

ECO - Electronic Cadastral Office

1. Zámer projektu

1.1. Zadanie projektu

E-správa (E-Governance)

Okrem podpory a zabezpečenia demokratickosti vládnutia (e-government), informačné a komunikačné technológie umožňujú aj podporu množstva ďalších procesov súvisiacich so štátnou správou. Tzv. e-správa (e-governance) zahŕňa podporu akýchkoľvek procesov medzi štátnou alebo lokálnou správou a fyzickými alebo právnickými osobami. Môže sa to týkať pobytu, nehnuteľností, zamestnania, finančnej podpory, školstva a pod.

Príklady e-správy zahŕňajú elektronické zabezpečenie podávania žiadosti o víza a udeľovania víz, komunikácie s finančnou správou, resp. s jej ľubovoľnou sekciou (daňový, živnostenský, colný úrad a pod.), poradenstva pre študentov študujúcich v zahraničí, registrácie privezeného dopravného prostriedku zo zahraničia a podania daňového priznania.

1.2. Zámer

Projekt je zameraný na spravovanie pozemkov, nehnuteľností na daných pozemkoch a príslušných práv k nim určených.

Program bude slúžiť Katastrálnemu úradu spravovať verejný register pozemkov a nehnuteľností a poskytne lepší prehľad o právnych úkonoch, ktoré sú spojené so správou. Do tejto činnosti sa zahŕňa evidovanie pozemkov a stavieb; členenie pozemkov; zapisovanie práv k pozemkom a nehnuteľnostiam; evidovanie majiteľov; spracovanie požiadavok o prevode vlastníckeho práva, zahŕňajúc kúpnu zmluvu, darovaciu zmluvu, zámennú zmluvu pozemkov; komunikácia s Úradom geodézie, kartografie a katastra ohľadom stavebnej činnosti; komunikácia s Daňovým úradom ohľadom výpočtu sumy dane za pozemok a nehnuteľnosť; zmena hraníc území; ukladanie pokút za niektoré priestupky na úseku geodézie a katastra.

Program taktiež bude slúžiť občanom. Občan, ktorý nebude registrovaný v danom programe, bude môcť prehľadávať parcely a bude mať prístup k vlastníkom a ich údajom. Registrovaný občan bude môcť komunikovať s úradmi spomenutými vyššie, podávať rôzne žiadosti týkajúce sa správy jeho vlastníctva, vystavovať zmluvy.

1.3. Splnenie zámeru

Vo finálnej verzii programu nie je zahrnutá celá myšlienka a funkcionalita pôvodného zámeru.

1.3.1. Nesplnené funkcie uvedené v zámere

Celkovo nie sú zahrnuté financie v danom programe. Tie mali byť spojené s Daňovým úradom, ktorý si teda vôbec nenašiel uplatnenie a do programu sa neimplementoval. Spolu s ním ani nehnuteľnosti nemajú vyčíslené dane, ktoré by musel občan platiť. Ďalej sa ani nezrealizovalo ukladanie pokút z úseku geodézie a katastra, keďže nie je vytvorený algoritmus na posudzovanie pokutovania občanov. Ďalej financie nehrajú rolu pri kúpnej zmluve pozemku, keďže nedochádza k prevodu peňazí v rámci programu. A ani pri vytvorení požiadavky („requestu“) (resp. vkladu do katastra) sa nevykonáva žiaden prevod peňazí (ktorý by mal byť v hodnote 33€, ktorý by sa mal zapísať do úradnej pokladnice).

Ďalej v adminskej (úradnej časti) nie je implementovaná funkcia, pre zmenu hraníc jednotlivých pozemkov.

Vo vykonateľných požiadavkách nie je zahrnutá požiadavka o vytvorení zámennej zmluvy.

1.3.2. Splnené funkcie uvedené v zámere

V úradnej časti úradník vie evidovať pozemky, pozemky vie členiť podľa oficiálneho členenia pozemkov, ďalej úradník vie evidovať budovy. K daným nehnuteľnostiam vie evidovať majiteľov. Úradník vie spracovávať požiadavky od používateľov, vie ich preposielať Úradu geodézie, kartografie a katastra, ak sa jedná o úpravy týkajúce sa nehnuteľnosti. Ak sa požiadavka týka úpravy majteľov, vie uvážiť o akceptovaní požiadavky.

Program vedia používať aj registrovaní aj neregistrovaní občania.

Neregistrovaný občan vie prehľadávať občanov a ich nehnuteľnosti.

Registrovaný používateľ vie navyše posilať žiadosti Katastrálnemu úradu a vie teda vykonávať zmeny v rámci svojho vlastníctva, nad ktorým má prehľad v rámci programu.

2. Štruktúra

2.1. Kritéria projektu

2.1.1. Splnené hlavné kritéria

Dedenie

- balík *CadasterObjects* – triedy *Land* a *RealEstate* dedia od abstraktnej triedy *CadasterObject*
- package *Ownership* – triedy *City* a *Owner* dedia od abstraktnej triedy *Ownership*

Polymorfizmus

- balík *CadasterObjects* – metóda *toString()* v triedach *CadasterObject*, *Land* a *RealEstate*

Agregácia

- balík *UserObject* - trieda *User* agreguje objekt triedy *Ownership*
- balík *Owners* - trieda *Ownership* agreguje polia tried *Land* a *RealEstate*
- agregácia sa využíva aj v triedach *CadastreObject*, *Land*, *RealEstate*, *Request*

Enkapsulácia

- v skoro každom balíku sa nachádza aspoň 1, kde sa využíva enkapsulácia
- napr. v balíku *CadsterObjects* trieda *CadasterObject* sa nachádza enkapsulácia

2.1.2. Splnené ďalšie kritéria

Použitie návrhových vzorov okrem návrhového vzoru Singleton

- *Visitor* – balík *ViewControllers* – trieda *MakeOwnerController* metóda *btnClicked*

Využitie: ak je metóda spustená pre usera s *UserType.CITIZEN*, vytvorí v ňom inštanciu triedy *Onwera*, inak vytvorí inštanciu triedy *City*

- *Observer* – balík *OtherFunctionality* – trieda *DataObserver* (ktorá implementuje interface *Observer*)

Využitie: ak sa vytvorí nový používateľ, nový vlastník v používateľovi alebo sa vytvorí nová nehnuteľnosť, všetky tieto informácie sa aktualizujú v zobrazení v uzle *TextArea*, ktorá dané informácie vypisuje v hlavnom okne používateľa

Ošetrovanie mimoriadnych stavov prostredníctvom vlastných výnimiek

- package *MyExeptions*
- príkl.1 výnimka *AddressFormatException* – využíva sa v balíku *CadasterObjects* v triede *Address*, v metóde *correctAddress()*, ktorá kontroluje správny zápis adresy (majiteľa, nehnuteľnosti). Ak adresa nebude spĺňať nejaké kritéria, hodí novú výnimku, ktorá je následne chytná v GUI, kde následne vytvorí a vypíše chybovú hlášku danej výnimky.

- príkl.2 výnimka *SameRegNumException* – využíva sa v balíku *OtherFunctionality* v triede *CadasterSearch*, v metóde *compareAddress()*, ktorá zisťuje, či úradník nezadal už existujúce registračné číslo v zadanom meste. Ak zadal, metóda hodí novú výnimku, ktorá je následne chytená v GUI, kde následne vytvorí a vypíše chybovú hlášku danej výnimky.

Poskytnutie grafického používateľského rozhrania oddelene od aplikačnej logiky

- celá vytvorená manuálne
- balík *View*

Použitie vnhiezdených tried a rozhraní

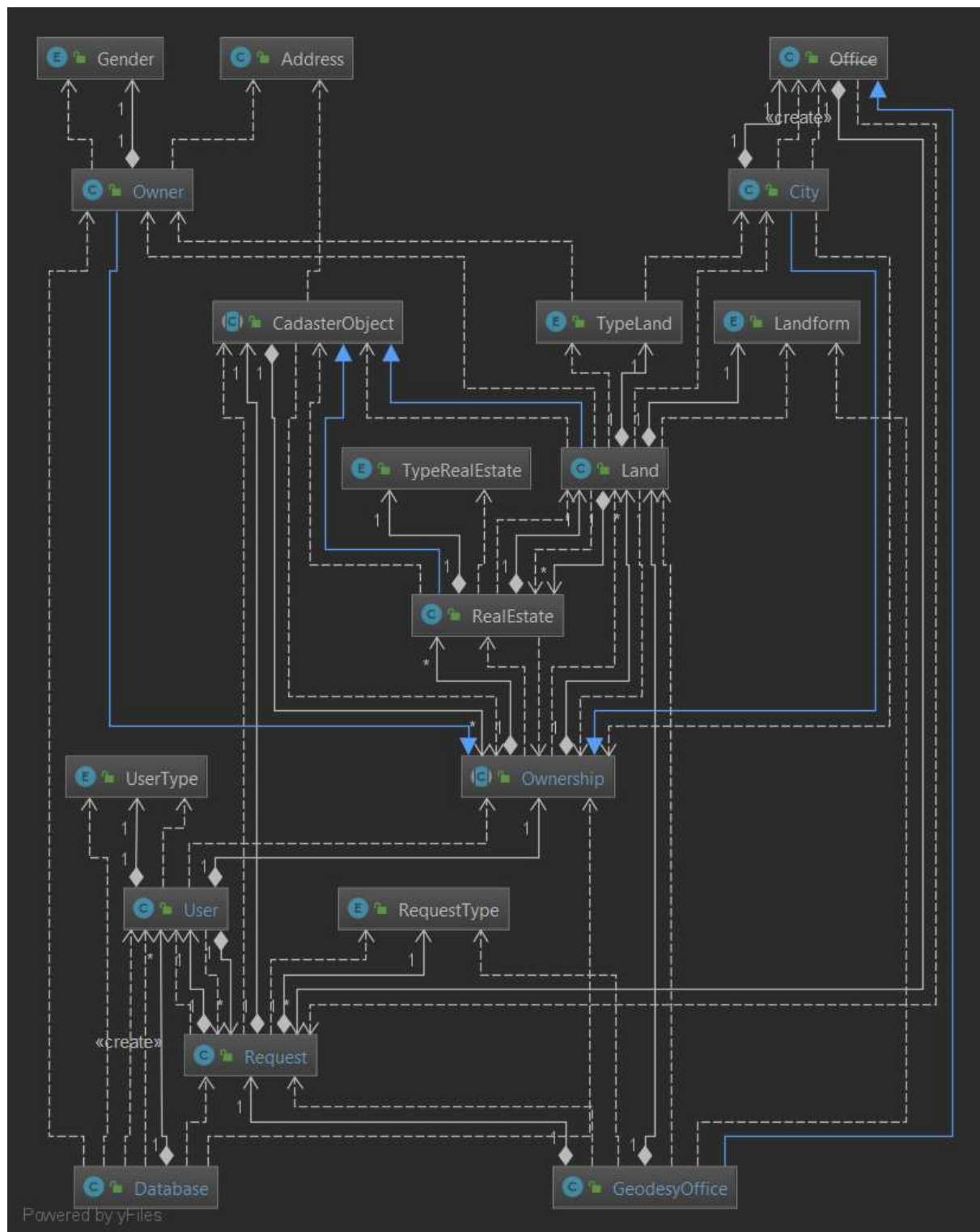
- balík *View* – trieda *RequestStage* (GUI trieda) – vnhiezdená trieda *RequestButtonHandler*
Využitie: daná trieda implementuje rozhranie *EventHandler<ActionEvents>* a využíva sa ako custom action handler pre každý zvlášť vytvorený button.

Explicitné použitie RTTI

- balík *OtherFunctionality* trieda *CadasterSearch* v metódach *compareRegNum()* a *compareAddress()*, trieda *PrintCadastre* v metóde *print()*
- balík *CadsterObjects* trieda *Land* v metóde *setTypeLand()*

2.2. Hierarchia tried

Uml diagram modelov



3. Comitty

3.1. Hlavné verzie programu

3.1.1. Commit: started printing users

- Používatelia sa serializovali do *HahhMapy*
- Vytvárali si používatelia a vlastníci v nich
- Vypisovali sa dané údaje v inom vlákne

3.1.2. Commit: fixed bug with loading updated database after office log out

- Plné serializovanie používateľov
- Prenášanie objektu *Database* z jedného *stage* do druhého pomocou atribútov (nie pôvodným neustálym deserializovaním databázy)
- Bezproblémové načítavanie

3.1.3. Commit: ownership have new custom exceptions

- Implementovanie vlastných výnimok *EmailFormatException* a *PhoneNumberException* v triede *Ownership*
- Vytváranie pozemkov
- Neskôr vytváranie budov

3.1.4. Commit: observer for printing done

- Vytvorený observer *DataObserver*, ktorý sledoval zmenu údajov pri vytvorení nových informácií, ktoré automaticky po zmene vypísal do *TextArea* na hlavnom okne používateľa

3.1.5. Commit: requests done

- Používatelia vedia vytvárať požiadavky a odosielať ich zvolenému administrátorovi
- Prideľovanie budovy k pozemkom a opačne podľa registračného čísla a ulice/ mesta
- (bohužiaľ vo finálnej verzii, požiadavky nefungujú tak, ak v danom commite, keďže program prechádzal pár zásadnými zmenami hlavne na úrovni používateľov)
- Aktualizované vypisovanie používateľov – nevykonáva sa daná akcia v inom vlákne, kvôli neustálemu padaniu vlákna (errorom)