



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

| | |
|--------------------------|--|
| Název: | Aplikace pro monitorování a správu aukcí na portálu eBay |
| Student: | Tomáš Polačok |
| Vedoucí: | doc. RNDr. Ing. Marcel Jiřina, Ph.D. |
| Studijní program: | Informatika |
| Studijní obor: | Webové a softwarové inženýrství |
| Katedra: | Katedra softwarového inženýrství |
| Platnost zadání: | Do konce zimního semestru 2020/21 |

Pokyny pro vypracování

Aukční portál eBay má svůj vnitřní nástroj pro hlídání aukcí (watchdog/watch lists), nicméně tento nástroj nefunguje spolehlivě a má řadu omezení. Cílem práce je vytvořit aplikaci, která by běžela na pozadí (na serveru/v cloudu) a hlídala běžící i právě nově zveřejněné aukce podle zadaných kritérií a následně by generovala/aktualizovala přehledy a současně upozorňovala uživatele (email, SMS). Rámcovými požadavky jsou zejména:

- možnost zadávat podrobné logické podmínky (kritéria) pro vyhledávání (co má vyhledávaný objekt splňovat - popis, cena apod.)
- možnost pracovat s více lokalizacemi portálu eBay (eBay.com, eBay.de, eBay.co.uk, eBay.fr, ...)
- třídění a kategorizace výsledků
- zasílání upozornění na mail/SMS
- generování strukturovaných přehledů (náhledů) a statistik
- uchovávání historických dat (včetně obrazových) pro budoucí strojové rozpoznávání, např. pomocí deep learningu (není součástí této práce)
- uživatelsky jednoduché a přívětivé rozhraní

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 18. února 2019



**FAKULTA
INFORMAČNÍCH
TECHNOLÓGIÍ
ČVUT V PRAZE**

Bakalárska práca

Aplikácia pre monitorovanie a správu aukcií na portáli eBay

Tomáš Polaček

Katedra softvérového inžinierstva

Vedúci práce: doc. RNDr. Ing. Marcel Jiřina, Ph.D.

2. mája 2019

Pod'akovanie

Týmto by som chcel poďakovať zadavateľovi a vedúcemu tejto bakalárskej práce doc. RNDr. Ing. Marcelovi Jiřinovi, Ph.D. za jeho rady, ochotu a pomoc počas celej realizácie práce. Ďalej by som chcel poďakovať rodine a priateľom za ich podporu.

Prehlásenie

Prehlasujem, že som predloženú prácu vypracoval samostatne a že som uviedol všetky informačné zdroje v súlade s Metodickým pokynom o etickej príprave vysokoškolských záverečných prác.

Beriem na vedomie, že sa na moju prácu vzťahujú práva a povinnosti vyplývajúce zo zákona č. 121/2000 Sb., autorského zákona, v znení neskorších predpisov. V súlade s ustanovením § 46 odst. 6 tohoto zákona týmto udeľujem bezvýhradné oprávnenie (licenciu) k užívaniu tejto mojej práce, a to vrátane všetkých počítačových programov ktoré sú jej súčasťou alebo prílohou a tiež všetkej ich dokumentácie (ďalej len „Dielo“), a to všetkým osobám, ktoré si prajú Dielo užívať.

Tieto osoby sú oprávnené Dielo používať akýmkoľvek spôsobom, ktorý nezníži hodnotu Diela, a za akýmkoľvek účelom (vrátane komerčného využitia). Toto oprávnenie je časovo, územne a množstevne neobmedzené. Každá osoba, ktorá využije vyššie uvedenú licenciu, sa však zaväzuje priradiť každému dielu, ktoré vznikne (čo i len čiastočne) na základe Diela, úpravou Diela, spojením Diela s iným dielom, zaradením Diela do diela súborného či zpracovaním Diela (vrátane prekladu), licenciu aspoň vo vyššie uvedenom rozsahu a zároveň sa zaväzuje sprístupniť zdrojový kód takého diela aspoň zrovnateľným spôsobom a v zrovnateľnom rozsahu ako je zprístupnený zdrojový kód Diela.

V Prahe 2. mája 2019

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2019 Tomáš Polačok. Všetky práva vyhrazené.

Táto práca vznikla ako školské dielo na FIT ČVUT v Prahe. Práca je chránená medzinárodnými predpismi a zmluvami o autorskom práve a právach súvisiacich s autorským právom. Na jej využitie, s výnimkou bezplatných zákonných licencií, je nutný súhlas autora.

Odkaz na túto prácu

Polačok, Tomáš. *Aplikácia pre monitorovanie a správu aukcií na portáli eBay*. Bakalárska práca. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2019.

Abstrakt

Táto bakalárska práca popisuje tvorbu softvérovej aplikácie slúžiacej na prácu s aukčným portálom eBay. Zahŕňa analýzu súčasného stavu vyhľadávania a monitorovania položiek na danom portáli, a taktiež analýzu požiadaviek kladených na výslednú aplikáciu. Následne obsahuje návrh vlastného riešenia vytvoreného na základe tejto analýzy a s ním súvisiaci výber vhodných technológií. Nakoniec popisuje implementáciu a vyhodnotenie výsledkov vytvorenej aplikácie.

Kľúčové slová eBay API, Spring aplikácia, vyhľadávanie aukcií, notifikácie, archivácia obrázkov

Abstract

This bachelor thesis describes creation of software application for working with eBay auction portal. It includes analysis of the current state of searching and monitoring items on a given portal, as well as analysis of the requirements for the final application. Consequently, it contains a design of my own solution created on the basis of this analysis and the related selection of suitable technologies. Lastly it describes implementation and evaluation of the results of the created application.

Keywords eBay API, Spring application, auction search, notifications, image archiving

Obsah

| | |
|--|-----------|
| Úvod | 1 |
| 1 Cieľ práce | 3 |
| 2 Analýza | 5 |
| 2.1 Existujúce riešenie | 5 |
| 2.1.1 Vyhľadávanie eBay | 5 |
| 2.1.2 Zhrnutie | 7 |
| 2.2 Prístup k dátam — eBay API | 7 |
| 2.2.1 Získanie prístupu | 7 |
| 2.2.2 Typy | 8 |
| 2.2.3 Obmedzenia | 9 |
| 2.3 Požiadavky | 10 |
| 2.3.1 Funkčné požiadavky | 10 |
| 2.3.2 Nefunkčné požiadavky | 11 |
| 2.4 Prípady použitia | 11 |
| 3 Návrh | 15 |
| 3.1 Forma aplikácie | 15 |
| 3.2 Architektúra | 16 |
| 3.3 Databázový model | 17 |
| 3.4 Archivácia obrázkov | 19 |
| 3.5 Užívateľské rozhranie | 19 |
| 3.5.1 Registrácia | 19 |
| 3.5.2 Prihlasovanie | 19 |
| 3.5.3 Vyhľadávanie | 21 |
| 3.5.4 Vytváranie vyhľadávania | 21 |
| 3.5.5 Výsledky vyhľadávania | 22 |
| 3.5.6 Detail položky | 22 |

| | | |
|----------|--|-----------|
| 3.5.7 | Zákazy predajcov a položiek | 23 |
| 3.5.8 | Archivované obrázky | 23 |
| 3.5.9 | Nápoveda | 24 |
| 3.5.10 | Nastavenie | 24 |
| 4 | Technológie | 29 |
| 4.1 | Backend | 29 |
| 4.1.1 | Java | 29 |
| 4.1.2 | Spring Boot | 30 |
| 4.1.3 | Hibernate | 30 |
| 4.1.4 | PostgreSQL | 30 |
| 4.1.5 | Twilio | 31 |
| 4.1.6 | eBay API + Java | 31 |
| 4.1.7 | Google Drive | 32 |
| 4.2 | Frontend | 33 |
| 4.2.1 | JQuery | 33 |
| 4.2.2 | Bootstrap | 33 |
| 4.2.3 | Thymeleaf | 33 |
| 4.3 | Dalšie nástroje | 33 |
| 4.3.1 | Apache Maven | 33 |
| 4.3.2 | Git | 34 |
| 4.3.3 | Spring Tools Suite | 34 |
| 4.4 | Nasadenie | 35 |
| 5 | Realizácia | 37 |
| 5.1 | Štruktúra aplikácie | 37 |
| 5.1.1 | Adresár java | 37 |
| 5.1.2 | Adresár resources | 38 |
| 5.2 | Dátová vrstva | 39 |
| 5.3 | Prezentačná vrstva | 40 |
| 5.4 | Validácia dát | 41 |
| 5.5 | Vyhľadávanie | 42 |
| 5.5.1 | Vytvorenie vyhľadávania | 42 |
| 5.5.2 | Žiadosť na Finding API | 42 |
| 5.5.3 | Spracovanie odpovede | 43 |
| 5.5.4 | Notifikácie | 43 |
| 5.5.5 | Aktualizácia vyhľadávania | 44 |
| 5.5.6 | Kategórie položiek | 44 |
| 5.6 | Peňažná mena | 45 |
| 5.6.1 | Získavanie prevodných kurzov | 45 |
| 5.6.2 | Konverzia | 45 |
| 5.7 | Archivácia | 45 |
| 5.7.1 | Načítanie detailu položky | 46 |
| 5.7.2 | Uloženie do archívu | 46 |

| | | |
|----------|---|-----------|
| 5.7.3 | Cache | 46 |
| 5.8 | Nasadenie | 47 |
| 5.9 | Licencie | 47 |
| 6 | Výsledky | 49 |
| 6.1 | Testovanie | 49 |
| 6.1.1 | Registrácia a prihlásenie | 50 |
| 6.1.2 | Telefónne číslo | 51 |
| 6.1.3 | Užívateľské preferencie a heslo | 51 |
| 6.1.4 | Vyhľadavanie | 52 |
| 6.1.5 | Zákaz | 52 |
| 6.1.6 | Detaily položky a archív | 53 |
| 6.1.7 | Zhrnutie | 56 |
| 6.2 | Vyhliadky do budúcnosti | 58 |
| 6.2.1 | Úprava užívateľského rozhrania | 58 |
| 6.2.2 | Práca s eBay položkami | 58 |
| 6.2.3 | Limity API | 58 |
| 6.2.4 | Viacjazyčnosť | 58 |
| 6.2.5 | Strojové učenie | 58 |
| | Záver | 59 |
| | Zoznam použitej literatúry | 61 |
| | A Zoznam použitých skratiek | 65 |
| | B Ďalšie screenshoty aplikácie | 67 |
| | C Obsah priloženého CD | 71 |

Zoznam obrázkov

| | | |
|------|--|----|
| 2.1 | eBay advanced search [1] | 6 |
| 2.2 | eBay API limity [2] | 9 |
| 2.3 | Prípady použitia | 14 |
| 3.1 | Webová aplikácia (prevzaté z [3]) | 15 |
| 3.2 | MVC | 16 |
| 3.3 | Databázový model | 18 |
| 3.4 | Wireframe — usporiadanie obrazovky a navigačný panel | 20 |
| 3.5 | Wireframe — registrácia | 20 |
| 3.6 | Wireframe — prihlasovanie | 21 |
| 3.7 | Wireframe — zoznam vyhľadávaní | 22 |
| 3.8 | Wireframe — vytváranie vyhľadávania | 23 |
| 3.9 | Wireframe — výsledky vyhľadávania | 24 |
| 3.10 | Wireframe — detail položky | 25 |
| 3.11 | Wireframe — modálne okno pre archívovanie | 25 |
| 3.12 | Wireframe — zákazky predajcov a položiek | 26 |
| 3.13 | Wireframe — archív | 26 |
| 3.14 | Wireframe — nápoveda | 27 |
| 3.15 | Wireframe — nastavenie | 28 |
| 4.1 | Diagram nasadenia | 35 |
| 5.1 | Štruktúra aplikácie | 38 |
| 6.1 | Registrácia | 50 |
| 6.2 | Prihlasovanie | 50 |
| 6.3 | Nastavenie | 51 |
| 6.4 | Tvorba vyhľadávania | 53 |
| 6.5 | Vyhľadávanie (obsahuje dáta z [1]) | 54 |
| 6.6 | Výsledky vyhľadávania (obsahuje dáta z [1]) | 54 |
| 6.7 | Vytvorenie záказu (obsahuje dáta z [1]) | 55 |

| | | |
|------|--|----|
| 6.8 | Zákazy (obsahuje dáta z [1]) | 55 |
| 6.9 | Detaily položky (obsahuje dáta z [1]) | 56 |
| 6.10 | Archivácia obrázkov (obsahuje dáta z [1]) | 57 |
| 6.11 | Archív (obsahuje dáta z [1]) | 57 |
| B.1 | Pomoc — obsahuje návody pre jednotlivé funkcionality aplikácie . | 67 |
| B.2 | Pomoc — Vytvorenie zákazu | 68 |
| B.3 | Pomoc — popis jednotlivých kritérií pri vytváraní vyhľadávania . . | 69 |
| B.4 | Pomoc — popis jednotlivých nastavení | 70 |

Zoznam častí kódu

| | | |
|-----|---|----|
| 5.1 | Dátová trieda | 40 |
| 5.2 | Repozitár | 40 |
| 5.3 | Vytvorenie fragmentu | 41 |
| 5.4 | Metóda pre vyhľadávanie | 42 |
| 5.5 | Vytvorenie kontextového objektu pre Trading API | 44 |
| 5.6 | Koncový bod pre pridanie novej kategórie | 46 |
| 5.7 | Koncový bod pre stiahnutie obrázku | 46 |

Úvod

Internetový obchod je novodobým výdobytkom, ktorý sa stáva stále populárnejším. Dôvodom je pohodlnosť, širší sortiment a často aj lepšie ceny. Jednotlivé položky sú často predávané v podobe dražby — je vymedzený určitý čas a na konci dražby položku získava ten, kto ponúkne najvyššiu cenu. Týmto spôsobom je možné kúpiť takmer čokoľvek zo sortimentu internetového obchodu a to často aj za veľmi výhodnú cenu.

Existuje veľa webových aukčných prostredí, kde sa takéto dražby uskutočňujú. Jedným z najväčších je internetový portál eBay. Užívateľ tam vyhľadáva požadovaný tovar tak, že vyšpecifikuje vyhľadávacie kritéria priamo na webovej stránke aukčného portálu. Môže však nastať prípad, že v momente keď užívateľ zadáva dané kritéria, žiadna položka nebude nájdená. Preto by bolo vhodné, keby existovala služba, ktorá bude v stanovených intervaloch monitorovať aukčný portál a v prípade výskytu novej položky spĺňajúcej zadané kritéria to oznámi užívateľovi.

V prvej kapitole opíšem cieľ tejto práce a dôvod, prečo som si danú tému zvolil. Pokračovať bude kapitola 2, kde vysvetlím existujúce riešenie tejto problematiky a následne zhrniem jednotlivé požiadavky a prípady použitia vo výslednej aplikácii. V ďalšej kapitole 3 popíšem výber formy aplikácie, architektúru, databázový model, možnosti ukladania obrázkov a návrh užívateľského rozhrania. Výber technológií zhrniem v kapitole 4, kde najprv spomeniem hlavné zvolené technológie pre tvorbu backendovej a frontendovej časti a následne popíšem ďalšie nástroje, ktoré som sa rozhodol použiť pri vývoji. Pokračovať bude kapitola 5, kde popíšem implementačné detaily, a následne kapitola 6, kde zhodnotím výsledky vytvorenej aplikácie.

Cieľ práce

Cieľ

Cielom tejto bakalárskej práce je navrhnúť a implementovať aplikáciu, ktorá bude slúžiť na vyhľadávanie a monitorovanie položiek na aukčnom portáli eBay — k tomu je najprv potrebné analyzovať existujúce riešenie, možnosti získavania dát a potrebné požiadavky aplikácie.

Užívateľ bude mať možnosť zadávať rôzne kritéria (cena, kľúčové slová, kategórie, ...), na základe ktorých aplikácia vyhledá položky, ktoré spĺňajú dané kritéria. Keďže eBay má pre viaceré krajiny lokalizované stránky, je potrebné, aby väčšinu týchto lokalizácií aplikácia podporovala. Užívateľ bude mať možnosť zobrazíť si výsledky vyhľadávania a následne ich triediť podľa určitých kritérií. Aplikácia bude všetky vyhľadávania v nastavenom časovom intervale aktualizovať a vlastníčkovi vyhľadávania oznámi prítomnosť nových položiek prostredníctvom emailovej, alebo SMS správy. Informácie o nájdených položkách budú archivované a užívateľ si bude môcť ukladať aj obrazové dáta týkajúce sa zvolených položiek. Prostredie, s ktorým užívateľ bude pracovať bude jednoduché, užívateľsky prívetivé a intuitívne.

Motivácia

Dôvod, prečo som si zvolil práve túto tému je to, že som sa chcel pokúsiť vytvoriť funkčnú aplikáciu od úplného začiatku. Ďalším dôvodom je snaha vytvoriť niečo, čo môže v budúcnosti pomôcť ľuďom ušetriť čas pri nakupovaní cez internet.

Analýza

Analýza je pre vývoj softvéru veľmi dôležitá a jej nedostatok môže často viesť k mnohým nepríjemnostiam pri neskorších častiach implementácie. Cena odstránenia chýb, ktoré sa vyskytnú pri finalizácii aplikácie je spravidla oveľa vyššia, ako cena vykonania dobrej analýzy, ktorá prispeje k prevencii vzniku týchto chýb. Práve preto má veľký význam vykonať ju čo najdôkladnejšie a vytvoriť si čo najpresnejší prehľad o funkcionalite, ktorá je požadovaná od výslednej aplikácie. V tejto kapitole som najprv popísal existujúce riešenie problematiky, následne funkčné a nefunkčné požiadavky na aplikáciu, a nakoniec prípady jej použitia.

2.1 Existujúce riešenie

V tejto časti som detailne popísal výsledky analýzy existujúceho riešenia od spoločnosti eBay. Výsledná aplikácia bude odstraňovať niektoré nedostatky tohto riešenia a zároveň podporovať jeho základnú funkcionalitu.

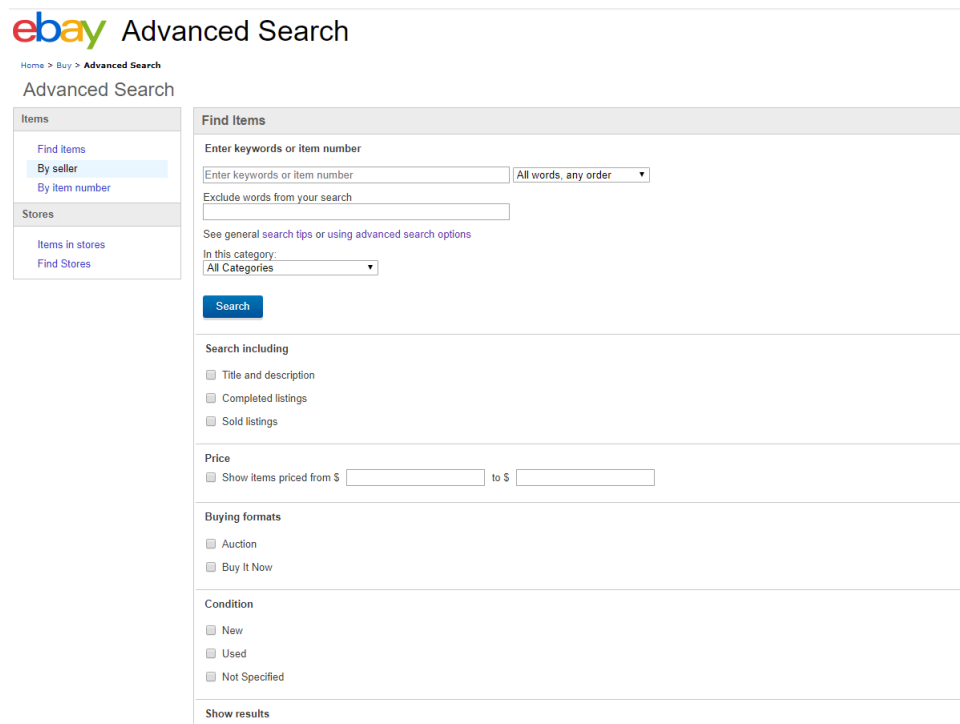
2.1.1 Vyhľadávanie eBay

Aukčný portál eBay [1] je webová stránka, ktorá sa zameriava na sprostredkovanie aukcií, ale taktiež je tam možné predávať a nakupovať normálnym spôsobom. Okrem www.ebay.com existujú ďalšie lokalizované stránky, ktoré poskytujú rovnaké služby pre rôzne regióny. Tieto stránky sa líšia hlavne v rozdelení kategórií, ale okrem toho poskytujú rovnaké služby na vyhľadávanie predávaných položiek (viď obrázok 2.1).

Užívateľ si môže zvoliť rôzne kritéria, na základe ktorých mu budú zobrazené výsledné predávané položky. Následne si to vyhľadávanie môže uložiť a daná stránka mu v prípade pribudnutia nových položiek, ktoré spĺňajú predtým navolené kritéria, odošle email s novými predávanými položkami. Existuje tiež možnosť zrušenia a editovania vytvoreného vyhľadávania. Táto služba poskytuje užívateľovi eBayu celkom jednoduchú možnosť sledovania predávaných

2. ANALÝZA

položiek, pretože oproti bežnému vyhľadávaniu stačí vytvorené vyhľadávanie len uložiť a následne bude užívateľ pravidelne notifikovaný o nových položkách prostredníctvom emailu. Medzi jednotlivé výhody a nevýhody som zaradil:



Obr. 2.1: eBay advanced search [1]

Výhody

- Vytvorenie vyhľadávania priamo na webovej stránke, kde je možné s nájdenými výsledkami priamo pracovať.
- Široká škála vyhľadávacích kritérií, kategórií a aspektov.

Nevýhody

- Vyhľadávanie na základe krajiny doručenia nie je podporované.
- Význam jednotlivých kritérií nie je jednoznačne vysvetlený pri tvorbe vyhľadávania.
- Neexistuje nastavenie periódy obnovovania. V prípade eBayu je perióda nastavená na jeden deň, preto sa o nových objektoch spĺňajúcich nastavené kritéria užívateľ dozvie len raz denne. Pri aukciách s možnosťou

„kúpiť hneď“ môže byť čas získania informácie o ich existencii rozhodujúci, teda v prípade využívania tohto vyhľadávania užívateľovi môžu uniknúť objekty o ktoré by potenciálne mal záujem.

- Notifikácie len pomocou emailových správ, čo v prípade, že sa užívateľ chce dozvedieť čo najskôr o predávanej položke, nemusí byť vždy to najlepšie.
- Vytvorené vyhľadávanie nájde prevažne len predávané objekty danej lokalizácie eBayu. Výnimkami sú iba obchodníci, ktorí spĺňajú špeciálne podmienky a tým je ich predaj viditeľný z viacerých lokalizovaných stránok.
- Cenové rozpätie ako kritérium vyhľadávania je možné zadať len v mene danej lokalizovanej stránky, kde sa vyhľadávanie vytvára.

2.1.2 Zhrnutie

Predchádzajúca časť opisuje riešenie, ktoré čiastočne spĺňa ciele tejto práce. Poskytuje dobré služby vyhľadávania a monitorovania predávaných položiek a v kombinácii s možnosťou priamej manipulácie s danými položkami tvorí dostatočne silný nástroj. Existuje však priestor na vylepšenie, predovšetkým odstránením nedostatkov zhrnutých v predchádzajúcej časti.

2.2 Prístup k dátam — eBay API

Pre prácu s jednotlivými lokalizáciami eBayu je potrebné mať spôsob, pomocou ktorého s nimi aplikácia bude môcť komunikovať a tak získavať žiadané dáta, ktoré následne môže ďalej spracovať. Presne k tomuto účelu poskytuje eBay verejné API [2] ktoré umožňuje nezávislým softvérom získavať poskytované dáta. Okrem získavania dát taktiež poskytuje možnosť vykonávať rôzne akcie užívateľského účtu prostredníctvom tokenu, ktorý aplikácia po súhlase užívateľa získa od eBayu.

2.2.1 Získanie prístupu

Aby vývojár mohol využívať toto API, potrebuje získať aplikačný prístupový kľúč, pomocou ktorého sa identifikuje pri využívaní funkcií, ktoré API poskytuje. Daný kľúč je nutnou súčasťou zaslaných žiadostí na toto API, avšak v prípade niektorých volaní je potrebný aj užívateľský prístupový token. Aplikačný prístupový kľúč je možné získať registráciou na [2] a následným vytvorením projektu a zažiadanim o kľúč pre daný projekt.

2.2.2 Typy

K rôznym účelom eBay poskytuje rozličné typy API. Jedno môže slúžiť napríklad na predávanie a nakupovanie, iné zase na vyhľadávanie rôznych štatistických údajov. V rámci tejto práce je potrebné mať možnosť získavať položky na základe rôznych špecifikácií, teda je potrebné využiť volanie takej metódy API, ktorá to umožní. Ďalej je potrebná možnosť získania rôznych špecifikácií, ktoré sa môžu pre jednotlivé lokalizácie líšiť. Pri využívaní API je taktiež potrebné zvoliť jedno z nasledujúcich dvoch prostredí:

Sandbox Slúži na testovacie účely. Je tam možné vytvárať rôzne instance položiek a účtov, ktoré je následne možné použiť na testovanie.

Production Obsahuje reálne dáta z produkčných serverov.

Z rôznych typov API, ktoré eBay poskytuje, som vybral za relevantné nasledujúce:

Finding API

Finding API poskytuje možnosť vyhľadávania predávaných objektov na základe definovaných kritérií. Pre použitie funkcií tohto API nie je potrebné mať užívateľský prístupový token, stačí len aplikačný prístupový kľúč. Z tohto API bude potrebná funkcia `findItemsAdvanced`, ktorá vráti predávané položky na základe zadaných kritérií. Táto funkcia združuje iné funkcie z tohto API, ktoré sa zameriavajú na špecifickú oblasť vyhľadávania objektov. Príkladom je funkcia `findItemsByKeywords`, ktorá vyhľadáva len na základe kľúčových slov a funkcia `findItemsByCategory` vyhľadávajúca predávané položky exkluzívne podľa kategórií. Funkcia `findItemsAdvanced` dokáže vyhľadávať súčasne podľa kategórií, kľúčových slov a ďalších kritérií, pre ktoré môžu existovať samostatné funkcie.

Trading API

Toto API poskytuje aplikáciám možnosť využívať viaceré možnosti eBay účtu. Pre využívanie tohto API potrebuje aplikácia získať autentizačný token od užívateľa. Prebieha to tak, že aplikácia presmeruje užívateľa na eBay, kde je potrebné povoliť prístup danej aplikácii k svojmu účtu. Následne aplikácia získa autentizačný token, ktorý je okrem aplikačného prístupového kľúča potrebný pri využívaní daného API. Jednotlivé eBay lokalizácie majú rôzne hierarchie kategórií.

Toto API, okrem možností správy užívateľského účtu, poskytuje taktiež iné možnosti, z ktorých v rámci tejto práce považujem za potrebné funkciu `GetCategories`, ktorá slúži na získanie kategórií zvolenej lokalizácie eBayu,

funkciu `GetCategoryMappings` na mapovanie kategórií pri ich zmenách a funkciu `GetItem`, pomocou ktorej je možné získať čo najdetailnejšie informácie o predávanej položke.

2.2.3 Obmedzenia

Vývojár nemôže jednotlivé typy API využívať neobmedzene, určitý typ API obsahuje isté maximum (viď obrázok 2.2), po ktorého dosiahnutí dané API odpovie na každé ďalšie volanie chybovou správou. Počet volaní jednotlivého API je každodenne vynulovaný, po čom je API možné naďalej využívať.

| API | All Applications | Compatible Applications |
|---------------------------------------|-------------------------------------|-------------------------------------|
| Search | | |
| Finding API | 5,000 API calls per day | 1.5 Million API calls per day |
| Buy (Browse, Commit, Checkout) | | |
| /buy/feed * | 10,000 API calls per day | 10,000 API calls per day |
| /buy/browse * | 5,000 API calls per day | 100,000 API calls per day |
| /buy/marketing * | 5,000 API calls per day | 10,000 API calls per day |
| /buy/offer * | 5,000 API calls per day | 50,000 API calls per day |
| /buy/order * | 5,000 API calls per day | 10,000 API calls per day |
| Buy (Browse & Commit) | | |
| Shopping API | 5,000 API calls per day | 1.5 Million API calls per day |
| Merchandising API | 5,000 API calls per day | 1.5 Million API calls per day |
| PlaceOffer * | (see Trading API limits under Sell) | (see Trading API limits under Sell) |
| Sell (SKU-Based Inventory) | | |
| /sell/account | 25,000 API calls per day | 25,000 API calls per day |
| /sell/inventory | 2 Million API calls per day | 5 Million API calls per day |
| /sell/compliance | 5,000 API calls per day | 1 Million API calls per day |
| /sell/fulfillment | 100,000 API calls per day | 2.5 Million API calls per day |
| /sell/marketing | 10,000 API calls per day | 25,000 API calls per day |
| /sell/analytics | 100 API calls per day | 500 API calls per day |

Obr. 2.2: eBay API limity [2]

Pre bežné aplikácie sú dané maxima dosť obmedzujúce, avšak je možné ich navýšiť v prípade kompatibility medzi aplikáciou a eBay API. Kompatibilitu je možné docieľiť splnením určitých požiadaviek a následným podaním žiadosti¹.

¹viď <https://developer.ebay.com/support/app-check>

2.3 Požiadavky

Modelovanie požiadaviek je neoddeliteľnou súčasťou tvorby softvéru, pretože bez nich nie je jasné čo a ako má fungovať. Požiadavky, ktoré by mala táto aplikácia pokrývať, som vytvoril na základe zadania a konzultácií s vedúcim tejto bakalárskej práce. Požiadavky rozdelené na funkčné a nefunkčné sú zhrnuté v nasledujúcich častiach.

2.3.1 Funkčné požiadavky

Tento typ požiadaviek popisuje aké funkčnosti má system podporovať. Popisujú konkrétne funkcionality, ktoré sú jednoducho testovateľné. V rámci tejto práce som vytvoril nasledujúce funkčné požiadavky:

Registrácia a prihlásenie užívateľa Aplikácia bude podporovať viac užívateľských účtov a tak je potrebné mať možnosť registrácie a následného prihlásenia.

Vytvorenie vyhľadávania Užívateľ bude mať možnosť vytvoriť si vyhľadávanie, kde si zvolí lokalizáciu eBay stránky a následne vyhľadávacie kritéria, predovšetkým kategórie a kľúčové slová, poprípade ďalšie upresňujúce informácie ako sú cena, počet ponúk a iné.

Zobrazenie a editovanie vyhľadávaní Užívateľ bude môcť svoje už vytvorené vyhľadávania zobraziť, upraviť a vymazať.

Zobrazenie a triedenie výsledkov vyhľadávania Užívateľ si bude môcť zobraziť nájdené výsledky svojho vyhľadávania a bude mať možnosť ich triediť podľa rôznych kritérií.

Zobrazenie detailov konkrétneho výsledku Užívateľ si pri zobrazení výsledkov vyhľadávania bude môcť zobraziť detaily konkrétneho výsledku. Súčasťou detailov o predaji budú obrázky a vývoj ceny predávanej položky.

Archivácia obrázkov položiek s príslušnou kategorizáciou Užívateľ si bude môcť pri prehliadaní detailov konkrétneho výsledku vyhľadávania archivovať obrázky výsledku pod určitou kategóriou. Bude môcť uložené obrázky z archívu aj vymazať.

Zobrazenie archivovaných obrázkov Užívateľ bude môcť zobraziť archivované obrázky podľa kategórií.

Manažment zákazov položiek a predajcov Užívateľ si bude môcť pridávať a odstraňovať zakazy určitých predajcov, alebo položiek v rámci konkrétneho vyhľadávania, alebo globálne pre celý užívateľský účet. Užívateľ bude môcť určitého predajcu, poprípade jednotlivý výsledok nejakého

vyhľadávania zakázať buďto len na jedno konkrétne vyhľadávanie, alebo globálne pre všetky vyhľadávania. Následne bude môcť tieto zákazy aj vymazať.

Preferencie a peňažná mena užívateľa Užívateľ si bude môcť upraviť určité preferencie, na základe ktorých sa mu automaticky vyplnia niektoré údaje pri vytváraní nového vyhľadávania. Súčasťou týchto preferencií bude aj možnosť zmeny peňažnej meny užívateľského účtu. To značí, že cenové kritéria a peňažné detaily predávaných objektov budú užívateľovi zobrazené vo vybranej mene.

Obnovenie vyhľadávania Aplikácia bude pravidelne po zvolenom časovom intervale obnovovať výsledky jednotlivých vyhľadávaní.

E-mailové notifikácie O nových položkách v rámci vytvoreného vyhľadávania dostane užívateľ oznámenie prostredníctvom emailovej správy, ktorá bude obsahovať najdôležitejšie detaily nájdených položiek.

Mobilné notifikácie Užívateľ bude mať možnosť pridať telefónne číslo, na ktoré mu budú doručované oznámenia o nových nájdených položkách vo forme SMS správy.

2.3.2 Nefunkčné požiadavky

Nefunkčné požiadavky neopisujú čo má systém robiť, ale ako to má robiť. Jedná sa napríklad o požiadavky na výkon, bezpečnosť, dostupnosť a spoľahlivosť systému. Popisujú požadované vlastnosti aplikácie, ktoré ale nemusia byť známe z jej zadania. Sú však rovnako dôležité pre správnu funkcionálnosť aplikácie. V rámci tejto práce som vytvoril nasledujúce nefunkčné požiadavky:

Jednoduché a užívateľsky prívetivé prostredie Systém by mal byť intuitívny a jednoduchý na používanie.

Responzivita Aplikácia by mala byť správne zobrazená pre rôzne zariadenia (počítač, tablet, mobil). Cieľom je, aby užívateľ mal možnosť plnohodnotne používať aplikáciu aj na mobilnom zariadení.

Rozšíriteľnosť Je vhodné, aby bolo možné v budúcnosti pridávať ďalšie rozšírenia, teda je potrebný dobrý dizajn a dokumentácia.

2.4 Prípady použitia

Prípady použitia znázorňujú akcie, ktoré užívatelia môžu vykonávať a odrážajú jednotlivé funkčné požiadavky (viď obrázok 2.3).

Registrovať Užívateľ sa môže zaregistrovať do aplikácie vyplnením registračného formulára, kde vyplní svoju emailovú adresu a heslo. Aplikácia následne zašle potvrdzovací email na zadanú emailovú adresu. Po potvrdení registračného emailu sa užívateľ môže prihlásiť do aplikácie.

Prihlásiť Užívateľ sa môže prihlásiť pomocou emailu a hesla, ktoré zadal pri registrácií. Je však nutné, aby predtým mal svoj účet aktivovaný.

Zmeniť užívateľské preferencie Užívateľ môže zmeniť svoje preferencie — kritéria, ktoré sú predvolené pri vytváraní nového vyhľadávania, predvolené nastavenie mobilných notifikácií pre nové vyhľadávanie, zobrazenú peňažnú menu a čas automatického odhlásenia pri neaktivite.

Pridať telefónne číslo Užívateľ môže pridať svoje mobilné telefónne číslo. Aplikácia mu následne zašle potvrdzovací kód v SMS správe, ktorým užívateľ verifikuje svoje telefónne číslo.

Zmeniť heslo Užívateľ môže zmeniť svoje prihlasovacie heslo.

Zobraziť vyhľadávania Užívateľ si môže zobraziť svoje vyhľadávania.

Vytvoriť vyhľadávanie Užívateľ môže vytvoriť vyhľadávanie pomocou formulára, v ktorom zadá požadované kritéria — lokalizáciu eBay, krajinu doručenia, typ predaja, kategórie, kľúčové slová, rozmedzie konca predaja, ponúk, cien a zopár ďalších možností. Jednou z vecí, ktorú musí užívateľ nastaviť je čas aktualizácie vyhľadávania.

Editovať vyhľadávanie Užívateľ môže zmeniť kritéria vyhľadávania rovnako ako pri tvorbe nového vyhľadávania.

Pozastaviť vyhľadávanie Užívateľ môže pozastaviť vyhľadávanie — vyhľadávanie sa nebude ďalej aktualizovať.

Spustiť vyhľadávanie Užívateľ môže pozastavené vyhľadávanie opätovne spustiť, vďaka čomu sa dané vyhľadávanie začne aktualizovať.

Vymazať vyhľadávanie Užívateľ môže vymazať vyhľadávanie, ktoré predtým vytvoril.

Upraviť mobilné notifikácie vyhľadávania Užívateľ môže pre určité vyhľadávanie definovať aké typy položiek mu má aplikácia oznamovať pomocou SMS správ.

Zobraziť výsledky vyhľadávania Užívateľ si môže zobraziť nájdené výsledky zvoleného vyhľadávania.

Triediť výsledky vyhľadávania Užívateľ môže výsledky vyhľadávania po ich zobrazení triediť podľa rôznych kritérií.

Pridať zákaz položky Užívateľ môže po zobrazení výsledkov vyhľadávania pridať zákaz určitej položky pre všetky vyhľadávania, alebo len pre dané vyhľadávanie, ktoré si zobrazil. Po vytvorení zákazu aplikácia vymaže z vyhľadávani všetky položky, ktoré podliehajú pod daný zákaz.

Pridať zákaz predajcu Užívateľ môže po zobrazení výsledkov vyhľadávania pridať zákaz predajcu vybranej položky pre všetky vyhľadávania, alebo len pre dané vyhľadávanie, ktoré si zobrazil. Po vytvorení zákazu aplikácia vymaže z vyhľadávani všetky položky, ktoré podliehajú pod daný zákaz.

Zobraziť zákazy pre vyhľadávanie Užívateľ si môže zobrazit zoznam zákazov predajcov a položiek pre určité vyhľadávanie.

Zobraziť zákazy pre užívateľa Užívateľ si môže zobrazit zoznam zákazov predajcov a položiek viazaných pre všetky vlastnené vyhľadávania.

Vymazať zákaz pre vyhľadávanie Užívateľ môže po zobrazení zoznamu zákazov pre vyhľadávanie vymazať jednotlivé zákazy.

Vymazať zákaz pre užívateľa Užívateľ môže po zobrazení zoznamu zákazov pre užívateľa vymazať jednotlivé zákazy.

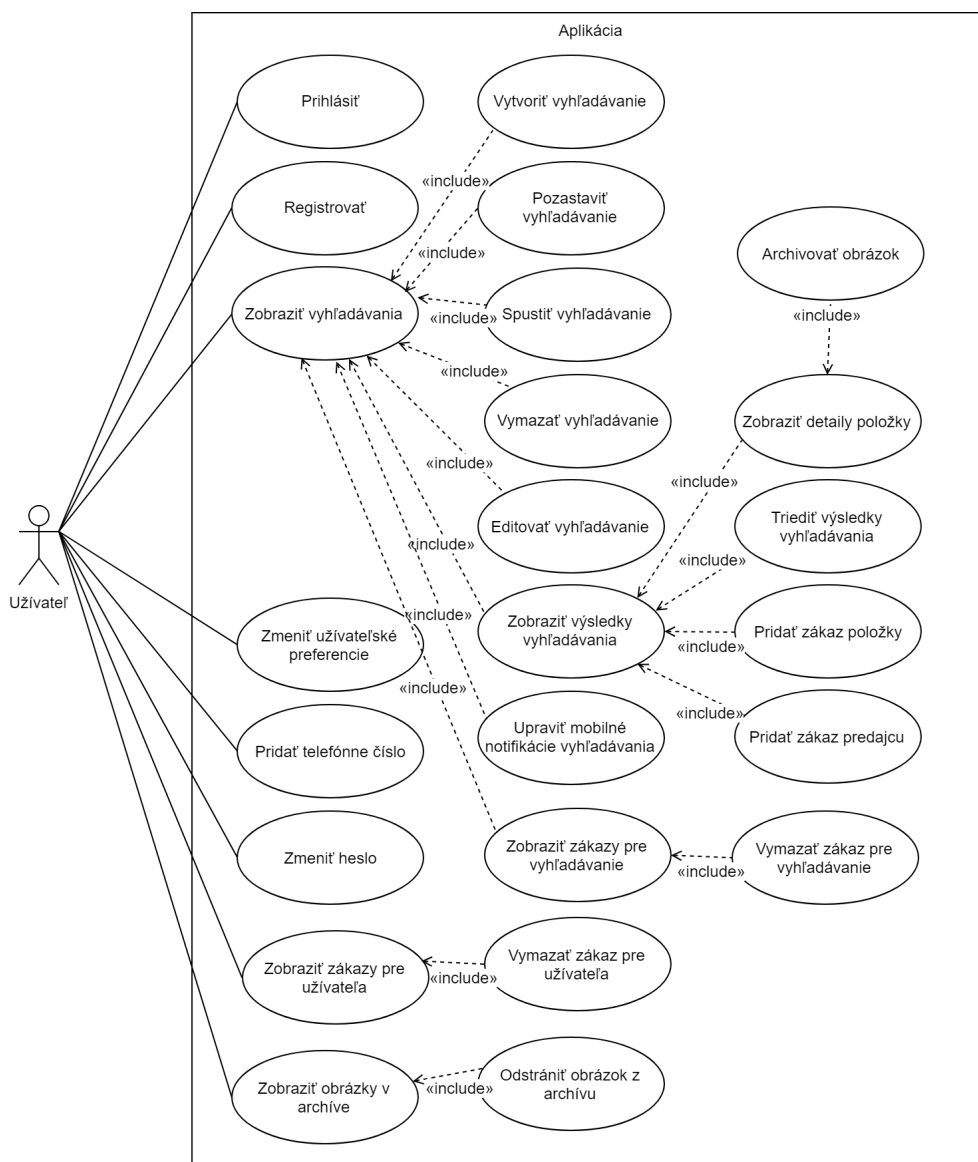
Zobraziť detaily položky Užívateľ môže po zobrazení výsledkov vyhľadávania zobrazit detaily nájdenej položky.

Archivovať obrázok Užívateľ môže po zobrazení detailu položky jednotlivé nájdene obrázky archivovať pod určitou kategóriou.

Zobraziť obrázky v archíve Užívateľ môže zobrazit archivované obrázky.

Odstráneniť obrázok z archívu Užívateľ môže po zobrazení archívu obrázkov jednotlivé obrázky vymazať.

2. ANALÝZA



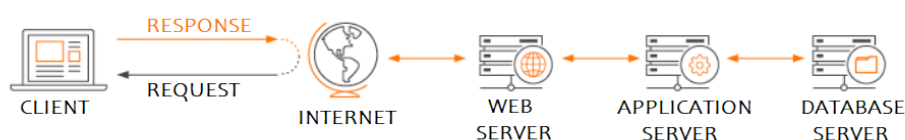
Obr. 2.3: Prípady použitia

Návrh

Návrh je ďalšia dôležitá časť vývoja softvéru, pretože sa v nej rozhoduje ako bude aplikácia implementovaná. V tejto kapitole opíšem formu aplikácie, ktorú som sa v rámci implementácie rozhodol použiť. Nasledovať bude architektúra, model databázy, spôsob archivovania obrázkov a návrh užívateľského rozhrania.

3.1 Forma aplikácie

Pre koncovú aplikáciu a služby, ktoré poskytuje, je dôležité ako k nim užívateľ bude môcť pristupovať. Rozhodoval som sa medzi desktopovou a webovou aplikáciou (viď obrázok 3.1). „Webová aplikácia je počítačový program, ktorý využíva webové prehliadače a webové technológie na vykonávanie úloh cez internet“ [3]. Moderné webové aplikácie poskytujú takmer rovnaké možnosti ako desktopové, avšak je nutné internetové pripojenie používateľa. Okrem toho má však mnoho výhod, napríklad nenúti užívateľa inštalovať samotnú aplikáciu a taktiež nie je potrebné inštalovať ďalšie aktualizácie. Ďalšou veľkou výhodou je možnosť využívania aplikácie pomocou mobilných zariadení. Práve preto som zvolil túto formu aplikácie v rámci tejto práce.



Obr. 3.1: Webová aplikácia (prevzaté z [3])

3.2 Architektúra

Softvérová architektúra je štruktúra celého softvéru, teda udáva ako sú rozložené jednotlivé komponenty a ako medzi sebou komunikujú. Aplikáciu je v rámci implementácie potrebné rozdeliť na väčšie celky, ktoré poskytujú určitú funkcionálnosť, pretože to zjednodušuje vývoj, rozširovanie a údržbu kódu.

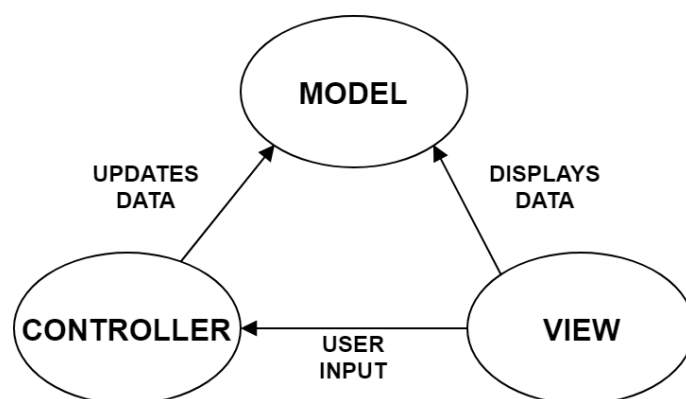
Model-View-Controller (MVC) je jedným z typov softvérovej architektúry. V danom type sú aplikácia a jej vývoj rozdelené do troch vzájomne prepojených častí. Vďaka tomu je možné sa zamerať len na konkrétnu časť aplikácie a jednotlivé časti môžu byť jednoducho vyvíjané paralelne. [4] Komunikácia jednotlivých častí je znázornená na obrázku 3.2. Pri použití MVC je aplikácia rozdelená na nasledujúce časti:

Model Predstavuje dátovú vrstvu, ktorá má na starosti všetko čo sa týka ukladania a načítania dát, ktoré aplikácia využíva. Ďalej sa tu nachádza aplikačná logika.

View Predstavuje prezentačnú vrstvu, ktorá je zodpovedná za zobrazenie obsahu užívateľovi, teda predstavuje užívateľské rozhranie.

Controller Predstavuje kontrolnú vrstvu, ktorá spracováva (prijíma a validuje) vstupy od užívateľa z prezentačnej vrstvy a následne mu posiela odpoveď, ktorá obsahuje dáta z dátovej vrstvy.

MVC architektúra je v rámci vývoja webových aplikácií veľmi populárna a existuje veľké množstvo frameworkov, ktoré jej použitie vynucujú. Tento typ architektúry som sa rozhodol použiť pretože poskytuje rozdelenie, ktoré mi jednotlivé časti umožní vyvíjať nezávislo a v budúcnosti zjednoduší rozširovanie aplikácie.



Obr. 3.2: MVC

3.3 Databázový model

Databázový model zobrazuje logickú štruktúru databázy, vrátane vzťahov a obmedzení, ktoré určujú spôsob ukladania a prístupu k údajom [5]. Pre potreby uchovávaní dát tejto aplikácie som zvolil relačnú databázu. Relačné databázy uchovávajú dáta v tabuľkách, kde jednotlivé typy objektov predstavujú tabuľky a objekty určitého typu predstavujú riadky danej tabuľky. Referencie na objekty sú ukladané v podobe cudzích kľúčov, ktoré predstavujú odkazy na záznamy v iných tabuľkách prostredníctvom ich identifikátorov.

Je dôležité si premyslieť, aké všetky objekty bude potrebné uchovávať a následne navrhnúť dobrý model, podľa ktorého bude možné danú databázu vytvoriť. Hlavné objekty, ktoré budem v rámci tejto databázy uchovávať sú:

User Pre užívateľa je potrebné uchovávať jeho email, prihlasovacie heslo, telefónne číslo, rôzne aktivačné príznaky a kódy. Rovnako je potrebné ukladať užívateľské preferencie, ktoré slúžia hlavne pre vytváranie nového vyhľadávania, ale taktiež obsahujú ďalšie nastavenia pre zjednodušenie využívania aplikácie. Nasledujúcimi položkami, ktoré patria k užívateľovi, sú kategórie obrázkov a obrázky, ktoré si užívateľ uložil. Nakoniec je potrebné uložiť zoznam položiek, ktoré mu už boli oznámené, a zoznam zakázaných predajcov a položiek.

Search Pre vyhľadávanie je potrebné uchovávať kritéria vyhľadávaných objektov, zoznam zakázaných predajcov a položiek, periodu aktualizácie a čas poslednej aktualizácie. Na vyhľadávanie sa taktiež viaže zoznam položiek, ktoré sa v danom vyhľadávaní od poslednej aktualizácie nachádzajú. Ďalej obsahuje preferencie, ktoré obsahujú napríklad formu zobrazenia výsledkov vyhľadávania (triediace kritérium, počet položiek) a povolené mobilné notifikácie.

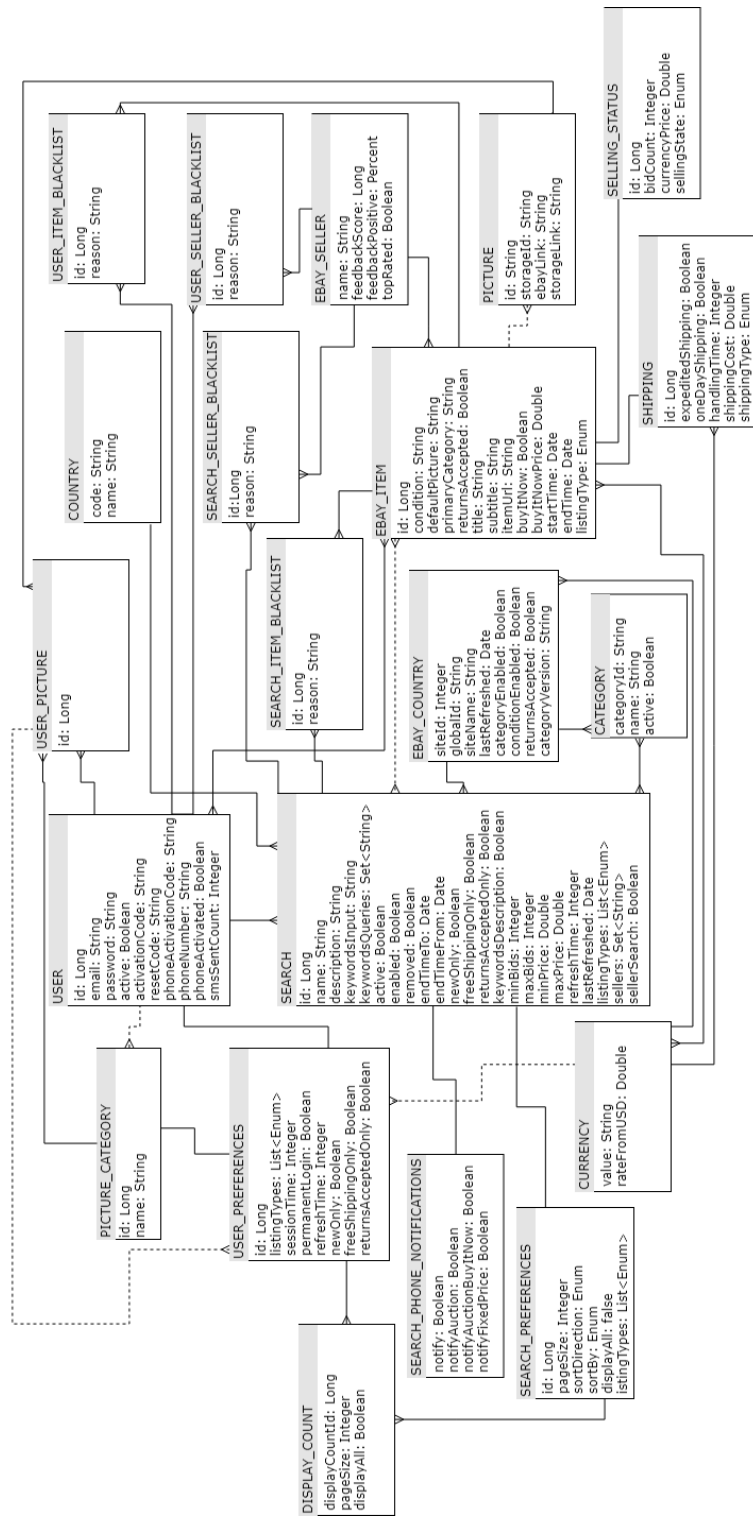
EbayItem Predávaná položka obsahuje detaily týkajúce sa predaja — krajina predaja, doprava, kategória položky, rôzne ceny (najvyššia ponuka, ponuka „kúpiť hneď“), počet ponúk, začiatok a koniec predaja a predajcu. K položke sa taktiež viaže zoznam obrázkov, ktoré sú k danej položke k dispozícii.

EbayCountry Pre rôzne lokalizácie eBayu je potrebné uchovávať identifikačné údaje, verziu kategórií a aktuálne kategórie, ktoré sa viažu k danej lokalizácii.

Picture Pre jednotlivé obrázky predávaných položiek je potrebné ukladať link na eBay úložisko a následne odkaz a identifikátor na aplikačnom úložisku, kde bude aplikácia archivovať dané obrázky.

Podrobnejší popis položiek jednotlivých objektov a ďalších potrebných objektov je možné vidieť na obrázku 3.3.

3. NÁVRH



Obr. 3.3: Databázový model

3.4 Archivácia obrázkov

Jedným z cieľov tejto práce je mať možnosť ukladať obrázky jednotlivých predávaných položiek. K tomu je potrebné zvoliť nejaké úložisko, kde bude možné bezpečne tieto obrázky uchovávať. Je možné použiť jednu z týchto možností:

Server Ukladať ich priamo na serveri kde beží aplikácia je síce rýchle riešenie, avšak môže priniesť problémy pri zmene servera, na ktorý je aplikácia nasadená. V tom prípade by bolo nutné obrázky zálohovať a následne ich na novom serveri obnoviť, resp. riešiť ich migráciu iným spôsobom.

Databáza Toto riešenie je síce jednoduché na implementáciu, ale môže spomaľovať chod aplikácie, keďže transakčné spracovanie veľkých dát je pomalé.

Externé úložisko V rámci tejto práce som sa rozhodol použiť toto riešenie. Nemusí to byť síce veľmi rýchle, ale nebude to spomaľovať chod databázy a prechod na iný server bude v prípade nutnosti jednoduchší.

3.5 Uživatelské rozhranie

V tejto časti budem písať o návrhu jednotlivých užívateľských obrazoviek, podľa ktorých implementujem užívateľské rozhranie. Cieľom tohto návrhu je vytvoriť podklad pre privetivé rozhranie, ktoré bude jednoduché na používanie. Jednotlivé návrhy som vytvoril použitím [6]. Každá z jednotlivých obrazoviek bude obsahovať navigačný panel (viď obrázok 3.4) pomocou ktorého bude možný pohyb medzi hlavnými časťami aplikácie.

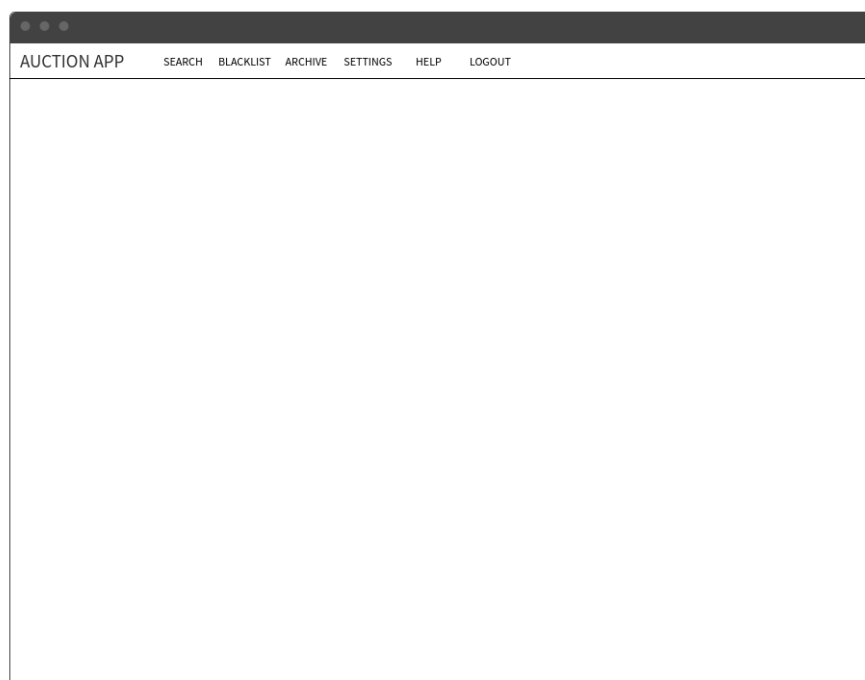
3.5.1 Registrácia

Užívateľ musí mať vytvorený účet pre používanie aplikácie. Tento účet si bude môcť vytvoriť vyplnením registračného formulára a následnou aktiváciou účtu. Pre registráciu som vytvoril wireframe na obrázku 3.5, kde je vidieť, že obsahuje len jednoduchý formulár, ktorý užívateľ pre úspešnú registráciu potrebuje vyplniť a následne potvrdiť tlačidlom. Obrazovka ďalej obsahuje tlačidlo, ktoré užívateľa presmeruje na prihlasovaciu stránku.

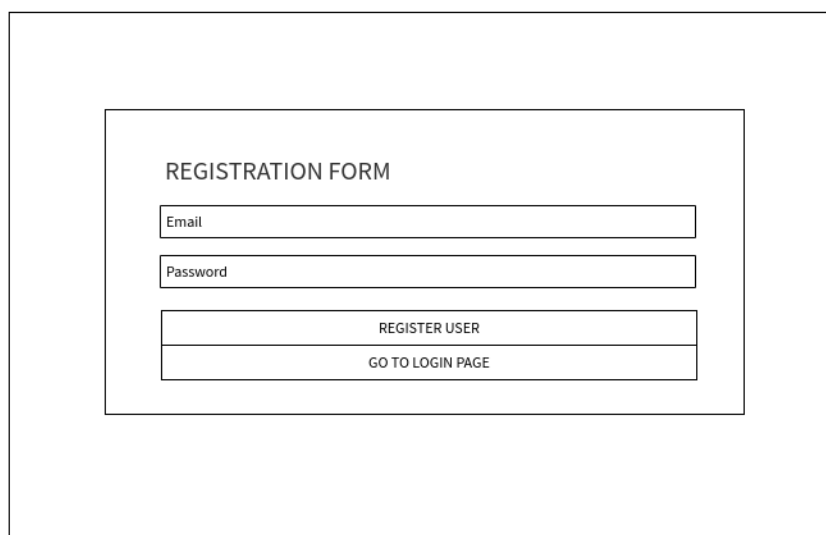
3.5.2 Prihlasovanie

Na obrázku 3.6 je možné vidieť wireframe pre prihlasovaciu obrazovku. Obsahuje jednoduchý formulár, kde užívateľ zadá svoje prihlasovacie meno (email) a heslo a následne klikne na tlačidlo pre prihlásenie. Okrem toho obsahuje dve ďalšie tlačidlá, jedno na presmerovanie na registračnú stránku a druhé na obnovenie hesla.

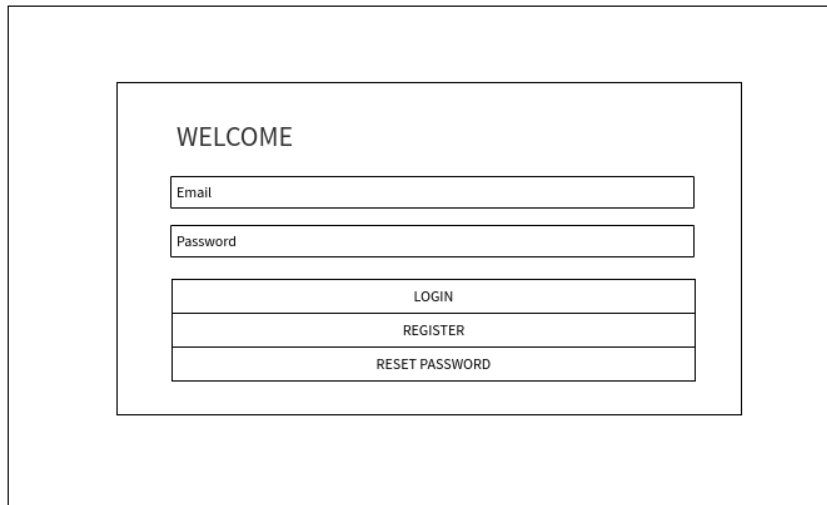
3. NÁVRH



Obr. 3.4: Wireframe — usporiadanie obrazovky a navigačný panel



Obr. 3.5: Wireframe — registrácia



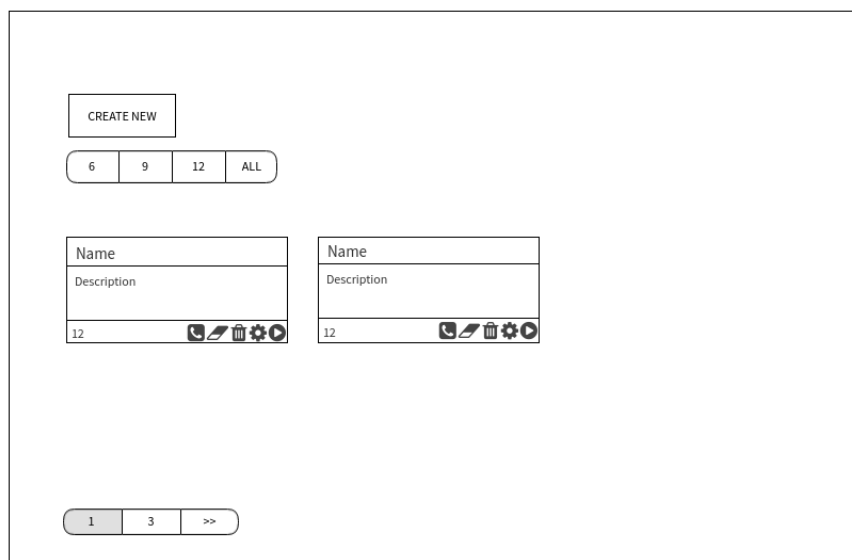
Obr. 3.6: Wireframe — prihlasovanie

3.5.3 Vyhľadávanie

Ako je vidieť na obrázku 3.7, obrazovka s vyhľadávania bude obsahovať tlačidlo, po ktorého kliknutí bude užívateľ presmerovaný na stránku pre vytvorenie nového vyhľadávania. Pod týmto tlačidlom sa nachádza výber počtu zobrazených vyhľadávanií pre jednu stránku. Ďalej táto obrazovka obsahuje zoznam všetkých vyhľadávanií, kde jednotlivé vyhľadávanie predstavuje karta, ktorá je zložená z 3 častí. Prvá časť obsahuje názov vyhľadávania, druhá jeho popis a posledná obsahuje počet výsledkov daného vyhľadávania a kliknutelné ikony umožňujúce ďalšie akcie. Kliknutie na celú kartu presmeruje užívateľa na stránku s výsledkami daného vyhľadávania. Jednotlivé ikony karty umožňujú úpravu notifikácií, úpravu zakázaných predajcov/objektov, editáciu, vymazanie a pozastavenie daného vyhľadávania. Ikona na správu zákazov presmeruje užívateľa na stránku so zákazmi. Ďalšie ikony otvoria modálne okno, ktoré užívateľovi podrobnejšie vysvetlí danú možnosť a užívateľ tam môže potvrdiť svoje rozhodnutie. Pod zobrazenými vyhľadávania sa nachádza výber stránky.

3.5.4 Vytváranie vyhľadávania

Obrazovka obsahuje formulár, kde užívateľ zadáva rôzne kritéria, na základe ktorých budú vyhľadávané jednotlivé objekty. Formulár obsahuje kombinované, textové a zaškrŕavacie polia pre výber kritérií a pre nastavenie ďalších vlastností vyhľadávania — meno, popis a perióda aktualizácie. Návrh tejto obrazovky je možné vidieť na obrázku 3.8.



Obr. 3.7: Wireframe — zoznam vyhľadávanií

3.5.5 Výsledky vyhľadávania

Na obrázku 3.9 je návrh obrazovky s výsledkami vyhľadávania. Obrazovka obsahuje kombinované pole pre zoradenie položiek, výber typu predaja a výber počtu zobrazených položiek pre stránku. Následne obsahuje výsledky vo forme kariet. Každá karta je rozdelená horizontálne na 4 časti. Prvá časť obsahuje obrázok výsledku. V druhej časti sú detaily ohľadom výsledku — aktuálna cena, počet ponúk, ak je to aukcia, cena „kúpiť hneď“ a cena dopravy. Tieto detaily predstavujú údaje na základe ktorých bude možné triediť výsledky. Nasleduje ďalšia časť, v ktorej je názov položky. Posledná časť obsahuje dátum ukončenia predaja a dve ikony, ktoré otvoria modálne okná s možnosťou vytvorenia zákazu pre konkrétnu položku alebo predajcu. Modálne okno obsahuje textové pole pre zadanie dôvodu, následne výber rozsahu zákazu a tlačidlo na potvrdenie. Pod zobrazenými položkami sa nachádza výber stránky.

3.5.6 Detail položky

Táto obrazovka obsahuje detaily položky nájdenej v určitom vyhľadávaní. Ako je vidieť na obrázku 3.10, obsahuje tri časti. V prvej časti sú všetky dôležité informácie o predávanom objekte (cena, lokácia, doprava, predajca, ...). V druhej časti sa nachádza galéria obrázkov daného výsledku. Pod galériou sa nachádza tlačidlo na archivovanie obrázkov 3.11. Po jeho kliknutí sa objaví modálne okno, kde si užívateľ bude môcť archivovať vybrané obrázky. Nakoniec sa na obrazovke nachádza graf, ktorý zobrazuje vývoj ceny položky.

Obr. 3.8: Wireframe — vytváranie vyhľadávania

3.5.7 Zákazy predajcov a položiek

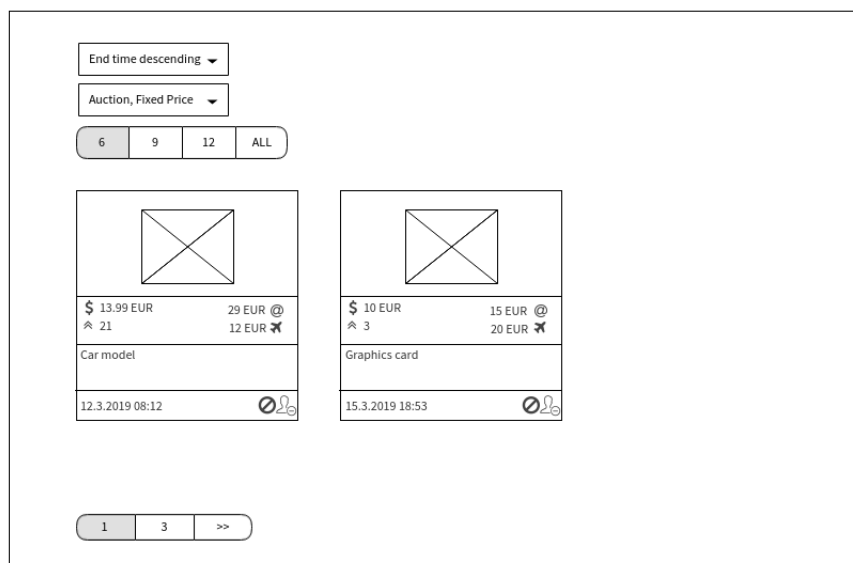
Na obrázku 3.12 je wireframe pre obrazovku určenú na manažment zákazov predajcov a položiek. Obrazovka obsahuje kombinované pole na prepnutie medzi zoznamom predajcov a položiek, a tabuľku obsahujúcu jednotlivé objekty vybraného okruhu zákazu.

Tabuľka obsahujúca zákazy predajcov obsahuje riadky, kde dané stĺpce predstavujú meno predajcu, dôvod na zákaz a ikonu na zrušenie zákazu. Tabuľka pre zákaz položky je podobná, akurát obsahuje ešte identifikátor daného objektu. Pod tabuľkou sa nachádza výber stránky.

3.5.8 Archivované obrázky

Jednotlivé archivované obrázky bude užívateľ môcť zobrazíť. Podľa obrázku 3.13 na to bude slúžiť obrazovka, ktorá obsahuje kombinované pole na výber

3. NÁVRH



Obr. 3.9: Wireframe — výsledky vyhľadávania

kategórie a následne obrázky, ktoré patria do vybranej kategórie. Každý obrázok bude obsahovať dve ikony. Jedna predstavuje odkaz na eBay stránku s položkou, ktorej patrí daný obrázok a druhá je na vymazanie obrázku zo zobrazenej kategórie, teda pri kliknutí sa otvorí modálne okno, ktoré užívateľovi bližšie vysvetlí význam danej akcie a následne užívateľ musí svoje rozhodnutie potvrdiť. Pod zobrazenými obrázkami sa nachádza výber stránky.

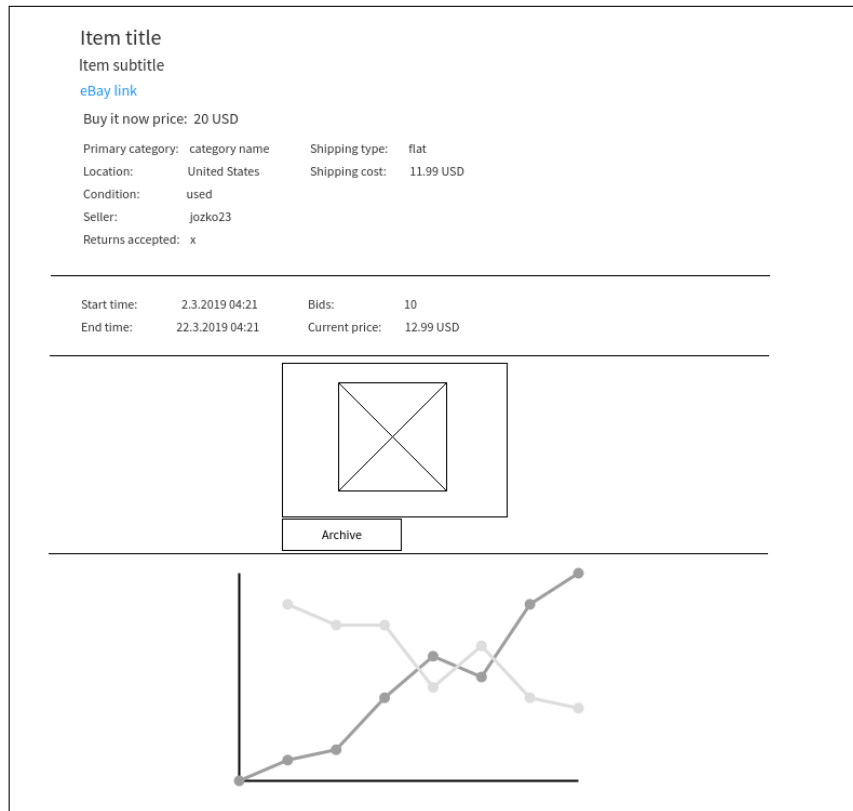
3.5.9 Nápoveda

Táto obrazovka obsahuje nápovedy k rôznym funkcionalitám aplikácie. V súlade s obrázkom 3.14 obsahuje zoznam funkcií, ku ktorým sú poskytnuté nápovedy. Keď užívateľ klikne na jednotlivú funkciu v zozname, obrazovka sa mu presunie na návod k danej funkcii. Jednotlivé nápovedy sú pod zoznamom.

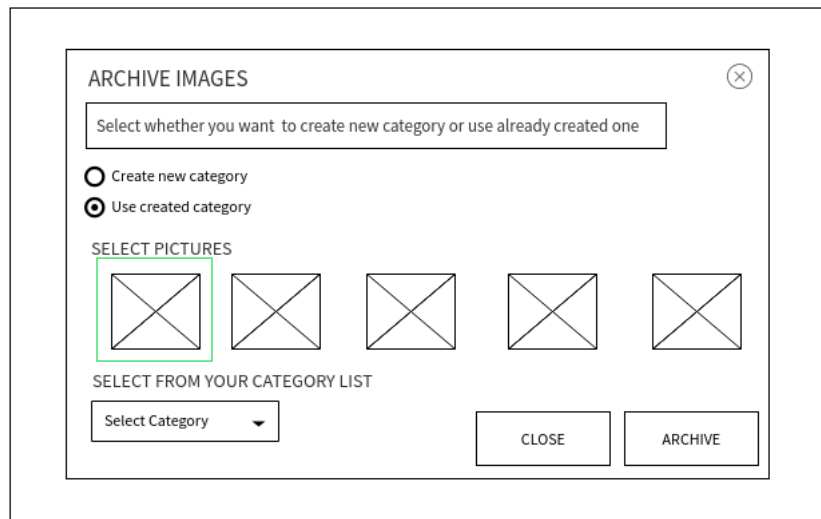
3.5.10 Nastavenie

Pre zmenu užívateľských nastavení som vytvoril wireframe viditeľný na obrázku 3.15. Obrazovka obsahuje dva formuláre. Jeden je na zmenu užívateľských preferencií a druhý na zmenu hesla.

V prvej časti formulára pre zmenu užívateľských preferencií je výber niektorých kritérií, ktoré budú predvyplnené pri tvorbe nového vyhľadávania a taktiež predvolené mobilné notifikácie pre nové vyhľadávanie. Druhá časť obsahuje výber dĺžky prihlásenia — užívateľ si môže zvoliť prihlásenie natrvalo alebo čas, po ktorého uplynutí bez prejavu aktivity, bude odhlásený.



Obr. 3.10: Wireframe — detail položky



Obr. 3.11: Wireframe — modálne okno pre archívovanie

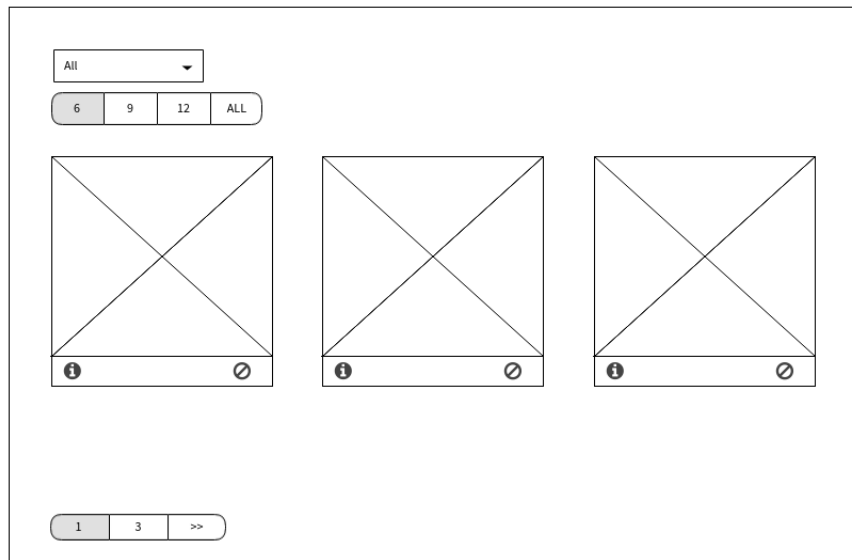
3. NÁVRH

A wireframe for a 'Sellers' page. At the top left is a dropdown menu labeled 'Sellers'. Below it is a table with three columns: 'NAME', 'REASON', and 'DELETE'. The table contains three rows of data. Below the table is a pagination control with three buttons: '1', '3', and '>>'.

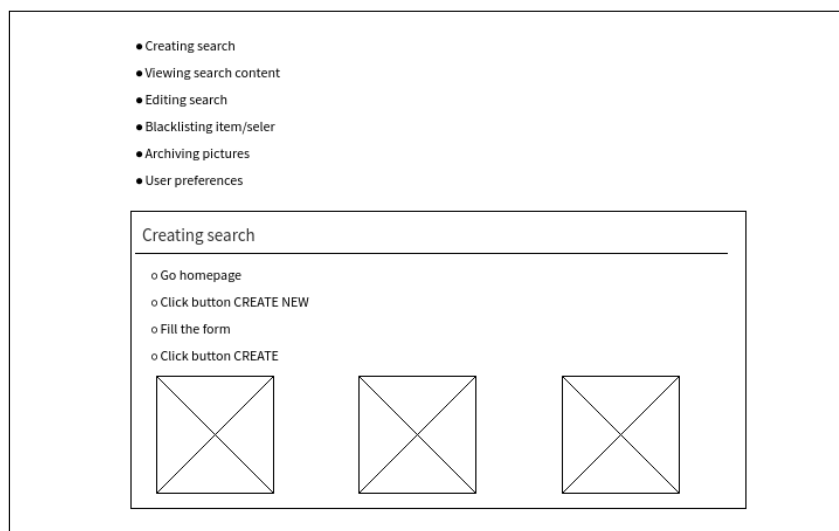
| NAME | REASON | DELETE |
|---------|---------------|--------|
| jozko12 | too expensive | X |
| tomas54 | dont like | X |
| ivan436 | | X |

1 3 >>

Obr. 3.12: Wireframe — zákazy predajcov a položiek



Obr. 3.13: Wireframe — archív



Obr. 3.14: Wireframe — nápoveda

3. NÁVRH

USER PREFERENCES

EBAY SITE

LISTING TYPES

SHIPPING COUNTRY

REFRESH TIME(MIN)

New only
 Free shipping
 Returns accepted

Enable notifications
 Notifications for auction without buy it now listings
 Notifications for auction with buy it now listings
 Notifications for fixed price listings

Stay logged in
INACTIVITY LOGOUT(MIN)

ACCOUNT CURRENCY

CHANGE PASSWORD

Obr. 3.15: Wireframe — nastavenie

Technológie

V tejto kapitole zhrniem hlavné technológie, ktoré som sa rozhodol použiť k vytvoreniu aplikácie v rámci tejto práce. Aplikácia má za úlohu poskytnúť užívateľovi možnosť vytvárať rôzne sledovania položiek na webových stránkach eBay, ktoré budú periodicky obnované, a možnosť archivovať obrázky nájdených položiek. Pre implementáciu je potrebné zvoliť vhodný backendové technológie, ktoré umožnia vytvorenie takého typu aplikácie. Takisto je potrebné vybrať vhodné technológie na vytvorenie prívetivého a intuitívneho užívateľského rozhrania. Okrem týchto technológií som následne popísal ďalšie nástroje, ktoré sú pre efektívnu implementáciu nutnosťou. Nakoniec táto kapitola obsahuje popis prostredia, ktoré som zvolil pre nasadenie vytvorenej aplikácie.

4.1 Backend

Backend predstavuje aplikačnú časť aplikácie. Nachádzajú sa tam triedy a metódy, ktoré sú potrebné pre správne fungovanie celého systému. Všetky požiadavky, ktoré od užívateľa aplikácia dostane sú následne spracované a vyhodnotené. Dané spracovanie predstavuje rôzne výpočty, algoritmy a komunikáciu s databázou. Toto všetko podlieha pod backendovú časť.

4.1.1 Java

Pre vývoj softvérovej aplikácie je veľmi dôležitý výber programovacieho jazyka, ktorý bude použitý pri jej implementácii. Každý programovací jazyk má určité výhody a nevýhody. V závislosti od toho, čo potrebujeme a čo máme k dispozícii, by sme mali zvážiť aký implementačný jazyk použiť. Ja som sa v rámci tejto práce rozhodol použiť Javu.

Java [7] je objektovo orientovaný jazyk, ktorý je určený na vývoj multiplatformových aplikácií. Vznikla v roku 1995 a je vyvíjaná aj v súčasnosti

spoločnosťou Oracle. Pre beh aplikácie je potrebná prítomnosť Java Virtual Machine, ktorá spracuje Java kód.

Jedným z hlavných dôvodov prečo som sa rozhodol použiť tento jazyk je jeho popularita a s ňou súvisjúca veľká komunita, čo môže v prípade problémov urýchliť ich riešenie, a taktiež obsahuje veľkú podporu z hľadiska rôznych pomocných knižníc pre určité technológie.

4.1.2 Spring Boot

Spring framework [8] je jedným z najrozšírenejších aplikačných frameworkov pre programovací jazyk Java. Je využívaný hlavne na vývoj webových aplikácií. Jedným zo základných prvkov tohto frameworku je Dependency injection (Vkladanie závislostí). Je to technika vkladania závislostí medzi komponenty, ktoré v dobe zostavovania programu na seba nemajú referenciu. Spring poskytuje podporné služby pre rôzne časti softvéru, ako je práca s dátami, bezpečnosť a mnoho ďalších.

Pri využívaní Spring frameworku je potrebné množstvo využívaných vecí nakonfigurovať. Daná konfigurácia sa zapisuje do konfiguračných súborov pomocou značkovacieho jazyka Extensible Markup Language (XML).

Spring Boot je jednou z poskytovaných služieb, jedná sa o nadstavbu základného Spring projektu, ktorá odstraňuje potrebu množstva konfigurácií, ktoré sú potrebné pre správne fungovanie aplikácie. Túto potrebu nahrádza preddefinovanými štartovacími závislosťami, ktoré zjednodušujú tvorbu aplikácie, a snahou automatickej konfigurácie všade, kde je to možné. Ďalej obsahuje vstavaný aplikačný server, takže nie je potrebné inštalovať a konfigurovať samostatný aplikačný server.

4.1.3 Hibernate

Pre vytvorenie databázy a následnú prácu s dátami, ktoré tam sú uložené, je vhodné použiť objektovo-relačné mapovanie. Objektovo-relačné mapovanie slúži na mapovanie jednotlivých tried aplikácie na tabuľky databázy. Pre programovací jazyk Java na to slúži Java Persistence API. Je to špecifikácia, ktorá definuje správu dát pre Java aplikácie. Hibernate [9] je jednou z najrozšírenejších implementácií Java Persistence API (JPA), ktorá umožňuje jednoduchú prácu s databázami od rôznych poskytovateľov.

4.1.4 PostgreSQL

PostgreSQL [10] je open source objektovo-relačný databázový systém. Daný systém som si zvolil kvôli jej spoľahlivosti a podpore v rámci Java aplikácií.

4.1.5 Twilio

Twilio [11] je technológia, ktorá poskytuje komunikačné služby hlavne v rámci mobilných sietí veľkého počtu krajín. Poskytuje možnosť skúšobného účtu s určitým kreditom, za ktorý je možné využívať jej služby. Z týchto dôvodov som sa rozhodol využiť túto technológiu na zasielanie SMS oznámení užívateľom.

4.1.6 eBay API + Java

Na využívanie eBay API je potrebné mať spôsob, ktorým je možné poslať požiadavky pre dané API a následne prijať a spracovať odpovede. V nasledujúcich častiach zhrniem možnosti, ktorými sa to dá realizovať a taktiež možnosť, ktorú som zvolil.

Web service definition language (WSDL)

WSDL je jazyk, ktorý slúži na popísanie funkcií webových služieb a taktiež prístupu k nim pomocou jazyka XML. K eBay API je stále možné získať najnovšiu verziu WSDL dokumentu.

V Jave je možné pomocou pluginov z WSDL vygenerovať triedy, ktoré odpovedajú objektom z daného dokumentu a následne je možné odpoveď volania API služieb konvertovať do týchto objektov. Avšak pri samotnom použití tohto generovania je často nutné implementovať ďalšie podporné funkcie, aby všetko správne fungovalo.

eBay software development kit(SDK)

Pre využívanie služieb Trading API poskytuje eBay nástroj eBay SDK [2]. Obsahuje rôzne triedy s podpornými funkciami, ktoré by pri samostatnom využití WSDL museli byť implementované.

Finding Kit

Finding kit [2] je pomocná knižnica, ktorá bola vytvorená pre uľahčenie používania Finding API. Podporuje všetky funkcie, ktoré je možné využívať prostredníctvom daného API.

Zhrnutie

V rámci tejto práce som sa rozhodol, že na využívanie služieb eBay API využijem eBay SDK a Finding Kit, pretože ich použitie je jednoduchšie oproti použitiu WSDL a zjednoduší to prácu aj ušetrí čas pri implementácií.

4.1.7 Google Drive

V rámci tejto práce som sa rozhodol použiť pre archivovanie obrázkov technológiu Google Drive [12]. Je to služba od spoločnosti Google poskytujúca úložisko dát. Je naň možné ukladať súbory rôznych formátov, ako sú napríklad videá, fotografie a iné. Pre jednotlivé súbory je taktiež možné editovať privilégia užívateľov, ktorí k nim majú prístup. Táto služba poskytuje 15 GB úložného priestoru zadarmo pre každý registrovaný účet. V prípade potreby je možné prejsť na platenú službu Google One, ktorá ponúka vyššie kapacity.

Google Drive API

Google pre vývojárov poskytuje Drive API [13], ktoré ponúka externým aplikáciám možnosť práce s Google Drive úložiskom. Pomocou daného API je možné využívať všetky funkcie, ktoré Google Drive poskytuje. Určité úložisko sa ale viaže na užívateľský účet, teda aplikácia, ktorá by ho chcela využívať potrebuje mať povolenie od užívateľa, ktorý bude musieť potvrdiť povolenie prístupu pre danú aplikáciu. Následne aplikácia obdrží prístupový token, ktorý je potrebný pri každom volaní API funkcií.

Servisný účet

Keďže pre využívanie služieb Google Drive API je potrebné mať užívateľský prístupový token, bolo by nutné po každom vypršaní platnosti daného tokenu získať nový token, čo je v prípade tejto práce, v ktorej chceme všetky obrázky ukladať len na jediné spoločné úložisko, nežiadúce. Pre vyriešenie tohto problému som použil tzv. servisný účet. Daný účet umožňuje využívanie Google Drive API bez externej verifikácie. „Aplikácia používa servisný účet na volanie služieb Google API tak, aby užívatelia neboli priamo zapojení“ [14]. Pre vytvorenie servisného účtu je potrebné:

1. Účet na <https://accounts.google.com/>.
2. Vytvoriť projekt na <https://console.cloud.google.com>.
3. V záložke APIs & Services vytvoriť pre daný projekt Credentials pre Service account key, teda kľúč pre servisný účet.
4. Zvoliť typ kľúča a projektu, pre ktorý je daný účet vytváraný.
5. Stiahnuť vytvorený servisný kľúč v zvolenom formáte.

Google API knižnica pre Javu

Na využívanie rôznych API služieb, ktoré poskytuje Google, bola vytvorená pomocná knižnica pre programovací jazyk Java. Pomocou nej je práca s API jednoduchá.

4.2 Frontend

Frontend predstavuje časť aplikácie, s ktorou pracuje koncový užívateľ. V tejto časti popisujem hlavné technológie použité na tvorbu užívateľského rozhrania.

4.2.1 JQuery

JQuery [15] je JavaScriptová knižnica, ktorá uľahčuje prácu a programovanie pomocou JavaScriptu. Manipulácia s prvkami HyperText Markup Language (HTML) dokumentu a ich vyhľadávanie sa vďaka tejto knižnici implementuje veľmi jednoducho.

4.2.2 Bootstrap

Bootstrap [16] je framework, ktorý slúži na vývoj responzívnych webových stránok. Obsahuje rôzne komponenty ako tabuľky, navigačné panely, tlačidlá a mnoho ďalších vecí, ktoré uľahčujú tvorbu HTML stránok. V rámci tohto frameworku som sa rozhodol použiť tému [17], ktorá štylizuje Bootstrap komponenty a tým docieľuje elegantný a jednoduchý vzhľad.

4.2.3 Thymeleaf

Thymeleaf [18] je Java knižnica na vytváranie šablón. Používa sa prevažne na spracovanie a vytvorenie HTML a XML, ale je možné ju taktiež využiť na spracovanie JavaScriptu. Podporuje prácu s kolekciami, reťazcami, číslami a inými štruktúrami.

Jednou z jeho hlavných výhod je integrácia s frameworkom Spring, konkrétne s jeho špecifikáciou Spring MVC. Objekty predané z modelovej vrstvy do prezentačnej vrstvy je následne možné pomocou knižnice Thymeleaf jednoducho využívať.

4.3 Dalšie nástroje

V tejto časti sa zaoberám ďalšími nástrojmi a ich jednotlivými funkcionalitami, ktoré som sa rozhodol využiť a ktoré sú potrebné pre efektívnu implementáciu. Jednotlivé typy technológií sú v dnešnej dobe nutnosťou pri vývoji aplikácií väčších rozmerov.

4.3.1 Apache Maven

Apache Maven [19] je nástroj pre správu, riadenie a automatizáciu buildu aplikácií napísaných prevažne v programovacom jazyku Java. Maven vznikol v roku 2002 a je aj naďalej vyvíjaný firmou Apache Software Foundation. Podľa prieskumu [20] využíva tento nástroj viac ako 75 % dotazovaných Java vývojárov.

Základné ciele nástroja Maven podľa [19] sú:

- Uľahčenie build procesu
- Poskytnutie jednotného build systému
- Poskytnutie kvalitných informácií o projekte
- Navádzanie na vývoj pomocou osvedčených praktík (best practices)
- Umožňuje transparentný prechod na nové funkcionality

Maven projekty obsahujú súbor `pom.xml`, čo je skratka pre Project Object Model. Tento súbor slúži na popis projektu ako XML objektu, kde môžu byť definované závislosti daného projektu na iné knižnice a projekty, ktoré následne maven vyhledá a stiahne z globálneho, poprípade lokálneho repozitára. Globálny repozitár je verejne prístupné úložisko Maven projektov, ktoré je možné použiť ako závislosti v nových projektoch. Nie vždy sa tam ale nachádzajú všetky potrebné knižnice, preto je ich niekedy potrebné nainštalovať do lokálneho repozitára. Každý Maven projekt má unikátne atributy `ArtifactId` a `GroupId`, ktoré ho charakterizujú.

4.3.2 Git

Verzovací systém je dôležitou súčasťou implementácie. Umožňuje ukladať verzie softvéru počas jeho vývoja a následne ich porovnávať. Jedným z hlavných prínosov daného systému je možnosť jednoduchej paralelnej práce tímu na jednom projekte. Jednotlivé zmeny si medzi sebou vývojári vedia synchronizovať.

Git [21] je distribuovaný verzovací systém, čo znamená, že neexistuje jedno centrálné úložisko, ale jednotliví vývojári majú na svojich zariadeniach pracovné kópie, ktoré fungujú ako vzdialené záložné kópie kódu.

Rozhodol som sa použiť Git, lebo s ním mám dobré skúsenosti a budem ho v rámci tohto projektu využívať na zálohovanie a verzovanie aplikácie.

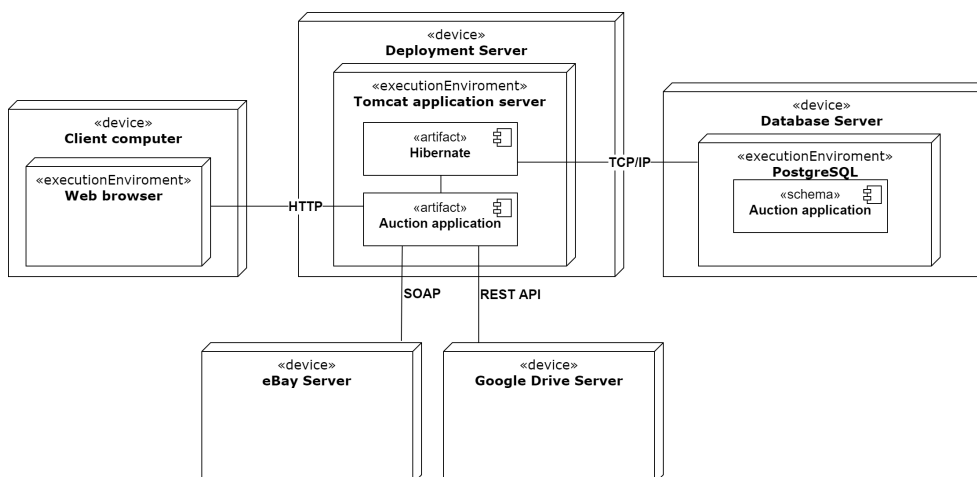
4.3.3 Spring Tools Suite

Eclipse [22] je vývojové prostredie, ktoré poskytuje základné pracovisko na implementáciu softvéru a taktiež rozsiahly systém pluginov, pomocou ktorých môžeme doceliť rozširujúcu funkcionality. Hlavným programovacím jazykom, pre ktorý je toto prostredie určené, je Java. Pomocou pluginov je však možné ho použiť aj na programovanie pomocou iných jazykov. Podľa [20] je Eclipse druhým najpoužívanejším vývojovým prostredím pre vývoj Java aplikácií a používa ho takmer 40 % dotazovaných vývojárov. Jeho výhodou oproti IntelliJ je to, že je zadarmo, takže je možné povedať, že je najpoužívanejším voľným vývojovým prostredím pre Javu.

Spring tool suite [23] je nadstavba Eclipse, ktorá podporuje tvorbu a inicializáciu Spring aplikácií. Bez tejto nadstavby je potrebné vytvoriť Spring projekt manuálne a pridávať všetky závislosti a konfigurácie, alebo je možné vytvoriť ho prostredníctvom inicializátora², pomocou ktorého je možné si zvoliť potrebné technológie a následne vygenerovať projekt. V prípade Spring Tools Suite je všetko možné vytvoriť prostredníctvom vývojového prostredia.

4.4 Nasadenie

Pre trvalý chod aplikácie je potrebné zabezpečiť prostredie, na ktorom môže aplikácia aj databázový server bežať. Pre testovacie účely som zvolil použitie platformy Google Cloud [14]. Poskytuje škálovateľné výpočtové prostriedky pre aplikačné a dátové potreby. Táto platforma poskytuje pre nový účet skúšobný kredit, ktorý je možné míňať na využívanie poskytovaných služieb. Z týchto služieb je pre túto aplikáciu potrebná služba App Engine, na ktorú je možné nasadiť aplikačný server a služba SQL, prostredníctvom ktorej je možné vytvoriť a prevádzkovať databázu, ktorú bude aplikácia využívať. Diagram nasadenia aplikácie je možné vidieť na obrázku 4.1.



Obr. 4.1: Diagram nasadenia

²vid <https://start.spring.io/>

Realizácia

V tejto kapitole popíšem implementáciu aplikácie počnúc štruktúrou projektu až k popisu jednotlivých častí. Stručne popíšem tvorbu dátovej a prezentačnej vrstvy a následne jednotlivé dôležité časti aplikácie. Na konci som zhrnul priebeh nasadzovania aplikácie na server.

5.1 Štruktúra aplikácie

V tejto časti popíšem členenie zdrojových častí aplikácie. Na obrázku 5.1 je znázornená štruktúra aplikácie — balíčky, zdroje a konfiguračné súbory. V zdrojovom adresári `src` sa nachádza adresár `main`, ktorý obsahuje všetky zdrojové kódy a konfiguráciu aplikácie, a adresár `test`, ktorý obsahuje zdrojové kódy a konfiguráciu potrebnú pre testovacie účely. Adresár `main` obsahuje 3 adresáre. Prvým adresárom je `appengine`. Obsahuje len jeden súbor, ktorý definuje zdroje, ktoré má Google Cloud pri nasadení aplikácie použiť. Ďalšími adresármi sú `java` a `resources`, ktoré sú popísané v nasledujúcich častiach.

5.1.1 Adresár java

V tomto adresári sa nachádzajú balíčky, ktoré obsahujú triedy potrebné pre správne fungovanie aplikácie. Každý balíček predstavuje v rámci aplikácie inú funkčnosť. Balíčky majú predponu `tomaspolacok.bachelor.application` a ich funkcie sú:

config Slúži na konfiguráciu bezpečnosti a vytváranie tried, ktoré umožňujú jednoduché využívanie služieb eBayu a Google Drive.

controllers Predstavuje kontrolnú časť MVC architektúry, takže slúži na prijímanie užívateľských požiadaviek a kontrolu zaslaných formularov, odovzdávanie objektov do prezentačnej vrstvy a presmerovania.

entities Obsahuje triedy, ktoré predstavujú jednotlivé tabuľky v databáze.

5. REALIZÁCIA



Obr. 5.1: Štruktúra aplikácie

enums Obsahuje enumerované typy, ktoré slúžia na zlepšenie čitateľnosti zdrojového kódu.

exceptions Obsahuje vlastné implementované výnimky.

helper Triedy ktoré majú pomocnú funkcionality.

main Obsahuje hlavnú triedu, ktorá zabezpečuje spustenie aplikácie.

repositories Pre jednotlivé modelové triedy obsahuje repozitáre, ktoré slúžia na komunikáciu s databázou.

services Predstavuje hlavnú logiku aplikácie, teda spracovanie užívateľských požiadaviek, komunikáciu s databázou a komunikáciu s rôznymi API.

5.1.2 Adresár resources

Tento adresár obsahuje 2 adresáre a 5 súborov. Jednotlivé adresáre sú:

static Obsahuje statické zdroje, ktoré sú potrebné pre správne fungovanie prezentačnej vrstvy. Je tvorený adresárom cs, ktorý obsahuje súbory

určené na štylizáciu HTML elementov, adresárom `img`, ktorý obsahuje statické obrázky, a posledným adresárom `js`, v ktorom sú uložené JavaScriptové súbory pre správnu funkcionálnosť niektorých komponentov v prezentačnej vrstve.

templates Obsahuje súbory HTML, ktoré predstavujú prezentačnú vrstvu. Sú zložené z HTML elementov a taktiež obsahujú nejaké JavaScriptové funkcie. Ďalej tam je použitá knižnica Thymeleaf (viď sekcia 4.2.3), ktorá následne z týchto súborov generuje HTML stránky.

Okrem toho obsahuje adresár `fragments`, v ktorom je jeden HTML súbor obsahujúci opakujúce sa časti kódu, ktoré sú následne pomocou knižnice Thymeleaf používané v ďalších HTML súboroch.

Jednotlivé súbory a ich významy sú:

application.properties Obsahuje konfiguráciu Spring aplikácie. Je možné definovať nové konštanty, ktoré je možné načítať do premenných v konkrétnych triedach.

import.sql Obsahuje Structured Query Language (SQL) dotazy na vloženie dát pri prvotnom spustení aplikácie.

messages.properties Obsahuje jednotlivé texty, ktoré sú používané v rôznych šablónach.

service_account.json JavaScript Object Notation (JSON) súbor, ktorý obsahuje verifikačné údaje, ktoré sú potrebné pre používanie služieb Google Drive.

spring-cloud-bootstrap.properties Meno Google Cloud projektu, v rámci ktorého má byť daná aplikácia nasadená.

5.2 Dátová vrstva

Dátovú vrstvu som implementoval použitím technológií zvolených v 4.1. Jednotlivé dátové triedy, na základe ktorých bolo vykonané objektovo-relačné mapovanie, som vytvoril podľa databázového modelu popísaného v 3.3 a nachádzajú sa v balíčku `entities`. Pre dátovej triedy som použil anotáciu `@Data` knižnice Lombok [24], ktorá automaticky vygeneruje metódy na zmenu a čítanie jednotlivých polí triedy a ďalšie pomocné metódy na porovnanie a prevod do reťazca. Príklad dátovej triedy sa nachádza na časti kódu 5.1.

Komunikácia medzi aplikáciou a databázou je zabezpečená pomocou tried v balíčku `repositories`. Daný balíček obsahuje pre jednotlivé dátové triedy repozitáre, ktoré sú implementáciou rozhrania `JpaRepository`. Dané rozhranie

umožňuje základné CRUD (Create, Read, Update, Delete) operácie nad databázou s daným dátovým typom, ale okrem toho obsahuje aj ďalšie pokročilejšie funkcie na triedenie a stránkovanie. Taktiež podporuje možnosť definovania vlastných SQL dotazov pomocou natívneho SQL ale aj pomocou Java Persistence Query Language (JPQL), čo je špeciálny dotazovací jazyk, ktorý nepracuje s relačnými tabuľkami, ale s dátovými triedami aplikácie. Definícia repozitára je znázornená na časti kódu 5.2.

```
@Data
@Entity
public class Country {
    @Id
    @Column(name = "code")
    private String code;

    @Column(name = "name")
    private String name;
}
```

Časť kódu 5.1: Dátová trieda

Použitie PostgreSQL a Hibernate ako implementácie JPA je možné jednoducho zameniť za iné pomocou zmeny konfigurácie a pridania príslušných závislostí.

```
public interface CountryRepository extends JpaRepository<Country,
    String> {}
```

Časť kódu 5.2: Repozitár

5.3 Prezentačná vrstva

Prezentačnú vrstvu som implementoval použitím technológií zvolených v časti 4.2 podľa návrhov z časti 3.5. Knižnica Thymeleaf umožňuje definovať časti jednotlivých šablón ako fragmenty, ktoré je možné potom obsiahnuť v ďalších šablónach. Takýmto spôsobom som vytvoril a používal navigačné menu (viď časť kódu 5.3) a jednotlivé zdroje na externé JavaScriptové a Cascading Style Sheets (CSS) knižnice. Pre mnoho komponentov (klikateľné ikony, textové polia, ...) som vytvoril nápovedy, ktoré si užívateľ vie zobrazit ako tooltipy.

Pri implementácii tejto vrstvy som použil niekoľko doplnkových knižníc, ktoré sú založené na Bootstrap, jQuery a jazyku JavaScript:

bootstrap-select [25] Využívané ako kombinované polia pre výber položiek v užívateľskom rozhraní.

bootstrap-datepicker [26] Využívané pre výber dátumu.

bootstrap-tagsinput [27] Využívané pre písanie kľúčových slov.

chart.js [28] Využívané na tvorbu grafov pri zobrazení detailu položky.

unitegallery [29] Využívané pre zobrazenie obrázkov pri zobrazení detailu položky.

fontawesome [30] Obsahuje ikony pre rôzne akcie v rámci užívateľského rozhrania.

```
<div th:fragment="nav(activeTab)">
.
.
.
<li class="nav-item" th:classappend="{activeTab == 'search'}?
    active : '' ">
    <a class="nav-link" th:href="@{/search}">Search</a>
</li>
.
.
</div>

<th:block th:include="fragments/includes :: nav('search')"></th:block>
```

Časť kódu 5.3: Vytvorenie fragmentu

5.4 Validácia dát

Pri komunikácií medzi prezentačnou a aplikačnou vrstvou dochádza k zasielaniu žiadostí klientov s rôznymi dátami, ktoré je potrebné po prijatí žiadosti validovať.

Validácia dát prijatých formulárov prebieha v kontrolnej vrstve pomocou triedy `BindingResult`, ktorá skontroluje prijatý objekt na základe anotácií pri jednotlivých položkách triedy daného objektu. V prípade chyby presmerujem užívateľa na stránku s vyplneným formulárom, kde budú zobrazené chybové správy viazané na jednotlivé položky kontrolovaného objektu.

Validácia ďalších dát, ktoré nie sú zasielané prostredníctvom formulárov, prebieha pomocou servisnej vrstvy. Ide hlavne o dáta, ktoré charakterizujú objekty, ktoré sa viažu na určitého užívateľa, teda je potrebné overiť, či užívateľ, ktorý sa snaží prístup k danému objektu, má k tomu dostatočné práva. V tomto prípade užívateľa presmerujem na chybovú stránku, kde sa mu zobrazí príslušná chybová správa.

5.5 Vyhľadávanie

V tejto časti sa budem venovať implementácií vyhľadávania. Najprv zhrniem vytváranie vyhľadávania a následné vytvorenie a zaslanie žiadosti na eBay API. Pokračovať budem spracovaním odpovede a implementáciou notifikácií. V závere popíšem zmeny kategórií pre jednotlivé lokalizácie eBayu.

5.5.1 Vytvorenie vyhľadávania

Jednotlivé požiadavky, ktoré sa týkajú vytvárania vyhľadávania spracováva kontrolna trieda `SearchController`. Žiadosť o vytvorenie nového vyhľadávania je prijatá metódou `searchCreateGet`. Daná metóda predvyplní určité údaje pre nové vyhľadávanie na základe užívateľských preferencií a presmeruje užívateľa na stránku s formulárom pre dané vyhľadávania. Následne, keď užívateľ vyplní a odošle formulár, metóda `searchCreatePost` prijme danú požiadavku a prebehne validácia vytvoreného vyhľadávania bližšie popísaná v časti 5.4.

5.5.2 Žiadosť na Finding API

V prípade úspešnej validácie prebieha ďalšie spracovanie vytvoreného vyhľadávania v servisnej triede `SearchService`. Metóda `searchFindItems` (viď časť kódu 5.4) využíva pomocnú triedu `SearchRequestMaker` pre vytvorenie objektu triedy `FindItemsAdvancedRequest` z knižnice Finding Kit (viď sekcia 4.1.6) a nastavenie jednotlivých kritérií podľa spracovávaného vyhľadávania.

```
public void searchFindItems(Search search, Boolean automatic) throws
    Exception {

    FindItemsAdvancedRequest request =
        SearchRequestMaker.createRequest(search);
    List<FindItemsAdvancedResponse> responses = new ArrayList<>();

    fixedPriceListings(request, responses, search, automatic);
    auctionListings(request, responses, search, automatic);

    itemService.processEbayResponse(responses, search, automatic);
}
```

Časť kódu 5.4: Metóda pre vyhľadávanie

Keďže má aplikácia podporovať hľadanie aukčných, ale aj normálne predávaných objektov, je nutné nastaviť parameter určujúci typ predaja. Pri tomto nastávaní určité problémy, pretože pre vyhľadávanie aukčných položiek je možné použiť kritéria, ktoré sú pre neaukčné položky neplatné. V prípade použitia daných kritérií v kombinácii s neaukčným typom predaja Finding API nevráti žiadne výsledky. Práve preto som to implementoval spôsobom, že v prípade

zvolenia oboch typov predaja sa najprv vytvorí a odošle žiadosť pre neaukčné položky a následne po nej žiadosť pre aukčné položky, kde použijem doplnujúce kritéria, ktoré sú validné pre daný typ predaja.

Užívateľ môže pri vytváraní vyhľadávania zadať výraz kľúčových slov, pri ktorých môže používať logické výrazy AND, OR, NOT a zátvorky. Následne je daný výraz spracovaný pomocou knižnice JBool Expressions [31], ktorá spracuje jednotlivé skupiny len na spojenia jednotlivých slov. Samotné Finding API poskytuje možnosť práce s výrazmi, kde sa vyskytujú logické operátory, avšak toto riešenie nie je možné použiť na zložitejší výraz. Preto najprv výraz spracujem na skupiny jednoduchších výrazov, ktoré viem následne poslať na Finding API.

Ďalšia vec, ktorú je potrebné nastaviť je poradové číslo stránky a počet položiek, ktoré má odpoveď vrátiť. Maximálny možný počet je 100 položiek na jednu stránku, preto som sa v rámci jednej odpovede rozhodol použiť toto maximum. V prípade, že nájdených výsledkov bude viac, aplikácia vygeneruje výnimku `TooManyItemsException`, prostredníctvom ktorej to bude užívateľovi oznámené. Dôvodom tohto obmedzenia je snaha zamedziť nadmernému počtu volaní Finding API pri zadaní veľmi generických kritérií.

Odpovede sú objekty triedy `FindItemsAdvancedResponse`, ktoré sú po vytvorení ukladané do poľa, ktoré je následne spracované.

5.5.3 Spracovanie odpovede

Spracovanie výsledkov vyhľadávania prebieha v servisnej triede `ItemService`, kde je z výsledkov volaní Finding API vytvorený zoznam objektov, ktoré predstavujú predávané položky. Následne sú zo zoznamu vymazané položky, ktoré má užívateľ zaradené medzi zakázanými podľa ich identifikátorov aj podľa zhody názvu. Nakoniec je pole položiek spolu so spracovávaným vyhľadávaním uložené do databázy.

5.5.4 Notifikácie

Po spracovaní výsledkov z predchádzajúcej časti sú z daného poľa výsledných položiek vymazané objekty, ktoré už boli oznámené užívateľovi. Medzi tieto objekty radím:

- Položky, ktoré boli nájdené v rámci vytvorenia vyhľadávania.
- Položky, ktoré boli nájdené v rámci automatického vykonania vyhľadávania a poslané užívateľovi emailom.

Notifikácie sú posielané pomocou servisnej triedy `NotificationService`. V rámci tejto aplikácie som vytvoril 2 typy notifikácií:

Emailová notifikácia Hlavička správy obsahuje meno vyhľadávania a počet nových položiek. Telo správy predstavuje HTML dokument vytvorený pomocou knižnice Thymeleaf a obsahuje obrázky položiek spolu so základnými informáciami, ktoré ich popisujú.

SMS notifikácia Telo poslanej SMS tvorí len názov vyhľadávania, ktoré si užívateľ vytvoril a počet objektov, ktoré spĺňajú notifikačné podmienky pre notifikácie pomocou SMS.

5.5.5 Aktualizácia vyhľadávania

Pomocou Spring anotácie `@Scheduled` metóda `refresh` v servisnej triede `SearchService` je vykonávaná každú minútu od skončenia vykonávania predchádzajúceho priebehu metódy. Metóda vyberie z databázy všetky vyhľadávania a pre každé porovnáva aktuálny čas s časom ich posledného obnovenia. Ak je daný rozdiel väčší než perióda obnovovania daného vyhľadávania, je zavolaná metóda `searchFindItems` popísaná v sekcii 5.5.2.

5.5.6 Kategórie položiek

V časti 2.2.2 som písal o hierarchii kategórií. Táto hierarchia aktualizovaná pomocou triedy `CategoryService`. Metóda `checkCategoryVersion` porovná verziu kategórií pre danú lokalizáciu eBayu uloženú v databáze s aktuálnou verziou pomocou volania Trading API funkcie `GetCategories`. V prípade rozdielu verzií je volaná metóda `GetCategoryMappings`, na základe výsledkov ktorej bude vykonaná aktualizácia hierarchie uloženej v databáze. V tejto aplikácii využívam len najvyššiu úroveň kategórií pri jednotlivých lokalizáciach, pretože predpokladám, že poskytuje dostatočnú možnosť špecifikácie so súčasným použitím kľúčových slov.

Na periodické aktualizovanie kategórií slúži metóda `refresh` z časti 5.5.5. Pre jednotlivé lokalizácie eBayu je uložený čas poslednej aktualizácie a daná metóda zistí pre ktoré lokalizácie ubehlo viac než 24 hodín, a tie následne aktualizuje. Aktualizácia prebehne tiež v prípade, že dojde k problému súvisiacemu s kategóriami pri vyhľadávaní (viď sekcia 5.5.2).

```
@Bean
public ApiContext getApiContext() {
    ApiContext apiContext = new ApiContext();
    ApiCredential cred = apiContext.getApiCredential();
    cred.seteBayToken(token);
    apiContext.setApiServerUrl(url);
    apiContext.setSite(SiteIDUtil.fromNumericalID(siteCode));
    return apiContext;
}
```

Časť kódu 5.5: Vytvorenie kontextového objektu pre Trading API

Pre využívanie Java SDK pre Trading API je potrebné vytvoriť kontext (viď časť kódu 5.5) s potrebnými údajmi (token, aplikačný kľúč, id eBay stránky), ktorý je následne možné používať na zasielanie požiadaviek.

5.6 Peňažná mena

Aplikácia má podporovať zobrazovanie výsledkov vyhľadávania v rôznych peňažných menách. Do databázy ukladám všetky cenové hodnoty v peňažnej mene americký dolár (USD). To zjednoduší ukladanie v prípade, že viacerí užívatelia budú mať prístup k jednej položke a budú chcieť vidieť jej cenu v rôznych menách. Vtedy budú peňažné vlastnosti daného objektu prekonvertované na žiadanú menu a zobrazené užívateľovi.

Jednotlivé funkcionality súvisiace s peňažnými menami sú implementované v servisnej triede `CurrencyService`. Najdôležitejšie funkcie danej triedy sú popísané v nasledujúcich častiach.

5.6.1 Získavanie prevodných kurzov

Na získavanie jednotlivých kurzov využívam API³, ktoré pravidelne aktualizuje kurzy. Keďže aplikácia ukladá jednotlivé položky v mene USD, je potrebné načítať práve kurzy medzi USD a inými podporovanými menami. Aktualizáciu prevodných kurzov vykonáva metóda `updateCurrencyRates`, ktorá je pomocou anotácie `@Scheduled` volaná raz za 24 hodín.

API poskytuje odpoveď vo formáte JSON. Pre jednotlivé meny podporované aplikáciou sú z odpovede načítané prevodové kurzy voči USD a následne uložené do databázy.

5.6.2 Konverzia

Pri každom zobrazovaní objektov a pri zasielaní notifikácií užívateľovi je potrebné konvertovať všetky objekty na užívateľom definovanú menu. Na tento účel slúži metóda `convertItemWithRate`. Parametrom metódy je objekt na konverziu, požadovaná mena a kurz k danej mene.

5.7 Archivácia

V tejto časti popíšem priebeh archivácie vybraných obrázkov. Najprv opíšem načítanie detailu vybranej položky, následne priebeh archivácie a zobrazenie obrázkov pre užívateľa.

Pre archivovanie na úložisko Google Drive je tam najprv potrebné vytvoriť zložku, do ktorej bude mať servisný účet právo zapisovať a jej identifikátor je následne používaný pri ukladaní jednotlivých obrázkov.

³<https://api.exchangeratesapi.io/latest?base=USD>

5.7.1 Načítanie detailu položky

Keď užívateľ pošle žiadosť na zobrazenie detailu nejakej položky sú z Trading API stiahnuté ďalšie detaily pomocou funkcie `GetItem`. Medzi tieto detaily patria aj všetky obrázky, ktoré predajca zverejnil a sú dostupné na eBay stránke popisujúcej predaj danej položky. Po tomto zobrazení je daná položka aktualizovaná a stránka, ktorá ju popisuje je zobrazená užívateľovi.

5.7.2 Uloženie do archívu

Jednotlivé metódy slúžiace na archiváciu sa nachádzajú v servisnej triede `ArchiveService`. Užívateľ si na stránke detailu položky môže otvoriť archivačné modálne okno, kde si môže zvoliť obrázky a kategóriu, ktoré chce archivovať.

```
@PostMapping(value= {"/category/add"})
public @ResponseBody String addCategory(@RequestParam
    ("categoryName") String categoryName) {
    PictureCategory cat = archiveService.saveCategory(categoryName);
    return cat.getId().toString();
}
```

Časť kódu 5.6: Koncový bod pre pridanie novej kategórie

Pri vytváraní novej kategórie je poslaná Asynchronous JavaScript + XML (AJAX) požiadavka na server, po ktorej spracovaní je nová kategória obrázkov uložená pre užívateľa (viď časť kódu 5.6). Následne je poslaná AJAX požiadavka pre uloženie vybraných obrázkov pod zvolenú kategóriu. Pomocou Google Drive API sú dané obrázky nahrané na úložisko.

5.7.3 Cache

Pri načítaní jednotlivých obrázkov klientom je stále potrebné načítavať ich, čo môže byť zbytočne pomalé. Preto pre zobrazovanie obrázkov či už v archíve, alebo pri detaile položky používam anotáciu `@Cacheable` pomocou ktorej server zistí, pre ktoré metódy si má pamätať návratové hodnoty pri rovnakých vstupných parametroch (viď časť kódu 5.7).

```
@ResponseBody
@RequestMapping(value = "/images/{imageId}", method =
    RequestMethod.GET, produces = MediaType.IMAGE_JPEG_VALUE)
@Cacheable("pictureResponse")
public byte[] getImageResource(@PathVariable String imageId)
    throws Exception {
    return archiveService.getPicture(imageId);
}
```

Časť kódu 5.7: Koncový bod pre stiahnutie obrázku

5.8 Nasadenie

Ako som spomínal v časti 4.4, aplikáciu som sa rozhodol nasadiť na Google Cloud. Na to je potrebné nainštalovať si Cloud SDK a následne sa tam prihlásiť. Potom už je nasadenie aplikácie jednoduché. Pomocou Maven pluginu pre integráciu Java aplikácie s Google Cloud systémom je možné zabezpečiť build a nasadenie aplikácie použitím príkazu `mvn package appengine:deploy`.

Daný príkaz vytvorí zo zdrojových súborov Java Archive (JAR) súbor, ktorý následne nasadí ako novú verziu aplikácie na Google Cloud server. Cloud SDK poskytuje v rámci nasadenej aplikácie viaceré možnosti⁴.

Funkčná verzia aplikácie je momentálne dostupná pre testovacie účely na adrese <http://www.auction-management-app.appspot.com>.

5.9 Licencie

Technológie, ktoré som použil v rámci vývoja tejto aplikácie spadajú pod licencie Apache Licence 2.0, BSD, MIT, GNU LGPL a CDDL. Väčšina týchto licencií patrí medzi permissívne licencie, takže umožňujú voľne využívať dané technológie bez väčších obmedzení. Licencia GNU LGPL je menej permissívna, avšak umožňuje používanie dynamicky linkovaného softvéru (patriaceho pod danú licenciu) bez nutnosti vydania softvéru pod rovnakou licenciou, poprípade bez nutnosti sprístupnenia zdrojového kódu.

⁴vid <https://cloud.google.com/sdk/gcloud/reference/>

Výsledky

V tejto kapitole opíšem výsledky tejto práce. Prvá časť obsahuje užívateľské testovanie rôznych scenárov, ktorých jednotlivé výsledky som vyhodnotil a na základe toho som v aplikácii pozmenil niektoré veci, ktoré sa ukázali ako problematické. Druhá časť popisuje výhliadky do budúcnosti, kde zhrniem jednotlivé možnosti rozšírenia a zlepšenia aplikácie, ktoré by mohli okrem poskytnutia nových funkcií taktiež zjednodušiť využívanie aplikácie

6.1 Testovanie

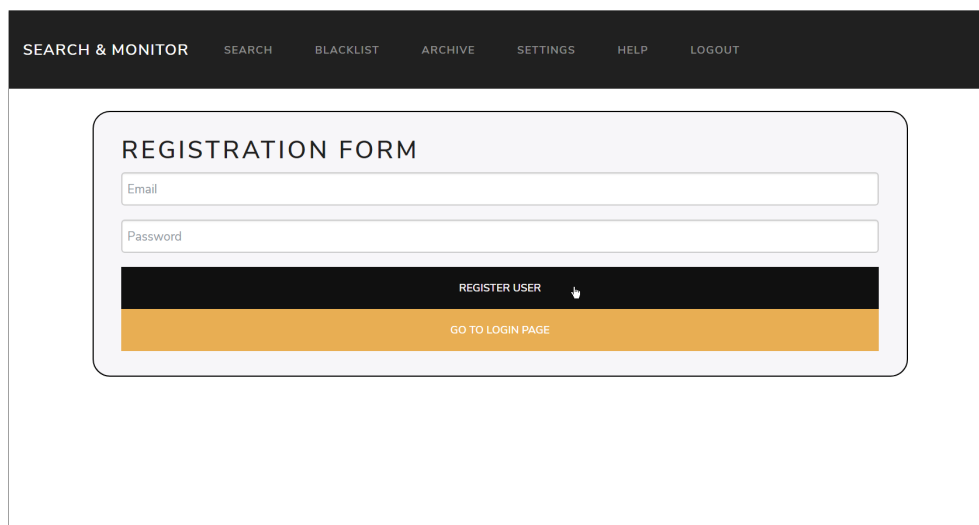
Aplikáciu som užívateľsky otestoval v spolupráci s 5 testovacími užívateľmi, pre ktorých som pripravil testovacie scenáre, ktoré by mali obsiahnuť základnú funkcionálnosť aplikácie. Jeden užívateľ aplikáciu testoval prostredníctvom prehliadača na mobilnom zariadení. Jednotlivé scenáre som následne na základe odozvy zhodnotil.

1. Zaregistrujte sa a prihláste sa.
2. Pridajte telefónne číslo pre svoj účet a aktivujte ho.
3. Zmeňte svoje užívateľské preferencie a heslo.
4. Vytvorte nové vyhľadávanie tak, aby vrátilo nejaké výsledky (príklad: eBay stránka „United States“, kľúčové slová „gtx&1060“, typ predaja „Auction“ a minimálny počet ponúk „15“). Následne povoľte všetky mobilné notifikácie pre vytvorené vyhľadávanie.
5. Prejdite do výsledkov vyhľadávania a vytvorte zákaz predajcu určitej položky pre konkrétne vyhľadávanie a následne daný zákaz vymažte.
6. Prejdite do výsledkov vyhľadávania a zobrazte detaily určitej položky. Archivujte obrázky prislúchajúce k vybranej položke a následne ich vymažte z archívu.

6. VÝSLEDKY

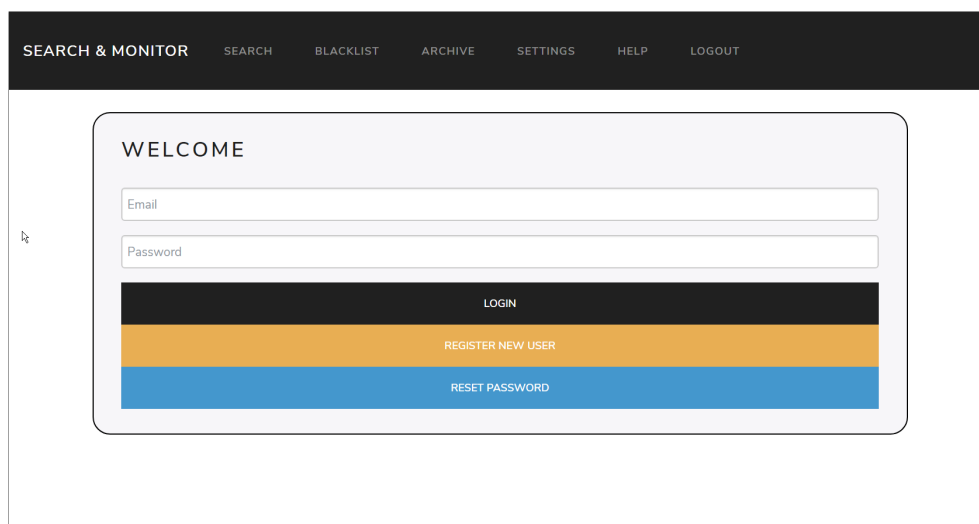
6.1.1 Registrácia a prihlásenie

Cieľom tohto scenára je zistiť či je registrácia a následné prihlásenie (viď obrázky 6.2 a 6.1) pre užívateľa dostatočne jednoduché a funkčné. Táto časť prebehla pri všetkých užívateľov bez problémov — všetkým užívateľom bol po registrácii zaslaný aktivačný email, pomocou ktorého svoj účet aktivovali a následne sa prihlásili do aplikácie.



The screenshot shows the 'REGISTRATION FORM' page. At the top, there is a dark navigation bar with the text 'SEARCH & MONITOR' and several menu items: 'SEARCH', 'BLACKLIST', 'ARCHIVE', 'SETTINGS', 'HELP', and 'LOGOUT'. The main content area features a light gray rounded rectangle containing the form. It has two input fields labeled 'Email' and 'Password'. Below these fields are two buttons: a black one labeled 'REGISTER USER' and an orange one labeled 'GO TO LOGIN PAGE'.

Obr. 6.1: Registrácia



The screenshot shows the 'WELCOME' page. It has the same dark navigation bar as the registration page. The main content area features a light gray rounded rectangle containing the login form. It has two input fields labeled 'Email' and 'Password'. Below these fields are three buttons: a black one labeled 'LOGIN', an orange one labeled 'REGISTER NEW USER', and a blue one labeled 'RESET PASSWORD'.

Obr. 6.2: Prihlasovanie

6.1.2 Telefónne číslo

Cieľom tohto scenára je overiť či je užívateľ schopný pridať a potvrdiť mobilné telefónne číslo pre svoj účet. Pri tomto scénari sa ukázalo, že aplikácia nezaslala aktivačné kódy. Dôvodom sa ukázala nemožnosť zasielania SMS správ neverifikovaným telefónnym číslam prostredníctvom skúšobného účtu technológie popísanej v časti 4.1.5. Po manuálnej verifikácii však prebehla aktivácia bez ďalších problémov.

6.1.3 Užívateľské preferencie a heslo

Cieľom je zistiť, či je užívateľ schopný jednoducho zmeniť svoje prihlasovacie heslo a preferencie pre svoj účet (viď obrázok 6.3) — predvolené nastavenie pre vyhľadávanie, maximálnu dobu nečinnosti pre odhlásenie a peňažnú menu. Zmeny preferencií a hesla prebehli pre všetkých užívateľov bez problémov.

USER PREFERENCES

SEARCH DEFAULTS

EBAY SITE

EBAY GERMANY ▾

LISTING TYPES

AUCTION ▾

SHIPPING COUNTRY

SELECT PREFERRED COUNTRY ▾

REFRESH TIME (MINUTES)

60

New Only

Free Shipping Only

Returns Accepted Only

Enable notifications

OTHER

Stay logged in

INACTIVITY LOGOUT (MINUTE)

60

ACCOUNT CURRENCY

PICK PREFERRED CURRENCY ▾

PHONE NUMBER

ADD PHONE NUMBER

SAVE PREFERENCES

Obr. 6.3: Nastavenie

6.1.4 Vyhľadavanie

Cielom tohto scenára je overiť schopnosť užívateľa vytvoriť vyhľadávanie (viď obrázok 6.4). Teda zvoliť si eBay lokalizáciu a ďalšie kritéria pre nájdenie žiadaných objektov, poprípade pracovať s chybovou správou v prípade zadania príliš generických kritérií. Vyhľadávanie zvládli všetci vytvoriť bez problémov. Jedinou výhradou bolo vymazanie definovaných kritérií pri zmene eBay stránky. Danú vadu som následne opravil, teda po zmene eBay lokalizácie pri tvorbe nového vyhľadávania budú zachované všetky doposiaľ zadané kritéria okrem kategórií, ktoré sa viažu na konkrétnu lokalizáciu.

Ďalšia časť tohto scenára má overiť, či užívateľ vie zmeniť nastavenie mobilných oznámení pre dané vyhľadávanie. Všetci užívatelia boli schopní nájsť tlačidlo pre zmenu mobilných nastavení a následne dané nastavenia zmeniť.

6.1.5 Zákaz

Cielom tohto scenára je zistiť, či užívateľ je schopný vytvoriť zákaz pre určitého predajcu a následne si zobrazit správny zoznam zákazov a daný zákaz vymazať. Vytvoriť zákaz bolo pre užívateľov jednoduché (viď obrázky 6.5, 6.6 a 6.7), trochu komplikácii však prinieslo následovne vymazanie.

Prvým problémom bolo rozlíšenie zoznamu zákazov pre užívateľa a pre jednotlivé vyhľadávanie. Niektorí testujúci užívatelia sa najprv dostali do zoznamu zákazov pre užívateľský účet, keďže to ponúka navigačné menu. Avšak po zistení, že sa vytvorený zákaz nenachádza na zobrazenej stránke, sa boli schopní dostať na správny zoznam.

Na základe tejto odozvy som trochu poupravil aplikáciu. Po úspešnom vytvorení vyhľadávania bude užívateľ presmerovaný na stránku s výsledkami, kde som pridal tlačidlá na jednotlivé akcie s vyhľadávaním, pomocou ktorých sa užívateľ bude môcť jednoduchšie dostať aj do zoznamu zákazov. Ďalej som pridal text popisujúci rozsah určitého zoznamu zákazov, aby bolo jasne špecifikované na akom zozname sa užívateľ nachádza (viď obrázok 6.8).

Ďalšou výhradou bolo, že po vymazaní zákazu neboli položky, ktorých sa zákaz týkal, obnovené v ovplyvnených vyhľadávaniach. Je tomu tak z dôvodu možných zmien kritérií vyhľadávania a vlastností položiek.

V prípade, že by som chcel, aby po zrušení zákazu boli položky naspäť pridané do všetkých vyhľadávanií, z ktorých boli prostredníctvom daného zákazu odstránené, by bolo nutné zistiť, či dané položky splňajú aktuálne kritéria vyhľadávania. K tomu ale aplikácia neuchováva dostatok informácií, keďže Finding API nezískava všetky informácie o položkách na základe ktorých sa dá vyhľadávať (napríklad popis). Ďalším problémom je neaktuálnosť odstránených položiek. Riešením by bolo po zrušení zákazu aktualizovať všetky užívateľové vyhľadávania, a tak by došlo k obnoveniu všetkých vymazaných položiek. Túto možnosť ale považujem za neefektívnu, preto som sa rozhodol ponechať súčasné správanie aplikácie.

SEARCH FORM

EBAY SITE
EBAY GERMANY ▾

NAME
[Text input field]

DESCRIPTION
[Text input field]

SHIPPING COUNTRY
SELECT PREFERRED COUNTRY ▾

LISTING TYPES
AUCTION ▾

CATEGORIES
SELECT <= 3 CATEGORIES ▾

KEYWORDS
[Text input field]
 Search in description

SELLERS
[Text input field]

END DATE FROM [Text input field] **END DATE TO** [Text input field]

MIN BIDS [Text input field with value 0] **MAX BIDS** [Text input field with value 0]

MIN PRICE(EUR) [Text input field with value 0] **MAX PRICE(EUR)** [Text input field with value 0]

REFRESH TIME(MINUTES)
[Text input field with value 60]

OTHER OPTIONS
 New Only
 Free Shipping Only

CREATE

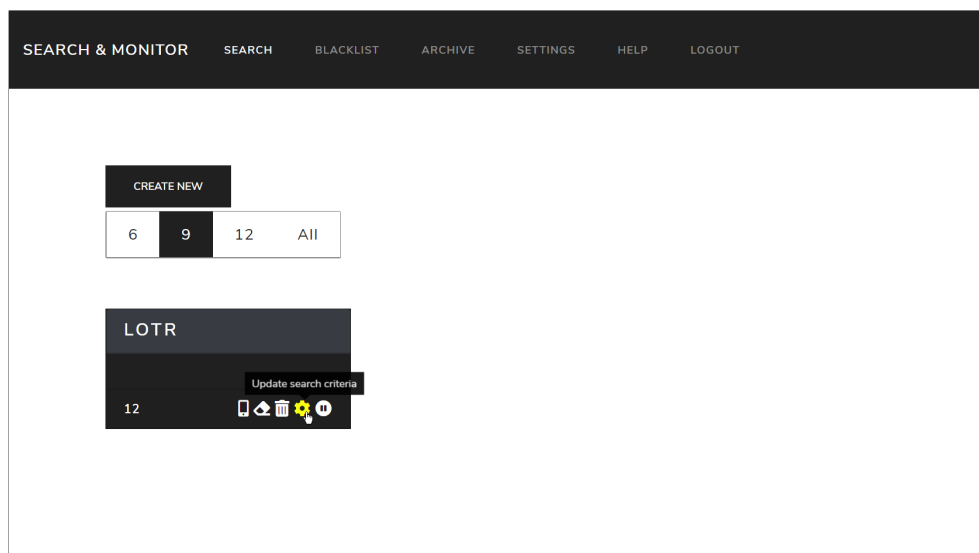
Obr. 6.4: Tvorba vyhľadávania

6.1.6 Detaily položky a archív

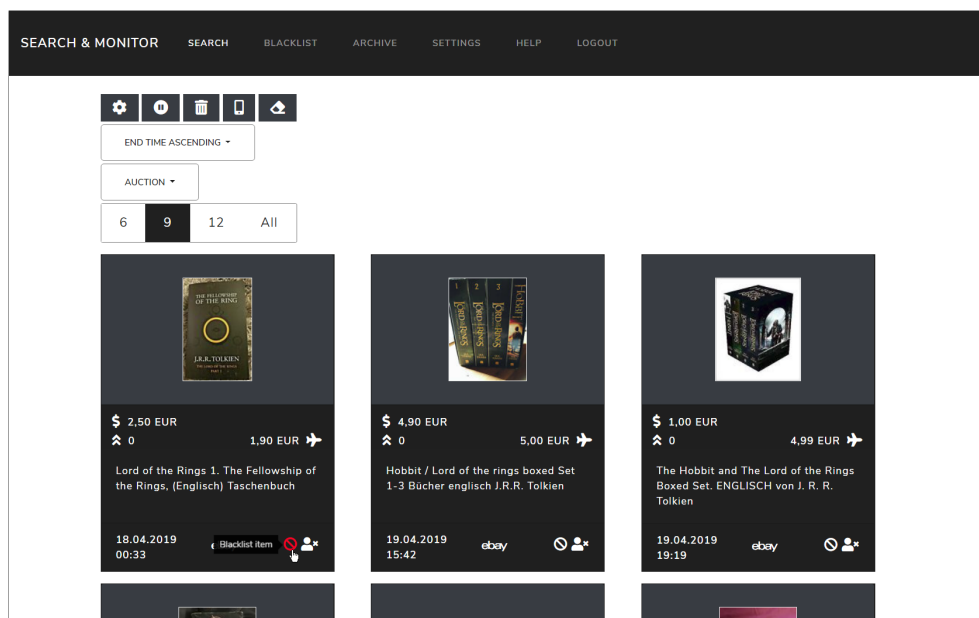
Cieľom je otestovať, či užívateľ dokáže zobraziť detail nájdenej položky vo vyhľadávaní a následne archivovať obrázky patriace danej položke (viď obrázky 6.9 a 6.10). Detaily položky zvládli zobraziť všetci bez problémov, ale jeden užívateľ mal problém s následným nájdením tlačidla pre archivovanie, ktoré sa nachádza pod galériou obrázkov.

Ďalej bolo cieľom tohto scenára zistiť, či si užívateľ vie archivované obrázky zobraziť a následne ich vymazať z archívu (viď obrázok 6.11). Táto časť prebehla bez problémov.

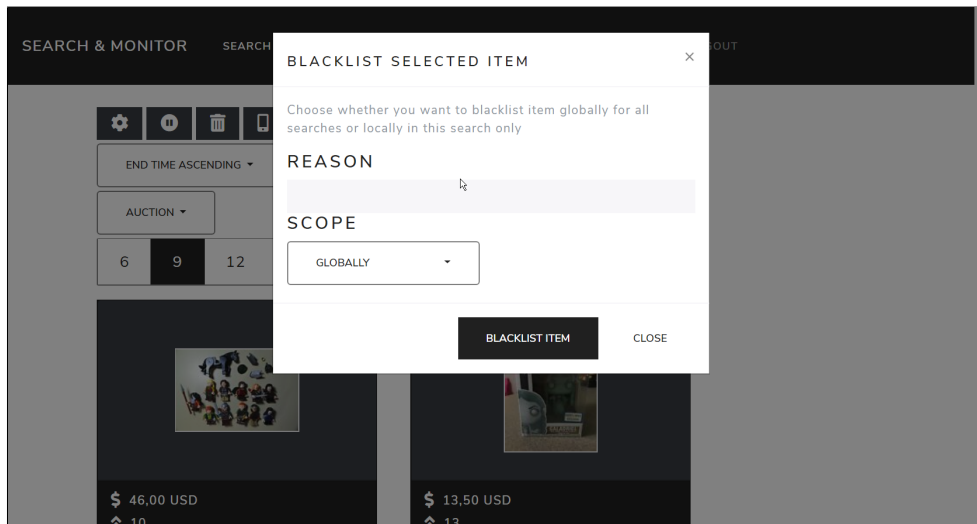
6. VÝSLEDKY



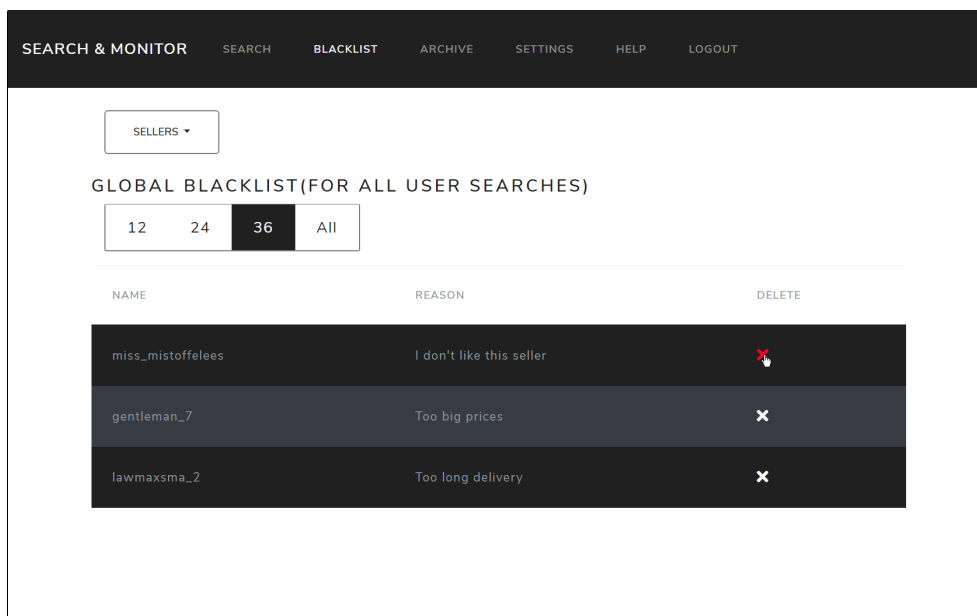
Obr. 6.5: Vyhľadávanie (obsahuje dáta z [1])



Obr. 6.6: Výsledky vyhľadávania (obsahuje dáta z [1])

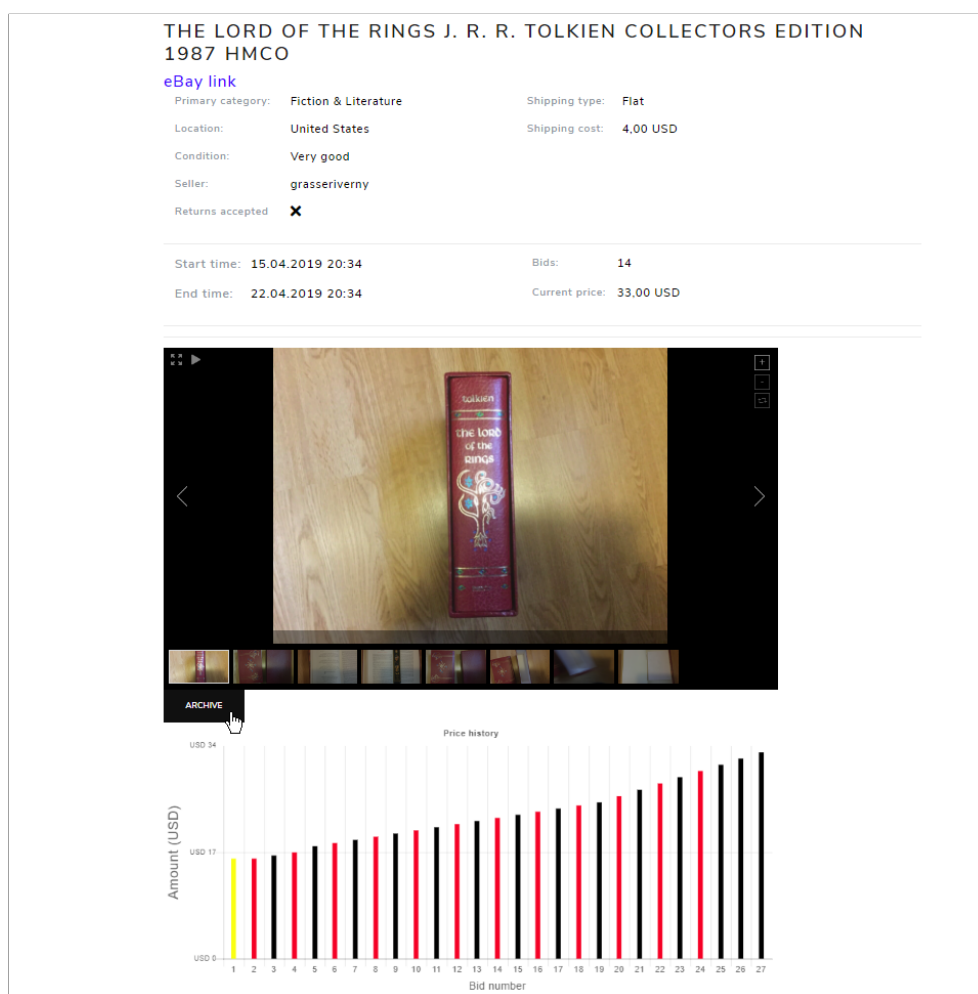


Obr. 6.7: Vytvorenie zákazu (obsahuje dáta z [1])



Obr. 6.8: Zákazy (obsahuje dáta z [1])

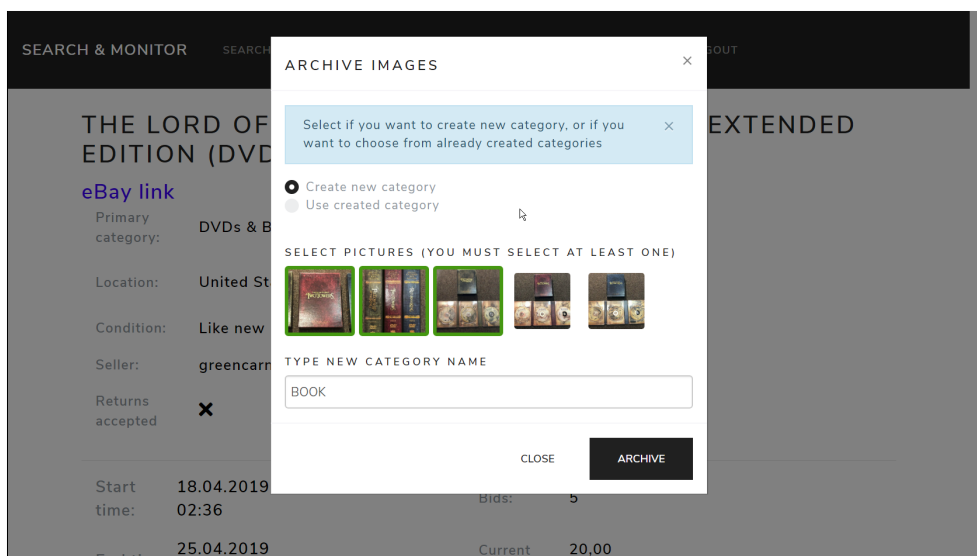
6. VÝSLEDKY



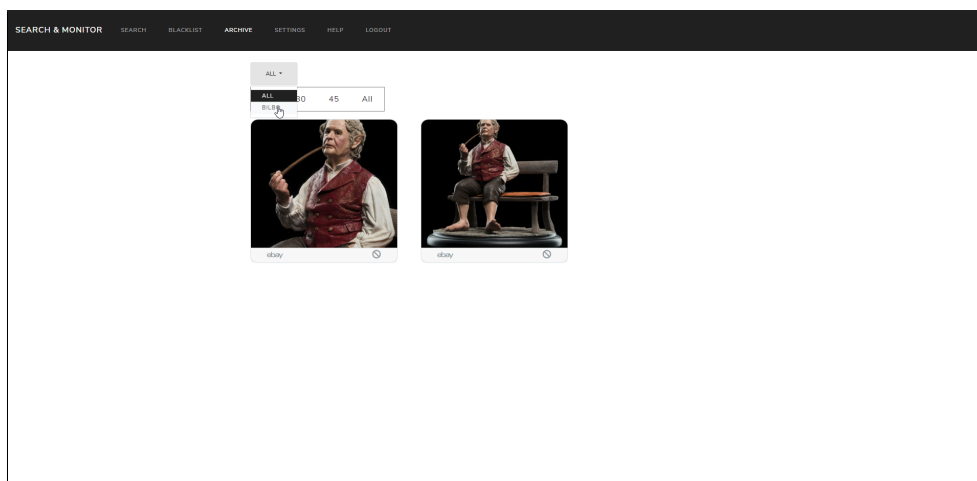
Obr. 6.9: Detaily položky (obsahuje dáta z [1])

6.1.7 Zhrnutie

Vďaka užívateľskému testovaniu som overil, že aplikácia je použiteľná nielen v desktopových, ale aj v mobilných prehliadačoch. Pri testovaní nedošlo k žiadnej chybe, ktorá by zabráňovala normálnemu používaniu aplikácie. Existuje však niekoľko menších nedostatkov, prevažne niektoré detaily užívateľského rozhrania — nejednoznačnosť zoznamu zákazov pre vyhľadávanie a užívateľa, a pozícia tlačidla pre archiváciu obrázkov. Zistenú nefunkčnosť SMS notifikácií je možné odstrániť použitím platených služieb technológie popísanej v časti 4.1.5, poprípade inej služby poskytujúcej podobné možnosti. Väčšinu zistených nedostatkov popísaných v predchádzajúcich častiach som upravil tak, aby bola užívateľovi zjednodušená práca s aplikáciou.



Obr. 6.10: Archivácia obrázkov (obsahuje dáta z [1])



Obr. 6.11: Archív (obsahuje dáta z [1])

6.2 Vyhlíadky do budúcnosti

Na základe užívateľského testovania, vlastného zhodnotenia výsledkov tejto práce a nápadov, ktoré vznikli počas implementácie, ďalej opisujem návrhy na zlepšenie a rozšírenie výslednej aplikácie.

6.2.1 Úprava užívateľského rozhrania

Užívateľské testovanie ukázalo zopár nedostatkov (viď sekcia 6.1.7), ktoré užívateľom prekážali v jednoduchom využívaní aplikácie. Jednotlivé problémy neboli závažné, avšak ich zmena by priniesla užívateľom príjemnejšiu prácu s aplikáciou.

6.2.2 Práca s eBay položkami

Aplikácia vie nájsť jednotlivé výsledky na základe zadaných kritérií. Bolo by dobré, ak by bolo možné pracovať s nájdenými položkami priamo prostredníctvom aplikácie. Pre to by bolo potrebné integrovať eBay s aplikáciou tak, aby užívateľ mohol aplikáciou udeliť prístup k svojmu eBay účtu a tá by ho následne mohla využívať prostredníctvom služieb API.

6.2.3 Limity API

Ako som spomínal v časti 2.2.3, využívanie eBay API je dosť obmedzené, takže v prípade trvalejšieho nasadenia aplikácie pre viac užívateľov by dané maximálne volania boli nedostatočné. Preto by bolo potrebné získať status kompatibilnej aplikácie a tak značne rozšíriť možnosti využívania API.

6.2.4 Viacjazyčnosť

Všetky texty v aplikácii som písal po anglicky. Možnosť zmeny jazyka by mohla niektorým užívateľom pomôcť v plnohodnotnejšom využívaní aplikácie. Vďaka frameworku Spring je to možné docieľiť jednoducho. Keďže všetky používané texty sú uložené v jednom súbore, pre použitie konkrétnej lokalizácie stačí vytvoriť ďalší súbor, kde budú preložené texty. Potom už je len treba zabezpečiť to, že aplikácia bude vedieť aký jazyk daný užívateľ používa.

6.2.5 Strojové učenie

V rámci tejto aplikácie bolo požadované uchovávanie historických dát (hlavne obrazových), na základe ktorých by mohlo byť implementované určité strojové učenie, pomocou ktorého by boli spomedzi nájdených položiek vylučované tie nežiadúce. Avšak pre tento účel je potrebné zabezpečiť veľké množstvo dát, pomocou ktorých by to mohlo byť realizované.

Záver

Cieľom tejto práce bolo navrhnúť a implementovať aplikáciu, pomocou ktorej bude užívateľ schopný vyhľadávať a sledovať položky spĺňajúce určité kritéria na aukčnom portáli eBay tak, aby mu prítomnosť nových položiek bola oznámená. Užívateľ by mal mať takisto možnosť zobrazit si nájdené výsledky, ktoré si následne bude môcť podľa určitých kritérií triediť. Aplikácia by mala uchovávať historické údaje, čo zahŕňa aj možnosť archivovania obrázkov jednotlivých položiek.

Najprv som analyzoval existujúce riešenie, možnosť komunikácie s aukčným portálom eBay a požiadavky na aplikáciu. Následne som na základe analýzy vytvoril návrh aplikácie a zvolil technológie umožňujúce dosiahnutia potrebnej funkcionality. Aplikáciu som po návrhu implementoval. Po implementácii som aplikáciu otestoval pomocou testovacích užívateľov a väčšinu nájdených nedostatkov som odstránil. Zistil som, že výsledná aplikácia je použiteľná a spĺňa rámcové požiadavky zadania.

Zoznam použitej literatúry

1. *eBay* [online]. eBay Inc., © 1995-2019 [cit. 2019-04-02]. Dostupné z: <https://www.ebay.com/>.
2. *eBay Developers Program* [online]. eBay Inc., © 1999-2019 [cit. 2019-03-31]. Dostupné z: <https://developer.ebay.com/>.
3. NDEGWA, Amos. What is a Web Application? *MaxCDN* [online]. 2016 [cit. 2019-03-31]. Dostupné z: <https://www.maxcdn.com/one/visual-glossary/web-application/>.
4. HOODA, Parikshit. Model-View-Controller(MVC) architecture for Node applications. *GeeksforGeek: A computer science portal for geeks* [online] [cit. 2019-03-31]. Dostupné z: <https://www.geeksforgeeks.org/model-view-controllermvc-architecture-for-node-applications/>.
5. LUCIDCHART. What is a Database Model. *LucidChart* [online] [cit. 2019-03-31]. Dostupné z: <https://www.lucidchart.com/pages/database-diagram/database-models>.
6. A PRODULE SYSTEMS PVT LTD. *MockFlow* [software] [cit. 2019-04-04]. Dostupné z: <https://www.mockflow.com/>.
7. ORACLE CORPORATION. *Java SE Development Kit 8* [software] [cit. 2019-03-31]. Dostupné z: <https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>.
8. PIVOTAL SOFTWARE. *Spring* [software] [cit. 2019-03-31]. Dostupné z: <https://spring.io/>.
9. RED HAT, INC. *Hibernate* [software] [cit. 2019-03-31]. Dostupné z: <http://hibernate.org/>.
10. POSTGRESQL GLOBAL DEVELOPMENT GROUP. *PostgreSQL* [software] [cit. 2019-03-31]. Dostupné z: <https://www.postgresql.org/>.
11. TWILIO INC. *Twilio* [software] [cit. 2019-03-31]. Dostupné z: <https://www.twilio.com/>.

12. *Google Drive* [online]. Google LLC, © 2019 [cit. 2019-04-13]. Dostupné z: <https://www.google.com/drive/>.
13. *Google Drive API* [online]. Google LLC, © 2019 [cit. 2019-04-13]. Dostupné z: <https://developers.google.com/drive/api/v3/about-sdk>.
14. GOOGLE INC. *Google Cloud* [software] [cit. 2019-03-31]. Dostupné z: <https://cloud.google.com/>.
15. JOHN RESIG. *jQuery* [software] [cit. 2019-03-31]. Dostupné z: <https://jquery.com/>.
16. MARK OTTO, JACOB THORNTON. *Bootstrap* [software] [cit. 2019-03-31]. Dostupné z: <https://getbootstrap.com/>.
17. THOMAS PARK. *Bootstrap Lux* [software] [cit. 2019-03-31]. Dostupné z: <https://bootswatch.com/lux/>.
18. DANIEL FERNÁNDEZ. *Thymeleaf* [software] [cit. 2019-04-23]. Dostupné z: <https://www.thymeleaf.org/>.
19. APACHE SOFTWARE FOUNDATION. *Apache Maven* [software] [cit. 2019-03-31]. Dostupné z: <https://maven.apache.org/what-is-maven.html>.
20. PARASCHIV, Eugen. The State of Java in 2018. *Baeldung* [online] [cit. 2019-03-31]. Dostupné z: <https://www.baeldung.com/java-in-2018>.
21. LINUS TORVALDS. *Git* [software] [cit. 2019-03-31]. Dostupné z: <https://git-scm.com>.
22. ECLIPSE FOUNDATION. *Eclipse* [software] [cit. 2019-03-31]. Dostupné z: <https://www.eclipse.org/>.
23. PIVOTAL SOFTWARE. *Spring Tools Suite 4* [software] [cit. 2019-03-31]. Dostupné z: <https://spring.io/tools>.
24. THE PROJECT LOMBOK AUTHORS. *Project Lombok* [software] [cit. 2019-04-05]. Dostupné z: <https://projectlombok.org/>.
25. SNAPAPPOINTMENTS. *Bootstrap-select* [software] [cit. 2019-03-31]. Dostupné z: <https://github.com/snapappointments/bootstrap-select/>.
26. UX SOLUTIONS. *Bootstrap-datepicker* [software] [cit. 2019-03-31]. Dostupné z: <https://github.com/uxsolutions/bootstrap-datepicker>.
27. TIM SCHLECHTER A ĎALŠÍ. *Bootstrap-tagsinput* [software] [cit. 2019-03-31]. Dostupné z: <https://github.com/bootstrap-tagsinput>.
28. EVERT TIMBERG A ĎALŠÍ. *Chart.js* [software] [cit. 2019-03-31]. Dostupné z: <https://www.chartjs.org/>.
29. VALIANO. *Unite Gallery* [software] [cit. 2019-03-31]. Dostupné z: <http://unitegallery.net/>.

30. FONT AWESOME TEAM. *Font awesome* [software] [cit. 2019-03-31]. Dostupné z: <https://fontawesome.com/>.
31. BEN PODGURSKY. *JBool Expressions* [software] [cit. 2019-03-31]. Dostupné z: https://github.com/bpodgursky/jbool_expressions.

Zoznam použitých skratiek

AJAX Asynchronous JavaScript + XML

API Application Programming Interface

CRUD Create, Read, Update, Delete

CSS Cascading Style Sheets

HTML Hypertext Markup Language

JAR Java Archive

JPA Java Persistence API

JSON JavaScript Object Notation

MVC Model-View-Controller

SDK Software Development Kit

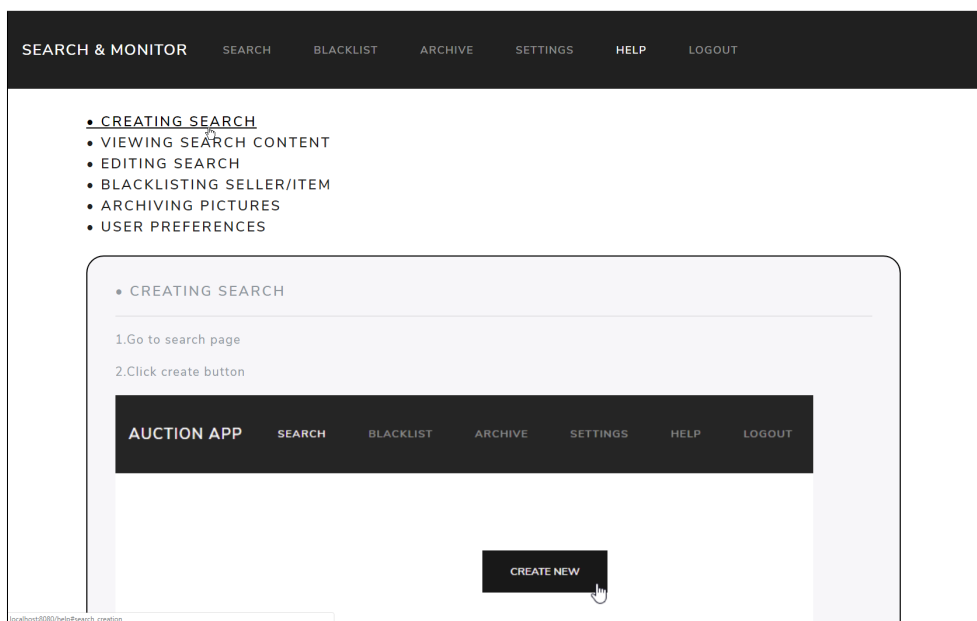
SQL Structured Query Language

USD United States dollar

WSDL Web Service Definition Language

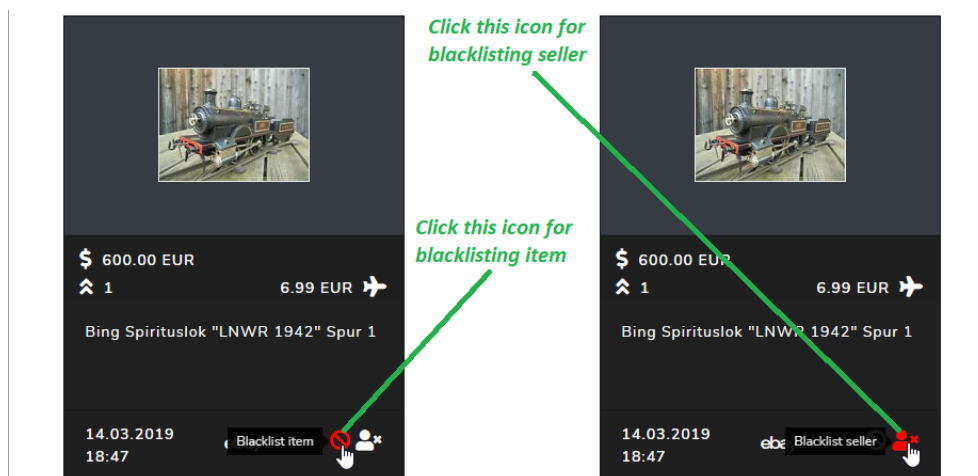
XML Extensible Markup Language

Ďalšie screenshoty aplikácie



Obr. B.1: Pomoc — obsahuje návody pre jednotlivé funkcionality aplikácie

B. ĎALŠIE SCREENSHOTY APLIKÁCIE



Obr. B.2: Pomoc — vytvorenie zákazu

SEARCH FORM

EBAY SITE

Pick eBay localization

NAME *Name for your characterization*

DESCRIPTION *Description for your characterization*

SHIPPING COUNTRY *Pick country to which seller can ship sold items*

LISTING TYPES *Pick listing types of desired items, different listing types are explained upon collapse*

CATEGORIES *Pick up to 3 categories of desired items*

KEYWORDS

Search in description *Pick whether keywords should be searched in description too*

Write keywords for your desired items, you can use symbols &(and), |(or), !(not) and quotation marks(encapsulating keyword in quotes mean you want exact sequence of specified words)

Example: (car & black & !red) |(train & green) -> Application will search for black cars which is not red and green train

SELLERS *Pick sellers of your desired items*

END DATE FROM *Ending time from date* **END DATE TO** *Ending time to date*

MIN BIDS *Minimal amount of bids(only auction listings)* **MAX BIDS** *Maximal amount of bids(only auciton listings)*

MIN PRICE(EUR) *Minimal price* **MAX PRICE(EUR)** *Maximal price*

REFRESH TIME(MIN) *Refresh time of search*

OTHER OPTIONS

New Only *Search only for items which are brand new*

Free Shipping Only *Search only for items for which shipping is free*

Obr. B.3: Pomoc — popis jednotlivých kritérií pri vytváraní vyhľadávania

B. ĎALŠIE SCREENSHOTY APLIKÁCIE

USER PREFERENCES

SEARCH DEFAULTS

EBAY SITE

Pick eBay localization

SHIPPING COUNTRIES

Pick country to which seller can ship sold items

LISTING TYPES

Pick listing types of desired items, different listing types are explain upon collapse

REFRESH TIME(MIN) *Pick refresh time for search*

New Only *Search only for items for which are brand new*

Free Shipping Only *Search only for items for which shipping is free*

Returns Accepted Only *Search only for items for which returns are accepted*

Enable notifications *Enable notifications for phone*

OTHER

Stay logged in *Stay logged in (or choose inactivity logout)*

INACTIVITY LOGOUT(MIN)

ACCOUNT CURRENCY

Pick currency in which you wish to see items and define your cost criteria

Add phone number

Obr. B.4: Pomoc — popis jednotlivých nastavení

Obsah priloženého CD

| | | |
|--|------------------|---|
| | readme.txt | stručný popis obsahu CD |
| | exe..... | adresár so spustiteľnou formou implementácie |
| | src | |
| | impl..... | zdrojové kódy implementácie |
| | thesis..... | zdrojová forma práce vo formáte L ^A T _E X |
| | text | text práce |
| | thesis.pdf..... | text práce vo formáte PDF |
| | thesis.ps..... | text práce vo formáte PS |