**Веб апликација за преглед и управљање катастарским подацима**

Оливер Којић

Факултет техничких наука

Универзитет у Новом Саду

Трг Доситеја Обрадовића 6, 21000 Нови Сад

kojic.sr12.2020@uns.ac.rs

**Сажетак - У овом раду описана је веб апликација [1] за катастар, која омогућава ефикасно управљање катастарским подацима. За реализацију су коришћене технологије као што су Spring Boot [2], Angular [3] и MySQL [4]. Применом предложеног решења, постигнуте су предности, као што су повећана ефикасност, транспарентност и аутоматизација у управљању подацима о непокретностима, праћење историја промена података над појединачном непокретношћу и интероперабилност са информационим системима матичара и нотара при преносу власништва над неким земљиштем. Као резултат рада, развијен је прототип апликације, који је у даљем тексту детаљније описан и демонстрирано је његово практично коришћење.**

***Кључне речи – катастар; непокретност; парцела; власништво; историја промена***

# Увод

Катастар има кључну улогу у вођењу евиденција о непокретностима и пружању релевантних информација о истим. Међутим, катастарски систем се суочава са неким проблемима који отежавају вођење евиденција и приступ подацима.

Један од главних проблема је ручно вођење папирних евиденција. То често доводи до потешкоћа у ефикасном вођењу евиденција непокретности, а расте и могућност за грешкама.

Други проблем који се јавља у традиционалном систему катастра јесте могуће дуготрајно чекање грађана на шалтерима приликом добијања папира о непокретности или обављања процедура као што је промена власништва над непокретношћу. Грађани неретко треба да се спреме на дуге редове, као и на ограничено радно време. Тиме могу да изгубе можда драгоцено време, а све то може представљати додатни терет онима не живе близу места где су шалтери.

Циљ овог семинарског рада је развој информационог система катастра који ће решити ове проблеме. Нови систем има за циљ побољшање ефикасности и транспарентности у управљању катастарским подацима, као и евиденција историје промена података над неком непокретношћу. Такође повећана је аутоматизација процеса тако што катастар има комуникацију са информационим системима матичара и нотара приликом преноса власништва над земљиштем. Кроз развој апликације омогућен је једноставан приступ подацима и повећана поузданост током вођења евиденција о непокретностима.

Поред тренутних изазова у управљању катастарским подацима, такође је важно размотрити и будућност. У данашње дигитално доба, где се подаци све више дигитализују и процеси постају аутоматизовани, информациони систем катастра би требало да постане основа за дигитално управљање подацима о непокретностима. Управљање катастарским подацима има кључну улогу у различитим областима као што су урбанизација и развој инфраструктуре, а самим тим и у области економије. Ефикасан и поуздан информациони систем катастра може да омогући инвеститорима, грађанима и осталим заинтересованим странама да брзо приступе релевантним подацима везаним за непокретности.

Остатак рада је организован као што је објашњено у наставку. У другом поглављу је дат преглед сродних решења и коришћених технологија. У трећем поглављу је наведена спецификација захтева. У четвртом поглављу је представљена спецификација дизајна. Пето поглавље објашњава имплементацију информационог система катастра. У шестом поглављу је приказана демонстрација. Седмо поглавље доноси закључна разматрања и правце будућих истраживања.

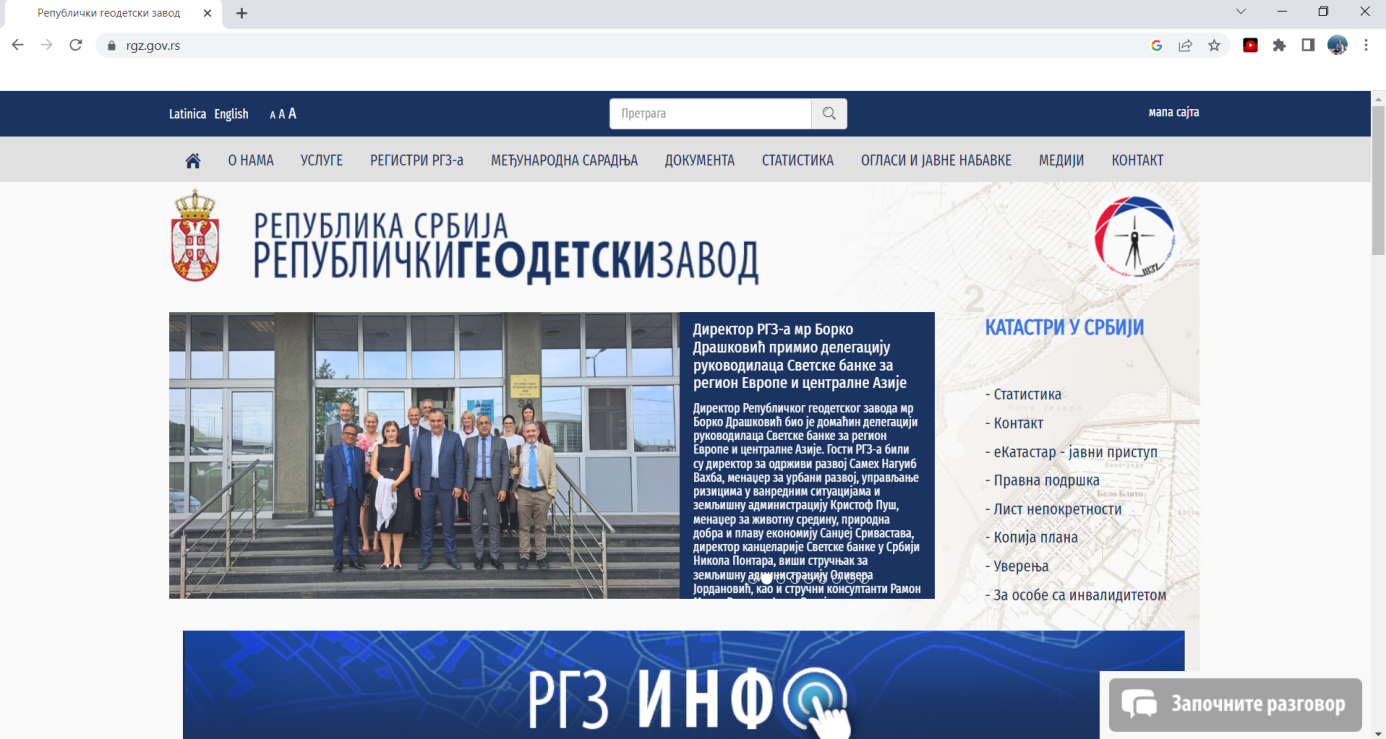
# Сродна решења и преглед коришћених технологија

У овом поглављу дат је преглед постојећих решења за развој информационог система катастра и наведене су технологије које омогућавају решавање проблема које су наведене у поглављу Увод овог семинарског рада.

## Сродна решења

Овај одељак доноси преглед сродних апликација намењених прегледу и управљању катастарским подацима.

Републички геодетски завод (РГЗ) [5] је српски национални завод који има кључну улогу у прикупљању, чувању и пружању информација о непокретностима у Републици Србији. Изглед почетне странице сајта Републичког геодетског завода је приказан на слици 1.



Слика 1 - Изглед почетне странице сајта Републичког геодетског завода

Добре стране РГЗ-а су:

* могућност прегледа основних информација о том заводу као што су њихова организациона структура, делокруг рада, историјат, стратегија развоја, преглед многих статистика.
* пружање важних информативних података, као што су списак катастарских општина, преглед вести везаних за геодетски завод, приказ контакт података.
* могућност прегледа важних катастарских докумената, као што су закони, уредбе, правилници, директиве, упутства итд. уз могућност преузимања сваког документа у

PDF [6] формату.

* пружање online услуга, као што је еШалтер за грађане која у себи обухвата многе online услуге као што су јавни приступ еКатастру, могућност еЗаказивања термина за предају докумената на шалтеру, могућност додавања еПримедбе на рад катастра итд.
* подржана претрага целог сајта Републичког геодетског завода на основу кључних речи, као и могућност бирања језика (српски, енглески) и писма (ћирилица, латиница).

Лоша страна РГЗ-а је да није у потпуности решен проблем могућег дуготрајног чекања грађана на шалтерима, јер и даље за обављање неких катастарских процедура попут промене власништва над земљиштем грађанин треба да прикупља потребну документацију и иде по шалтерима, уместо да се то више дигитализује и аутоматизује.

## Коришћене технологије

Овај одељак пружа преглед и објашњења технологија помоћу којих је могућ развој информационог система катастра.

За развој серверског дела апликације коришћен је Spring Boot. Spring Boot је open-source [7] алат који олакшава коришћење Java[8]-базираних радних оквира за креирање микросервисних[9] и веб апликација. Spring Boot пружа добру платформу за развој

stand-alone[10] и продукционих апликација који се могу једноставно покренути. Spring Boot нуди неке предности, као што су лако разумевање и развијање апликација, повећана продуктивност и смањење времена развоја апликација.

За развој клијентског дела апликације коришћен је Angular. Angular је развојна платформа изграђена на TypeScript-у [11]. Као платформа, Angular обухвата компоненте за изградњу скалабилних веб апликација, колекције добро интегрисаних библиотека које покривају широк спектар функција (рутирање, управљање формама, клијент-сервер [12] комуникацију итд) и пакет алатки за програмере који помажу при развијању, тестирању и ажурирању кода.

За чување података коришћена је MySQL база података. Према DB-Engines-у [13], MySQL се рангира као друга најпопуларнија база података на свету, иза Oracle [14] базе података. MySQL користе многе познате апликације, као што су Facebook [15], Twitter [16] и други. MySQL је open-source база података, која је по својој природи релациона [17] и може да ради на

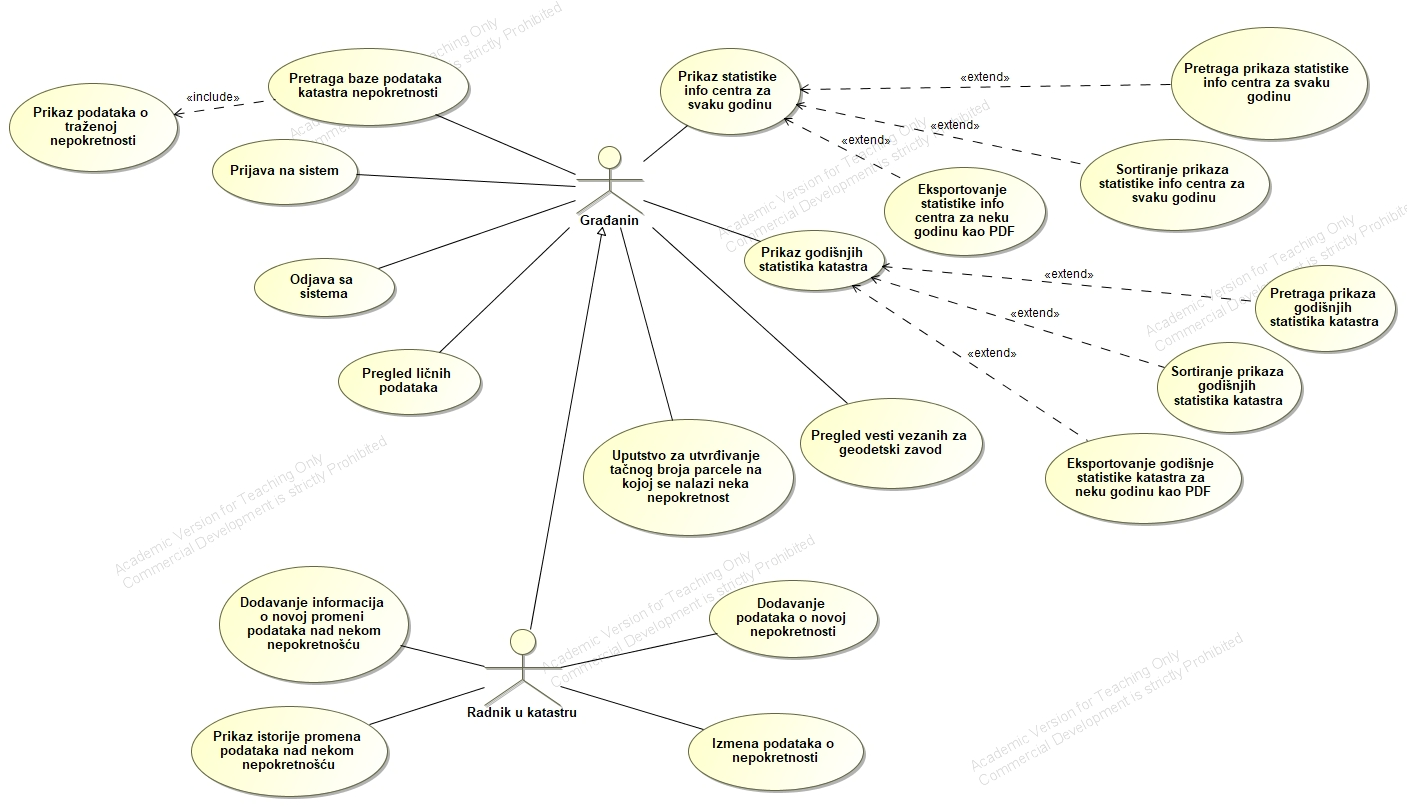
клијент-сервер или embedded [18] системима. Неки од бенефита коришћења ове базе података су једноставно коришћење, поузданост, скалабилност, висока доступност итд.

# Спецификација захтева

У овом поглављу су објашњени функционални и нефункционални захтеви софтверског решења представљеног у овом раду.

## Спецификација функционалних захтева

У овом одељку су описани функционални захтеви које је потребно да испуњава софтверско решење информационог система катастра. Функционални захтеви овог софтверског решења су представљени UML [19] дијаграмом случајева коришћења, као што је приказано на слици 2.



Слика 2 – Приказ UML дијаграма случајева коришћења

Табела 1 приказује опис случаја коришћења “Пријава на систем“.

|  |  |
| --- | --- |
| Назив | Пријава на систем |
| Учесници | Грађанин, Радник у катастру |
| Предуслови | - |
| Кораци | 1. Грађанин/Радник у катастру бира опцију за пријаву.  2. Грађанин/Радник у катастру уноси корисничко име и лозинку.  3. Грађанин/Радник у катастру потврђује унос. |
| Резултат | Грађанин/Радник у катастру је пријављен на систем. |
| Изузеци | Погрешно корисничко име или лозинка. |

Табела 1 – Опис случаја коришћења “Пријава на систем“

Табела 2 приказује опис случаја коришћења “Одјава са система“.

|  |  |
| --- | --- |
| Назив | Одјава са система |
| Учесници | Грађанин, Радник у катастру |
| Предуслови | Грађанин/Радник у катастру је пријављен на систем. |
| Кораци | 1. Грађанин/Радник у катастру бира опцију за одјаву са система.  2. Грађанин/Радник у катастру потврђује унос. |
| Резултат | Грађанин/Радник у катастру је одјављен са система. |
| Изузеци | - |

Табела 2 – Опис случаја коришћења “Одјава са системa“

Табела 3 приказује опис случаја коришћења “Преглед личних података“.

|  |  |
| --- | --- |
| Назив | Преглед личних података |
| Учесници | Грађанин, Радник у катастру |
| Предуслови | Грађанин/Радник у катастру је пријављен на систем. |
| Кораци | 1. Грађанин/Радник у катастру бира опцију за преглед личних података. |
| Резултат | Грађанин/Радник у катастру је прегледао своје личне податке. |
| Изузеци | - |

Табела 3 – Опис случаја коришћења “Преглед личних података“

Табела 4 приказује опис случаја коришћења “Претрага базе података катастра непокретности“.

|  |  |
| --- | --- |
| Назив | Претрага базе података катастра непокретности |
| Учесници | Грађанин, Радник у катастру |
| Предуслови | Грађанин/Радник у катастру је пријављен на систем. |
| Кораци | 1. Грађанин/Радник у катастру уноси број парцеле.  2. Грађанин/Радник у катастру потврђује унос. |
| Резултат | Грађанин/Радник у катастру је претражио базу података катастра непокретности. |
| Изузеци | Унесени број парцеле не постоји у бази података катастра непокретности. |

Табела 4 – Опис случаја коришћења “Претрага базе података катастра непокретности“

Табела 5 приказује опис случаја коришћења “Приказ података о траженој непокретности“.

|  |  |
| --- | --- |
| Назив | Приказ података о траженој непокретности |
| Учесници | Грађанин, Радник у катастру |
| Предуслови | Грађанин/Радник у катастру је претходно извршио претрагу базе података катастра непокретности. |
| Кораци | 1. Апликација приказује податке о траженој непокретности попут адресе, потеса, површине, начина коришћења земљишта, имена и презимена садашњег и претходног власника. |
| Резултат | Грађанин/Радник у катастру је прегледао податке о траженој непокретности. |
| Изузеци | - |

Табела 5 – Опис случаја коришћења “Приказ података о траженој непокретности“

Табела 6 приказује опис случаја коришћења “Приказ статистике info центра за сваку годину“.

|  |  |
| --- | --- |
| Назив | Приказ статистике info центра за сваку годину |
| Учесници | Грађанин, Радник у катастру |
| Предуслови | Грађанин/Радник у катастру је пријављен на систем. |
| Кораци | 1. Грађанин/Радник у катастру бира опцију за приказ статистика info центра.  2. Апликација табеларно приказује сваку годину од почетка рада и за сваку годину приказује податке попут броја колико пута су катастарски радници одговорили на позив, просечног времена одговора на позив, броја одговора на мејлове грађана итд. |
| Резултат | Грађанин/Радник у катастру је прегледао приказ статистика info центра катастра. |
| Изузеци | - |

Табела 6 – Опис случаја коришћења “Приказ статистике info центра за сваку годину“

Табела 7 приказује опис случаја коришћења “Претрага приказа статистике info центра за сваку годину“.

|  |  |
| --- | --- |
| Назив | Претрага приказа статистике info центра за сваку годину |
| Учесници | Грађанин, Радник у катастру |
| Предуслови | Грађанину/Раднику у катастру је приказана страница са табеларним приказом статистика info центра катастра за сваку годину рада. |
| Кораци | 1. Грађанин/радник у катастру уноси годину.  2. Грађанин/радник у катастру потврђује унос. |
| Резултат | Грађанин/Радник у катастру је претражио податке о статистици info центра катастра за унесену годину. |
| Изузеци | За унесену годину не постоји статистика info центра катастра. |

Табела 7 – Опис случаја коришћења “Претрага приказа статистике info центра за сваку годину“

Табела 8 приказује опис случаја коришћења “Сортирање приказа статистике info центра за сваку годину“.

|  |  |
| --- | --- |
| Назив | Сортирање приказа статистике info центра за сваку годину |
| Учесници | Грађанин, Радник у катастру |
| Предуслови | Грађанину/Раднику у катастру је приказана страница са табеларним приказом статистика info центра катастра за сваку годину рада. |
| Кораци | 1. Грађанин/радник у катастру бира колону по којој жели да сортира табеларни приказ статистика info центра катастра. |
| Резултат | Грађанин/Радник у катастру је сортирао податке о статистици info центра катастра по одабраној колони. |
| Изузеци | - |

Табела 8 – Опис случаја коришћења “Сортирање приказа статистике info центра за сваку годину“

Табела 9 приказује опис случаја коришћења “Експортовање статистике info центра за неку годину као PDF“.

|  |  |
| --- | --- |
| Назив | Експортовање статистике info центра за неку годину као PDF |
| Учесници | Грађанин, Радник у катастру |
| Предуслови | Грађанину/Раднику у катастру је приказана страница са табеларним приказом статистика info центра катастра за сваку годину рада. |
| Кораци | 1. Грађанин/радник у катастру бира одговарајућу годину чије податке жели да експортује као PDF. |
| Резултат | Грађанин/Радник у катастру је експортовао податке за одабрану годину у PDF формату. |
| Изузеци | - |

Табела 9 – Опис случаја коришћења “Експортовање статистике info центра за неку годину као PDF“

Табела 10 приказује опис случаја коришћења “Приказ годишњих статистика катастра“.

|  |  |
| --- | --- |
| Назив | Приказ годишњих статистика катастра |
| Учесници | Грађанин, Радник у катастру |
| Предуслови | Грађанин/Радник у катастру је пријављен на систем. |
| Кораци | 1. Грађанин/Радник у катастру бира опцију за приказ годишњих статистика катастра.  2. Апликација табеларно приказује сваку годину од почетка рада и за сваку годину приказује податке попут броја поднетих захтева, броја решених захтева, броја новорегистрованих парцела, броја захтева за промену власништва итд. |
| Резултат | Грађанин/Радник у катастру је прегледао приказ статистика катастра. |
| Изузеци | - |

Табела 10 – Опис случаја коришћења “Приказ годишњих статистика катастра“

Табела 11 приказује опис случаја коришћења “Претрага приказа годишњих статистика катастра“.

|  |  |
| --- | --- |
| Назив | Претрага приказа годишњих статистика катастра |
| Учесници | Грађанин, Радник у катастру |
| Предуслови | Грађанину/Раднику у катастру је приказана страница са табеларним приказом статистика катастра за сваку годину рада. |
| Кораци | 1. Грађанин/радник у катастру уноси годину.  2. Грађанин/радник у катастру потврђује унос. |
| Резултат | Грађанин/Радник у катастру је претражио податке о статистици катастра за унесену годину. |
| Изузеци | За унесену годину не постоји статистика катастра. |

Табела 11 – Опис случаја коришћења “Претрага приказа годишњих статистика катастра“

Табела 12 приказује опис случаја коришћења “Сортирање приказа годишњих статистика катастра“.

|  |  |
| --- | --- |
| Назив | Сортирање приказа годишњих статистика катастра |
| Учесници | Грађанин, Радник у катастру |
| Предуслови | Грађанину/Раднику у катастру је приказана страница са табеларним приказом статистика катастра за сваку годину рада. |
| Кораци | 1. Грађанин/радник у катастру бира колону по којој жели да сортира табеларни приказ статистика катастра. |
| Резултат | Грађанин/Радник у катастру је сортирао податке о статистици катастра по одабраној колони. |
| Изузеци | - |

Табела 12 – Опис случаја коришћења “Сортирање приказа годишњих статистика катастра“

Табела 13 приказује опис случаја коришћења “Експортовање годишње статистике катастра за неку годину као PDF“.

|  |  |
| --- | --- |
| Назив | Експортовање годишње статистике катастра за неку годину као PDF |
| Учесници | Грађанин, Радник у катастру |
| Предуслови | Грађанину/Раднику у катастру је приказана страница са табеларним приказом статистика катастра за сваку годину рада. |
| Кораци | 1. Грађанин/радник у катастру бира одговарајућу годину чије податке жели да експортује као PDF. |
| Резултат | Грађанин/Радник у катастру је експортовао податке за одабрану годину у PDF формату. |
| Изузеци | - |

Табела 13 – Опис случаја коришћења “Експортовање годишње статистике катастра за неку годину као PDF“

Табела 14 приказује опис случаја коришћења “Преглед вести везаних за геодетски завод“.

|  |  |
| --- | --- |
| Назив | Преглед вести везаних за геодетски завод |
| Учесници | Грађанин, Радник у катастру |
| Предуслови | Грађанин/Радник у катастру је пријављен на систем. |
| Кораци | 1. Грађанин/радник у катастру бира опцију за преглед вести везаних за геодетски завод. |
| Резултат | Грађанин/Радник у катастру је прегледао вести везане за геодетски завод. |
| Изузеци | - |

Табела 14 – Опис случаја коришћења “Преглед вести везаних за геодетски завод“

Табела 15 приказује опис случаја коришћења “Упутство за утврђивање тачног броја парцеле на којој се налази нека непокретност“.

|  |  |
| --- | --- |
| Назив | Упутство за утврђивање тачног броја парцеле на којој се налази нека непокретност |
| Учесници | Грађанин, Радник у катастру |
| Предуслови | Грађанин/Радник у катастру је пријављен на систем. |
| Кораци | 1. Грађанин/Радник у катастру бира опцију за преглед упутства за утврђивање тачног броја парцеле. |
| Резултат | Грађанин/Радник у катастру је прегледао упутство за утврђивање тачног броја парцеле. |
| Изузеци | - |

Табела 15 – Опис случаја коришћења “Упутство за утврђивање тачног броја парцеле на којој се налази нека непокретност“

Табела 16 приказује опис случаја коришћења “Додавање информација о новој промени података над неком непокретношћу“.

|  |  |
| --- | --- |
| Назив | Додавање информација о новој промени података над неком непокретношћу |
| Учесници | Радник у катастру |
| Предуслови | Радник у катастру је пријављен на систем. |
| Кораци | 1. Радник у катастру бира опцију за додавање информација о новој промени података над неком непокретношћу.  2. Радник у катастру уноси датум промене, име и презиме старог власника, име и презиме новог власника и непокретност за коју је везана да промена.  3. Радник у катастру потврђује унос. |
| Резултат | Радник у катастру је додао информације о новој промени података над неком непокретношћу. |
| Изузеци | Погрешна непокретност, стари или нови власник. |

Табела 16 – Опис случаја коришћења “Додавање информација о новој промени података над неком непокретношћу“

Табела 17 приказује опис случаја коришћења “Приказ историје промена података над неком непокретношћу“.

|  |  |
| --- | --- |
| Назив | Приказ историје промена података над неком непокретношћу |
| Учесници | Радник у катастру |
| Предуслови | Радник у катастру је пријављен на систем. |
| Кораци | 1. Радник у катастру бира опцију за приказ историје промена података над неком непокретношћу.  2. Радник у катастру уноси број парцеле за коју жели да погледа историјат промена.  3. Радник у катастру потврђује унос. |
| Резултат | Радник у катастру је прегледао историјат промена података над непокретношћу чији је број парцеле унео. |
| Изузеци | Не постоји унесени број парцеле. |

Табела 17 – Опис случаја коришћења “Приказ историје промена података над неком непокретношћу“

Табела 18 приказује опис случаја коришћења “Додавање података о новој непокретности“.

|  |  |
| --- | --- |
| Назив | Додавање података о новој непокретности |
| Учесници | Радник у катастру |
| Предуслови | Радник у катастру је пријављен на систем. |
| Кораци | 1. Радник у катастру бира опцију за додавање података о новој непокретности.  2. Радник у катастру уноси адресу, број парцеле, површину, потес и начин коришћења земљишта.  3. Радник у катастру потврђује унос. |
| Резултат | Радник у катастру је додао податке о новој непокретности. |
| Изузеци | Непокретност са унетим бројем парцеле већ постоји у бази података. |

Табела 18 – Опис случаја коришћења “Додавање података о новој непокретности“

Табела 19 приказује опис случаја коришћења “Измена података о непокретности“.

|  |  |
| --- | --- |
| Назив | Измена података о непокретности |
| Учесници | Радник у катастру |
| Предуслови | Радник у катастру је пријављен на систем. |
| Кораци | 1. Радник у катастру бира опцију за измену података о непокретности.  2. Радник у катастру уноси број парцеле чије податке жели да мења, затим име и презиме старог и име и презиме новог власника.  3. Радник у катастру потврђује унос. |
| Резултат | Радник у катастру је изменио податке о непокретности. |
| Изузеци | - |

Табела 19 – Опис случаја коришћења “Измена података о непокретности“

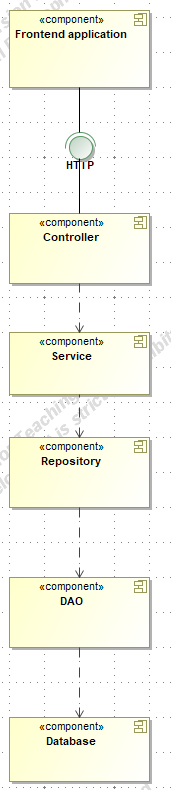
## Спецификација нефункционалних захтева

У овом одељку су описани нефункционални захтеви које је потребно да испуњава софтверско решење информационог система катастра. Апликација подржава следеће нефункционалне захтеве:

* портабилност, односно циљне платформе и прилагодљив дизајн. То је остварено кроз коришћење Boostrap [20] компоненти, које омогућавају једноставно прилагођавање изгледа апликације различитим уређајима и ширинама уређаја.
* компатибилност са форматима датотека за export. То је остварено кроз могућност корисника експортују и преузимају податке у стандардизованом и широко прихваћеном формату као што је PDF.
* компатибилност са другим системима (интероперабилност). То је остварено кроз повезаност са информационим системима матичара и нотара да би се дигитализовао и аутоматизовао процес преноса власништва над земљиштем. Тиме је омогућена ефикасна размена података између наведених система.
* одржавање система. То је остварено кроз добре праксе програмирања и тиме се омогућава лакше одржавање и проширивост система без нарушавања постојећих функционалности.
* употребљивост. То је остварено кроз интуитиван кориснички интерфејс и јасну навигацију кроз апликацију, чиме се корисницима омогућава да могу ефикасно и једноствано користити апликацију.
* безбедност, односно заштита података. Технологије које су коришћене за реализацију софтверског решења подразумевано нуде неке важне безбедносне механизме. На пример, Angular има безбедносне механизме који аутоматски, без додатних подешавања, пружају подршку за заштиту од SQL Injection [21] и XSS [22] напада у пољима за унос текста.

# Спецификација дизајна

Ово поглавље објашњава дизајн софтверског решења за информациони систем катастра. Дизајн софтверског решења је представљен помоћу UML дијаграма компоненти, као што је приказано на слици 3.

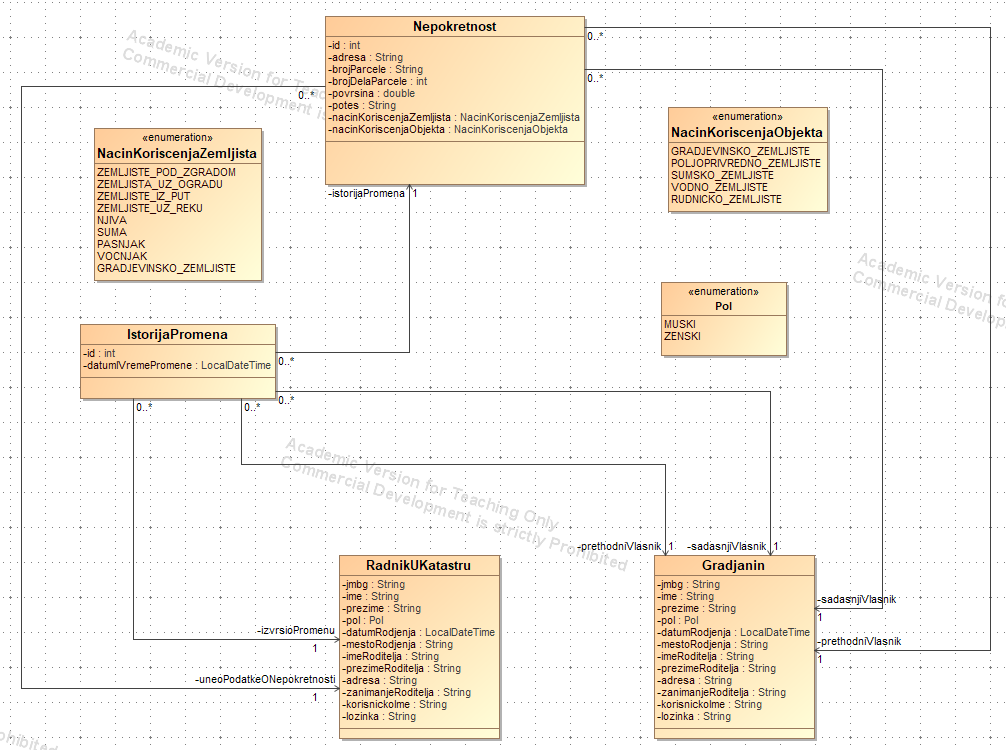


Слика 3 – Приказ UML дијаграма компоненти

Frontend апликација је компонента која представља клијентску апликацију информационог система катастра и омогућава корисницима да ступе у интеракцију са системом. Серверски део апликације се састоји од више компоненти. То су контролери, сервиси, репозиторијуми и DAO слој. База података је представљена посебном компонентом по имену Database.

Frontend апликација и контролери комуницирају везом путем HTTP [23] интерфејса, преко које frontend апликација може да шаље захтеве за одређеним акцијама и добија одговоре. Контролер компонента је зависна од сервисног слоја који садржи главну пословну логику система, док је сервисни слој зависан од репозиторијума који је одговоран за манипулацију подацима смештеним у бази података. Репозиторијуми су зависни од DAO (Data Access Object) слоја који пружа приступ подацима у бази података. Такође, DAO слој је зависан од базе података јер треба да приступа подацима који су смештени у бази података да би их проследио репозиторијуму, овај сервисном слоју, а овај контролеру да би клијентска апликација могла да врати одговор кориснику система.

На слици 4 је помоћу UML дијаграма класа представљен објектни модел система.



Слика 4 – Дијаграм класа

Класа RadnikUKatastru репрезентује радника који ради у катастру и садржи његове личне податке и податке за аутентификацију. Класа Gradjanin репрезентује грађанина Републике Србије и садржи његове личне податке и податке за аутентификацију. Подаци о непокретностима су моделовани класом Nepokretnost, а подаци о историјама промена над неком непокретношћу класом IstorijaPromena.

Кардиналитети релација између класа Nepokretnost и Gradjanin омогућавају једној непокретности да има тачно једнок садашњег и тачно једног претходног власника (под “претходни власник“ се мисли само на власника који је пре садашњег власника поседовао ту непокретност, а не на листу свих претходних власника који су поседовали ту непокретност током историје), а да један грађанин може да има више непокретности чији је садашњи или претходни власник.

Кардиналитет релације између класа Nepokretnost и RadnikUKatastru омогућава једној непокретности да има тачно једног катастарског радника који је унео податке о непокретности у систем, а да радник у катастру може да има више непокретности чије је податке унео у систем.

Кардиналитети релација између класа IstorijaPromena и Gradjanin омогућава једној историји промене да садржи податке о тачно једном садашњем и тачном једном претходном власнику (под “претходни власник“ се мисли само на власника који је пре садашњег власника поседовао ту непокретност, не на листу свих претходних власника који су поседовали ту непокретност током историје), а да један грађанин може да има више историја промена где је уписан као садашњи или претходни власник неке непокретности.

Кардиналитет релације између класа IstorijaPromena и RadnikUKatastru омогућава једној историји промена да има тачно једног катастарског радника који је креирао ту историју промена, а да један радник у катастру може да име више историја промена које је унео у систем.

Кардиналитет релације између класа Nepokretnost и IstorijaPromena омогућава једној историји промена да има тачно једну непокретност на коју се она односи, а да једна непокретност може да има више историја промена над њоме.

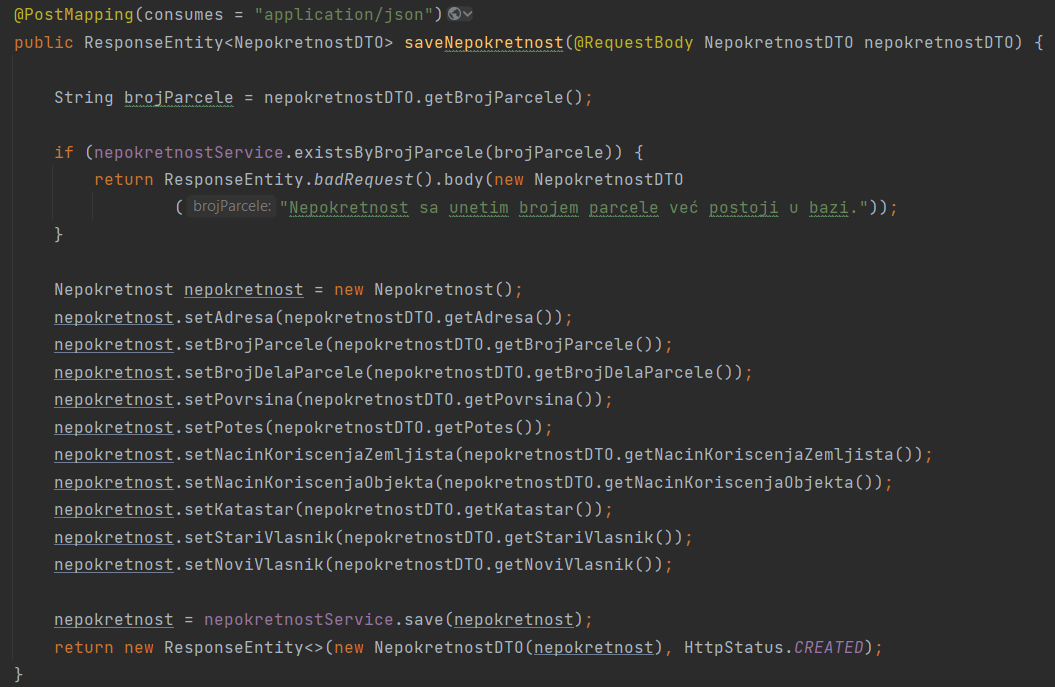
# Имплементација

Ово поглавље приказује начин на који су имплементиране неке од функција информационог система катастра. Објашњена је имплементација додавања података о новој непокретности, ажурирање података о непокретности, претраге базе података катастра непокретности, приказа историје промена података над неком непокретношћу и додавања података о новој историји промена над непокретоношћу. Имплементација наведених функционалности је објашњена и на серверском и на клијентском делу апликације.

* 1. ***Имплементација серверског дела апликације***

Овај одељак садржи начин који су имплементиране неке од функција информационог система катастра на серверском делу апликације.

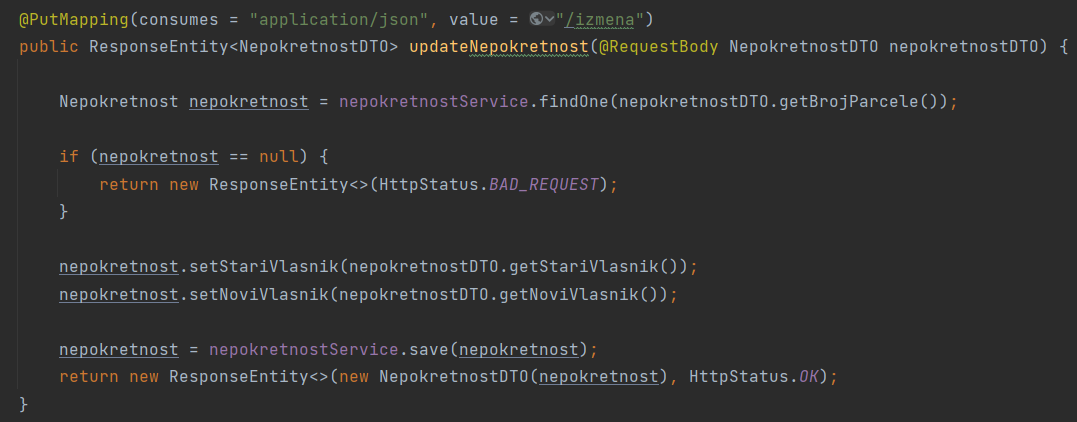
За додавање података о новој непокретности имплементирана је метода у једном контролеру приказана на листингу 1.



Листинг 1 – Додавање података о новој непокретности (серверски код)

Ова метода са именом saveNepokretnost() има @PostMapping анотацију која указује да је у питању HTTP POST захтев. Овој методи се прослеђује објекат који репрезентује конкретну непокретност, то јест DTO (Data Transfer Object) за конкретну непокретност. У телу методе се прво проверава да ли постоји непокретност са унетим бројем парцеле позивом методе existsByBrojParcele(brojParcele) чија се имплементација налази у класи NepokretnostService. Ако постоји, прави се HTTP одговор са статусом 400 (Bad request) и поруком која садржи информације о грешци. Ако непокретност са унетим бројем парцеле не постоји, креира се нова инстанца објекта класе Nepokretnost и попуњавају се његови атрибути вредностима (уносе се адреса, број парцеле, број дела парцеле, површина, потес, начин коришћења земљишта, начин коришћења објекта, име катастарског радника који уноси непокретност, име и презиме старог и новог власника) и то на основу вредности који се добијају из објекта класе NepokretnostDTO. Након тога се позива save() метода из класе NepokretnostService да би се сачували подаци о новој непокретности у базу података. После се креира објекат класе ResponseEntity<> који враћа одговор са статусом која означава да су успешно сачувани подаци о новој непокретности.

За ажурирање података о непокретности имплементирана је метода у једном контролеру приказана на листингу 2.



Листинг 2 – Ажурирање података о непокретности (серверски код)

Ова метода са називом updateNepokretnost() има @PutMapping анотацију која указује да је у питању HTTP PUT захтев. Овој методи се прослеђује објекат који репрезентује конкретну непокретност, то јест DTO (Data Transfer Object) за конкретну непокретност. У телу методе се прво добавља постојећа непокретност по броју парцеле позивом методе findOne() чија имплементација се налази у класи NepokretnostService. Затим се проверава да ли подаци о добављеној непокретности по броју парцеле имају null вредност. Ако имају, то значи да непокретност са унетим бројем парцеле не постоји у бази података и враћа се објекат класе ResponseEntity са статусом 400 (Bad Request). Ако добављени подаци о непокретности постоје у бази података, ажурирају се атрибути који репрезентују старог и новог власника на основу вредности из објекта класе NepokretnostDTO. Затим се позива save() метода из класе NepokretnostService да би се сачувале промене у бази података и на крају се креира објекат класе ResponseEntity<> који враћа одговор са статусом 200 о успешном ажурирању података о непокретности.

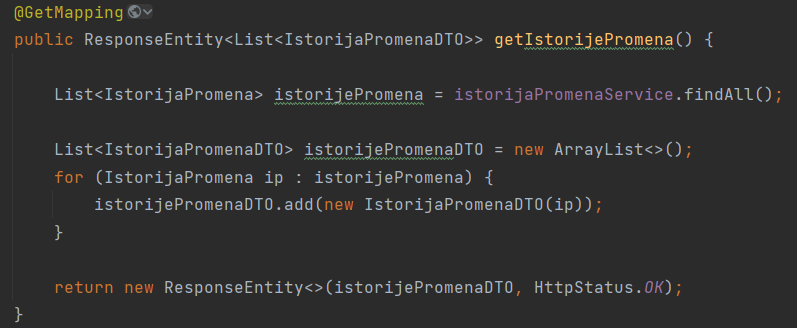
За претрагу базе података катастра непокретности имплементирана је метода у једном контролеру приказана на листингу 3.



Листинг 3 – Претрага базе података катастра непокретности (серверски код)

Ова метода са називом getNepokretnost() има @GetMapping анотацију која указује да је у питању HTTP GET захтев. Прво се позива метода findAll() чија се имплементација налази у класи NepokretnostService и која враћа листу свих непокретности који постоје у бази података. Затим се креира празна листа која садржи објекте класе NepokretnostDTO и они репрезентују DTO (Data Transfer Object) за конкретне непокретности. После тога, итерацијама кроз листу свих непокретности се за сваку непокретност креира нови објекат типа NepokretnostDTO помоћу конструктора те класе. На крају се креира објекат класе ResponseEntity<> који враћа одговор са статусом 200 (OK).

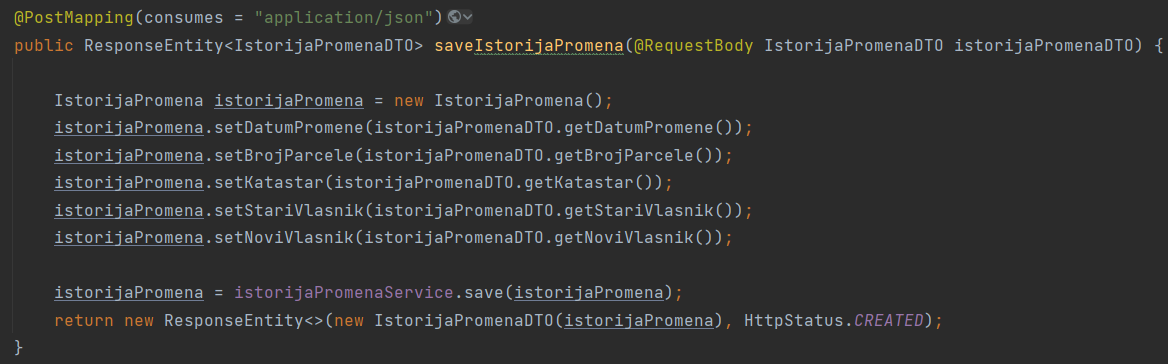
За приказ историје промена над неком непокретношћу имплементирана је метода у једном контролеру приказана на листингу 4.



Листинг 4 – Приказ историје промена над неком непокретношћу (серверски код)

Ова метода са називом getIstorijePromena() има @GetMapping анотацију која указује да је у питању HTTP GET захтев. Прво се позива метода findAll() чија се имплементација налази у класи IstorijaPromenaService и која враћа листу свих историја промена над свим непокретностима који постоје у бази података. Затим се креира празна листа која садржи објекте класе IstorijaPromenaDTO и они репрезентују DTO (Data Transfer Object) за конкретне податке о историјама промена. После тога, итерацијама кроз листу свих историја промена се за сваку историју промене креира нови објекат типа IstorijaPromenaDTO помоћу конструктора те класе. На крају се креира објекат класе ResponseEntity<> који враћа одговор са статусом 200 (OK).

За додавање података о новој историји промена над неком непокретношћу имплементирана је метода у једном контролеру приказана на листингу 5.

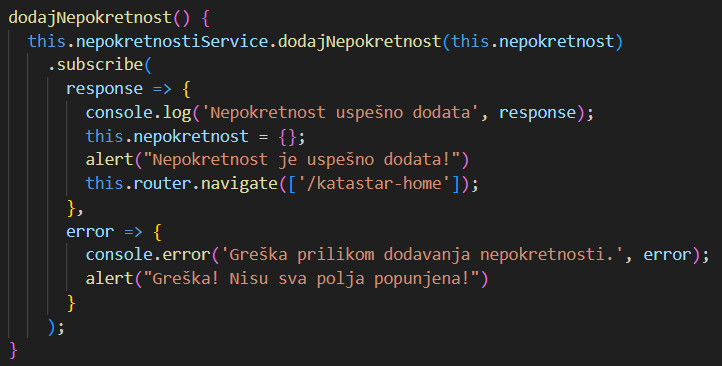


Листинг 5 – Додавање података о новој историји промена над неком непокретношћу (серверски код)

Ова метода са именом saveIstorijaPromena() има @PostMapping анотацију која указује да је у питању HTTP POST захтев. Овој методи се прослеђује објекат који репрезентује конкретну историју промене, то јест DTO (Data Transfer Object) за конкретну историју промене. У телу методе се креира нова инстанца класе IstorijaPromena и попуњавају се његови атрибути вредностима (уносе се датум промене, број парцеле, име катастарског радника који уноси непокретност, име и презиме старог и новог власника) и то на основу вредности који се добијају из објекта класе IstorijaPromenaDTO. Након тога се позива save() метода из класе IstorijaPromenaService да би се сачували подаци о новој историји промена у базу података. После се креира објекат класе ResponseEntity<> који враћа одговор са статусом која означава да су успешно сачувани подаци о новој историји промене.

* 1. ***Имплементација клијентског дела апликације***

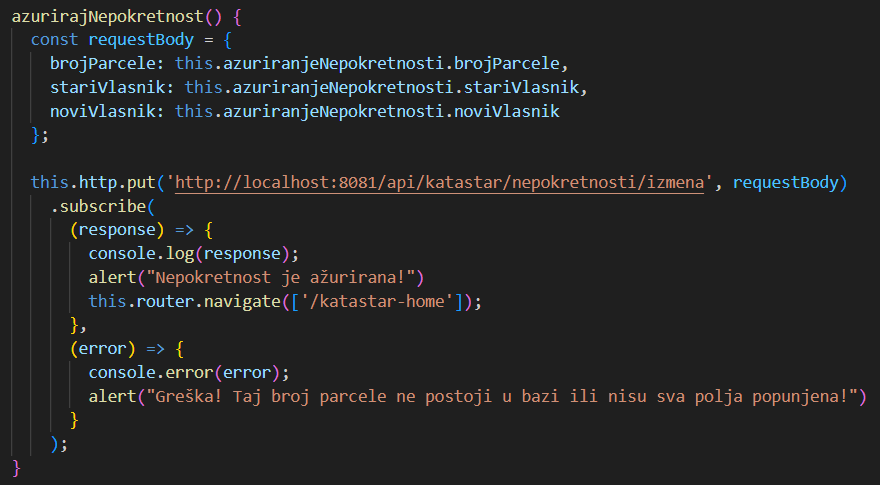
За додавање података о новој непокретности, на основу кода у серверском делу апликације који је приказан на листингу 1, имплементирана је метода у једном TypeScript фајлу приказана на листингу 6.



Листинг 6 – Додавање података о новој непокретности (клијентски код)

У овој функцији под називом dodajNepokretnost() се прво позива истоимена функција dodajNepokretnost() чија имплементација се налази у Angular сервису по имену NepokretnostService. Та функција тог сервиса је одговорна за слање HTTP захтева на сервер ради додавања података о новој непокретности и позива се са аргументом која је глобална варијабла са именом nepokretnost и која је иницијализована на празну листу. Када сервер одговори на захтев онда се користи метода subscribe() над објектом класе Observable који се враћа из позване функције са именом dodajNepokretnost(). Преко subscribe() методе се региструје одговор сервера. Ако је одговор успешан, извршава се callback функција која је прослеђена као први аргумент методе subscribe(), што значи да ће се приказати порука "Nepokretnost uspešno dodata" у конзоли, а затим се ресетује варијабла nepokretnost на празну листу и приказује се alert порука. Затим се кориснику приказује страница /katastar-home помоћу Angular рутирања кроз компоненте. Са друге стране, ако се деси грешка приликом захтева онда се извршава callback функција која је прослеђена као други аргумент subscribe() методи, то јест приказује се порука о грешци и у конзоли и као alert порука.

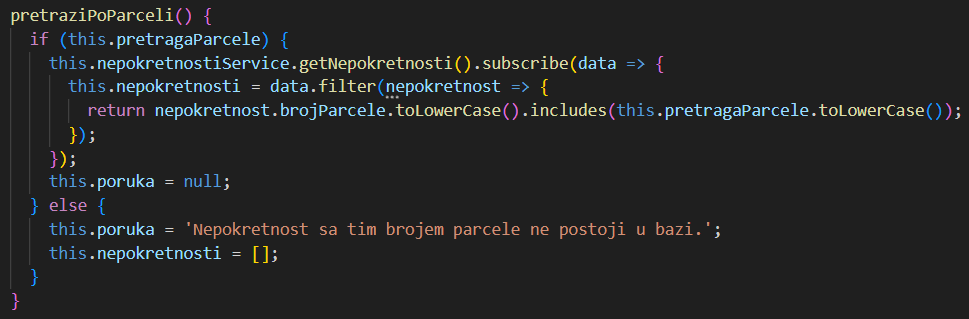
За ажурирање података о непокретности, на основу кода у серверском делу апликације који је приказан на листингу 2, имплементирана је метода у једном TypeScript фајлу приказана на листингу 7.



Листинг 7 – Ажурирање података о непокретности (клијентски код)

У овој функцији под називом azurirajNepokretnost() се прво декларише константа requestBody која садржи податке о непокретнисти које треба ажурирати. Ти подаци се преузимају из глобалне варијабле под називом azuriranjeNepokretnosti која је иницијализована на празну листу. Након тога користи се метода за слање HTTP PUT захтева ка серверз. Путања захтева је "http://localhost:8081/api/katastar/nepokretnosti/izmena", а претходно споменута константа requestBody се шаље на сервер као тело тог захтева. Функција subscribe() се користи за “претплату“ на резултате извршавања захтева. У случају успешног одговора од сервера, тај добијени одговор се исписује у конзоли и приказује се порука да су подаци о непокретности успешно ажурирани користећи alert, а затим се кориснику приказује страница /katastar-home помоћу Angular рутера. У случају да се десила нека грешка, она се исписује у конзоли и приказује се одговарајући текст кориснику путем alert поруке.

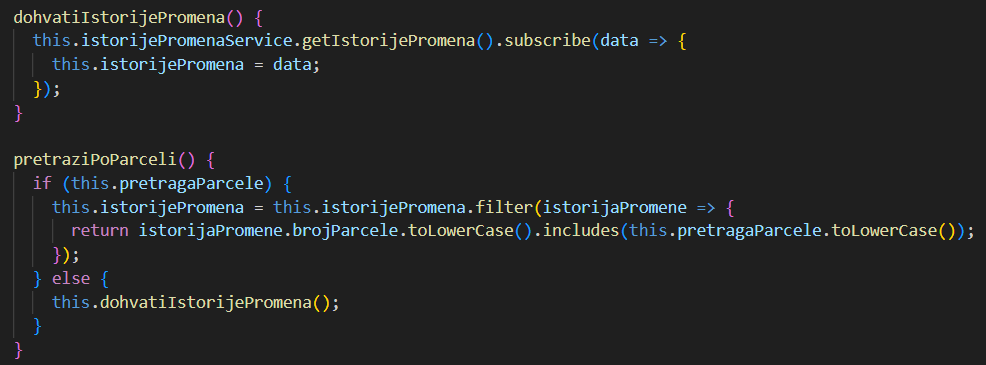
За претрагу базе података катастра непокретности, на основу кода у серверском делу апликације који је приказан на листингу 3, имплементирана је метода у једном TypeScript фајлу приказана на листингу 8.



Листинг 8 – Претрага базе података катастра непокретности (клијентски код)

У овој функцији под називом pretraziPoParceli() се прво у if исказу проверава вредност глобалне варијабле pretragaParcele која представља унету вредност за претрагу. Ако та вредност постоји онда се извршава блок кода унутар if исказа. Позива се функција getNepokretnosti() чија се имплементација налази у Angular сервису под називом NepokretnostService и помоћу ње се добија тражена непокретност са сервера. Функција getNepokretnosti() враћа објекат класе Observable, на који се ради “претплата“ помоћу subscribe() методе. Кад стигну подаци са сервера, варијабла data ће садржати податке о траженој непокретности. Подаци о непокретности се филтрирају коришћењем filter() методе преко које се за сваку непокретност проверава да ли вредност за број парцеле садржи унету вредност у пољу за претрагу. Филтрирана непокретност се додељује глобалној променљивој nepokretnosti која ће се приказати кориснику на корисничком интерфејсу. Ако вредност променљиве pretragaParcele из поља за претрагу не постоји у бази података, извршава се блок кода унутар else исказа. То значи да ће се приказати одговарајућа порука о грешци.

За приказ историје промена над неком непокретношћу, на основу кода у серверском делу апликације који је приказан на листингу 4, имплементиранe су две методе у једном TypeScript фајлу приказани на листингу 9.

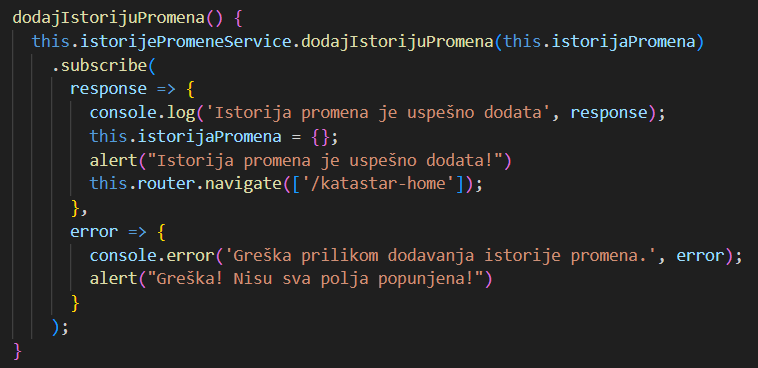


Листинг 9 – Приказ историје промена над неком непокретношћу (клијентски код)

У функцији под називом dohvatiIstorijePromena() позива се функција чија имплементација се налази у Angular сервису IstorijaPromenaService. Ова метода враћа објекат класе Observable који садржи листу свих историја промена над свим непокретностима. Кад сервер пошаље податке, параметар data у subscribe() методи ће садржати листу свих историја промена над свим непокретностима. Та листа се касније додељује глобалној променљивој istorijePromena и на тај начин ће се приказати резултати на корисничком интерфејсу.

У другој функцији, под називом pretraziPoParceli() се прво проверава вредност глобалне променљиве pretragaParcele која представља унету вредност у пољу за претрагу. Ако та вредност постоји, извршава се блок кода унутар if исказа. Листа историја промена се филтрира коришћењем filter() методе, где се за сваку историју промена проверава да ли вредност за број парцеле садржи унету вредност у пољу за претрагу. Касније се филтриране историје промена додељују глобалној променљивој istorijePromena и приказаће се резултат претраге на корисничком интерфејсу. Ако вредност у пољу за претрагу не постоји, извршава се блок кода унутар else исказа. Тада се позива функција dohvatiIstorijePromena() да би се добиле подаци о свим историјама промена над свим непокретностима са сервера и ажурирала листа у глобалној променљивој istorijePromena.

За додавање података о новој историји промена над неком непокретношћу, на основу кода у серверском делу апликације који је приказан на листингу 5, имплементирана је метода у једном TypeScript фајлу приказана на листингу 10.



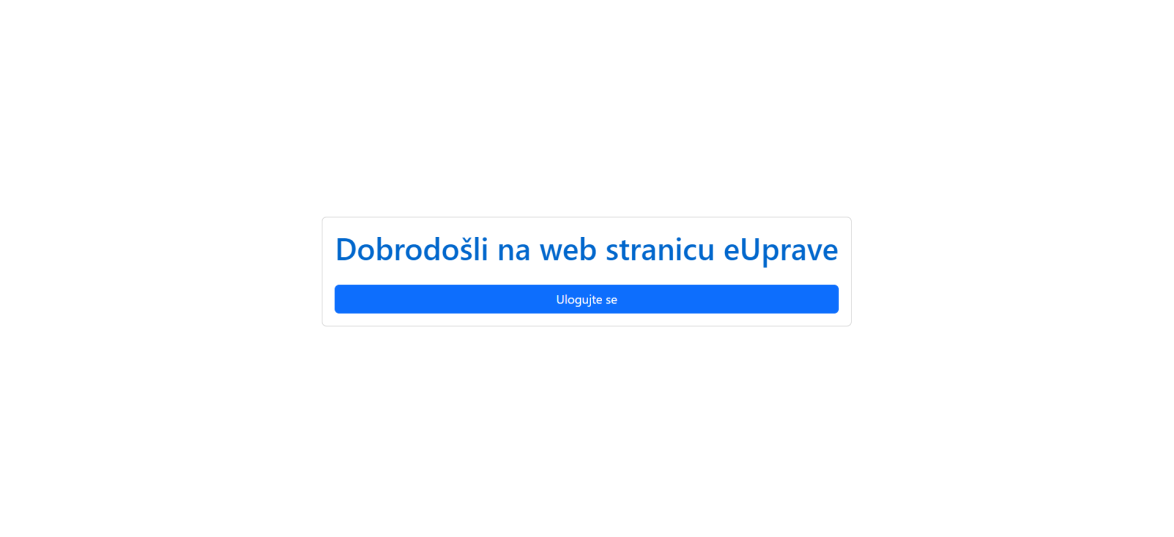
Листинг 10 – Додавање података о новој историји промена над неком непокретношћу (клијентски код)

У функцији под називом dodajIstorijuPromena() прво се позива метода dodajIstorijuPromena() чија се имплементација налази у Angular сервису IstorijePromeneService и прослеђује јој се глобална варијабла istorijaPromena као тело захтева. Резултат позива функције се добија помоћу subscribe() методе позване над објектом класе Observable. Ова метода прима две callback функције. Прва функција се извршава када се након успешног захтева добије одговор од сервера. Тада response параметар садржи одговор са сервера. Друга функција се извршава ако се догоди нека грешка приликом слања захтева на сервер. У том случају error параметар садржи информације о грешци. Ако је захтев успешан, исписује се одговарајућа порука кориснику у конзоли, а глобална варијабла istorijaPromena се ресетује на празну листу како би се очистили унети подаци и омогућило додавање података о новој историји промена. На крају се кориснику приказује alert порука и позива се Angular рутер који ће кориснику приказати /katastar-home страницу. Уколико захтев није успешан, прво се исписује порука о грешци у конзоли, па се приказује одговарајућа alert порука како би корисник добио повратну информацију о грешци.

# Демонстрација

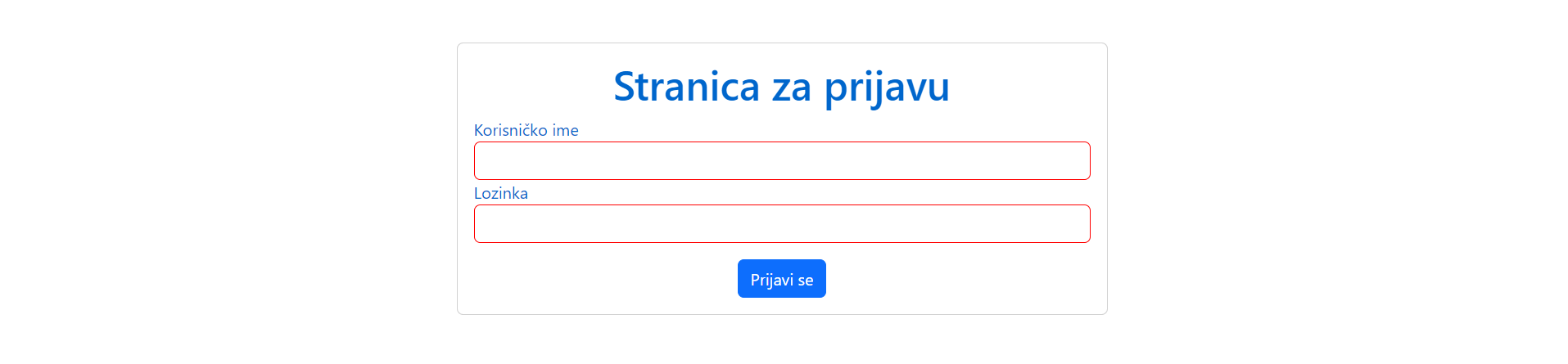
Ово поглавље приказује начин коришћења софтверског решења за информациони систем катастра.

Приликом покретања апликације кориснику се приказује прозор као што је илустровано на слици 3.



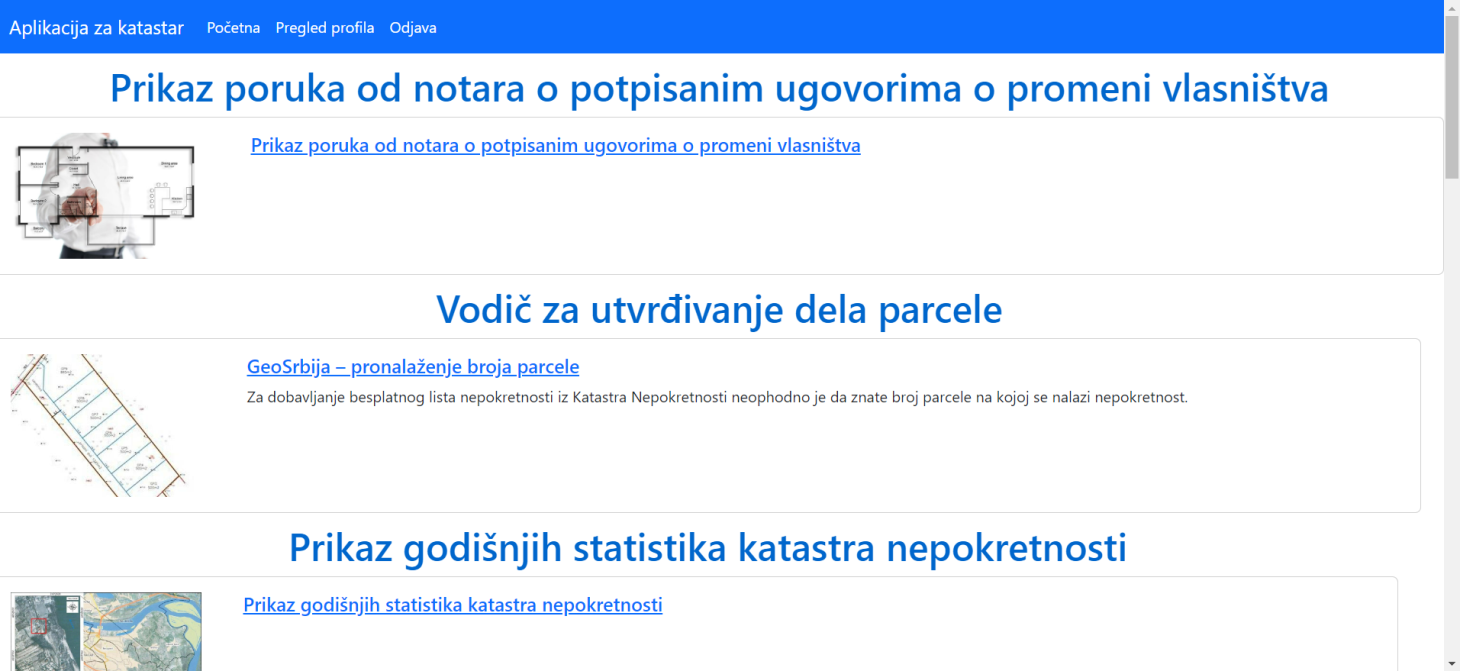
Слика 3 – Изглед прозора који се приказује приликом покретања апликације

У централном делу прозора се налази порука добродошлице и дугме који корисник може изабрати да би му се приказала форма за пријаву на систем. Приказ форме за пријаву на систем је илустрована на слици 4.



Слика 4 – Приказ форме за пријаву на систем

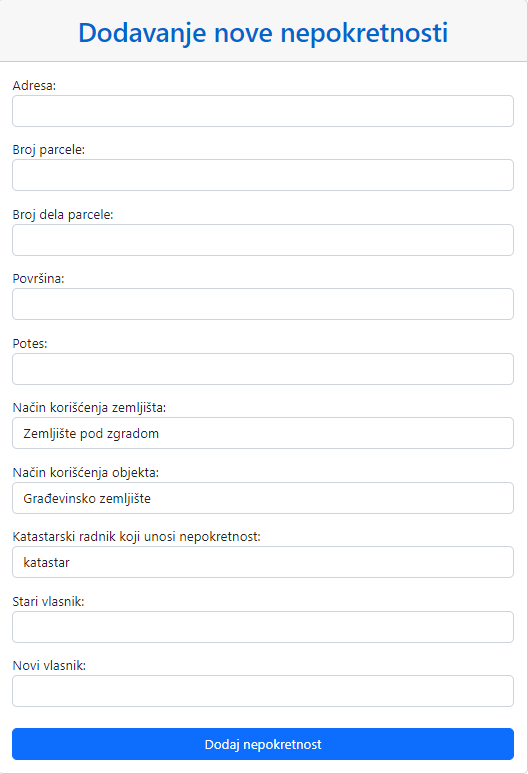
Форма за пријаву на систем садржи поља за унос корисничког имена и лозинке, а у доњем делу форме је дугме којим корисник потврђује унос. Уколико су унети тачни креденцијали за аутентификацију, кориснику ће се приказати почетна страница апликације у зависности од улоге која је кориснику додељена у систему (катастарски радник или грађанин). Уколико корисник има додељену улогу да је катастарски радник, њему ће се након успешне пријаве на систем приказати почетна страница као што је илустровано на слици 5.



Слика 5 – Изглед почетне странице катастарског радника

У горњем делу странице се налази трака за навигацију са опцијама за приказ почетне странице, преглед профила и одјаву са система. Остатак странице садржи приказ опција које катастарски радник може да извршава у складу са функционалностима које може да обавља, а које су приказане на слици 2 помоћу UML дијаграма случајева коришћења.

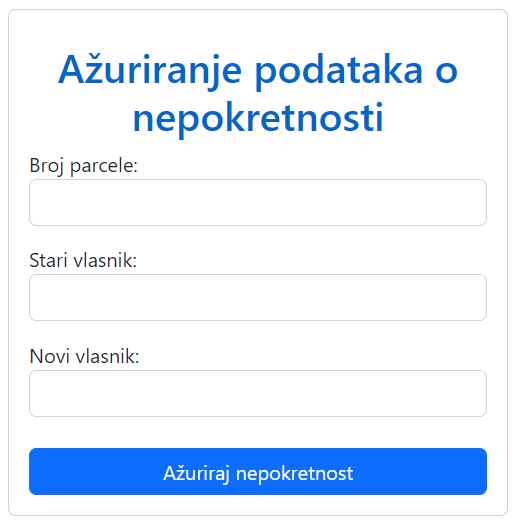
Уколико на почетној страници катастарски радник одабере опцију за додавање података о новој непокретности, њему се приказује форма као што је илустровано на слици 6.



Слика 6 – Изглед форме за унос података о новој непокретности

Унутар форме за унос података о новој непокретности налазе се поља за унос одговарајућих података о новој непокретности, као што су унос адресе, броја парцеле, броја дела парцеле, површине, потеса, начина коришћења земљишта, начина коришћења објекта, као и унос имена и презимена новог и старог власника. У доњем делу форме је дугме којим катастарски радник може да потврди унос и, уколико су одговарајуће валидације за свако поље успешни, сачува податке о новој непокретности.

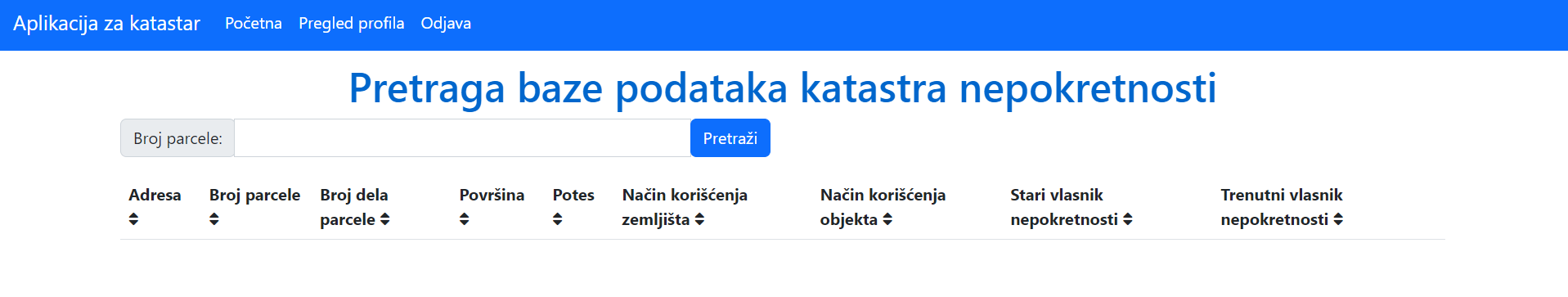
Уколико на почетној страници катастарски радник одабере опцију за ажурирање података о непокретности, њему се приказује форма као што је илустровано на слици 7.



Слика 7 – Изглед форме за ажурирање података о непокретности

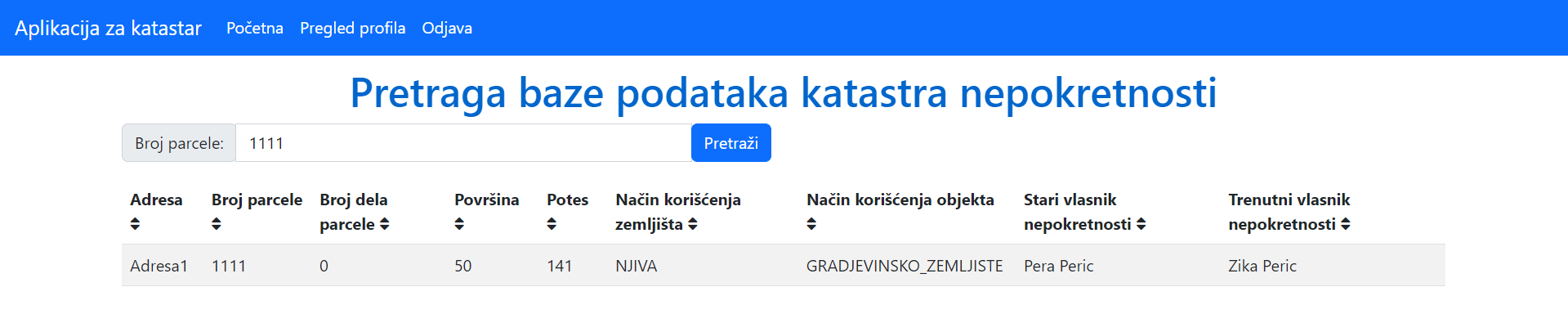
Унутар форме за ажурирање података о непокретности налазе се поља за унос одговарајућих података о непокретности, као што су унос броја парцеле непокретности чији се подаци ажурирају, као и унос имена и презимена новог и старог власника. У доњем делу форме је дугме којим катастарски радник може да потврди унос и, уколико су одговарајуће валидације за свако поље успешнаи, ажурира податке о непокретности.

Уколико на почетној страници катастарски радник одабере опцију за претрагу базе података катастра непокретности, њему се приказује страница као што је илустровано на слици 8.



Слика 8 – Изглед странице за претрагу базе података катастра непокретности

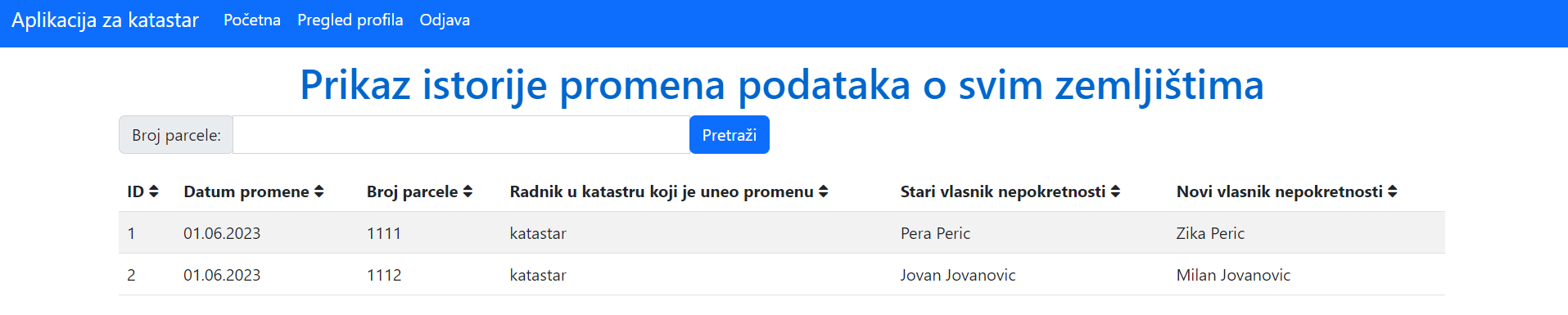
У горњем делу странице се налази трака за навигацију са опцијама за приказ почетне странице, преглед профила и одјаву са система. У централном делу странице је поље за унос броја парцеле за претрагу, а у доњем делу странице је празна табела са заглављем. Уколико катастарски радник унесе у поље за претрагу неки број парцеле непокретности која постоји у бази података, подаци о траженој непокретности се приказују катастарском раднику на начин на који је приказано на слици 9.



Слика 9 - Изглед странице за претрагу базе података катастра непокретности са конкретним претраженим подацима

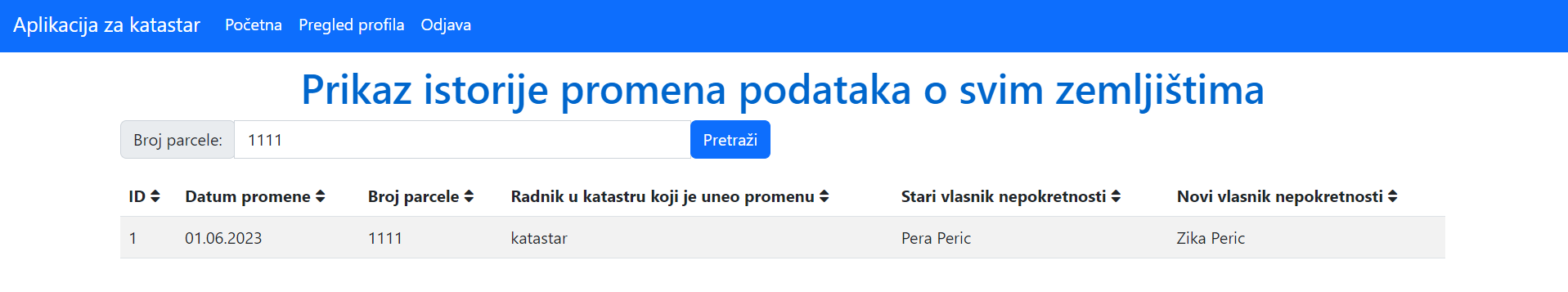
У централном делу странице је у поље за претрагу по броју парцеле унета вредност “1111“ и потврђен унос одабиром опције за претрагу и подаци из базе података о непокретности се приказују катастарском раднику на начин као што је приказано у табели у доњем делу странице.

Уколико на почетној страници катастарски радник одабере опцију за приказ историје промена података над неком непокретношћу, њему се приказује страница као што је илустровано на слици 10.



Слика 10 – Изглед странице за приказ историје промена података над неком непокретношћу

У горњем делу странице се налази трака за навигацију са опцијама за приказ почетне странице, преглед профила и одјаву са система. У централном делу странице је поље за унос броја парцеле по ком се приказају историје промена само над непокретношћу са унесеним бројем парцеле, а у доњем делу странице је табела у којој се приказују све историје промена над свим непокретностима. Уколико катастарски радник унесе у поље за претрагу неки број парцеле непокретности која постоји у бази података, подаци о историјама промена над траженом непокретношћу се приказују катастарском раднику на начин како је приказано на слици 11.



Слика 11 - Изглед странице за приказ историја промена података са конкретним претраженим подацима

У централном делу странице је у поље за претрагу по броју парцеле унета вредност “1111“ и потврђен унос одабиром опције за претрагу и подаци из базе података о историјама промена над непокретношћу се приказују катастарском раднику на начин као што је приказано у табели у доњем делу странице.

Уколико на почетној страници катастарски радник одабере опцију за додавање података о новој историји промене над неком непокретношћу, њему се приказује форма као што је илустровано на слици 12.



Слика 12 - Изглед форме за додавање података о новој историји промене над неком непокретношћу

Унутар форме за додавање података о новој историји промене над неком непокретношћу налазе се поља за унос одговарајућих података о новој историји промене, као што су унос броја парцеле непокретности чији се подаци ажурирају, као и унос имена и презимена новог и старог власника. У доњем делу форме је дугме којим катастарски радник може да потврди унос и, уколико су одговарајуће валидације за свако поље успешни, сачува податке о новој историји промене над неком непокретношћу.

# Закључак

У овом раду је представљено софтверско решење за информациони систем катастра и демонстриран је рад прототипске апликације. Приказано решење олакшава катастарским радницима да воде евиденцију о непокретностима, а у исто време омогућава грађанима да не морају да иду по шалтерима и чекају да би добили неке папире о непокретносима или да би обавили неке процедуре везане за катастар.

Систем нуди повећану ефикасност, транспарентност и аутоматизацију у управљању подацима о непокретностима, могућност праћења историје промена података над појединачним непокретностима, као и интероперабилност са информационим системима матичара и нотара при преносу власништва над неким земљиштем. Омогућен је једноставан приступ подацима и повећана поузданост током вођења евиденција о непокретностима. Овим системом нису обухваћене неке функције, као на пример:

* пружање неких информативних података важних за катастар, као што су списак катастарских општина или приказ контакт података путем којих грађани могу да контактирају катастарски завод путем мејла или телефона. Ове могућности би се могле реализовати додавањем нових клијентских компоненти који ће садржати статичке податке о списку катастарских општина или контакт подацима, или да се ти подаци чувају у новим табелама у бази података катастра, па да се у серверском коду искодира пословна логика преко које ће се подаци динамички учитавати у новододате компоненте.
* преглед важних катастарских докумената попут неких закона, уредби, правилника, директива итд. и могућност њиховог експортовања у неком формату. Те могућности би се могле реализовати чувањем тих докумената у неком формату на неком екстерном месту попут Google drive-а [24] или Firebase-a [25] и увезивањаем истих у одговарајуће делове клијентских компоненти.
* додатне online услуге, попут еШалтера, еЗаказивања, еПримедби које би се могле реализовати без коришћења додатних платформи или библиотека, него кодирањем додатне пословне логике у серверском коду и повезивањем те логике са клијентским кодом.
* могућност локализације клијентске апликације да би, осим српског језика, подржао и друге језике попут енглеског, немачког и других. То би се могло реализовати коришћењем интернационализације садржаја.
* претрага базе података катастра непокретности би могла да подржи претрагу и по другим кротеријумима, попут адресе, потеса итд. или могућност комбиноване претраге по више критеријума. То би се могло реализовати коришћењем технологија попут ElasticSearch-a. [26]

Систем приказан у овом раду, у поређењу са сродним решењима, пружа једноставност при коришћењу, а истовремено обезбеђује све потребне функционалности. С обзиром на то да у развоју већина анализираних решења учествују тимови инжењера, очекивано је да она поседује извесне предности у односу на решење представљено у овом раду.

Коришћење система је омогућено путем веб апликације креиране за Windows[27] оперативни систем. Даљим усавршавањем овог решења би могла да се омогући подршка за мобилне платформе, као што су Android[28] и IoS[29], како би оно било доступно што већем броју корисника. У томе може помоћи коришћење радних оквира као што су Flutter[30], React Native[31] и други.

# Литература

[1] StackPath, LLC. 2023. What is a Web Application?. Преузето 30.06.2023. са

<https://www.stackpath.com/edge-academy/what-is-a-web-application/>

[2] VMware, Inc. or its affiliates. 2023. Spring Boot. Преузето 30.06.2023. са <https://spring.io/projects/spring-boot>

[3] Google. 2023. Angular. Преузето 30.06.2023. са <https://angular.io/>

[4] Oracle. 2023. MySQL. Преузето 30.06.2023. са <https://www.mysql.com/>

[5] Републички геодетски завод. 2023. Републички геодетски завод. Преузето 30.06.2023. са <https://www.rgz.gov.rs/>

[6] Adobe. 2023. What is a PDF? Portable Document Format | Adobe Acrobat. Преузето 30.06.2023. са <https://www.adobe.com/acrobat/about-adobe-pdf.html>

[7] Red Hat, Inc. 2019. What is open source?. Преузето 30.06.2023. са <https://www.redhat.com/en/topics/open-source/what-is-open-source>

[8] Oracle. 2023. Java | Oracle. Преузето 30.06.2023. са <https://www.java.com/en/>

[9] Chris Richardson. 2023. What are microservices?. Преузето 30.06.2023. са <https://microservices.io/>

[10] abhishekaslk. 2023. What is Standalone Application? – GeeksforGeeks. Преузето 30.06.2023. са <https://www.geeksforgeeks.org/what-is-standalone-application/>

[11] Microsoft. 2023. TypeScript: JavaScript With Syntax For Types. Преузето 30.06.2023. са <https://www.typescriptlang.org/>

[12] syedmodassirali. 2022. Client-Server Model – GeeksforGeeks. Преузето 30.06.2023. са <https://www.geeksforgeeks.org/client-server-model/>

[13] solid IT gmbh. 2023. DB-Engines Ranking. Преузето 30.06.2023. са

<https://db-engines.com/en/ranking>

[14] Oracle. 2023. Database | Oracle. Преузето 30.06.2023. са <https://www.oracle.com/database/>

[15] Meta. 2023. Facebook – log in or sign up. Преузето 30.06.2023. са <https://www.facebook.com/>

[16] X Corp. 2023. Explore / Twitter. Преузето 30.06.2023. са <https://twitter.com/>

[17] Oracle. 2023. What is a Relational Database | Oracle. Преузето 30.06.2023. са <https://www.oracle.com/database/what-is-a-relational-database/>

[18] javatpoint. 2021. What is an Embedded system? Definition and Tutorial – javatpoint. Преузето 30.06.2023. са <https://www.javatpoint.com/embedded-system-tutorial>

[19] Object Management Group®, Inc. 2023. Welcome To UML Web Site!. Преузето 30.06.2023. са <http://www.uml.org/>

[20] Bootstrap team and contributors. 2023. Bootstrap – The most popular HTML, CSS and JS library in the world. Преузето 30.06.2023. са <https://getbootstrap.com/>

[21] PortSwigger Ltd. 2023. What is SQL Injection? Tutorial & Examples | Web Security Academy. Преузето 30.06.2023. са <https://portswigger.net/web-security/sql-injection>

[22] Synopsys, Inc. 2023. What Is Cross Site Scripting (XSS) and How Does It Works | Synopsys. Преузето 30.06.2023. са <https://www.synopsys.com/glossary/what-is-cross-site-scripting.html>

[23] Refsnes Data. 2023. What is HTTP. Преузето 30.06.2023. са <https://www.w3schools.com/whatis/whatis_http.asp>

[24] Google. 2023. Personal Cloud Storage & File Sharing Platform – Google. Преузето 30.06.2023. са <https://www.google.com/drive/>

[25] Google. 2023. Firebase. Преузето 30.06.2023. са <https://firebase.google.com/>

[26] Elasticsearch B.V. 2023. Elasticsearch Platform — Find real-time answers at scale | Elastic. Преузето 30.06.2023. са <https://www.elastic.co/>

[27] Microsoft. 2023. Experience the Power of Windows 11 OS, Computers, & Apps | Microsoft. Преузето 30.06.2023. са <https://www.microsoft.com/en-us/windows>

[28] Google. 2023. Android - Secure & Reliable Mobile Operating System. Преузето 30.06.2023. са <https://www.android.com/>

[29] Apple Inc. 2023. Apple. Преузето 30.06.2023. са <https://www.apple.com/ios>

[30] Google. 2023. Flutter - Build apps for any screen. Преузето 30.06.2023. са <https://flutter.dev/>

[31] Meta Platforms, Inc. 2023. React Native · Learn once, write anywhere. Преузето 30.06.2023. са <https://reactnative.dev/>