|  |  |  |
| --- | --- | --- |
|  | УНИВЕРЗИТЕТ У НОВОМ САДУ  **ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У НОВОМ САДУ** |  |

Божидар Марић

**Софтверски пакет за подршку рада запослених у школским установама**

ДИПЛОМСКИ РАД

- Основне академске студије -

Нови Сад, 2020

|  |  |  |  |
| --- | --- | --- | --- |
| Редни број, **РБР**: | |  | |
| Идентификациони број, **ИБР**: | |  | |
| Тип документације, **ТД**: | | Монографска документација | |
| Тип записа, **ТЗ**: | | Текстуални штампани материјал | |
| Врста рада, **ВР**: | | Дипломски (Bachelor) рад | |
| Аутор, **АУ**: | | Божидар Марић | |
| Ментор, **МН**: | | Др Милан Челиковић, ванредни професор | |
| Наслов рада, **НР**: | | Софтверски пакет за подршку рада запослених у образовним институцијама | |
| Језик публикације, **ЈП**: | | Српски / ћирилица | |
| Језик извода, **ЈИ**: | | Српски | |
| Земља публиковања, **ЗП**: | | Република Србија | |
| Уже географско подручје, **УГП**: | | Војводина | |
| Година, **ГО**: | | 2020 | |
| Издавач, **ИЗ**: | | Ауторски репринт | |
| Место и адреса, **МА**: | | Нови Сад; трг Доситеја Обрадовића 6 | |
| Физички опис рада, **ФО**: (поглавља/страна/ цитата/табела/слика/графика/прилога) | | 5/69/0/90/46/0/0 | |
| Научна област, **НО**: | | Електротехника и рачунарство | |
| Научна дисциплина, **НД**: | | Примењене софтверско инжењерство | |
| Предметна одредница/Кqучне речи, **ПО**: | | Базе података 2 | |
| **УДК** | |  | |
| Чува се, **ЧУ**: | | У библиотеци Факултета техничких наука, Нови Сад | |
| Важна напомена, **ВН**: | |  | |
| Извод, **ИЗ**: | | У овом раду презентован је пројекат за реализацију дела информационог система једне школе. Циљ изабраног дела информационог система је да поједностави процес евиденције рада запосленима у школи. Описани информациони систем омогућава учитељима и наставницима да организују предавања, часове, контролне, домаће, евидентирају присуство ученика и оцене радове истих. На основу података унетих у базу система, даље је могуће прегледати испланиране часове за одређени датум, као и послати информације везане за успех ученика њиховим родитељима путем мејла. Модел шеме базе података пројектована је помоћу пакета Entity Framework. Пројекат је затим реализован на платформи Visual Studio 2019. | |
| Датум прихватања теме, **ДП**: | |  | |
| Датум одбране, **ДО**: | |  | |
| Чланови комисије, **КО**: | Председник: |  |
|  | Члан: |  | Потпис ментора |
|  | Члан, ментор: |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Accession number, **ANO**: | |  | |
| Identification number, **INO**: | |  | |
| Document type, **DT**: | | Monographic publication | |
| Type of record, **TR**: | | Textual printed material | |
| Contents code, **CC**: | | Bachelor Thesis | |
| Author, **AU**: | | Božidar Marić | |
| Mentor, **MN**: | | Millan Čeliković, Associate Professor, Ph.D | |
| Title, **TI**: | | A software tool for school System Support Service | |
| Language of text, **LT**: | | Serbian | |
| Language of abstract, **LA**: | | Serbian | |
| Country of publication, **CP**: | | Republic of Serbia | |
| Locality of publication, **LP**: | | Vojvodina | |
| Publication year, **PY**: | | 2020 | |
| Publisher, **PB**: | | Author’s reprint | |
| Publication place, **PP**: | | Novi Sad, Dositeja Obradovica sq. 6 | |
| Physical description, **PD**: (chapters/pages/ref./tables/pictures/graphs/appendixes) | | 5/69/0/90/46/0/0 | |
| Scientific field, **SF**: | | Electrical and computer engineering | |
| Scientific discipline, **SD**: | | Applied software engineering | |
| Subject/Key words, **S**/**KW**: | | Databases 2 | |
| **UC** | |  | |
| Holding data, **HD**: | | The Library of Faculty of Technical Sciences, Novi Sad, Serbia | |
| Note, **N**: | |  | |
| Abstract, **AB**: | | This thesis presents the specification and implementation of a section of an information system for a school. The aim of the selected section of the information system is to simplify the process of recording employees work. The described information system allows for teachers to organise lectures, lessons, tests, homeworks, record student attendance and student grades. Based on the information present in system's database it is further possible to review lessons planned for a certain date and to send informations pertaining to student’s achievements to their parents via email. The database schematic was created using the Entity Framework package. The project was afterwards implemented using the Visual Studio 2019 platform. | |
| Accepted by the Scientific Board on, **ASB**: | |  | |
| Defended on, **DE**: | |  | |
| Defended Board, **DB**: | President: |  |
|  | Member: |  | Menthor's sign |
|  | Member, Mentor: |  |  |

|  |  |  |
| --- | --- | --- |
|  | УНИВЕРЗИТЕТ У НОВОМ САДУ ⚫ **ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА**  21000 НОВИ САД, Трг Доситеја Обрадовића 6 | Датум: |
|  |
| **ЗАДАТАК ЗА ДИПЛОМСКИ РАД** | Лист/Листова: |
|  |

*(Податке уноси предметни наставник - ментор)*

| Студијски програм: | **Рачунарство и аутоматика** |
| --- | --- |
| Руководилац стидијског програма: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Студент: | **Божидар Марић** | Број индекса: | **ПР 32/2016** |
| Област: |  | | |
| Ментор: | **др Милан Челиковић** | | |
| НА ОСНОВУ ПОДНЕТЕ ПРИЈАВЕ, ПРИЛОЖЕНЕ ДОКУМЕНТАЦИЈЕ И ОДРЕДБИ СТАТУТА ФАКУЛТЕТА ИЗДАЈЕ СЕ ЗАДАТАК ЗА ДИПЛОМСКИ РАД, СА СЛЕДЕЋИМ ЕЛЕМЕНТИМА:   * проблем – тема рада; * начин решавања проблема и начин практичне провере резултата рада, ако је таква провера неопходна; | | | |

**НАСЛОВ ДИПЛОМСКОГ РАДА:**

|  |
| --- |
| **Софтверски пакет за подршку рада запослених у школским установама** |

**ТЕКСТ ЗАДАТКА:**

|  |
| --- |
| * Проучити аспекте практичне примене изабраних алата за пројектовање и имплементацију шеме базе података и алата за пројектовање и имплементацију апликација информационих система. * Испројектовати сегмент концептуалне и имплементационе шеме базе података, потребан за развој дела информационог система прихватилишта за животиње, а намењен за проналажење животиње са карактеристикама дефинисаним од стране потенцијалног усвојитеља. * Имплементирати испројектовани сегмент шеме базе података на изабраном систему за управвљање базама података. * Имплементирати апликативно софтверско решење за део система који покрива функције за подршку процеса анализе упитника и проналажење животиње са карактеристикама дефинисаним од стране потенцијалног усвојитеља. |

|  |  |
| --- | --- |
| Руководилац студијског програма: | Ментор рада: |
|  |  |

|  |
| --- |
| Примерак за: - Студента; - Ментора |

Образац **Q2.НА.04-03** - Издање 3

Садржај

[1. Увод..........................................................................................................................................................1](#_Toc52484302)

[2. Опис реалног система 2](#_Toc52484303)

[2.1 Типови корисника и њихове карактеристике 2](#_Toc52484304)

[2.2 Функционални захтеви 3](#_Toc52484305)

[2.2.1 Основне корисничке функционалности 3](#_Toc52484306)

[2.2.2 Администраторске функционалности 3](#_Toc52484307)

[2.2.3 Функционалности запослених 4](#_Toc52484308)

[2.3 Перспектива система 4](#_Toc52484309)

[3. Опис шеме базе података 7](#_Toc52484310)

[3.1 Концептуална шема базе података 7](#_Toc52484311)

[3.1.1 Појмови коришћени при моделовању концептуалних шема база података 8](#_Toc52484312)

[3.2 Модел шеме базе података 8](#_Toc52484313)

[3.2.1 Део модела шеме базе података за смештање података о корисницима 8](#_Toc52484314)

[3.2.2 Део модела шеме базе података за смештање података о запосленима 9](#_Toc52484315)

[3.2.3 Део модела шеме базе података за смештање података о ученицима 10](#_Toc52484316)

[3.2.4 Део модела шеме базе података за смештање података о предметима 10](#_Toc52484317)

[3.2.5 Део модела шеме базе података за смештање информација о предавањима и часовима 11](#_Toc52484318)

[3.2.6 Део модела шеме базе података за смештање података о контролним тачкама 13](#_Toc52484319)

[3.3 Базе података генерисане из модела шема базе података 14](#_Toc52484320)

[3.3.1 Изглед база података генерисаних за потребе информационог система 15](#_Toc52484321)

[3.4 Опис ентитета и асоцијација у моделима 17](#_Toc52484322)

[4. Опис апликативног решења 23](#_Toc52484323)

[4.1 Пријем животиње 24](#_Toc52484324)

[4.2 Уређивање профила животиње 25](#_Toc52484325)

[4.4 Креирање упитника 28](#_Toc52484326)

[4.5 Попуњавање упитника 30](#_Toc52484327)

[4.6 Покретање потраге 31](#_Toc52484328)

[4.6 Начин рачунања поклапања 32](#_Toc52484329)

[4.7 Приказ поклапања и промена статуса 35](#_Toc52484330)

[4.8 Креирање и штампање извештаја о поклапањима за животињу 36](#_Toc52484331)

[5. Закључак 37](#_Toc52484332)

[Речник појмова 40](#_Toc52484333)

[Скраћенице 42](#_Toc52484334)

[Литература 44](#_Toc52484335)

[Биографија 46](#_Toc52484336)

# 1. Увод

У дипломском раду описан је информациони систем једне школске установе. Примарни циљ oписаног информационог система је да пружи подршку евиденције рада запосленима у школи. Софтверска подршка за наведени процес омогућава запосленима да информације неопходне за рад у образовној установи имају увек на располагању на једном месту. Без софтверске подршке запослени су принуђени да са собом носе неопходне информације у папирној форми што непотребно компликује рад запослених. Узимајући у обзир количину документације укључене у процес, постоји вероватноћа да ће доћи до губљења или занемаривања дела података, или грешке приликом обраде.

Такође, родитељи ученика, често немају доступне све потребне информације о успеху свог детета. Због тога, они су принуђени да долазе лично у школу да траже те информације, што може представљати проблем родитељима који су на послу у време рада школе.

Информациони систем доприноси прегледности и доступности података и убрзава и олакшава рад запослених, при чему се истовремено повећава степен информисаности родитеља о академским успесима своје деце. Смањење папирологије, запосленима омогућава да се више посвете раду са својим ученицима, што је примарни део њиховог посла.

Поред Увода, Закључка, Речника појмова , Литературе и Биографије, овај дипломски рад садржи три поглавља:

* **Поглавље 2. Опис реалног система** – у овом поглављу описани су типови корисника, функционални захтеви и перспектива система,
* **Поглавље 3. Опис шема базе података** – у овом поглављу представљени је модел базе података и дат опис свих ентитета и релација присутних у систему и
* **Поглавље 4. Опис апликативног решења** – у овом поглављу приказано је програмско решење за функционалне захтеве описане у поглављу 2.

# 2. Опис реалног система

У овом поглављу описани су типови корисника, њихове карактеристике и надлежности у информационом систему основне школе. Потом су наведени функционални захтеви које систем треба да реализује, као и перспективе таквог система.

## 2.1 Типови корисника и њихове карактеристике

У информационом систему основне школе присутни су следећи типови корисника:

1. Учитељ:
   * задужен за једно одељење,
   * организује предавања, часове, домаће и контролне задатке за своје одељење,
   * евидентира присуство ученика на часовима,
   * оцењује радове ученика и
   * шаље извештај о тренутним успесима својих ученика њиховим родитељима
2. Наставник:

* предаје један предмет,
* организује предавања, часове, домаће и контролне задатке из свог предмета, а у оквиру одељења којима предаје,
* оцењује радове ученика и
* шаље извештај о тренутним успесима ученика у одељењима којима предаје на његовом предмету.

1. Администратор система:

* уноси информације о тренутним запосленима у школској установи,
* ажурира информације о одељењима школе,
* ажурира информације о ученицима који похађају школу,
* ажурира информације о учионицама које у школи постоје и
* ажурира информације о предметима и њиховим областима који се у школи предају.

Карактеристике типова корисника информационог система су:

1. *Учитељи* – предају више предмета једном одељењу. У систему је неопходно да постоји одељење којем је учитељ додељен, као и предмети за разред који одговара разреду одељења којем учитељ предаје, у супротном учитељ не може ефикасно да користи информациони систем.
2. *Наставници* – предају један предмет једном или више одељења. У систему је неопходно да постоји предмет који наставник држи и барем једно одељење којем наставник предаје, иначе наставник не може да користи систем на планирани начин.
3. *Администратор система* – задужен за иницијализацију базе података неопходним информацијама које би запосленима омогућиле што ефикасније коришћење информационог система. Неопходне информације обухватају креирање корисничких налога за запослене, евидентирање учионица, ученика који похађају школу, одељења која постоје у школи, предмета који се предају и сл.

## 2.2 Функционални захтеви

У оквиру поглавља наведени су и описани функционални захтеви које информациони систем треба да испуни. Ради лакшег разумевања, захтеви су груписани према типовима корисника за које су везани и то у три групе: основне корисничке, администраторске и функционалности запослених.

### 2.2.1 Основне корисничке функционалности

Основне корисничке функционалности обухватају групу захтева везане за управљање корисничким налозима у оквиру информационог система једне школске установе. У наведене функционалности спадају:

1. **Регистрација запослених** – обавља се од стране администратора система. Приликом регистрације новог запосленог администратор уноси основне информације о запосленом и додељује му улогу у оквиру система у складу са којом запослени може да позива друге функционалности система.
2. **Пријава на систем** – сви претходно регистровани корисници система су у могућности да позову ову функционалност. Након успешне пријаве корисник добија приступ систему у складу са својом улогам.
3. **Ажурирање информација о личности** – запослени који имају направљен кориснички налог су у могућности да позову ову функционалност. Изузев администратора, податке о личности могу видети само они корисници на које се ти подаци односе.
4. **Ажурирање лозинке** – сви запослени могу да промене своју лозинку, при чему се од њих тражи да је промене при првој пријави на систем.

### 2.2.2 Администраторске функционалности

Функционалности описане у оквиру овог поглавља намењене су да подрже рад запослених у школској установи и представљају основу за даљу употребу информационог система од стране запосленог. Њих може позвати искључиво администратор система и у њих спадају:

1. **Ажурирање корисничких налога** – администратор може направити нови кориснички налог и обрисати постојећи. По креирању корисничког налога, одговорност одржавања информација о личности у ажурном стању прелази на запосленог за ког је тај кориснички налог креиран, те је онемогућено да администратор мења једном унете податке. У случају прекида радног односа запосленог, он се мора уклонити из система од стране одговорног лица, те је ова функционалност везана за самог администратора и само је он може позвати.
2. **Ажурирање учионица** – запосленима је потребно обезбедити валидне информације о постојећим учионицама у школској установи ради планирања часова. Како не би требало да се те информације мењају често и од стране било ког корисника система ова функционалност је везана само за администратора информационог система.
3. **Ажурирање информација о предметима** – запосленима је такође неопходно обезбедити информације о предметима који се предају у школској установи. Креирање, мењање и брисање ових информација одговорност је искључиво администратора система.
4. **Ажурирање информација о ученицима** – у систему је неопходно евидентирати све ученике који похађају школску установу како би запослени имали информације о ученицима којима предају. Унос, модификација и брисање ових информација мора обављати одговорно лице, те је ово још једна од функционалности администратора система.
5. **Ажурирање информација о одељењима** – како се часови одржавају у групама ученика и обзиром да запосленима морамо омогућити информације о групама ученика којима држе часове, одговорност је администратора система да те информације унесе у систем. Унос подразумева креирање, модификацију и брисање одељења, али и додавање постојећих ученика у одељења, као и додељивање одељења запосленима, ако тим одељењима предају.

### 2.2.3 Функционалности запослених

У овом поглављу описане су функционалности које запослени који имају кориснички налог могу да позову уз освит на разлике, ако их има, између позива од стране учитеља и од стране наставника. Обухвата следеће функционалности:

1. **Креирање часова** – запосленима је омогућено евидентирање планираних часова, при чему запослени бира предмет и његову област за коју планира час, као и одељење и учионицу у којој ће час бити одржан. Дозвољено је брисање и модификација часова.
2. **Евидентирање присуства ученика** – запосленима је омогућено да евидентирају присутне ученике на часовима које су унели у систем.
3. **Креирање домаћих задатака и контролних** – у циљу евидентирања рада ученика на неком предмету, запосленима је дозвољено да креирају, мењају и бришу домаће и контролне задатке из предмета које предају, при чему су ти задаци повезани са је одељењем којем су намењени.
4. **Креирање предавања** – ради лакшег планирања области које ће запослени да прелази у неком дану, омогућено је креирање, мењање и брисање предавања. Предавања, за разлику од часова, нису везана за учионицу нити одељење јер она представљају подсетник запосленом да за одређени дан жели да испредаје одређену област.
5. **Оцењивање ученичких радова** – запосленом је такође омогућено да оцењује домаће и контролне задатке које је је претходно задао одређеном одељењу.
6. **Преглед распореда часова** – ради додатног олакшања при планирању часова, запосленима је омогућен преглед постојећих часова за неки задати датум.
7. **Информисање родитеља о успеху ученика** – запосленима је такође омогућено да пошаљу тренутне информације о постигнутом успеху ученика на мејл адресу родитеља тог ученика.

## 2.3 Перспектива система

Примерна намена информационог система, описаног у овом дипломском раду, је да олакша рад запосленима у образовним институцијама. Увођењем информационог система у образовне установе, запосленима је омогућено да потребне информације обрађују електронским путем на једноставнији начин, тиме смањујући време које би се у супротном трошило на документацију где би се те информације иначе чувале. Такође, аутоматским слањем извештаја о тренутном успеху ученика његовим родитељима избегава се потреба за родитељским састанцима или позивима. Заобилажењем информисања сваког родитеља појединачно о академским постигнућима његовог детета, оставља се више простора запосленима у школама да се посвете свом истинском задатку, што је едукација његових ученика. У светлу тренутне глобалне ситуације, требало би напоменути да је ова функционалност система од посебног значаја, јер се помоћу ње смањује потреба за лични сусрет родитеља и запосленог

Могуће проширење система било би прављење подсистема за статистичку обраду података о ученицима, како би се ти извештаји могли користити у унапређењу рада са ученицима. Даља проширења су такође могућа прављењем корисничких налога и за родитеље, који би се онда могли пријавити на систем и самостално у било ком тренутку погледати информације о свом детету, што би наравно захтевало да се информациони систем имплементира у облику веб апликације. Овакво проширење би погодовало и запосленима, јер би им омогућило да свој рад планирају са било које локације, не само са рачунара на којем би се информациони систем налазио.

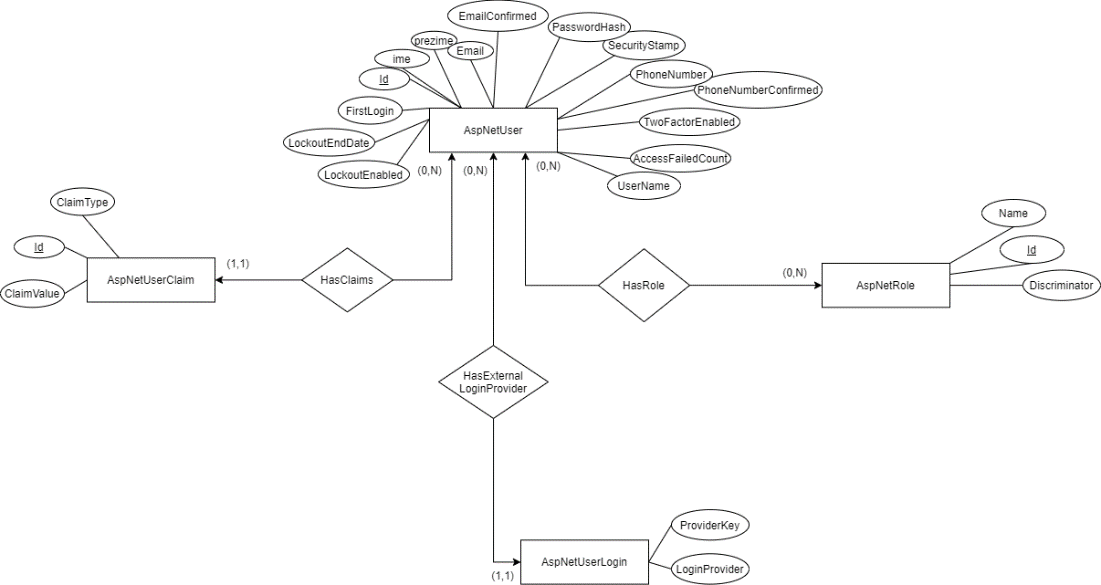
# 

# 3. Опис шеме базе података

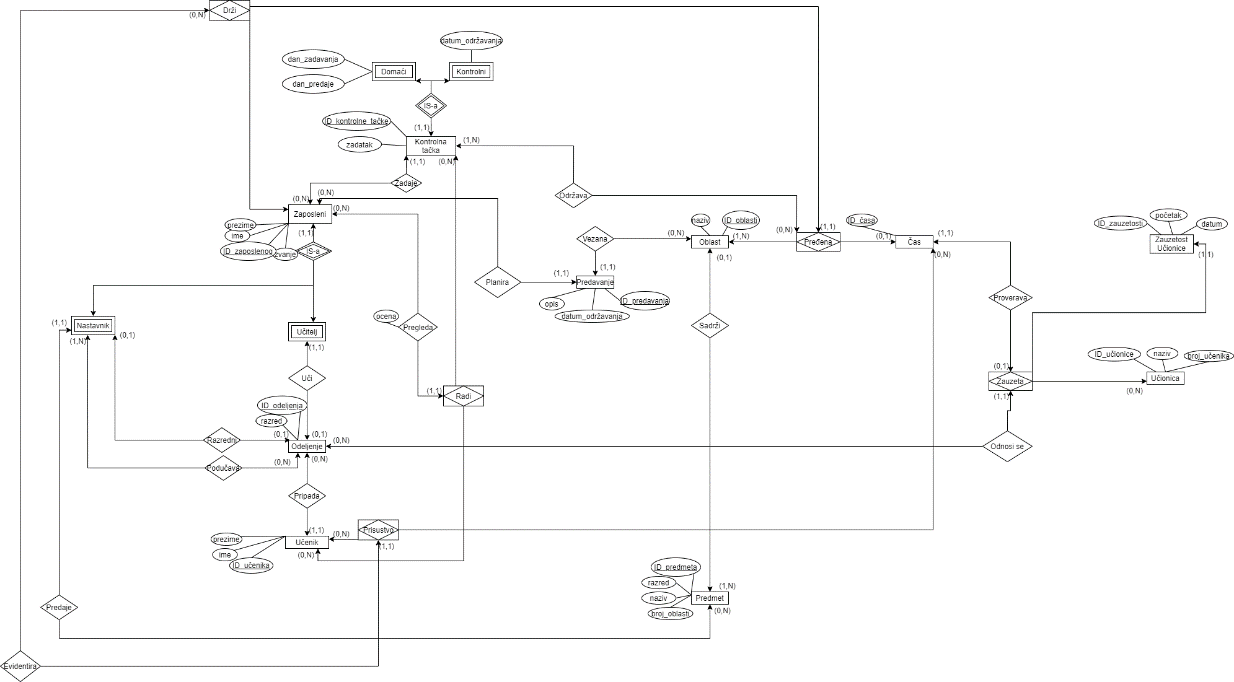
У овом поглављу биће описане концептуалне шеме база података коришћене за развој информационог система, а затим и модели шема база података коришћене за генерисање базе. Такође, биће анализирана и сама база података генерисана на основу модела шема база података употребом функционалности из *Entity Framework* пакета. На крају, дати су описи ентитета и асоцијација у оквиру модела креираног помоћу *Entity Framework Designer* алата.

## 3.1 Концептуална шема базе података

Концептуални модели шема база података засновани су на анализи информација које су неопходне за што једноставнији рад запослених у образовним институцијама. Модели су презентован у форми ЕР дијаграма и налази се на сликама испод (слика 3.1 и слика 3.2). Како су све информације са тог модела пренете у модел шеме базе података направљен помоћу Entity Framework пакета, детаљи типова ентитета и типова повезника биће објашњени у следећем поглављу, док ће се овде размотрити појмови коришћени у моделу и начин на који су ти појмови преведени тако да их Entity Framework пакет разуме.



*Слика 3.1 – Концептуална шема базе података за смештање информација о корисницима*



*Слика 3.2 – Концептуална шема базе података за смештање информација које су потребне за рад запослених*

### 3.1.1 Појмови коришћени при моделовању концептуалних шема база података

Основни појмови у ЕР дијаграмима су типови ентитета са њиховим обележјима и типови повезника. Типови ентитета представљају ресурс пословања у реалном систему, обележја типова ентитета су особине које тај ресурс поседује, док тип повезника представља однос у којем се ресурси реалног система налазе. При моделовању концептуалне шеме базе података, поред наведених појмова коришћени су још и герунд и ИС-А хијерархија. Герунд представља тип повезника који се може посматрати као тип ентитета у релацији са неким другим типом ентитета, а ИС-А хијерархија нам омогућава да моделујемо суперкласу и поткласе, односно наслеђивање [1].

При моделовању шеме базе података употребом Entity Framework Designer алата, типови ентитета представљају се преко концепта *Entity*, обележја се додају сваком *Entity*-у као *ScalarProperty*, док се типови повезника моделују помоћу *Association* концепта. У случајевима када имамо тип повезника са неким обележјем, то обележје се преноси у тип ентитета који има максимални кардиналитет 1. Тип повезника који са обе стране има максимални кардиналитет „више“ се у случајевима, ако има неко додатно обележје или представља тип ентитета у релацији са другим типом ентитета (прецизније ако је у питању герунд) моделује посебним *Entity*-ем. Герунд, осим у претходно описаном случају, прати иста правила као и обичан тип повезника при моделовању у Entity Framework Designer-у. ИС-А хијерархија се моделује при креирању *Entity*-а који су поткласе у хијерархији, тако што се изпадајућег менија означеног као *Base type* изабере који од већ постојећих ентитета представља суперкласу.

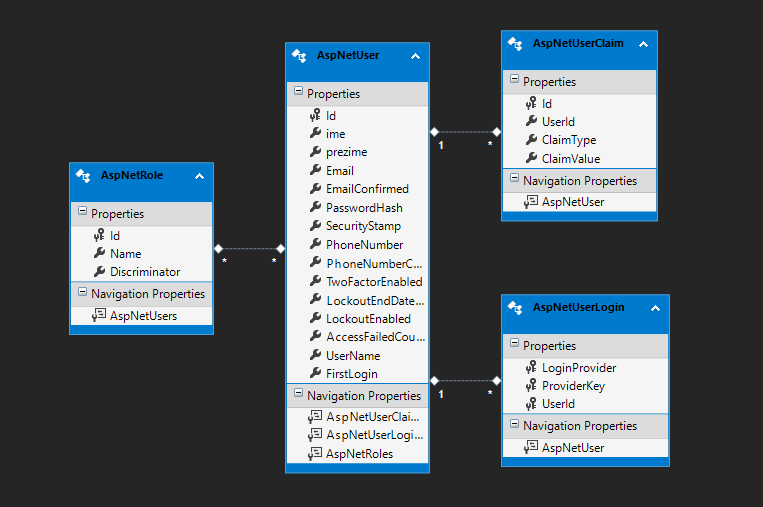
Уколико се модел базе података прави употребом *Code First* приступа, у оквиру класе која представља тип ентитета, обележја наводимо преко пропертија одговарајућег типа, ИС-А хијерархију дефинишемо сходно правилима наслеђивања програмског језика који користимо, док се типови повезника реализују пропертијима класног типа који означавају тип ентитета са којим је класа у релацији.

## 3.2 Модел шеме базе података

Модели шема база података засноване су на концептуалној шеми базе података према правилима која су описана у претходном полављу, уз посебан обзир на могућности за даља проширења. У циљу веће прегледности модел је представљен кроз више делова.

### 3.2.1 Део модела шеме базе података за смештање података о корисницима

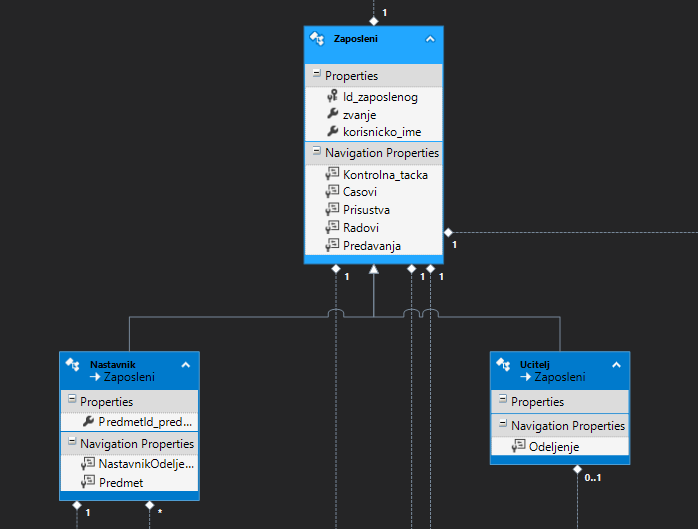
Корисничке информације представљају личне информације везане за запослене у школи, те су оне смештене у посебну базу ради додатне заштите. За креирање нове базе искоришћене су класе добијене у пакету *Microsoft.AspNet.Identity*. Изглед модела шеме базе података са овим информацијама дат је на слици испод (слика 3.3). Заједничке особине корисника представљене су типом ентитета *AspNetUser*. За корисника се везује тачно једна улога у оквиру система која је представљена типом ентитета *AspNetRole*. Типови ентитета *AspNetUserClaim* и *AspNetUserLogin* у тренутној имплементацији нису коришћени, али могу бити искоришћени за даљи развој апликације. *AspNetUserClaim* може се користити уместо улога за одређивање шта све корсиник има могућност да уради у оквиру нашег информационог система, док се у *AspNetUserLogin* чувају информације о начинима на који корисник може да се пријави на систем коришћењем сервиса који нису део нашег система. Ова два типа ентитета би нам могла бити од користи када бисмо даље развијали информациони систем у веб апликацију, о чему је било речи у поглављу *2.3 Перспектива система*.



*Слика 3.3 – Део дијаграма мдоела шеме базе података за смештање података о корисницима*

### 3.2.2 Део модела шеме базе података за смештање података о запосленима

Овај део модела шеме базе података обухвата типове ентитета и повезника потребних за чување података о запосленима (*слика 3.4*). Типови ентитета приказани на слици испод представљају збир информација које су неопходне за даљи рад запосленог, односно помоћу ових ентитета и њихових особина могуће је повезати запосленог са свим информацијама које су му потребне за рад на систему, као и приказати информације о раду који је у оквиру система запослени већ обавио.

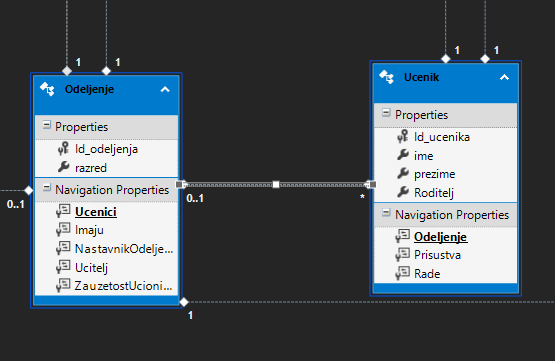


*Слика 3.4 – Део дијаграма модела шеме базе података за смештање података о запосленима*

Разлика између *Properties* и *NavigationProperties* је у томе што се ови други аутоматски генеришу из асоцијација између типова ентитета, тако на пример *Predmet* представља тип ентитета Предмет са којим је тип ентитета Наставник повезан. Такође, треба напоменути да је *korisnicko\_ime* особина искоришћена да повеже одређеног запосленог са његовим корисничким информацијама које се налазе у другој бази. Повезивање запосленог са његовим корисничким информацијама одрађено је на апликативном нивоу и о томе ће бити више речи у неком од следећих поглавља.

### 3.2.3 Део модела шеме базе података за смештање података о ученицима

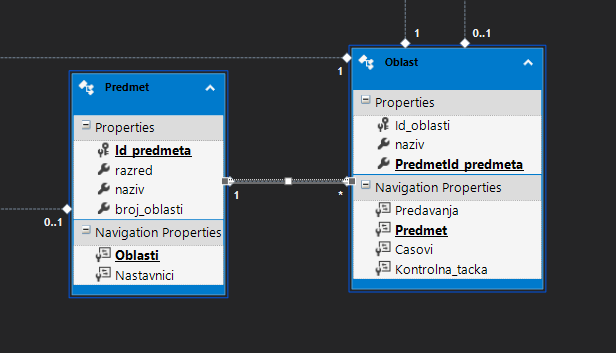
Подаци о ученицима који похађају школску установу као и о припадности једног ученика неком одељењу приказани су на слици испод (слика 3.5). На слици је подебљана асоцијација која представља припадност ученика једном одељењу, при чему су навигационе особине креиране на основу те асоцијације на слици приказане подебљаним словима.



*Слика 3.5 – Део дијаграма модела шеме базе података за смештање података о ученицима и одељењима*

### 3.2.4 Део модела шеме базе података за смештање података о предметима

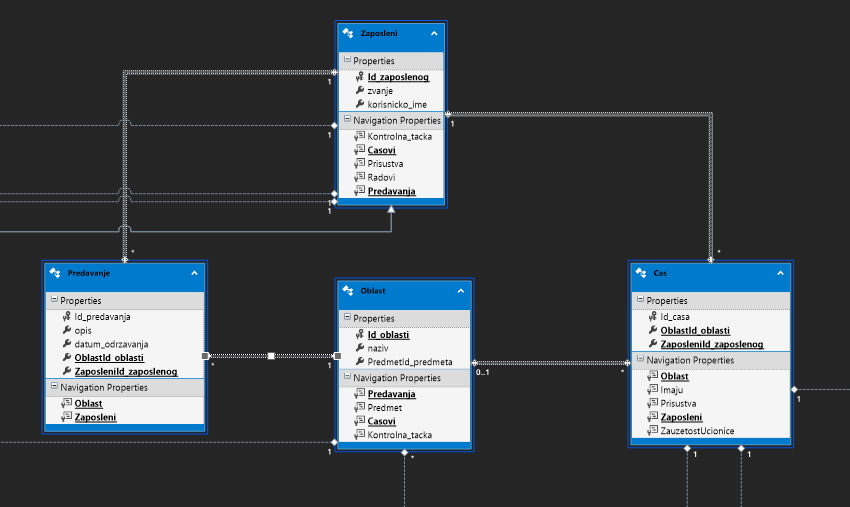
За рад запослених у образовној институцији неопходне су информације о предметима који се предају као и о областима које ти предмети садрже. На слици 3.6 приказани су типови ентитета искоришћени за складиштење тих информација. Веза између типа ентитета Предмет и типа ентитета Област, као и изгенерисане навигационе особине, су подебљане на слици. Такође, како предмет садржи више области дошло је до преноса страног кључа из Предмета у Област, ради бољег повезивања табела у бази и ефикаснијег приступа информацијама из апликације, што се може и приметити на слици.



*Слика 3.6 – Део дијаграма модела шеме базе података за смештање информација о предмету и његовим областима*

### 3.2.5 Део модела шеме базе података за смештање информација о предавањима и часовима

Једна од основних функционалности која мора бити омогућена запосленима образовне установе је да испланирају предавања и евидентирају часове. На слици испод (слика 3.7) можемо видети како изгледају и на који начин су повезани типови ентитета Предавање и Часови.



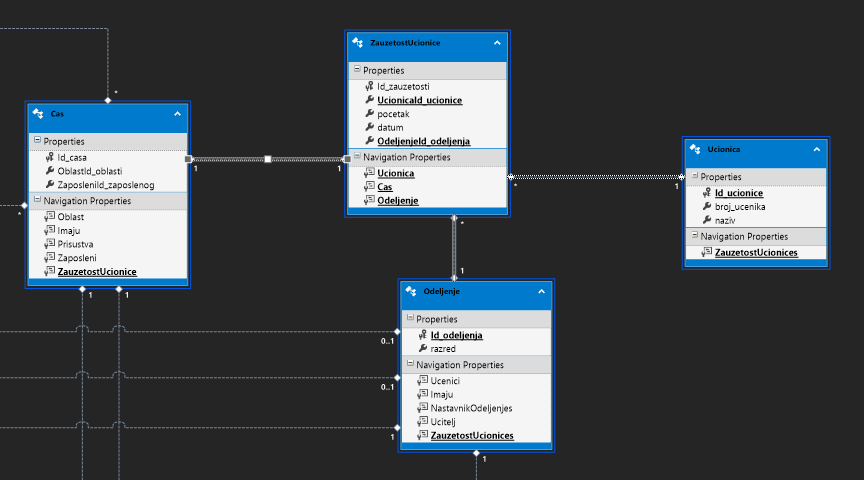
*Слика 3.7 – Део дијаграма модела шеме базе података за смештање информација о предавањима и часовима*

Као што се на слици може видети, Област је повезана и са Часом и са Предавањем, а како једна област може бити везана за више Часова и више Предавања, тако је дошло до преноса страног кључа из Области у Предавање, односно Час. Такође, један запослени може да држи више часова и да испланира више предавања, те је страни кључ пренесен из Запосленог у Час и Предавање.

Приликом евидентирања часова од интереса је и учионица у којој ће се час одржавати, као и одељење којем се час одржава. На слици испод (слика 3.8) приказани су типови ентитета Час, Учионица, Заузетост Учионице и Одељење, као и повезаност између њих.

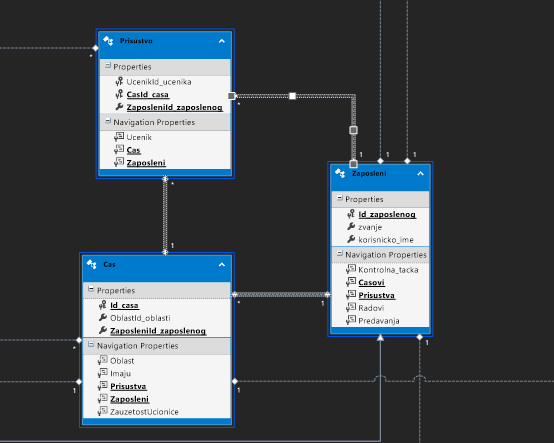
Учионица представља тип ентитета који садржи информације о учионицама које у образовној установи постоје, док је Заузетост Учионице тип ентитета у којој се бележе информације о датуму и времену када је учионица заузета, како не би дошло до двоструког резервисања учионица. Тип ентитета Одељење се у овом случају користи да се у оквиру Заузетости Учионице забележи које Одељење је заузето у одређено време.

Увођењем Заузетости Учионице као посебан тип ентитета омогућили смо да се све информације о заузетостима како учионице, тако и одељења налазе на једном месту. На основу повезаности између Часа и Заузетости Учионице, можемо проверити и заузетост Запосленог, јер као што се види на слици 3.7 у оквиру Часа је садржана информација о Запосленом који је евидентирао тај Час.



*Слика 3.8 – Део дијаграма модела шеме базе података за смештање информација о заузетости учионица и одељења*

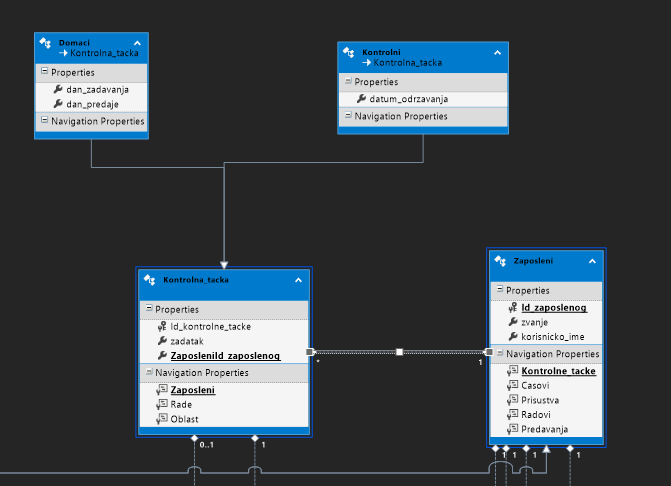
Иако се час планира између осталог и према одељењу којем ће бити држан, то не мора да значи да су сви ученици тог одељења били на одређеном часу. С обзиром на то да је битна информација да ли је ученик био на часу, направљен је посебан тип ентитета Присуство који ће садржати информацију о томе који је Ученик био на ком Часу. На слици испод (слика 3.9) можемо видети поменуте типове ентитета и везе између њих.



*Слика 3.9 – Део дијаграма модела шеме базе података за смештање информација о присутности ученика на часу*

### 3.2.6 Део модела шеме базе података за смештање података о контролним тачкама

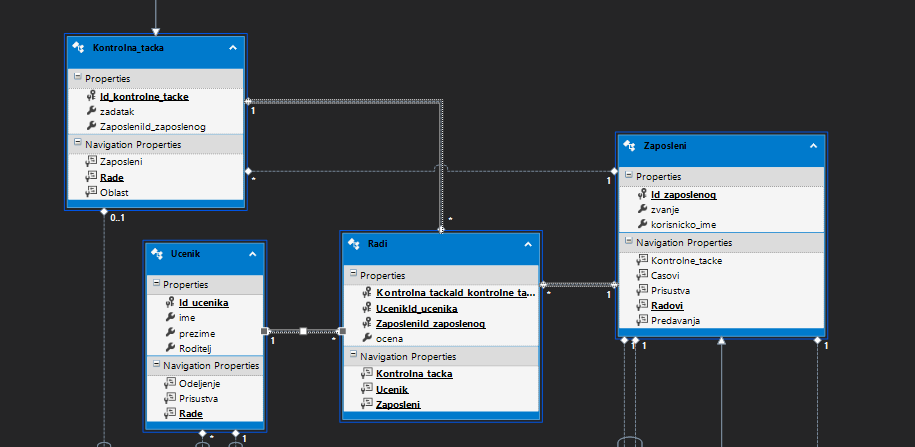
Друга веома битна функционалност која треба бити омогућена запосленима је да евидентирају контролне и домаће задатке које планирају, као и да оцене радове ученика. На слици 3.10 приказани су типови ентитета Контролна тачка, Контролни и Домаћи као и начин на који смо повезали те типове са типом ентитета Запослени.



*Слика 3.10 – Део дијаграма модела шеме базе података за смештање података о контролним и домаћим задацима*

На слици можемо приметити да смо повезали тип ентитета Контролна тачка са типом ентитета Запослени, узимајући у обзир да је Контролна тачка родитељски тип ентитета у односу на Контролни и Домаћи, у навигационој особини Контролне\_тачке типа ентитета Запослени имаћемо информације и о Домаћима и о Контролнима у складу са правилима наслеђивања.

Ради праћења постигнутог успеха ученика у школи, неопходна је евиденција његових оцена које је остварио на контролним, односно домаћим задацима. Ради бележења ових информација направљен је нови тип ентитета Ради. Овај тип ентитета и његову повезаност са Учеником, Запосленим и Контролном тачком види се на слици 3.11.



*Слика 3.11 – Део дијаграма модела шеме базе података за смештање података о контролним и домаћим задацима које ученик ради*

На слици се такође види да смо Ради повезали само са Контролном тачком, али као што је већ речено, у складу са правилима наслеђивања, ово нам је довољно да у навигациону особину Контролна\_тачка сместимо било Контролни, било Домаћи. Помоћу особине *ocena* у типу ентитета Ради бележимо коју је оцену ученик добио на задатом контролном, то јест домаћем.

## 3.3 Базе података генерисане из модела шема базе података

Модели шема података описани у претходном поглављу представљају основу за моделовање базе података која је искоришћена за генерисање саме базе података помоћу интегрисане програмске подршке у оквиру Microsoft Visual Studi-a 2019, а преко пакета Entity Framework 6.4.4. Поменути пакет омогућава да се на основу дефинисаног модела изгенерише база података, али и класе у C# које представљају типове ентитета.

На основу типова ентитета изгенерисане су табеле у бази и класе које представљају те ентитете. Асоцијације између типова ентитета су у оквиру класе претворене у пропертије, док су у бази креирани страни кључеви према правилима о преносу страног кључа.

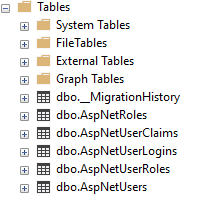
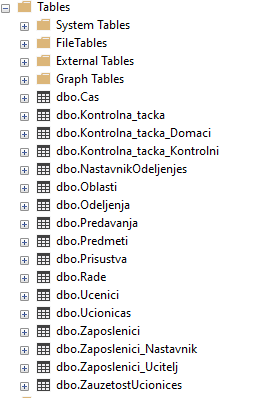
За потребе информационог система при моделовању базе података коришћен је и Code First и Model First приступ. Разлике између ова два приступа су у томе што се у Code First приступу моделују класе и на основу њих креира база, док се у Model First приступу, као што назив и сугерише, прво креира модел базе података и онда, на основу њега, сама база података. Ефективно, базе приликом генерисања прате иста правила, тако на пример асоцијације у моделу, као што је и речено, у бази простају страни кључеви, а у класама пропертији. С друге стране пропертији у класама, при генерисању базе на основу њих, се такође претварају у стране кључеве сходно правилима о преносу страног кључа.

Разлог за коришћење оба приступа моделовању базе података, огледа се у могућностима које класе имплементиране у пакету Microsoft.AspNet.Identity поседују. Наиме, класе у оквиру овог пакета се најчешће користе за чување корисничких информација, што је и разлог зашто су управо ове класе искоришћене за моделовање базе података о корисницима. Сам пакет нуди напредне функционалности, као што су чување лозинке у кодираном облику и додељивање улога корисницима у систему. Ове особине поменутог пакета биће детаљније објашњене када буде описивано апликативно решење информационог система.

### 3.3.1 Изглед база података генерисаних за потребе информационог система

У оквиру овог дела поглавља биће представљен изглед самих база података које су генерисане према претходно наведеним правилима. Како су за потребе информационог система направљене две базе, једна за потребе складиштења корисничких информација и друга за смештање информација о самој школској установи, обе од њих биће презентоване примерима.

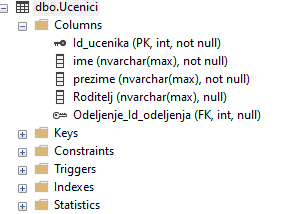
За потребе хостовања базе података коришћен је *Microsoft SQLExpress* сервер, а као СУБП коришћен је *Microsoft SQL Server Management Studio*. СУБП је употребљен за креирање база података и за потребе прегледа начина на који су подаци смештани у базу података. На основу модела шеме базе података који су претходно описани креиране су табеле у базама (слика 3.12), док је платформа Microsoft Visual Studio 2019 омогућила рад са подацима у бази.

*Слика 3.12 –Изглед табела у базама података коришћене у информационом систему, у оквиру Microsoft SQL Server Management Studio-а*

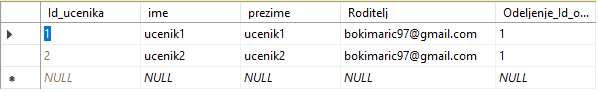
На слици 3.12 види се једина разлика приликом креирања база података помоћу Code First и Model First приступа. Та разлика огледа се у додатној табели која није изгенерисана на основу нашег модела, а под називом \_MigrationHistory. Додатна табела служи са чување метаподатака о миграцијама које су у оквиру развојног окружења направљене.

Као што је већ речено, генерисање базе података прати иста правила без обзира на приступ који користимо, стога ће изглед начина на који се информације у оквиру једне табеле смештају бити приказан на најрепрезентативнијој табели у информационом систему, а то је табела Ucenici. На слици испод (слика 3.13) види се табела са информацијама вазаним за њу које су организоване у потфолдере.



*Слика 3.13 –Изглед табеле Ucenici, која је одабрана да репрезентује начин на који СУБП складишти информације о табелама у оквиру базе*

У оквиру фолдера *Columns* садржане су информације о колонама које табела садржи. Као што се на слици 3.13 види примарни кључ табеле Id\_ucenika визуелно је означен сликом обојеног кључа, док је страни кључ према табели *Odeljenja* означен необојеним кључем. Обележја ime, prezime и Roditelj представљени су сликом која симболизује колону у табели. Треба напоменути, да је разлог постојања страног кључа у оквиру ове табеле моделована асоцијација између ентитета Ucenik и Odeljenjе у оквиру Entity Framework Designer-а. Такође, назив страног кључа је аутоматски генерисан на основу правила Entity Framework Designer-а.



*Слика 3.14 –Излед података смештених у табелу Ucenici*

На слици изнад (слика 3.14) види се начин на који СУБП уписује податке у оквиру табеле Ucenici. Значење података које смештамо у базу описани су у следећем поглављу.

## 3.4 Опис ентитета и асоцијација у моделима

У овом поглављу представљен је табеларни приказ свих ентитета у базама података који су коришћени у информационом систему и описи асоцијација које ентитет има ка другим ентитетима.

#### Ентитет AspNetRoles

Ентитет *AspNetRoles* моделује могуће улоге корисника. Свака улога дефинисана помоћу торке у оквиру табеле означава посебан тип корисника у оквиру информационог система.

*Табела 3.1 – Обележја у ентитету АspNetRoles*

|  |  |  |  |
| --- | --- | --- | --- |
| **Назив обележја** | **Тип података** | **Nullable** | **Опис обележја** |
| Id | String | Не | Примарни кључ |
| Name | String | Не | Назив улоге |
| Discriminator | String | Не | Опис улоге |

#### Ентитет AspNetUser

Ентитет *AspNetUser* моделује корисника информационог система. Ентитет се налази у оквиру пакета *Microsoft.AspNet.Identity*, и проширен је обележјима ime, prezime и FirstLogin. Подаци из табеле настале од овог ентитета користе се за бележење корисничких информација.

*Табела 3.2 – Обележја у ентитету AspNetUser*

|  |  |  |  |
| --- | --- | --- | --- |
| **Назив обележја** | **Тип података** | **Nullable** | **Опис обележја** |
| Id | String | Не | Примарни кључ |
| ime | String | Да | Име корисника |
| prezime | String | Да | Презиме корисника |
| Email | String | Да | Е-мејл корисника |
| EmailConfirmed | Boolean | Не | Е-мејл је потврђен |
| PasswordHash | Hash Strng | Да | Лозинка |
| SecurityStamp | Hash String | Да | Рандом вредност која се мења кад год се лозинка промени |
| PhoneNumber | String | Да | Телефон запосленог |
| PhoneNumberConfirmed | Boolean | Не | Телефон је потврђен |
| TwoFactorEnabled | Boolean | Не | Двострука аутентификација је омогућена |
| LockoutEndDateUtc | Date Time | Да | Датум до ког корисник нема приступ систему |
| LockoutEnabled | Boolean | Не | Закључавање корисника ван система је омогућено |
| AccessFailedCount | Integer | Не | Број неуспелих покушаја пријаве на систем |
| UserName | String | Не | Корисничко име |
| FirstLogin | Boolean | Не | Означава прву пријаву на систем. |

AspNetUser ентитет има асоцијацију са *AspNetRoles* ентитетом која представља тип корисника у информационом систему.

#### Ентитет Cas

Ентитет *Cas* моделује школски час који запослени заказује помоћу информационог система.

*Табела 3.3 – Обележја у ентитету Cas*

|  |  |  |  |
| --- | --- | --- | --- |
| **Назив обележја** | **Тип података** | **Nullable** | **Опис обележја** |
| Id\_casa | Integer | Не | Примарни кључ |
| OblastId\_oblasti | Integer | Да | Страни кључ |
| ZaposleniId\_zaposlenog | Integer | Не | Страни кључ |
| ZauzetostUcionice\_Id\_zauzetosti | Integer | Не | Страни кључ |

Као што се може и приметити на основу присуства страних кључева у табели изнад, Ентитет *Cas* има асоцијације према ентитетима *Oblast*, *Zaposleni* и *ZauzetostUcionice.*

#### Ентитет Kontrolna\_tacka

Ентитет *Kontrolna\_tacka* представља контролне тачке које запослени може да испланира. У оквиру система је направљена да буде надтип ентитетима *Kontrolni* и *Domaci*.

*Табела 3.4 – Обележја у ентитету Kontrolna\_tacka*

|  |  |  |  |
| --- | --- | --- | --- |
| **Назив обележја** | **Тип података** | **Nullable** | **Опис обележја** |
| Id\_kontrolne\_tacke | Integer | Не | Примарни кључ |
| zadatak | String | Не | Опис контролне тачке |
| ZaposleniId\_zaposlenog | Integer | Не | Страни кључ |
| OblastId\_oblasti | Integer | Не | Страни кључ |

Ентитет *Kontrolna\_tacka* има асоцијације према ентитетима *Zaposleni* и *Oblast.*

#### Ентитет Domaci

Ентитет *Domaci* представља домаћи задатак који запослени може да евидентира. Његов надтип је ентитет *Kontrolna\_tacka* и од тог ентитета се наслеђује примарни кључ.

*Табела 3.5 – Обележја у ентитету Domaci*

|  |  |  |  |
| --- | --- | --- | --- |
| **Назив обележја** | **Тип података** | **Nullable** | **Опис обележја** |
| Id\_kontrolne\_tacke | Integer | Не | Примарни кључ |
| dan\_zadavanja | Date Time | Не | Датум када се домаћи задаје |
| dan\_predaje | Date Time | Не | Датум када се домаћи предаје на оцењивање |

#### Ентитет Kontrolni

Ентитет *Kontrolni* моделује контролни задатак који запослени може да дефинише.

*Табела 3.6 – Обележја у ентитету Kontrolni*

|  |  |  |  |
| --- | --- | --- | --- |
| **Назив обележја** | **Тип података** | **Nullable** | **Опис обележја** |
| Id\_kontrolne\_tacke | Integer | Не | Примарни кључ |
| datum\_odrzavanja | Date Time | Не | Датум када се контролни одржава |

#### Ентитет NastavnikOdeljenje

Ентитет представља везу између ентитета *Nastavnik* и *Odeljenje*. Овај ентитет је моделован како би се у њему бележиле информације о наставницима који предају одређеним одељењима.

*Табела 3.7 – Обележја у ентитету NastavnikOdeljenje*

|  |  |  |  |
| --- | --- | --- | --- |
| **Назив обележја** | **Тип података** | **Nullable** | **Опис обележја** |
| NastavnikId\_zaposlenog | Integer | Не | Примарни кључ |
| OdeljenjeId\_odeljenja | Integer | Не | Примарни кључ |
| Razredni | Boolean | Не | Наставник је одељењу разредни |

Ентитет има асоцијације са ентитетима *Nastavnik* и *Odeljenje,* с тим што су пренети страни кључеви искоришћени за прављење комбинованог примарног кључа

#### Ентитет Predmet

Ентитет *Predmet* моделује предмете који се предају у школи.

*Табела 3.8 – Обележја у ентитету Predmet*

|  |  |  |  |
| --- | --- | --- | --- |
| **Назив обележја** | **Тип података** | **Nullable** | **Опис обележја** |
| Id\_predmeta | Integer | Не | Примарни кључ |
| razred | Short Integer | Не | Разред којем је предмет намењен |
| naziv | String | Не | Назив предмета |
| broj\_oblasti | Short Integer | Не | Број области предмета |

#### Ентитет Oblast

Ентитет *Oblast* представља област једног предмета.

*Табела 3.9 – Обележја у ентитету Oblast*

|  |  |  |  |
| --- | --- | --- | --- |
| **Назив обележја** | **Тип података** | **Nullable** | **Опис обележја** |
| Id\_oblasti | Integer | Не | Примарни кључ |
| naziv | String | Не | Назив области |
| PredmetId\_predmeta | Integer | Не | Страни кључ |

Овај ентитет има једну асоцијацију и то према ентитету *Predmet* за који је везан.

#### Ентитет Ucenik

Ентитет садржи обележја која описују ученике који похађају школску установу.

*Табела 3.10 – Обележја у ентитету Ucenik*

|  |  |  |  |
| --- | --- | --- | --- |
| **Назив обележја** | **Тип података** | **Nullable** | **Опис обележја** |
| Id\_ucenika | Integer | Не | Примарни кључ |
| ime | String | Не | Име ученика |
| prezime | String | Не | Презиме ученика |
| Roditelj | String | Да | Е-мејл родитеља ученика |
| OdeljenjeId\_odeljenja | Integer | Да | Страни кључ |

Ентитет има асоцијацију према ентитету *Odeljenje* која симболизује припадност одељењу.

#### Ентитет Odeljenje

Ентитет моделује одељења школске установе.

*Табела 3.11 – Обележја у ентитету Odeljenje*

|  |  |  |  |
| --- | --- | --- | --- |
| **Назив обележја** | **Тип података** | **Nullable** | **Опис обележја** |
| Id\_odeljenja | Integer | Не | Примарни кључ |
| razred | Short Integer | Не | Разред одељења |
| UciteljId\_zaposlenog | Integer | Да | Страни кључ |

*Odeljenje* има асоцијацију према ентитету *Ucitelj.*

#### Ентитет Predavanje

Ентитет представља предавање које један запослени може да евидентира у оквиру информационог система.

*Табела 3.12 – Обележја у ентитету Predavanje*

|  |  |  |  |
| --- | --- | --- | --- |
| **Назив обележја** | **Тип података** | **Nullable** | **Опис обележја** |
| Id\_predavanja | Integer | Не | Примарни кључ |
| opis | String | Не | Кратак опис планираног предавања |
| datum\_odrzavanja | Date Time | Не | Датум када се предавање одржава |
| OblastId\_oblasti | Integer | Не | Страни кључ |
| ZaposleniId\_zaposlenog | Integer | Не | Страни кључ |

*Predavanje* има асоцијације према *Zaposlenom* који га планира и *Oblasti* за коју је везан.

#### Ентитет Prisustvo

Ентитет служи за бележење да је ученик присуствовао одређеном часу.

*Табела 3.13 – Обележја у ентитету Prisustvo*

|  |  |  |  |
| --- | --- | --- | --- |
| **Назив обележја** | **Тип података** | **Nullable** | **Опис обележја** |
| UcenikId\_ucenika | Integer | Не | Примарни кључ |
| CasId\_casa | Integer | Не | Примарни кључ |
| ZaposleniId\_zaposlenog | Integer | Не | Страни кључ |

*Prisustvo* има асоцијације према ентитетима *Ucenik, Cas* и *Zaposleni,* при чему су пренесени страни кључеви из часа и ученика искоришћени за формирање примарног кључа ове табеле.

#### Ентитет Radi

Ентитет садржи обележја која повезују одређену контролну тачку коју је ученик радио и у којој се садржи информација о оцени коју је на њој добио.

*Табела 3.14 – Обележја у ентитету Radi*

|  |  |  |  |
| --- | --- | --- | --- |
| **Назив обележја** | **Тип података** | **Nullable** | **Опис обележја** |
| Kontrolna\_tackaId\_kontrolne\_tacke | Integer | Не | Примарни кључ |
| UcenikId\_ucenika | Integer | Не | Примарни кључ |
| ZaposleniId\_zaposlenog | Integer | Не | Примарни кључ |
| ocena | Short Integer | Не | Оцена коју је ученик добио на контролној тачци |

Ентитет има асоцијације према ентитетима *Kontrolna\_tacka, Ucenik* и *Zaposleni,* при чему су пренесени страни кључеви из ових асоцијација искоришћени за формирање примарног кључа.

#### Ентитет Ucionica

Ентитет представља учионицу која у школској установи постоји.

*Табела 3.15 – Обележја у ентитету Ucionica*

|  |  |  |  |
| --- | --- | --- | --- |
| **Назив обележја** | **Тип података** | **Nullable** | **Опис обележја** |
| Id\_ucionice | Integer | Не | Примарни кључ |
| broj\_ucenika | Integer | Не | Број ученика које учионица смешта |
| naziv | String | Не | Назив учионице |

#### Ентитет Zaposleni

Ентитет моделује запосленог у школској установи и служи да помоћу њега и његових подтипова имамо повезане информације о забележеном раду запосленог у оквиру информационог система.

*Табела 3.16 – Обележја у ентитету Zaposleni*

|  |  |  |  |
| --- | --- | --- | --- |
| **Назив обележја** | **Тип података** | **Nullable** | **Опис обележја** |
| Id\_zaposlenog | Integer | Не | Примарни кључ |
| zvanje | String | Не | Звање које је запослени стекао |
| korisnicko\_ime | String | Не | Корисничко име запосленог у систему |

#### Ентитет Nastavnik

Ентитет је подтип ентитета *Zaposleni* од ког наслеђује примарни кључи повезује *Nastavnika* са *Predmetom* који предаје преко асоцијације између та два ентитета.

*Табела 3.17 – Обележја ентитета Nastavnik*

|  |  |  |  |
| --- | --- | --- | --- |
| **Назив обележја** | **Тип података** | **Nullable** | **Опис обележја** |
| Id\_zaposlenog | Integer | Не | Примарни кључ |
| PredmetId\_predmeta | Integer | Да | Страни кључ |

#### Ентитет Ucitelj

Ентитет моделује учитеља у школској установи и представља подтип ентитета *Zaposleni* од ког наслеђује примарни кључ.

*Табела 3.18 – Обележја у ентитету Ucitelj*

|  |  |  |  |
| --- | --- | --- | --- |
| **Назив обележја** | **Тип података** | **Nullable** | **Опис обележја** |
| Id\_zaposlenog | Integer | Не | Примарни кључ |

#### Ентитет ZauzetostUcionice

Ентитет служи за бележење информација о заузетости учионице.

*Табела 3.19 – Обележја у ентитету ZauzetostUcionice*

|  |  |  |  |
| --- | --- | --- | --- |
| **Назив обележја** | **Тип података** | **Nullable** | **Опис обележја** |
| Id\_zauzetosti | Integer | Не | Примарни кључ |
| UcionicaId\_ucionice | Integer | Не | Страни кључ |
| OdeljenjeId\_odeljenja | Integer | Не | Страни кључ |
| datum | Date Time | Не | Датум када је учионица заузета |
| pocetak | Time | Не | Почетак часа од 45 минута када је учионица заузета |

*ZauzetostUcionice* има асоцијације са ентитетима *Odeljenje* и *Ucionica* што се у табели види присуством страних кључева.

# 4. Опис апликативног решења

У овом поглављу биће описано апликативно решење уз кратак освит на архитектуру софтвера и пакете коришћене при реализацији информационог система. При презентовању апликативног решења функционалности које ће бити објашњене поделићемо према типу корисника.

Апликативно решење је реализовано као рачунарска апликација, где је кориснички интерфејс имплементиран коришћењем WPF графичког подсистема, водећи се MVVM програмским обрасцем.

## 4.1 Софтверска архитектура

При развијању информационог система у апликативном решењу је коришћена сервисно оријентисана архитектура која се заснива на подели имплементираних функционалности на групе према одређеним особинама, а у циљу повећања скалабилности и поновне употребљивости самог програмског кода[2].

Функционалности развијене за потребе информационог система су подељене на четири логички независна пројекта у оквиру *solution*-а посматране апликације. Од пројеката у апликацији постоје:

* *AuthTesting* – управља корисничким информацијама како на апликативном новоу тако и на нивоу базе података
* *OsnovnaSkola* – садржи шему модела базе података на основу које је изгенерисана база података нужна за рад запослених у школи, као и класе за приступ подацима у бази
* *OsnovnaSkolaPL* – у оквиру овог пројекта издвојена је пословна логика апликације
* *OsnovnaSkolaUI* – дефинише кориснички графички интерфејс

Размена информација између претходно описаних пројеката реализована је употребом WCF канала и то на следећи начин: пројекат *OsnovnaSkolaUI* представља клијента у односу на пројекат *OsnovnaSkolaPL,* који репрезентује сервер и где је дефинисана пословна логика која користи елементе за приступ бази дефинисане у оквиру пројекта *OsnovnaSkola*; такође је отворена нова конекција између *OsnovnaSkolaPL*, који представља, у овом случају, клијента у односу на пројекат *AuthTesting*, који има улогу сервера и приступа бази са корисничким информацијама.

Ради реализовања сервисно оријентисане архитектуре, односно у циљу постизања жељене одвојености програмске логике на различите пројекте у којима ће се обрађивати специфични захтеви и давати приступ само одређеним подацима од значаја, употребљен је управо WCF који обезбеђује захтеване функционалности без великих измена у начину моделовања и руковања класама. WCF комуникација је развијена управо за потребе комуникације између сервиса у сервисно оријентисаним апликацијама [3]. Она такође пружа могућност сигурног протока информација кроз мрежу коришћењем добро знаних стандарда као што је на пример SSL протокол.

## 4.2 Коришћени пакети

Microsoft Visual Studio развојно окружење подржава механизам преко ког програмери могу развијати, делити и искористити постојеће имплементације које су скупљене у „пакете“ компајлираног кода. Тај механизам се назива *NuGet,* и он дефинише како су пакети креирани, хостовани и употребљавани у оквирима .*NetFramework*-а [4].

За потребе апликативног решења додата су и искоришћена два *NuGet* пакета под називима: *„Microsoft.AspNet.Identity“* верзија 2.2.3 и *„Microsoft.EntityFramework“* верзија 6.4.4.

Функционалности које пакет EntityFramework пружа огледају се у раду са базом. У претходним поглављима више пута је поменут Entity Framework Designer, алат искоришћен за креирање дијаграма модела шеме базе података о запосленима. Овај алат се налази у оквиру поменуте верзије EntityFramework пакета. Пакет такође омогућује писање класа за рад са базом, као и креирање базе података Code First приступом.

Из пакета AspNet.Identity употребљене су класе *IdentityUser* и *IdentityRole*, за моделовање информација о корисницима и њиховим улогама у систему респективно. *IdentityUser* класа садржи већину информација које су неопходне за моделовање корисника у оквиру неког информационог система, због чега је употребљена као родитељска класа у односу на класу *ApplicationUser* преко које смо моделовали корисника, где су њој додати пропертији: ime, prezime и FirstLogin.

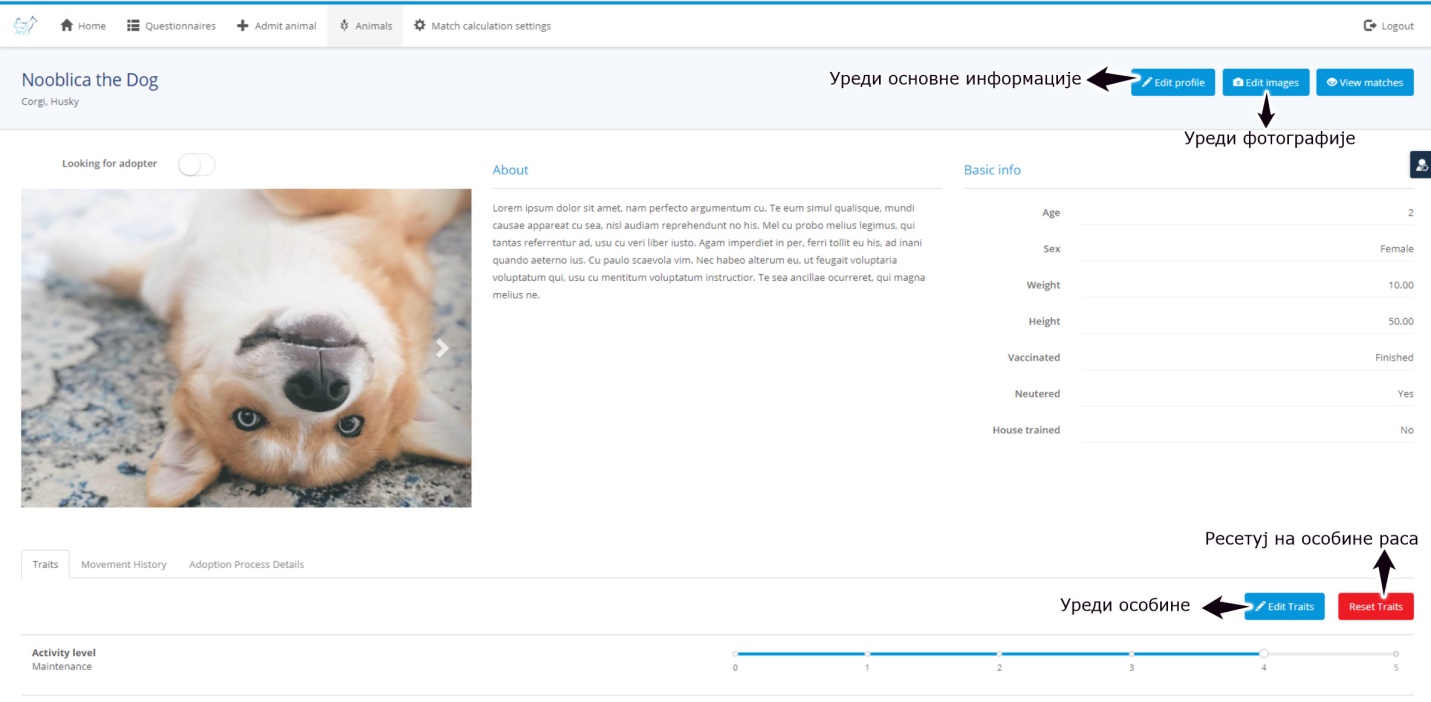
Оно што је од великог интереса за развој апликације је начин на који се лозинка складишти у бази, а што је функционалност AspNet.Identity пакета. Наиме, пакет нам за приступ бази пружа већ имплементирану класу под називом *UserManager,* а која при упису корисничких информација лозинку смешта у енкриптованом облику добијеном *hesh*-ирањем унете лозинке. Такође, помоћу те класе је могуће верификовати корисника преко методе *CheckPassword*, која од аргумената прима објекат типа *ApplicationUser* (којег проналазимо на основу унетог корисничког имена) и string променљиву која репрезентује лозинку коју корисник уноси. Сама метода *hash*-ира прослеђену лозинку и пореди је са вредношћу која већ постоји у бази, ако се унета и постојећа лозинка не поклапају метода враћа false.

При упису нових корисничких информација у базу лозинка се *hash*-ира аутомаским позивом тзв. *„Key Derivation Function“* која је названа *Rfc2898DeriveBytes(String, Int32, Int32)* [5] и која је уграђена у сам .NET Framework. Иста функција се позива приликом позива *CheckPassword* методе *UserManager* класе, с тим што се у оквиру те функције врши и провера да ли се лозинка прослеђена *CheckPassword* методи после *hash*-ирања поклапа са оном вредношћу која стоји у бази у колони *PasswordHash* за одговарајућег корисника.

Као што се може приметити, једном унета лозинка се из базе никада неће повући у свом читком облику, већ ће се употребљавати само њена *hash*-ирана вредност. Ове и бројне друге могућности које AspNet.Identity пакет пружа, као што су на пример: двофактроска аутентификација, пријава на систем преко других сервиса, забрана кориснику приступ систему и сл. су разлог зашто је пакет искоришћен при моделовању базе корисничких информација. Требало би напоменути да наведене функционалности нису имплементиране у оквиру постојећег апликативног решења али да представљају одличну основу за потенцијална проширења. Такође, постоје и новије верзије пакета са додатним проширењима, али оне не могу да се користе у оквиру .NetFramework платформе.

## 4.2 Уређивање профила животиње

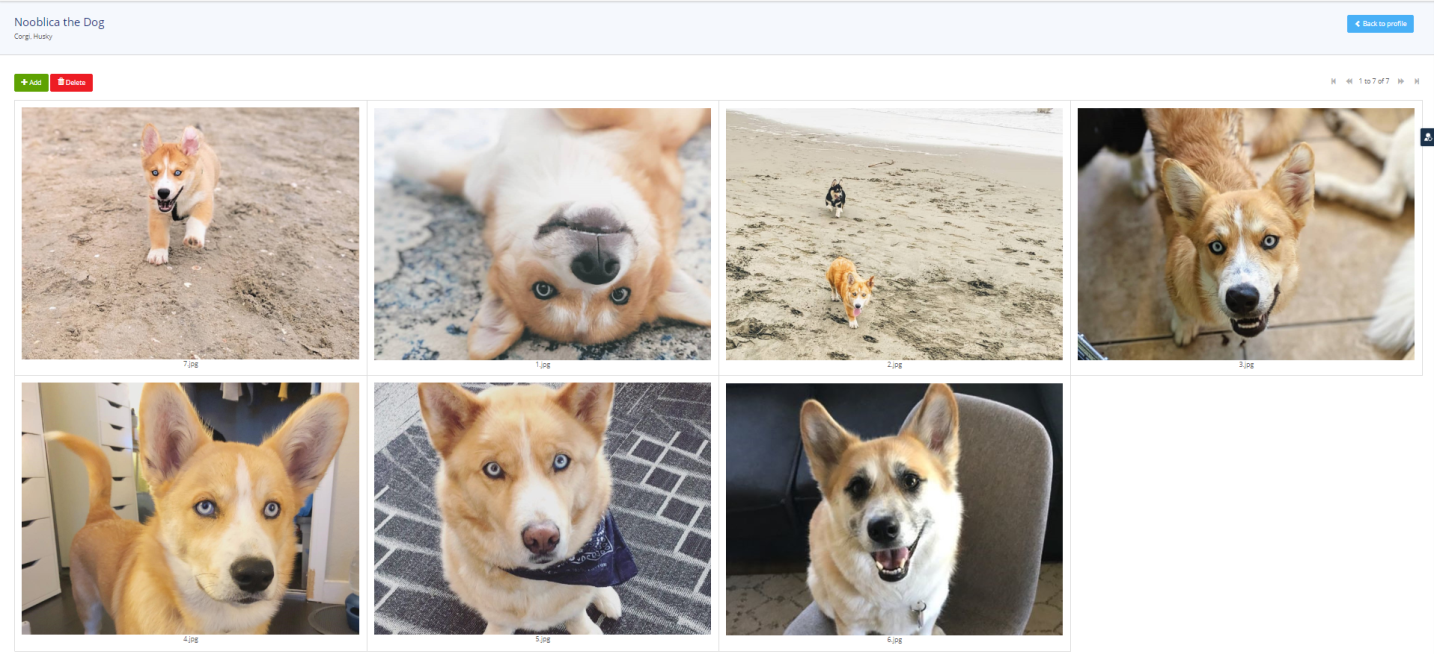
Уређивање профила животиње омогућено је радницима задуженим за пријем и бригу о животињама, као и радницима задуженим за процес усвајања. Уређивање профила подељено је на сегменте како би форме биле фокусиране на конкретан задатак и краће. Приступ формама за уређивање профила врши се са самог профила (*слика 4.5*).



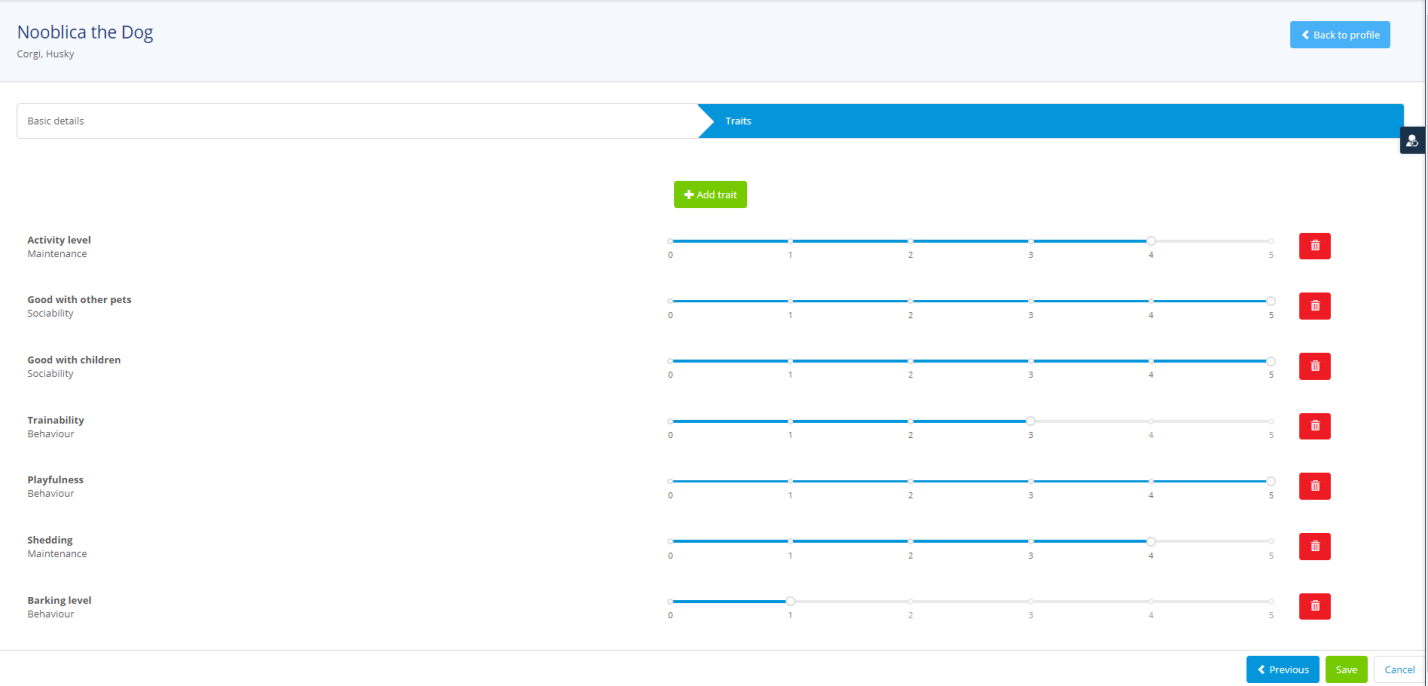
*Слика 4.5 – Профил животиње са означеним дугмићима за приступ формама за уређивање*

Кликом на дугме *Edit profile* радницима се отвара форма за уређивање основних информација о животињи. У оквиру ове форме могуће је уредити име, расу, пол и остале основне информације о животињи. Форма има исти облик као форма за уређивање основних информација о животињи у оквиру пријема животиње.

Кликом на дугме *Edit images* отвара се форма за уређивање фотографија животиње (*слика 4.6*). Кликом на дугме *Add* у оквиру ове форме радник додаје нову слику, док клик на дугме *Delete* омогућава брисање селектоване слике. Приликом брисања могуће је селектовати више слика одједном. Приликом додавања нове слике, радник истовремено може да дода само једну слику. Могуће побољшање ове форме представља омогућавање истовременог додавања више слика.

 *Слика 4.6 – Уређивање фотографија животиње*

За уређивање особина радницима су на располагању два начина. Кликом на дугме *Reset traits* покреће се аутоматско рачунање испољености особина животиње на основу њених раса. Овакав начин доделе особина нарочито је користан за животиње које су тек приспеле у прихватилиште и немају додељене особине. Други начин за уређивање особина животиње је ручном доделом особина. Клик на дугме *Edit traits* отвара форму за уређивање особина (*слика 4.7*). Могуће је уклонити постојеће и додати нове особине животињи. За сваку додељену особину врши се избор изражености особине, где вредност 0 значи да особина није изражена, док вредност 5 представља врло изражену особину. Кликом на дугме *Save* радник може да сачува измене, док кликом на дугме *Cancel* може да одустане од измена.

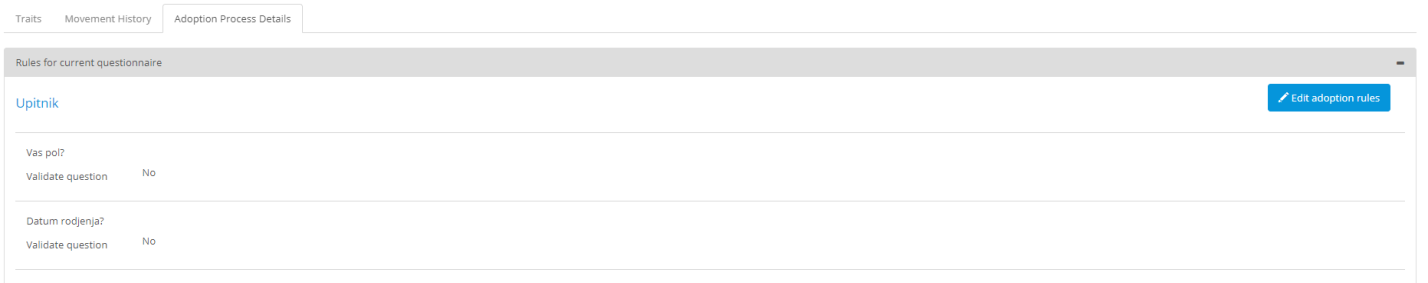


*Слика 4.7 – Форма за уређивање особина животиње*

Радници задужени за процес усвајања могу да измене жељене вредности одговара на питања из упитника о личним подацима усвојитеља. Уколико животиња нема постављене жељене вредности, креирање се врши кликом на дугме *Create rules* у оквиру таба *Adoption Process Details* (*слика 4.8*), док се отварање форме за уређивање постојећих жељених вредности врши кликом на дугме *Edit* *rules* у оквиру истог таба (*слика 4.9*). Клик на било које од ова два дугмета преусмерава радника на форму за уређивање жељених одговара (*слика 4.10*).

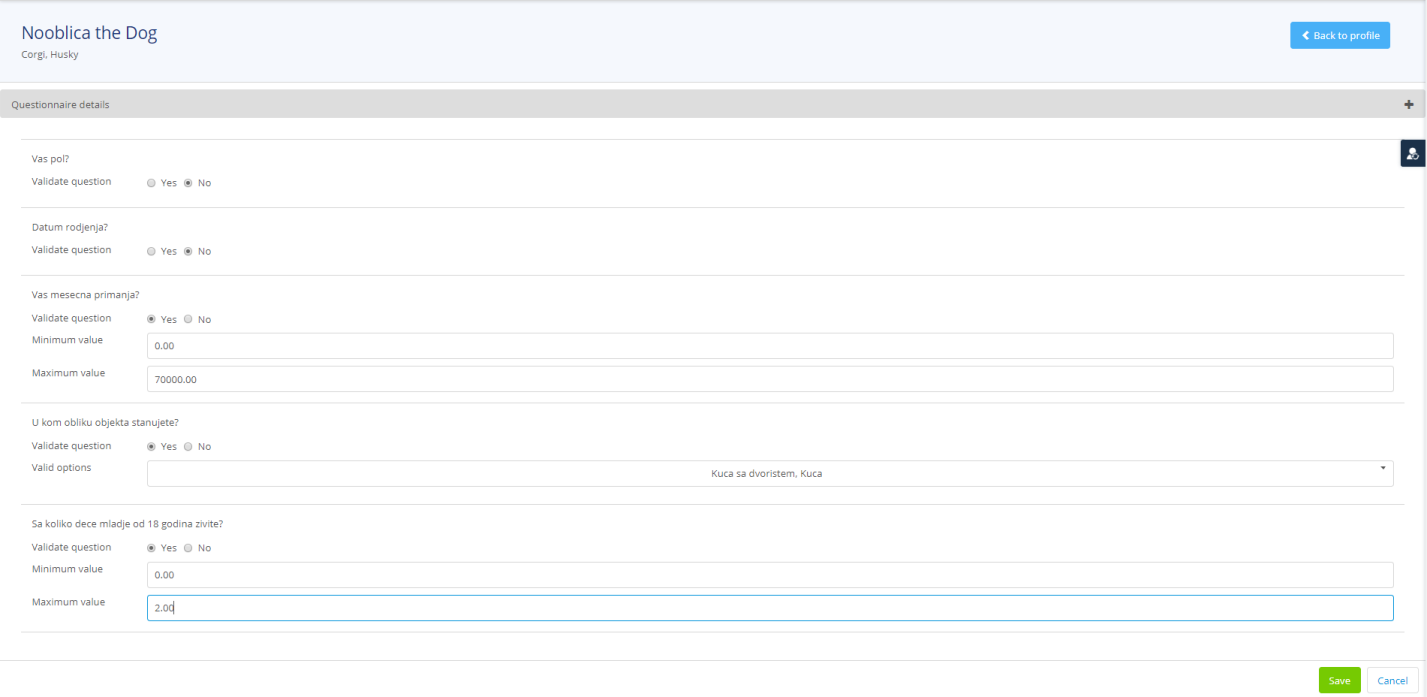


*Слика 4.8 – Дугме за креирање правила*



*Слика 4.9 – Дугме за уређивање постојећих правила*

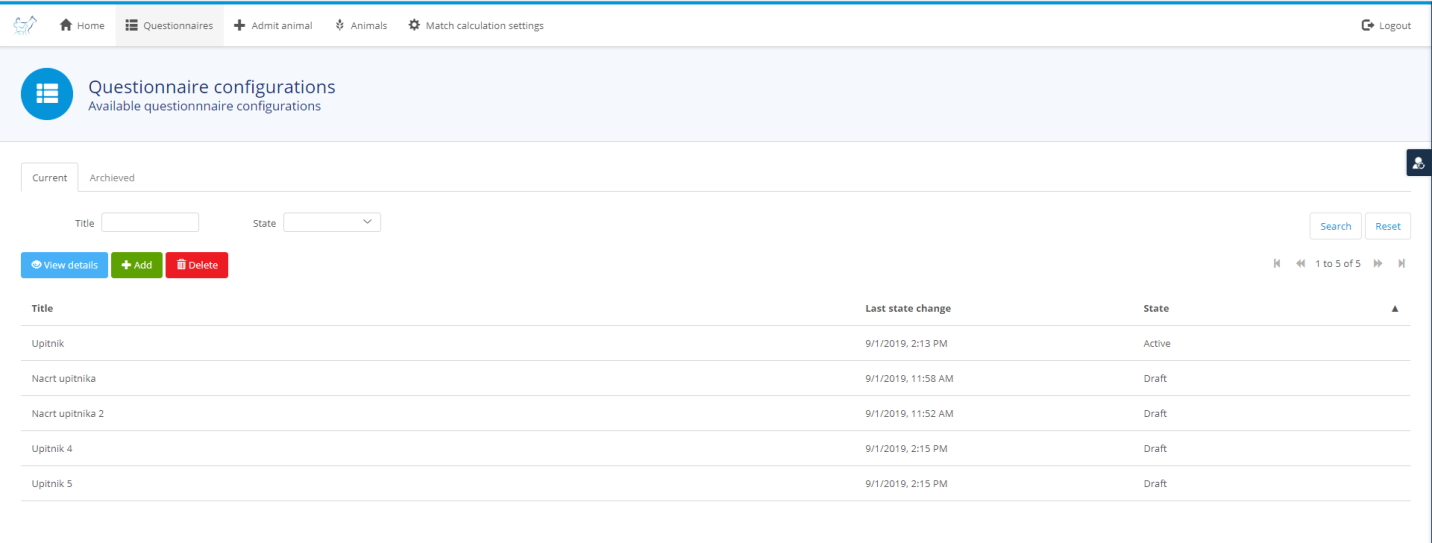
За свако од питања упитника радник може да изабере да ли се оно бодује или не. Уколико је изабрано да се питање бодује, радник бира валидне одговоре. За питања на која се одговара бројем или датумом избор валидних одговара врши се избором опсега. За питања на која се одговара селекцијом једног од понуђених одговара, радник може да изабере све опције које се сматрају валидним одговор. Кликом на дугме *Save* радник може да сачува измене, док кликом на дугме *Cancel* може да одустане од измена.



*Слика 4.10 – Форма за уређивање вредности жељених одговара у упитнику*

## 4.4 Креирање упитника

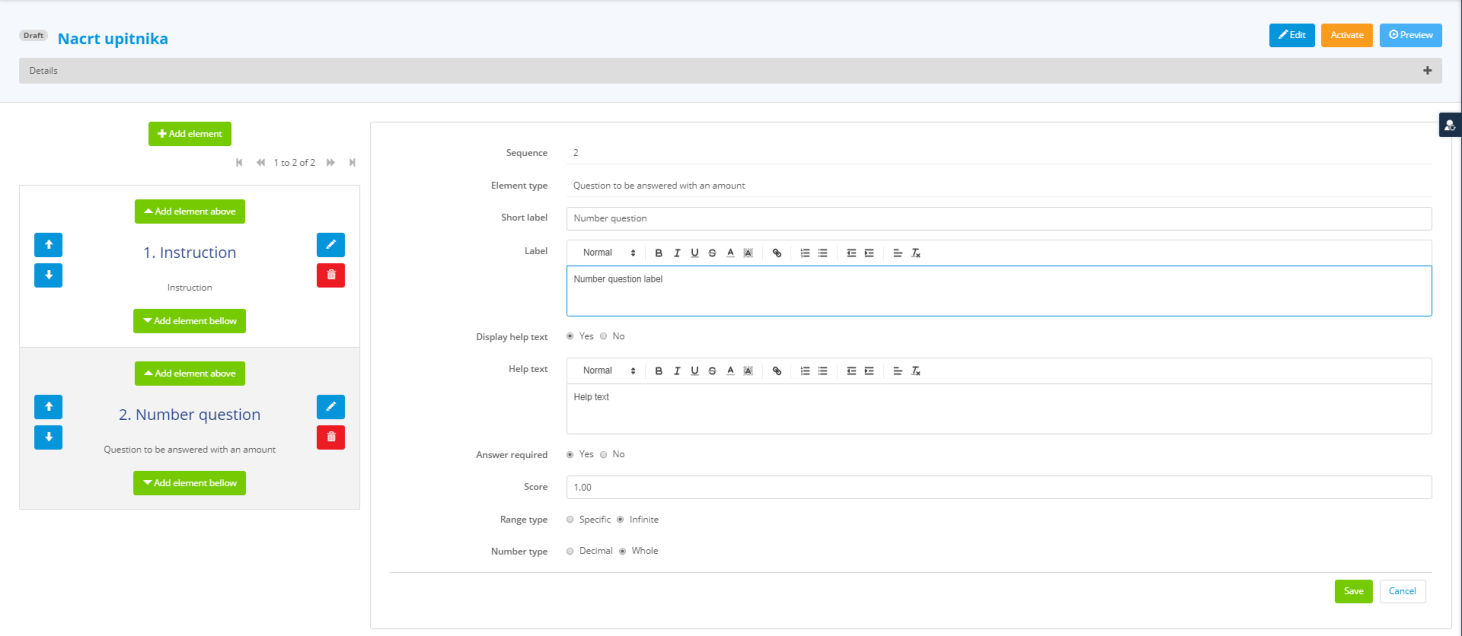
Радник у процесу усвајања може да креира упитник са питањима о личним информацијама усвојитеља. У једном тренутку може да постоји само један активан упитник, док нацрта и архивираних упитника може да буде произвољно много. Преглед свих упитника подељен је у два таба (*слика 4.11*). У оквиру првог таба приказан је активни упитник и нацрти упитника. Активни упитник ће се увек налазити на првом месту у оквиру табеле свих упитника. У оквиру другог таба приказани су архивирани упитници. У оквиру таба са активним упитником и нацртима, радник може да дода нови упитник кликом на дугме Add, или да обрише неки од постојећих нацрта кликом на дугме *Delete*. Двостурки клик или селекција и клик на дугме *View details* преусмерава на страницу са детаљима о упитнику.



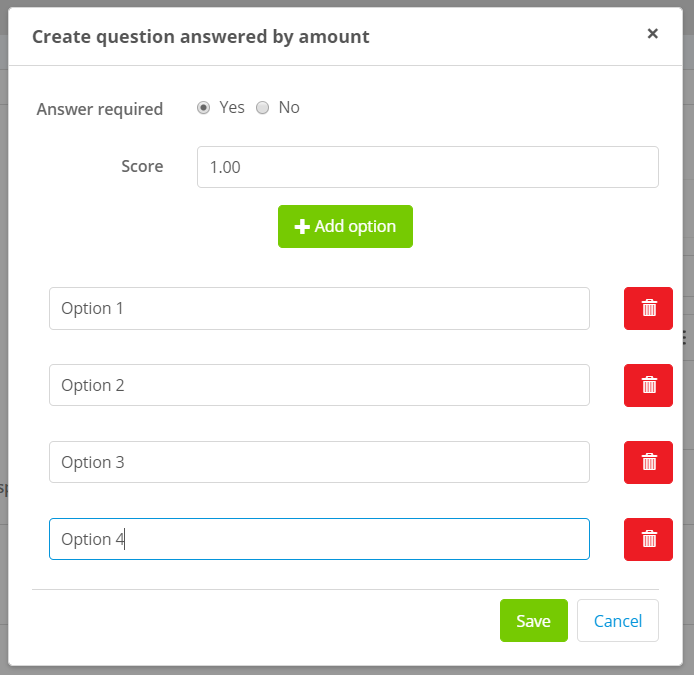
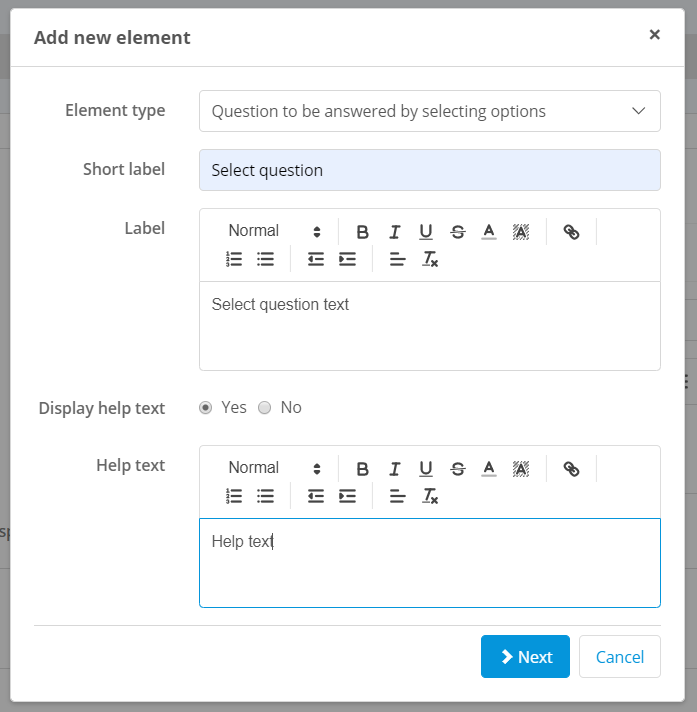
*Слика 4.11 – Преглед свих упитника*

У оквиру странице са детаљима о упитнику приказана су питања (*слика 4.12*). Уколико упитник има статус нацрта, омогућено је додавање, брисање, уређивање и промена редоследа питања. Додатно, омогућен је преглед изгледа упитника кликом на дугме *Preview* и промена статуса упитника у активан кликом на дугме *Activate*.

У левом делу прозора приказан је преглед свих питања. У десном делу прозора приказане су додатне информације о селектованом питању. За упитнике са статусом нацрта, приказ у десном делу прозора је у облику форме и омогућава измену питања. За упитнике са статусом активан или архивиран приказ у десном делу прозора је текстуалан. Приказана поља зависе од типа питања.

*Слика 4.12 – Приказ детаља о упитнику*

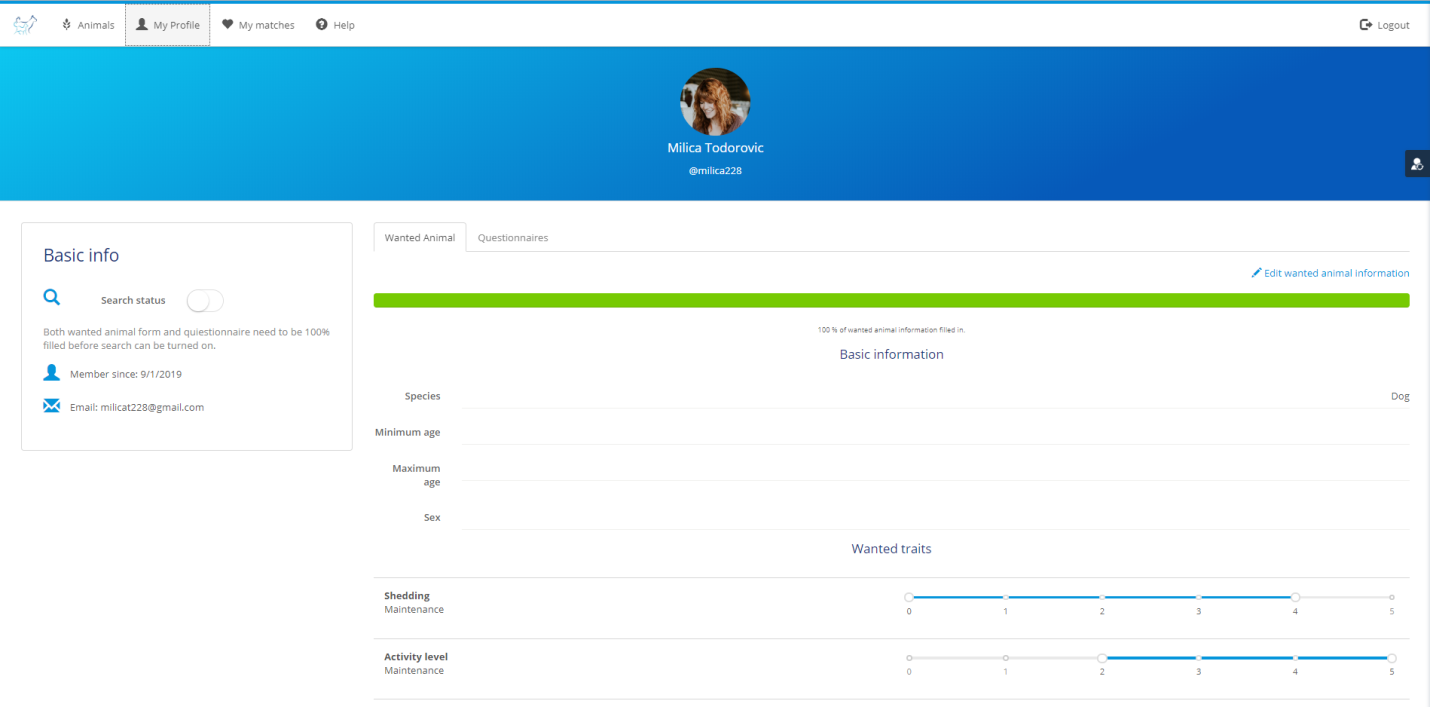
Клик на дугме *Add element* отвара форму за додавање новог елемента упитника у оквиру блокирајућег модалног дијалога (*слика 4.13*). У зависности од типа елемента, форма може имати једну или две стране. На првој страни форме врши се избор типа елемента – инструкција или неко од питања, као и унос текста, избор да ли постоји помоћни текст и унос помоћног текста. Уколико се изабере тип елемента инструкција, друга страна форме не постоји. За све типове питања постоји друга страна форме, где се уносе информације потребне за формирање тог типа питања. Поља са додатним информацијама зависе од типа питања, а свим питањима заједничко је поље *Score* које представља број бодова које питање носи приликом анализе упитника.



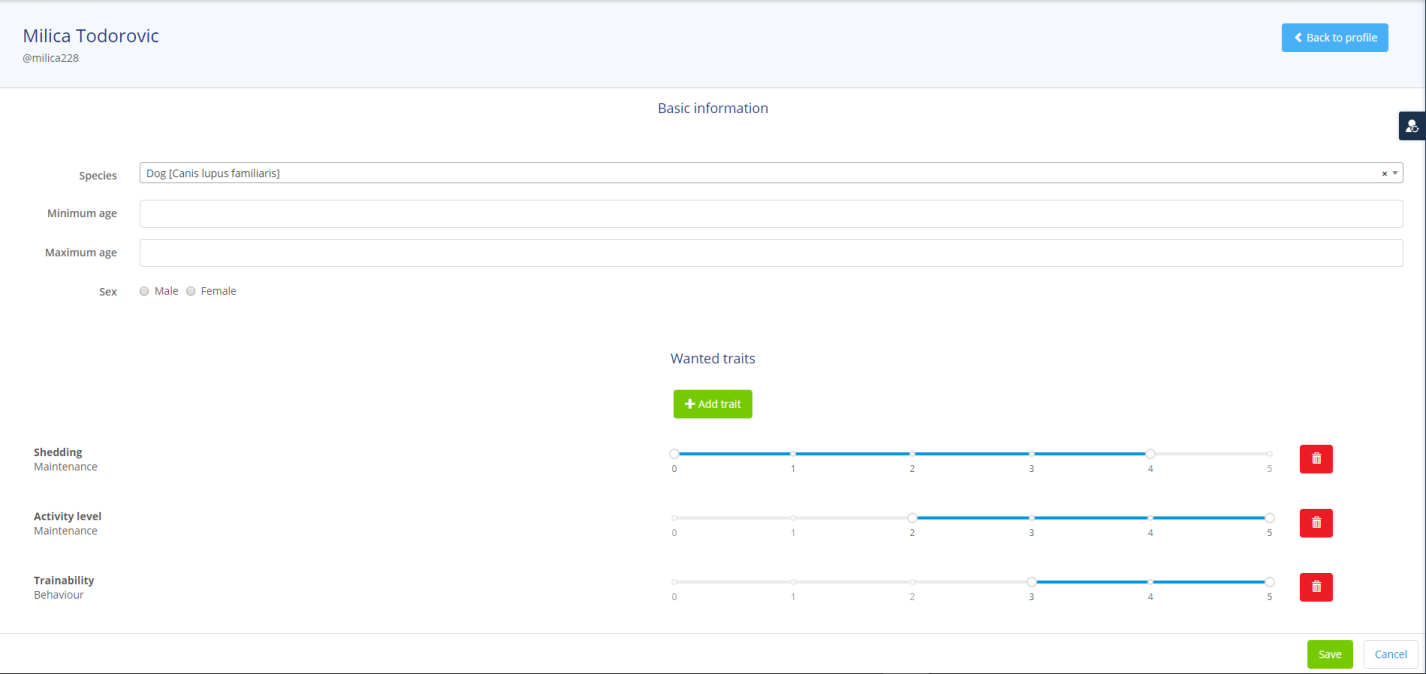
*Слика 4.13 – Блокирајући модални дијалог за додавање новог питања*

## 4.5 Попуњавање упитника

Да би покренуо потрагу животиње, потенцијални усвојитељ мора претходно да попуни упитник о жељеној животињи и упитник о себи. Приступ упитницима потенцијални усвојитељ може да изврши са своје профилне странице (*слика 4.14*). У оквиру своје профилне странице усвојитељ такође може да вити проценат попуњености обе упитника и своје тренутне одговоре.

 *Слика 4.14 – Профилна страница потенцијалног усвојитеља*

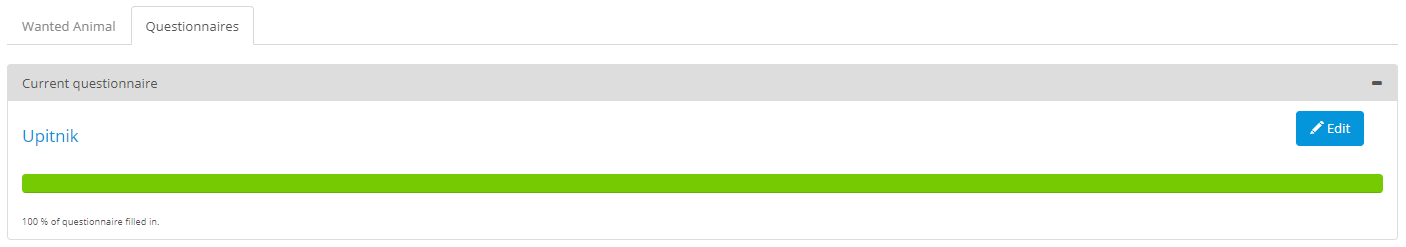
Клик на дугме *Edit wanted animal information* води на страницу за попуњавање упитника о жељеној животињи (*слика 4.15*). У оквиру овег упитника усвојитељ може да изабере врсту, пол и године жељене животиње. Даље, усвојитељ може да уреди које особине жели да животиња има и у којој мери оне да буду изражене. Додавање нове особине врши се кликом на дугме *Add trait*. За сваку изабрану особину усвојитељ може да изабере доњу и горњу границу жељене изражености. Брисање изабране особине врши се кликом на дугме са иконицом канте за смеће. Чување попуњене форме може се извршити кликом на дугме *Save*, а одустајање од промена кликом на дугме *Cancel*.



*Слика 4.15 – Упитник о жељеној животињи*

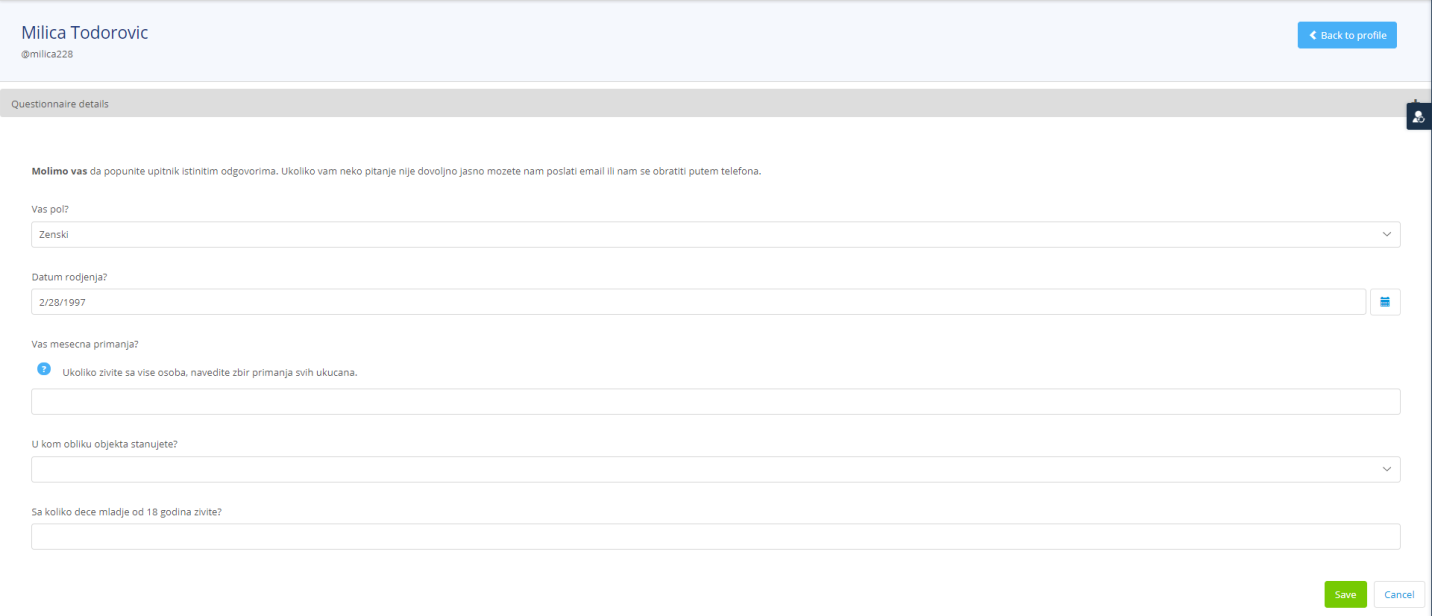
Уколико потенцијални усвојитељ први пут жели да попуњава упитник са личним информацијама, то може да уради кликом на дугме *Start questionnaire* у оквиру таба *Questionnaires* (*слика 4.16*). Уколико усвојитељ жели да уреди одговоре, то може да уради кликом на дугме *Edit* у истом табу (*слика 4.17*).

*Слика 4.16 – Дугме за покретање попуњавања упитника о личним подацима*



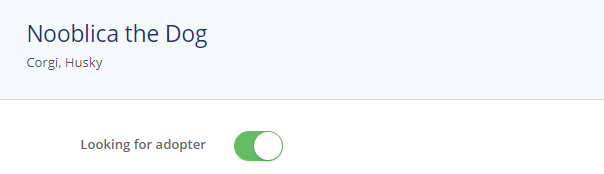
*Слика 4.17 – Дугме за покретање измене већ постојећих одговара на упитник о личним подацима*

Клик на било који од два наведена дугмета преусмерава усвојитеља на страницу у оквиру које може да уреди своје одговоре на питања (*слика 4.18*). У оквиру упитника приказана су редом питања и инструкције. Облик поља за унос одговора зависи од типа питања. Уколико за питање постоји задата помоћ, она је исписана испод самог питања, поред знака упитника. Усвојитељ може у било ком тренутку да сачува тренутно стање упитника кликом на дугме *Save*, или да поништи измене кликом на дугме *Cancel*.

*Слика 4.18 – Упитник о личним информацијама*

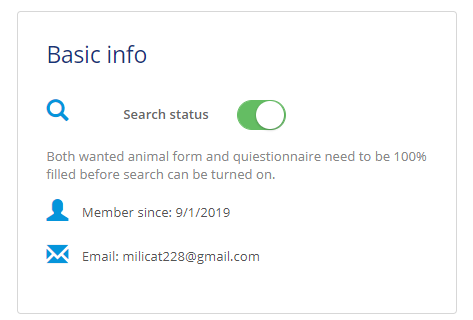
## 4.6 Покретање потраге

Покретање потраге за усвојитељем неке животиње врши радник задужен за процес усвајања. Покретање потраге врши се помоћу клизача у горњем левом углу профилне странице животиње (*слика 4.19*). Када се потрега покрене, систем проверава степен поклапања особина животиње са особинама које су усвојитељи који активно траже животињу исказали као жељене. Разматрају се се само оне комбинације животиње и усвојитеља за које већ није пронађено поклапање. Животиње са активираном потрагом разматрају се сваки пут када неки усвојитељ покрене своју потрагу. Уређивање профила није могуће за животиње које траже усвојитеља.



*Слика 4.19 – Дугме за покретање потраге за усвојитељем*

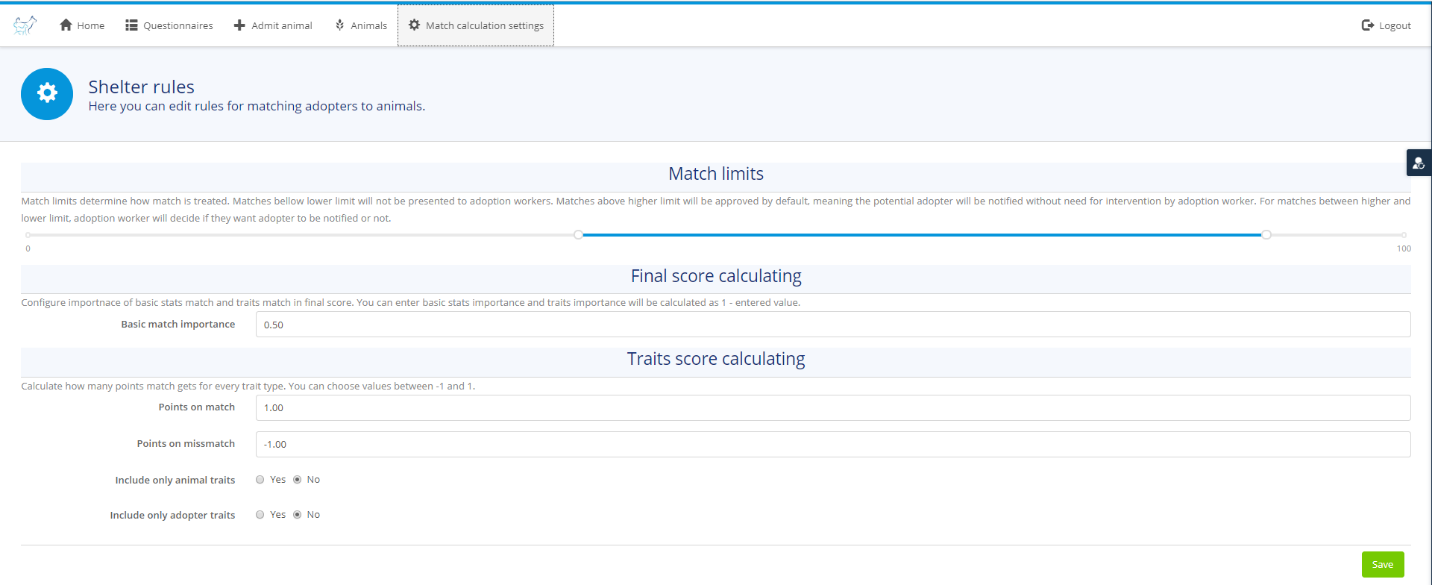
Покретање потраге за животињом врши потенцијални усвојитељ. Покретање се може извршити помоћу клизача у горњем левом углу профила усвојитеља (*слика 4.20*), онда када је усвојитељ у целости попунио оба упитника. При покретању потраге, систем проверава степен поклапања жељених особина са особинама животиња којима се тражи усвојитељ, а које већ немају поклапање са датим усвојитељем. Усвојитељи са активираном потрагом разматрају се сваки пут када се покрене потрага за усвојитељем за неку животињу. Уређивање одговора није могуће за усвојитеље код којих је активирана потрага за животињом.



*Слика 4.20 – Дугме за покретање потраге за животињом*

## 4.6 Начин рачунања поклапања

Приликом рачунања поклапања користе се оба упитника попуњена од стране усвојитеља, особине животиње, жељене вредности одговара за животињу и коефицијенти за рачунање поклапања дефинисани на нивоу прихватилишта. Уређивање коефицијената за рачунање поклапања могу да врше радници задужени за усвајање (*слика 4.21*). Промене коефицијената не утичу на постојећа поклапања.

*Слика 4.21 – Форма за промену коефицијената за рачунање поклапања*

Да би се разматрало поклапање између усвојитеља и животиње, потребно је да животиња буде оне расе која је изабрана у оквиру упитника о жељеној животињи. Уколико су у оквиру овог упитника наведене године и/или жељени пол животиње, они такође морају одговарати, да би се уопште анализирало поклапање. Уколико су наведени услови испуњени рачуна се поклапање животиње и усвојитеља по особинама и по упитнику са личним информацијама. Укупан резултат поклапања се рачуна као:

где је *q* коефицијент значаја поклапања личних информација. Сва укупна поклапања која су мања од доње границе се одбацују и не чувају у бази. Сва поклапања између доње и горње границе се чувају у бази и дају на увид раднику задуженом за процес усвајања, који може да их прихвати или одбије. Сва усвајања изнад горње границе се прихватају и приказују се и потенцијалном усвојитељу који даље може да их прихвати или одбије.

Поклапање особина рачуна се поређењем особина које животиња има и особина које је усвојитељ означио као жељене. Особина која се нађе у обе листе, а да је вредност коју животиња испољава унутар жељене границе наведене од стране усвојитеља сматра се поклопљеном. Уколико се особина нашла у обе листе, али је вредност задана за животињу изван граница заданих од стране усвојитеља, сматра се да је дошло до неслагања. Особине које испољава само животиња, а усвојитељ их није навео, као и особине које је усвојитељ навео, а нису наведене код животиње се одвојено гледају и бодују у складу са коефицијентима заданим за њих. Поклапање се рачуна као:

Број бодова за особине се рачуна по формули:

где *а* представља број бодова додељених за поклопљене особине, *b* број бодова за неусаглашене особине, *c* број бодова за особине које само животиња испољава, а *d* број бодова за особине које је само усвојитељ навео.

Могући број бодова рачуна се по формули:

где се под укупним бројем особина подразумева број јединствених особина у обе листе, а коефицијент *а* представља број бодова додељених за поклопљене особине.

За рачунање поклапања личних информација користе се попуњен упитник са личним информацијама и жељене вредности дефинисане за сваку животињу. Валидирају се и бодују само она питања која су означена за бодовање у правилима за дату животињу. Одговор се сматра валидним уколико је у заданим границама по правилима, или празан. Бодови који се додељују дефинишу се приликом дефинисања питања у упитнику појединачно за свако питање. Поклапање личних информација се рачуна по следећој формули:

Број бодова се рачуна према формули:

где коефицијент *i* итерира кроз питања на које је одговор валидан.

Могући број бодова се рачуна као:

где коефицијент *j* итерира кроз сва питања која је требало валидирати и бодовати за дату животињу.

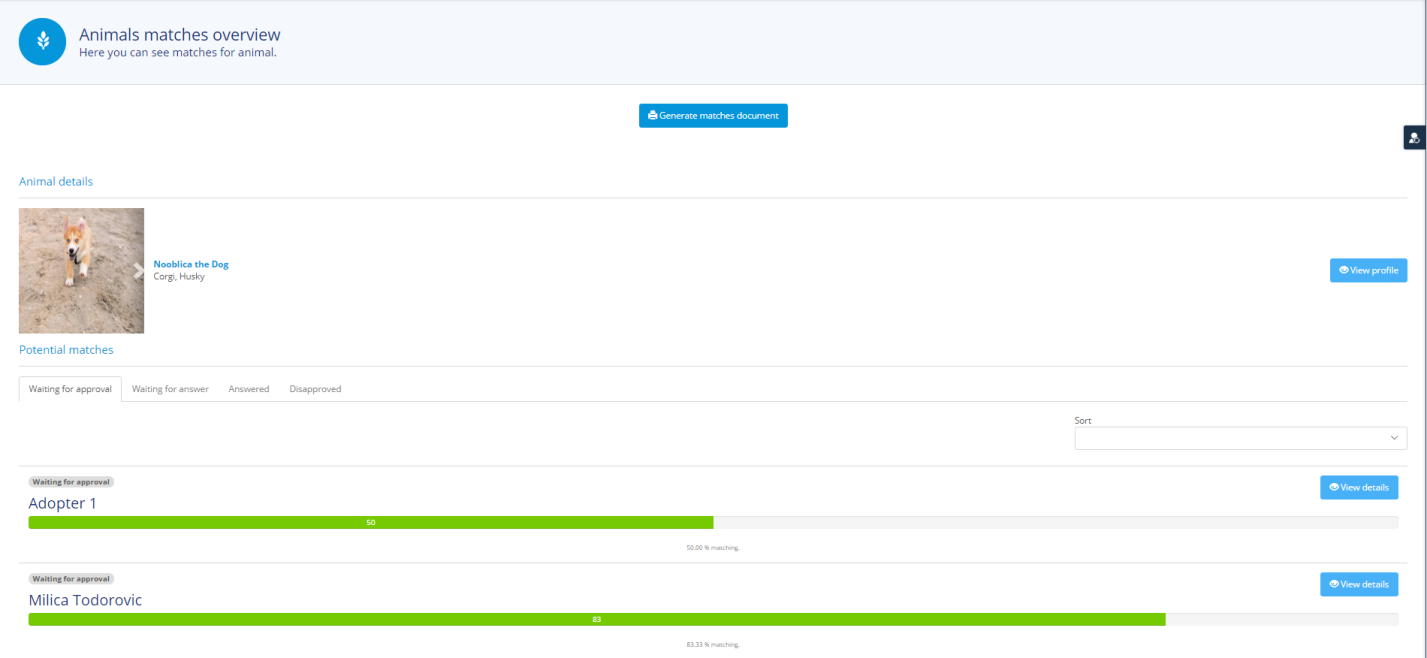
Наведене формуле представљају један од предлога и нису једини могући начин рачунања поклапања. Једна од могућих измена је дефинисање броја бодова које питање носи за сваку животињу појединачно. Надаље, алгоритам се може променити на начин да се задају почетне вредности за бодовање, које затим систем сам прилагођава учећи на основу прихватања и одбијања пронађених поклапања. На пример, уколико се особина *x* није поклопила за особу и животињу, али је укупно поклапање прихваћено од стране радника и усвојитеља, смањује се број бодова који се одузима када се особина *x* не поклопи.

Даље измене алгоритма за рачунање поклапања представља примена вештачке интелигенције за употребу проналажења животиња сличних онима које усвојитељ означи као њему одговарајуће. За оваква проширења потребно је да информациони систем има претходно прикупљене податке на основу кога би алгоритам учио. Информациони систем описан у овом раду може да помогне у прикупљању таквих података.

Предности изабраног алгоритма за проналажење поклапања су једноставност за имплементацију и могућност презентовања резултата на човеку јасан начин.

## 4.7 Приказ поклапања и промена статуса

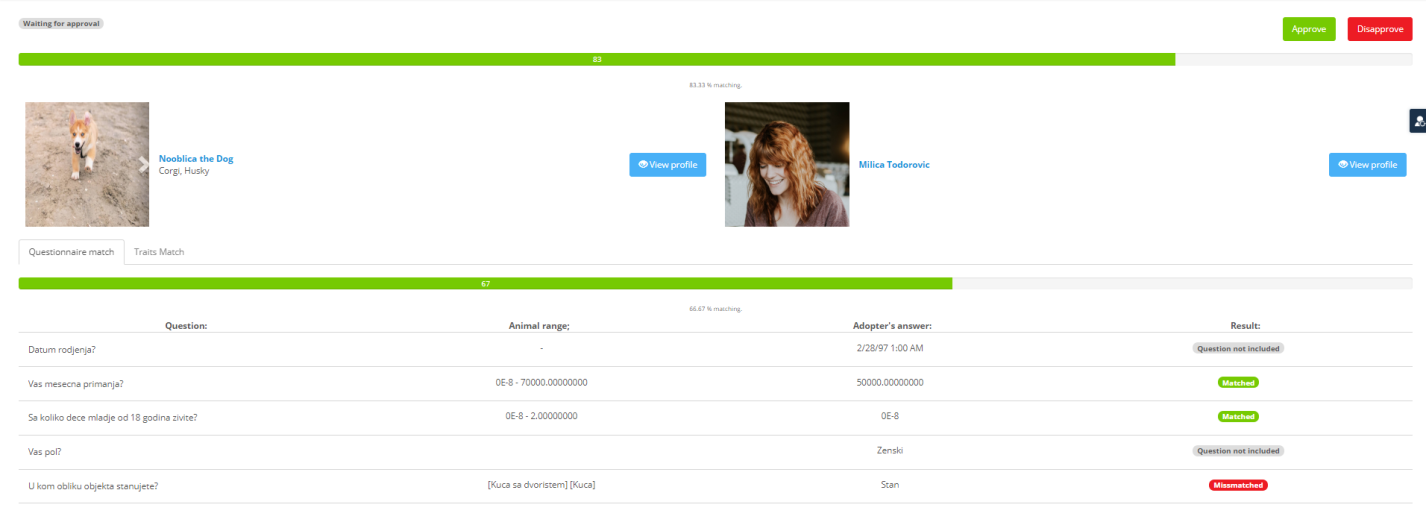
Радник на процесу усвајања може да види сва поклапања која је систем пронашао за неку животињу. Потенцијални усвојитељ може да види поклапања која је систем пронашао за њега, а да су одобрена од стране радника задужених за процес усвајања. У оквиру прегледа свих поклапања за животињу (*слика 4.22*), поклапања су раздвојена по статусима. Могуће је извршити сортирање поклапања по проценту укупног поклапања, поклапања особина или поклапања личних информација. На сличан начин организован је и преглед свих поклапања пронађених за усвојитеља. Увид у детаље поклапања могућ је кликом на дугме *View details*.

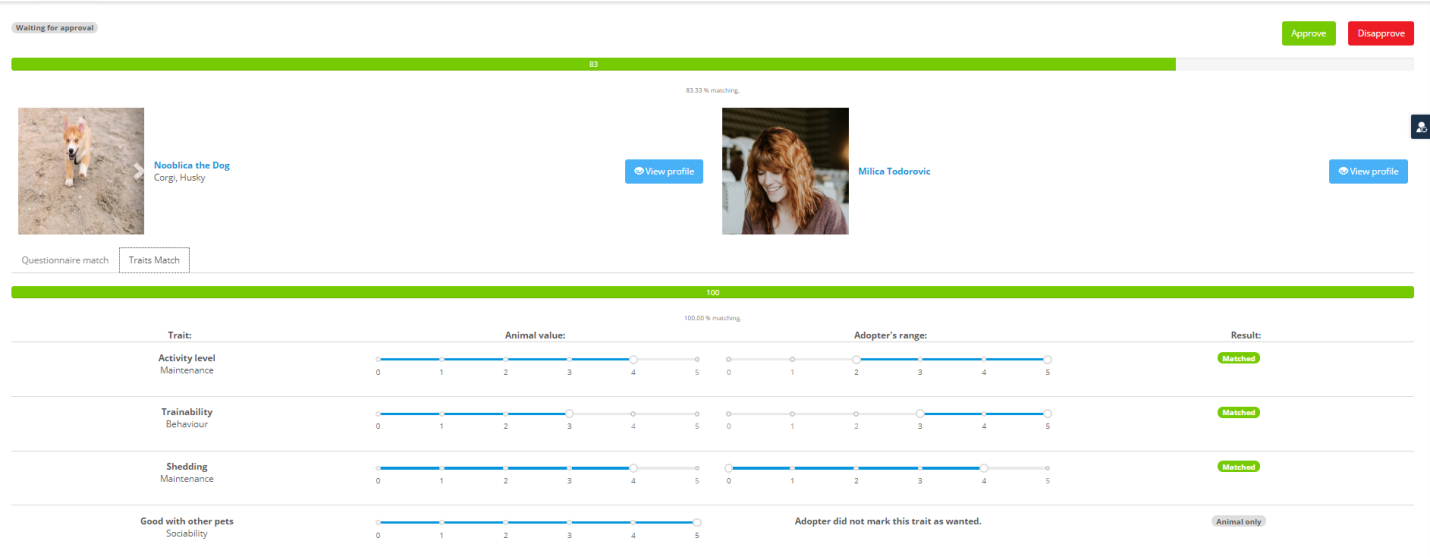
*Слика 4.22 – Преглед свих поклапања за животињу*

Страница са детаљима поклапања доступна је радницима задуженим за процес усвајања. Уколико је поклапање одобрено од стране радника, онда је страница за његов преглед доступна и потенцијалном усвојитељу.

На самом врху странице налази се приказ статуса поклапања, као и приказ укупног поклапања. Ако поклапање чека на одобрење радника, на врху странице се налазе и дугмићи којима он може да прихвати или одбије поклапање. Уколико поклапање чека на одговор усвојитеља, онда ће се њему у горњем десном углу налазити дугмићи којима може да одбије или прихвати поклапање.

У оквиру странице са детаљима поклапања дати су линкови ка профилу животиње и линкови ка профилу усвојитеља. Надаље, у оквиру ове странице приказано је разлагање поклапања. У оквиру таба *Questionnaire match* приказано је поклапање одговора на питања о личним информацијама у односу на жељене вредности одговора за животињу (*слика 4.23*). У оквиру таба *Traits Match* приказано је поклапање особина које животиња има са особинама које је усвојитељ означио као жељене (*слика 4.24*). Овакав начин приказа показује корисницима на који начин је израчунато поклапање, па тиме олакшава одлуку да ли поклапање да одбију или прихвате.

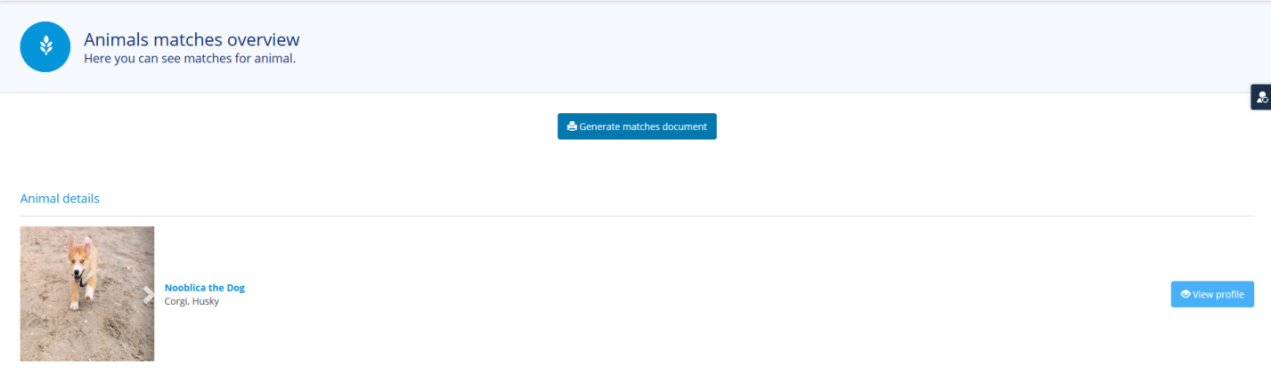
*Слика 4.23 – Детаљи о преклапању, преглед поклапања личних особина*



*Слика 4.24 – Детаљи о преклапању, преглед поклапања особина*

## 4.8 Креирање и штампање извештаја о поклапањима за животињу

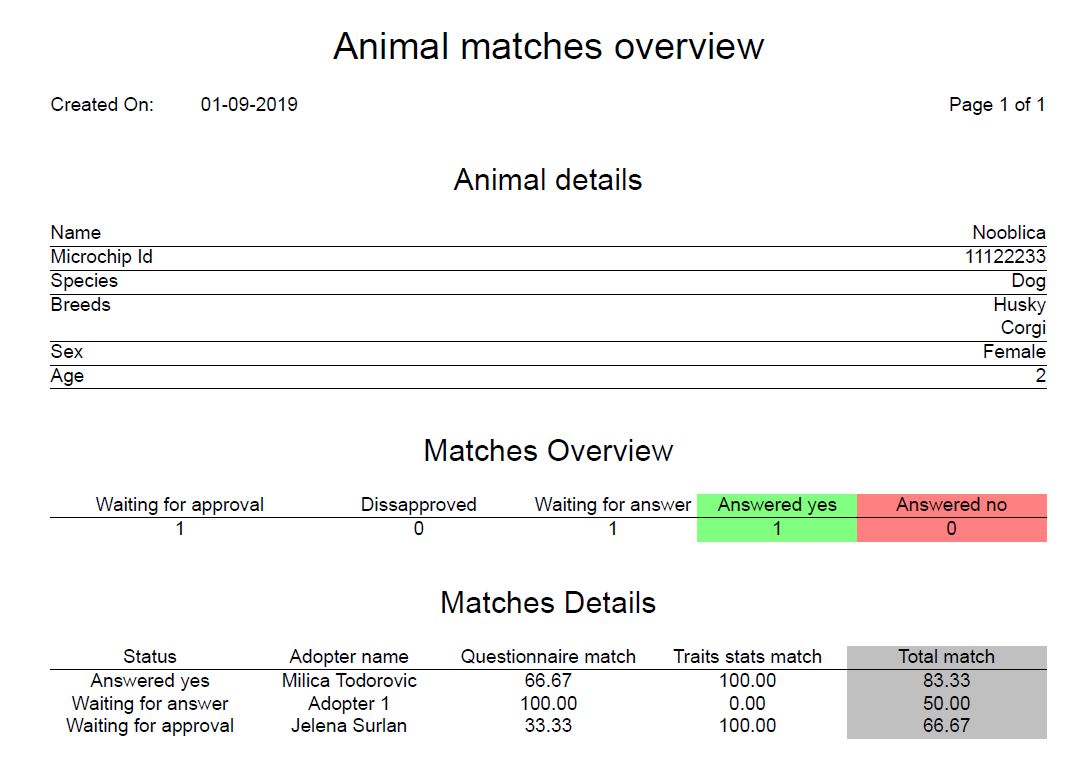
Извештај о пронађеним поклапањима за животињу омогућава преглед статистике поклапања за неку животињу. Овакав извештај може да служи да олакша избор усвојитеља или као помоћ при статистичким анализама. Могу га генерисати радници задужени за процес усвајања за било коју животињу. Генерисање извештаја покреће се кликом на дугме *Generate matches document* у оквиру прегледа свих поклапања пронађених за животињу (*слика 4.25*).



*Слика 4.25 – Дугме за генерисање извештаја о поклапањима за животињу*

Пример изгледа извештаја дат је на слици *4.26*.

Могуће унапређење генерисаног извештаја било би проширење извештаја детаљима о сваком поклапању, слично као што су приказани у оквиру страница са детаљима појединачних извештаја.

*Слика 4.26 – Пример извештаја о поклапањима за животињу*

# 5. Закључак

Софтверски пакет за подршку рада прихватилишта за животиње, описан у овом дипломском раду, подржава раднике прихватилишта приликом обраде упитника и проналаска одговарајућег пара животиње и усвојитеља и потенцијалног усвојитеља приликом подношења захтева за тражење животиње.

Увођењем информационог система смањује се време обраде упитника. Тиме је смањено време које радници троше на ову активност, што за последицу има повећање времена које им остаје за обављање других активности везаних за бригу о животињама. Потенцијални усвојитељ нема потребу да прихватилишту доставља упитнике попуњене на папиру, него упитнике може да попуни електронски. Смештањем података о животињама и потенцијалним усвојитељима у базу података информационог система, олакшава се претрага датих података. У супротном би радник прихватилишта приликом тражења усвојитеља морао да тражи податке у архиви упитника, што може да одузме време. Систем такође омогућава креирање детаљних профила животиња, који се дају на увид потенцијалним усвојитељима. Овим се усвојитељ добро информише и смањује се вероватноћа да животиња не буде онаква, какву он жели.

Такође, увођењем информационог система у прихватилиште за животиње, ствара се могућност за анализу података о поклапању животиња и усвојитеља и исходу усвајања. Тиме би потенцијално било могуће допринети проналаску животиње која више одговара усвојитељу, што би смањило број усвајања након којих животиња бива враћена, напуштена или поклоњена, јер не одговара новом власнику.

Информациони систем, описан у овом раду, може бити унапређен и проширен. Унос података о животињи обавља радник прихватилишта. Део ових података могао би се преузети из *Централне базе података о обележавању животиња,* чиме би се унос података убрзао. У оквиру упитника о личним подацима, тренутно су подржана три најчешћа типа питања. Једно од могућих побољшања било би увођење нових типова питања. Проширења су могућа и у оквиру извештаја о пронађеним поклапањима за животињу, где би извештај могао бити проширен прегледом сваког поклапања.

Даља проширења су такође могућа и додавањем подсистема за прављење распореда активности радника, распореда посета усвојитеља прихватилишту, праћење и набавку ресурса потребних за бригу о животињама.

Такође, део система намењен за анализу упитника и проналажење поклапања може бити унапређен и проширен додацима заснованим на вештачкој интелигенцији. Овакви додаци би, на пример, могли да предлажу потенцијалном усвојитељу животиње сличне оним које су му се допале или мењају значај питања у упитницима у односу на окончање поклапања у коме је то питање било поклопљено.

# Речник појмова

**Code First –** приступ моделовању базе података, где се прво класе конфигуришу и на основу њих генерише база

**Model First –** приступ моделовању базе података, где се прво помоћу неког алата конфигурише дијаграм базе података и онда, у складу са њим, сама база података

**Метаподаци – „**Подаци о подацима“; подаци који описују карактеристике неких других података

**Миграција –** у овом контексту, начин на који пакет Entity Framework базу података синхронизује са текућим моделом базе података, при коришћењу Code First приступа моделовању базе података

# Скраћенице

**СУБП –** систем за управљање базама података

**WPF –** *Windows Presentation Foundation*, графички подсистем за развијање корисничког интерфејса у Windows базираним апликацијама

**MVVM –** *Model–View–ViewModel* је структурни програмски образац по којем се одваја графички кориснички интерфејс (View) од модела бизнис логике (Model), док се за размену информација између њих користи посебан модел (ViewModel)

**WCF –** *Windows Communication Foundation*, сервисно оријентисани модел размене порука који омогућава програмима размену порука преко рачунарске мреже или локално на асинхрон начин

**SSL –** *Secure Sockets Layer*, криптографски протокол за безбедну комуникацију преко мреже

# Литература

1. Презентација на предмету Базе података 1 на ПСИ –<http://www.acs.uns.ac.rs/sites/default/files/4_BP_ER_Model_Osnove_1.pdf>
2. "Service-oriented architecture explained." – Hashimi, Sayed, *ONDot Net.com, August* 18 (2003).
3. Документација WCF, .NetFramework – <https://docs.microsoft.com/en-us/dotnet/framework/wcf/whats-wcf>
4. Документација NuGet, .NetFramework – <https://docs.microsoft.com/en-us/nuget/what-is-nuget>
5. Објашњење .NetFramework-ове уграђене *hash* функције, StackOverflow – <https://stackoverflow.com/questions/20621950/asp-net-identitys-default-password-hasher-how-does-it-work-and-is-it-secure>

# Биографија

Божидар Марић рођен је 12. децембра 1997. године у Липничком Шору. Завршио је основну школу „Свети Сава“ у Липничком Шору, а затим природни смер гимназије „Вук Караџић“ у Лозници. Школске 2016/2017. године уписује Факултет техничких наука у Новом Саду, на студијском програму Примењено софтверско инжењерство. Положио је све испите прописане планом и програмом.