

Sem vložte zadání Vaší práce.

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



Bakalářská práce

Mobilní aplikace pro vyhledávání ukradených předmětů

Lukáš Kratochvíl

Vedoucí práce: Ing. Martin Klíma, Ph.D.

12. května 2015

Poděkování

Rád bych poděkoval panu Ing. Martinovi, Ph.D. za průběžné konzultace, cenné zkušenosti a odborné vedení při vypracování této 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. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 12. května 2015

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2015 Lukáš Kratochvíl. 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

Kratochvíl, Lukáš. *Mobilní aplikace pro vyhledávání ukradených předmětů*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.

Abstrakt

Tato práce se zabývá vývojem mobilní aplikace pro vyhledávání ukradených předmětů, kterou by měla využívat Policie ČR. Aplikace bude používána při kontrolách uměleckých děl pro případ, že by předměty byly zavedené v databázi odcizených cenností, a tudíž by nebyly majetkem osob, které tyto předměty převáží.

Klíčová slova Android, hledání odcizených předmětů, REST rozhraní, JSON, mobilní aplikace, Policie ČR

Abstract

This thesis deals with a development of a mobile application which should be used by Police of the Czech Republic. Purpose of the application is to search for stolen objects and to check whether a database of stolen valuables contains certain work of art which is carried by a person who is being inspected.

Keywords Android, searching for stolen objects, REST interface, JSON, mobile device, Police CR

Obsah

Úvod	1
Motivace práce	1
Cíl práce	1
1 Analýza	3
1.1 Současný stav	3
1.2 SimIG	5
1.3 Zhodnocení	11
2 Návrh	13
2.1 Uživatelské role	13
2.2 Scénáře	13
2.3 Požadavky	16
2.4 Případy užití	19
2.5 Výběr platformy	25
2.6 Operační systém Android	25
2.7 Vývoj vedoucí k operačnímu systému Android	27
2.8 Architektura operačního systému Android	27
2.9 Verze operačního systému Android	28
2.10 Diagram tříd	29
2.11 Návrh grafického rozhraní	32
2.12 Průchod aplikací	41
3 Implementace	61
3.1 Využití technologie	61
3.2 Focení objektu	62
3.3 Načítání fotografií z galerie	62
3.4 Vyplnění metadat	63
3.5 Vytvoření požadavku	63
3.6 Odesílání a přijímání dat	64

4 Testování	65
4.1 Testování autorem	65
4.2 Testování použitelnosti	65
4.3 Výsledky testování	66
4.4 Vyhodnocení	69
Závěr	73
Zhodnocení testu použitelnosti	73
Další vývoj aplikace	74
Literatura	75
A Seznam použitých zkratk	79
B Obsah příloženého CD	81

Seznam obrázků

1.1	Komunikace aplikace a serveru	6
1.2	Průběh komunikace aplikace a serveru	8
2.1	Grafická podoba scénáře	14
2.2	Diagram případu užití pro běžného uživatele	19
2.3	Diagram případu užití pro správce	20
2.4	Diagram případu užití pro příjemce zprávy	20
2.5	Schéma tříd	30
2.6	Návrh úvodní obrazovky	32
2.7	Návrh fotoaparátu	33
2.8	Návrh galerie obrázků	34
2.9	Návrh zobrazení detailu fotografie z galerie	35
2.10	Návrh galerie vybraných obrázků	36
2.11	Návrh ohraničení objektu na fotografii	37
2.12	Návrh vyplnění metadat k předmětu	38
2.13	Návrh obrazovky při přijímání výsledků	39
2.14	Návrh obrazovky po vyhodnocení výsledků a jejich zobrazení	40
2.15	Návrh zobrazení detailu přijatého výsledku	41
2.16	Přechody	42
2.17	Úvodní obrazovka	43
2.18	Obrazovka s fotoaparátem připraveným k focení	44
2.19	Galerie s fotografiemi	45
2.20	Detail fotografie	47
2.21	Galerie s vybranými fotografiemi k porovnání	48
2.22	Obrazovka s ohraničením oblasti fotografie	49
2.23	Obrazovka s výběrem pro odebrání fotografie z výběru	51
2.24	Výběr typu předmětu	52
2.25	Vyplnění informací popisujících objekt	53
2.26	Odesílání požadavku, pokud zařízení není připojeno k internetu	55
2.27	Odesílání požadavku, pokud zařízení je připojeno k internetu	56

2.28	Zobrazení výsledků po prohledání databáze	57
2.29	Detail jednoho z výsledků	58
2.30	Vyplnění elektronické zprávy	59
4.1	Málo viditelná zaškrtačací tlačítka	70
4.2	Nejasný účel nástroje	70
4.3	Překřížení ohraničení	71

Seznam tabulek

1.1	REST rozhraní služby SimIG	7
2.1	Rozšířenost operačních systémů[14]	25
2.2	Verze operačního systému Android[19]	28

Úvod

Motivace práce

V současné době velmi často média řeší dřevěné, umělecky dokonalé sošky, jejichž původ sahá někdy až o několik století nazpět, ukradené z kostelů, či obrazy v hodnotě milionů korun, které majitelům někdo odcizil z trezoru během noci. Krádeži lze občas předejít. Pokud má ale pachatel prostředí dobře zmapované a je dokonale připraven, sebelepší systém může selhat a vlastník o svůj klenot přijde. Později se předměty velice těžko hledají. Většina jich končí v zahraničí na černém trhu nebo u jiných sběratelů, kde je ve většině případů nulová šance je nalézt. Existuje však možnost, jak zachránit cennost před předáním v cílové lokaci, a to při jejím převozu z místa krádeže.

Zásadní otázka zní, jak poznat zločince s ukradenými předměty od vlastníka uměleckých skvostů?

Policie ČR využívá databázi obrázků ukradených předmětů, se kterou mohou policisté věc porovnat. Avšak je příliš časově náročné projít databází s tisíci záznamy, a přestože policista může použít filtry, které sníží počet objektů na stovky, ztratil by nejspíše trpělivost.

V tomto případě by policistům ulehčila práci aplikace porovnávající pořízené fotografie s databází. Výsledky by uživateli zobrazila v podobě seznamu seřazeného podle míry podobnosti s originálem. To by zrychlilo průběh vyhodnocování výsledků a také jej zpřesnilo.

Cíl práce

Cílem mé bakalářské práce je navrhnout a vhodně zvoleným způsobem naimplementovat aplikaci, která bude odesílat uživatelem pořízené fotografie podezřelých předmětů s přiloženými informacemi o objektu na server, kde se porovnají s databází ukradených předmětů. Jak již bylo řečeno, aplikace bude

ÚVOD

komunikovat se serverem, na kterém je spuštěna služba SimIG vyhodnocující a odesílající výsledky na základě přijatých požadavků.

Analýza

1.1 Současný stav

K vyhledávání obrázku slouží velmi oblíbené a populární webové služby. Většina jich však vyhledává obrázky skrze celý internet a ne jenom z uzavřené databáze s aktualizovaným obsahem. Tyto aplikace jsou dobré pro běžné uživatele, kteří nepřirazují vyhledávání tak vysokou prioritu nebo nehledají nad jasně vyhraněným souborem obrázků (odcizených předmětů), jako to bude v tomto případě.

1.1.1 Webové služby

Existuje velké množství webových služeb zabývajících se hledáním obrázků buď podle obrazové předlohy, textového zadání, nebo kombinací obou přístupů. Zde je uvedeno několik příkladů internetových stránek, pomocí kterých lze vyhledávat obrázky.

1.1.1.1 Google Images

Velice populární americká firma Google vyvinula vyhledávací službu Google Images[1] založenou na vyhledávání pomocí obrázku. Uživatel může vložit obrázek buď ze svého počítače, nebo může vložit jeho URL (Universal resource locator) adresu. Poté pomocí sofistikovaného algoritmu (jehož přesná podoba není firmou zveřejněna) najde a zobrazí ve vyhledávači podobné obrázky a na základě jmen, odkazů a textů přiložených k nalezeným obrázkům použije tato klíčová slova k dalšímu vyhledávání.

Prohledávání funguje na principu procházení webových stránek, jejichž součástí textu je i hledaný obrázek. Vyhledávání je realizováno jako hledání klíčových slov v názvech souboru, popiscích a okolním textu obrázku. Výsledek je uživateli vrácen jako odkaz na stránku v podobě zmenšených obrázků. Jak lze jistě vytušit, může nastat problém, pokud metadata obrázku neobsahují patřičné informace, nebo naopak je obsahuje obrázek, který se původnímu vůbec

nepodobá. Proto je tu další fáze vyhledávání pomocí vizuálních deskriptorů, ze kterých spočítá podobnost obrázku[2].

Postupem času firma Google přidala další zajímavá vylepšení jako například filtrování podle velikosti, podle barev, nebo i klasifikaci obrázku pomocí metadat získaných od uživatelů, kteří je doplňují formou zábavné hry[3].

1.1.1.2 Mufin

MUFIN[4] (Multi-Feature Indexing Network) je systém vyvíjený na Fakultě informatiky Masarykovy univerzity v Brně. Jedná se o systém založený na vyhledávání pomocí vizuálních deskriptorů, které lze získat z libovolného obrázku, a tudíž je lze univerzálně používat. Pro takové vyhledávání je nutné mít efektivně naimplementovaný index pro vyhledávání ve vysokodimenzionálních vektorových prostorech. Dále se obrázky porovnávají pomocí funkce vyhodnocující jejich vzdálenost (čím víc jsou obrázky vzdálené, tím víc si jsou nepodobné).

Pomocí zpracování poddotazů paralelním způsobem a využití distribuované vyhledávací sítě dokáže aplikace MUFIN Image Search vyhledávat v minimálním čase ve zhruba sto milionech objektech na webové galerii Flickr. Výsledkem jsou vizuálně podobné obrázky, v čemž opět může být problém. Podobné obrázky nemusí obsahovat stejný obsah a přitom si jsou podobné vzhledem ke vzdálenosti mezi předlohou a výsledkem.

Tyto nedokonalosti vývojáři vyřešili přidáním textových metadat v další fázi hledání. Mezi textové informace lze zařadit například datum pořízení, GPS souřadnice nebo třeba popularitu obrázku mezi uživateli. Dále lze indexováním podle vybraných kritérií např. podle barvy, tvaru (krabici od bot budu vyhledávat spíše podle barvy než tvaru), zmenšit odchýlení od původního obrázku. Pro vyhledávání se používá 5 deskriptorů podle standardu MPEG-7 (Motion picture experts group) – 2 popisují tvary, 3 barvy. Vzdálenost je počítána jako vážený průměr těchto pěti vzdáleností[2][5].

1.1.1.3 TinEye

TinEye[6] je další webovou službou, využívající čistě reverzní vyhledání fotografií (vyhledávání podle obrázkového vzoru). Stačí zadat do vyhledávače URL adresu či rovnou vložit obrázek z lokálního zařízení. Poté prohledá svou databázi a vrátí podobné obrázky. Výhodou TinEye je bezesporu vyhledání i upravených obrázků. Tato funkcionality se může zdát velice užitečná třeba v případě, že se chcete ujistit, jestli vám fotografie z webové stránky či blogu nebyly ukradeny.

Určitě uživatelé kladně přijmou fakt, že tato služba je nabízena i jako doplněk do internetových prohlížečů Google Chrome nebo Firefox.

Firma TinEye za poplatek nabízí využívání aplikačního rozhraní pro vyhledávání obrázků. Tato placená služba (oproti neplacenému vyhledávání skrze

webovou stránku) nabízí vyhledání obrázků, které se od originálu liší ořezáním nebo jinou úpravou obrázku. I když se jedná o placenou verzi vyhledávání, uživatel si nemůže vytvořit vlastní databázi obrázků.

To je možné ve službě MatchEngine[7]. Tato služba vytvoří ke každému obrázku v databázi, kterou si uživatel sám vytvořil, unikátní příznak. Díky němu je vyhledávací algoritmus schopen najít různé modifikace (například otočení, převrácení, zmenšení) originálu. Služba nerozpozná, zda se na různých snímcích jedná o tutéž osobu. Také ji nelze použít jako čtečku UPC (Universal product code) nebo QR (Quick response) kódů.

1.1.2 Mobilní aplikace

1.1.2.1 Google Goggles

Google Goggles[8] je aplikace vyvinutá firmou Google. Uživatel zapne tento program a pomocí fotoaparátu v mobilním zařízení vyfotí obrázek. Pokud služba najde podobný obrázek ve své databázi, zobrazí uživateli užitečné informace týkající se vyfoceného objektu. Tato aplikace dokáže přečíst a přeložit text do jiných jazyků nebo může být použita jako čtečka UPC a QR kódů. Dokonce je schopna vyřešit sudoku, které uživatel vyfotí.

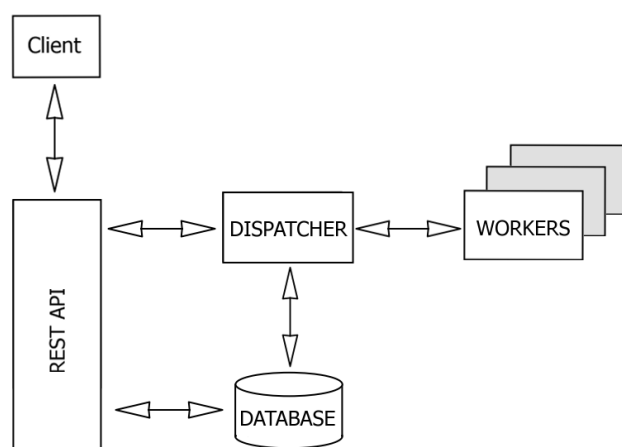
1.1.2.2 Unravl

Unravl[9] je další mobilní aplikací, která dokáže vyhledávat podle obrázkové předlohy. Pro vyhledávání používá kromě svých technologií také výše zmíněné služby Google Images nebo TinEye. Díky tomu, že využívá i jiné služby, dokáže získat výsledky z více zdrojů a poté je sloučit. Dalším rozdílem oproti ostatním je schopnost oříznout nebo otočit obrázek, který se odesílá na server[10]. Ve výsledku to znamená, že lze vybrat pouze část obrázku, kterou chce uživatel vyhledat v databázi. Kladem této aplikace je možnost výběru, jak může být obrázek odeslán. Uživatel v rámci aplikace může vybrat již existující obrázek z galerie, vyfotit ho a následně odeslat nebo zadat jeho URL adresu[11].

1.2 SimIG

Služba přijímá požadavky odeslané klientem, zpracovává je a výsledky vrací zpět klientovi. Všechna komunikace mezi klientem a serverem probíhá skrze RESTové (Representational state transfer) aplikační rozhraní. Data jsou přenášena ve formě JSON (JavaScript object notation). Přenos dat je zprostředkován skrze HTTPS (Hypertext transfer protocol secure) protokol (obr. 1.1).

Práce serveru začíná přijetím požadavku ve formě řetězce. Dále je požadavek přeposlán k volné výpočetní jednotce. Tímto způsobem je zajištěna možnost zpracovávat úlohy paralelně, pokud existuje více výpočetních jednotek. Jednotka získá z JSON řetězce bitmapu zakódovanou v base64 a oblast,



Obrázek 1.1: Komunikace aplikace a serveru

kerou v ní uživatel ohraničil. Poté pomocí algoritmu porovná dekodovaný obrázek s databází. Vyhledávání je založené na porovnávání lokálních deskriptorů v podobě binárních řetězců. Lokálním deskriptorem se rozumí příznak popisující pouze část obrázku. Oproti tomu globální deskriptor popisuje celý obrázek (například histogram). Jednotlivé znaky v řetězci se určí porovnáním dvou pixelů v okolí význačného bodu. Význačný bod je definován jako bod v obrázku, který je snadno detekovatelný a lze ho najít i po působení změn v obrázku (například pootočení, zmenšení nebo zvětšení). Následně se na sebe deskriptory pokusí namapovat. Čím větší plochu namapované deskriptory vytvoří, tím více si jsou obrázky podobné.

1.2.1 RESTové aplikační rozhraní

Jelikož klientská aplikace komunikuje se serverem (odesílá požadavky, přijímá odpovědi), musí být dotazování se serveru jednoznačně určeno. Toho je docíleno využitím RESTového rozhraní.

REST je architektura navržená pro distribuované prostředí, což znamená, že klient odešle data na serverovou část, kde jsou zpracována a následně odeslána. Všechny příkazy jsou předávány pomocí HTTP (Hypertext transfer protocol) protokolu, kdy se používají stejné metody (GET, POST, PUT, DELETE, ...) pro přístup k webovým stránkám a odesílání dat na server. Dále pro jednoznačnost při odesílání příkazů určuje REST každému zdroji jeho URI (Uniform product identifier)[12].

Popis RESTového rozhraní služby SimIG je popsán v tabulce 1.1.

- Tělo požadavku odeslaného metodou POST obsahuje informace o obrázku a další data vztahující se k samotnému vyhledávání. Dále {dataset_id} představuje jméno datasetu (množina reprezentující ob-

Tabulka 1.1: REST rozhraní služby SimIG

URI identifikátor	Metoda	Tělo požadavku	Výsledek
api/v1.0/datasets/ {dataset_id} /images/similar	POST	JSON řetězec	ID úlohy zpracovávající odeslaný požadavek (task_id)
api/v1.0/tasks/ {task_id}	GET	prázdný	JSON řetězec obsahující výsledky
api/v1.0/tasks/ {task_id}	PATCH	JSON řetězec	prázdný

rázky) podle toho, do jaké kategorie uživatel předmět zařadil. Tato metoda vrací odpověď, v jejíž hlavičce se nachází ID úlohy. Ta zpracovává odeslaný požadavek.

- Metodou GET se aplikace dotazuje serveru na úlohu. V těle přijaté odpovědi jsou mimo jiné stav vyhledávání, URL adresy nalezených podobných obrázků a jejich míra podobnosti. Průběh zpracování se klient dozví z těla odpovědi v číselné formě z klíče „progress“.
- Metodou PATCH lze úlohu na serveru pozastavit, úplně zastavit, nebo v ní pokračovat, podle toho, jaký stav (cancel, susped, resume) aplikace odešle v těle JSON požadavku. Pro zastavení vyhledávání se odešle požadavek.

```
{
  "state": "suspend"
}
```

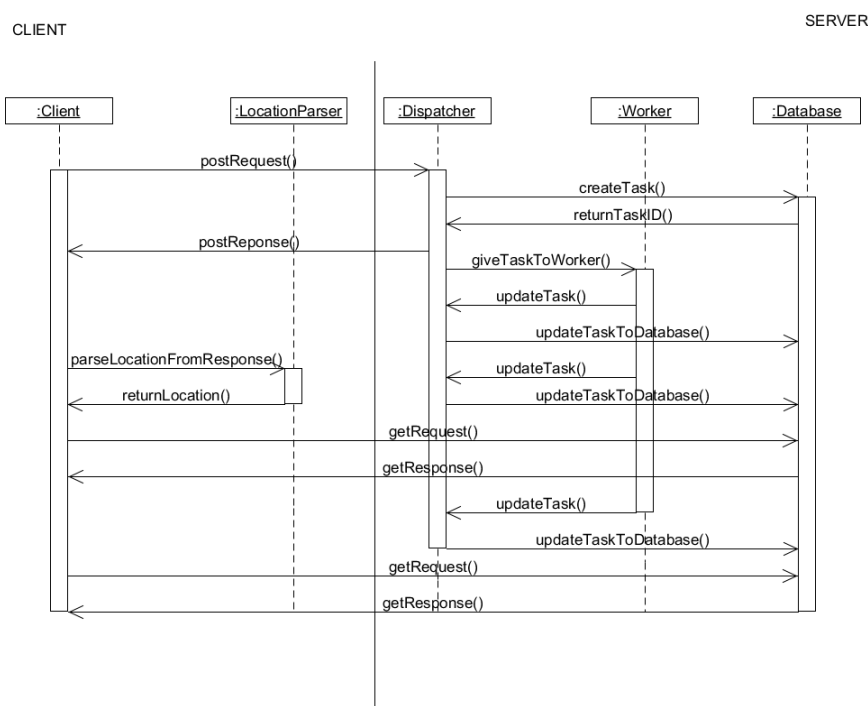
Jak již bylo zmíněno, server pracuje s architekturou REST, takže má jasně dané rozhraní, pomocí kterého k němu klientská aplikace přistupuje. Je zde definovaných 5 operací, které se využívají například pro vypsání obrázku v datasetu, přidání, odebrání, přesunutí do jiného datasetu nebo získání metadat obrázku z databáze, vypsání všech datasetů, přidání nebo odebrání datasetu. Zmíněné operace jsou určeny pro správce databáze, tudíž je běžný uživatel aplikace za účelem porovnání nebude provádět.

Komunikace je graficky znázorněna pomocí obrázku 1.2.

1.2.2 Přenos dat ve formě JSON

Odeslat jakoukoli informaci ze současných mobilních zařízení na server nebo ji přijmout není žádný problém. Větším problémem je sjednotit formát dat,

1. ANALÝZA



Obrázek 1.2: Průběh komunikace aplikace a serveru

aby byla pro obě strany čitelná. Výhodnější je data před odesláním uspořádat do předem určených a dle určitých pravidel uspořádaných struktur, jako je například XML (Extensible markup language) nebo JSON. A právě JSON je využit pro komunikaci mezi klientem a serverem.

JSON je strukturovaný, lehce analyzovatelný způsob zápisu objektů. Oproti tomu je XML sice obsáhlé, ale velice flexibilní, proto je vhodné pro udržení přidaných informací o textu, jako je například formátování. Dále JSON dokáže používat jen předem dané typy, oproti tomu XML si může definovat vlastní[13].

Pro názornou ukázkou uvedu POST požadavek. Dotaz obsahuje obrázek zakódovaný v base64 (query), pole s relativními souřadnicemi jednotlivých ohraničení (segmentations) a další informace jako jsou filtry pro vyhledávání v databázi (width), počet deskriptorů použitých při porovnávání (descriptors-Limit) a počet nejlépe ohodnocených obrázků, které server uživateli vrátí (bestImagesCount). Jak je vidět v příkladu, server je schopen u jednoho obrázku zpracovat více oblastí. Vytvářená aplikace bude odesílat požadavek vždy jen s jedním ohraničením.

První příklad představuje JSON notaci.

{

8


```

"filter": {
  "width": {
    "$gt": 2000
  }
},
"query": "data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAA
AQABAAD/4dRQRXhpZgAASUkqAAgAAAAMAAABBAABAA
AAAwAwAAAEbBAABAAAAkAkAAA8BAGATAAAAngAAA
.
.
.",
"descriptorsLimit": 400,
"bestImagesCount": 10,
"segmentations": [
  {
    "type": "manual",
    "params": {
      "type": "points",
      "points": [
        {
          "x": 0.1,
          "y": 0.2
        },
        {
          "x": 0.9,
          "y": 0.12
        },
        {
          "x": 0.8,
          "y": 0.7
        },
        {
          "x": 0.1,
          "y": 0.8
        }
      ]
    }
  }
]
}

```

Druhý příklad nesoucí tutéž informaci znázorňuje XML formátování.

<request>

```
<filter>
  <width>
    <greater type="int" value="2000">
  </width>
</filter>
<query type="string">
  data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAA
  AQABAAD/4dRQRXhpZgAASUkqAAgAAAAMAAABBAABAA
  AAwAwAAAEBBAAABAAAkAkAAA8BAgAIAAAAngAAA
  .
  .
  .
</query>
<descriptorsLimit type="int">400</descriptorsLimit>
<bestImagesCount type="int">10</bestImagesCount>
<segmentations>
  <type>manual</type>
  <params>
    <type>points</type>
    <points>
      <point x="0.1" y="0.2">
      <point x="0.9" y="0.12">
      <point x="0.8" y="0.7">
      <point x="0.1" y="0.8">
    </points>
  </params>
</segmentations>
</request>
```

Na příkladech je vidět rozdíl mezi jednotlivými formáty. XML má schopnost detailněji popsat složité struktury, které aplikace odesílá nebo přijímá ze serveru. Avšak lepší přehledností kódu dominuje JSON.

V tomto případě se bude odesílat objekt s takovými typy parametrů, které JSON podporuje, takže pokud jde o výběr formátování informací pro přenos dat, je JSON lepší volbou než XML.

Pro jednodušší práci s JSON objekty obsahuje operační systém Android knihovny `org.json.JSONObject` a `org.json.JSONArray`. Tyto nástroje na vytváření požadavků a zpracovávání odpovědí se jeví jako velice užitečné už jenom z toho důvodu, že je lze libovolně kombinovat a vnořovat je do sebe. Jelikož se v JSON objektu informace ukládají ve formě klíč-hodnota, nespornou výhodou je jednoduché vytažení dat, které probíhá zadáním klíče a získáním dané hodnoty.

1.3 Zhodnocení

V současné době existují služby na vyhledávání pomocí obrázkové předlohy. Většina z nich je přístupná pomocí internetového prohlížeče. Některé nabízí veřejné aplikační rozhraní, díky němuž je službu možné využívat i v mobilních aplikacích. Avšak je tu problém s obsahem databází, se kterým se bude pořízený snímek porovnávat.

Vyvíjená aplikace by potřebovala pracovat s databází, ve které by byly uloženy pouze fotografie ukradených objektů. Vytvoření vlastní kolekce obrázků nabízí za poplatek pouze jedna služba a to MatchEngine od firmy TinEye.

Zajímavým prvkem je oříznutí fotografie, které je možné provést u mobilní aplikace Unravl. Tato funkcionality je velice užitečná, protože se díky ní zmenší pravděpodobnost nalezení nesprávného výsledku.

Bohužel na trhu neexistuje software, jež by zkombinoval potřebné funkce, které současné služby a aplikace nabízí. Jedinou volbou je tedy služba SimIG, která nemá veřejně přístupné rozhraní. Ta obsahuje databázi s ukradenými předměty a podporuje ohraničení objektu na fotografii, která je odeslána na porovnání.

Návrh

2.1 Uživatelské role

Pro tuto aplikaci byly definovány následující role.

2.1.1 Běžný uživatel aplikace

Běžný uživatel je každý uživatel, který bude využívat aplikaci k účelům, k jakým byla vytvořena, tudíž právě pro něj je určen scénář obsažený v této práci. Konkrétně bude moci fotit objekty, následně je vybrat z galerie nebo vybrat ty, které už vyfoceny byly. Bude moci ohraničit oblast, přidat k ní další textové informace, odeslat ji, přijmout výsledek zpracovaný serverem a odeslat pomocí elektronické pošty.

2.1.2 Uživatel přijímající výsledek

Tuto roli může získat každý uživatel, kterému pomocí elektronické pošty pošle běžný uživatel výsledky vyhledávání, jež sám provedl. Údaje bude moci odeslat pouze po úplném skončení vyhledávání a zobrazení nalezených děl na mobilním zařízení.

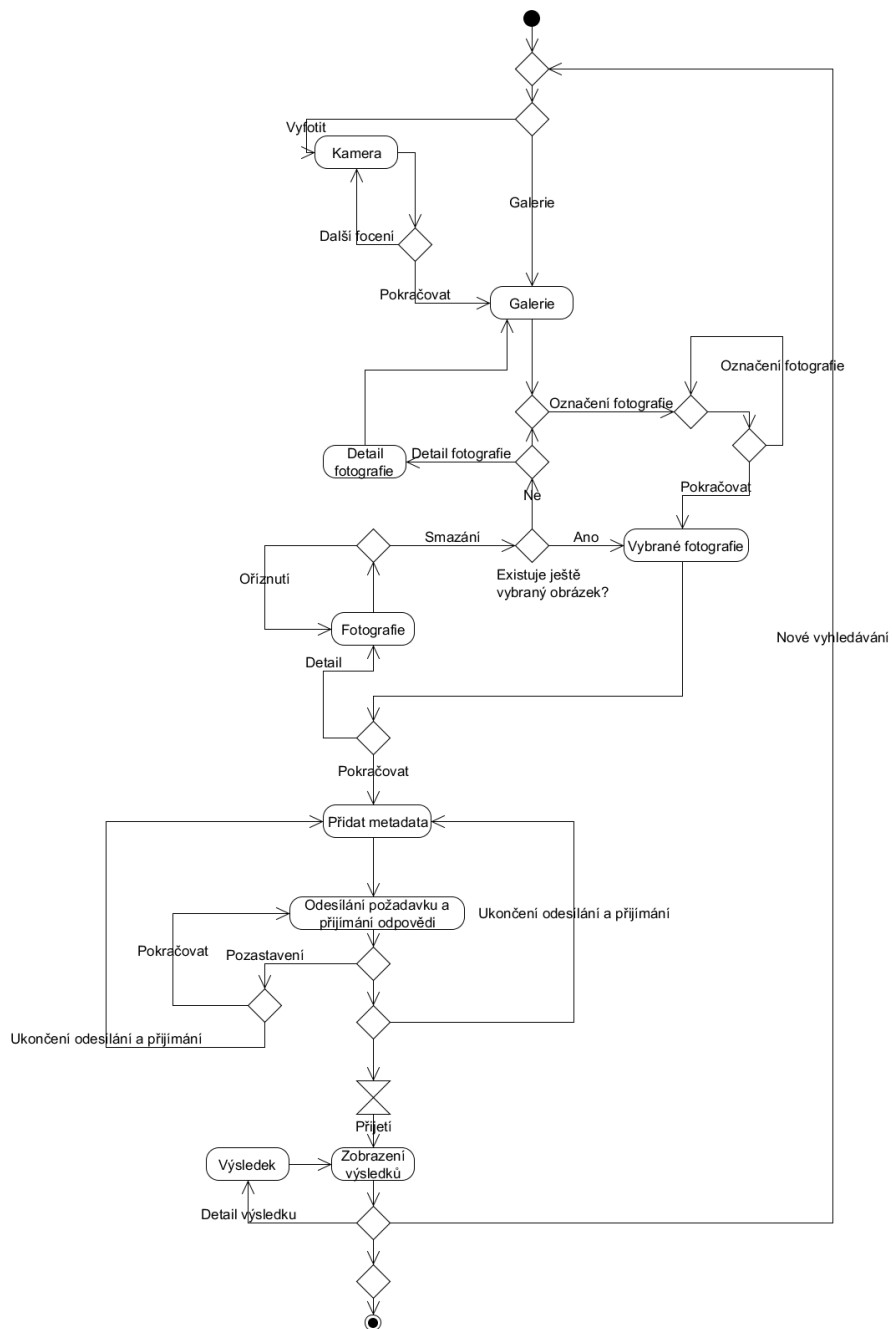
2.1.3 Správce databáze

Správce má za úkol starat se o databázi, přidávat nové předměty do databáze, odebírat již nalezené, mazat úlohy, přidávat a odebírat datasey (množina reprezentující obrázky).

2.2 Scénáře

Z důvodu lepší představy průchodu aplikace je obecný scénář převeden do grafické podoby (obr. 2.1).

2. NÁVRH



Obrázek 2.1: Grafická podoba scénáře

2.2.1 První scénář

Pro názornější ukázkou vytvořím dvě osoby: pana Františka Nováka (sběratele cenných předmětů) a pana Josefa Koudu (policistu).

Pan Novák je vášnivým sběratelem cenných předmětů. Jednou z jeho vlastností je i opatrnost, proto si pan Novák pořídil fotografie všech svých předmětů pro případ, že by mu byly odcizeny. Naneštěstí se tato nemilá událost odehrála, panu Františkovi byly cennosti odcizeny. Šel proto na policii a předal policistům fotografie ukradených předmětů, které úředník nahrál do databáze na serveru.

Při běžné policejní kontrole pan Kouda procházel automobil, který měl namířeno do zahraničí. Všiml si starožitných děl v zavazadlovém prostoru, a tak se chtěl ujistit, zda nejsou v databázi odcizených předmětů.

Vzal si mobilní telefon, nastavil připojení k internetu a spustil aplikaci na vyhledávání ukradených předmětů. Zde měl na výběr, jestli pořídí novou fotografii, nebo vybere k porovnání již existující obrázek v telefonu. V tomto případě otevřel fotoaparát a objekt vyfotil pro jistotu z několika úhlů, aby byla větší pravděpodobnost nalezení předmětu v databázi.

Po focení pokračoval pomocí tlačítka na další obrazovku, kde se mu otevře galerie (do tohoto bodu by se mohl dostat výběrem druhé možnosti po spuštění aplikace). Zde si ještě prohlédl obrázky jejich zvětšením na 100 % obrazovky bez ořezání.

Dále policista Kouda vybral obrázky, na kterých byl objekt nejlépe zachycen. V dalším kroku se mu zobrazí vybrané fotografie a u každé z nich ohraničí relevantní část, aby bylo jasné, jaká část obrázku má být porovnána. Pokud se mu výběr nepovede tak, jak by si představoval, může část přidat, odebrat, nebo rovnou celý obsah smazat a označit znova.

Na další stránce policista přidá textové informace upřesňující hledaný předmět, jako výšku, šířku, materiál, ze kterého je vyroben, a jiné další atributy.

V dalším kroku aplikace odešle požadavek, který pan Kouda vytvořil, a bude zobrazovat průběžné výsledky vyhledávání a stav vyjádřený procenty. V tomto kroku může uživatel zpracovávání pozastavit nebo i úplně zastavit a vrátit se na vyplňování metadat.

Po přijetí všech výsledků se panu Josefovi zobrazí všechny výsledky s ohodnocením v podobě hvězdiček. Pokud si policista nebude jistý, může si výsledek zobrazit detailněji (fotografie se mu zvětší na 100 % obrazovky bez ořezání, pod fotografií budou přidružené informace vztahující se k obrázku). V tento okamžik pan Kouda rozhodne, zda se jedná o ukradený předmět. Pokud si ale není jistý, může ještě jednotlivé fotografie označit a odeslat je elektronickou zprávou třeba svému kolegovi nebo nadřízenému.

2.2.2 Druhý scénář

Dalším ukázkovým scénářem může být nákup cenných předmětů ve starožitnictví.

Sběratel přišel do obchodu a začal si prohlížet vystavené zboží. Zalíbila se mu lampa, kterou by rád koupil, ale než se rozhodl zaplatit, chtěl se ujistit, že není kradená.

Vzal tedy mobilní telefon, spustil aplikaci na vyhledávání ukradených předmětů a vyfotil lampu. Při dalším kroku vybral vhodné obrázky k porovnání.

Pokračováním se dostal na galerii vybraných obrázků, kde pro přesnější porovnání označil oblast obrázku, ve kterém se nachází lampa.

Po ohraničení při dalším kroku vyplnil informace vztahující se k předmětu. Jelikož se v oblasti starožitností pohybuje už delší dobu, vyplnil všechny nabízené atributy a přidal několik vlastních poznatků.

Odeslal požadavek na server a čekal na odpověď. Po vrácení výsledku si prohlédl zobrazené položky, které byly nejvíce podobné lampě, a ručně je porovnal s originálem. Porovnáním předmětu s výsledky vyhledávání zjistil, že lampa v databázi není, a mohl si ji bez obav koupit.

2.3 Požadavky

2.3.1 Funkční požadavky

- F1 – Pořízení fotografie

Aplikace umožní pořizovat obrázky objektu pomocí fotoaparátu v mobilním zařízení.

- F2 – Vyfocení více fotografií

Uživatel bude mít možnost pořídít více fotografií bez neustálého zapínání foťáku. Aplikace bude focení schopná do té doby, dokud uživatel nepřejde na další krok.

- F3 – Uložení fotografie na externí úložiště

Pořízené fotografie budou uloženy do složky DCIM (Digital camera images). V úložišti obrázky zůstanou uchovány i po vypnutí aplikace.

- F4 – Přístup a práce s úložištěm

Obrázky bude možné nalézt ve složce DCIM, a jelikož nebudou uloženy jen dočasně, bude možné je po vypnutí a zapnutí aplikace vybrat a odeslat na porovnání. Aplikace bude schopna načíst i ostatní obrázky, které se budou nacházet ve stejné složce jako fotografie pořízené aplikací.

- F5 – Ohraničení relevantní oblasti jednotlivých obrázků

Aplikace umožní označit oblast vybrané fotografie, kterou uživatel bude chtít odeslat na porovnání. Podstatná oblast nemusí být vybrána u žádné, jedné nebo i u všech fotografií.

- F6 – Vyplnění metadat

Uživatel bude moci vyplnit další informace týkající se předmětu, které k němu budou přidruženy a odeslány na server. Podle typu atributu uživateli aplikace nabídne 3 druhy přiřazení hodnoty (výběr z daného seznamu, textovou hodnotu, číselnou hodnotu). Dále se bude moci číselná hodnota dělit na ty, ke kterým lze doplnit veličinu, a na ty, ke kterým ne.

- F7 – Vytvoření požadavku

Prvním krokem k vytvoření požadavku bude vybrání fotografií. Dále se uživateli zobrazí fotografie jím již vybrané, kde bude moci ohraničit důležitou oblast obrázku. Další součástí požadavku jsou přidávané textové informace týkající se objektu určeného k porovnání s databází.

- F8 – Odesílání požadavku

Po dokončení vytváření požadavku (výběru fotografií, ohraničení relevantních oblastí a přidání textové informace) aplikace jeho konečnou podobu odešle na server ke zpracování.

- F9 – Přerušení vyhledávání

Pokud se uživatel rozhodne vyhledávání z nějakého důvodu přerušit, použije tlačítko, které odešle na server příkaz, aby prohledávání pozastavil. Aplikace bude pokračovat po opětovném stisknutí tlačítka.

- F10 – Zastavení vyhledávání

Jestliže uživatel bude chtít vyhledávání ukončit, stiskne tlačítko „Zastavit“ ukončit vyhledávání. Pokud bude chtít opět vyhledávat, bude muset odeslat požadavek znovu, což znamená, že server začne vyhledávat od začátku.

- F11 – Přijímání odpovědí na požadavky

Po odeslání požadavku na server se aplikace pravidelně bude ptát serveru na výsledky vyhledávání. Ty budou postupně zobrazovány. Po dokončení porovnávání se na straně klienta zobrazí výsledky.

- F12 – Výběr více obrázku k porovnání

Při vytváření dotazu bude moci uživatel vybrat více obrázků k porovnání. U každého obrázku bude moci určit oblast. Výběr více obrázků jednoho předmětu zvýší pravděpodobnost nalezení předmětu v databázi. Obrázky budou v galerii seřazeny sestupně podle data pořízení.

2. NÁVRH

- F13 – Ohodnocení výsledků pomocí hvězdiček
Výsledky vyhledávání zobrazené aplikací budou ohodnoceny pomocí hvězdiček, kde nejmenší možný počet bude nula a maximální pět. Hvězdičky budou znázorňovat míru shody.
- F14 – Odeslání vybraných výsledků pomocí elektronické pošty
Po dokončení bude moci uživatel vybrat a odeslat výsledky pomocí elektronické pošty. Výsledky se odešlou v těle zprávy pomocí URL adres obrázků.
- F15 – Zobrazení obrázků jednotlivých výsledků s přidruženými informacemi
Aplikace umožní zobrazit uživateli detail výsledků v podobě obrázku a informací, které ho upřesňují.
- F16 – Přiblížování a oddalování obrázku
Pokud uživatel zobrazí detail fotografie v galerii (při výběru obrázku k porovnání), bude moci přiblížit nebo oddálit obrázek.
- F17 – Nutnost výběru alespoň jednoho obrázku k porovnání
Uživatel bude muset vybrat alespoň jednu fotografii v galerii, kterou bude chtít použít k vyhledávání. V opačném případě mu aplikace nedovolí pokračovat.

2.3.2 Nefunkční požadavky

- N1 – Mobilní aplikace
Aplikace, která bude výsledkem této práce, bude používána v terénu, tudíž bude muset být přizpůsobena okolnostem, za kterých bude sloužit. Proto se bude jednat o mobilní aplikaci, jež půjde nainstalovat a spustit na zařízeních, jako jsou mobilní telefony či tablety.
- N2 – Klient aplikace
Z důvodu úspory vnitřní paměti a výkonu mobilního zařízení, bude aplikace z hlediska komunikace přes internet vyvíjena jako klientská strana, která bude komunikovat se serverem. Server bude přijímat požadavky, které bude zpracovávat a následné výsledky pošle zpět na mobilní zařízení.
- N3 – REST API (Application programming interface)
Pomocí RESTového rozhraní bude aplikace odesílat požadavky na server, který je přečte a dále zpracuje.

- N4 – Grafické uživatelské rozhraní

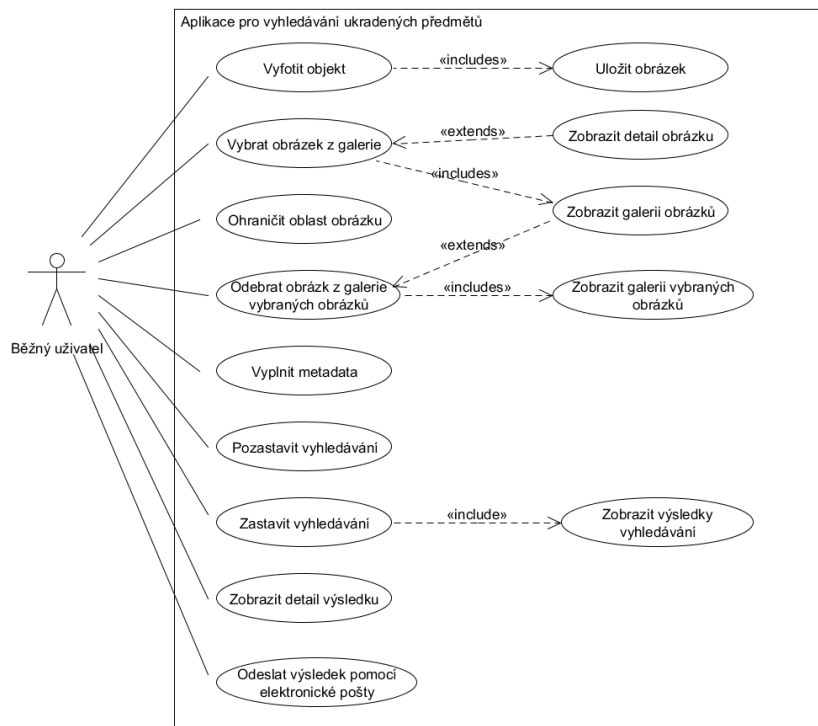
Aplikace bude nabízet takové grafické uživatelské rozhraní, na které jsou uživatelé dotykových mobilních zařízení s operačním systémem Android zvyklí.

- N5 – Lazy loading

Při procházení galerie bude aplikace průběžně načítat jednotlivé obrázky, které mají být uživateli zobrazeny na displeji zařízení, do paměti telefonu. Již načtené obrázky se z paměti nebudou mazat.

2.4 Případy užití

Pro správné pochopení aplikace musí být vytvořeny postupy, které popisují jednotlivé kroky využívající předepsanou funkcionalitu programu. Pro lepší znázornění interakce mezi uživatelem a aplikací jsou vytvořeny diagramy případů užití pro role běžného uživatele (obr. 2.2), správce (obr. 2.3) a příjemce zprávy (obr. 2.4).

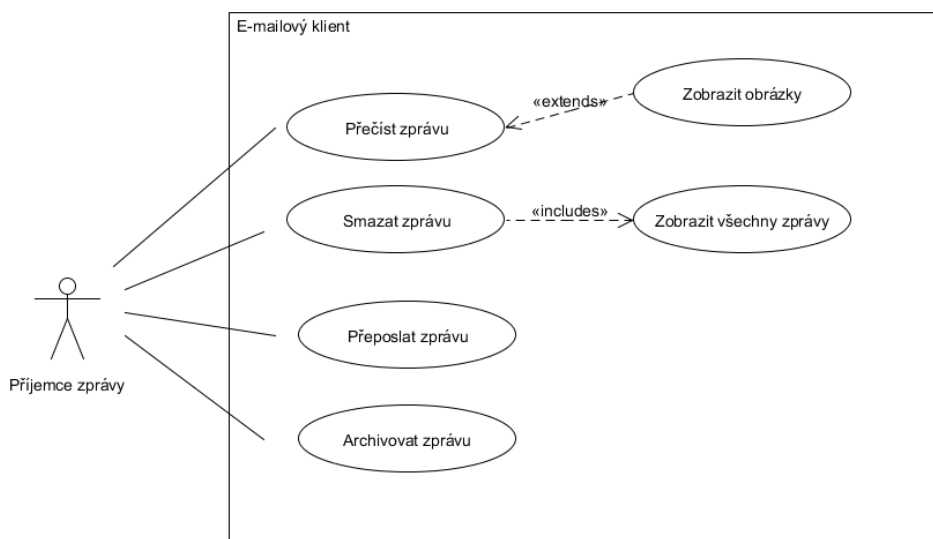


Obrázek 2.2: Diagram případu užití pro běžného uživatele

2. NÁVRH



Obrázek 2.3: Diagram případu užití pro správce



Obrázek 2.4: Diagram případu užití pro příjemce zprávy

2.4.1 Vyfocení předmětu

Umožňuje pořídit fotografii předmětu určeného primárně k odeslání a porovnání.

1. Po zapnutí aplikace uživatel vybere možnost focení objektu pomocí tlačítka „Kamera“.
2. Vybráním této volby se otevře fotoaparát, který umožní pořídit obrázky předmětu.
3. Pomocí tlačítka „Vyfotit“ se pořídí fotografie a uloží se do složky DCIM.
4. Uživatel může po zpracování a uložení vyfocené fotografie pořizovat další snímky (například pokud bude chtít odeslat více obrázků stejného předmětu ke zpracování a tím zvýšit pravděpodobnost nalezení správného výsledku).
5. Pokud se uživatel rozhodne pokračovat na další krok, stačí stisknout šipku symbolizující pokračování vpřed.

2.4.2 Vybrání obrázku z galerie

Umožňuje si prohlédnout galerii pořízených obrázků, detail jednotlivých fotografií a vybrání těch, které uživatel bude chtít odeslat.

1. Po zapnutí aplikace uživatel vybere možnost zobrazení galerie obrázků pomocí tlačítka „Galerie“.
2. Vybráním této volby se zobrazí galerie pořízených obrázků a fotografií, která čerpá data z externího úložiště zařízení.
3. V případě, že je ve složce více fotografií, než se vejde na obrazovku, se uživatel bude moci v galerii vertikálně posouvat.
4. Kliknutím na položku v galerii se fotografie zobrazí v maximální podobě, která se vejde na obrazovku bez ořezání.
5. Pomocí gesta dvou prstů si uživatel bude moci fotografii přiblížit nebo oddálit.
6. Pokud bude uživatel dále chtít s nějakou fotografií pracovat, označí ji.
7. Dokud nebude vybrána alespoň jedna fotografie, nemůže uživatel pokračovat ve vytváření požadavku.

Alternativa:

1. Uživatel vybere po zapnutí aplikace možnost „Kamera“.
2. Dalším krokem je stisknutí tlačítka znázorňující pokračování.
3. Následující kroky jsou stejné jako v základním případě od kroku 2.

2.4.3 Ohraničení oblasti obrázku

Umožňuje ohraničení důležité oblasti obrázku, kterou chce uživatel poslat na porovnání.

1. Nejprve si uživatel vybere alespoň jeden obrázek z galerie všech fotografií a teprve poté bude moci pokračovat na další krok.
2. Aplikace zobrazí fotografie, které uživatel v předešlém kroku vybral. Pokud bude vybraných obrázků více, než se vejde na obrazovku, uživatel se bude moci vertikálně posouvat mezi jednotlivými položkami (stejně jako při vybírání obrázků).
3. Po kliknutí na jednotlivé fotografie aplikace zobrazí jejich maximální podobu, která se vejde na obrazovku bez ořezání.
4. Pokud není vybraná žádná oblast, uživatel pomocí prstu vybere takovou, která zachycuje objekt, jenž chce dále poslat na porovnání.
5. Po výběru oblasti uživatelem aplikace zkontroluje, zda se čára nikde nekříží, a pokud k tomu dojde, celá oblast se vymaže a uživatel bude mít možnost nějakou opět vybrat. Oblast se automaticky uzavře nejkratší možnou úsečkou, která také projde kontrolou křížení. Výsledné ohraničení má podobu jedné uzavřené čáry.
6. Pokud uživatel úspěšně dokončí ohraničení oblasti, může ještě oblast upravit buď přidáním oblasti, nebo naopak odebráním. Po dokončení ohraničení se automaticky zvolí možnost přidat oblast.
7. Jestliže chce uživatel přidat oblast k již vybrané, musí dodržet pravidla stejná, jako při vybírání oblasti samotné (čára nesmí překřížit sebe samu) a navíc musí výběr začínat i končit uvnitř již existující oblasti, aby byla dodržena podmínka jedné uzavřené oblasti.
8. Pokud uživatel vybral část obrázku, kterou dále nechce mít obsaženou ve výběru, vybere nástroj na odebrání oblasti. Opět platí pravidlo nepřekřížení jako v minulém případě. Pro dodržení celistvosti výběru bude muset čára začínat i končit mimo existující výběr.
9. Může nastat i případ, kdy se uživatel rozhodne celý výběr smazat a označit oblast znovu. V tom případě stačí stisknout tlačítko pro vymazání oblasti a uživatel může rovnou vybrat novou oblast.
10. Pro uložení výběru musí uživatel stisknout tlačítko symbolizující pokračování a dostane se zpět na galerii vybraných obrázků, kde budou zobrazeny fotografie i s ohraničením. Pokud ho však nebude chtít uložit, stačí stisknout tlačítko pro návrat na předešlý krok.

2.4.4 Vyplnění metadat

Umožňuje přidat doplňující textové informace k upřesnění popisu daného předmětu.

1. Po vybrání a ohraničení obrázků se uživateli zobrazí dialogové okno, kde si vybere, jaký předmět uživatel vyfotil a pošle na porovnání.
2. V dalším kroku se zobrazí obrazovka s příloženými informacemi. Podle zvolené možnosti v dialogovém okně se uživateli nabídne výběr z nabídky atributů, ke kterým uživatel doplní jejich hodnotu. Jedná-li se o hodnotu, kterou lze vyjádřit číslem, uživatel bude moci zadat pouze číselnou hodnotu. Hodnota může být ještě upřesněna fyzikální jednotkou. Další možností výběru hodnoty je vytvoření seznamu. V ostatních případech uživatel napíše textovou hodnotu.
3. Jestli uživatel bude znát více než jednu informaci týkající se předmětu, může pomocí tlačítka přidat další atribut, ke kterému bude moci opět přidat hodnotu. Stejně tak, když se uživatel rozmyslí a bude chtít odebrat jakýkoli atribut, ho může smazat.
4. Pokud uživatel smaže poslední atribut, otevře se mu opět dialogové okno, kde si bude moci vybrat z kategorií, do kterých předmět zapadá.
5. Atribut si uživatel bude moci vytvořit, pokud si z předem navolených nevybere. Potom se mu otevře dialogové okno, kde si vybere, zda chce zadat číselnou či textovou hodnotu. Při výběru číselné hodnoty se automaticky přidá i okénko, do kterého napíše zvolenou jednotku, pokud bude mít smysl nějakou zadat.

2.4.5 Přerušování vyhledávání

Umožňuje pozastavit vyhledávání vytvořeného požadavku.

1. Aby mohlo být vyhledávání přerušeno, nejdříve musí být odeslán požadavek na server.
2. Po odeslání se uživateli zobrazí obrazovka s průběhem vyhledávání, kde bude mít možnost přerušit vyhledávání pomocí tlačítka „Přerušit“.
3. Serverová aplikace dostane příkaz a pozastaví vyhledávání.
4. Pro pokračování uživatel bude moci na stejné stránce stisknout tlačítko „Pokračovat“. Vyhledávání bude pokračovat v místě, kde bylo přerušeno.

2.4.6 Zastavení vyhledávání

Umožňuje zastavit vyhledávání vytvořeného požadavku. Princip je zde podobný jako u přerušování vyhledávání.

1. Aby mohlo být vyhledávání zastaveno, nejdříve musí být odeslán požadavek na server.
2. Po odeslání se uživateli zobrazí obrazovka s průběhem vyhledávání, kde bude mít možnost zastavit vyhledávání pomocí tlačítka „Zastavit“.
3. Serverová aplikace dostane příkaz a ukončí vyhledávání.
4. Aplikace vrátí uživatele na krok s vyplňováním metadat.

2.4.7 Zobrazení výsledku

Umožňuje zobrazit výsledek vyhledávání a jeho detail.

1. Uživateli se zobrazí výsledky přijaté ze serveru jako jednotlivé položky v seznamu, kde uvidí obrázek, míru shody znázorněnou hvězdičkami a přidružené informace týkající se předmětu.
2. Po kliknutí na libovolný výsledek se zobrazí jeho detail v podobě fotografie rozložené na obrazovce bez ořezání a přidružených metadat.

2.4.8 Odeslání výsledků pomocí elektronické pošty

Umožňuje odeslat vybrané výsledky pomocí elektronické pošty.

1. Uživatel vybere z nabídky výsledků takové obrázky, které bude chtít odeslat jinému uživateli.
2. Po vybrání všech obrázků přejde pomocí tlačítka znázorňující odeslání na obrazovku, kde zadá emailovou (Electronic mail) adresu příjemce. Tu musí vyplnit, aby zpráva mohla být odeslána, dále může vyplnit předmět nebo přidat libovolný textový obsah.
3. Součástí elektronické zprávy budou samozřejmě i URL odkazy na obrázky, které uživatel vybral v prvním kroku. Odkazy budou součástí obsahu emailu a budou vloženy před přidaný text v těle zprávy.

Tabulka 2.1: Rozšířenost operačních systémů[14]

Operační systém	%
Android	81,2
iOS	15,0
Windows Phone	3,0
ostatní	0,7

2.5 Výběr platformy

Důležité je určit, pro který operační systém bude nejuvhodnější aplikaci vyvinout.

Podle tabulky 2.1 je vidět, že Microsoft Phone a ostatní operační systémy jsou velmi málo rozšířené, a proto nebudou podrobněji rozebrány.

Významným operačním systémem na trhu je systém iOS od firmy Apple, od kterého konkurence často kopíruje nové funkcionality. Tento systém je známý svou uzavřeností, kterou lze považovat za pozitivum i negativum podle toho, jak je na ni nahlíženo. Jednoznačným kladem je bezpečnost, kterou zařízení s iOS disponují. Pro uživatele je však systém hůře přístupný.

Google Android je v přístupu přesným opakem předešlého systému. Tvůrci tohoto operačního systému vsadili na otevřenost, tudíž je možné, že si stažením neznámého softwaru do zařízení uživatel nainstaluje i nebezpečný vir. Avšak práce s obsahem systému je přímější a není potřeba podpůrných aplikací jako je tomu v případě iOS, kde lze přesouvat soubory z mobilního zařízení a počítače pouze pomocí aplikace iTunes[15].

Po shrnutí je vidět, že operační systémy iOS a Android sází na opačné přístupy (jeden je otevřený, druhý uzavřený). Každý má své výhody i nevýhody. Funkcionalitu aplikace však ani jeden z nich neomezuje, proto byl výběr určen podle rozšířenosti systému. V tom případě je jasnou volbou operační systém Android.

2.6 Operační systém Android

Android je nejrozšířenějším operačním systémem na mobilních zařízeních jako jsou mobilní telefony a tablety. Dokonce ho najdeme i v některých novějších televizorech. Jedním z důvodů, proč je tento systém tak oblíbený, je cenová dostupnost některých zařízení, která ho využívají.

Dalším kladem je otevřenost systému. Tímto způsobem přitáhl velkou spoustu vývojářů, kteří přispívají svým dílem k oblíbenosti platformy. A naopak, více prodaných telefonů znamená větší množství stažené aplikace a ve většině případů i větší zisk programátorům – oboustranně prospěšná spolupráce[16].

2. NÁVRH

Jako u každého systému, který se neustále vyvíjí, se i u systému Android vytváří nové verze přinářející nové funkce, knihovny a rozhraní. S průběžným vývojem by časem mohl přijít problém v podobě nekompatibility novějších zařízení se staršími aplikacemi. Avšak díky zpětné kompatibilitě se těmto komplikacím operační systém lehce vyhne, protože nové verze obsahují stejné funkce jako ty předešlé. Ovšem je nutné si nejdříve určit, od jaké verze operačního systému by měla aplikace fungovat bez potíží, protože vyšší verze mohou obsahovat funkce, které nižší nepodporují[17]. V mém případě jsem si zvolil dolní hranici verzi 4.0.

2.7 Vývoj vedoucí k operačnímu systému Android

V minulosti, kdy se vyvíjelo výhradně v nízko-úrovňovém jazyce C nebo C++, musel vývojář dopředu znát technickou stránku zařízení, pro které vyvíjel aplikace. Jak časem technologie postoupila, stal se tento přístup zastaralým. Platformy jako je třeba Symbian tento problém částečně vyřešily, avšak potřebovaly komplexní kód v C/C++ a těžkopádné aplikační rozhraní, aby mohly být vyvíjeny pro více druhů zařízení.

Velkým pokrokem bylo představení MIDletu (aplikace využívající MIDP – Mobile information device profile), které jsou spouštěny na JVM (Java virtual machine), na procesu tvářícím se jako hardwarová vrstva a umožňující vytvářet aplikace běžící na různých zařízeních. Bohužel tato výhoda má i druhou stranu, a to omezený přístup k hardwaru zařízení.

Tento operační systém nabízí nové možnosti pro mobilní aplikace otevřeným vývojovým prostředím postaveném na otevřeném Linuxovém jádře. Přístup k hardwaru je umožněn všem aplikacím skrze knihovny aplikačních rozhraní.

V Androidu si jsou všechny aplikace rovnocenné. Jak aplikace třetích stran, tak původní Android aplikace jsou psány se stejnými aplikačními rozhraními a spouštěny ve stejném běhovém prostředí. Uživatel může odstranit a nahradit jakékoli původní aplikace jinou alternativou.

2.8 Architektura operačního systému Android

Operační systém Android je postaven na pěti základních elementech, jádru, knihovnách, běhovém prostředí, rámci aplikace a aplikační vrstvě:

- Jádro systému je nejnižší vrstvou v celé architektuře. Vytváří abstraktní vrstvu mezi hardwarem a zbylými čtyřmi vyššími vrstvami. Dalšími vlastnostmi této vrstvy jsou zabudované ovladače, podpora správy sítě, paměti, procesu, bezpečnosti a energie.
- Knihovny operačního systému Android jsou další vrstvou nad jádrem. Knihovny jsou napsané v jazyce C/C++ jako jsou knihovny libc, OpenSSL.
- Běhové prostředí Android je třetí vrstvou, bez které by nebyl mobilní telefon s operačním systémem Android takový, jaký ho známe dnes. Tato vrstva obsahuje jádro knihoven a DVM (Dalvik Virtual Machine). DVM je registrově orientovaný virtuální stroj, který umožňuje běh několika instancí aplikace na jednom zařízení.
- Rámec aplikace zprostředkovává mnoho služeb, které vývojář využívá ve své aplikaci v podobě Java tříd. Obsahuje například nástroje na vy-

Tabulka 2.2: Verze operačního systému Android[19]

Verze	Název	API	Rozšířenost
2.2	Froyo	8	0,4 %
2.3.3 - 2.3.7	Gingerbread	10	6,4 %
4.0.3 - 4.0.4	Ice Cream Sandwich	15	5,7 %
4.1.x	Jelly Bean	16	16,5 %
4.2.x		17	18,6 %
4.3		18	5,6 %
4.4	KitKat	19	41,4 %
5.0	Lollipop	21	5,0 %
5.1		22	0,4 %

tváření uživatelského rozhraní, nástroje pro spouštění jiných aktivit či aplikací nebo služeb, které běží na pozadí mobilního telefonu.

- Aplikační vrstva je poslední a nejvyšší vrstvou výše zmíněné architektury. V této vrstvě jsou všechny aplikace, jak původní, tak stažené, nainstalovány. Všechny aplikace na této vrstvě využívají stejné aplikační rozhraní a využívají třídy z rámce aplikace[18].

2.9 Verze operačního systému Android

Pro lepší přehlednost jsou jednotlivé verze systému Android, jejich rozšířenost, název a API zobrazeny v tabulce 2.2.

Dále následuje výčet verzí a úprav, které jednotlivé verze přinesly[20].

2.9.1 Froyo

- Optimalizace rychlosti a paměti
- Možnost vytvořit WiFi (Wireless fidelity) hotspot
- Možnost instalace aplikací na paměťovou kartu
- Zvýšení rychlosti díky JIT (Just in time) kompilátoru
- Podpora numerických a alfanumerických hesel

2.9.2 Gingerbread

- Upravená virtuální klávesnice
- Podpora SIP (Session initiation protocol) pro volání přes internet
- Podpora videohovorů

- Podpora senzorů jako je gyroskop nebo barometr
- Zlepšení správy prostředků

2.9.3 Ice Cream Sandwich

- Vylepšení rozeznávání hlasu
- Rozpoznávání podle obličeje/odemykání telefonu obličejem
- Vylepšení fotoaparátu (pořízení panorama, přibližování popřípadě oddalování při natáčení)
- Možnost přistupovat k aplikacím přímo ze zamčené obrazovky
- Android Beam (funkce umožňující odesílání dat pomocí NFC – Near field communication)

2.9.4 Jelly Bean

- Podpora OpenGL
- Vylepšení fotoaparátu (pomocí Photo Sphere lze pořizovat 360° fotografie)
- Schopnost přidávat ovládací prvky bez nutnosti práv správce zařízení
- Podpora více uživatelských účtů

2.9.5 KitKat

- Omezení přístupu aplikacím k externímu úložišti
- Optimalizace zařízení se slabým výkonem
- Zakázán přístup třetím stranám ke statistickým údajům o baterii
- Představen Android Runtime

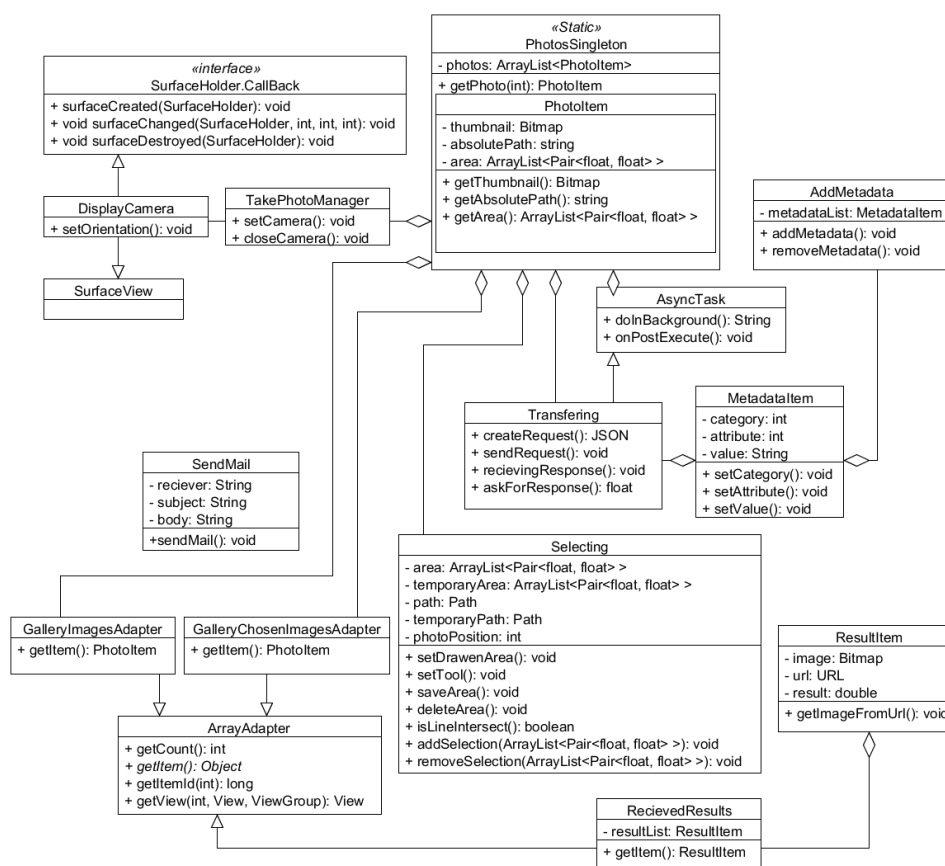
2.9.6 Lollipop

- Optimalizace Android Runtime (například garbage collector)
- Podpora 64 bitových procesorů
- Podpora používání více SIM karet najednou

2.10 Diagram tříd

Součástí návrhu je diagram tříd (obr. 2.5) a jejich krátký popis.

2. NÁVRH



Obrázek 2.5: Schéma tříd

2.10.1 PhotoItem

Tato třída je implementována jako singleton, tedy po jejím zavolání existuje její instance celou dobu běhu aplikace. Tato instance obsahuje pole držící informace o jednotlivých obrázcích (samotný obrázek, jméno, absolutní cestu vedoucí k souboru, ohraničenou oblast). Tato třída je používána pro zobrazení obrázků v galerii.

2.10.2 MetadataItem

Třída MetadataItem reprezentuje položky v seznamu, jež obsahují jednotlivé informace blíže popisující předmět.

2.10.3 ResultItem

ResultItem reprezentuje položky, které klientská aplikace přijme ze serveru. Výsledek obsahuje obrázek předmětu, URL adresu obrázku, jeho ohodnocení

a další atributy popisující objekt.

2.10.4 DisplayCamera

Abychom mohli fotit, je potřeba zobrazovat to, co je vidět pomocí vestavěného fotoaparátu. Pro tento účel je vytvořena třída DisplayCamera, která dědí z třídy SurfaceView, díky které lze vykreslovat obrázky, a která implementuje rozhraní SurfaceHolder.CallBack, jež umožňuje pořádit obrázek.

2.10.5 GalleryImagesAdapter, GalleryChosenImagesAdapter

Tyto dvě třídy dědí ze stejného předka a to z třídy ArrayAdapter, pomocí které uživateli zobrazují na obrazovce jednotlivé položky.

2.10.6 Selecting

Třída, která umožňuje ohraničit relevantní oblast obrázku. Selecting obsahuje metody na ohraničení, přidání oblasti, odebrání oblasti, smazání celého ohraničení, uložení ohraničení a kontrolu, zda se čára nekříží.

2.10.7 Transferring

Tato třída obstarává komunikaci se serverem, tedy odesílá požadavky v podobě JSON řetězců pomocí HTTP metod a přijímá odpovědi ve stejné formě jako požadavky. Transferring dědí z třídy AsyncTask, která umožňuje asynchronní práci dvou vláken. V tomto případě jde o hlavní vlákno a vlákno obstarávající komunikaci se serverem.

2.10.8 AddMetadata

Pro vytvoření a přidání jednotlivých metadat, které popisují hledaný předmět, slouží třída AddMetadata. Třída obsahuje list jednotlivých položek (MetadataItem), které obsahují informace v podobě klíče a odpovídající hodnoty.

2.10.9 RecievedResults

Třída RecievedResults, která dědí z třídy ArrayAdapter, zobrazuje přijaté výsledky ze serveru.

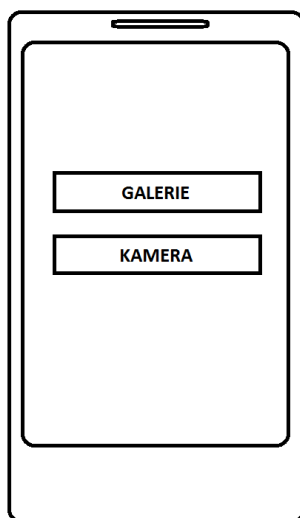
2.10.10 SendMail

Tato třída zprostředkovává odeslání vybraných fotografií pomocí elektronické pošty.

2.11 Návrh grafického rozhraní

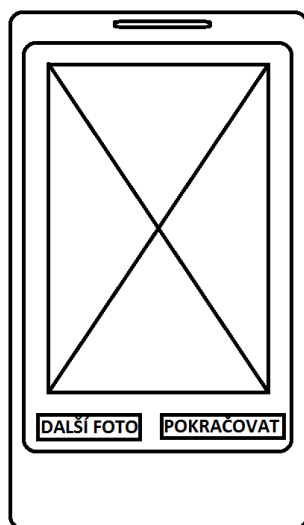
Důležitým faktorem návrhu aplikace je její grafické rozhraní. Návrh jednotlivých obrazovek je nezbytný pro konzultaci se zadavatelem. Často se tím předejde nesrovnalostem a všichni si mohou jasně představit, jak bude aplikace vypadat, komunikovat, a mohou ji upravovat podle potřeb.

Prvotní návrh grafického rozhraní vyvíjené aplikace vypadal následovně.



Obrázek 2.6: Návrh úvodní obrazovky

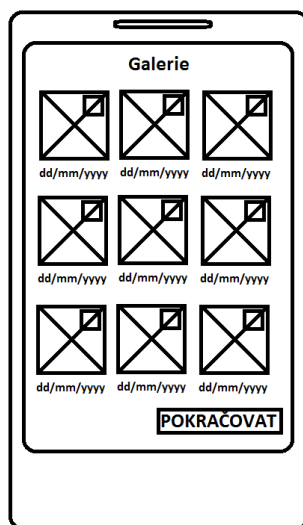
Při zapnutí aplikace uživatel vybere buď zapnutí fotoaparátu nebo galerii již vyfocených předmětů (obr. 2.6).



Obrázek 2.7: Návrh fotoaparátu

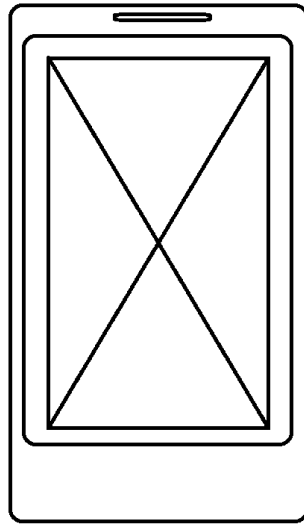
Po stisknutí tlačítka „Kamera“ se zobrazí fotoaparát (obr. 2.7) a uživatel může pořizovat fotografie. Zde uživatel pořídí fotografii, která se mu zobrazí na displeji. Poté se rozhodne, jestli pokračovat na další krok, nebo pořídít další.

2. NÁVRH



Obrázek 2.8: Návrh galerie obrázků

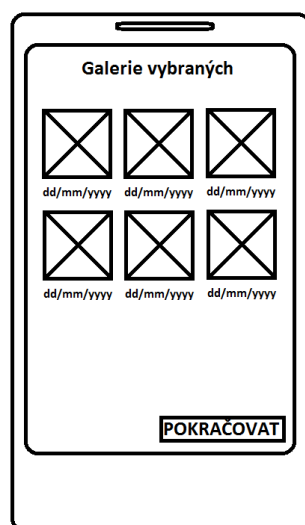
Na této obrazovce (obr. 2.8) uživatel vybere obrázky, které chce odeslat na porovnání s databází. Pokud bude chtít, může také zobrazit detail jednotlivých položek. Položky jsou seřazeny od nejnovější po nejstarší podle času a data pořízení.



Obrázek 2.9: Návrh zobrazení detailu fotografie z galerie

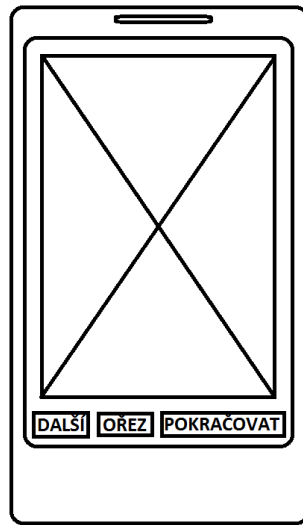
Po kliknutí na jakoukoli položku se zobrazí její podoba v co nejvyšší velikosti bez nutnosti ořezání (obr.2.9).

2. NÁVRH



Obrázek 2.10: Návrh galerie vybraných obrázků

Zde se uživateli zobrazí vybrané položky z galerie všech obrázků z předešlého kroku (obr. 2.10). Po vybraní položky se mu otevře možnost objekt na fotografii ohraničit.



Obrázek 2.11: Návrh ohraničení objektu na fotografii

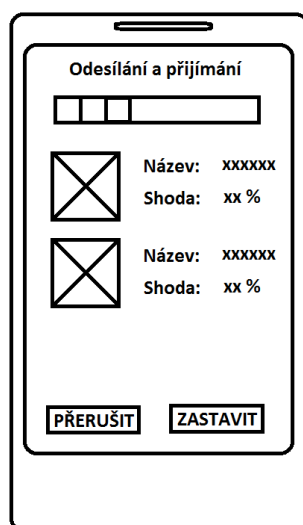
Ohraničit fotografii může uživatel pomocí doteku. Obrázek se zobrazí v co největší velikosti bez nutnosti ohraničení (obr. 2.11).

The image shows a mobile application interface for entering metadata. The form is titled "Metadata" and contains the following elements:

- A dropdown menu labeled "Dílo" with a right-pointing arrow.
- A dropdown menu labeled "Socha" with a right-pointing arrow and a red "X" icon to its right.
- A dropdown menu labeled "Výška" with a right-pointing arrow.
- A text input field containing the number "20" followed by the unit "cm" and a red "X" icon to its right.
- A green plus sign (+) icon below the input fields.
- A button labeled "POKRAČOVAT" at the bottom right of the form.

Obrázek 2.12: Návrh vyplnění metadat k předmětu

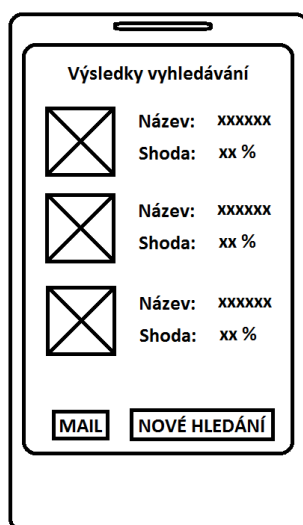
Po vybrání fotografií a ohrazení jejich podstatných částí uživatel může přidat informace o objektu, které upřesní vyhledávání (obr. 2.12).



Obrázek 2.13: Návrh obrazovky při přijímání výsledků

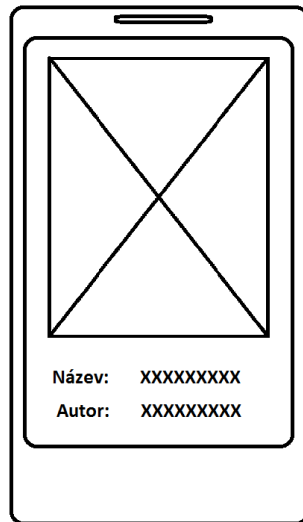
Během přijímání se uživateli zobrazují průběžné výsledky i průběh celkového zpracování požadavku na serveru (obr. 2.13). Dále může být vyhodnocování ukončeno, nebo přerušeno.

2. NÁVRH



Obrázek 2.14: Návrh obrazovky po vyhodnocení výsledků a jejich zobrazení

Jakmile vyhodnocení dosáhne 100 %, zobrazí se uživateli množina (o předem dané velikosti) výsledků seřazená od největší podobnosti obrázků po nejmenší (obr. 2.14).



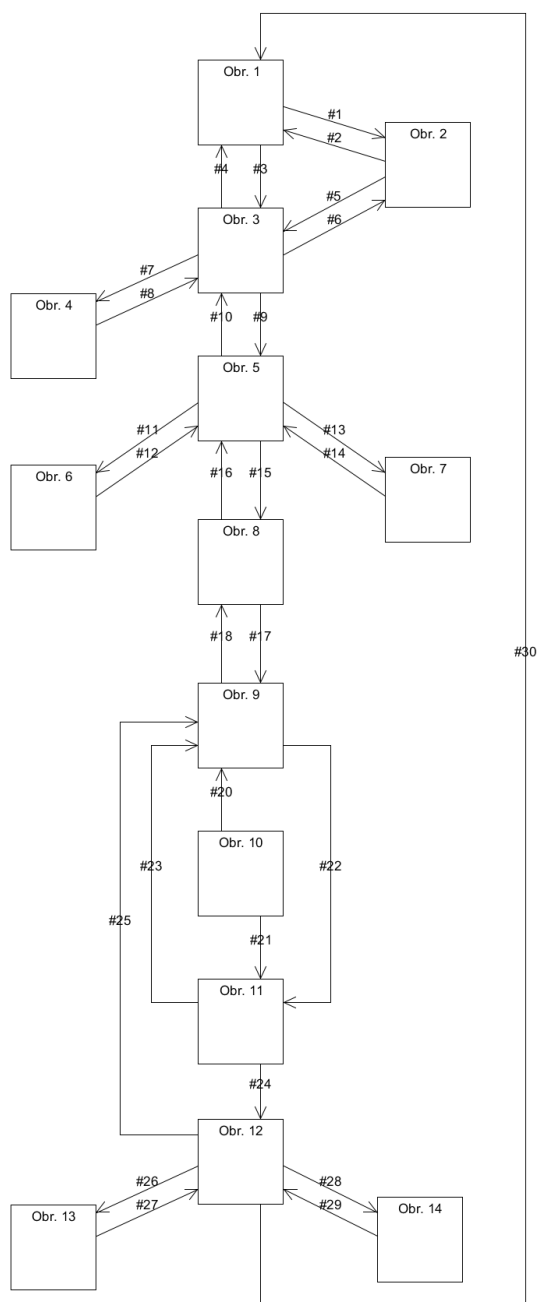
Obrázek 2.15: Návrh zobrazení detailu přijatého výsledku

Po kliknutí na výsledek v seznamu výsledků se uživateli zobrazí jeho detail obsahující fotografii a přidružené informace popisující předmět (obr. 2.15).

2.12 Průchod aplikací

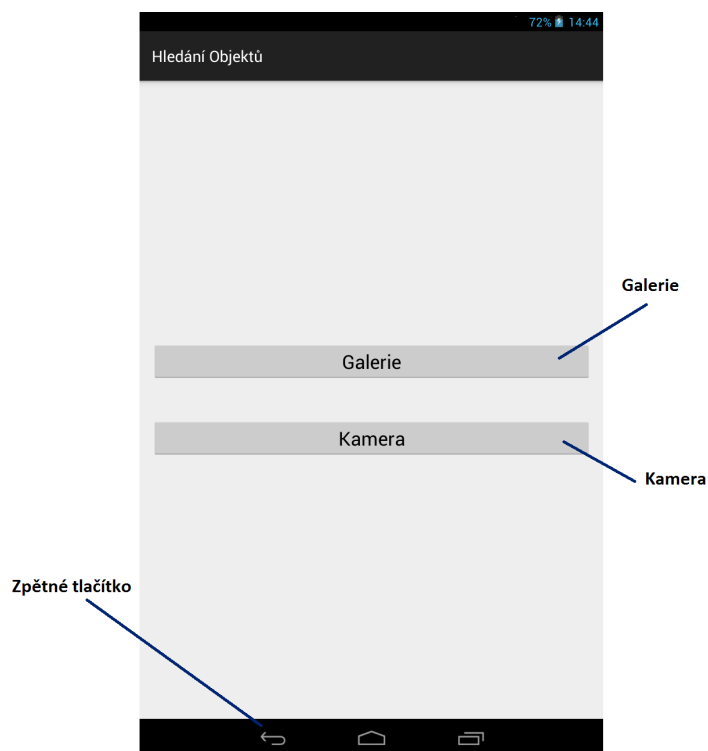
Pro lepší představu průchodu aplikací je přiložen graf s označenými přechody mezi jednotlivými obrazovkami (obr. 2.16).

2. NÁVRH



Obrázek 2.16: Přechody

Po spuštění aplikace bude mít uživatel na výběr ze dvou možností: „Galerie“ a „Kamera“ (obr. 2.17). Při jakémkoli výběru budou následovat další kroky

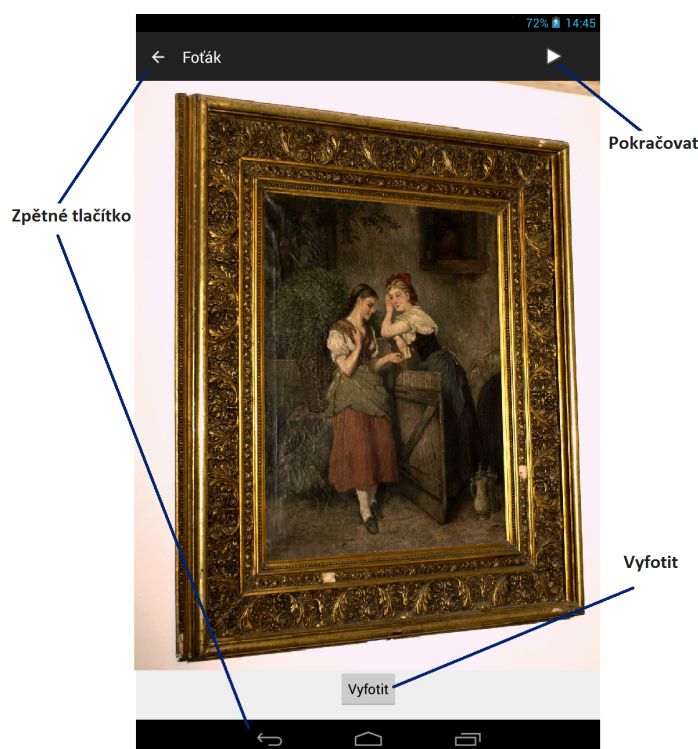


Obrázek 2.17: Úvodní obrazovka

pro získání výsledku.

- Zpětné tlačítko – Stisknutím se aplikace vypne.
- Galerie (přechod #3) – Po stisknutí tlačítka se zobrazí galerie s obrázky uloženými ve složce DCIM.
- Kamera (přechod #1) – Po stisknutí tlačítka se otevře kamera a uživatel bude moci vyfotit podezřelý předmět.

2. NÁVRH

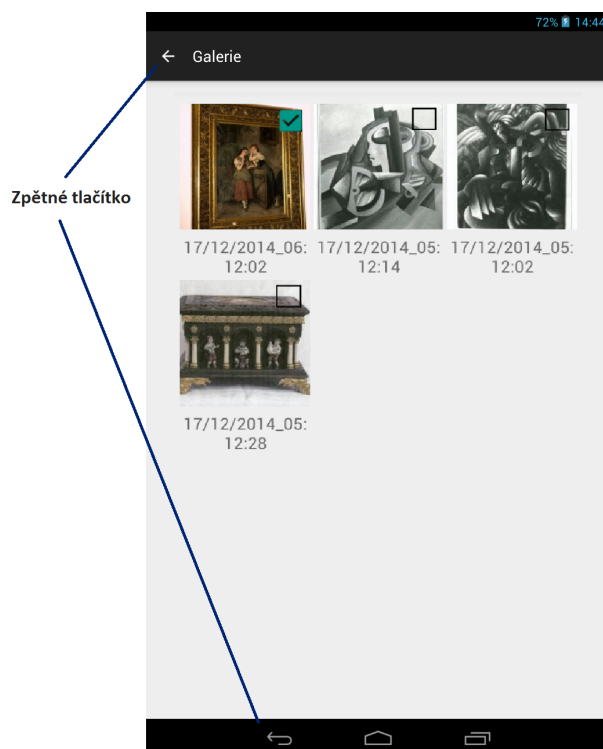


Obrázek 2.18: Obrazovka s fotoaparátem připraveným k focení

Pokud uživatel vybere možnost „Kamera“, otevře se fotoaparát na zadní straně mobilního zařízení, pomocí něž je možné pořídit obrázek cennosti, který se poté uloží do složky DCIM (obr. 2.18). Objekt může být vyfocen libovolně krát za sebou (například z různých úhlů) pro zvětšení pravděpodobnosti nálezu nejpodobnějšího výsledku.

Pokračováním se uživatel dostane na galerii obrázků v telefonu, stejně jako kdyby na úvodní obrazovce vybral možnost „Galerie“.

- Zpětné tlačítko (přechod #2) – Po stisknutí se uživatel vrátí k úvodnímu výběru „Galerie“ a „Kamera“.
- Vyfotit – Stisknutím tohoto tlačítka uživatel pořídí fotografii zobrazenou na displeji. Po vyfocení a zpracování obrázku je foták opět připraven pořizovat fotografie.
- Pokračovat (přechod #5) – Stisknutím tohoto tlačítka uživatel přejde na další krok, kde se mu zobrazí galerie s fotografiemi uloženými v mobilním zařízení.



Obrázek 2.19: Galerie s fotografiemi

Galerie s fotografiemi v mobilním zařízení se otevře při vybrání „Galerie“ po spuštění aplikace nebo pokud uživatel fotí objekt a rozhodne se pokračovat stisknutím tlačítka „Pokračovat“ na obrazovce se spuštěným fotoaparátem (obr. 2.19). Stisknutím obrázku v galerii se zobrazí tak, aby bez ořezání zabíral 100 % obrazovky.

Na následující obrazovce uživatel vybírá fotografie, které chce v dalších krocích odeslat na server, na němž se pomocí programu porovnají s databází odcizených předmětů.

Pomocí zaškrťovacího tlačítka se označí obrázky určené k odeslání. Kliknutím na miniaturu obrázku se zobrazí v plné velikosti (přechod #7). Pokud uživatel nezaškrtně alespoň jednu položku, nebude moci pokračovat na další krok.

- Zpětné tlačítko (přechod #4 popřípadě #6) – Po stisknutí se uživatel vrátí na úvodní obrazovku, pokud se na nynější krok dostal stisknutím tlačítka „Galerie“, nebo se vrátí na obrazovku s foťákem, kde bude schopen opět pořizovat fotografie, pokud se na nynější krok dostal pomocí „Pokračovat“ po focení.
- Pokračovat (přechod #9) – Stisknutím tohoto tlačítka se uživatel do-

2. NÁVRH

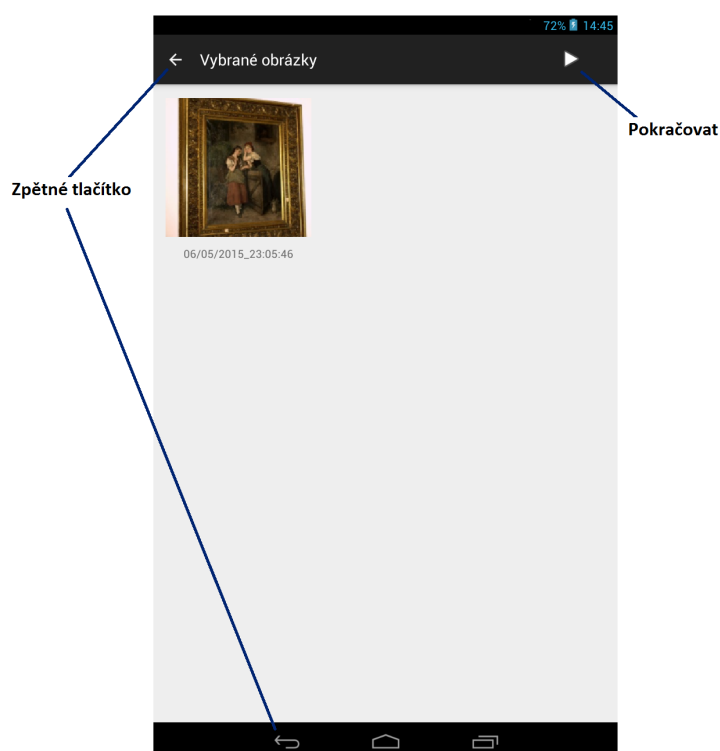
stane na další krok, kterým je galerie vybraných obrázků. Pokud není označena ani jedna fotografie, tlačítko není zobrazeno a nelze pokračovat.



Obrázek 2.20: Detail fotografie

Pokud uživatel klikne na fotografii v galerii, zobrazí se na obrazovce v plné velikosti – fotografie bude zobrazena tak, aby se celá vešla na obrazovku a nebyla nikde ořezána (obr. 2.20). Fotografie může být přiblížena nebo oddálena pomocí dvou prstů.

- Zpětné tlačítko (přechod #8) – Po stisknutí se zobrazí galerie se všemi obrázky zaškrtnutými tak, jak byly těsně před otevřením obrázku.



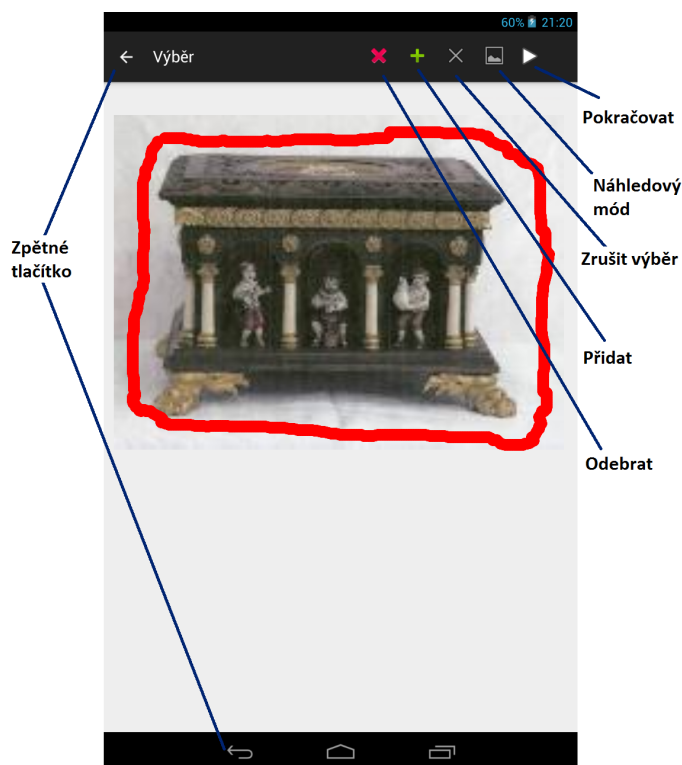
Obrázek 2.21: Galerie s vybranými fotografiemi k porovnání

V případě, že uživatel bude po označení obrázků určených k porovnání pokračovat, zobrazí se opět galerie, ale tentokrát jenom s obrázky, které byly určeny ke zpracování (obr. 2.21). Zde mohou být ještě upraveny nebo odebrány ze seznamu.

Přejití na další krok uživateli umožní vyplnit metadata související s obrázky vybraného objektu, která se vloží do požadavku.

Kliknutím na obrázek se otevře plná velikost (obrázek bude tak velký, aby se vešel na celou obrazovku bez ořezání) fotografie a uživatel bude moci označit pouze relevantní oblast obrázku (přechod #11 popřípadě #13). Ohraničená fotografie se pozná podle toho, že na miniatuře bude zobrazen výběr, který uživatel provedl.

- Zpětné tlačítko (přechod #10) – Po stisknutí se uživateli zobrazí galerie se všemi obrázky označenými tak, jak byly vybrány těsně před pokračováním na obrazovku s galerií vybraných obrázků. Pokud byl obrázek odebrán ze seznamu, nebude v galerii zaškrtnut.
- Pokračovat (přechod #15) – Po stisknutí se uživatel dostane na další krok, kterým je doplnění informací o obrázku.



Obrázek 2.22: Obrazovka s ohraničením oblasti fotografie

Pokud uživatel klikne na obrázek v galerii vybraných fotografií, zobrazí se mu fotografie s možností výběru oblasti (obr. 2.22). Každý objekt by měl být v obrázku ohraničen kvůli zvýšení pravděpodobnosti najetí nejpodobnějšího obrázku v databázi. Jestliže zůstane fotografie bez ohraničení objektu, může pozadí zbytečně znepřesnit porovnávání a konečný výsledek. Uživatel bude obrázek označovat pomocí tahu prstu, kterým ohraničí oblast, kterou považuje za důležitou a kterou bude chtít odeslat ke zpracování. Obrázek s ohraničením bude v galerii označen. Po zvednutí prstu se ohraničení dokončí pomocí nejkratší úsečky mezi počátečním a koncovým bodem, která oblast uzavře. Pokud se ohraničení bude křížit, po oddálení prstu zmizí.

Pokud uživatel není s vybranou oblastí spokojen, může celý výběr jednoduše zrušit a provést znovu, anebo pomocí panelu nástrojů výsledný obrazec upravit. Bude mít na výběr ze dvou možností úpravy: přidat oblast, odebrat oblast.

- Přidat oblast: Pro přidání nového výběru k již označené oblasti na fotografii uživatel vybere v panelu tlačítko umožňující tuto možnost. Nyní uživatel tahem prstu z vyznačené oblasti ohraničí novou oblast, kterou bude chtít připojit k té původní, a vrátí se zpět do již vyznačené části. Přidaný výběr bude ohraničen protínajícími se body, čarou mezi těmito

2. NÁVRH

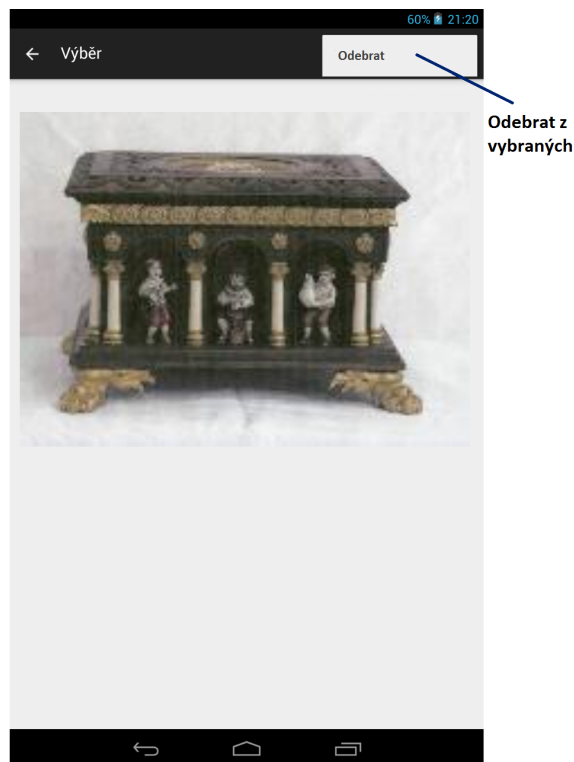
body a nově nakreslenou hranicí oblasti. Pokud se nově nakreslená křivka neprotne se stávající, jak je popsáno, tah zmizí a přidání se neprovede. Pokud se křivka protne sama se sebou, opět zmizí a požadovaná akce se neprovede.

- Odebrání oblasti: Pro odebrání části ohraničení fotografie uživatel vybere tlačítko k tomu určené, jako tomu bylo v předešlém případě (tlačítko se chová obdobným způsobem). K využití této funkcionality se musí pomocí prstu opět protnout nově vytvořená křivka začínající vně existujícího výběru s ohraničením oblasti. Poté uživatel vybere část, kterou bude chtít odebrat z výběru, a vrátí se zpět mimo původní ohraničení. Pro odebírání platí obdobná omezení jako pro přidávání.

Pokud uživatel nebude s výběrem spokojen, bude ho moci pomocí nástrojového panelu upravit. Buď označí oblast, kterou nechce zahrnout do výběru, ta se následně odebere, nebo naopak přidá oblast, kterou bude chtít zahrnout do ohraničení.

Fotografie bude moci být i přiblížena. Po přiblížení či oddálení bude ohraničení obsahovat stejnou část fotografie. Bude reagovat na změny.

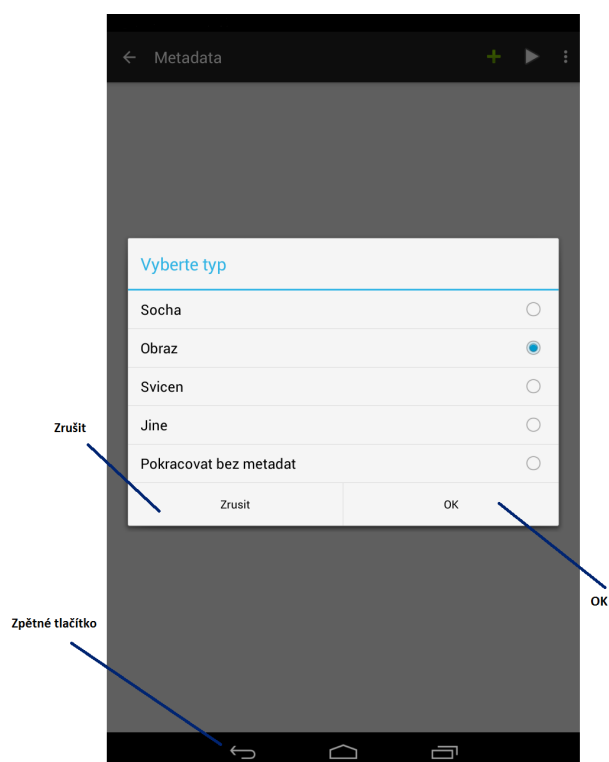
- Zpětné tlačítko (přechod #12) – Po kliknutí se uživatel vrátí na galerii vybraných fotografií a označení, které vytvořil, se zruší.
- Pokračovat (přechod #12) – Po stisknutí se uloží výběr a uživatel se vrátí na galerii vybraných fotografií. Pokud uživatel bude chtít ještě zhlédnout nebo upravit ohraničení, stačí, aby znovu klikl na obrázek, a otevře se náhled i s ohraničením.
- Zrušit výběr – Stisknutím tohoto tlačítka se ohraničení zruší a uživatel může znovu ohraničit oblast.
- Přidat – Po vybrání tohoto nástroje aplikace uživateli umožní vybrat pomocí prstu oblast, kterou chce zahrnout do výběru.
- Odebrat – Výběrem tohoto nástroje aplikace uživateli umožní ohraničit pomocí prstu oblast, kterou nechce zahrnout do výběru.
- Náhledový mód – Po stisknutí se zruší předešlý výběr nástroje (přidání/odebrání oblasti) a uživatel bude moci fotografii přibližovat nebo oddalovat.



Obrázek 2.23: Obrazovka s výběrem pro odebrání fotografie z výběru

Při možnosti ohraničení objektu se po stisknutí třech teček otevře menu s položkou „Odebrat“ (obr. 2.23). Pokud se uživatel rozhodne, že obrázek není vhodný na porovnání, jak si původně myslel, může ho tímto výběrem odebrat z vybraných fotografií.

- Zpětné tlačítko (přechod #14) – Po kliknutí se uživatel vrátí na galerii vybraných fotografií a označení, které vytvořil, se zruší (pokud nějaké vytvořil).
- Odebrat z vybraných – Stisknutím se obrázek odebere z množiny vybraných fotografií určených k porovnání (při vrácení se na výběr obrázků určených k odeslání nebude odebraný obrázek zaškrtnutý). Po odebrání se uživatel vrátí do galerie vybraných fotografií, kde obrázek, který se rozhodl smazat z výběru, není.



Obrázek 2.24: Výběr typu předmětu

Pokračováním se uživateli zobrazí obrazovka s možností přiložit přidavné informace, které mají pomoci s vyhledáváním výsledků. Uživatel musí dříve, než začne vyplňovat samotné údaje, specifikovat o jakou cenu se jedná pomocí jednoduchého dialogového okna a podle toho bude mít na výběr údaje, které k danému objektu lze přiřadit a zároveň jsou relevantní (obr. 2.24).

Pokud si uživatel nevybere ani z jedné kategorie, bude moci pokračovat výběrem „Pokračovat bez metadat“. Tímto způsobem se odešle fotografie na porovnání bez dalších doplňujících informací.

Aplikace bude nabízet i možnost jiného předmětu, respektive nabídne atributy určené pro více než jednu kategorii místo specifických (například počet svíček pro svícen).

- Zpětné tlačítko (přechod #16) – Stisknutím se uživatel dostane na předchozí obrazovku galerie s vybranými obrázky.
- OK (přechod #17) – Zvýrazněná položka se vybere a uživatel bude moci přidávat pouze data upřesňující daný typ ceny.
- Zrušit (přechod #16) – Dialogové okno se zavře a uživatel se dostane na předchozí obrazovku galerie s vybranými obrázky.



Obrázek 2.25: Vyplnění informací popisujících objekt

Před odesláním může uživatel vyplnit textové informace vztahující se k předmětu. Přidružených informací se může odeslat více než jedna.

Po stisknutí „OK“ v dialogovém okně se zobrazí obrazovka se seznamem vlastností (obr. 2.25). Pomocí tlačítka „Přidat“ se do seznamu vloží nový řádek. Uživatel si vybere, o jaký atribut se bude jednat (vlevo), a vedle něj (napravo) se aktualizuje pole na vyplnění textové nebo číselné hodnoty, či vybrání z množiny předvyplněných hodnot.

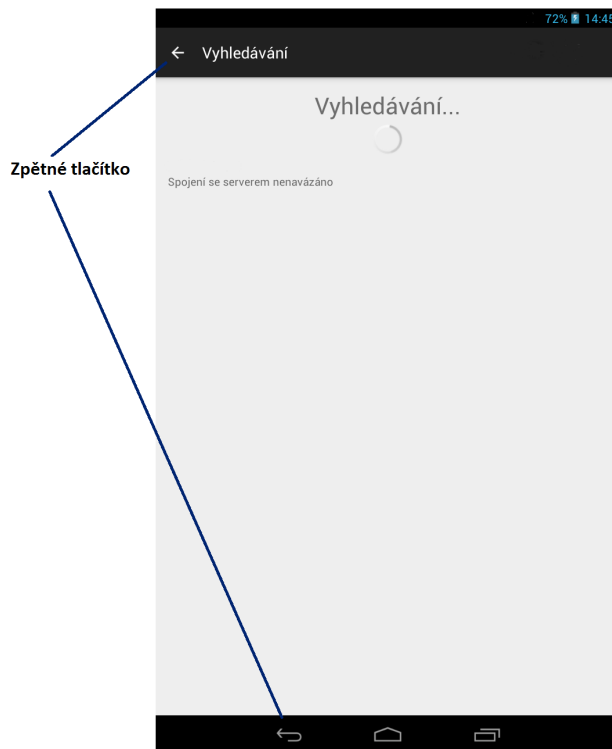
Pokud uživatel vybere možnosti „Jiné“, napravo od seznamu se zobrazí textové pole, kde uživatel vepíše hodnotu atributu „Materiál“. Pokud uživatel bude chtít do pole, kde je třeba vyplnit číselná hodnota (například výška), vepsat textovou odpověď, nebude možné zapsat jakýkoli jiný znak kromě číslice. Pokud se bude jednat o číselný údaj určený ve fyzikálních jednotkách, bude si uživatel moci vybrat, ve kterých danou informaci zadá (například jestli výška bude v centimetrech, metrech, nebo milimetrech).

Pokud dialogové okno uživateli nenabídne atribut, který by rád vyplnil, bude mít možnost vybrat položku s vlastním vyplněním informace. Při výběru se otevře dialogové okno s možností výběru mezi textovou a číselnou formou hodnoty, kterou chce uživatel zadat. Podle toho, kterou možnost vybere, se zobrazí řádek v seznamu s poli, do kterých vepíše atribut, jeho hodnotu a popřípadě jednotku, pokud se bude jednat o číselnou hodnotu. Pokud uživatel

2. NÁVRH

stiskne tlačítko „Zrušit“, nebo klikne vně dialogového okna, atribut řádku, který změnil na „Vlastní“, se nastaví na první hodnotu v seznamu atributu.

- Zpětné tlačítko (přechod #18) – Stisknutím se smažou všechny dosud vyplněné a přidané informace a opět se otevře dialogové okno s výběrem typu předmětu. Okno se bude chovat jako kdyby se na stránku přišlo z předchozího kroku (galerie s vybranými fotografiemi).
- Smazat řádek – Stisknutím tohoto tlačítka se smaže celý řádek, vedle kterého je zobrazen.
- Přidat – Po stisknutí se na konec seznamu přidá nový řádek, na kterém uživatel bude moci vybrat, jaký atribut k obrázku bude chtít přiložit. Pokud se bude jednat o položku, která je jedinečná (například výška obrazu) a která je již vybraná v nějakém z předešlých řádků, nebude se už v seznamu vyskytovat.
- Atribut – V rozbalovacím seznamu si uživatel vybere atribut, který chce přiložit k obrázku jako doplňující informaci.
- Hodnota – Podle atributu se zobrazí pole, do kterého se napíše jakýkoli text (1), nebo číslo (2), nebo se zobrazí rozbalovací okno s možností výběru hodnoty (3).
- Jednotky – Zobrazí se seznam s jednotkami, které doplní celkovou informaci o hodnotě, pokud bude přidání hodnoty veličiny dávat smysl.
- Pokračovat (přechod #19 popřípadě #22) – Po stisknutí tohoto tlačítka se odešle požadavek na server a přejde se na další krok.

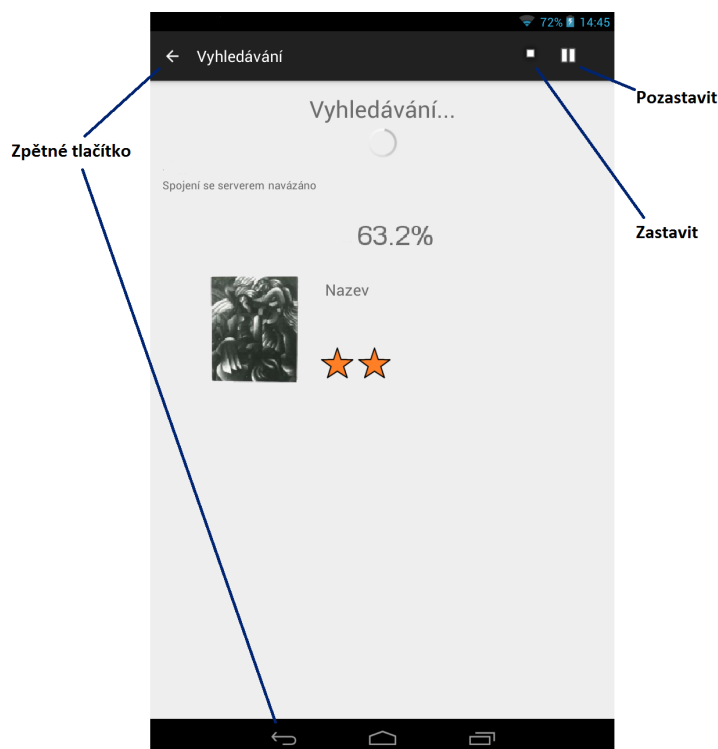


Obrázek 2.26: Odesílání požadavku, pokud zařízení není připojeno k internetu

Obrazovka „Vyhledávání“ s nápisem „Spojení se serverem nenavázáno“ se ukáže, pokud uživatel není připojen k internetu (obr. 2.26). Po navázání spojení započne odesílání a přijímání výsledku (přechod #21).

- Zpětné tlačítko (přechod #20) – Po stisknutí se uživatel vrátí na obrazovku, kde se mu zobrazí metadata, která v předešlém kroku vyplnil a připojil k požadavku.

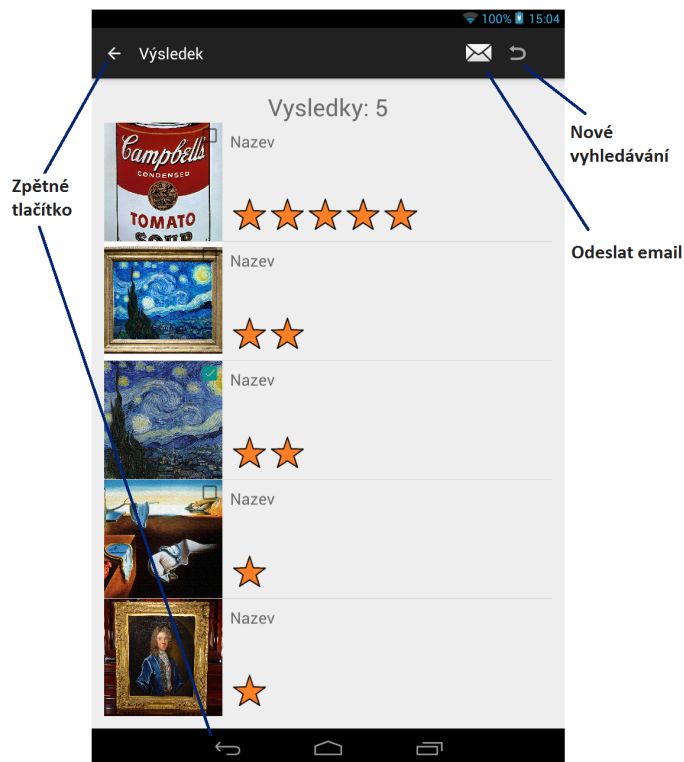
2. NÁVRH



Obrázek 2.27: Odesílání požadavku, pokud zařízení je připojeno k internetu

Při připojení během předešlého kroku (obrazovka s nápisem „Spojení se serverem nenavázáno“) nebo při samotném vstupu na tuto obrazovku, pokud je připojení již navázáno, se po automatickém odeslání požadavku na obrazovce objeví procentuální průběh procesu vyhodnocování na serveru a průběžné výsledky (obr. 2.27). Pokud proces porovnávání skončí (dosáhne 100 %), automaticky se zobrazí konečné výsledky (přechod #24).

- Zpětné tlačítko (přechod #23) – Po stisknutí se uživatel vrátí na obrazovku, kde se mu zobrazí metadata, která v předešlém kroku vyplnil a připojil k požadavku.
- Zastavit (přechod #23) – Po stisknutí tohoto tlačítka se ukončí proces porovnávání a aplikace se vrátí do stavu, kdy uživatel přidává a vyplňuje přidružené informace. Tato metadata jsou vyplněna tak, jak byla před odesláním na server.
- Pozastavit – Po stisknutí tohoto tlačítka se pozastaví porovnávání a aplikace čeká na další rozhodnutí uživatele („Pokračovat“ nebo „Zpětné tlačítko“). Průběžné výsledky budou stále zobrazeny.



Obrázek 2.28: Zobrazení výsledků po prohledání databáze

Poslední krok, kdy se uživateli zobrazí výsledky s náhledem ceny, vybranými informacemi a podobností vyjádřenou hvězdičkami (obr. 2.28). Uživatel si může kliknout na objekt, který se mu zobrazí na nové stránce i s příloženými informacemi o něm (přechod #26). Na obrazovce bude také uvedeno, kolik výsledků bylo nalezeno. Vybrané výsledky budou moci být odeslány na email, který uživatel vyplní. Odeslány budou URL adresy obrázků, kde je příjemce zprávy bude moci nalézt (přechod #28).

- Zpětné tlačítko (přechod #25) – Stisknutím se uživatel dostane zpět na stránku s metadaty vyplněnými tak, jak byly před pokračováním a odesláním požadavku na server.
- Nové vyhledávání (přechod #30) – Uživatel se dostane na první krok, kde bude mít na výběr mezi otevřením galerie, nebo vyfocení nového obrázku.
- Odeslat email – Kliknutím na toto tlačítko se uživateli zobrazí nová stránka, kde vyplní emailovou adresu (popřípadě předmět a obsah), kam bude chtít odeslat URL adresy fotografií, které jsou označené.

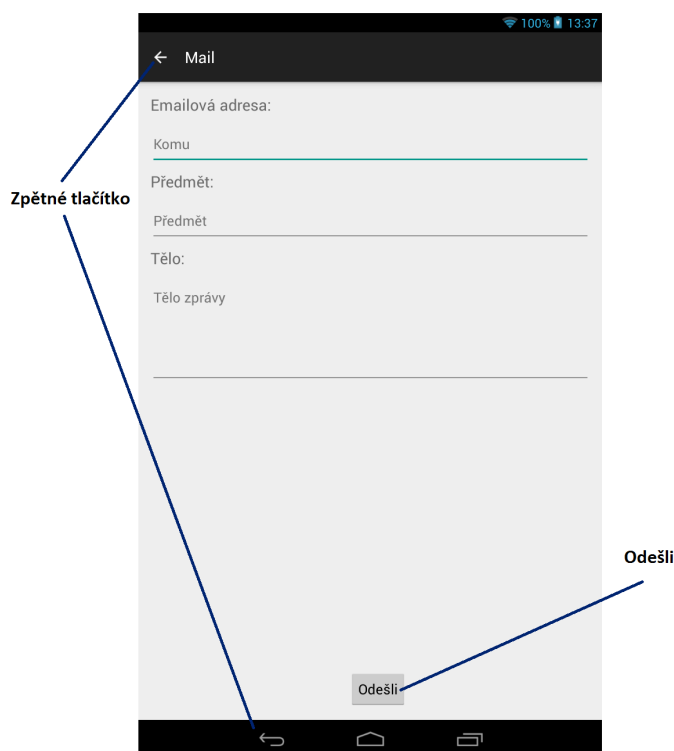
2. NÁVRH



Obrázek 2.29: Detail jednoho z výsledků

Po kliknutí na řádek s libovolným výsledkem se zobrazí jeho detail, ve kterém se uživateli zobrazí fotografie a další atributy, které jsou k výslednému obrázku přiloženy (obr. 2.29). Pokud se všechna data nevejdou na obrazovku, bude možné posunout se podle potřeby níže nebo výše, aby se uživatel mohl dostat ke všem informacím přiloženým k obrázku.

- Zpětné tlačítko (přechod #27) – Po stisknutí se uživatel vrátí na stránku s výslednými fotografiemi a jejich ohodnocením.



Obrázek 2.30: Vyplnění elektronické zprávy

Uživatel bude mít také možnost odeslat své výsledky ostatním osobám pomocí elektronické pošty. Stačí, aby zadal adresu (ta je povinná), komu chce výsledky odeslat. Dále může vyplnit předmět zprávy a její dodatečný obsah, který se připojí k automaticky přidanému obsahu v podobě URL adres, na kterých se obrázky nachází (obr. 2.30).

- Zpětné tlačítko (přechod #29) – Po stisknutí se uživatel vrátí na obrazovku s výsledky vyhledávání. Pokud vyplní jakékoli políčko, vrátí se zpět na obrazovku s výsledky, a když bude chtít opět zprávu odeslat, políčka budou prázdná.
- Odeslat (přechod #29) – Stisknutím se výsledky odešlou i s případným předmětem a obsahem pomocí emailového klienta, kterého mobilní zařízení nabízí.

Implementace

3.1 Využité technologie

3.1.1 Zabudovaná kamera v mobilním zařízení

Jedním z funkčních požadavků je, že aplikace bude schopna pořizovat fotografie. V současné době je již standardem, že mobilní telefony či tablety mají zabudovaný fotoaparát.

Android sám o sobě nabízí komponentu na focení[21], avšak pro zdejší případ není vhodná. Po vyfocení se totiž zobrazí pořizená fotografie a uživatel se rozhoduje, zda ve focení pokračovat, nebo fotografii přímo smazat. Tento přístup se neshoduje s funkcionalitou aplikace, tudíž bylo nutné vytvořit vlastní implementaci fotografování.

Nejdříve bylo potřeba povolit přístup k fotoaparátu v souboru `AndroidManifest.xml`[22]. Poté se vytvořila instance kamery, pomocí které se získávaly informace z fotoaparátu mobilního zařízení. Nakonec se skrze rozhraní `SurfaceView` přenášel uživateli na displej obraz zachycený telefonem či tabletem.

3.1.2 Externí úložiště

Pokud má aplikace používat již pořízené obrázky z galerie nebo pokud má pořizovat obrázky, které mají existovat i po ukončení aplikace, musí být uloženy na předem vybraném úložišti. V tomto případě se bude jednat o externí úložiště. Před využíváním této možnosti se musí nastavit v konfiguračním souboru práva, která umožní přístup k externímu úložišti.

```
<uses-permission  
    android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

Nevýhodou tohoto přístupu oproti internímu úložišti je fakt, že nemusí být vždy přístupný. Například pokud uživatel připojí zařízení pomocí USB (Uni-

versal serial bus) kabelu k počítači nebo odebere medium s úložištěm (SD kartu – Secure digital) z mobilního zařízení.

Jeho nespornou výhodou je možnost sdílení dat (obrázků), které interní úložiště nenabízí[23].

3.2 Focení objektu

Jednou ze základních funkcionalit je focení objektu pomocí vestavěného fotoaparátu. Aby focení bylo uživateli příjemné, nesmí ho zatěžovat zdlouhavými procedurami. Proto hned odpadl případ, že by po každém vyfocení uživatel musel potvrdit, jestli chce fotografii uložit nebo ji rovnou smazat.

Aplikace tedy dovoluje stále pořizovat fotografie bez přerušení, dokud uživatel neuzná za vhodné pokračovat na další krok. Po vyfocení několika snímků se bude moci přesunout do galerie, kde si je bude moci prohlédnout. Pokud nebude dostatečně spokojen s pořízenými fotografiemi, bude se moci vrátit k focení a pokračovat v získávání lepších obrázků.

3.3 Načítání fotografií z galerie

Je zřejmé, že aplikace bude muset mít přístup k úložišti fotografií, které uživatel pořídil, aby je následovně mohla zobrazit. Nejlepší způsob práce s nimi by byl, kdyby se nemusely fotografie při každém zobrazení načítat. Tento problém je vyřešen tak, že se při prvním spuštění galerie vytvoří instance třídy, která existuje i mimo aktivitu, ve které je vytvořena. Takto je předejito opakovanému načítání obrázků například při přechodu mezi aktivitami.

Dále bylo potřeba vyřešit, jak fotografie nejefektivněji (paměťově i časově) načítat a pracovat s nimi. V potaz přišly dva přístupy.

Jeden přístup spočívá v načtení všech fotografií do paměti, kterou programu přiřadí operační systém. Výhodou tohoto přístupu je rychlost při otevírání náhledu fotografie, či práce s jednotlivými fotografiemi. Vše je předem načteno, tudíž není potřeba obrázky znovu nahrávat do paměti. Avšak velkou nevýhodou je omezená velikost paměti, kterou je možné tímto způsobem přístupu vyčerpat.

Proto je tento problém řešen způsobem lazy loading. Aplikace načte pouze ty fotografie, které jsou uživateli zobrazeny (jsou vidět na obrazovce). Při pohybu mezi jednotlivými objekty se postupně obrázky načítají a ukládají do paměti. Tudíž pokud se uživatel bude chtít dostat zpět do horní části galerie, obrázky se již nebudou načítat pomalu z úložiště, nýbrž z paměti telefonu, která je mnohonásobně rychlejší.

3.4 Vyplnění metadat

Aplikace by měla počítat s tím, že uživatel nebude umět přesně a sám určit atributy vztahující se k danému předmětu, tudíž by mu měla napomáhat při vytváření požadavku, konkrétně vyplňování metadat. Otázka zní, jak nejvíce usnadnit práci běžnému uživateli?

Ruční vypisování všech atributů, které ho napadnou, nebo na které si v danou chvíli vzpomene, není, vzhledem k předešlé otázce, v tomto případě nejvhodnější volbou. Ne vždy si uživatel, který neprošel kurzy architektury a historie, vzpomene a sám popíše, jakými prvky je který umělecký směr charakteristický. Třeba by si vzpomněl, pokud by mu někdo poradil nebo by viděl, které možnosti existují a zároveň připadají v úvahu. Proto bude mít aplikace několik možností výběru atributů podle toho, jaký typ předmětu uživatel vybere, a také hodnota bude vyplňována různými způsoby, které budou záviset na vybraném atributu.

V ideálním průběhu by se uživateli objevila obrazovka, kde by si nejdříve vybral typ objektu, podle kterého se mu nabídnou položky vztahující se k danému předmětu. U prstenu pro nás bude důležitější spíše materiál, ze kterého je vyroben, nebo drahé kameny, které v něm jsou zasazeny, než například rám, který je relevantní možností pro obraz. Tato volba uživateli zúží výběr atributu cennosti.

Dále se ukáže obrazovka, kde uživatel bude moci vyplnit jednu či více informací vztahujících se k objektu. Možnosti budou uvedeny v podobě seznamu. Pokud uživatel v seznamu nenajde atribut, který by rád vyplnil, pak bude mít ještě možnost ručního vyplnění. Po vybrání této možnosti se aplikace zeptá, zda se jedná o textovou hodnotu atributu, nebo číselnou hodnotu. V prvním případě se uživateli zobrazí dvě políčka, kde vyplní atribut a jeho hodnotu. V druhém případě se zobrazí tři políčka, která nesou informaci o atributu, hodnotě, která musí mít číselnou formu, a popřípadě i jednotce, ve které je hodnota zadána.

Po výběru atributu se zvolí jedna ze tří možností, která určí způsob vyplnění hodnoty.

Prvními dvěma možnostmi jsou výše zmiňované textové a číselné hodnoty, které mohou obsahovat i jednotku, kterou ze seznamu vybere uživatel.

Třetí možností je seznam, ze kterého uživatel zvolí jednu možnost. K atributu „Umělecký styl“ se těžko bude vymýšlet text, proto je lepší dát uživateli na výběr z množiny stylů a tím mu usnadnit práci nebo napovědět, pokud si zrovna nebude schopen vzpomenout.

3.5 Vytvoření požadavku

Pro zjištění výsledku musí být nejdříve vytvořen požadavek, který charakterizuje dílo, jež bude uživatel chtít porovnat s databází.

Prvním krokem k vytvoření požadavku je získání obrázků díla do mobilního zařízení a vybrání vhodných fotografií k následnému vyhodnocení. Fotografií může pořídit policista kolik bude chtít, avšak musí počítat s tím, že větší počet fotografií odeslaných na server se bude odesílat i zpracovávat déle.

Z množiny vhodných kandidátů pro zjištění podobnosti lze dále u jednotlivých obrázků vybrat oblast, která zachycuje pouze důležitou část. Ohraničení není nutné, tudíž je jen na uživateli, zda vybere oblast na fotografii, kde je předmět zachycen.

Poslední fází tvorby požadavku je přidání metadat. Ta mohou být uživatelem vyplněna a přiložena k fotografiím. Informace zadané v tomto kroku nejsou nutné, požadavek může být odeslán i bez dalších informací o cennosti, ale jejich přiložením se urychlí hledání na straně serveru (k porovnání budou vybrány jen takové obrázky, které se budou v těchto přidružených datech shodovat).

3.6 Odesílání a přijímání dat

Velmi důležitou součástí tohoto softwaru je komunikace se serverem. Tomu musí aplikace odeslat data ve formátu JSON. Tam se data zpracují a výsledek se odešle zpět uživateli.

Součástí funkčních požadavků je výběr více snímků k odeslání. Služba však neumí přijímat v jednom požadavku více než jeden zakódovaný obrázek. Tudíž je potřeba, aby tento problém vyřešila aplikace.

Možným řešením je odeslání obrázků, kde každý bude obsažen v jednom požadavku. Služba tedy obdrží tolik požadavků, kolik uživatel vybere obrázků. Každý požadavek bude obsahovat stejná metadata. Aplikace získá od serveru URI jednotlivých úloh, na které se bude po odeslání všech požadavků postupně dotazovat.

Po dokončení každé úlohy program obdrží pole výsledků, která sjednotí. Pole, jež vznikne po sjednocení všech polí obdržných od jednotlivých úloh, nebude obsahovat žádné položky, které by měly stejnou URL adresu obrázku.

Testování

Důležitou součástí, jež bývá často podceňována nebo opomíjena, je testování vyvíjeného softwaru. Důkladným otestováním se z produktu odstraní velké množství chyb, které by mohly odradit budoucí i současné uživatele či zákazníky. V případě jednoduchých programů je proces testování kratší a odstranění chyb snadnější. Je velice pravděpodobné, že se z nich odstraní většina chyb. Oproti tomu ve složitých a obsáhlých programech je vytvoření všech možných scénářů pro otestování hodně náročné.

4.1 Testování autorem

Pro jednodušší zjištění a zvýšení kvality jsem průběžně s vývojem softwaru testoval jednotlivé aktivity a funkcionality aplikace. Za tímto účelem bylo využíváno zařízení Prestigio Multipad 7.0 3G s operačním systémem Android 4.2.2. Pro zjištění výstupů metod a vypisování chybových hlášek jsem používal třídu Log.

4.2 Testování použitelnosti

Po důkladném otestování autorem je třeba, aby otestovali aktivity i uživatelé, kteří zatím s aplikací neměli žádné zkušenosti. Pro tento případ bylo zvoleno testování použitelnosti, které prověří funkčnost softwaru.

Důležitým kritériem tohoto typu testu je výběr cílové skupiny, jež bude software testovat. Každá skupina by měla být sestavena z lidí, kteří by měli danou aplikaci v budoucnu využívat[24].

Pro tento případ je vybrána skupina lidí mající alespoň středoškolské vzdělání ukončené maturitní zkouškou. Splnění této podmínky je jedním z předpokladů pro přijetí k Policii ČR.

Všechny testy byly provedeny na zařízení Prestigio Multipad 7.0 3G s operačním systémem Android 4.2.2.

4.2.1 Popis testu

Základní myšlenkou tohoto testu je zjistit, zda je ovládání intuitivní a vybraný uživatel (tester) dosáhne požadovaného výsledku a získá požadované informace. Každý testovaný dostane navržený scénář, pomocí kterého by měl dosáhnout předem určených výsledků.

4.2.2 Scénář

Pořídte obrázkový podklad předmětu pro vyhledávání v databázi, ohraničte předmět na obrázku nebo obrázcích a zhodnoťte výsledky.

Řešení:

1. Po spuštění aplikace se stiskne tlačítko „Kamera“. Poté se pomocí tlačítka „Vyfotit“ vyfotí několik snímků.
2. Pro pokračování je potřeba vybrat alespoň jednu položku v galerii.
3. Na další obrazovce se zobrazí galerie vybraných fotografií, kde se po kliknutí na položku zobrazí fotografie a nástroje (v horní části obrazovky), pomocí kterých se ohraničí oblast určená k porovnání. Dále se uloží ohraničení pomocí tlačítka pro pokračování.
4. Pokračováním z galerie vybraných obrázků se zobrazí okno, kde je třeba vybrat kategorii, do které předmět spadá. Po vybrání kategorie se na obrazovce zobrazí pole pro vyplnění doplňujících informací v podobě atributu a jeho hodnoty. Může být přidáno více informací pomocí tlačítka pod seznamem metadat, nebo může být informace odebrána tlačítkem umístěným vlevo od jména atributu.
5. Po dokončení porovnávání na straně serveru se odešlou výsledky na klientskou stranu, kde se zobrazí v podobě seřazeného seznamu podle míry shody.

4.3 Výsledky testování

4.3.1 Testeři

4.3.1.1 Tester 1

- Pohlaví: muž
 - Věk: 22
 - Vzdělání: středoškolské s maturitní zkouškou
1. Tento krok tester zvládl bez problémů, avšak pořídil pouze jednu fotografii.

2. Tester chtěl vybrat obrázek určený k odeslání kliknutím na položku, čímž se mu otevřel jeho detail. Po vrácení se do galerie si všiml zaškrťovacího tlačítka a označil obrázek, který byl určen k porovnání.
3. Uživatel nejdříve klikl na nástroj pro přidání ohraničení k již existujícímu. Ohraničení dále proběhlo bez problémů.
4. Tento krok tester zvládl bez problémů cestou popsanou v řešení.
5. Tento krok tester zvládl bez problémů cestou popsanou v řešení.

4.3.1.2 Tester 2

- Pohlaví: muž
 - Věk: 21
 - Vzdělání: středoškolské s maturitní zkouškou
1. Tento krok tester zvládl bez problémů cestou popsanou v řešení.
 2. Tester chtěl vybrat obrázek určený k odeslání kliknutím na položku, čímž se mu otevřel jeho detail. Po vrácení se do galerie si všiml zaškrťovacího tlačítka a označil obrázky, které byly určeny k porovnání.
 3. Uživatel nejdříve klikl na nástroj pro přidání ohraničení k již existujícímu. Ohraničení dále proběhlo bez problémů.
 4. Tento krok tester zvládl bez problémů cestou popsanou v řešení.
 5. Tento krok tester zvládl bez problémů cestou popsanou v řešení.

4.3.1.3 Tester 3

- Pohlaví: žena
 - Věk: 22
 - Vzdělání: středoškolské s maturitní zkouškou
1. Tento krok tester zvládl bez problémů, avšak pořídil pouze jednu fotografii.
 2. Tester chtěl vybrat obrázek určený k odeslání kliknutím na položku, čímž se mu otevřel jeho detail. Po vrácení se do galerie si všiml zaškrťovacího tlačítka a označil obrázek, který byl určen k porovnání.
 3. Tento krok tester zvládl bez problémů cestou popsanou v řešení i se smazáním nevyvedeného ohraničení a opětovného označení oblasti.

4. TESTOVÁNÍ

4. Tento krok tester zvládl bez problémů cestou popsanou v řešení.
5. Tento krok tester zvládl bez problémů cestou popsanou v řešení.

4.3.1.4 Tester 4

- Pohlaví: žena
 - Věk: 23
 - Vzdělání: vysokoškolské
1. Tento krok tester zvládl bez problémů, avšak pořídil pouze jednu fotografii.
 2. Tester chtěl vybrat obrázek určený k odeslání kliknutím na položku, čímž se mu otevřel jeho detail. Po vrácení se do galerie se vrátil až na úvodní obrazovku, kde vybral možnost „Galerie“. Po chvíli si všiml zaškrťovacího tlačítka a označil obrázek, který byl určen k porovnání.
 3. Při ohrazení se čára překřížila, tudíž byla celá doposud ohrazená oblast smazána. Tester následně úspěšně ohrazení předmět bez křížení.
 4. Tento krok tester zvládl bez problémů cestou popsanou v řešení.
 5. Tento krok tester zvládl bez problémů cestou popsanou v řešení.

4.3.1.5 Tester 5

- Pohlaví: muž
 - Věk: 22
 - Vzdělání: středoškolské s maturitní zkouškou
1. Tento krok tester zvládl bez problémů, avšak pořídil pouze jednu fotografii.
 2. Tester chtěl vybrat obrázek určený k odeslání kliknutím na položku, čímž se mu otevřel jeho detail. Po vrácení se do galerie si po chvíli všiml zaškrťovacího tlačítka a označil obrázek, který byl určen k porovnání.
 3. Uživatel nejdříve klikl na nástroj pro přidání ohrazení k již existujícímu. Ohrazení dále proběhlo bez problémů.
 4. Tento krok tester zvládl bez problémů cestou popsanou v řešení.
 5. Tento krok tester zvládl bez problémů cestou popsanou v řešení.

4.3.1.6 Tester 6

- Pohlaví: muž
 - Věk: 22
 - Vzdělání: středoškolské s maturitní zkouškou
1. Tento krok tester zvládl bez problémů, avšak pořídil pouze jednu fotografii.
 2. Tester chtěl vybrat obrázek určený k odeslání kliknutím na položku, čímž se mu otevřel jeho detail. Po vrácení se do galerie se vrátil až na úvodní obrazovku, kde vybral možnost „Galerie“. Po chvíli si všiml zaškrťovacího tlačítka a označil obrázek, který byl určen k porovnání.
 3. Při ohraničení se čára překřížila, tudíž byla celá doposud ohraničená oblast smazána. Tester následně úspěšně ohraničil předmět bez křížení.
 4. Tento krok tester zvládl bez problémů cestou popsanou v řešení.
 5. Tento krok tester zvládl bez problémů cestou popsanou v řešení.

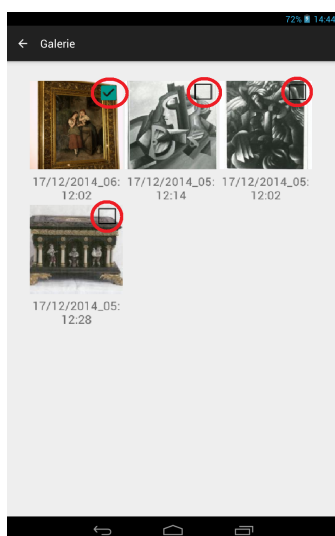
4.3.1.7 Tester 7

- Pohlaví: muž
 - Věk: 23
 - Vzdělání: vysokoškolské
1. Tento krok tester zvládl bez problémů, avšak pořídil pouze jednu fotografii.
 2. Tester chtěl vybrat obrázek určený k odeslání kliknutím na položku, čímž se mu otevřel jeho detail. Po vrácení se do galerie si všiml zaškrťovacího tlačítka a označil obrázek, který byl určen k porovnání.
 3. Tento krok tester zvládl bez problémů cestou popsanou v řešení.
 4. Tento krok tester zvládl bez problémů cestou popsanou v řešení.
 5. Tento krok tester zvládl bez problémů cestou popsanou v řešení.

4.4 Vyhodnocení

Uživatelé, kteří testovali aplikaci, se nejčastěji lišili od řešení v kroku, kde se vybírají obrázky k odeslání. Nikdo z nich si nevšiml zaškrťovacího tlačítka, které znázorňuje vybrané obrázky určené k porovnání. Místo toho v domnění, že mohou vybrat jen jeden obrázek, klikali přímo na položku, čímž otevřeli pouze její detail. (obr. 4.1).

4. TESTOVÁNÍ



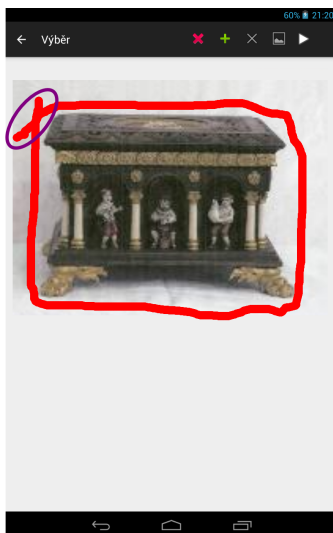
Obrázek 4.1: Málo viditelná zaškrťovací tlačítka

Z výsledků testů je patrné, že uživatelé před výběrem oblasti předpokládají nutnost vybrat nástroj na ohraničení. Funkce počátečního výběru se od přidávání oblasti liší tím, že je potřeba, aby ohraničení již existovalo. Stisknutím zmíněného tlačítka při absenci ohraničení tedy aplikace nástroj nevybere a zůstane u počátečního výběru (obr. 4.2).



Obrázek 4.2: Nejasný účel nástroje

Dalším problémovým krokem bylo ohrazení obrázku, kde 2 testeři překřídili čáru samu se sebou, čímž se ohrazení zrušilo (obr. 4.3).



Obrázek 4.3: Překřížení ohrazení

Další kroky byly v souladu s ideálním scénářem pro vytvoření a odeslání požadavků a následného vyhodnocení výsledků.

Závěr

Cílem bakalářské práce bylo vyvinout mobilní aplikaci na vyhledávání ukradených předmětů. Program pracuje jako klientská část odesílající požadavky obsahující zakódovanou fotografii a přidružené informace týkající se podezřelého předmětu ve formátu JSON na server, který požadavek zpracuje a odešle výsledky, které se uživateli na zařízení zobrazí. Pomocí těchto výsledků vypracovaných SimIG aplikací může uživatel porovnat podobu nalezených výsledků a hmotného předmětu.

Pro tvorbu této aplikace bylo nutné si uvědomit, pro jaký operační systém bude aplikace nejvhodnější. Musíme ale také navrhnout průběh vytváření požadavků, které se následně odešlou. Důležitou roli hrál formát dat, ve kterém služba SimIG přijímá a odesílá data. Nicméně tato služba je ve fázi vývoje. Proto je místo ní vytvořen testovací server, jež odpovídá na každý požadavek odeslaný pomocí metody POST stejným JSON řetězcem. Tímto způsobem je ověřeno, že aplikace přijímá odpovědi a následně je zpracuje.

Dále bylo důležité otestovat aplikaci. Program byl průběžně testován autorem a následně i samotnými uživateli, kteří neměli v době testování o aplikaci žádné informace.

Zhodnocení testu použitelnosti

Po vyhodnocení výsledků testování bylo zjištěno, že uživatel, který s aplikací nemá žádné zkušenosti, nepředpokládá možnost výběru více obrázků. K tomu přispívá i fakt, že většina pořídila pouze jednu fotografii. Aplikace tedy nedostatečně vybízí uživatele k výběru více obrázků k odeslání a následnému porovnání. Tento problém by mohlo vyřešit větší a výraznější zaškrtačací tlačítko při výběru v galerii.

Dále uživatelé při výběru oblasti klikali na tlačítko znázorňující přidání oblasti ještě před její existencí. Tento problém je možné odstranit zobrazením nástroje pro přidání a odebrání ohraničení až po jejím vytvoření.

Další problémovou částí bylo ohraničení relevantní oblasti obrázku. Překřížení se dopustili 2 ze 7 testerů. Jako řešení vidím úpravu ve způsobu ohraničování. Pokud by se čára překřížila pouze v jednom bodě, „nadbytečné“ křivky, které nejsou součástí uzavřeného výběru, by byly smazány. Pokud by se však překřížila ve více bodech, výběr by byl opět smazán.

Další vývoj aplikace

Z časových důvodů nebyla realizována všechna navržená funkcionalita (pozastavení úlohy porovnávající obrázků s databází, zobrazování průběžných výsledků při vyhledávání, přibližování obrázků pomocí gest). S tím je spjata také grafická stránka programu. Ta by mohla být postupem času upravena. Změna vzhledu by mohla přinést větší přehlednost a zvýšit míru využívání této aplikace. Proto si myslím, že je zde prostor pro pokračování na tomto projektu.

Literatura

- [1] Google, Inc.: Google Images[online]. 2015, [Cit. 2015-7-5]. Dostupné z: <https://images.google.com/>
- [2] Budíková, P.: Image search: kde slova nestačí[online]. 2010, [Cit. 2015-7-5]. Dostupné z: <http://ics.muni.cz/bulletin/articles/652.html>
- [3] Bartošek, M.: Nástroje Google. 5. Google Image Search[online]. 2009, [Cit. 2015-7-5]. Dostupné z: <http://ics.muni.cz/bulletin/articles/623.html>
- [4] DISA: MUFIN[online]. 2013, [Cit. 2015-7-5]. Dostupné z: <http://mufin.fi.muni.cz/imgsearch/similar>
- [5] Kohoutková, P.: MUFIN: buďte v obraze![online]. 2009, [Cit. 2015-7-5]. Dostupné z: <http://ikaros.cz/mufin-budte-v-obraze>
- [6] Ideé Inc.: TinEye[online]. 2015, [Cit. 2015-7-5]. Dostupné z: <http://www.tineye.com/>
- [7] Ideé Inc.: MatchEngine[online]. 2015, [Cit. 2015-7-5]. Dostupné z: <http://services.tineye.com/MatchEngine>
- [8] Google, Inc.: Google Goggles[online]. 2015, [Cit. 2015-7-5]. Dostupné z: <https://play.google.com/store/apps/details?id=com.google.android.apps.unveil&hl=cs>
- [9] Avaranic, Inc.: Unravl[online]. 2014, [Cit. 2015-7-5]. Dostupné z: <http://www.unravl-app.com/>
- [10] Ajinky: Reverse Image Search Engines, Apps And Its Uses[online]. 2015, [Cit. 2015-7-5]. Dostupné z: <http://beebom.com/2015/02/reverse-image-search-engines-apps-and-its-uses>

- [11] Google, Inc.: Unravl - Search by Image[online]. 2015, [Cit. 2015-7-5]. Dostupné z: <https://play.google.com/store/apps/details?id=com.avaaranic.wtis&hl=en>
- [12] Allamaraju, S.: *RESTful Web Services Cookbook*. O'Reilly Media, Inc, 2010, ISBN 978-0-596-80168-7.
- [13] About Poonam Dhanvani: Difference between JSON and XML | JSON vs. XML[online]. 2013, [Cit. 2015-7-5]. Dostupné z: <http://freefeast.info/tutorials-for-beginners/dotnet-tutorials/difference-between-json-and-xml-json-vs-xml/>
- [14] Pavlíček, M.: Android drží spolu s iOS více než 96operačními systémy[online]. 2015, [Cit. 2015-7-5]. Dostupné z: <http://mobilenet.cz/clanky/android-spolu-s-ios-drzi-vice-nez-96-podil-mezi-operacnimi-systemy-18872>
- [15] Meralai, M.: Poradna: Chci chytrý telefon, jaký zvolit operační systém? Android, iOS nebo Windows Phone?[online]. 2013, [Cit. 2015-7-5]. Dostupné z: <http://mobile-effect.cz/clanky/poradna-chci-chytry-telefon-jaky-zvolit-operacni-system-android-ios-nebo-windows-phone-148678>
- [16] Stevenson, A.: Top 10 Android benefits over Apple iPhone[online]. 2014, [Cit. 2015-7-5]. Dostupné z: <http://www.v3.co.uk/v3-uk/news/2336474/top-10-android-benefits-over-apple-iphone>
- [17] Google, Inc.: Remaining Backward Compatible[online]. 2015, [Cit. 2015-7-5]. Dostupné z: <http://developer.android.com/training/search/backward-compat.html>
- [18] Meier, R.: *Professional Android 4 Application Development*. John Wiley & Sons, Inc, 2012, ISBN 978-1-118-10227-5.
- [19] Google, Inc.: Dashboards[online]. 2015, [Cit. 2015-7-5]. Dostupné z: <https://developer.android.com/about/dashboards/index.html#Platform>
- [20] Seralo: Android versions comparison[online]. 2015, [Cit. 2015-7-5]. Dostupné z: <http://socialcompare.com/en/comparison/android-versions-comparison>
- [21] Google, Inc.: Taking Photos Simply[online]. 2015, [Cit. 2015-7-5]. Dostupné z: <https://developer.android.com/training/camera/photobasics.html>
- [22] Google, Inc.: Camera[online]. 2015, [Cit. 2015-7-5]. Dostupné z: <http://developer.android.com/reference/android/hardware/Camera.html>

- [23] Google, Inc.: Storage Options[online]. 2015, [Cit. 2015-7-5]. Dostupné z: <http://developer.android.com/guide/topics/data/data-storage.html#filesExternal>

- [24] Nielsen Norman Group: Turn User Goals into Task Scenarios for Usability Testing[online]. 2014, [Cit. 2015-7-5]. Dostupné z: <http://www.nngroup.com/articles/task-scenarios-usability-testing/>

Seznam použitých zkratk

- API** Application programming interface
- CR** Czech Republic
- ČR** Česká republika
- DCIM** Digital camera images
- email** Electronic mail
- HTTP** Hypertext transfer protocol
- HTTPS** Hypertext transfer protocol secure
- JIT** Just in time
- JSON** JavaScript object notation
- JVM** Java virtual machine
- MIDP** Mobile information device profile
- MPEG** Motion picture experts group
- MUFIN** Multi-feature indexing network
- NFC** Near field communication
- QR** Quick response
- REST** Representational state transfer
- SD** Secure digital
- SIP** Session initiation protocol
- UPC** Universal product code

A. SEZNAM POUŽITÝCH ZKRATEK

URI Uniform resource identifier

URL Uniform resource locator

USB Universal serial bus

WiFi Wireless fidelity

XML Extensible markup language

Obsah přiloženého CD

	readme.txt.....	stručný popis obsahu CD
	apk.....	adresář s instalačním souborem
	src	
	impl.....	zdrojové kódy implementace
	thesis.....	zdrojová forma práce ve formátu $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$
	text	
	thesis.pdf.....	text práce ve formátu PDF