/uml_use_case_diagram.htm)
- [17] GILOVITZ, Ze'ev. Finding Dominant Image Colours Using Python. *Blog Of Ze'ev Gilovitz* [online]. 2014 [cit. 2016-03-07]. Dostupné z: <http://blog.zeevgilovitz.com/detecting-dominant-colours-in-python/>
- [18] ZUCCONI, Alan. How to find the main colours in an image. *Web Of Alan Zucconi* [online]. 2015 [cit. 2016-03-07]. Dostupné z: <http://www.alanzucconi.com/2015/05/24/how-to-find-the-main-colours-in-an-image/>
- [19] ARTHUR, David a Sergei VASSILVITSKII. *K-means++: The Advantages of Careful Seeding* [online]. Stanford University [cit. 2016-03-09]. Dostupné z: <http://ilpubs.stanford.edu:8090/778/1/2006-13.pdf>
- [20] What is K-means clustering? *OnMyPhD* [online]. [cit. 2016-03-09]. Dostupné z: [http://www.onmyphd.com/?p=k-means.clustering&ckattempt=1#h3\\_clusters](http://www.onmyphd.com/?p=k-means.clustering&ckattempt=1#h3_clusters)

- 
- [21] SCHUESSLER, Zachary. *Delta E: Color Difference Algorithms* [online]. [cit. 2016-03-13]. Dostupné z: <http://zschuessler.github.io/DeltaE/learn/>
- [22] FERONATO, Emanuele. Color difference algorithm. *Web Of Emanuele Feronato* [online]. [cit. 2016-03-15]. Dostupné z: <http://www.emanueleferonato.com/2009/08/28/color-differences-algorithm/>
- [23] *The International Commission on Illumination* [online]. Austria, 2000 [cit. 2016-03-15]. Dostupné z: <http://www.cie.co.at/index.php/LEFTMENU/About+us>
- [24] *How to Make a Domain Model: Tutorial* [online]. Pace University [cit. 2016-03-25]. Dostupné z: [http://csis.pace.edu/~marchese/CS389/L8/DomainModel-UML\\_short.pdf](http://csis.pace.edu/~marchese/CS389/L8/DomainModel-UML_short.pdf)
- [25] Activity Diagrams. *SourceMaking* [online]. [cit. 2016-03-26]. Dostupné z: <https://sourcemaking.com/uml/modeling-business-systems/external-view/activity-diagrams>
- [26] Activity Diagram. *Technopedia* [online]. [cit. 2016-03-26]. Dostupné z: <https://www.techopedia.com/definition/27489/activity-diagram>
- [27] Client/server: client/server model, client/server architecture. *SearchNetworking* [online]. 2008 [cit. 2016-04-01]. Dostupné z: <http://searchnetworking.techtarget.com/definition/client-server>
- [28] BEAL, Vangie. The Differences Between Thick & Thin Client Hardware. *Webopedia* [online]. 2006 [cit. 2016-04-01]. Dostupné z: [http://www.webopedia.com/DidYouKnow/Hardware\\_Software/thin\\_client.asp](http://www.webopedia.com/DidYouKnow/Hardware_Software/thin_client.asp)
- [29] WAYNER, Peter. Java vs. Node.js: An epic battle for developer mind share. *InfoWorld* [online]. 2015 [cit. 2016-04-01]. Dostupné z: <http://www.infoworld.com/article/2883328/java/java-vs-nodejs-an-epic-battle-for-developer-mindshare.html>
- [30] EGGER, F. N. Lo-Fi vs. Hi-Fi Prototyping: how real does the real thing have to be? [online]. „Teaching HCI“ workshop OZCHI2000. Sydney (Australia), 2000. [cit. 2016-04-01]. Dostupné z: <http://www.telono.com/en/articles/lo-fi-vs-hi-fi-prototyping-how-real-does-the-real-thing-have-to-be/>
- [31] SNOZOVÁ, Martina. Heuristická analýza. *Inflow* [online]. 2013 [cit. 2016-04-03]. Dostupné z: <http://www.inflow.cz/heuristicka-analyza>

- [32] Google Inc. Android Developers: *Design* [online]. [cit. 2016-04-01]. Dostupné z: <http://developer.android.com/design/index.html>
- [33] ŽIKOVSKÝ, P. *Návrh uživatelského rozhraní*. Praha, ČVUT, 2011. Ne-publikované přednášky.
- [34] CHANDLER, Bryson. Which is better, native app development vs. cross platform app? *Quora* [online]. 2016 [cit. 2016-04-06]. Dostupné z: <https://www.quora.com/Which-is-better-native-app-development-vs-cross-platform-app>
- [35] Google Inc. Android Developers: *Android Studio Overview* [online]. [cit. 2016-04-06]. Dostupné z: <http://developer.android.com/tools/studio/index.html>
- [36] Google Inc. Android Developers: *Tools Help* [online]. [cit. 2016-04-06]. Dostupné z: <http://developer.android.com/tools/help/index.html>
- [37] Google Inc. Android Developers: *Support Library* [online]. [cit. 2016-04-06]. Dostupné z: <http://developer.android.com/intl/ru/tools/support-library/index.html>
- [38] Google Inc. Android Developers: *Support Library Features* [online]. [cit. 2016-04-06]. Dostupné z: <http://developer.android.com/intl/ru/tools/support-library/features.html#design>
- [39] Android Universal Image Loader. *GitHub* [online]. [cit. 2016-03-15]. Dostupné z: <https://github.com/nostra13/Android-Universal-Image-Loader>
- [40] Android Parallax Recycle View. *GitHub* [online]. [cit. 2016-03-16]. Dostupné z: <https://github.com/kanytu/android-parallax-recyclerview>
- [41] Google Inc. Android Developers: *Android NDK* [online]. [cit. 2016-03-16]. Dostupné z: <http://developer.android.com/intl/ru/tools/sdk/ndk/index.html>
- [42] Salted Password Hashing: Doing it Right. *CrackStation* [online]. 2016 [cit. 2016-04-06]. Dostupné z: <https://crackstation.net/hashing-security.htm>
- [43] Asymmetric cryptography: Public-key cryptography. *Search-Security* [online]. 2008 [cit. 2016-04-06]. Dostupné z: <http://searchsecurity.techtarget.com/definition/asymmetric-cryptography>



- 
- [44] Hash-based Message Authentication Code (HMAC). *Search-Security* [online]. 2010 [cit. 2016-04-06]. Dostupné z: <http://searchsecurity.techtarget.com/definition/Hash-based-Message-Authentication-Code-HMAC>
- [45] *Express: Fast, unopinionated, minimalist web framework for Node.js* [online]. [cit. 2016-04-10]. Dostupné z: <http://expressjs.com/>
- [46] SQLite3. *GitHub* [online]. [cit. 2016-04-10]. Dostupné z: <https://github.com/mapbox/node-sqlite3>
- [47] *Multer: node.js middleware* [online]. [cit. 2016-04-10]. Dostupné z: <https://ewiggin.gitbooks.io/expressjs-middleware/content/multer.html>
- [48] Imagemagick. *GitHub* [online]. [cit. 2016-04-10]. Dostupné z: <https://github.com/yourdeveloper/node-imagemagick>
- [49] Ursa. *GitHub* [online]. [cit. 2016-04-10]. Dostupné z: <https://github.com/quartzjer/ursa>
- [50] Crypto. *Nodejs* [online]. [cit. 2016-04-10]. Dostupné z: <https://nodejs.org/api/crypto.html>
- [51] Google Inc. Android Developers: *Loading Large Bitmaps Efficiently* [online]. [cit. 2016-04-25]. Dostupné z: <http://developer.android.com/intl/ru/training/displaying-bitmaps/load-bitmap.html>
- [52] Cwac-cam2. *GitHub* [online]. [cit. 2016-04-27]. Dostupné z: <https://github.com/commonsguy/cwac-cam2>



## Seznam použitých zkratek

- GUI** Graphical user interface
- API** Application programming interface
- FR** Functional requirement
- NFR** Non-functional requirement
- REST** Representational state transfer
- UC** Use case
- RGB** Red, green, blue
- NP** Non-deterministic polynomial-time
- CIE** Commission internationale de l'éclairage
- CMYK** Cyan, magenta, yellow, key (black)
- UML** Unified modeling language
- HTTP** Hypertext transfer protocol
- JSON** JavaScript object notation
- IDE** Integrated development environment
- ARGB** Alpha, red, green, blue
- lo-fi** Low fidelity
- hi-fi** High fidelity
- SDK** Software development kit
- AVD** Android virtual device

## A. SEZNAM POUŽITÝCH ZKRATEK

---

**QEMU** Quick EMUlator

**CPU** Central processing unit

**GPU** Graphics processing unit

**ADB** Android debug bridge

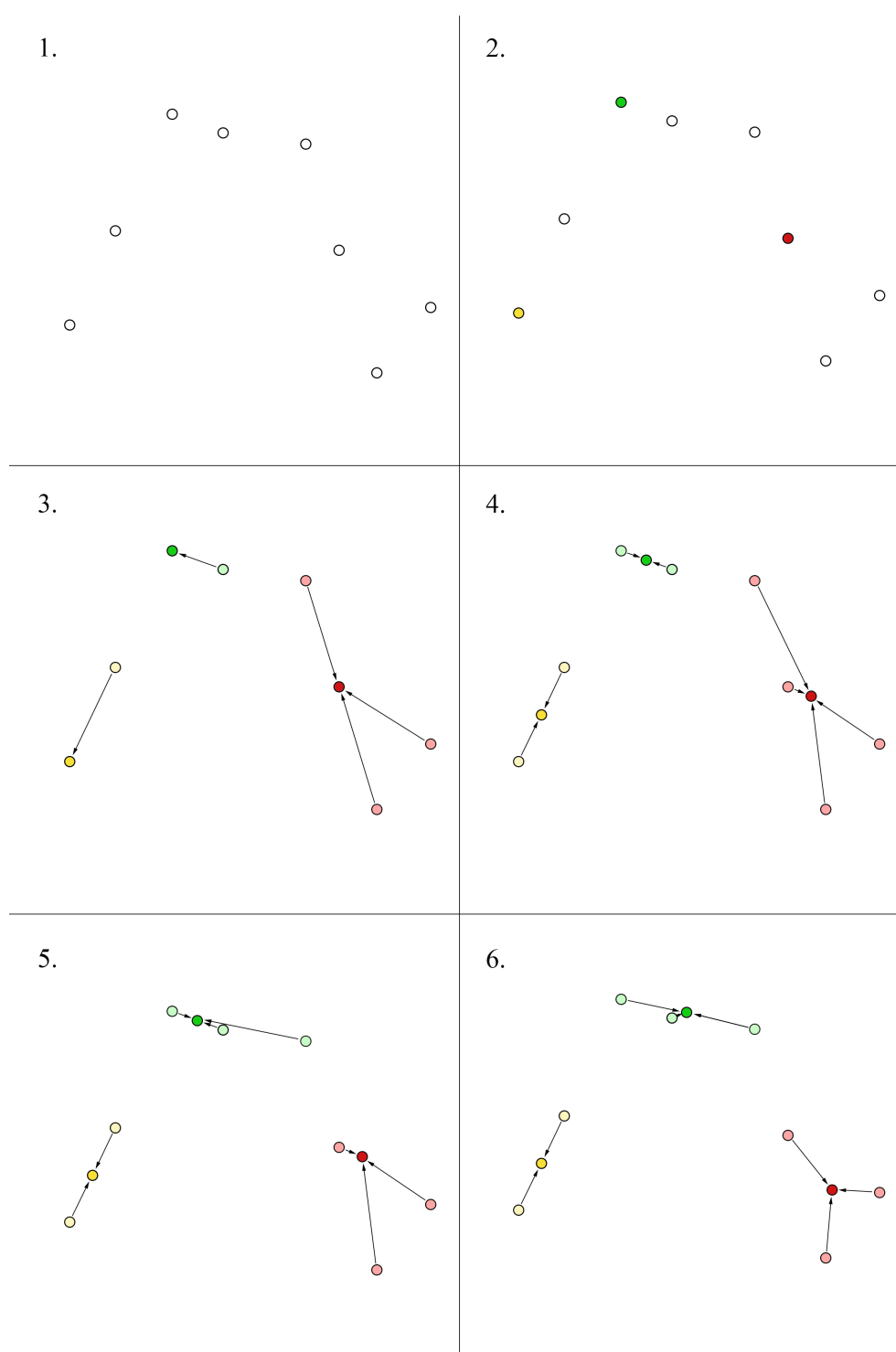
**NDK** Native development kit

**URI** Uniform resource identifier

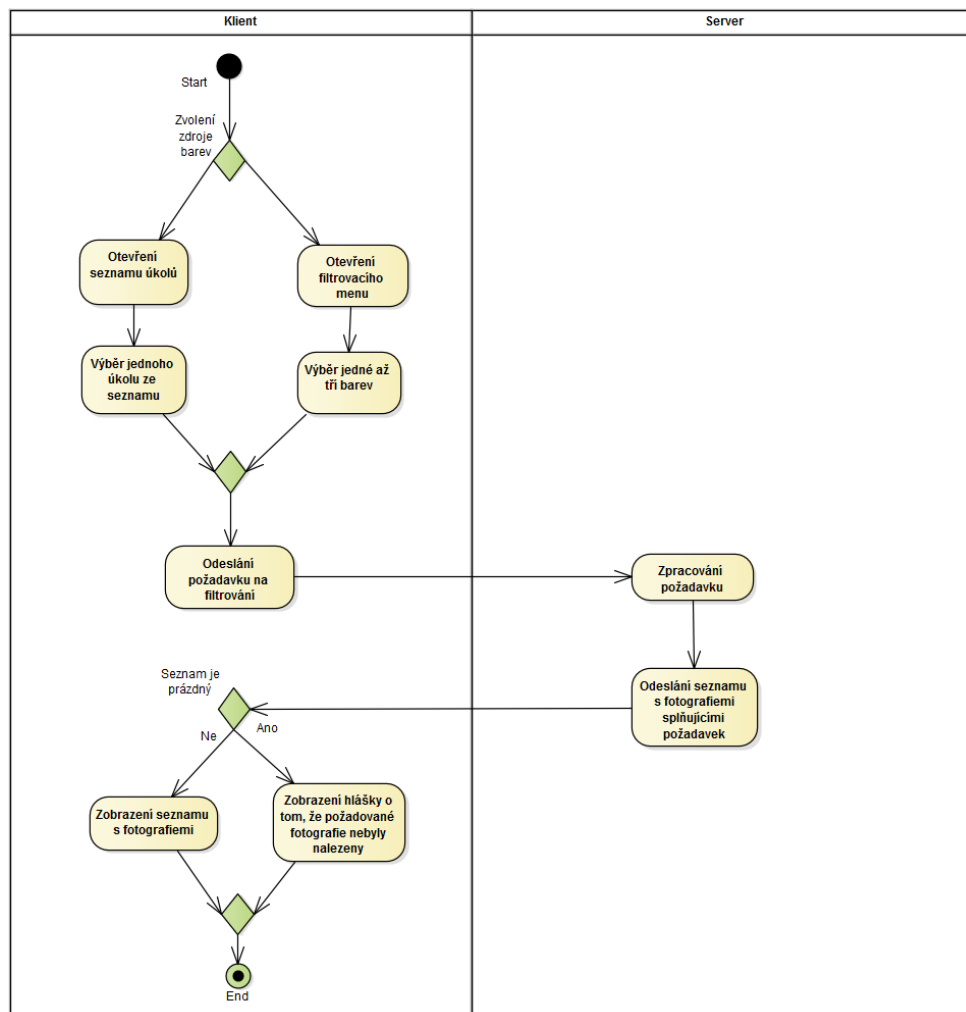
**HMAC** Keyed-hash message authentication code

**NONCE** Number used only once

## Obrázky a schémata

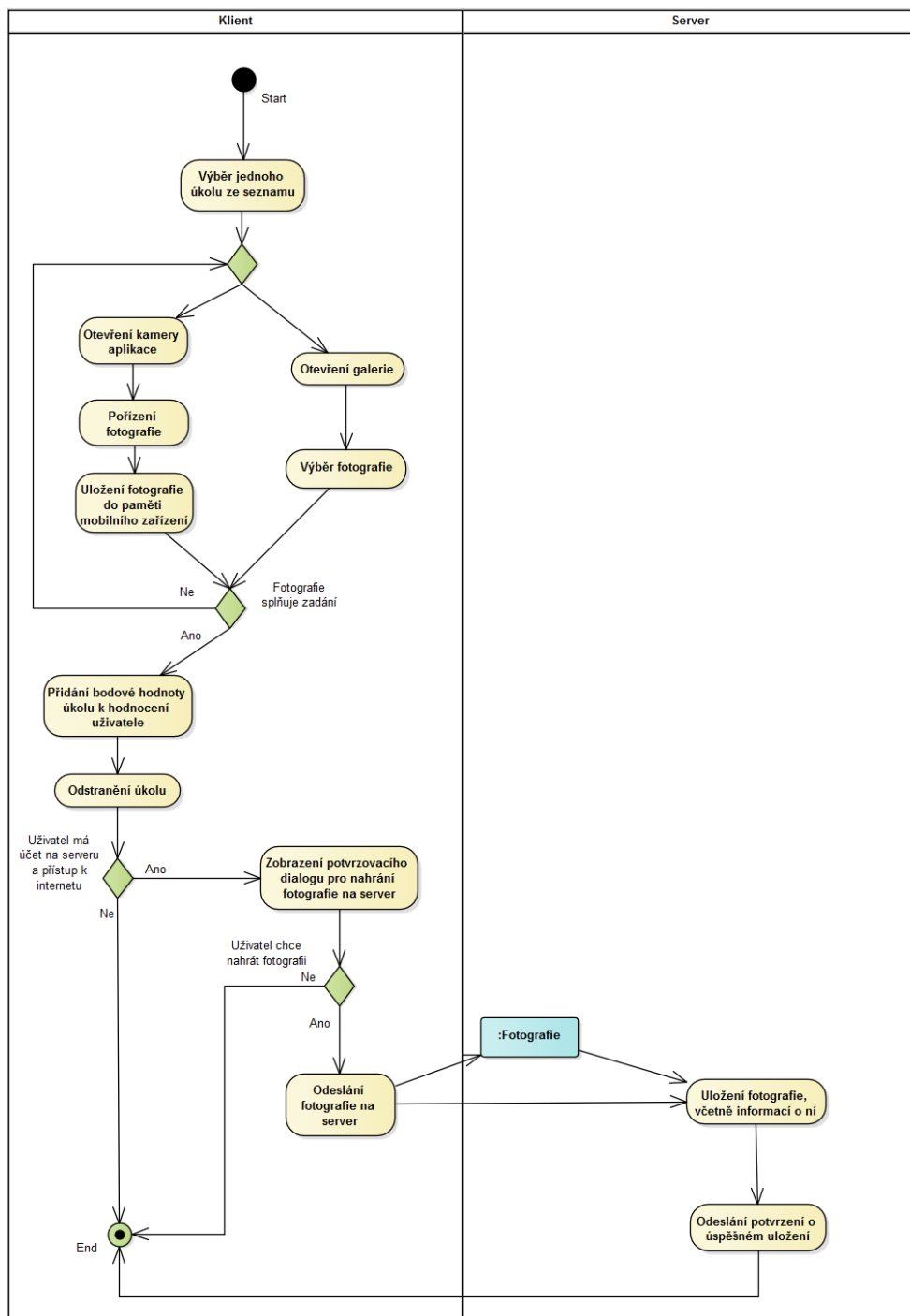


Obrázek B.1: Ukážka běhu algoritmu k-means



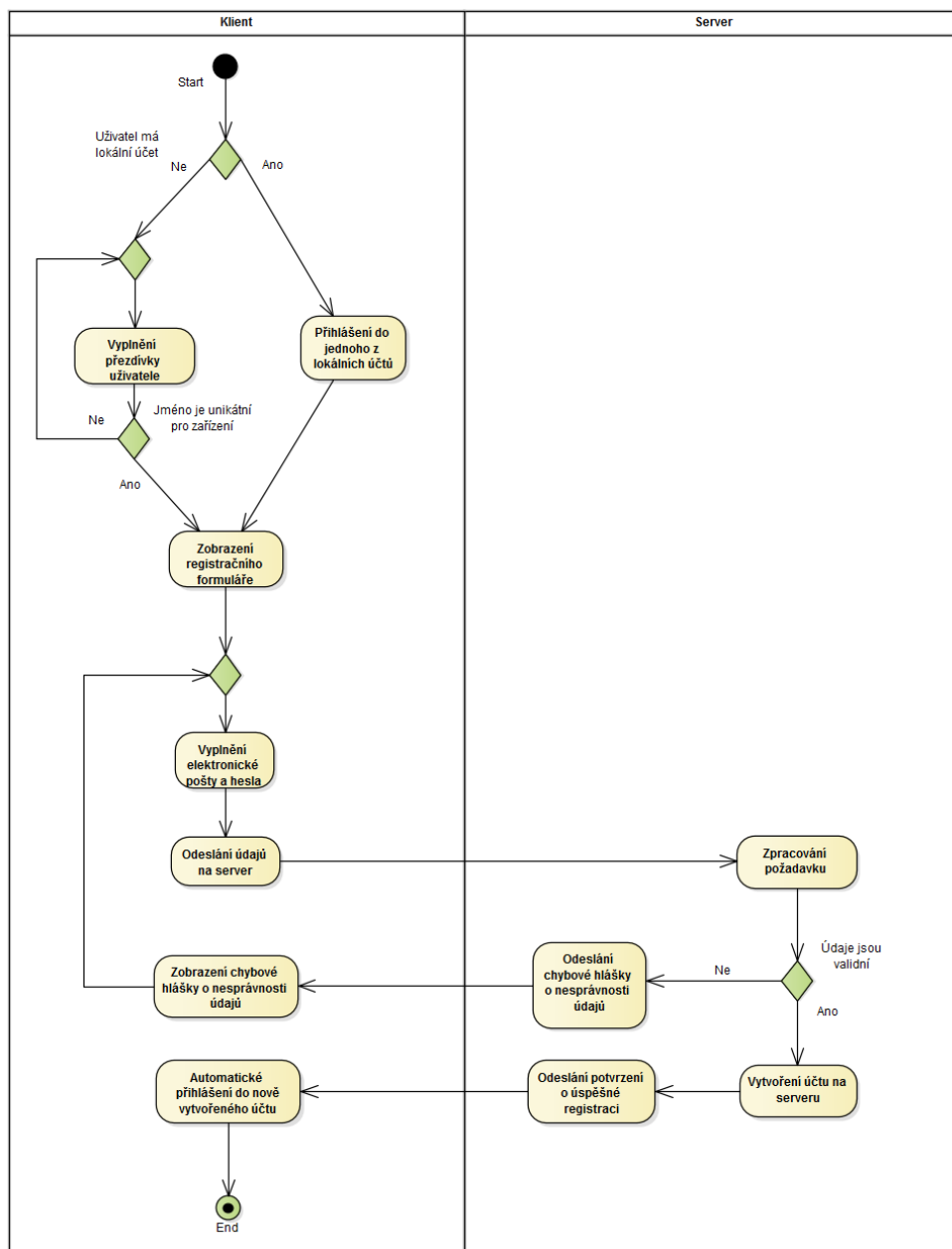
Obrázek B.2: Diagram aktivit pro filtrování fotografií

## B. OBRÁZKY A SCHÉMATA

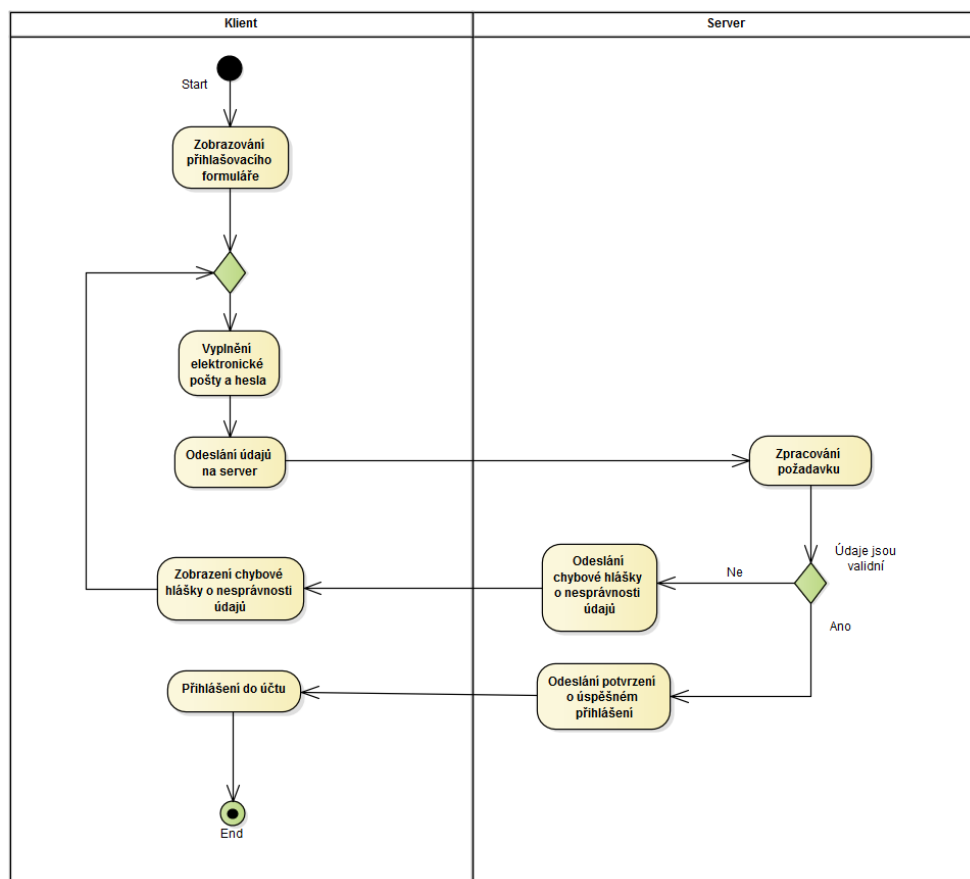


Obrázek B.3: Diagram aktivit pro splnění úkolu





Obrázek B.4: Diagram aktivit pro registraci na serveru



Obrázek B.5: Diagram aktivit pro přihlášení do účtu





---

**Formule**

$$\Delta E_{94}^* = \sqrt{\left(\frac{\Delta L^*}{k_L S_L}\right)^2 + \left(\frac{\Delta C_{ab}^*}{k_C S_C}\right)^2 + \left(\frac{\Delta H_{ab}^*}{k_H S_H}\right)^2}$$

$$\Delta L^* = L_1^* - L_2^*$$

$$C_1^* = \sqrt{a_1^{*2} + b_1^{*2}}$$

$$C_2^* = \sqrt{a_2^{*2} + b_2^{*2}}$$

$$\Delta C_{ab}^* = C_1^* - C_2^*$$

$$\Delta H_{ab}^* = \sqrt{\Delta E_{ab}^{*2} - \Delta L^{*2} - \Delta C_{ab}^{*2}} = \sqrt{\Delta a^{*2} + \Delta b^{*2} - \Delta C_{ab}^{*2}}$$

$$\Delta a^* = a_1^* - a_2^*$$

$$\Delta b^* = b_1^* - b_2^*$$

$$S_L = 1$$

$$S_C = 1 + K_1 C_1^*$$

$$S_H = 1 + K_2 C_1^*$$

Obrázek C.1: Formule pro počítání Delta E94, převzato z [21]

$$\Delta E_{00}^* = \sqrt{\left(\frac{\Delta L'}{k_L S_L}\right)^2 + \left(\frac{\Delta C'}{k_C S_C}\right)^2 + \left(\frac{\Delta H'}{k_H S_H}\right)^2} + R_T \frac{\Delta C'}{k_C S_C} \frac{\Delta H'}{k_H S_H}$$

$$\Delta L' = L_2^* - L_1^*$$

$$\bar{L} = \frac{L_1^* + L_2^*}{2} \quad \bar{C} = \frac{C_1^* + C_2^*}{2}$$

$$a_1' = a_1^* + \frac{a_1^*}{2} \left(1 - \sqrt{\frac{\bar{C}^\gamma}{\bar{C}^\gamma + 25^\gamma}}\right) \quad a_2' = a_2^* + \frac{a_2^*}{2} \left(1 - \sqrt{\frac{\bar{C}^\gamma}{\bar{C}^\gamma + 25^\gamma}}\right)$$

$$\bar{C}' = \frac{C_1' + C_2'}{2} \text{ and } \Delta C' = C_2' - C_1' \quad \text{where } C_1' = \sqrt{a_1'^2 + b_1'^2} \quad C_2' = \sqrt{a_2'^2 + b_2'^2}$$

$$h_1' = \text{atan2}(b_1', a_1') \pmod{360^\circ}, \quad h_2' = \text{atan2}(b_2', a_2') \pmod{360^\circ}$$

$$\Delta h' = \begin{cases} h_2' - h_1' & |h_1' - h_2'| \leq 180^\circ \\ h_2' - h_1' + 360^\circ & |h_1' - h_2'| > 180^\circ, h_2' \leq h_1' \\ h_2' - h_1' - 360^\circ & |h_1' - h_2'| > 180^\circ, h_2' > h_1' \end{cases}$$

$$\Delta H' = 2\sqrt{C_1' C_2'} \sin(\Delta h'/2), \quad \bar{H}' = \begin{cases} (h_1' + h_2' + 360^\circ)/2 & |h_1' - h_2'| > 180^\circ \\ (h_1' + h_2')/2 & |h_1' - h_2'| \leq 180^\circ \end{cases}$$

$$T = 1 - 0.17 \cos(\bar{H}' - 30^\circ) + 0.24 \cos(2\bar{H}') + 0.32 \cos(3\bar{H}' + 6^\circ) - 0.20 \cos(4\bar{H}' - 63^\circ)$$

$$S_L = 1 + \frac{0.015 (\bar{L} - 50)^2}{\sqrt{20 + (\bar{L} - 50)^2}} \quad S_C = 1 + 0.045 \bar{C}' \quad S_H = 1 + 0.015 \bar{C}' T$$

$$R_T = -2\sqrt{\frac{\bar{C}'^\gamma}{\bar{C}'^\gamma + 25^\gamma}} \sin \left[ 60^\circ \cdot \exp \left( - \left[ \frac{\bar{H}' - 275^\circ}{25^\circ} \right]^2 \right) \right]$$

Obrázek C.2: Formule pro počítání Delta E00, převzato z [21]

## Seznam úkolů aplikace

- Barevné úlohy
  - Zobrazit seznam úloh
  - Zobrazit fotografie vyhovující kritériím dané úlohy
  - Zobrazit detaily fotografie a indikátor úspěchu
  - Splnit úlohu
  - Smazat úlohu
  - Potvrdit odstranění úlohy
  - Zobrazit galerii
  - Zvolit fotografii z galerie
- Filtrování
  - Zobrazit filtrovací menu
  - Zvolit barvy pro filtrování
  - Zobrazit seznam vyfiltrovaných fotografií
- Fotografie
  - Zobrazit detaily fotografie
  - Skrýt detaily fotografie
  - Přidat do oblíbených/Odebrat z oblíbených
  - Přejít na následující/předchozí fotografii v seznamu
  - Zobrazit poslední nahrané fotografie na server
  - Obnovit seznam s naposledy na server nahranými fotografiemi

- Účty uživatelů
  - Zobrazit účet přihlášeného uživatele
  - Nahrát fotografii na server
  - Přidat fotografii na svůj účet z galerie
  - Přidat fotografii na svůj účet pomocí kamery aplikace
  - Zobrazit dialog pro potvrzení nahrávání na server
  - Zobrazit informace o jiném uživateli a jeho fotografiích
  - Smazat fotografii z účtu
  - Nastavit úvodní obrázek uživatele
  - Obnovit stránku s účtem uživatele
- Registrace a přihlášení
  - Odhlásit se
  - Přihlásit se jako lokální uživatel
  - Zobrazit seznam dostupných lokálních uživatelů
  - Zobrazit formulář pro přihlášení registrovaného uživatele
  - Zobrazit formulář pro vyplnění přezdívky
  - Přihlásit se jako registrovaný uživatel
  - Zaregistrovat se jako lokální uživatel
  - Zaregistrovat se na serveru
  - Zobrazit formulář pro registraci na serveru
- Kamera
  - Zapnout kameru aplikace
  - Nastavit zaostření
  - Zapnout/vypnout blesk
  - Přiblížení/oddálení při focení
  - Pořídit fotografii



---

## Výsledky testů použitelnosti

### E.1 Pre-test dotazník

1. Jaká je Vaše věková kategorie?

Kategorie 11-18: **20 %**, kategorie 18-25: **40 %**, kategorie 25-35: **20 %**, kategorie 35-50: **20 %**.

2. Jak byste ohodnotil své zkušenosti s použitím sociálních sítí na mobilních zařízeních?

Používám každý den: **40 %**, používám několikrát týdně: **20 %**, někdy jsem to použil/použila: **20 %**, nikdy jsem to nepoužil/nepoužila: **20 %**.

3. Je fotografování Vaším koníčkem?

Ano: **40 %**, trochu: **40 %**, ne: **20 %**.

4. Jak byste ohodnotil své zkušenosti s operačním systémem Android?

Pokročilý uživatel: **40 %**, běžný uživatel: **40 %**, vlastním telefon jen pro volání: **20 %**.

### E.2 Post-test dotazník

1. Bylo používání aplikace bez registrace jednoduché a pochopitelné?

Ano: **40 %**, spíše ano: **60 %**

2. Byl průběh registrace na serveru pochopitelný?

Ano: **100 %**

3. Bylo jednoduché splnit barevný úkol?

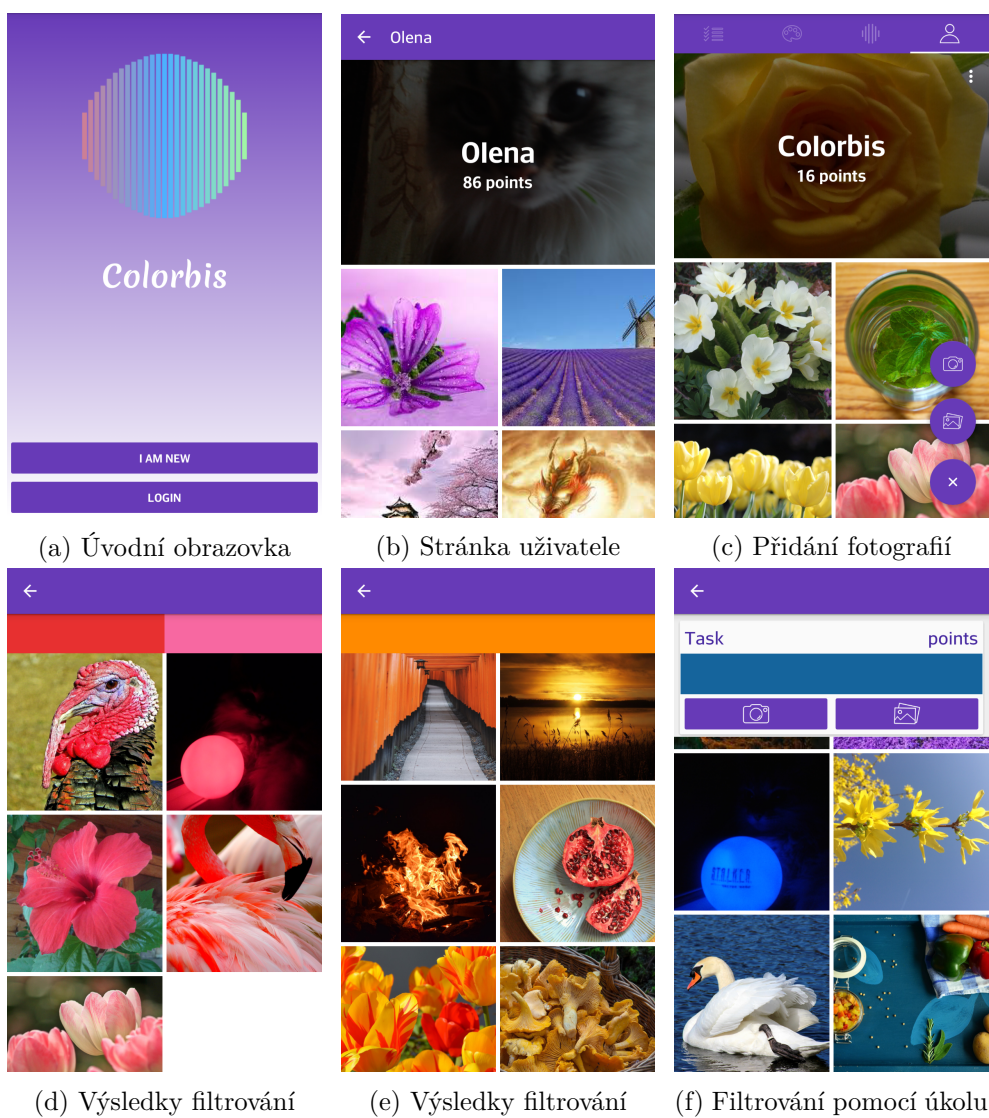
Ano: **60 %**, spíše ano: **40 %**

## E. VÝSLEDKY TESTŮ POUŽITELNOSTI

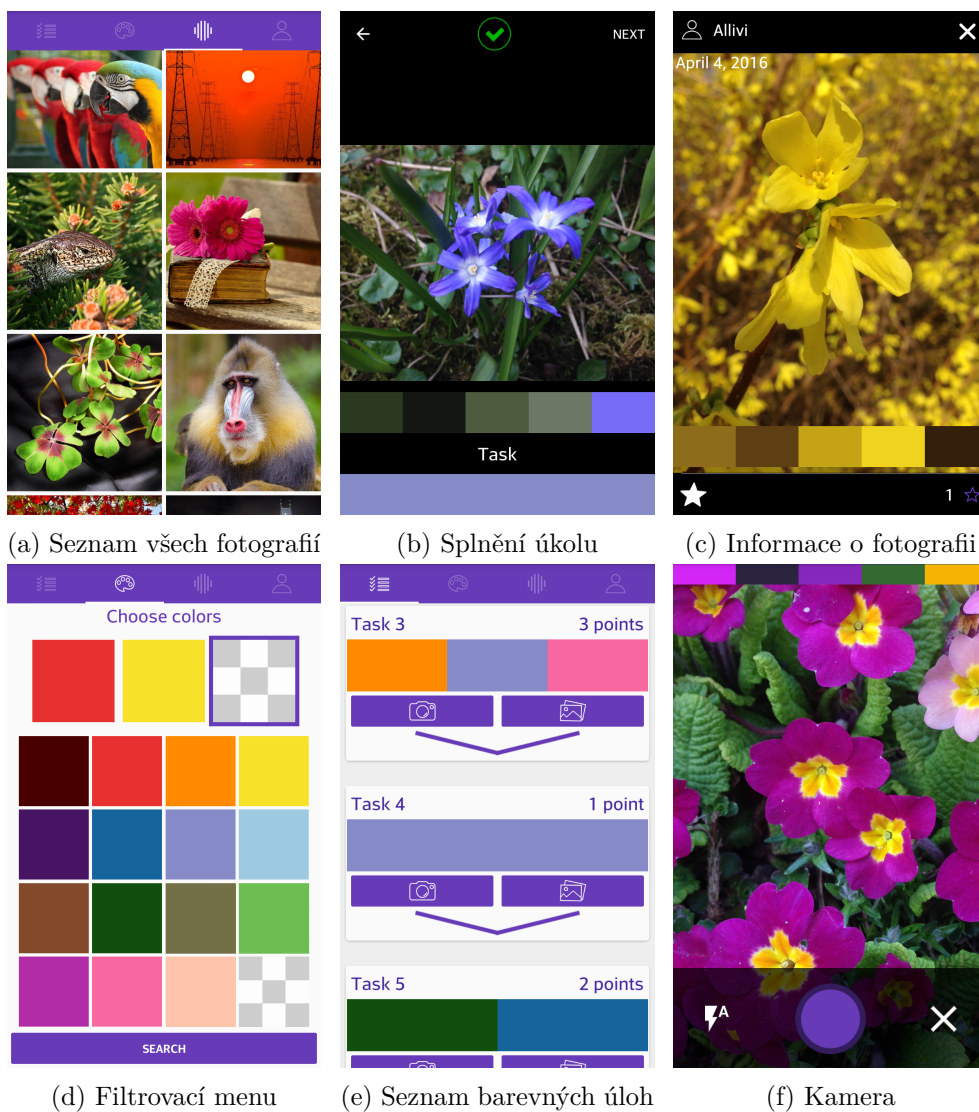
---

4. Je filtrování obrázků podle jedné až tří barev pochopitelné?  
Ano: **100 %**
5. Bylo jednoduché přidat fotografii na svůj účet po registraci?  
Ano: **100 %**
6. Jaký je Váš celkový dojem z používání aplikace? (ohodnoťte prosím od 1 do 10)  
průměr: **9,6**
7. Jaký je Váš dojem z myšlenky aplikace? (ohodnoťte prosím od 1 do 10)  
průměr: **9**

## **Ukázka výsledné aplikace**



Obrázek F.1: První část ukázky výsledné aplikace



Obrázek F.2: Druhá část ukázky výsledné aplikace



## Obsah přiloženého CD

_ colorbis.apk.....	instalační soubor
_ src	
_  _ thesis.....	zdrojová forma práce
_  _ implementation.....	ukázka zdrojových kódů
_ text	
_  _ thesis.pdf .....	text práce
_ documentation.....	programátorská dokumentace
_ attach.....	přiložené soubory