ВИСОКА ТЕХНИЧКА ШКОЛА

СТРУКОВНИХ СТУДИЈА У НОВОМ САДУ

СМЕР: ЕЛЕКТРОТЕХНИЧКИ

СТУДИЈСКИ ПРОГРАМ: ИНФОРМАЦИОНЕ ТЕХНОЛОГИЈЕ

ПРЕДМЕТ: СОФТВЕРСКО ИНЖЕЊЕРСТВО

**АПЛИКАЦИЈА ЗА ЗАКАЗИВАЊЕ ТЕРМИНА У ПАНСИОНУ ЗА КУЋНЕ ЉУБИМЦЕ - *PETKEEPER***

Нови Сад, јун 2019. године

**САДРЖАЈ**

[1. Увод 3](#_Toc41254596)

[1.1. Сажетак 3](#_Toc41254597)

[1.2. Намена апликације 3](#_Toc41254598)

[1.3. Претпоставке 3](#_Toc41254599)

[2. Циљ рада 4](#_Toc41254600)

[2.1. Функционална апликација 4](#_Toc41254601)

[2.2. Тимски рад 4](#_Toc41254602)

[2.3. Контрола верзије 4](#_Toc41254603)

[2.4. Дотнет и јаваскрипт 4](#_Toc41254604)

[2.5. Израда пројектне документације 4](#_Toc41254605)

[3. предуслови и припремне радње за развој апликације 5](#_Toc41254606)

[3.1. Идентификација заинтересованих страна 5](#_Toc41254607)

[3.2. Захтеви система 5](#_Toc41254608)

[3.2.1. Функционалност веб апликације 6](#_Toc41254609)

[3.2.2. Минимални системски захтеви 6](#_Toc41254610)

[3.2.3. Минимални хардверски захтеви 6](#_Toc41254611)

[3.3. Категорије корисника 7](#_Toc41254612)

[3.3.1. Администратор 7](#_Toc41254613)

[3.3.2. Регистровани корисник 7](#_Toc41254614)

[4. имплементација 7](#_Toc41254615)

[4.1. Случајеви коришћења 7](#_Toc41254616)

[4.1.1. Случајеви коришћења корисника са улогом администратора 8](#_Toc41254617)

[4.1.2. Случајеви коришћења корисника са улогом регистрованог корисника 9](#_Toc41254618)

[4.2. Дијаграм класа 10](#_Toc41254619)

[4.3. Дијаграм активности 11](#_Toc41254620)

[4.4. Шема релационе базе података 12](#_Toc41254621)

[5. Тестирање 12](#_Toc41254622)

[5.1. Списак функионалних захтева 12](#_Toc41254623)

[5.2. Тестирање функционалности интерфејса 13](#_Toc41254624)

[6. Дизајн 14](#_Toc41254625)

[7. Могућа проширења 15](#_Toc41254626)

[8. Апликација у продукцији 15](#_Toc41254627)

[9. Активности тима на изради пројекта 16](#_Toc41254628)

[9.1. Тим 16](#_Toc41254629)

[9.2. Предуслови за израду пројекта 16](#_Toc41254630)

[9.3. Комуникација у току израде пројекта 16](#_Toc41254631)

[9.4. Управљање пројектом 16](#_Toc41254632)

[9.5. Контрола верзије 19](#_Toc41254633)

[9.6. Анализа рада и одлука 19](#_Toc41254634)

[9.6.1. Моделовање апликације 19](#_Toc41254635)

[9.6.2. Контрола верзије 19](#_Toc41254636)

[9.6.3. Комуникација у оквиру тима 19](#_Toc41254637)

[9.6.4. Управљање пројектом 20](#_Toc41254638)

[9.6.5. Програмски оквир 20](#_Toc41254639)

[10. Закључак 20](#_Toc41254640)

# Увод

## Сажетак

Пројекат *Pet Keeper* је предиспитна обавеза из предмета „Софтверско инжењерство. Настао је као потреба да се кроз практични рад имплементирају знања стечена у току теоријске наставе и да студенти стеккну искуство о проласку кроз све фазе пројектовања и израде софтверске апликације у оквиру тима.

## Намена апликације

Апликација је намењена предузетницима из области пружања пансионских услуга за кућне љубимце. Трендови у овој области су такви да приступ ресурсима треба омогућити већем броју корисника оваквих услуга путем *online* креирања резервација и на тај начин повећати број потенцијалних клијената.

## Претпоставке

Анализом постојећих веб презентација у РС домену дошло се до закључка да тренутно не постоји могућност заказивања термина у пансиону путем интернета. Из тог разлога, одлучено је да се направи веб апликација која ће омогућити потенцијалним корисницима ову врсту услуге и на тај начин проширити и побољшати пословање фирме која овакву услугу нуди на тржишту.

# Циљ рада

Циљ рада обухвата више засебних целина:

## Функционална апликација

Крајњи производ ће бити функионална апликација која ће корисницима омогућити заказивање термина у пансиону за кућне љубимце путем интернета.

## Тимски рад

Организовање тима за израду пројекта који се састоји од 4 члана био је предуслов за почетак израде апликације. Кроз овакав начин рада, студенти ће се упознати са расподелом и организацијом послова у оквиру групе.

## Контрола верзије

Како ће пројекат бити резултат тимског рада, постоји потреба за контролом верзије израде апликације. Кроз активности на GITHUB сервису студенти ће се упознати и савладати основе контроле верзирања и стећи вредно искуство у овој врсти активности која је постала стандард приликом израде било које врсте пројекта.

## Дотнет и јаваскрипт

Пројекат ће бити израђен у ДОТНЕТ технологији при чему ће студенти моћи да се упознају и савладају основе коришћења објектно-оријентисаног програмирања при изради веб апликације у клијент-сервер укружењу. У пројекту ће се такође користити и јаваскрипт као незаобилазна компонента сваке веб апликације.

## Израда пројектне документације

Сви чланови тима ће учествовати у изради пројектне документације која је обавезни део у изради пројекта.

# предуслови и припремне радње за развој апликације

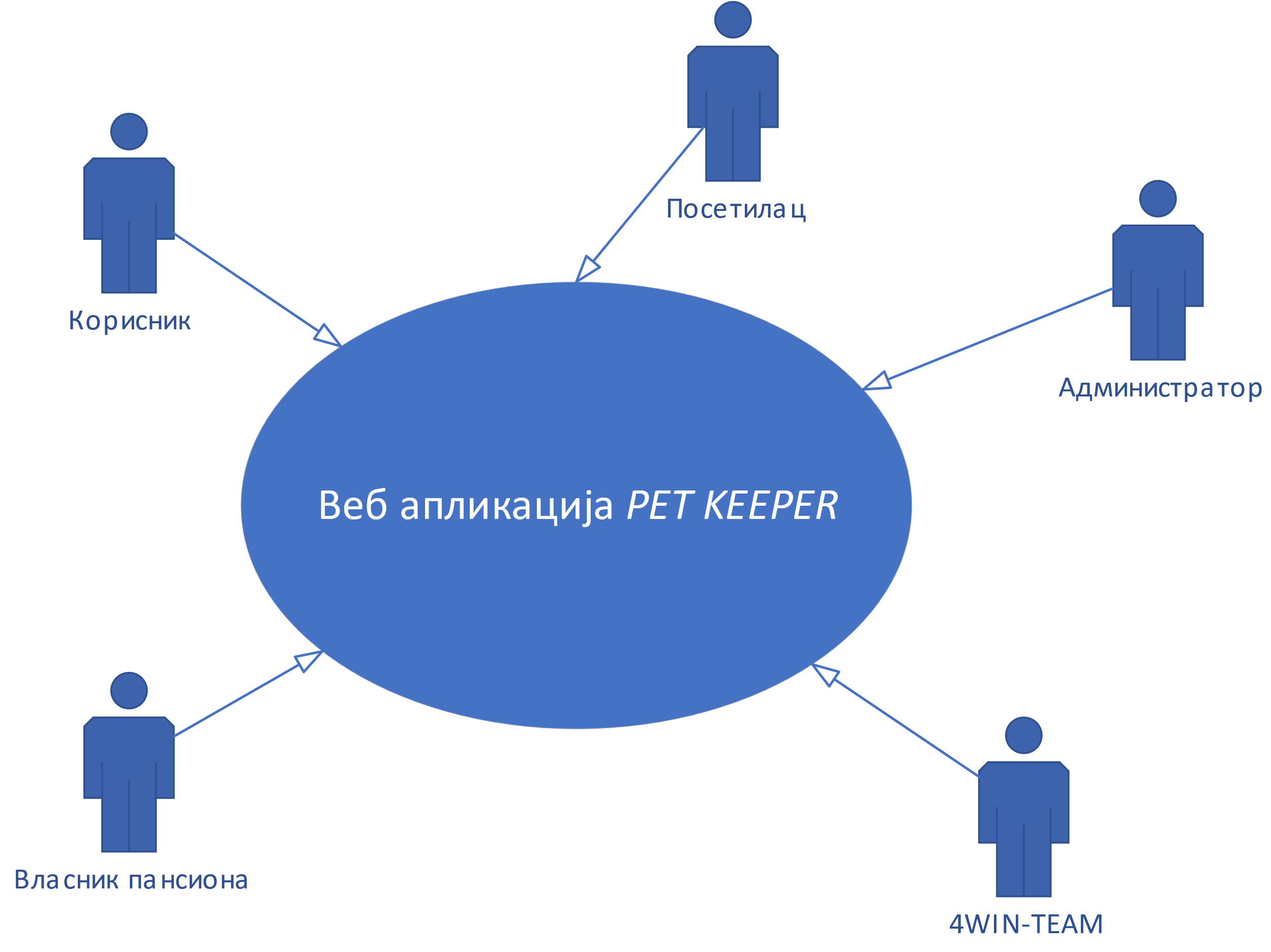
У овом поглављу биће речи о испуњавању софтверских и хардверских предуслова као и о припремним радњама које је потребно извршити пре приступања развоју апликације.

## Идентификација заинтересованих страна

Заинтересоване стране у пројекту подразумевају:

* Свако ко управља системом
* Свако ко има користи од система
* Свако ко директно или индиректно учествује у куповини система
* Организација задужена за дизајн и израду система

На слици 1. су приказане заинтересоване стране,



Слика 1. Заинтересоване стране

## Захтеви система

Након разговора са заинтересованим странама дефинисане су следеће компоненте система:

* Функционалност веб апликације
* Минимални системски захтеви
* Минимални хардверски захтеви

### Функционалност веб апликације

#### Веб презентација

Веб презентација подразумева:

* увид у основне информације о пансиону (насловна страница)
* контакт форму за евентуална питања и сугестије (контакт)
* кратко представљање оснивача пансиона (о нама)

#### Кориснички интерфејс

Кориснички интерфејс подразумева два типа:

* **Интерфејс за регистроване кориснике** – могућност да корисник креира и мења резервације
* **Интерфејс за администратора** – могућност прегледа, измене и брисања свих резервација у систему, управљање улогама корисника итд.

О функционалности веб апликације биће више речи у поглављу 4 – Имплементација.

### Минимални системски захтеви

#### Клијентска страна

Системски захтеви са клијентске стране подразумевају функционални веб претраживач:

* *Firefox*
* *Google Chrome*
* *Safari*
* *IE*
* …

#### Серверска страна

Системски захтеви са серверске стране подразумевају следеће:

* Оперативни систем – минимум Windows7
* DotNet окружење – минимум верзија 2.2
* MSSQL база података – минимум верзија 2008
* Веб сервер – минимум Apache 2.4 или IIS 2017???

### Минимални хардверски захтеви

#### Клијентска страна

Минимални хардверски захтеви са клијентске стране подразумевају следеће:

* Процесор Интел или АМД новије генерације
* 2 *GB* РАМ меморије

#### Серверска страна

* 64бит ИНТЕЛ или АМД процесор новије генерације
* 4*GB* РАМ меморије
* 100*GB* слободног простора на хард диску

## Категорије корисника

### Администратор

Администратор треба да има следеће могућности:

* Дефинише нове улоге
* Мења улоге корисника
* Има увид у све резервације
* Могућност да врши измене у резервацијама
* Могућност да брише резервације
* Могућност да поставља статус резервације

### Регистровани корисник

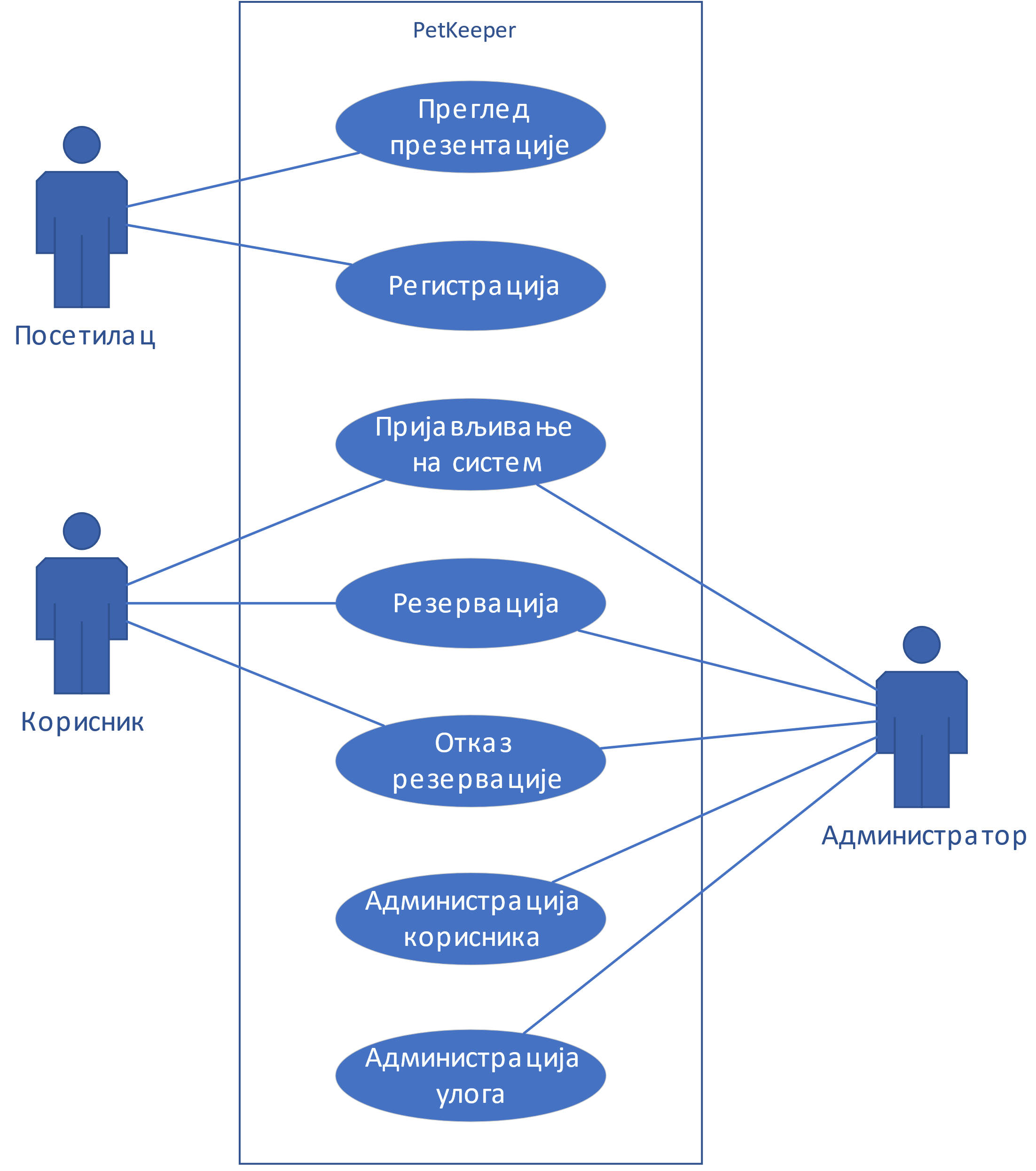
Регистровани корисник има следеће могућности:

* Може да креира резервацију
* Може да прави измене у резервацији у зависности од статуса резервације
* Може да брише резервацију у зависности од статуса резервације
* Може да види детаље резервације

# имплементација

## Случајеви коришћења

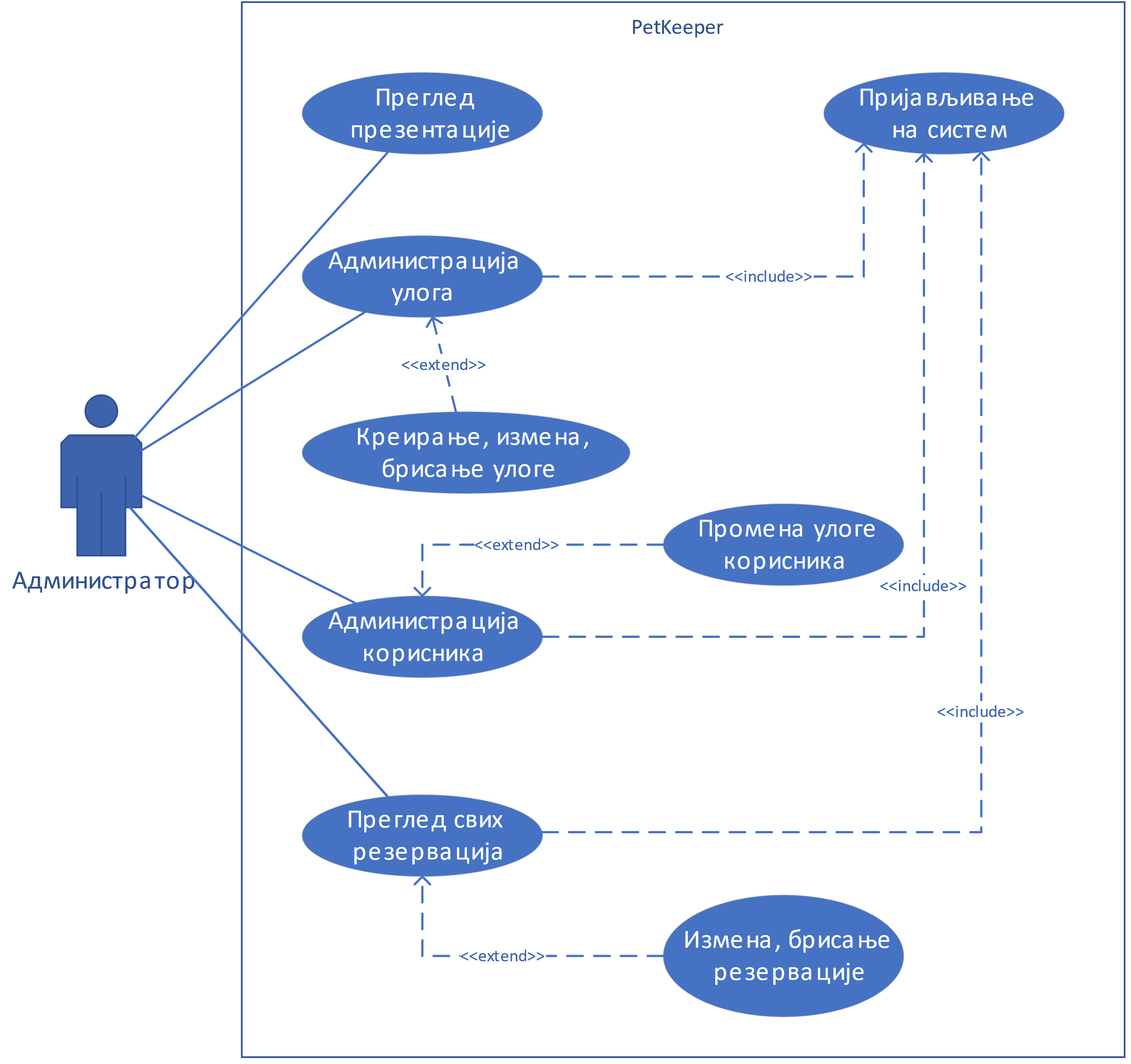
У овом поглављу су приказани случајеви коришћења у зависности од улоге корисника. На слици 2 приказан је дијаграм случаја коришћења високог нивоа апстрактности где су приказани актери система.



Слика 2.

### Случајеви коришћења корисника са улогом администратора

На слици 3 приказан је случај коришћења актера са улогом администратора.



Слика 3.

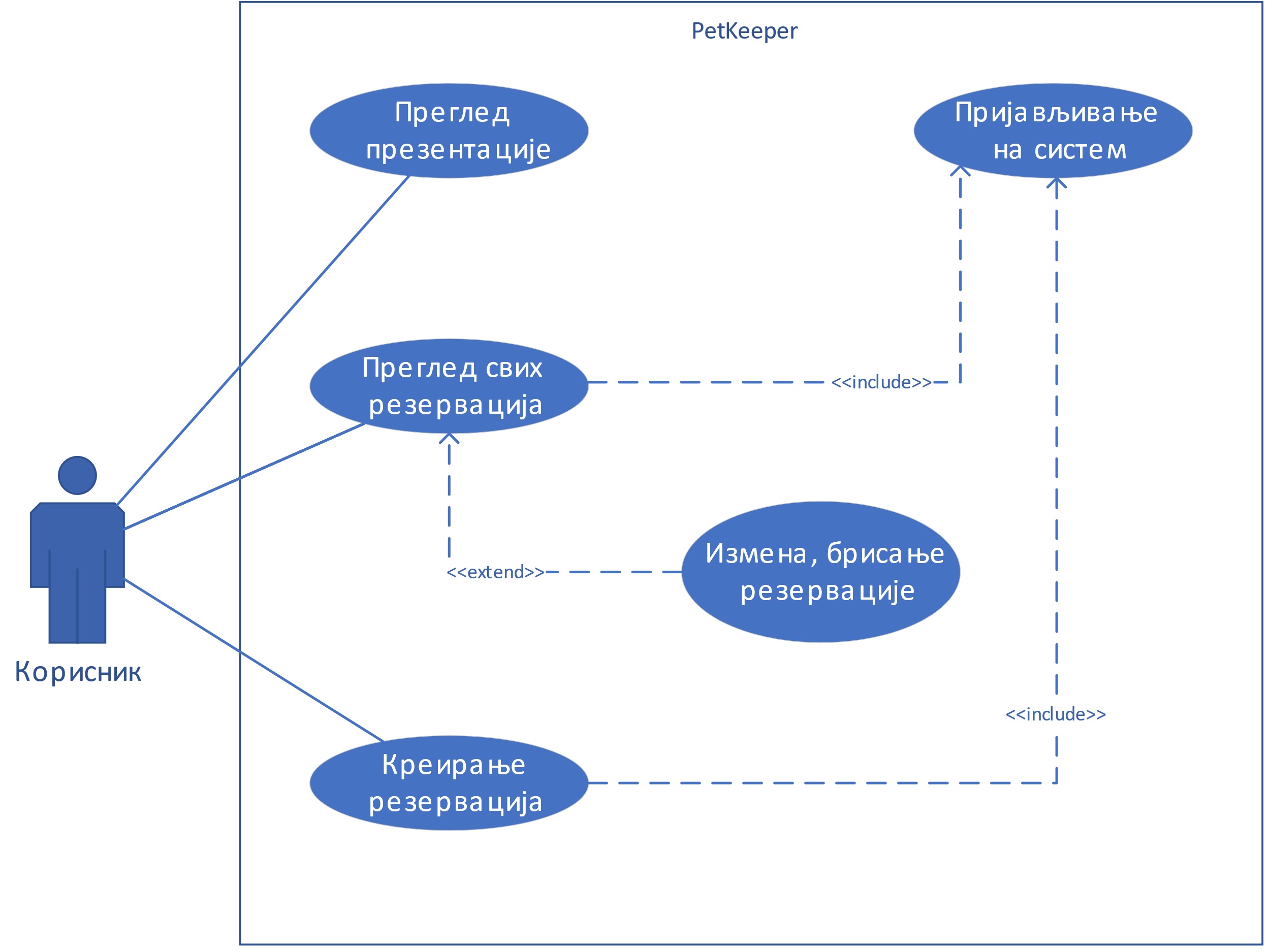
У табели 1 приказани су случајеви коришћења актера који има административне привилегије, услов и резултат.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Случај коришћења | Опис | Актер | Услов | Резултат |
| СК01 | Преглед презентације | Администратор | Стартована апликација | Администратор може да прегледа све странице презентације |
| СК02 | Администрација улога | Администратор | Успешно пријављивање | Администратор може да креира, мења и брише улоге |
| СК03 | Администрација корисника | Администратор | Успешно пријављивање | Администратор може да мења улоге корисника |
| СК04 | Преглед свих резервација | Администратор | Успешно пријављивање | Администратор кроз табеларни приказ може да мења и брише резервације |

Табела 1.

### Случајеви коришћења корисника са улогом регистрованог корисника

На слици 4 приказан је случај коришћења актера са улогом корисника.



Слика 4

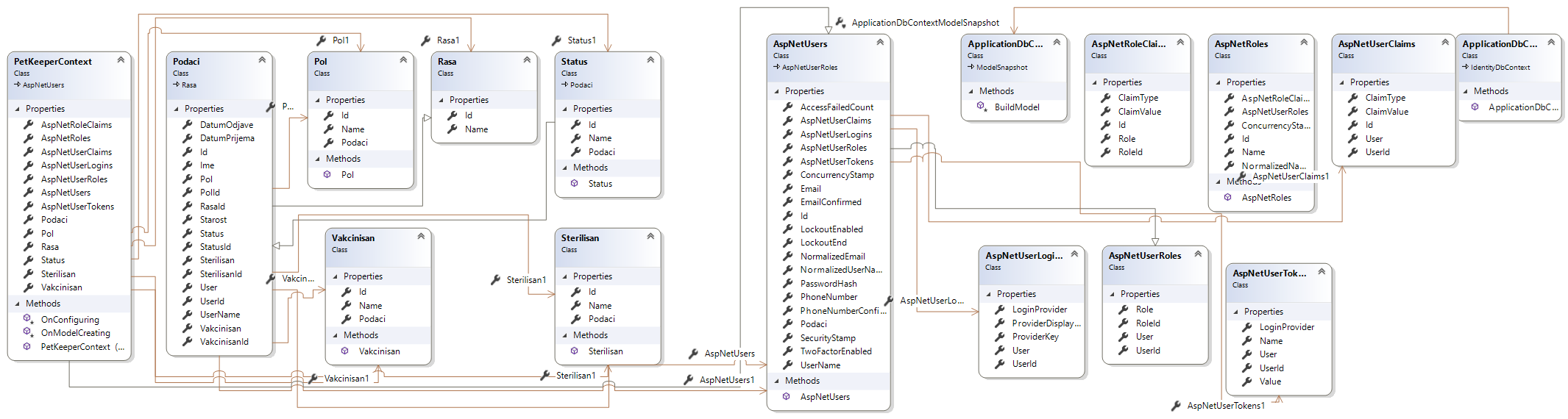
У табели 2 приказани су случајеви коришћења актера који има административне привилегије, услов и резултат.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Случај коришћења | Опис | Актер | Услов | Резултат |
| СК05 | Преглед презентације | Корисник | Стартована апликација | Корисник може да прегледа све странице презентације |
| СК06 | Регистрација корисника | Корисник | Стартована апликација | Корисник може да се региструје на систем |
| СК07 | Преглед резервација направљене од стране пријављеног корисника | Корисник | Успешно пријављивање | Корисник може да мења и брише резервације у зависности од статуса резервације |
| СК08 | Креирање резервације | Корисник | Успешно пријављивање | Корисник може да креира резервацију |

Табела 2.

## Дијаграм класа

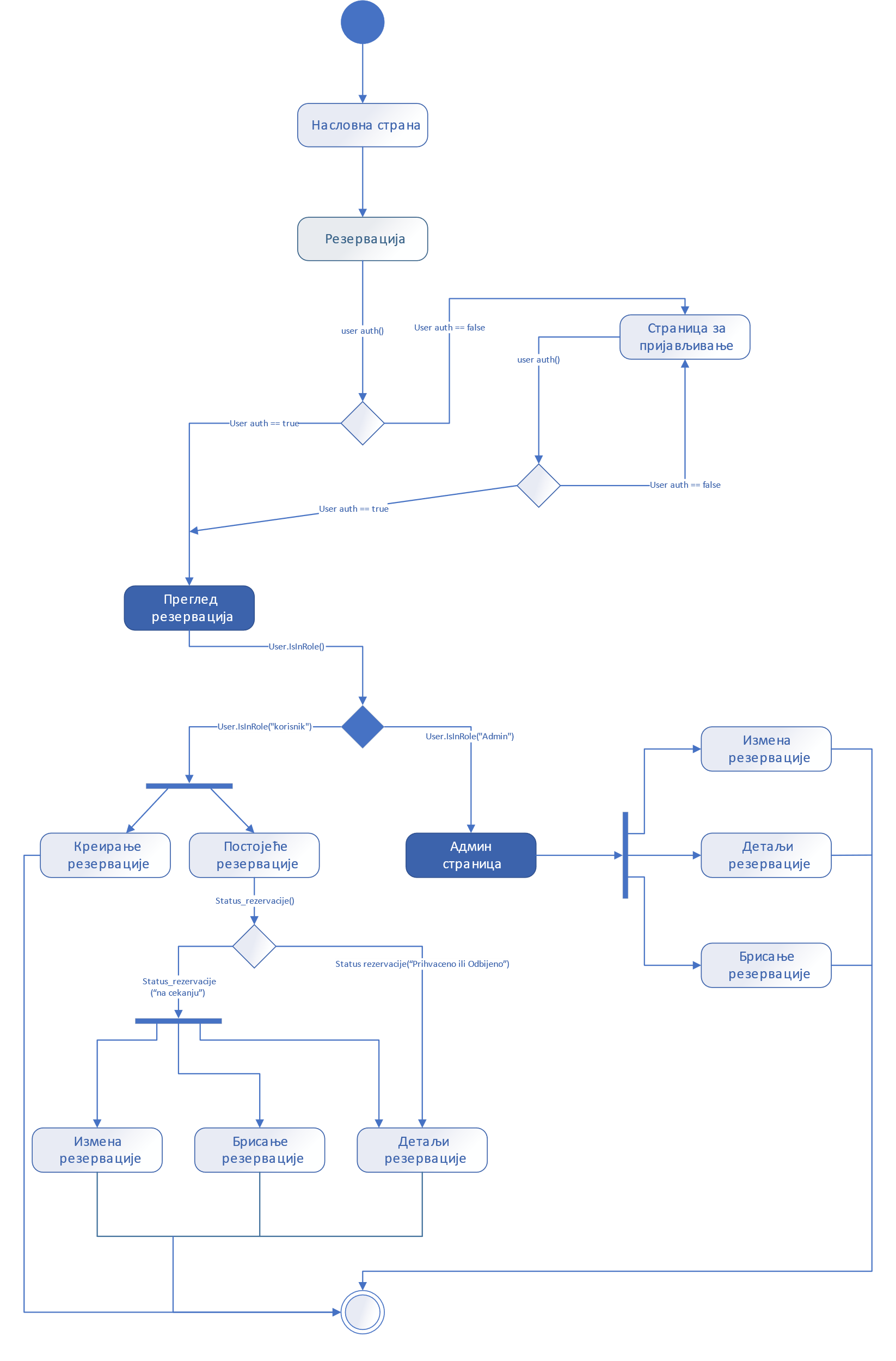
На слици 5 приказане су класе и њихова међусобна повезаност и итеракција



Слика 5.

## Дијаграм активности

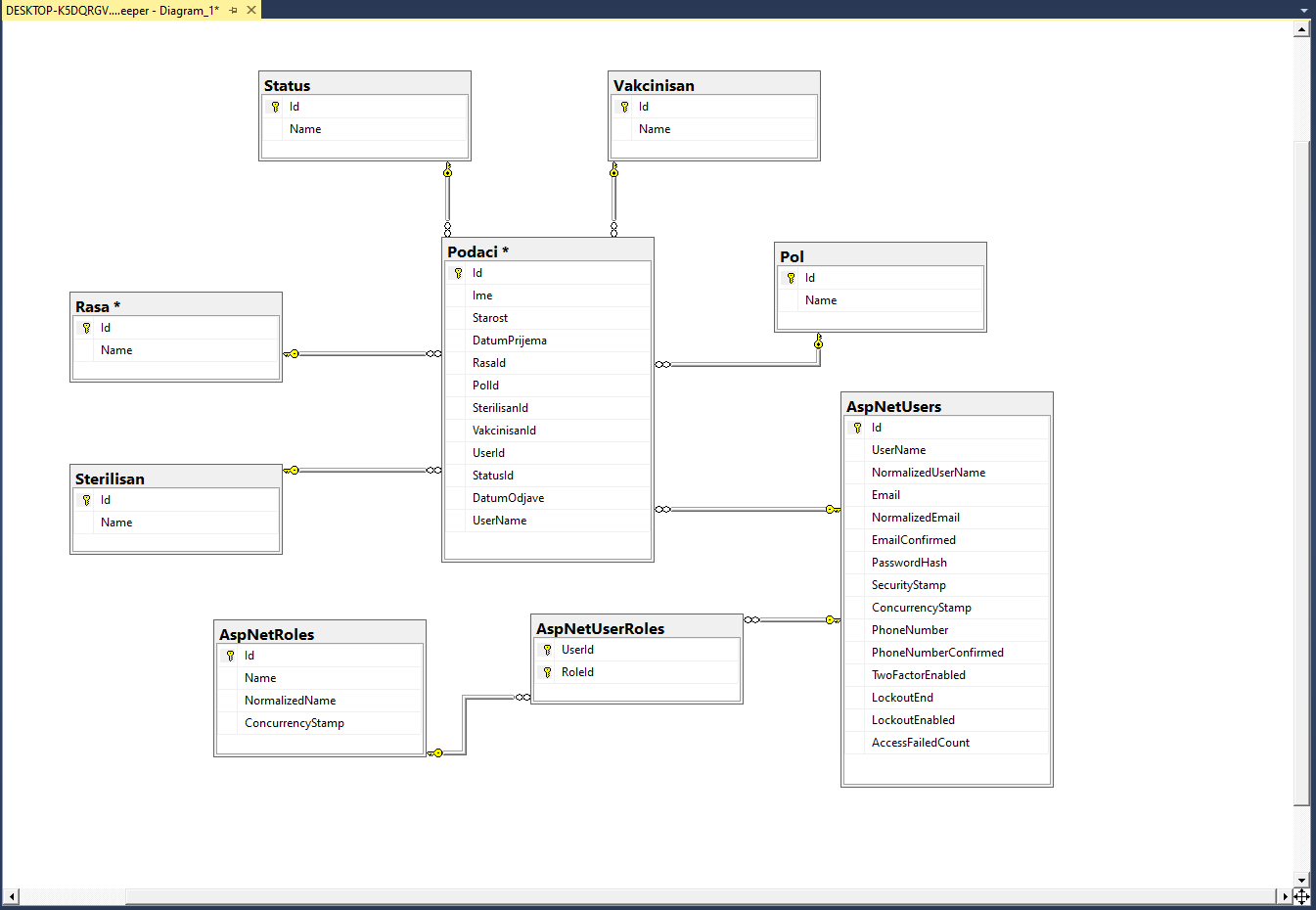
На слици 6 приказан је дијаграм активности странице „Резервација“.



Слика 6.

## Шема релационе базе података

На слици 7 приказана је шема релационе базе података.



Слика 7.

# Тестирање

## Списак функионалних захтева

У табели 3 приказани су функционални захтеви апликације.

|  |  |
| --- | --- |
| Функионални захтев | Опис |
| ФЗ-01 | Апликација ће имати 2 типа аутентификације посебно за корисника и посебно за администратора |
| ФЗ-02 | Веб презентација ће бити доступна свим посетиоцима |
| ФЗ-03 | Корисник може да види само своје резервације |
| ФЗ-04 | Корисник може да врши измене и да брише своје резервације у односу на статус |
| ФЗ-05 | Администратор може да види све резервације |
| ФЗ-06 | Администратор може да врши измене и да брише на резервације |
| ФЗ-07 | Администратор може да креира улоге |
| ФЗ-08 | Администратор може да мења улоге корисника |
| ФЗ-09 | Посетилац може да се региструје |
| ФЗ-10 | Посетилац не може да се региструје ако не испуни захтеве по питању комплексности лозинке |

Табела 3.

## Тестирање функционалности интерфејса

Списак тестова и њихових описа налазе се у табели 4.

|  |  |  |
| --- | --- | --- |
| Функционални захтев | Тест случај | Опис теста |
| ФЗ-01 | ТС-01 | Тестирање логовања на систем за администратора |
|  | ТС-02 | Тестирање логовања на систем за регистрованог корисника |
| ФЗ-03 | ТС-03 | Регистровати 2 корисника који ће направити по 2 резервације и тестирати да ли су им доступне само оне које су они креирали |
| ФЗ-04 | ТС-04 | Као администратор поставити статус једне резервације на „Прихваћено“. Када је статус резервације „Прихваћено“, заглавље је зелене боје и корисник не може да брише или мења резервацију |
|  | ТС-05 | Као администратор поставити статус једне резервације на „Одбијено“.Када је статус резервације „Одбијено“, заглавље је црвене боје и корисник не може да брише или мења резервацију |
|  | ТС-06 | Као администратор поставити статус једне резервације на „На чекању“. Када је статус резервације „На чекању“, заглавље нема боју и корисник може да брише или мења резервацију |
| ФЗ-06 | ТС-07 | Тестирати да ли администратор постојеће резервације може да мења |
|  | ТС-08 | Тестирати да ли администратор постојеће резервације може да брише |
| ФЗ-08 | ТС-09 | Администратор ће тест кориснику променити улогу над „Админ“. Тестирати да ли је корисник добио администраторске привилегије |
| ФЗ-09 | ТС-10 | Тестирати да ли посетилац може да се региструје на систем |
| ФЗ-10 | ТС-11 | При регистрацији тест корисника у поље за лозинку унети само 4 карактера. Тестирати да ли ће корисник добити обавештење о сувише једноставној лозинци. |

Табела 4.

У табели 5 дат је приказ резултата тестова.

|  |  |  |
| --- | --- | --- |
| Број тест случаја | Очекивани резултат | Добијени резултат |
| ТС-01 | Успешан | Успешан |
| ТС-02 | Успешан | Успешан |
| ТС-03 | Успешан | Успешан |
| ТС-04 | Успешан | Успешан |
| ТС-05 | Успешан | Успешан |
| ТС-06 | Успешан | Успешан |
| ТС-07 | Успешан | Успешан |
| ТС-08 | Успешан | Успешан |
| ТС-09 | Успешан | Успешан |
| ТС-10 | Успешан | Успешан |
| ТС-11 | Успешан | Успешан |

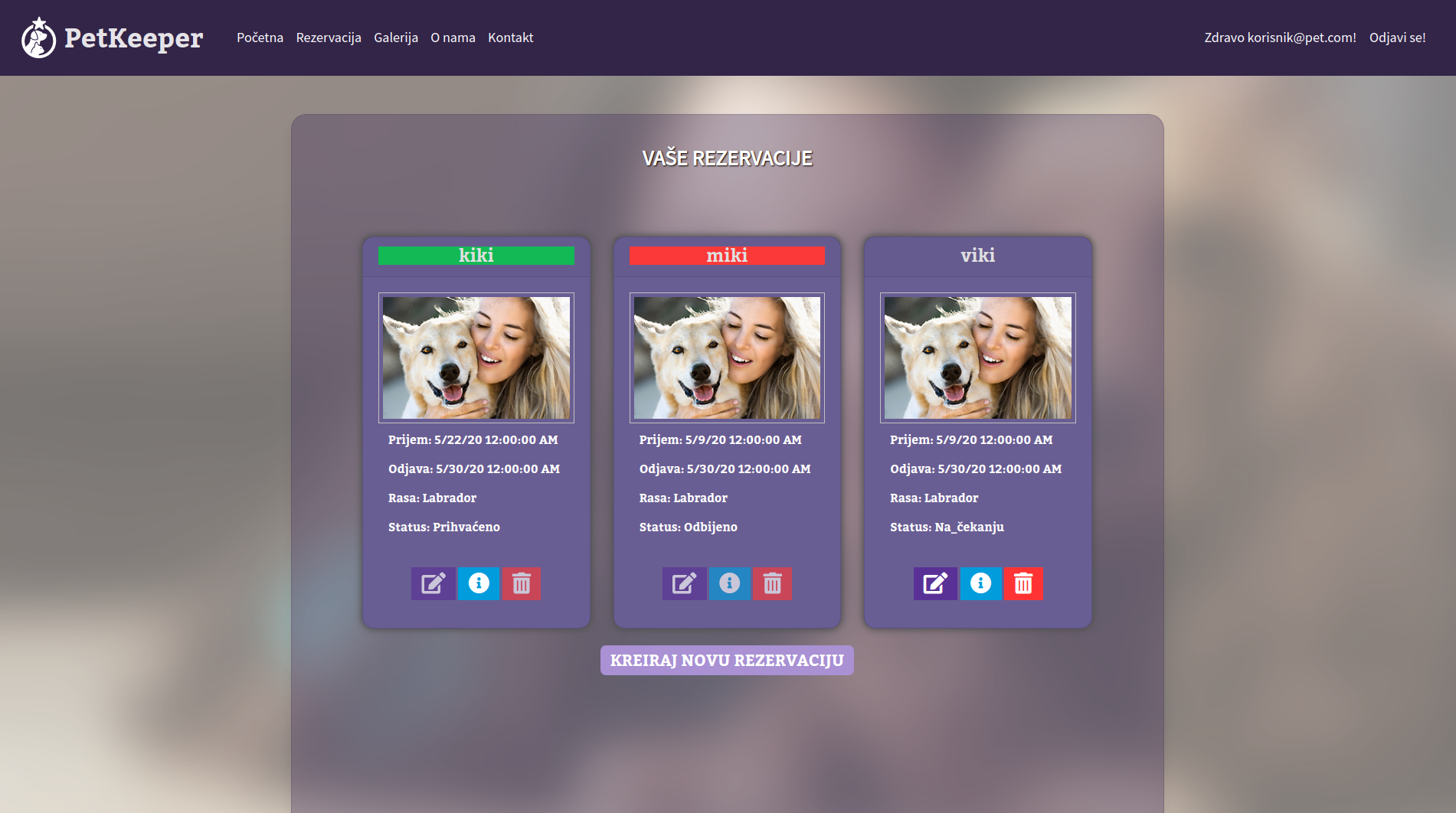
Табела 5.

# Дизајн

Приликом дизајна изгледа апликације коришћен је BOOTSTRAP v4 како би веб презентација била респонзивна на уређајима мање резолуције(мобилни телефони, таблети).

Како би корисник имао бољи преглед својих резервација, страница је дизајнирана на начин да уместо табеле приказује картице са основним подацима о резервацији. На овај начин је знатно побољшан приказ на уређајима мање резолуције.

На слици 7 приказан је изглед десктоп резолуције за регистрованог корисника.

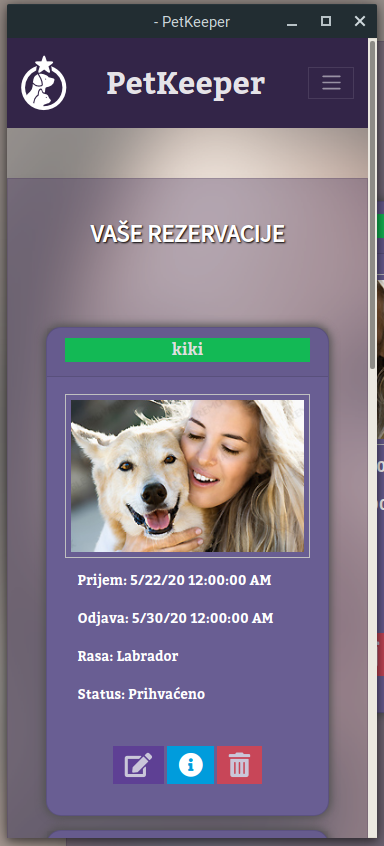


Слика 7.

Заглавље картице има различите боје у зависности од статуса резервације:

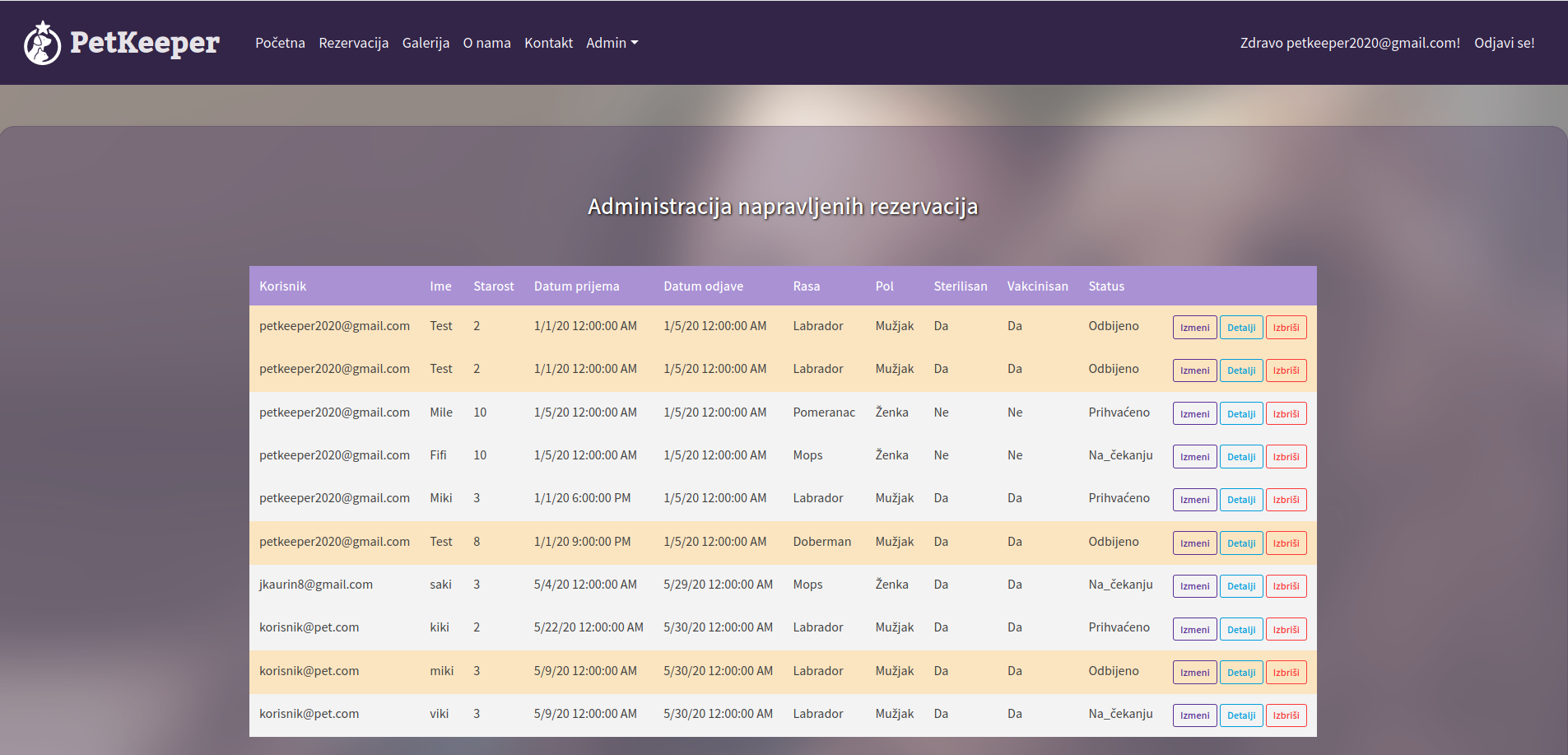
* Зелена – резервација је прихваћена
* Црвена – резервација је одбијена
* Без боје – резервација је у статусу чекања

На слици 8 приказан је изглед странице са резервацијама на мобилном телефону.



Слика 8.

Администратор због прегледа свих резервација у систему има табеларни приказ као што је приказано на слици 9.



Слика 9.

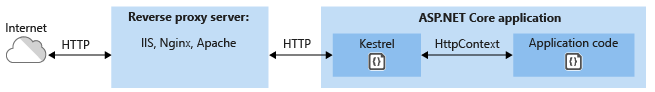
# Могућа проширења

Апликација је урађена у МВЦ технологији која пружа могућност проширења без мењања кода који је већ написан. Као могућа проширења можемо навести следећа:

* Додавањем улоге „Ветеринар“ могуће је направити модул за заказивање ветеринарских услуга.
* Додавањем улоге „Тренер“ могуће је направити модул за заказивање термина за тренинг кућног љубимца
* Сва обавештења о заказаним терминима би се слала путам СМС-а како кориснику који је направио разервацију, тако и ветеринару или тренеру који би имали увид о предстојећим активностима.
* Додавање простора разних капацитета и опремљености за кориснике који имају специфичне захтеве.
* Могућност резервације директним бирањем простора.
* Плаћање путем кредитне картице.
* Плаћање путем PayPal-a.

# Апликација у продукцији

Стављање апликације у продукцију врши се помоћу интерног DOTNET сервера KESTREL(слично TOMCAT-u). Како би повећали сигурност и омогућили модуларност биће коришћена технологија реверзног проксија као што је приказано на слици 10.



Слика 10.

# Активности тима на изради пројекта

## Тим

Тим је формиран почетком месеца марта у саставу:

1. Јелена Каурин МИТ 24/19 – кодирање, докумантација
2. Марко Козомора 16/19 – дизајн, докумантација
3. Спасоја Стојшић 25/19 – кодирање, тестирање
4. Драгослав Бабић МИТ 21/19 – организација, дизајн, докумантација

## Предуслови за израду пројекта

Одлука да се апликација изради на Дотнет платформи подразумевала је да се испуне следећи предуслови за успешну израду пројекта:

* Инсталација и конфигурација Дотнет окружења
* Инсталација и конфигурација релационе базе података
* Инсталација и конфигурација интегрисаног окружења за развој апликације (IDE app)
* Креирање заједничког јавног репозиторијума на систему за контролу верзија (ГИТ)

У табели 6 приказана су окружења за развој апликације.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Оперативни систем | *SDK* | *IDE* апликација |
| Јелена Kаурин | Window 10 | Dotnet 2.2 | Visual Studio 2017 |
| Марко Козомора | Windows 7 | Dotnet 2.2 | Visual Studio 2017 |
| Спасоја Стојшић | Windows 7 | Dotnet 2.2 | Visual Studio 2017 |
| Драгослав Бабић | Ubuntu 18.04 LTS | Dotnet 2.2 | Visual Studio Code 1.4 |

Табела 6.

## Комуникација у току израде пројекта

За комуникацију између чланова тима коршћене су апликације које подржавају све врсте оперативних система и типова уређаја а то су:

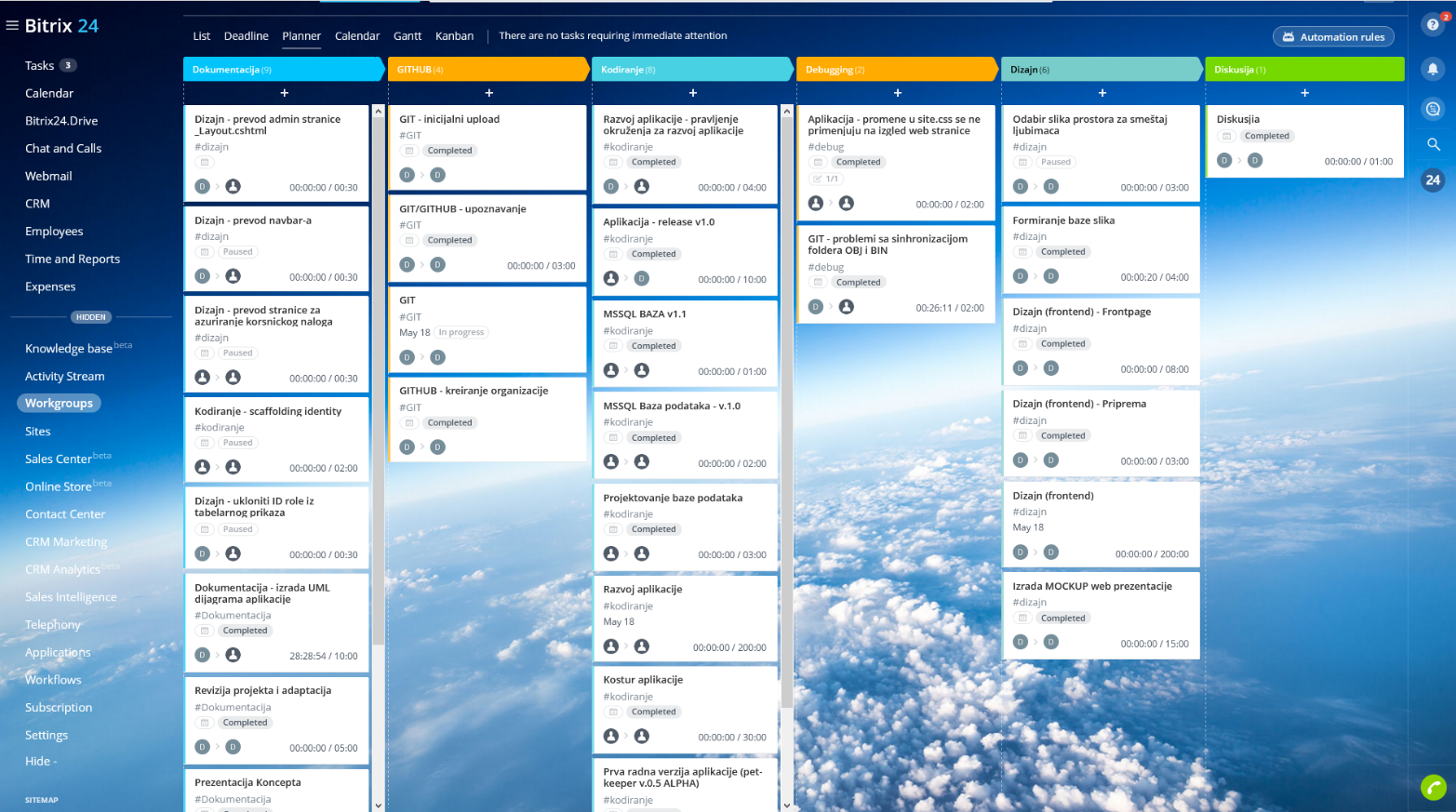
* VIBER – размена текстуалних порука
* ZOOM – коришћен код детаљнијих разрада апликације (аудио-видео конференције)

## Управљање пројектом

За управљање пројектом изабран је сервис BITRIX24. Бесплатна верзија пружила је основну функционалност која је била довољна за развој апликације и између осталог укључује следеће:

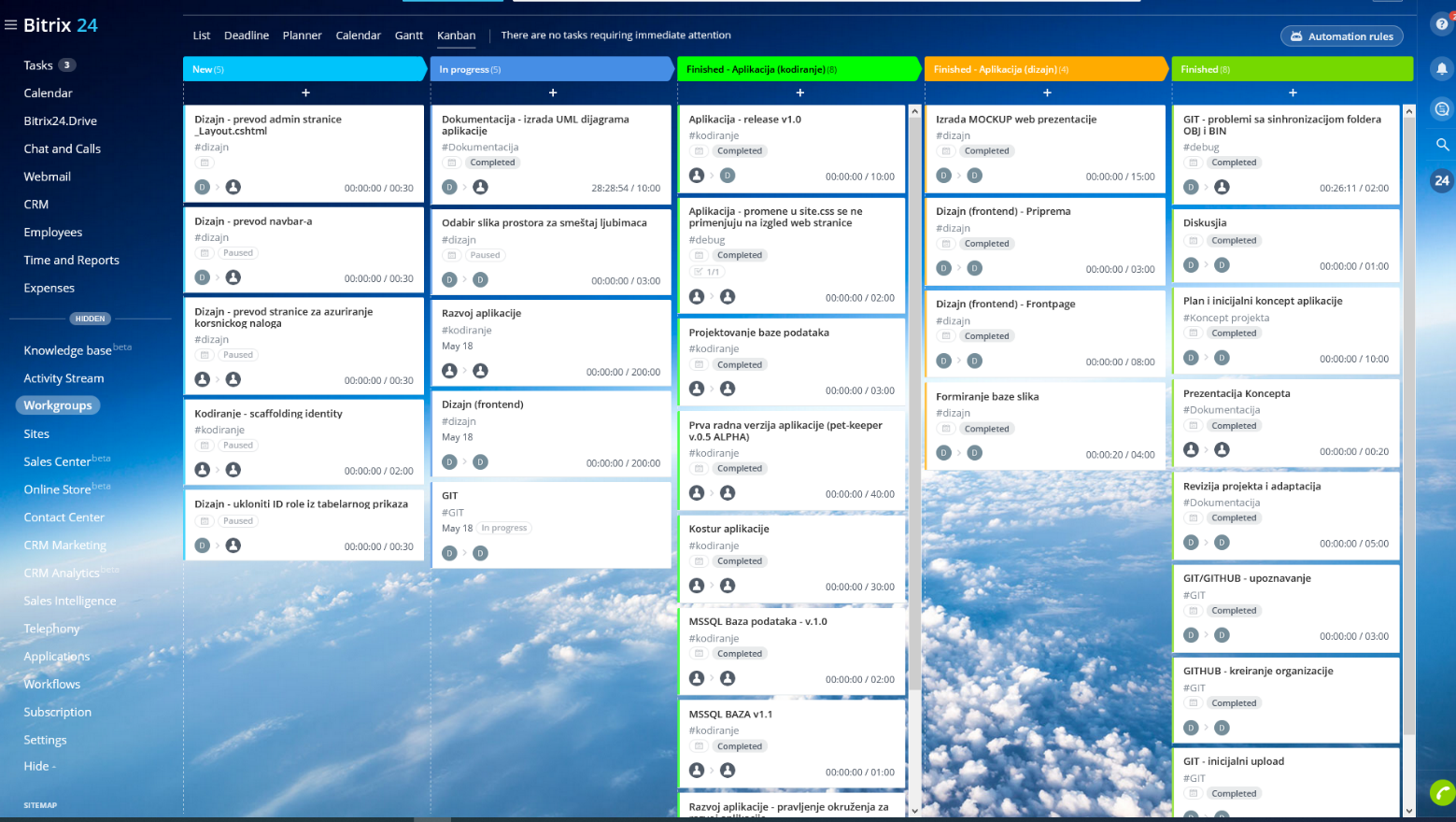
* Листа задатака
* Планер
* Календар
* Гант табелу
* Канбан

На слици 10 приказан је детаљ планера.



Слика 10.

На слици 11 приказан је детаљ Канбан табле.



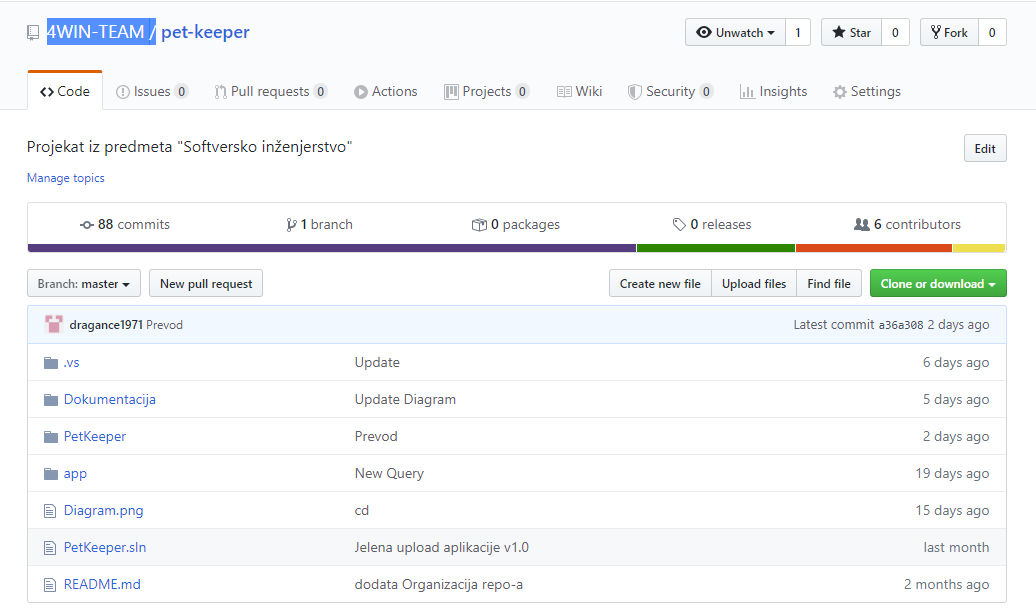
Слика 11.

На слици 12 приказан је гантограм задатака и време одређено за завршетак сваког задатка посебно.

Слика 12.

## Контрола верзије

Као систем за контролу верзије коришћен је *GITHUB.* Апликација и документација су од почетка израде пројекта јавно доступни на адреси <https://github.com/4WIN-TEAM/pet-keeper> . На слици 13 приказан је репозиторијум *4WIN-TEAM/pet-keeper*.



Слика 13.

## Анализа рада и одлука

Након завршетка пројекта тим је направио осврт на целокупан рад и одлуке које је донео у току израде и утврдио следеће:

### Моделовање апликације

Изузетну пажњу треба посветити моделовању апликације због чињенице да свака накнадна преправка изискује додатне напоре и ангажовање. Након завршетка прве пилот верзије апликације приметили смо недостатак пар атрибута у бази података, али смо захваљујући доброј комуникацији у оквиру тима то брзо уочили и исправили тако да није дошло да великог застоја у самој изради апликације.

### Контрола верзије

Иако смо као чланови тима имали одређена теоријска и практична знања о раду на GITHUB сервису, дешавало се да чланови тима истовремено праве измене на истим датотекама што је доводило до конфликта и проблема у синхронизацији пројекта. Како је време одмицало тако смо стицали више знања и искуства у коришћењу сервиса који је незаобилазан у оквиру тимског рада.

### Комуникација у оквиру тима

Користећи бесплатне сервисе као што су VIBER и ZOOM, комуникација у оквиру тима била је на доста добром нивоу, мада смо дошли до закључка да коришћење необједињених сервиса за размену текстуалних порука и датотека(нпр. садржај екрана) које су везане за конкретне задатке није најоптималније решење(VIBER – претрага постављених порука није могућа, ZOOM – временско ограничење сесије).

### Управљање пројектом

Коришћење специјализованих сервиса за управљање пројектима је битан фактор у организацији и праћењу израде апликације. Као тим, одабрали смо сервис BITRIX24 који је у пробној(trial) верзији пружио и више него довољан број могућности како би ток израде пројекта могао бити испраћен и документован у потпуности. Наравно, да би наставили са коришћењем овог сервиса, потребно је издвојити значајна финансијска средства.

### Програмски оквир

Као тим, определили смо се за DOTNET CORE који је програмски оквир отвореног кода и могуће га је покренути на свим значајнијим оперативним системима(WINDOWS, MAC OS, LINUX). MVC(model, view, controller) архитектура нам је пружила могућност да раздвојимо апликацију у три логичке целине. Предност оваквог приступа је могућност накнадних промена без преправљања већ написаног програмског кода.

Проблем на који смо наишли се огледа у усклађивању верзије SDK(software development kit) и његовој компатибилности са претходим и будућим верзијама програмског оквира. Такође остаје питање да ли су се за овакву врсту апликације могли користити неки други програмски оквири који користе мање захтевне скрипт језике (*JavaScript, PHP*).

# Закључак

Постигнут је примарни циљ пројекта, а то је израда апликације за резервисање термина у пансиону за кућне љубимце. Након урађених тестова, апликација се у потпуности показала као функионална и спремна за експлоатацију у продукционом окружењу.

Мање видљива али значајна чињеница јесте да смо као чланови тима стекли непроцењиво искуство у оквиру тимског рада које до сада нисмо имали, укључујући проблеме и њихова решења. Такође смо стекли практична знања о фазама израде пројекта и активностима које оне захтевају.