

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO  
FAKULTETA ZA MATEMATIKO IN FIZIKO

Kristjan Dragovan Širnik

**Aplikacija za podporo delovanju  
društev področja socialnega varstva**

DIPLOMSKO DELO

INTERDISCIPLINARNI UNIVERZITETNI  
ŠTUDIJSKI PROGRAM PRVE STOPNJE  
RAČUNALNIŠTVO IN MATEMATIKA

MENTOR: doc. dr. Rok Rupnik

Ljubljana, 2019

COPYRIGHT. Rezultati diplomske naloge so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavo in koriščenje rezultatov diplomske naloge je potrebno pisno privoljenje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

*Besedilo je oblikovano z urejevalnikom besedil L<sup>A</sup>T<sub>E</sub>X.*

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Društva na področju socialnega varstva dobivajo sredstva od različnih finan- cerjev, večji del pa prejema od države. Država praviloma zahteva različna poročila. Proučite informacijske potrebe društev na področju socialnega var- stva. Na podlagi tega zasnujte in razvite aplikacijo, ki podpira delovanje to- vrstnih društev s poudarkom področju poročanja. Pri tem uporabite orodja in tehnologije po lastnem izboru.



# Kazalo

Povzetek

Abstract

<b>1</b>	<b>Uvod</b>	<b>1</b>
1.1	Časovne kategorije . . . . .	2
<b>2</b>	<b>Uporabljene tehnologije</b>	<b>5</b>
2.1	C# in Microsoft .NET Framework . . . . .	5
2.1.1	Knjižnica LINQ . . . . .	6
2.2	Microsoft Azure . . . . .	7
2.3	MSSQL . . . . .	7
2.4	Transakcijski SQL . . . . .	7
2.5	PowerDesigner . . . . .	8
2.6	MySQL Workbench . . . . .	8
2.7	draw.io . . . . .	9
2.8	JSON . . . . .	9
<b>3</b>	<b>Analiza in načrtovanje podatkovne baze</b>	<b>11</b>
3.1	Podatkovni model . . . . .	11
3.2	Tabele . . . . .	13
3.3	Indeksi . . . . .	22
3.3.1	Osebe . . . . .	22
3.3.2	Delavnice . . . . .	23

3.3.3	Moduli in pravice . . . . .	23
<b>4</b>	<b>Programski objekti v podatkovni bazi</b>	<b>25</b>
4.1	Procedure . . . . .	25
4.2	Prožilci (angl. <i>Triggers</i> ) . . . . .	27
<b>5</b>	<b>Analiza funkcionalnosti programa</b>	<b>29</b>
<b>6</b>	<b>Načrtovanje funkcionalnosti programa</b>	<b>33</b>
6.1	Prijava . . . . .	33
6.2	Meni . . . . .	34
6.3	Šifranti . . . . .	34
6.4	Delavnice, individualna dela in donacije . . . . .	35
6.5	Cilji . . . . .	35
6.6	Dnevne prisotnosti . . . . .	36
6.7	Poročila . . . . .	36
<b>7</b>	<b>Pregled delovanja programa</b>	<b>45</b>
7.1	Pravice . . . . .	45
7.2	Prijava . . . . .	46
7.3	Meni . . . . .	46
7.4	Šifranti . . . . .	47
7.4.1	Delavnice . . . . .	47
7.4.2	Osebe . . . . .	48
7.4.3	Pravice . . . . .	50
7.5	Delavnice . . . . .	51
7.6	Individualna dela . . . . .	53
7.7	Cilji . . . . .	55
7.8	Donacije . . . . .	57
7.9	Dnevne prisotnosti . . . . .	58
7.10	Poročila . . . . .	60
7.10.1	Poročilo donacij . . . . .	61
7.10.2	Poročilo delavnic . . . . .	62

7.10.3	Splošno poročilo oseb . . . . .	64
7.10.4	Poročilo oseb . . . . .	65
7.10.5	Poročilo individualnih del . . . . .	66
<b>8</b>	<b>GDPR</b>	<b>69</b>
8.1	Splošno o uredbi . . . . .	69
8.1.1	Načela pri zbiranju in obdelavi osebnih podatkov . . . . .	71
8.2	Rešitev v programu . . . . .	72
8.2.1	Deaktivacija osebe . . . . .	73
8.2.2	Izvoz podatkov o osebi . . . . .	74
8.2.3	Spremljanje poteka programa . . . . .	75
<b>9</b>	<b>Sklepne ugotovitve</b>	<b>77</b>
	<b>Literatura</b>	<b>78</b>



# Seznam uporabljenih kratic

<b>kratica</b>	<b>angleško</b>	<b>slovensko</b>
<b>MSSQL</b>	Microsoft SQL Server	Microsoft SQL Server
<b>T-SQL</b>	Transact-SQL	Transakcijski SQL
<b>PB</b>	Database	Podatkovna baza
<b>GDPR</b>	General Data Protection Regulation	Splošno uredba o varstvu podatkov
<b>EU</b>	European Union	Evropska unija



# Seznam uporabljenih strokovnih besed

angleški izraz	predlagani prevod	v diplomski nalogi uporabljen izraz
Serialization	Serializacija	Serializacija
Deserialization	Deserializacija	Deserializacija



# Povzetek

**Naslov:** Aplikacija za podporo delovanju društev področja socialnega varstva

**Avtor:** Kristjan Dragovan Širnik

V namen olajšanja priprave poročil za financerje določenega društva na področju socialnega varstva smo razvili program, ki bo poročila generiral sam. Diplomaska naloga vključuje opis različnih pristopov, ki so uporabljeni pri razvoju programa in uporabljene tehnologije. Program zajema več modulov, in sicer: vnos šifrantov, vnos in urejanje izvedb delavnic, individualnih del, dnevne prisotnosti, donacij, ciljev in njihovih evalvacij ter generiranje poročil naštetega. Program vsebuje tudi prijavo in omejitev dostopa do modulov njegovih uporabnikov. V podatkovni bazi, prilagojeni in optimizirani za potrebe programa, so shranjeni vsi podatki, potrebni za njegovo delovanje. Navedena podatkovna baza je v diplomski nalogi tudi podrobneje predstavljena. Zaradi shranjevanja osebnih podatkov je v nalogi opisana tudi splošna uredba o varstvu podatkov in njena implementacija v programu.

**Ključne besede:** razvoj, program, podatkovna baza, poročila.



# Abstract

**Title:** Application to support work of the societies from the area of social security

**Author:** Kristjan Dragovan Širnik

In order to make writing of reports easier for financiers of different social care organizations we developed a program that generates reports on its own. This diploma thesis contains a description of various approaches that are used in developing this program as well as techniques and technology in use. The program includes multiple modules that is establishing a register of codes, input and editing of workshops being carried out, individual tasks, daily presence, donations, goals and their evaluations as well as generating reports of the aforementioned modules. The program also contains log-on to the system and access restriction for modules of its users. The database, adjusted and optimized, carries all the data needed for its normal action. The database is described in detail in this thesis. Due to storing of personal data, this thesis also contains a general data protection regulation and its implementation in this program.

**Keywords:** development, program, database, reports.



# Poglavje 1

## Uvod

V društvu<sup>1</sup> na področju socialnega varstva (v nadaljevanju: društvo) se vsako leto soočajo s problemom, ko morajo za ministrstvo in druge financerje izdati poročilo o izvedenih dejavnostih. Ogromno količino podatkov je potrebno razvrstiti, prešteti, analizirati in povzeti, vendar za to primanjkuje časa. V ta namen smo se odločili, za diplomsko delo in za društvo, razviti program, ki bo olajšal društvu spremljanje in analizo izvedenih aktivnosti in generiral poročila za ministrstvo. Diplomsko delo predstavlja opis nastanka omenjenega programa, od predstavitve problema, do podrobne predstavitve zaključnega produkta. Prav tako bomo na kratko opisali pristope, ki smo jih uporabili pri razvoju programa, predstavili uporabljene tehnologije in orodja, s katerimi smo si pomagali pri izdelavi končnega produkta. Podrobneje bomo razložili tudi ustvarjeno PB in predstavili tabele, indekse, procedure in prožilce ter njihovo realizacijo. Nazadnje bomo na kratko omenili še GDPR in opisali, kako smo implementirali rešitev v našem programu.

Društvo izvaja različne dejavnosti, kot so delavnice, individualno delo, opredelitev in evalvacijo ciljev posameznikov, ki so vključeni v društvo. Trenutno vsa poročila o izvedenih dejavnostih evidentirajo ročno. Ker pa so delavnice, individualna dela in ostale dejavnosti raznovrstne in številčne, potre-

---

<sup>1</sup>Za namene lažjega razumevanja besedila diplomske naloge smo program, ki deluje na področju socialnega varstva, poimenovali kot „društvo.“

bujajo veliko časa za pripravo vseh potrebnih poročil. Dejavnosti morajo med drugim razvrstiti v različne časovne kategorije, glede na pogostost izvedbe (dnevno, večkrat tedensko, tedensko, večkrat mesečno, mesečno, večkrat letno, letno ali občasno).

V društvu je treba prav tako beležiti prisotnost vsakega udeleženca. Na podlagi prisotnosti se udeležence razvrsti v časovne skupine, saj je v poročilih potrebno za vsako časovno skupino podati število pripadajočih udeležencev za izbrano obdobje.

Društvo se delno financira tudi s pomočjo različnih donacij, ki so lahko finančne, v obliki storitev ali v materialu. Društvo mora voditi evidenco, kdaj so prejeli kakšno donacijo in kaj je ta donacija vsebovala.

Z namenom olajšanja dela v zvezi z evidentiranjem dela in pripravo poročil smo se odločili razviti program, v katerega bo možno evidentirati vse izvedbe delavnic, cilje in njihove evalvacije, donacije, prisotnost udeležencev in individualno delo. Ko bo ministrstvo zahtevalo poročilo, bodo zaposleni tako lahko izbrali obdobje in tip poročila, program pa bo sam razvrstil delavnice in osebe v določene časovne kategorije. V primeru individualnega dela jim bo program prikazal mesečni koledar, kjer bodo v posameznem dnevu izpisane osebe, ki so bile ta dan vključene v individualno delo. Prav tako bo omogočen prikaz števila vseh izvedenih individualnih del v mesecu in izbranem obdobju, kot tudi število različnih oseb, ki so bile v navedenem obdobju vključene v individualno delo.

## 1.1 Časovne kategorije

V programu so glede na potrebe financerjev opredeljene različne časovne kategorije, značilnosti katerih bomo v nadaljevanju podrobneje pojasnili.

**Dnevna** dejavnost se je v izbranem obdobju izvajala vsak (delovni) dan.

**Večkrat tedenska** dejavnost se je v izbranem obdobju izvajala večkrat na teden, vendar ne vsak (delovni) dan in ne le enkrat tedensko.

**Tedenska** dejavnost se je v izbranem obdobju izvajala enkrat na teden.

**Večkrat mesečna** dejavnost se je v izbranem obdobju izvajala večkrat v mesecu, vendar ne vsak teden in večkrat kot le enkrat na mesec.

**Mesečna** dejavnost se je v izbranem obdobju izvajala enkrat na mesec.

**Večkrat letna oz. občasna** dejavnost se je v izbranem obdobju izvajala večkrat na leto, vendar ne vsak mesec.

**Letna** dejavnost se je izvajala točno enkrat na leto [15].



# Poglavje 2

## Uporabljene tehnologije

### 2.1 C# in Microsoft .NET Framework

C# je splošno namenski, objektno orientiran programski jezik, ki zajema močne deklarativne, funkcionalne, generične in komponentno usmerjene programske discipline. Jezik prav tako omogoča zmožnost refleksije, kar pomeni, da v teku programa preverja in pridobiva podatke o svoji lastni strukturi ter po možnosti spremeni svojo lastno strukturo in vedenje. Razvilo ga je podjetje Microsoft v okviru ogrodja .NET. Organizaciji Ecma in ISO pa sta ga odobrili kot standard [1, 2, 21].

Microsoft .NET je ogrodje za razvijanje programske opreme, ki teče pretežno na operacijskih sistemih Microsoft Windows. Najnovejša različica je 4.7.2, ki smo jo tudi uporabili pri razvoju našega programa. .NET vključuje obsežno knjižnico razredov FCL - „Framework Class Library“ in zagotavlja medopravilnost jezikov, kar pomeni, da lahko uporablja kodo, napisano v drugih programskih jezikih, skladnih s specifikacijo CLI. Programi, napisani za .NET, se izvajajo v programskem okolju CLR - „Common Language Runtime“. Programsko okolje CLR zagotavlja višjo varnost, upravljanje s spominom in upravljanje z izjemami [13, 17].

### 2.1.1 Knjižnica LINQ

Vredno je omeniti tudi knjižnico LINQ (angl. *Language Integrated Query*), ki smo jo pri razvoju velikokrat uporabljali, saj nam je olajšala iskanje podatkov, kot tudi filtriranje in obdelovanje le-teh.

LINQ je komponenta Microsoft .NET Frameworka. Je knjižnica tehnologij, ki temeljijo na integraciji zmogljivosti poizvedb, neposredno v jezik C#. Razširja jezik z dodatnimi poizvedbenimi izrazi, ki so podobni stavkom SQL in se uporabljajo za obdelavo podatkov iz nizov, enumerabilnih razredov, XML dokumenotv ter relacijskih PB. Z uporabo sintakse poizvedb lahko izvajamo filtriranje, razvrščanje in združevanje podatkov oziroma elementov z minimalno količino kode. Knjižnica vključuje naslednje razširitve jezika:

- poizvedbeno sintakso,
- implicitne spremenljivke,
- anonimne tipe spremenljivk,
- inicializator objektov,
- lambda izraze.

LINQ definira veliko operatorjev poizvedb. Največkrat uporabljene v razvitem programu so:

**Select** izbira elementov,

**Where** filtriranje elementov,

**Sum/Min/Max/Average** operator, ki sprejme funkcijo, ki iz elementa vzame številčno vrednost in jo uporabi za nadaljno obdelovanje,

**OrderBy** resporejanje elementov na podlagi vhodne funkcije,

**GroupBy** združevanje podatkov na podlagi pogojev iz vhodne funkcije,

**Distinct** odstranjevanje podvojenih vrednosti,

**Contains** preverjanje obstoja elementa,

**Count** vrne število elementov, ki ustrežajo podanemu pogoju [10, 11].

## 2.2 Microsoft Azure

Azure je oblačna računalniška storitev, ki jo je Microsoft razvil z namenom gradnje, testiranja, uvajanja ter upravljanja aplikacij in storitev prek podatkovnih centrov Microsofta. Navajajo, da Azure podpira več kot 600 storitev, od osnovnih računskih storitev, storitev upravljanja s podatki, pa vse do strojnega učenja. Microsoft ponuja tudi emulator, ki emulira storitve Azure za razvojne namene. Z njegovo pomočjo lahko aplikacijo oziroma program razvijamo in preizkušamo, brez da bi morali ustvariti naročnino na Azure in s tem dodatne stroške. Po končanem razvoju Azure zagotovi enostaven prenos [12, 26].

## 2.3 MSSQL

MSSQL je sistem za upravljanje relacijskih podatkovnih baz. Razvilo ga je podjetje Microsoft. Kot strežnik baze ima primarno nalogo shranjevanja in pridobivanja podatkov, kot to zahtevajo druge programske aplikacije [14].

## 2.4 Transakcijski SQL

Osnovni jezik MSSQLa je T-SQL (angl. *Transact-SQL*), ki je Microsoftova in Sybaseova razširitev SQLa, ki se uporablja za interakcijo z relacijskimi PB. Vključuje proceduralno programiranje, lokalne spremenljivke ter različne podporne funkcije, kot sta na primer obdelava nizov in obdelava podatkov [24].

## 2.5 PowerDesigner

SAP PowerDesigner je ogrodje za modeliranje in oblikovanje programske opreme, ki ga izdeluje Sybase in je trenutno v lasti SAPa. Pomembnejše lastnosti, za katere PowerDesigner vključuje podporo, so: modeliranje poslovnih procesov, generiranje kode, modeliranje podatkov, modeliranje skladišča podatkov, objektno modeliranje, generiranje poročil, XML modeliranje, ki podpira XML shemo in DTD standarde [20].

PowerDesigner smo uporabili za pretvorbo sheme PB v skripte, ki so nam PB kreirale, in za kreiranje razredov, ki so nam predstavljali osnovo za program. Dobljene razrede smo nato dopolnili do funkcionalnosti, ki smo jo želeli imeti. Vsakemu smo dodali upravljanje s transakcijami. To pomeni, da smo vsako zapisovanje in pridobivanje podatkov obvili s transakcijo. Če ta še ni obstajala, smo jo ustvarili in zaprli ob končanem ukazu, sicer pa smo dopolnili obstoječo. Prav tako smo spremenili zapisovanje, in sicer smo namesto SQL ukazov v kodi v PB ustvarili procedure, opisane v naslednjem poglavju, v kodi napolnili parametre procedure in jo poklicali. Opisam pristop s proceduram je bolj pregleden, saj je celotna SQL koda ločena.

## 2.6 MySQL Workbench

MySQL Workbench je vizualno orodje za oblikovanje baze podatkov, ki združuje razvoj, upravljanje, načrtovanje, izdelavo in vzdrževanje SQL v enotno integrirano razvojno okolje za sistem baze podatkov MySQL. Pomembnejše lastnosti, za katere MySQL Workbench vključuje podporo, so: modeliranje podatkov, upravljanje PB, spremljanje učinkovitosti, selitev PB in urejevalnik SQL [16].

MySQL Workbench smo uporabili le za izris sheme PB, iz katere smo nato lahko zgenerirali PB in razrede.

## 2.7 draw.io

draw.io je odprtokodni tehnološki sklad za gradnjo diagramov poteka aplikacij. Z njim lahko ustvarimo diagrame poteka, diagrame procesa, UML, ER in mrežne diagrame. Omogoča možnost izdelave diagramov z uporabo spletnega brskalnika, kjer lahko ustvarjene diagrame delimo na aplikaciji Confluence in JIRA, lahko pa jih shranimo tudi na Google Drive. Na voljo imamo tudi namizno različico, v primeru, da nimamo vedno na voljo internetne povezave [5].

draw.io smo uporabili za izris diagramov poteka našega programa, na katere bomo naleteli v naslednjih poglavjih.

## 2.8 JSON

JSON (angl. *JavaScript Object Notation*) je standardna oblika zapisa datotek. Ustvarjen je bil iz programskega jezika JavaScript. Format podatkov je oblika besedila, ki je neodvisna od jezika, zaradi česar je primeren za izmenjavo podatkov. Številni programski jeziki že podpirajo serializacijo in deserializacijo podatkov v obliki JSON.

Zgrajen je na dveh strukturah, in sicer zbirki parov ime - vrednost, ter urejenem seznamu vrednosti, ki je običajno v večini jezikov realizirano kot matrika, vektor, seznam ali zaporedje. Vrednost lahko predstavlja objekt, zapis, struktura, slovar ali celo seznam [8, 9].



## Poglavje 3

# Analiza in načrtovanje podatkovne baze

Po razgovoru z naročnikom, ko smo imeli podane vse zahteve in želje, kaj naj program podpira in kako naj deluje, smo se lotili analize in načrtovanja podatkovne baze. Zahteve in želje smo razdelili na več delov, tako da smo na koncu enostavno sestavili entitetno-relacijsko shemo. Nato smo se lotili normalizacije do tretje normalne forme. Entitetno-relacijsko shemo smo nato prilagodili in popravili. Popravljenost smo prikazali na sliki 3.1. Nadaljevali smo s prilagajanjem PB in sicer smo dodali indekse, ki so nam koristili pri povečanju učinkovitosti PB.

### 3.1 Podatkovni model

V orodju PowerDesigner smo ustvarili podatkovni model, ki nam je omogočil kreiranje skript za ustvarjanje PB, kot tudi generiranje objektov uporabljenih v programu. Podatkovni model, prikazan na sliki 3.1, je bil prenešen iz orodja PowerDesigner v orodnje MySQL Workbench zaradi lepšega prikaza tabel in povezav med njimi.



## 3.2 Tabele

Tabela je zbirka podatkovnih elementov. Vsaka tabela ima določeno število stolpcev in poljubno število vrstic. Vsako vrstico identificira ena ali več vrednosti, ki se pojavijo v določeni podmnožici stolpcev. Ta posebna izbira stolpcev, ki enolično identificirajo vrstice, se imenuje primarni ključ [23].

V naši podatkovni bazi imamo 14 tabel. Tabele se med seboj logično povezujejo, glede na to, kaj se v določeni tabeli referencira. Vsaka tabela ima svoj primarni ključ. Pri večini tabel smo zaradi hitrejšega povezovanja in iskanja vrstic uvedli umetne primarne ključe. Kljub umetnim ključem se nekatere vrednosti nekaterih stolpcev še vedno ne smejo ponoviti. To so stolpci, ki so bili za lažje iskanje ustvarjeni na podlagi naročnikove želje. V nadaljevanju bomo opisali tabele in razložili njihov pomen.

Tabela **Osebe** predstavlja vse osebe, vnešene v sistem. Za vsako osebo hranimo naslednje podatke:

**ID\_USER** predstavlja primarni ključ tabele in je obvezen, unikaten ter generiran avtomatično.

**Ime** predstavlja ime osebe in je obvezen podatek.

**Priimek** predstavlja priimek osebe in je obvezen podatek.

**Datum\_rojstva** predstavlja datum rojstva osebe in je obvezen podatek.

**Tip** predstavlja vrednost, ki pove, ali je oseba administrator, zaposlen, prostovoljec, udeleženec ali nič od naštetega. Tudi ta podatek je obvezen, saj na podlagi tega podatka v programu razvrščamo osebe v omenjene kategorije.

**Uporabnisko\_ime** predstavlja uporabniško ime administratorja, zaposlenega ali prostovoljca, s katerim se lahko prijavi v program. Podatek je neobvezen.

**Aktiven** predstavlja aktivnost zaposlenih, administratorja in prostovoljcev.

V primeru večkratnega napačnega vnosa gesla se spremeni vrednost

na neaktivno, zato se navedeni ne morejo več prijaviti tudi ob pravilnem vnosu uporabniškega imena in gesla. Udeleženci imajo privzeto vrednost tega stolpca, ki je obvezen.

**Geslo** predstavlja geslo za zaposlene, administratorja in prostovoljce, s katerim se lahko prijavijo v program. Podatek je neobvezen, saj udeleženci nimajo možnosti dostopa do programa.

**Geslo\_old** predstavlja staro geslo, s pomočjo katerega preverjamo varnost novega gesla. Podatek je neobvezen.

**Email** predstavlja email zaposlenih, administratorja in prostovoljcev. Podatek je neobvezen, saj emailov udeležencev ne potrebujemo.

**Datum\_zadnje\_menjave\_gesla** predstavlja datum, na katerega so uporabniki programa zadnjič geslo zamenjali. Na podlagi tega datuma programu damo zahtevo za posodobitev gesla, ob prekinitvi pa uporabniku programa onemogočimo dostop, dokler ga ne zamenja. Podatek je neobvezen.

**Zadnja\_prijava** predstavlja datum, kdaj se je administrator, zaposlen ali prostovoljec nazadnje prijavil v sistem. Podatek je neobvezen.

**GDPR\_deaktiviran** predstavlja vrednost, ki pove, če je oseba deaktivirana zaradi njene prošnje po izbrisu osebnih podatkov. Podatek je neobvezen.

**GDPR\_deaktiviran\_datum** predstavlja datum, na katerega smo izbrisali osebne podatke osebe. Podatek je neobvezen.

**GDPR\_razlog** predstavlja razlog, zaradi katerega smo morali odstraniti osebne podatke osebe. Podatek je neobvezen.

Tabela **Delavnice** vsebuje vse delavnice, ki se izvajajo v društvu. Vsako delavnico opišemo z naslednjimi podatki:

**Sifra\_delavnice** je primarni ključ tabele in je obvezna in unikatna.

**Ime\_delavnice** predstavlja kratko ime delavnice in je obvezen podatek.

**Opis\_delavnice** predstavlja dodaten opis delavnice in je neobvezen podatek.

**Potrebno\_vodstvo** predstavlja vrednost, ki pove, če je za izvedbo te delavnice potrebno vodstvo ali se lahko izvaja brez njega. Podatek je neobvezen, saj je privzeta vrednost - v primeru, da podatka ni - da delavnica potrebuje vodstvo.

**Datum\_zadnje\_spremembe** predstavlja datum, kdaj se je posamezna delavnica spremenila. Podatek ni obvezen, a se napolni ob vsaki spremembi podatka v tabeli.

**Zadnja\_sprememba\_oseba** predstavlja identifikator osebe, ki je delavnico spremenila.

Tabela **Moduli** predstavlja vse module, ki jih program ponuja. Tabela nam olajša razporeditev pravic po različnih modulih in vsebuje naslednje tri podatke:

**ID\_MODUL** je primarni ključ tabele in je obvezen, unikatni in generiran avtomatično.

**Sifra\_modula** predstavlja unikatno šifro, po kateri razpoznamo posamezen modul. Je obvezen podatek.

**Opis\_modula** predstavlja kratek opis, za kaj se modul uporablja. Podatek je neobvezen.

V tabeli **Pravice** hranimo vse pravice, ki jih posamezna oseba lahko ima. Vsaka pravica omejuje določeno funkcionalnost programa. Tabela ima štiri attribute, in sicer:

**ID\_PRAVICE** je primarni ključ tabele in je obvezen, unikatni in generiran avtomatično.

**ID\_MODUL\_MODUL** je tuji ključ, ki je obvezen in poveže pravico z modulom, v katerega pripada.

**Sifra\_pravice** je unikatna šifra, po kateri razpoznamo posamezno pravico. Je obvezen podatek.

**Opis\_pravice** predstavlja kratek opis, kaj pravica v programu omogoča.

V tabeli **Ima\_pravico** hranimo vse pravice, ki jih posamezna oseba ima. Da lahko pravice uporabimo v tabeli, hranimo naslednje podatke:

**ID\_IMA\_PRAVICO** je primarni ključ tabele in je obvezen, unikatni in generiran avtomatično.

**ID\_USER** je tuji ključ, ki določi, na katero osebo se pravica nanaša. Je obvezen podatek.

**ID\_PRAVICE** je tuji ključ, ki določi, za katero pravico smo podelili dovoljenje. Je obvezen podatek.

**Dovoljenje** predstavlja vrednost, ki pove, če imamo podeljeno pravico do posamezne funkcionalnosti programa. Je obvezen podatek. V primeru, da zapis za določeno pravico in osebo manjka, program za privzeto vzame, da do te funkcionalnosti oseba nima dostopa.

Tabela **Cilji** je namenjena hranjenju podatkov o vnešenih ciljih udeležencev. Vsebuje naslednje attribute:

**ID\_CILJA** je primarni ključ tabele in je obvezen, unikatni in generiran avtomatično.

**ID\_USER** je tuji ključ, ki pove, kateri osebi posamezen cilj pripada. Je obvezen podatek.

**Datum\_cilja** predstavlja datum, ko se je cilj ustvaril. Je obvezen podatek.

**Cilj** predstavlja podroben opis cilja in je prav tako obvezen podatek.

**Zaključen** predstavlja, če je bil cilj dosežen ali se še vedno spremlja njegov napredek. Podatek ni obvezen.

**Zaključen\_dne** predstavlja datum, na katerega je bil cilj dosežen. Podatek ni obvezen.

**Datum\_zadnje\_spremembe** predstavlja datum, kdaj se je cilj spremenil. Podatek ni obvezen, vendar ga v programu ob vsaki spremembi napolnimo, tako da bo vrednost vedno vnešena.

**Zadnja\_sprememba\_oseba** predstavlja identifikator osebe, ki je cilj spremenila.

Vsak cilj je možno evalvirati. V ta namen je ustvarjena tabela **Evalvacije**, ki vsebuje naslednje attribute:

**ID\_EVALVACIJE** je primarni ključ tabele in je obvezen, unikaten in generiran avtomatično.

**ID\_CILJ** poveže evalvacijo s ciljem. Je tuji ključ in je obvezen podatek.

**Evalvacija** predstavlja podroben opis evalvacije. Je obvezen podatek.

**Datum\_evalvacije** predstavlja datum, na katerega je bila evalvacija narejena.

**Datum\_zadnje\_spremembe** predstavlja datum, kdaj se je evalvacija spremenila. Podatek ni obvezen, vendar ga v programu ob vsaki spremembi napolnimo, tako da bo vrednost vedno vnešena.

**Zadnja\_sprememba\_oseba** predstavlja identifikator osebe, ki je evalvacijo spremenila.

V tabelo **Donacije** shranimo vse prejete donacije. Vsaka donacija ima shranjene naslednje podatke:

**ID\_DONACIJE** je primarni ključ tabele in je obvezen, unikaten in generiran avtomatično.

**Opis\_donacije** predstavlja opis, kaj so prejeli. Podatek ni obvezen, saj v primeru denarne donacije te ni potrebno še dodatno opisati.

**Datum\_donacije** predstavlja datum, kdaj je bila donacija prejeta, in je obvezen podatek.

**Znesek** predstavlja vsoto doniranega denarja. Podatek ni obvezen, saj donacija ni nujno denarna.

**Oseba\_vnosa** predstavlja identifikator osebe, ki je donacijo vnesla v sistem. Podatek je obvezen.

**Datum\_zadnje\_spremembe** predstavlja datum, kdaj se je donacija spremenila. Podatek ni obvezen.

**Zadnja\_sprememba\_oseba** predstavlja identifikator osebe, ki je donacijo spremenila.

Za vsakega udeleženca je treba hraniti podatek, koliko časa je bil prisoten v društvu. V ta namen smo ustvarili tabelo **Dnevna\_prisotnost** z naslednjimi atributi:

**ID\_DPRIS** je primarni ključ tabele in je obvezen, unikaten in generiran avtomatično.

**ID\_USER** je tuji ključ, ki pove, za katero osebo je zapisana posamezna dnevna prisotnost. Je obvezen podatek.

**Datum\_prisotnosti** predstavlja datum dnevne prisotnosti in je obvezen podatek.

**Ure** predstavlja število ur, ko je bil posamezen udeleženec prisoten. Je obvezen podatek.

**Datum\_zadnje\_spremembe** predstavlja datum, kdaj se je dnevna prisotnost spremenila. Podatek ni obvezen.

**Zadnja\_sprememba\_oseba** predstavlja identifikator osebe, ki je dnevno prisotnost spremenila.

Udeleženci imajo lahko izvedenih poljubno mnogo individualnih del. Zaposlene pa zanima, kaj so na individualnih delih reševali in koliko del je posamezen udeleženec izvedel. V ta namen smo ustvarili tabelo **Individualno\_delo** z atributi:

**ID\_IND\_DELO** je primarni ključ tabele in je obvezen, unikaten in generiran avtomatično.

**ID\_USER** predstavlja osebo, ki je imelo izvedeno individualno delo. Je tuji ključ in obvezen podatek.

**Datum\_ind\_dela** predstavlja datum, kdaj se je individualno delo izvajalo, in je obvezen podatek.

**Cas\_zacetka\_ind\_dela** predstavlja, kdaj se je individualno delo začelo, in je obvezen podatek.

**Cas\_konca\_ind\_dela** predstavlja, kdaj se je individualno delo končalo, in je obvezen podatek.

**Zaznamek** predstavlja sprotne zaznamke izvajalcev individualnega dela in je neobvezen podatek.

**Datum\_zadnje\_spremembe** predstavlja datum, kdaj se je individualno delo spremenilo. Podatek ni obvezen.

**Zadnja\_sprememba\_oseba** predstavlja identifikator osebe, ki je individualno delo spremenila.

Vsako individualno delo mora izvajati zaposleni ali prostovoljec. Ker pa je lahko na enem individualnem delu več izvajalcev, smo ustvarili tabelo **Izvajal\_individualno\_delo**, kjer se hranijo ti izvajalci. Tabela vsebuje tri attribute:

**ID\_IZVAJALCI\_INDIVIDUALNEGA\_DELA** je primarni ključ tabele in je obvezen, unikatni in generiran avtomatično.

**ID\_IND\_DELO** je tuji ključ in je del primarnega indeksa tabele, ki poveže izvajalca z izvedbo individualnega dela. Je obvezen podatek.

**ID\_USER** je prav tako tuji ključ in je del primarnega indeksa tabele, ki poveže individualno delo z izvajalcem. Je obvezen podatek.

Za beleženje izvedb delavnic smo ustvarili tabelo **Izvedba\_delavnice**, kamor se shranijo naslednji podatki:

**ID\_IZVEDBE** je primarni ključ tabele in je obvezen, unikatni in generiran avtomatično.

**Sifra\_delavnice** je tuji ključ, ki določa, katera delavnica se je izvedla, in je obvezen podatek.

**Datum\_izvedbe** predstavlja datum, kdaj se je delavnica izvajala. Je obvezen podatek.

**Zacetek\_izvedbe** predstavlja, kdaj se je izvedba delavnice začela, in je obvezen podatek.

**Konec\_izvedbe** predstavlja, kdaj se je izvedba delavnice končala, in je obvezen podatek.

**Opombe** predstavlja sprotne opombe in opažanja izvajalcev in ni obvezen podatek.

**Izjemna\_izvedba** predstavlja vrednost, ki določa, ali je bila izvedba delavnice taka, da se druge delavnice na ta dan niso mogle izvajati. To nam koristi pri generiranju poročil, saj moramo v primeru, ko obstaja taka delavnica, ta dan preskočiti pri razvrščanju izbrane delavnice v časovne kategorije.

**Datum\_zadnje\_spremembe** predstavlja datum, kdaj se je izvedba delavnice spremenila. Podatek ni obvezen.

**Zadnja\_sprememba\_oseba** predstavlja identifikator osebe, ki je izvedbo delavnice spremenila.

Na delavnicah je prisotnih več oseb. V ta namen smo ustvarili tabelo **Prisotnost\_na\_delavnici**. Tabela sestavljajo naslednji atributi:

**ID\_PRISOTNOST\_NA\_DELAVNICI** je primarni ključ tabele in je obvezen, unikaten in generiran avtomatično.

**ID\_IZVEDBE** je tuji ključ in del primarnega indeksa tabele, ki poveže izvedbo delavnice z osebo, ki je bila na njej prisotna. Je obvezen podatek.

**ID\_USER** je tuji ključ in je del primarnega indeksa tabele, ki poveže osebo z izvedbo delavnice. Je obvezen podatek.

**Tip\_prisotnosti** določa, kaj je posamezna oseba na tej delavnici počela. Možni vrednosti sta dve, in sicer: oseba je lahko delavnico vodila ali jo je obiskala. Ta podatek je prav tako obvezen.

V namen spremljanja poteka programa smo dodali tabelo **Log** z atributi, navedenimi spodaj. Tu beležimo vsak dostop do različne funkcionalnosti programa. Tabela ni povezana z nobeno drugo, saj v primeru brisanja osebe oziroma modula ne smemo pobrisati vrednosti iz te tabele, saj sicer zgubimo nadzor. V primeru, da osebo izbrisemo, še zmeraj lahko vidimo, da je nekdo dostopal do določenih modulov. Tabela smo morali izločiti iz normalizacije, saj bi sicer pokvarili njen namen.

**ID\_LOG** je primarni ključ tabele in je obvezen, unikaten in generiran avtomatično.

**Datum\_cas** predstavlja točen datum in čas, kdaj je prišlo do dostopa oziroma spremembe podatkov. Je obvezen podatek.

**Oseba** hrani vrednost identifikatorja osebe in je obvezen podatek.

**Modul** predstavlja šifro modula, od koder se je zapis ustvaril. Je obvezen podatek.

**Opis** predstavlja kratek opis storjenega dogodka. Je obvezen podatek.

**Napaka** predstavlja podroben opis napake, v primeru da je do te prišlo. Podatek ni obvezen.

### 3.3 Indeksi

Indeksi so podatkovne strukture, s katerimi povečamo učinkovitost PB. Je datoteka, katere vsebina omogoča hiter dostop do zapisov v osnovni fizični datoteki, ki je z indeksom indeksirana. Poznamo primarne in sekundarne indekse. Prvi je izveden po primarnem ključu, ki mora biti obvezen in unika-ten, saj določa privzeto urejenost zapisov osnovne datoteke ter se uporablja za povezovanje tabel. Vsaka tabela ima lahko največ en primarni indeks. Slednji je izveden po podatkovnih elementih, ki nastopijo v tabeli, a niso ključi. Prav tako ločimo goste in redke indekse. Razlika med njima je, da gosti indeks vsebuje zapis za vsako vrstico v tabeli, redek indeks pa vsebuje manj zapisov in ti nastopajo v paru - kazalec na skupino polj in vrednost ključa zadnjega zapisa v skupini [6].

V naši PB smo ustvarili nekaj sekundarnih indeksov, s čimer smo po-  
spešili delovanje razvitega programa. V nadaljevanju smo našli dodatno  
ustvarjene indekse in razložili njihov pomen.

#### 3.3.1 Osebe

V tabeli „Osebe“ smo dodali dva sekundarna indeksa, in sicer na stolpca „Ime“ in „Priimek“. Po teh dveh poljih uporabniki programa največkrat iščejo podatke. Za povezovanje tabel smo uporabili primarni indeks, ki je umetno ustvarjen. Za primarni indeks bi lahko vzeli tudi EMŠO oseb ali davčno številko, vendar naročniki programa ne želijo hraniti teh podatkov o

osebah, zato je bilo neizbežno ustvariti dodaten, unikaten, samodejno nastavljen stolpec.

### **3.3.2 Delavnice**

V tabeli „Delavnice“ smo dodali le en sekundaren indeks. Ta je bil ustvarjen na stolpcu „Ime\_delavnice“, z enakim namenom kot v prejšnjem razdelku - uporabniki programa, kljub želji uvedbe šifer delavnic, ki so primarni ključ in tudi primarni indeks te tabele, še vedno velikokrat iščejo po imenu delavnice.

### **3.3.3 Moduli in pravice**

V tabeli „Moduli“ smo dodali sekundarni indeks na stolpec „Sifra\_modula“. V tabeli „Pravice“ pa smo prav tako dodali sekundarni indeks na stolpec „Sifra\_pravice“. V programu pogosto preverjamo, ali ima uporabnik pravico dostopa do določenega modula v programu. Za povezovanje med tabelama „Pravice“ in „Moduli“ smo uporabili primarni ključ in primarni indeks tabele „Moduli“, vendar največkrat se sklicujemo na šifro modula, kot tudi na šifro pravice. Z dodanima indeksoma smo pospešili preverjanje dostopov in s tem samo nalaganje programa.



## Poglavje 4

# Programski objekti v podatkovni bazi

Z namenom olajšanja povezovanja med različnimi sistemi, smo v podatkovno bazo dodali procedure. Prav tako smo želeli pospešiti izpisovanje poročil s prožilci. V tem poglavju bomo predstavili procedure in razložili, zakaj prožilcev nismo mogli uporabiti.

### 4.1 Procedure

Shranjene procedure (angl. *stored procedure*) so postopki, ki dostopajo do sistema za upravljanje relacijskih baz podatkov. Ti postopki so shranjeni v podatkovnem slovarju baze podatkov. Procedure vključujejo validacijo podatkov, kot tudi mehanizme za nadzor dostopa. Shranjene procedure nam olajšajo shranjevanje in posodabljanje podatkov, saj lahko z enim klicem izvedemo več SQL stavkov. Prav tako lahko različne aplikacije pokličejo iste procedure, kar močno olajša delo, ko pride do povezovanj več aplikacij. Obstajajo tudi gnezdene shranjene procedure, kar pomeni, da znotraj ene izvedbe shranjene procedure pokličemo in izvedemo drugo. Procedura ima lahko deklarirane spremenljivke (angl. *declared variables*) za obdelavo podatkov in kazalce, ki omogočajo izvajanje zank. Procedura lahko vsebuje

SQL stavke za nadzor pretoka, kot so „IF, WHILE, CASE“. Procedure lahko vrnejo nize rezultatov, kot tudi spremenijo zapise v PB [22].

Za potrebe našega programa smo za vsak razred ustvarili dve proceduri, in sicer prva služi za vstavljanje in posodabljanje, druga pa za brisanje. Prva ima toliko vhodnih parametrov, kolikor stolpcev ima ciljna tabela. V proceduri nato preverimo, če vrstica že obstaja in jo moramo le posodobiti ali pa taka vrstica še ne obstaja in smo podatek dodali na novo. Preverjanje se izvaja tako, da preverimo identifikator - primarni indeks. Privzeta vrednost pri dodajanju je -1, v nasprotnem primeru bo le-ta različen od -1, oziroma neka pozitivna številka. V primeru dodajanja program pričakuje povratno informacijo o primarnem indeksu. Procedura ga mora zaradi tega vrniti. V primeru avtomatskega kreiranja vrne le-tega, v nasprotnem primeru pa vrne pripadajoči podatek. S tem zagotovimo tudi potrditev vstavljanja.

Procedura za brisanje sprejme le primarni indeks, na podlagi katerega nato izbriše pripadajočo vrstico. Določene procedure, katerih tabela ima še povezane tabele, izbriše vrstice še iz povezanih tabel, na podlagi primarnega indeksa prvotne tabele, preko katerega so podatki med seboj povezani.

V nadaljevanju je prikazana oblika vseh shranjenih procedur. Zaradi splošnosti predpostavljamo, da parametri po vrsti pripadajo stolpcem v tabeli. Na primer parameter z imenom „Parameter1“ pripada stolpcu v tabeli z imenom „Stolpec1“.

```
CREATE PROCEDURE [dbo].[sp_<MODUL>_<ImeProcedure>]
    @Identifikator AS INT
    [ {@Parameter1 AS TIP} ]
AS
BEGIN
    IF @Identifikator = -1
    BEGIN
        INSERT INTO Ime_tabele(
            {Stolpec1}
        )
        VALUES(
            {@Parameter1}
        )
        SELECT SCOPE_IDENTITY()
    END

    ELSE
    BEGIN
        UPDATE Ime_tabele
            SET {Stolpec1 = @Parameter1}
            WHERE [Identifikator] = @Identifikator
    END
END
```

## 4.2 Prožilci (angl. *Triggers*)

Prožilec je proceduralna koda, ki se samodejno izvede ob določenem dogodku, ki se zgodi v PB. Poznamo DDL, DML in prijavne prožilce. DML prožilci se sprožijo, ko uporabnik skuša spremeniti podatke s pomočjo jezika DML. Ti dogodki so „INSERT, UPDATE“ in „DELETE“. DDL prožilci se sprožijo,

ko pride do ukazov „CREATE, ALTER“ in „DROP“, ter nekatere sistemске shranjene procedure, ki izvajajo podobne DDL operacije. Slednji pa se sprožijo, ko se vzpostavi seja uporabnika. Prožilci se lahko izvedejo pred ali po dogodku. Večinoma se uporabljajo za vzdrževanje celovitosti informacij v PB [3, 4].

V naši podatkovni bazi prožilcev nismo uporabili. Možnost vključitve letih bi lahko bila pri razvrščanju delavnic in oseb v časovne kategorije. Torej ob vsaki dodani delavnici bi pregledali, ali se je kategorija spremenila ali je ostala enaka. V tabelo delavnic bi morali dodati nov stolpec, kamor bi se kategorija shranila. Težav pri tej implementaciji je več. Ena od njih je počasno vstavljanje, posodabljanje in brisanje zaradi dodatnega preračunavanja. Naslednja nastane pri definiranju, kaj polje v tabeli predstavlja. Polje bi lahko predstavljalo, v katero časovno kategorijo delavnica spada v trenutnem letu. V tem primeru je težava, kdaj se poročilo kreira, saj v primeru, da se ustvari poročilo kasneje, kot vnese prva izvedba delavnice v novem letu, izgubimo pomen polja, saj bi morali preračunati znova. Prav tako bi le pospešilo kreiranje poročila v obdobju od začetka leta do dneva, ko je uporabnik poročilo zahteval. Ker pa kreiranje poročil uporabnik lahko požene kadarkoli, za katerokoli obdobje, tega podatka ponovno nimamo shranjenega in moramo razvrščanje pognati ponovno. Zato smo se odločili, da prožilce ne pri delavnicah ne pri osebah ne ustvarimo. Po testiranju s strani naročnika, ko bodo videli, koliko krat letno bodo potrebovali izpis poročil, bomo po potrebi prožilce in polja dodali.

## Poglavje 5

# Analiza funkcionalnosti programa

Diagram primerov uporabe je grafični prikaz interakcij med uporabnikom programa in različnimi primeri uporabe, ki jih lahko uporabnik izvaja. Diagram primerov uporabe omogoča pregled sistema in na razumljiv način razvijalcem in uporabnikom predstavi vse funkcionalnosti sistema. Narejen je v jeziku UML (angl. *Unified Modeling Language*), ki standardizira modeliranje računalniških sistemov [18, 28].

Za lažjo predstavitev funkcionalnosti programa naročniku smo ustvarili diagram primerov uporabe. Diagram je prikazan na sliki 5.1 in prikazuje primere uporabe programa. Za zagon programa mora uporabnik najprej imeti program nameščen na računalnik. Po zagonu se uporabniku prikaže prijavno okno, kamor vnese prijavne podatke. V primeru, da je geslo ali uporabniško ime napačno, se uporabniku izpiše obvestilo. V nasprotnem primeru pa iz podatkovne baze pridobimo podatke o uporabniku in njegove pravice.

Po uspešni prijavi lahko uporabnik začne urejati šifrate. Tu lahko dodaja, spreminja ali odstranjuje delavnice, kot tudi osebe. Pri osebah ponudimo še možnost spreminjanja gesla in aktivacije drugih uporabnikov. Za vsako osebo lahko nastavimo pravice, do katerih funkcionalnosti programa

lahko dostopa. Prav tako pa lahko osebne podatke osebe izbrišemo iz podatkovne baze. Kot vidimo v diagramu, se vsaka sprememba zapiše v PB na strežniku.

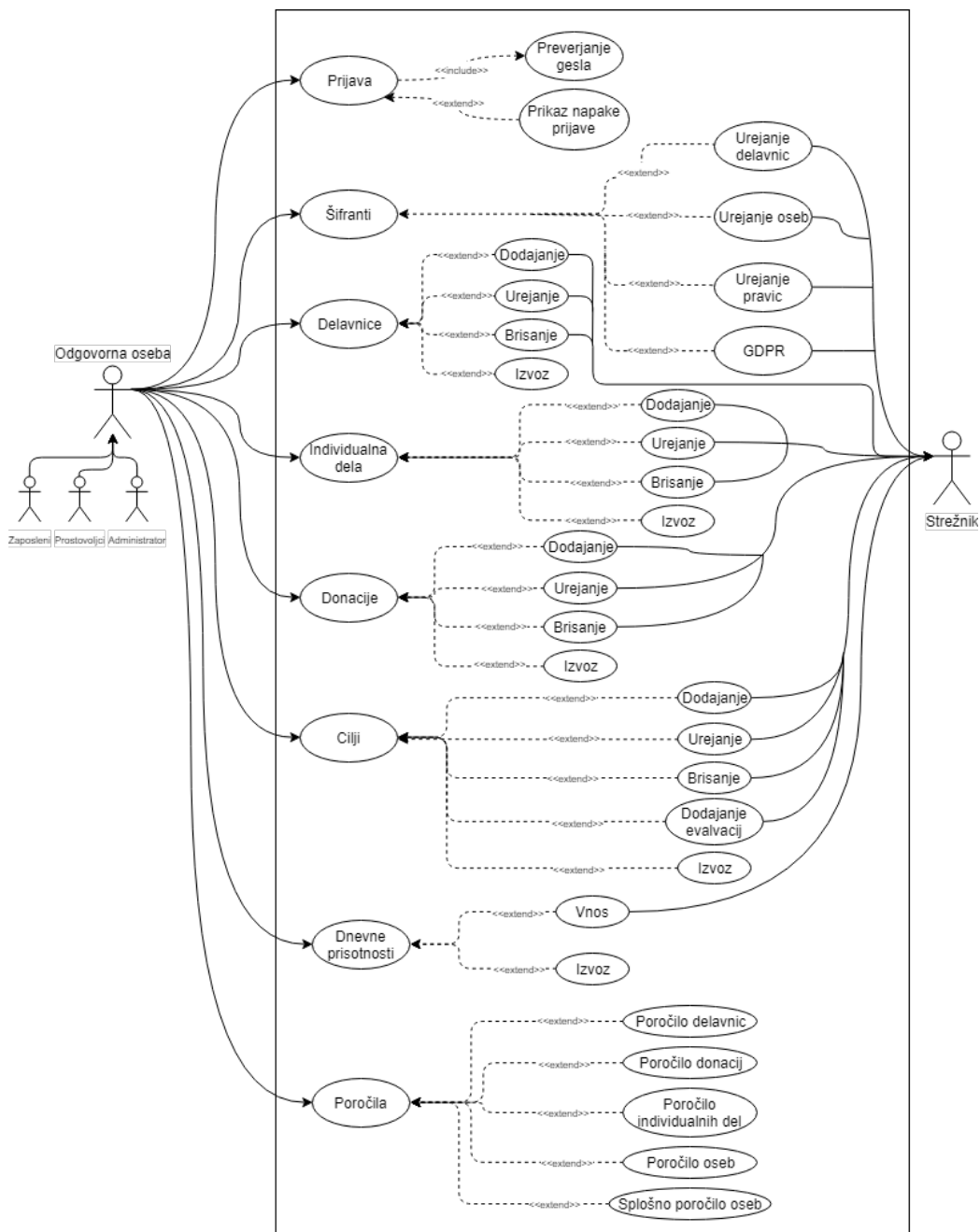
Naslednja možnost je, da uporabnik želi urejati izvedbe delavnice. Tu lahko izvedbe dodaja, ureja, briše ali izvozi v datoteko. Pri vsaki spremembi vnosa izvedbe se ta zapiše v PB. Možnosti urejanja individualnih del in donacij ne bomo posebej obrazložili, saj so identična delavnicam.

Uporabnik ima prav tako možnost urejati cilje. Ti se od urejanja delavnic razlikujejo le v tem, da lahko uporabnik doda tudi evalvacijo na cilj, ki se prav tako zapiše v PB.

Uporabnik lahko vnaša dnevne prisotnosti udeležencev in tudi izvozi prikazano obdobje v novo datoteko. Tako kot v prejšnjih možnostih se tudi tu vsaka sprememba vnosa zapiše v PB.

Uporabnik lahko iz vseh vnosov pripravi različne izpise poročil, ki se delijo na:

- Poročilo delavnic - razvrstitev delavnic v časovne kategorije,
- Poročilo individualnih del - mesečni prikaz oseb, ki so bile vključene v individualno delo,
- Poročilo donacij - izpis števila in vrednosti vseh prejetih donacij,
- Poročilo oseb - razvrstitev oseb v določene časovne kategorije na podlagi dnevnih prisotnosti,
- Splošno poročilo oseb - prikaz števila vseh vključenih udeležencev in povprečno število ur vključenosti enega udeleženca v določenem obdobju.



Slika 5.1: Diagram primerov uporabe.



## Poglavje 6

# Načrtovanje funkcionalnosti programa

Po potrditvi prikaza funkcionalnosti s strani naročnika smo naredili načrt, kako naj program deluje. Tega smo ustvarili s pomočjo diagrama poteka uporabe. Za vsak modul smo zaradi preglednosti ustvarili nov diagram. Diagrame smo naročniku prikazali in razložili ter po potrditvi začeli z razvojem. Pri razvoju smo sledili diagramu, kar nam je močno olajšalo in pospešilo delo, saj smo imeli celoten potek že ustvarjen. V nadaljevanju vam bomo predstavili diagrame poteka za vsak modul posebej.

### 6.1 Prijava

Na sliki 6.1 je prikazan diagram poteka prijave. Ob vsakem zagonu programa je treba vnesti uporabniško ime in geslo. V primeru, da so podatki nepravilni, povečamo števec, ki šteje število napačnih poskusov in prikažemo napako. Ko presežemo število dovoljenih poskusov prijave, osebo z vnešenim uporabniškim imenom deaktiviramo in prikažemo napako. V nasprotnem primeru, če so bili vnešeni podatki pravilni, preverimo, ali je oseba aktivna. Če ni, torej je bila že enkrat deaktivirana, prikažemo napako, v nasprotnem primeru pa pregledamo starost gesla. Če je geslo pretečeno, podamo zah-

tevo za spremembo gesla. Ta ne sme biti enak trenutnemu in prejšnjemu. Ob uspešni posodobitvi gesla uporabnika spustimo v sistem, v nasprotnem primeru zaključimo z izvajanjem. Po uspešni prijavi naložimo uporabnikove pravice, tako da lahko v nadaljevanju omogočamo in onemogočamo dostope do posameznih modulov in funkcionalnosti. Zaradi spremljanja poteka programa vsak napačen vnos in vsak zahtevek zabeležimo.

## 6.2 Meni

Na sliki 6.2 je prikazan diagram poteka menija. Prijavljena oseba iz menija dostopa do vseh modulov, ki jih program ponuja, oziroma do vseh modulov, do katerih ima dostop. Uporabniška odločitev zažene izbran modul v novi niti. V primeru, da uporabnik izbere izhod iz programa, se zapre vse aktivne procese in zaključi z izvajanjem.

## 6.3 Šifranti

Na sliki 6.3 je prikazan diagram poteka modula šifrantov. Prijavljena oseba na tem mestu upravlja z delavnicami, osebami in njenimi pravicami ter deaktivira osebe.

Ob zagonu modula se najprej preverijo pravice in prijavljeni osebi se prilagodi dostop do modula.

Uporabnik ima možnost dodajanja ali urejanja delavnice ali osebe. V primeru urejanja obstoječe podatke napolnimo in prikažemo. Pred shranjevanjem se naredi preverjanje, če je vnos podatkov pravilen. Tu se preveri, če so se vsa obvezna polja napolnila. V primeru nepravilnosti se izpiše napaka in označi vsa pomankljiva oz. neustrezno izpolnjena polja. V nasprotnem primeru pa posodobimo oziroma ustvarimo zapis v PB.

Pri spreminjanju pravic oseb se naložijo vse pravice in osebe. Ob izbiri osebe naložimo njene pravice. Prijavljena oseba nato izbere oziroma odstrani pravice. Ob shranjevanju pregledamo vse pravice, če je pravica izbrana,

to pravico osebi dodelimo, sicer ji jo odvzamemo. Prav tako preverjamo, če je prijavljena oseba shranila spremembe, preden je zamenjala osebo, ki ji je pravice spreminjala. Če spremembe niso shranjene, prijavljeno osebo vprašamo, če želi shraniti spremembe.

Prijavljena oseba lahko tudi deaktivira osebe, ki so podale zahtevek za izbris osebnih podatkov. Po izbiri osebe in vnosa razloga, ki mora biti vnešen, se osebni podatki iz PB izbrišejo oziroma postavijo na neveljavne.

Vse spremembe in napake beležimo v sistem z namenom večjega nadzora.

## 6.4 Delavnice, individualna dela in donacije

Na sliki 6.4 je prikazan diagram poteka modula izvedb delavnic, individualnih del in donacij. Te tri smo prikazali na enem diagramu poteka, saj so s stališča načrtovanja povsem enaka.

Na začetku preverimo pravice in prilagodimo dostop. Nato po vnešenih kriterijih iskanja naložimo vse obstoječe podatke. V primeru odločitve za dodajanje ali urejanje prijavljeno osebo preusmerimo na okno, kjer vnaša potrebne podatke. V primeru urejanja napolnimo že obstoječe podatke iz PB. Pred shranjevanjem preverimo pravilnost vnešenih podatkov. Ob nepravilnem vnosu prikažemo napako, sicer shranimo.

Pri brisanju preverimo obstoj izbrane izvedbe, individualnega dela ali donacije. Po potrditvi s strani prijavljene osebe izbrano izbrišemo.

Vsako izvedbo, individualno delo in donacijo je možno izvoziti. V ta namen ustvarimo nov dokument, kamor zapišemo vse podatke o izbranem vnosu. Dokument prijavljena oseba shrani na zeleno lokacijo.

Vsako spremembo, napako in dostop zabeležimo v sistem.

## 6.5 Cilji

Na sliki 6.5 je prikazan diagram poteka modula ciljev udeležencev. Diagram je enak prejšnjemu, z razliko, da na začetku ponujamo še vnos evalvacije za

posamezen cilj. Torej v primeru, da želi prijavljena oseba dodati evalvacijo za izbran cilj, jo preusmerimo na okno, kjer vnaša potrebne podatke za evalvacijo izbranega cilja. Pred shranjevanjem se preveri pravilnost podatkov in ob nepravilnem vnosu prikažemo napako, sicer evalvacijo shranimo.

## 6.6 Dnevne prisotnosti

Na sliki 6.6 je prikazan diagram poteka modula dnevnih prisotnosti. Prijavljena oseba na tem mestu beleži, koliko časa je bil posamezen udeleženec prisoten v društvu.

Na začetku preverimo pravice prijavljene osebe in prilagodimo dostop. Nato naložimo že obstoječe podatke o dnevni prisotnosti udeležencev. Prijavljena oseba lahko nato s klikom na polje spremeni vrednost polja. Ob shranjevanju se vrednosti iz polj za posameznega udeleženca shranijo v PB.

Oseba lahko spremeni prikazani mesec in leto. Ob spremembi se preveri, če so spremembe shranjene. Če niso, prijavljeno osebo vprašamo po shranjevanju, sicer naložimo podatke za izbrani mesec in leto.

Dnevne prisotnosti je možno izvoziti. Podatki iz prikazanega meseca in leta se shranijo v nov dokument. Nastali dokument prijavljena oseba shrani na poljubno mesto.

## 6.7 Poročila

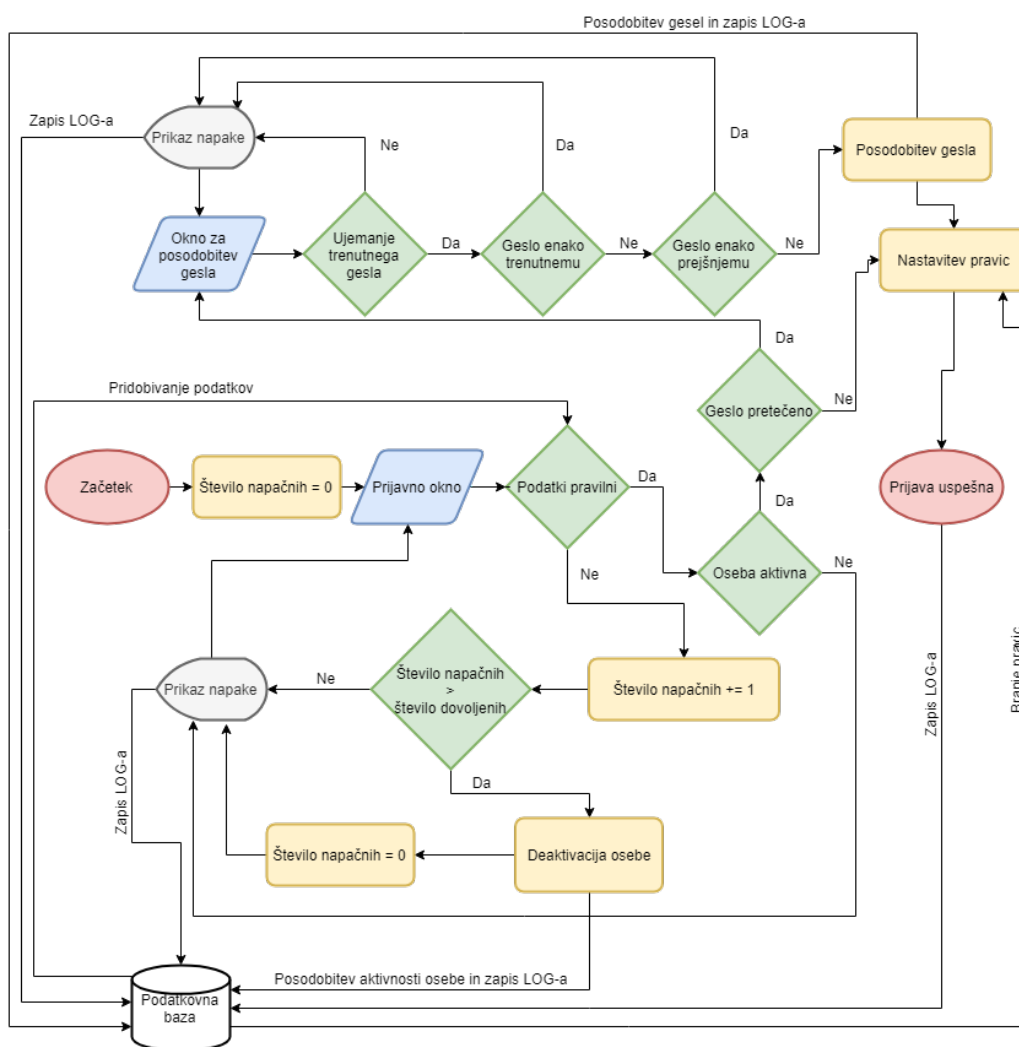
Na sliki 6.7 je prikazan diagram poteka modula poročil. Prijavljena oseba na tem mestu pridobiva vsa poročila, ki so: poročilo donacij, poročilo individualnih del, poročilo delavnic, poročilo oseb in splošno poročilo oseb.

Na začetku preverimo pravice prijavljene osebe in prilagodimo dostop. Oseba nato izbere tip poročila in obdobje. Pri poročilu donacij naložimo vse donacije v obdobju in prikažemo pripadajoče poročilo. Pri poročilu individualnih del naložimo vsa individualna dela v izbranem obdobju, jih preštujemo, preštujemo različne osebe ter prikažemo rezultate in razvrstitve.

Pri splošnem poročilu oseb naložimo dnevne prisotnosti oseb v izbranem obdobju, preštejemo različne osebe, ki so bile prisotne v društvu v izbranem obdobju, izračunamo povprečno prisotnost in prikažemo rezultate. Pri poročilu delavnic naložimo vse izvedbe delavnic v izbranem obdobju in jih razvrstimo v pripadajoče časovne kategorije. Če pridemo do mejne razvrstitve, torej je program v dvomih, v katero kategorijo naj delavnico razvrsti, vprašamo prijavljeno osebo, v katero od dveh kategorij naj to delavnico umesti. Nato prikažemo rezultate razvrščanj. Pri poročilu oseb naložimo dnevne prisotnosti oseb v izbranem obdobju in jih razvrstimo v časovne kategorije. Pri mejnih umestitvah program prijavljeno oseba vpraša, v katero od dveh kategorij naj osebo umesti. Na koncu prikažemo rezultate razvrščanj.

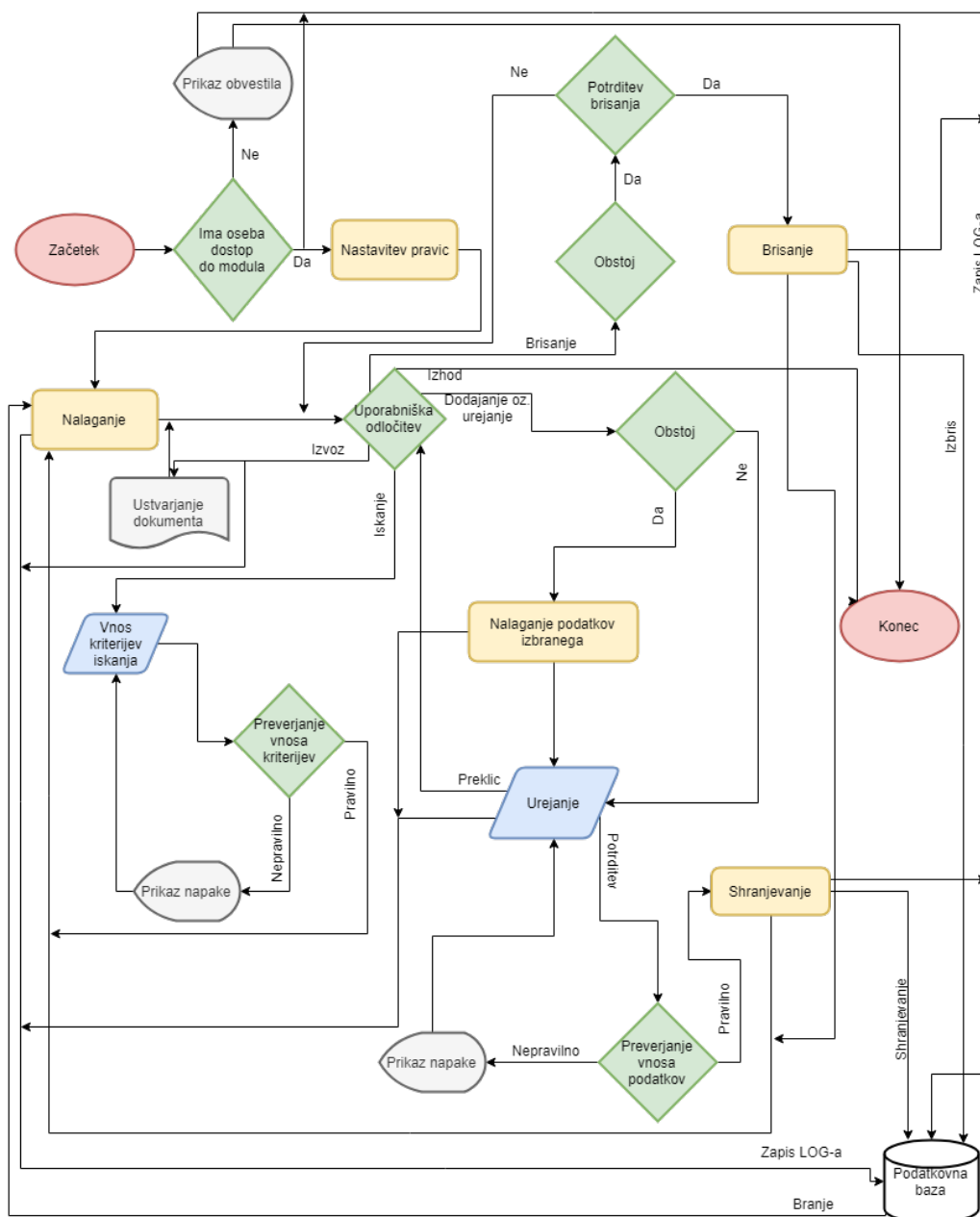
Vsa poročila je možno izvoziti. Podatki iz poročila se shranijo v dokument, ki ga prijavljena oseba shrani na poljubno mesto.

Vsako generiranje poročila se zapiše v PB za lažje spremljanje poteka programa.



Slika 6.1: Diagram poteka prijave.





Slika 6.4: Diagram poteka modula delavnic, individualnih del in donacij.









# Poglavje 7

## Pregled delovanja programa

V tem razdelku si bomo podrobneje ogledali ustvarjen produkt. Vsak razdelek bo imel prikazan diagram poteka, ki mu pripada, ter nekaj posnetkov zaslonov za lažjo predstavo. Za vsak del bomo tudi opisali, čemu služi.

Vsi podatki o osebah, vsa imena in opisi delavnic, vse opombe in vsi zaznamki individualnih del, ciljev in njihovih evalvacij, kot tudi vsi podatki dnevnih prisotnosti so izmišljeni in niso resnični.

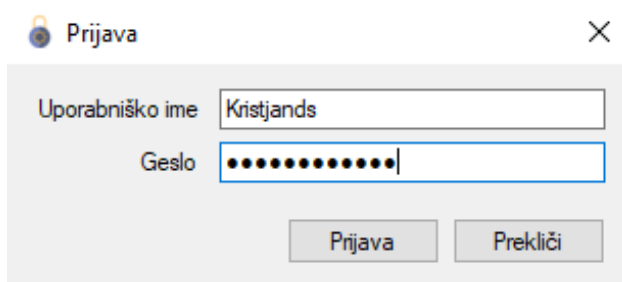
### 7.1 Pravice

Do programa lahko dostopajo le zaposleni s podanim uporabniškim imenom in geslom, ki jih registrira administrator oziroma zaposlen s podeljeno pravico dodajanja oseb. Program je razširjen in ne želimo, da imajo vsi zaposleni dostop do vseh modulov. To smo rešili z uvedbo pravic. Za dostop do modula smo ustvarili novo pravico in za vsako funkcionalnost le-tega prav tako. Ob prijavi naložimo pravice osebe, nato pa ob vsakem nalaganju posameznega modula blokiramo dostop do funkcionalnosti, do katerih prijavljena oseba nima pravic. Za lažje razumevanje se bomo v nadaljevanju sklicevali na odgovorno osebo, kadar bomo mislili na osebo, ki je zaposlena na društvu in ima podeljeno pravico za modul, na katerega se sklicujemo

## 7.2 Prijava

Ob zagonu programa prikažemo prijavno okno, kot ga vidimo na sliki 7.1. Vnesti moramo uporabniško ime in geslo. V primeru neuspešne prijave sporočimo napako. Če se uporabnik z določenim uporabniškim imenom zaradi napačnega gesla trikrat neuspešno prijavi, tega uporabnika blokiramo, tako da se v sistem ne more več prijaviti niti s pravilnim geslom. Deaktivacijo uporabnika zabeležimo v sistem. Ponovno ga lahko aktivira le odgovorna oseba.

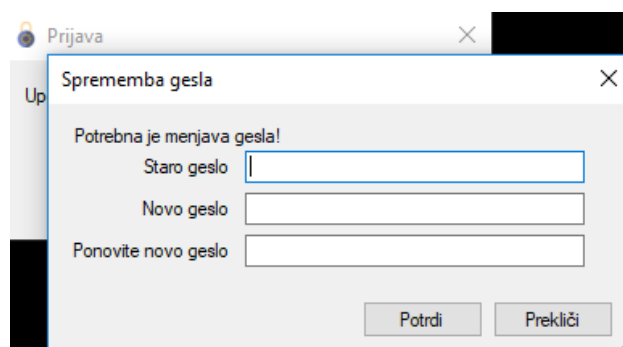
Ob pravilnem vnosu uporabniškega imena in gesla preverimo starost gesla. Če je geslo starejše od treh mesecev, prikažemo okno za posodobitev gesla, kot ga vidimo na sliki 7.2. Uporabnik mora vnesti trenutno geslo in novo geslo, ki pa ne sme biti enako prejšnjemu geslu, kot tudi ne geslu pred tem. Po posodobitvi gesla uporabnika prijavimo v sistem.



Slika 7.1: Prikaz prijave.

## 7.3 Meni

Osnovno okno je okno menija, ki je prikazano na sliki 7.3. Ob kliku na zelen modul ustvarimo novo nit, kjer zaženemo izbrani modul. Če je za izbrani modul nit že ustvarjena in ni ustavljena, nove niti ne zaženemo, ampak le postavimo okno modula v ospredje.



Slika 7.2: Prikaz menjave pretečenega gesla po prijavi.

## 7.4 Šifranti

Šifranti se delijo na štiri dele, in sicer na:

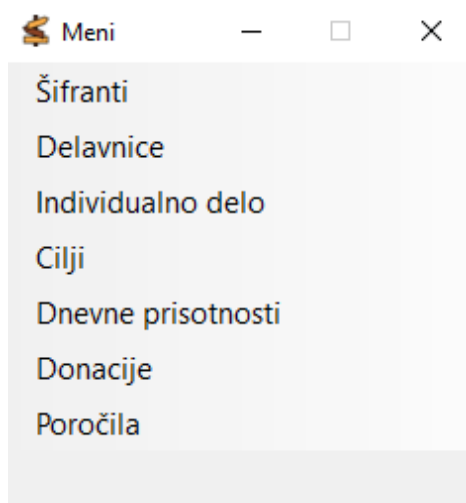
- delavnice,
- osebe,
- pravice,
- GDPR.

Prve tri si bomo ogledali v tem razdelku, na GDPR pa se bomo osredotočili v poglavju 8.

### 7.4.1 Delavnice

Ob izbiri urejanja šifrantov delavnic se odgovornim osebam najprej izpišejo vse delavnice, kot to vidimo na sliki 7.4. Program omogoča tudi iskanje in razvrščanje le-teh. Za iskanje je treba najprej izbrati kriterij in vnesti želeno besedno zvezo, za razvrščanje pa je treba le klikniti na stolpec, po katerem želimo razvrščati.

Če odgovorna oseba želi urediti delavnico, se ob kliku na urejanje prikaže okno, kot ga vidimo na sliki 7.5. Okno se napolni z že obstoječimi podatki, v



Slika 7.3: Prikaz okna menija.

primeru da le-ti obstajajo. Ob shranjevanju se podatki preverijo in shranijo v PB v primeru, da so podatki pravilni, v nasprotnem primeru se izpiše napaka. Enako okno se nam prikaže v primeru, ko želimo dodati novo delavnico, razlika je le v imenu okna.

Delavnico je možno tudi izbrisati, vendar če je bila delavnica že izvedena, program onemogoči njeno brisanje. Izbrisati se jo da le takrat, ko so izbrisane vse njene izvedbe. Brisanje delavnic se odsvetuje, saj drugače analiza ni nujno točna.

## 7.4.2 Osebe

Pri urejanju oseb se odgovornim osebam odpre okno, prikazano na sliki 7.6. Podobno kot pri delavnicah se tudi tu najprej izpišejo vse osebe. Prav tako je možno iskanje in razvrščanje le-teh. Iskanje poteka enako kot pri delavnicah, izbere se kriterij iskanja in vnese iskalni niz. Kriteriji iskanj oseb se razlikujejo od kriterijev iskanj delavnic. Izbere se lahko več kriterijev, vnešeni pa se izpisujejo zraven polja iskanja.

Na tem mestu imajo odgovorne osebe možnost za spreminjanje pozablje-

Sifra	Ime	Opis	Zadnja sprememba	Zadnja sprememba dne
0001	Kuhanje	Dnevno kuhanje v centru	Kristjan Dragovan Šimik	06-01-2019 14:27:28
0002	Čiščenje	Tedensko čiščenje v centru	Kristjan Dragovan Šimik	06-01-2019 14:28:04
0003	Pospravljanje okolice	Pospravljanje okolice centra	Kristjan Dragovan Šimik	06-01-2019 14:28:07
0005	Okrasitev centra	Okrasitev centra za posebne dogodke	Kristjan Dragovan Šimik	06-01-2019 18:17:21
0006	Družabne igre	Poučevanje s pomočjo družabnih iger	Kristjan Dragovan Šimik	06-01-2019 14:28:01
0007	Racunalištvo	Učenje uporabe racunalnika	Kristjan Dragovan Šimik	06-01-2019 14:28:23
0008	Dopustovanje	Odhod na dopust (letovanje/zimovanje) z uporabniki	Kristjan Dragovan Šimik	06-01-2019 14:28:27
0009	Gledališče	Obisk gledališča	Kristjan Dragovan Šimik	06-01-2019 14:28:33
0011	Muzej	Obisk muzeja	Kristjan Dragovan Šimik	06-01-2019 14:28:35
0013	Trgovina		Kristjan Dragovan Šimik	06-01-2019 14:28:40
0014	Izlet	Odhod na izlet	Kristjan Dragovan Šimik	06-01-2019 14:28:43
0015	Pogovorne delavnice		Kristjan Dragovan Šimik	06-01-2019 14:28:46
0016	Obisk župana		Kristjan Dragovan Šimik	06-01-2019 14:28:48
0004	Okrasitev okolice	Okrasitev okolice centra	Kristjan Dragovan Šimik	06-01-2019 18:17:11
0010	Kino	Obisk kina	Kristjan Dragovan Šimik	06-01-2019 14:28:56
0012	Sprehod	Sprehod po okolici	Kristjan Dragovan Šimik	06-01-2019 14:29:00
0017	Tedensko čiščenje		Kristjan Dragovan Šimik	17-01-2019 16:21:07

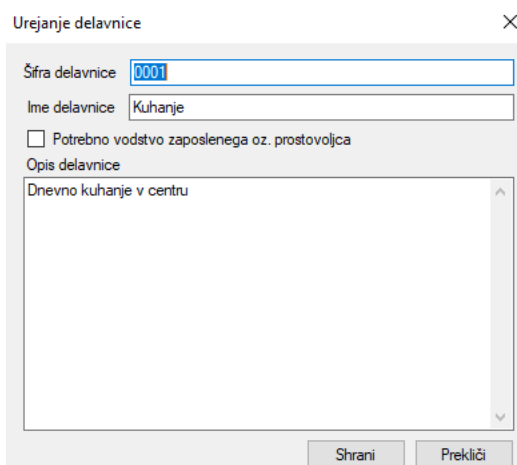
Slika 7.4: Prikaz okna šifranta delavnic.

nih gesel zaposlenih in administratorjev. Kot pri vsaki menjavi gesel, se tudi tu preverja, ali geslo ustreza kriterijem ter ali je geslo enako prejšnjim dvema.

Odgovorne osebe imajo možnost deaktivacije in ponovne aktivacije zaposlenih. To pomeni, da se osebo, ki je trikrat vnesla napačno osebo, lahko aktivira, da lahko ponovno dostopa do programa.

Vsako osebo se lahko tudi izbriše, postopek je enak prejšnjemu. Če je oseba že bila prisotna na neki dejavnosti, bo sistem onemogočil brisanje. Da osebo lahko izbrišemo, jo moramo najprej odstraniti iz vseh dejavnosti. Brisanje oseb se odsvetuje, saj drugače analiza ni točna.

Ob urejanju in dodajanju oseb se odgovornim osebam odpre okno, prikazano na sliki 7.7. Enako kot pri delavnicah se tudi tu okno napolni z obstoječimi podatki, če ti obstajajo. Pred shranjevanjem se podatki preverijo. V primeru, da so obvezna polja prazna, se prikaže napaka in označijo se polja, pri katerih je do napake prišlo. Preverjanje podatkov poteka na podlagi izbranega tipa osebe. Če se izbere tip osebe administrator ali zaposleni je poleg ostalih obvezno vnesti še polji uporabniško ime in email. V primeru,



Slika 7.5: Prikaz okna za urejanje šifrantov delavnic.

da se vnaša udeleženca društva ali prostovoljca pa je dovolj vnesti ime, priimek, spol in datum rojstva. Če so bili podatki ob preverjanju pravilni, torej so bila vsa obvezna polja pravilno vnešena, se vrednosti v PB posodobijo oziroma dodajo.

### 7.4.3 Pravice

Pri podeljevanju in odstranjevanju pravic odgovorna oseba izbere osebo, kateri želi podeliti oziroma odvzeti pravice. Ko osebo izbere, se iz PB naložijo podatki, katere pravice izbrana oseba že ima. Odgovorna oseba nato v seznamu obkljuka oziroma odkljuka pravice. Ob shranjevanju se nato seznam prebere, odkljukane pravice odstrani, če le-te obstajajo, obkljukane pa doda, če še ne obstajajo. Oseba, kateri so se pravice posodobile, bo imela dostop do modulov oziroma blokado le-teh ob naslednji prijavi, saj se pravice naložijo le takrat.

Tip uporabnika	Ime, priimek	Datum rojstva	Email	Uporabniško ime	Aktivnost
Admin	Kristjan Dragovan Šimik	06-06-1996	kristjands@outlook.com	Kristjands	Aktiven
Zaposleni	Tina Šimik	04-06-1975	tina.simik.75@yahoo.com	TinaŠimik	Aktiven
Uporabniki	Franc Novak	25-03-1965			Neaktiven
Uporabniki	Jožef Krajc	01-10-1965			Neaktiven
Uporabniki	Ivan Krajnc	16-12-1970			Neaktiven
Uporabniki	Žiga Potocnik	14-06-1974			Neaktiven
Uporabniki	Petra Kos	01-07-1977			Neaktiven
Uporabniki	Ana Zupancic	06-05-1980			Neaktiven
Uporabniki	Marija Horvat	13-08-1969			Neaktiven
Uporabniki	Nina Krajnc	16-05-1970			Neaktiven
Uporabniki	Maja Makica	13-10-1986			Neaktiven
Uporabniki	Barbara Blagorja	08-10-1990			Neaktiven
Uporabniki	Taja Jerebica	06-01-1975			Neaktiven
Uporabniki	Katrina Ojo	06-03-1988			Neaktiven
Uporabniki	Martina Apotekar	13-09-1977			Neaktiven
Prostovoljci	Miha Delcnjak	11-03-1978			Neaktiven
Prostovoljci	Martina Harapin	18-05-1990			Neaktiven
Prostovoljci	Andrej Plimon	23-01-1987			Neaktiven
Prostovoljci	Barbara Poderžan	27-06-1986			Neaktiven
Uporabniki	Katja Vidmar	08-08-1977			Neaktiven
Uporabniki	Tim Mlakar	04-04-1979			Neaktiven

Slika 7.6: Prikaz okna šifranta oseb.

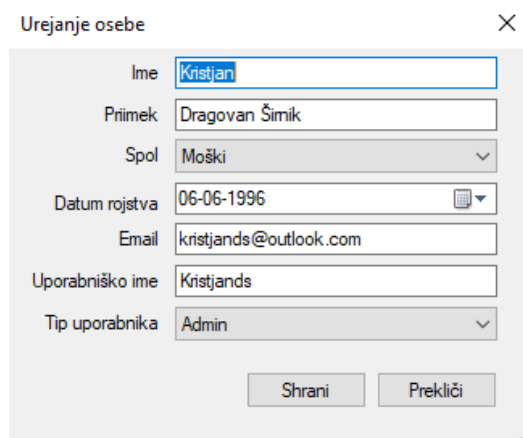
## 7.5 Delavnice

Na tem mestu bodo odgovorne osebe vnašale in pregledovale vse izvedbe delavnic. Vnos vseh izvedb je obvezen, saj sicer poročilo ne bo pravilno.

Najprej se odpre okno, prikazano na sliki 7.9. Tu so prikazane - v primeru, da nimamo nastavljenega iskalnika po datumu - vse izvedbe tekočega leta. Ker je izvedb lahko veliko, ima odgovorna oseba možnost iskanja po enem ali več kriterijih. Prav tako je na voljo možnost razvrščanja, ki se zgodi ob kliku na stolpec. Možni kriteriji za iskanje so:

- iskanje po delavnici,
- iskanje po izvajalcih,
- iskanje po udeležencih,
- iskanje v določenem razponu datuma.

Vnešeni kriteriji se izpisujejo zraven vnosnih polj iskalnika.



Ime	Kristjan
Primek	Dragovan Šimik
Spol	Moški
Datum rojstva	06-06-1996
Email	kristjands@outlook.com
Uporabniško ime	Kristjands
Tip uporabnika	Admin

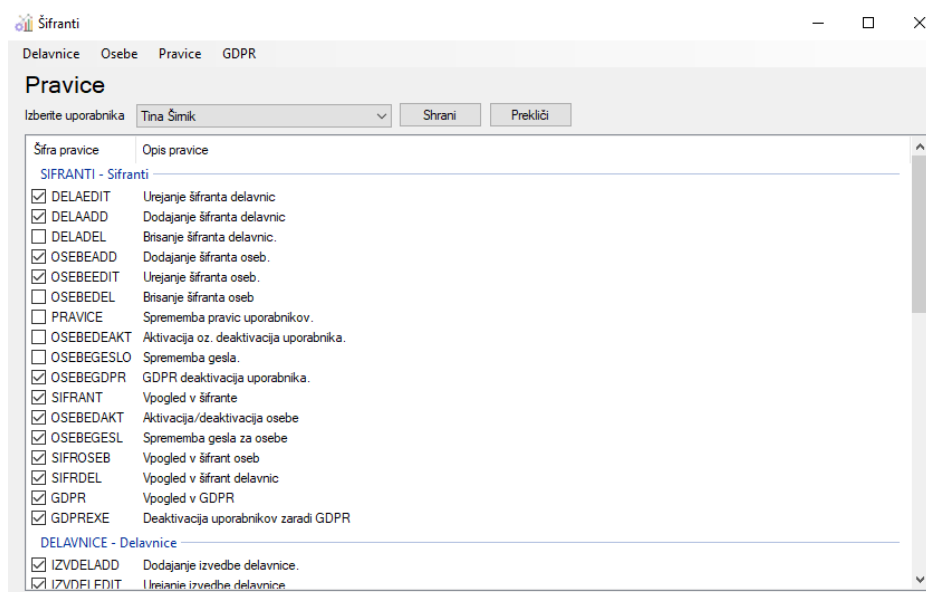
Shrani Prekliči

Slika 7.7: Prikaz okna za urejanje šifranta oseb.

Odgovorna oseba ima možnost dodajanja, urejanja, izvoza v PDF datoteko in brisanja, katerega pa ne priporočamo, saj lahko pride do drugačne razvrstitve delavnice v časovne kategorije. Pri dodajanju in urejanju se odpre okno, ki omogoča vnos podrobnosti o izvedbi, prikazano na sliki 7.10. Vsaka izvedba mora imeti zabeleženo delavnico, ki so jo izvajali, datum izvedbe, čas začetka in konca. Prav tako mora biti naveden vsaj en izvajalec delavnice, kot tudi en udeleženec. Po tem, kdo je izvajalec delavnice, se delavnice med seboj razlikujejo. Privzeto so izvajalci zaposleni in prostovoljci, vendar lahko ob dodajanju šifranta delavnice odgovorna oseba določi, da njena izvedba ne potrebuje nadzora. V tem primeru je lahko izvajalec delavnice tudi udeleženec društva.

Ob brisanju izvedbe program odgovorni osebi odpre okno za potrditev brisanja. To se zgodi zaradi varnosti, da ne prihaja do pomot brisanja. Prav tako je mogoče izbrisati le eno delavnico naenkrat.

Vsako izvedbo delavnice je možno shraniti v PDF dokument. Po izbiri delavnic, ki jih želimo izvoziti, oziroma po shranjevanju delavnice, se pojavi okno za izbiro mesta shranjevanja novo nastale datoteke. Ustvarijo se novi PDF dokumenti na izbrani lokaciji z vnaprej predpisanim imenom, ki se določi na podlagi imena in datuma izvedbe delavnice.



Slika 7.8: Prikaz okna za dodeljevanje in odstranjevanje pravic.

## 7.6 Individualna dela

Na tem mestu bodo odgovorne osebe vnašale in pregledovale vsa individualna dela. Modul služi interni analizi, saj zaposlene zanima, v katerem mesecu oziroma obdobju je izvedenih največ individualnih del, kar jim omogoča boljše pripravo v prihodnosti. Zanima jih tudi količina in obdobje obiskov individualnih del posameznika, saj se lahko v nadaljevanju bolj posvetijo udeležencu, ki je v tistem obdobju izvedel več individualnih del.

Najprej se odpre okno, prikazano na sliki 7.11, ki je podobno oknu za vnos izvedb delavnic. Tu so prikazana individualna dela tekočega leta, v primeru, da ni nastavljeno iskanje po datumu. Ker je individualnih del lahko veliko, ima odgovorna oseba možnost iskaja po enem ali več kriterijih, kot tudi možnost razvrščanja, ki se zgodi ob kliku na stolpec. Možni kriteriji za iskanje so:

- iskanje po udeležencih,
- iskanje po izvajalcih,

The screenshot shows the 'Delavnice' application interface. On the left, there are navigation options: 'Dodaj', 'Uredi', 'Izbrisi', 'Izpis', and 'Izvozi v PDF'. The main area contains search filters: 'Išči po' (empty), 'Dodaj filter', 'Išči', 'Osveži', and 'Delavnica: kuhanje'. Below the filters, there are checkboxes for 'Išči po datumu' (checked) and 'Prikaži vse' (unchecked), with date ranges '01-01-2018' and '31-12-2018'. The main table has the following columns: 'Delavnica', 'Datum', 'Trajanje', 'Izvajalci', 'Obiskovalci', and 'Zadnja sprememba'. The table contains 20 rows of task data.

Delavnica	Datum	Trajanje	Izvajalci	Obiskovalci	Zadnja sprememba
Kuhanje	04-01-2018 00:00:00	03:00	Katarina Hadner	Franc Novak, Jožef Kralj, Ivan Krajnc, Nina Krajnc, Maja Makica	Kristjan Dragovan Šimik (
Kuhanje	05-01-2018 00:00:00	03:00	Tim Mizoni	Jožef Kralj, Petra Kos, Ana Zupancic, Katarina Ojo, Martina Apotekar	Kristjan Dragovan Šimik (
Kuhanje	08-01-2018 00:00:00	03:00	Tim Mizoni	Ziga Potocnik, Petra Kos, Ana Zupancic, Barbara Blagonja, Taja Jerebica	Kristjan Dragovan Šimik (
Kuhanje	09-01-2018 00:00:00	03:00	Luka Tonc	Katarina Ojo, Katja Vidmar, Tim Mlakar, Martin Rovšček, Marko Cimmanjanc	Kristjan Dragovan Šimik (
Kuhanje	10-01-2018 00:00:00	03:00	Luka Tonc	Martina Apotekar, Tim Mlakar, Katja Žilic, Martina Rovšček, Marko Cimmanjanc	Kristjan Dragovan Šimik (
Kuhanje	11-01-2018 00:00:00	03:00	Luka Tonc	Ana Zupancic, Marija Horvat, Taja Jerebica, Katarina Ojo, Martina Apotekar, Katja Vidmar	Kristjan Dragovan Šimik (
Kuhanje	12-01-2018 00:00:00	03:00	Tim Mizoni	Ivan Krajnc, Petra Kos, Ana Zupancic, Nina Krajnc, Katarina Ojo, Martina Apotekar, Katja Vidmar	Kristjan Dragovan Šimik (
Kuhanje	15-01-2018 00:00:00	03:00	Katarina Hadner	Ivan Krajnc, Ziga Potocnik, Petra Kos, Taja Jerebica, Katarina Ojo	Kristjan Dragovan Šimik (
Kuhanje	16-01-2018 00:00:00	03:00	Katarina Hadner	Maja Makica, Barbara Blagonja, Taja Jerebica, Katja Vidmar, Tim Mlakar	Kristjan Dragovan Šimik (
Kuhanje	17-01-2018 00:00:00	03:00	Tina Simik	Ivan Krajnc, Ziga Potocnik, Petra Kos, Marko Panger, Katja Žilic	Kristjan Dragovan Šimik (
Kuhanje	18-01-2018 00:00:00	03:00	Tina Simik	Ziga Potocnik, Petra Kos, Ana Zupancic, Katarina Ojo, Martina Apotekar	Kristjan Dragovan Šimik (
Kuhanje	19-01-2018 00:00:00	03:00	Katarina Hadner	Jožef Kralj, Ivan Krajnc, Barbara Blagonja, Martina Apotekar, Tim Mlakar	Kristjan Dragovan Šimik (
Kuhanje	22-01-2018 00:00:00	03:00	Katarina Hadner	Ivan Krajnc, Ziga Potocnik, Petra Kos, Katja Vidmar, Stanko Jaunik	Kristjan Dragovan Šimik (
Kuhanje	23-01-2018 00:00:00	03:00	Tina Simik	Ziga Potocnik, Ana Zupancic, Maja Makica, Taja Jerebica, Martina Apotekar	Kristjan Dragovan Šimik (
Kuhanje	24-01-2018 00:00:00	03:00	Tina Simik	Barbara Blagonja, Taja Jerebica, Katja Vidmar, Marko Panger, Marko Cimmanjanc	Tina Simik (
Kuhanje	25-01-2018 00:00:00	03:00	Tim Mizoni	Barbara Blagonja, Katarina Ojo, Martina Apotekar, Katja Žilic, Marko Cimmanjanc	Tina Simik (
Kuhanje	26-01-2018 00:00:00	03:00	Katarina Hadner	Petra Kos, Ana Zupancic, Nina Krajnc, Maja Makica, Taja Jerebica	Tina Simik (
Kuhanje	29-01-2018 00:00:00	03:00	Luka Tonc	Jožef Kralj, Ivan Krajnc, Barbara Blagonja, Katja Vidmar, Tim Mlakar	Tina Simik (
Kuhanje	30-01-2018 00:00:00	03:00	Katarina Hadner	Ana Zupancic, Marija Horvat, Nina Krajnc, Katarina Ojo, Katja Vidmar	Tina Simik (
Kuhanje	31-01-2018 00:00:00	03:00	Katarina Hadner	Tim Mlakar, Stanko Jaunik, Marko Panger, Marko Cimmanjanc	Tina Simik (
Kuhanje	01-02-2018 00:00:00	03:00	Tina Simik	Ziga Potocnik, Maja Makica, Petra Kos, Katarina Ojo, Nina Krajnc	Kristjan Dragovan Šimik (
Kuhanje	01-04-2018 00:00:00	03:00	Katarina Hadner	Maja Makica, Petra Kos, Martina Apotekar, Katja Vidmar, Jožef Kralj	Kristjan Dragovan Šimik (

Slika 7.9: Prikaz osnovnega okna izvedb delavnic.

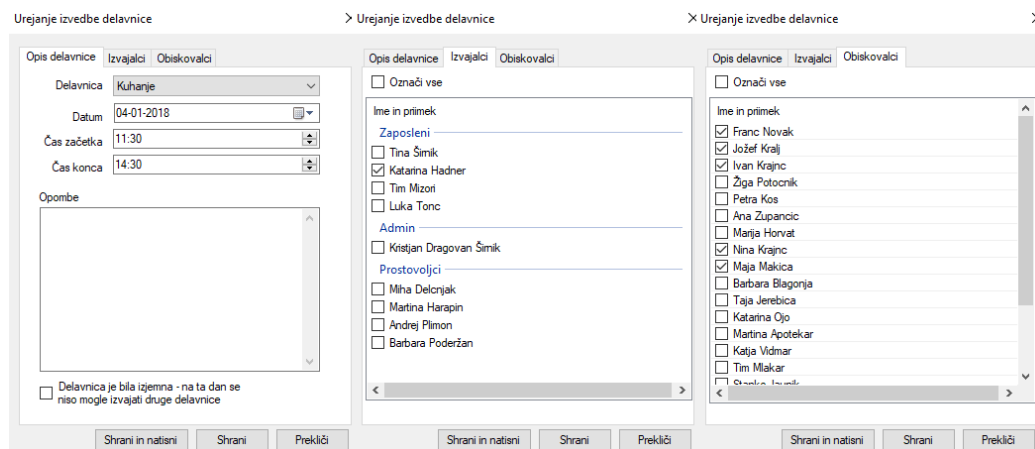
- iskanje v določenem razponu datuma.

Vnešeni kriteriji se izpisujejo zraven vnosnih polj iskalnika.

Tudi pri tem modulu ima odgovorna oseba možnost dodajanja, urejanja, izvoza v PDF datoteko in brisanja. Pri dodajanju in urejanju se odpre okno, ki omogoča vnos podrobnosti o posameznem individualnem delu, prikazano na sliki 7.12. Vsako individualno delo mora imeti zabeleženega udeleženca, na katerega se je individualno delo nanašalo, datum dela, čas začetka in konca. Prav tako mora biti naveden vsaj en izvajalec. Za razliko od izvedb delavnic, kjer so lahko pri določenih delavnicah izvajalci tudi udeleženci, so tu lahko le zaposleni in prostovoljci.

Ob brisanju individualnega dela program odgovorni osebi odpre okno za potrditev in tudi tu je mogoče izbrisati le eno delo naenkrat.

Vsako individualno delo je možno shraniti tudi v PDF dokument. Po izbiri del, ki jih želimo izvoziti, oziroma po shranjevanju le-tega, se pojavi okno za izbiro mesta shranjevanja nove datoteke. Ustvarijo se novi PDF dokumenti na izbrani lokaciji z vnaprej predpisanim imenom, ki se določi na podlagi imena in priimka udeleženca ter datuma izvedbe dela.



Slika 7.10: Prikaz okna za dodajanje in urejanje izvedbe delavnice.

## 7.7 Cilji

V tem modulu bodo odgovorne osebe vnašale cilje udeležencev in njihove evalvacije. Modul je bil narejen zaradi internih potreb, za lažje iskanje ciljev udeležencev in za lažje spremljanje in vodenje evalvacij le-teh.

Na sliki 7.13 je prikazano osnovno okno. Okno je zaradi konsistentnosti za lažjo uporabo programa podobno prejšnjima dvema. Kot pri prejšnjih dveh primerih tudi tu prikažemo cilje tekočega leta. Zaradi lažjega iskanja določenih ciljev imamo tudi na tem modulu možnost iskanja po enem ali več kriterijih, kot tudi razvrščanja. Možni kriteriji za iskanje so:

- iskanje po udeležencih,
- iskanje po cilju,
- iskanje po številu evalvacij,
- iskanje v določenem razponu datuma.

Kot pri prejšnjih dveh modulih, ima odgovorna oseba tudi tu možnost dodajanja, urejanja, izvoza v PDF datoteko in brisanja cilja. Pri dodajanju in urejanju ciljev se odpre okno, prikazano na sliki 7.14, ki omogoča vnos

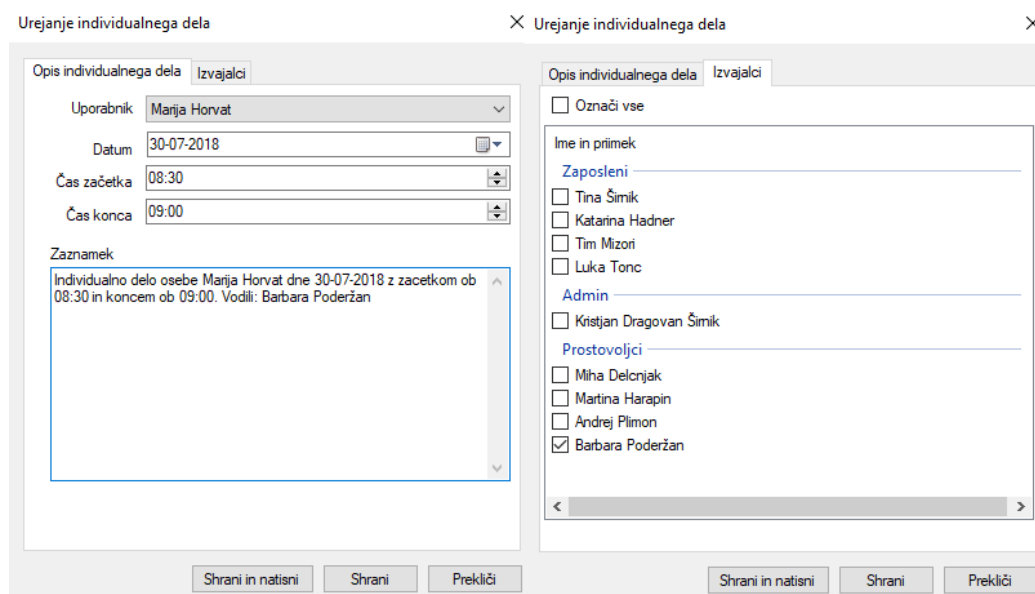
Uporabnik	Datum	Trajanje	Izvajalci	Zadnja sprememba	Zadnja sprememba dne
Ária Zupanc	16-04-2018 ...	00:30	Andrej Plimon	Kristjan Dragovan Šimik	06-01-2019 21:31:27
Marija Horvat	30-07-2018 ...	00:30	Barbara Poderžan	Kristjan Dragovan Šimik	06-01-2019 21:31:27
Nina Kranj	05-11-2018 ...	00:30	Katarina Hadner	Kristjan Dragovan Šimik	06-01-2019 21:31:27
Katja Vidmar	07-03-2018 ...	00:30	Barbara Poderžan	Kristjan Dragovan Šimik	06-01-2019 21:34:57
Stanko Jaunik	09-08-2018 ...	00:30	Katarina Hadner	Kristjan Dragovan Šimik	06-01-2019 21:34:57
Katja Žlic	12-06-2018 ...	00:30	Martina Harapin	Kristjan Dragovan Šimik	06-01-2019 21:36:23
Katarina Ojo	29-01-2018 ...	00:30	Tina Šimik	Kristjan Dragovan Šimik	06-01-2019 21:39:10
Barbara Blagonja	01-03-2018 ...	00:30	Martina Harapin	Kristjan Dragovan Šimik	06-01-2019 21:39:11
Katarina Ojo	03-04-2018 ...	00:30	Tim Mizori	Kristjan Dragovan Šimik	06-01-2019 21:39:11
Barbara Blagonja	11-05-2018 ...	00:30	Miha Delcrljak	Kristjan Dragovan Šimik	06-01-2019 21:39:11
Katarina Ojo	11-06-2018 ...	00:30	Luka Tonc	Kristjan Dragovan Šimik	06-01-2019 21:39:11
Barbara Blagonja	17-07-2018 ...	00:30	Andrej Plimon	Kristjan Dragovan Šimik	06-01-2019 21:39:11
Katarina Ojo	15-08-2018 ...	00:30	Luka Tonc	Kristjan Dragovan Šimik	06-01-2019 21:39:11
Barbara Blagonja	17-09-2018 ...	00:30	Andrej Plimon	Kristjan Dragovan Šimik	06-01-2019 21:39:11
Katarina Ojo	17-10-2018 ...	00:30	Tim Mizori	Kristjan Dragovan Šimik	06-01-2019 21:39:11
Barbara Blagonja	21-11-2018 ...	00:30	Andrej Plimon	Kristjan Dragovan Šimik	06-01-2019 21:39:12
Katarina Ojo	21-12-2018 ...	00:30	Katarina Hadner	Kristjan Dragovan Šimik	06-01-2019 21:39:12
Žiga Potoonik	11-01-2018 ...	00:30	Tim Mizori	Kristjan Dragovan Šimik	06-01-2019 21:40:05
Petra Kos	24-01-2018 ...	00:30	Katarina Hadner, Martina Harapin	Kristjan Dragovan Šimik	06-01-2019 21:40:05
Jožef Kralj	21-02-2018 ...	00:30	Barbara Poderžan	Kristjan Dragovan Šimik	06-01-2019 21:40:05
Ivan Kranj	06-03-2018 ...	00:30	Luka Tonc	Kristjan Dragovan Šimik	06-01-2019 21:40:05
Žiga Potoonik	19-03-2018 ...	00:30	Luka Tonc	Kristjan Dragovan Šimik	06-01-2019 21:40:05
Petra Kos	30-03-2018 ...	00:30	Tim Mizori, Andrej Plimon	Kristjan Dragovan Šimik	06-01-2019 21:40:06

Slika 7.11: Prikaz osnovnega okna individualnih del.

podrobnosti posameznega cilja. Vsak cilj mora biti opisan, vsebovati mora datum, kdaj je bil cilj ustvarjen in na katerega udeleženca se navezuje. Za razliko od prejšnjih dveh imamo tukaj možnost tudi dodajanja, urejanja in brisanja evalvacij. Evalvacije so prikazane v seznamu in so urejene po datumu, kot to vidimo na sliki 7.15. Vsaka evalvacija zahteva opis in datum. Evalvacijo lahko dodamo tudi iz glavnega okna, paziti je potrebno le, da je izbran pravi cilj, na katerega jo želimo dodati. Možno je tudi označiti, da je bil cilj že zaključen. V tem primeru se pojavi polje, kamor se vnese datum zaključka. V primeru, da je cilj zaključen, se onemogoči naknadno dodajanje, urejanje in brisanje evalvacij.

Ob brisanju ciljev, enako kot prej, program odgovorni osebi odpre okno za potrditev. Ob brisanju se izbrišejo tudi vse evalvacije izbranega cilja. Prav tako je možno izbrisati le en cilj naenkrat.

Vsak cilj je možno shraniti v PDF dokument. Postopek je enak prejšnjima dvema, ime datoteke pa je sestavljeno iz imena in priimka osebe ter datuma cilja.



Slika 7.12: Prikaz okna za dodajanje in urejanje individualnega dela.

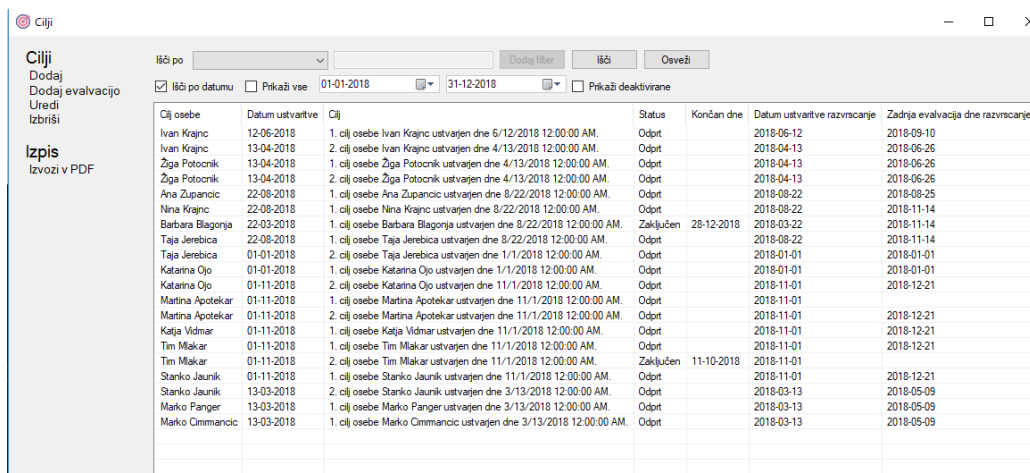
## 7.8 Donacije

Modul služi za hranjenje in analizo prejetih donacij. Društvo mora imeti zabeleženo, koliko donacij so prejeli in kakšne so te donacije bile - denarne ali materialne. Zaposlene pa zanima, koliko je bilo prejetih donacij v določenem obdobju.

Kot je prikazano na sliki 7.16, je tudi tu osnovno okno podobno prejšnjim. Izpišejo se nam podatki o vseh donacijah tekočega leta. Prav tako imajo odgovorne osebe možnost razvrščanja po stolpcih in iskanja po:

- osebi vnosa,
- opisu donacije,
- določenem razponu datuma.

Tudi tu imajo odgovorne osebe možnost dodajanja, urejanja, izvoza v PDF datoteko in brisanja donacij. Pri dodajanju in urejanju se odpre okno, ki je prikazano na sliki 7.17, kamor vnesemo podrobnosti donacije. Vsaka



The screenshot shows a web application window titled 'Cilji'. On the left, there is a sidebar with navigation options: 'Dodaj', 'Dodaj evalvacijo', 'Uredi', 'Izbrisi', and 'Izvozi v PDF'. The main area contains a search bar and a table of goals. The table has columns: 'Cilj osebe', 'Datum ustvaritve', 'Cilj', 'Status', 'Končan dne', 'Datum ustvaritve razvrščanje', and 'Zadnja evalvacija dne razvrščanje'. The table lists 20 goals, each with a unique ID and a description of the goal and its creation date. The status of each goal is either 'Odprt' or 'Zaključen'. The end date and evaluation dates are also provided for each goal.

Cilj osebe	Datum ustvaritve	Cilj	Status	Končan dne	Datum ustvaritve razvrščanje	Zadnja evalvacija dne razvrščanje
Ivan Krajnc	12-06-2018	1. cilj osebe Ivan Krajnc ustvarjen dne 6/12/2018 12:00:00 AM.	Odprt		2018-06-12	2018-09-10
Ivan Krajnc	13-04-2018	2. cilj osebe Ivan Krajnc ustvarjen dne 4/13/2018 12:00:00 AM.	Odprt		2018-04-13	2018-06-26
Ziga Fotočnik	13-04-2018	1. cilj osebe Ziga Fotočnik ustvarjen dne 4/13/2018 12:00:00 AM.	Odprt		2018-04-13	2018-06-26
Ziga Fotočnik	13-04-2018	2. cilj osebe Ziga Fotočnik ustvarjen dne 4/13/2018 12:00:00 AM.	Odprt		2018-04-13	2018-06-26
Ana Zupancic	22-09-2018	1. cilj osebe Ana Zupancic ustvarjen dne 8/22/2018 12:00:00 AM.	Odprt		2018-09-22	2018-09-25
Nina Krajnc	22-09-2018	1. cilj osebe Nina Krajnc ustvarjen dne 8/22/2018 12:00:00 AM.	Odprt		2018-09-22	2018-11-14
Barbara Blagorja	22-03-2018	1. cilj osebe Barbara Blagorja ustvarjen dne 8/22/2018 12:00:00 AM.	Zaključen	28-12-2018	2018-03-22	2018-11-14
Taja Jerebica	22-09-2018	1. cilj osebe Taja Jerebica ustvarjen dne 8/22/2018 12:00:00 AM.	Odprt		2018-09-22	2018-11-14
Taja Jerebica	01-01-2018	2. cilj osebe Taja Jerebica ustvarjen dne 1/1/2018 12:00:00 AM.	Odprt		2018-01-01	2018-01-01
Katarina Ojo	01-01-2018	1. cilj osebe Katarina Ojo ustvarjen dne 1/1/2018 12:00:00 AM.	Odprt		2018-01-01	2018-01-01
Katarina Ojo	01-11-2018	2. cilj osebe Katarina Ojo ustvarjen dne 11/1/2018 12:00:00 AM.	Odprt		2018-11-01	2018-12-21
Martina Apotekar	01-11-2018	1. cilj osebe Martina Apotekar ustvarjen dne 11/1/2018 12:00:00 AM.	Odprt		2018-11-01	2018-11-01
Martina Apotekar	01-11-2018	2. cilj osebe Martina Apotekar ustvarjen dne 11/1/2018 12:00:00 AM.	Odprt		2018-11-01	2018-12-21
Katja Vidmar	01-11-2018	1. cilj osebe Katja Vidmar ustvarjen dne 11/1/2018 12:00:00 AM.	Odprt		2018-11-01	2018-12-21
Tim Mikar	01-11-2018	1. cilj osebe Tim Mikar ustvarjen dne 11/1/2018 12:00:00 AM.	Odprt		2018-11-01	2018-12-21
Tim Mikar	01-11-2018	2. cilj osebe Tim Mikar ustvarjen dne 11/1/2018 12:00:00 AM.	Zaključen	11-10-2018	2018-11-01	2018-11-01
Stanko Jaunik	01-11-2018	1. cilj osebe Stanko Jaunik ustvarjen dne 11/1/2018 12:00:00 AM.	Odprt		2018-11-01	2018-12-21
Stanko Jaunik	13-03-2018	2. cilj osebe Stanko Jaunik ustvarjen dne 3/13/2018 12:00:00 AM.	Odprt		2018-03-13	2018-05-09
Marko Panger	13-03-2018	1. cilj osebe Marko Panger ustvarjen dne 3/13/2018 12:00:00 AM.	Odprt		2018-03-13	2018-05-09
Marko Cimmanic	13-03-2018	1. cilj osebe Marko Cimmanic ustvarjen dne 3/13/2018 12:00:00 AM.	Odprt		2018-03-13	2018-05-09

Slika 7.13: Prikaz osnovnega okna ciljev.

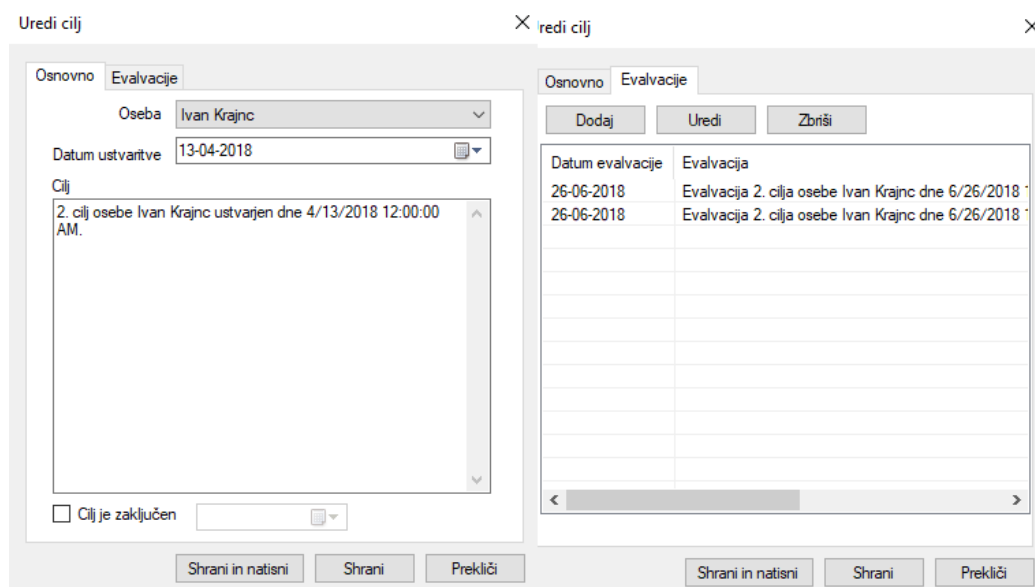
donacija zahteva vnos datuma prejema in opis. Datum prejema se predlaga kot trenutni datum, osebo vnosa pa določimo iz trenutno prijavljene osebe in je ni možno spremeniti. Zapišemo lahko tudi doniran znesek, če je bila donacija take oblike.

Ob brisanju ni sprememb, program odgovorni osebi odpre okno za potrditev in prav tako je možno izbrisati le eno donacijo naenkrat.

Tudi donacijo je mogoče izvoziti v PDF dokument. Postopek je enak prejšnjim, ime datoteke je sestavljeno iz datuma prejema donacije in nekega naključnega niza, v primeru, da je bilo več donacij na isti datum.

## 7.9 Dnevne prisotnosti

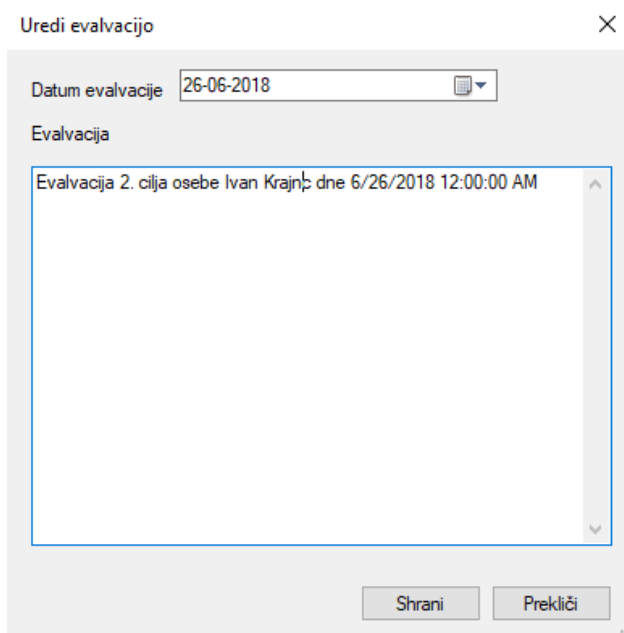
Modul dnevnih prisotnosti se uporablja pri zapisovanju, koliko časa je bil posamezen udeleženec prisoten v društvu, tudi če ni obiskal nobene delavnice. Odgovorne osebe morajo ministrstvu sporočiti, koliko udeležencev je bilo prisotnih v določenih časovnih kategorijah. Prav tako morajo sporočiti povprečno število ur prisotnosti v mesecu na enega udeleženca ter število udeležencev, ki je tisti mesec obiskalo društvo.



Slika 7.14: Prikaz okna za dodajanje in urejanje cilja.

Na sliki 7.18 je prikazano osnovno in edino okno tega modula. Na prvi pogled je podobno Excelu, saj imamo v prvem stolpcu našete vse udeležence, v naslednjih pa so našeti dnevi prikazanega meseca. V polja se zapisuje decimalne številke, koliko ur je bil na tisti dan prisoten določen udeleženec. Med meseci se odgovorna oseba lahko premika s pomočjo puščic na vrhu okna, oziroma tako, da izbere zeleno leto in mesec. V primeru, da spremeni datum prikazanega meseca in da trenutni ni bil shranjen, a ima spremembe, program prikaže pogovorno okno za shranjevanje. Če oseba želi shraniti spremembe, se spremembe shranijo, preden se naloži izbrani mesec, sicer se spremembe zavržejo. Da program ve, ali je prišlo do spremembe ali ne, imamo na objektu dnevne prisotnosti oznako, ali se je ta posodobil ali ne. S tem tudi zmanjšamo število posodobitev, saj se pri shranjevanju najprej preveri, ali so bile na izbranem objektu spremembe. Če niso, se ne naredi posodobitve v PB, ampak se objekt ignorira.

Dnevne prisotnosti se lahko izvozijo v PDF dokument. Odgovorna oseba izbere zelen mesec in leto ter pritisne gumb za izvoz in izbere pot, kamor



Slika 7.15: Prikaz okna za dodajanje in urejanje evalvacije cilja.

se bo datoteka shranila. Ime datoteke je sestavljeno iz izbranega meseca in leta. Zaradi velikosti tabele in velikosti papirja formata A4 smo morali tabelo razbiti na več delov. Orientacija strani je ležeča, z namenom, da se lahko izpišejo vsi stolpci skupaj. Število vrstic pa smo omejili. Če število vrstic preseže omejitve, se zgodi razbijanje na več strani. Pri omejitvi se stran zaključi in na novi strani se na vrhu tabele, zaradi lažjega branja, ponovno izpišejo datumi meseca.

## 7.10 Poročila

Modul poročil se uporablja za vsa razvrščanja in izpise, ki jih društvo potrebuje za notranje analize, kot tudi za ministrstvo. Odgovorna oseba se odloči med petimi različnimi vrstami poročil, vsaka od njih pa ima različne vrste vhodnih parametrov. Pri vsakem pa je treba določiti obdobje, ki ga želimo analizirati. Vrste poročil in možni dodatni parametri so:

Datum	Opis	Znesek	Oseba vnosa	Zadnja sprememba	Zadnja sprememba dne
24-05-2018	1. denarna donacija 83.45 € dne 5/24/2018 12:00:00 AM	83.45 €	Kristjan Dragovan Šimik	Kristjan Dragovan Šimik	10-01-2019
10-02-2018	2. nedenarna donacija dne 2/10/2018 12:00:00 AM		Kristjan Dragovan Šimik	Kristjan Dragovan Šimik	10-01-2019
12-12-2018	3. denarna donacija 65.81 € dne 12/12/2018 12:00:00 AM	65.81 €	Kristjan Dragovan Šimik	Kristjan Dragovan Šimik	10-01-2019
12-12-2018	4. nedenarna donacija dne 12/12/2018 12:00:00 AM		Kristjan Dragovan Šimik	Kristjan Dragovan Šimik	10-01-2019
12-12-2018	5. nedenarna donacija dne 12/12/2018		Kristjan Dragovan Šimik	Kristjan Dragovan Šimik	31-01-2019
22-04-2018	6. denarna donacija 87.33 € dne 4/22/2018 12:00:00 AM	87.33 €	Kristjan Dragovan Šimik	Kristjan Dragovan Šimik	10-01-2019
22-04-2018	7. denarna donacija 19.11 € dne 4/22/2018 12:00:00 AM	19.11 €	Kristjan Dragovan Šimik	Kristjan Dragovan Šimik	10-01-2019
22-04-2018	8. denarna donacija 45.73 € dne 2/22/2018 12:00:00 AM	45.73 €	Kristjan Dragovan Šimik	Kristjan Dragovan Šimik	10-01-2019
22-02-2018	9. denarna donacija 27.74 € dne 2/22/2018 12:00:00 AM	27.74 €	Kristjan Dragovan Šimik	Kristjan Dragovan Šimik	10-01-2019
22-02-2018	10. nedenarna donacija dne 2/22/2018 12:00:00 AM		Kristjan Dragovan Šimik	Kristjan Dragovan Šimik	10-01-2019
22-02-2018	11. denarna donacija 13.44 € dne 2/22/2018 12:00:00 AM	13.44 €	Kristjan Dragovan Šimik	Kristjan Dragovan Šimik	10-01-2019
22-02-2018	12. nedenarna donacija dne 2/22/2018 12:00:00 AM		Kristjan Dragovan Šimik	Kristjan Dragovan Šimik	10-01-2019
22-02-2018	13. nedenarna donacija dne 2/22/2018 12:00:00 AM		Kristjan Dragovan Šimik	Kristjan Dragovan Šimik	10-01-2019
22-02-2018	14. nedenarna donacija dne 2/22/2018 12:00:00 AM		Kristjan Dragovan Šimik	Kristjan Dragovan Šimik	10-01-2019
22-02-2018	15. denarna donacija 23.84 € dne 2/22/2018 12:00:00 AM	23.84 €	Kristjan Dragovan Šimik	Kristjan Dragovan Šimik	10-01-2019
22-02-2018	16. nedenarna donacija dne 2/22/2018 12:00:00 AM		Kristjan Dragovan Šimik	Kristjan Dragovan Šimik	10-01-2019
22-02-2018	17. denarna donacija 87.04 € dne 2/22/2018 12:00:00 AM	87.04 €	Kristjan Dragovan Šimik	Kristjan Dragovan Šimik	10-01-2019
22-02-2018	18. nedenarna donacija dne 2/22/2018 12:00:00 AM		Kristjan Dragovan Šimik	Kristjan Dragovan Šimik	10-01-2019
22-02-2018	19. nedenarna donacija dne 2/22/2018 12:00:00 AM		Kristjan Dragovan Šimik	Kristjan Dragovan Šimik	10-01-2019
22-02-2018	20. nedenarna donacija dne 2/22/2018 12:00:00 AM		Kristjan Dragovan Šimik	Kristjan Dragovan Šimik	10-01-2019
02-07-2018	21. denarna donacija 19.67 € dne 7/2/2018 12:00:00 AM	19.67 €	Kristjan Dragovan Šimik	Kristjan Dragovan Šimik	10-01-2019
02-07-2018	22. nedenarna donacija dne 7/2/2018 12:00:00 AM		Kristjan Dragovan Šimik	Kristjan Dragovan Šimik	10-01-2019
02-07-2018	23. nedenarna donacija dne 7/2/2018 12:00:00 AM		Kristjan Dragovan Šimik	Kristjan Dragovan Šimik	10-01-2019

Slika 7.16: Prikaz osnovnega okna donacij.

**Poročilo donacij** nima dodatnih parametrov.

**Poročilo delavnic**, kjer lahko podamo delavnico, za katero želimo imeti poročilo, oziroma če ne podamo delavnice, izpišemo za vse.

**Splošno poročilo oseb**, kjer lahko obkljukamo, če želimo ustvariti poročilo za celotno leto, izpisano po mesecih. V tem primeru v poljih za vnos obdobja upoštevamo le leto. V nasprotnem primeru generiramo poročilo v izbranem obdobju.

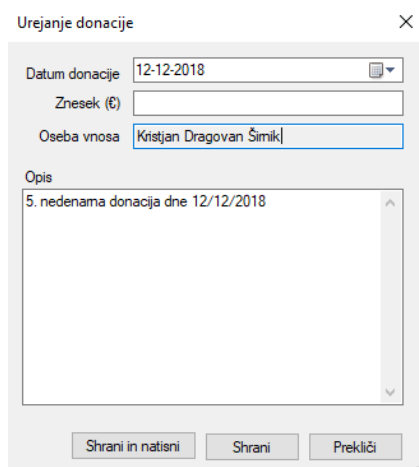
**Poročilo oseb** nima dodatnih parametrov.

**Poročilo individualnih del**, kjer lahko izberemo osebo, ki nas zanima, oziroma če ne izberemo osebe, izpišemo za vse.

Na sliki 7.19 je prikazano osnovno okno poročil. V naslednjih razdelkih pa si bomo podrobneje ogledali vsako poročilo posebej.

### 7.10.1 Poročilo donacij

Pri poročilu donacij za izbrano obdobje prikažemo le skupen znesek in število vseh donacij, ki so jih prejeli. Poročilo lahko shranimo v PDF dokument.



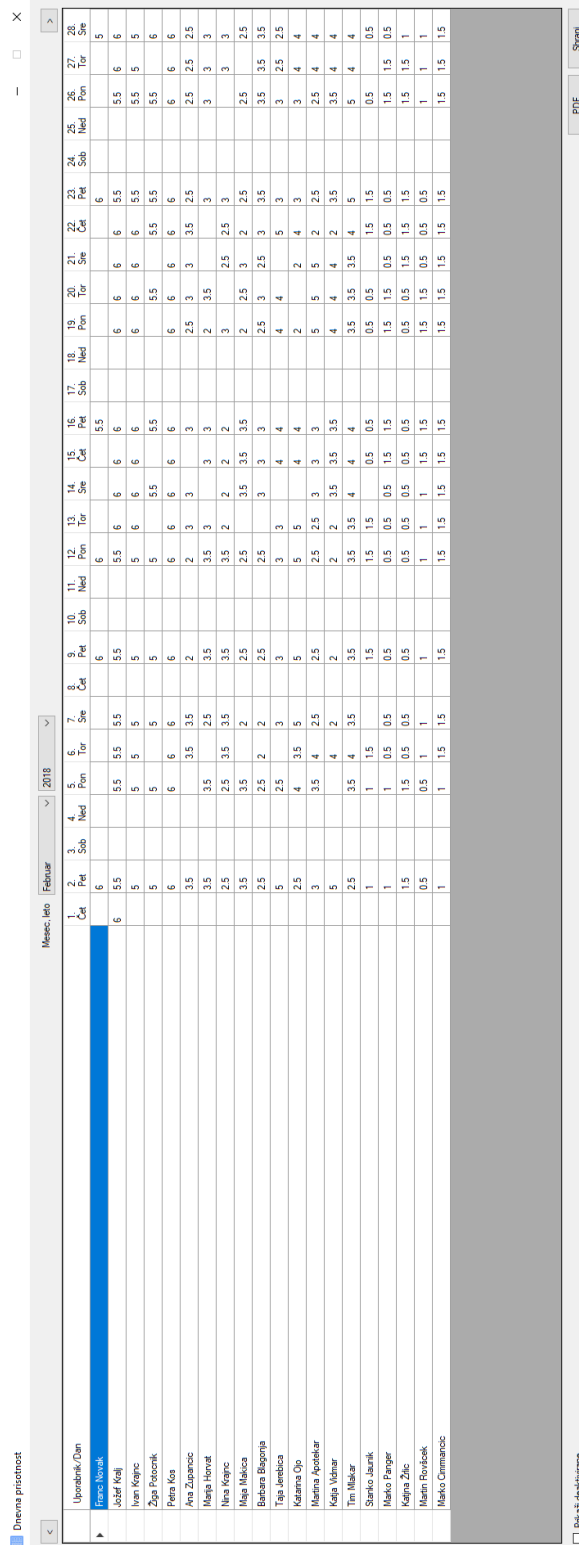
Slika 7.17: Prikaz okna za dodajanje in urejanje donacij.

Prikaz poročila je prikazan na sliki 7.20.

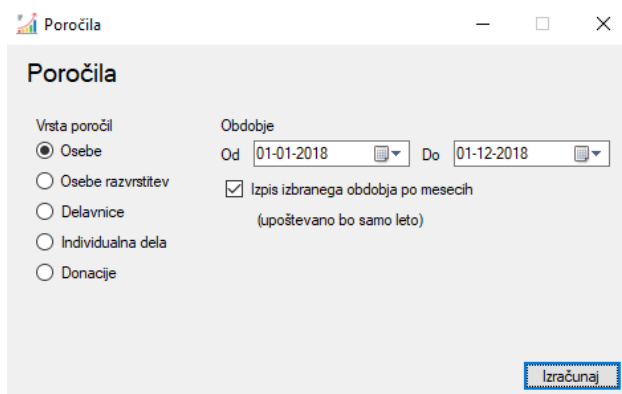
### 7.10.2 Poročilo delavnic

Pri poročilu delavnic moramo vsako izvedeno delavnico v izbranem obdobju razvrstiti v eno od časovnih kategorij na podlagi tega, kako se je izvajala. Časovne kategorije smo opisali v poglavju 1. Pri pregledovanju in razvrščanju smo upoštevali toleranco, torej v primeru, da bi se delavnica razvrstila v drugo kategorijo zaradi nekaj neizvedb, smo uporabnika vprašali, ali se delavnico vseeno razvrsti v določeno kategorijo. Ko uporabnika vprašamo, mu tudi izpišemo, kolikokrat delavnica ni bila izvedena, da se lažje odloči. Toleranca je definirana na podlagi časovne kategorije in števila pričakovanih izvedb za to kategorijo. V primeru dveh zaporednih neizvedb delavnice po definiciji časovnih kategorij se delavnica nemudoma razvrsti v drugo časovno kategorijo. Kljub toleranci pa upoštevamo tudi celodnevne delavnice, oziroma delavnice, ki so se izvajale tako, da ni bila možna izvedba drugih. V primeru izvedbe take delavnice na pregledujoč dan neizvedba pregledujoče delavnice ni bila upoštevana.

Odgovorno osebo lahko zanima razvrstitev le ene delavnice. V tem pri-



Slika 7.18: Prikaz okna dnevni prisotnosti.



Slika 7.19: Prikaz osnovnega okna poročil.

meru izbere to delavnico na prejšnjem oknu in na tem mestu se prikaže razvrstitev le izbrane delavnice.

Samo poročilo, kot ga vidite na sliki 7.21, je enostavno prebrati. Prikazana so imena delavnic ter časovna kategorija. Klik na delavnico odpre koledar izbranega obdobja z odebeljenimi datumi, kjer se je delavnica izvajala. Tako lahko odgovorna oseba preveri pravilnost razvrstitve, oziroma v primeru zanimanja zaporedja izvajanja si to lahko pogleda kar na tem mestu.

Tudi to poročilo se lahko izvozi v PDF datoteko. Poročilo se razdeli na več strani, v primeru da je veliko število delavnic.

### 7.10.3 Splošno poročilo oseb

Za vsakega udeleženca se shranjuje splošna dnevna prisotnost na društvu. To pomeni, da se šteje čas, ko je udeleženec v društvu, tudi če ni prisoten na nobeni delavnici. Iz teh podatkov pridobimo splošno poročilo oseb. V poročilu prikažemo število vseh vključenih udeležencev in povprečno število ur vključenosti enega udeleženca v obdobju. Povprečno število ur pridobimo po enostavni formuli 7.1, podani s strani ministrstva. Prav tako se prikažejo tudi imena in priimki vseh udeležencev, ki so bili prisotni v društvu v izbranem obdobju.

Donacije

Poročilo donacij za obdobje 01-01-2018 do 01-12-2018

Opis	Vrednost
Skupen znesek pridobljenih donacij v izbranem obdobju	1533.37 €
Število vseh donacij v izbranem obdobju	57

Shrani Prekliči

Slika 7.20: Prikaz okna poročil donacij.

$$\frac{\sum_{x \in \sigma_{urednevneprisotnosti}} x}{\sum_{x \in G_{count(id\_user)}(\pi_{id\_user}(dnevneprisotnosti))} x} \quad (7.1)$$

V primeru, da je odgovorna oseba obkljukala možnost izpisa poročila celega leta po mesecih, se prikaže poročilo, prikazano na sliki 7.22. Postopek, opisan zgoraj, se zažene za vsak mesec v izbranem letu posebej.

Poročilo je mogoče izvoziti v PDF dokument. Ta se ustvari tako, da prvi del zapiše na prvo stran, osebe, ki so bili prisotne, pa na drugo stran. Tako poskrbimo za preglednost.

#### 7.10.4 Poročilo oseb

Poročilo oseb je podobno poročilu delavnic. Tudi za to vrsto poročil uporabimo dnevne prisotnosti udeležencev. Za vsako časovno kategorijo je treba navesti število udeležencev, ki tej kategoriji pripada. V ta namen smo realizirali seznam za vsako časovno kategorijo in pregledovali vsakega udeleženca posebej. Tudi pri razvrščanju oseb smo upoštevali toleranco, torej v primeru, da bi se oseba razvrstila v drugo kategorijo le zaradi tega, ker je manjkala nekaj dni, program vpraša odgovorno osebo, ali kljub temu razvrsti upo-

Delavnica	Število izvedb	Udeleženi	Delavnica izvajala
Kuhanje	228	4 - 7	Dnevno
Čiščenje	232	4 - 5	Dnevno
Pospravljanje okolice	130	3 - 4	Večkrat tedensko
Okraitev centra	3	6 - 13	Občasno
Družabne igre	213	4 - 7	Občasno
Racunalništvo	262	4 - 10	Dnevno
Dopustovanje	8	20	Občasno
Trgovina	130	3 - 4	Večkrat tedensko
Izlet	10	10 - 14	Občasno
Pogovome delavnice	232	4 - 6	Dnevno
Obisk župana	1	14	Občasno
Okraitev okolice	2	7 - 8	Občasno
Tedensko čiščenje	8	4 - 7	Občasno

Slika 7.21: Prikaz okna poročil delavnic.

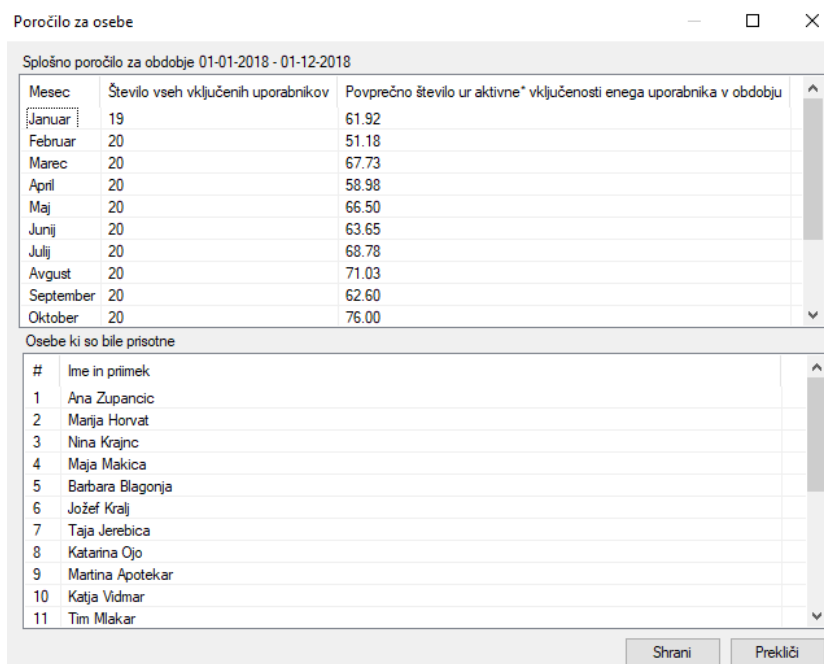
rabnika v pregledujočo časovno kategorijo. Prav tako ga izločimo iz določene časovne kategorije v primeru dveh zaporednih manjkanj po definiciji časovnih kategorij.

Prikaz poročila vidimo na sliki 7.23. Prikažejo se vse časovne kategorije in število udeležencev, ki pripadajo določeni kategoriji. Na koncu pa je izpisano tudi skupno število udeležencev, ki so bili v izbranem obdobju prisotni v društvu.

Poročilo je možno izvoziti v PDF dokument. Ta se oblikuje tako, da na eno stran v tabelo zapiše vse časovne kategorije in zraven število udeležencev, ki tej kategoriji pripada.

### 7.10.5 Poročilo individualnih del

Pri poročilu individualnih del se naredi mesečni prikaz izbranega obdobja. Prikazuje se izključno mesece, ki so v izbranem obdobju. Odgovorna oseba se premika po mesecih s pomočjo puščic. V celicah so naštete osebe, ki so imele tisti dan izvedeno individualno delo. Prav tako prikažemo skupno število oseb, ki so izvedle individualno delo v prikazanem mesecu, kot tudi v izbranem obdobju. Izpiše se tudi skupno število individualnih del v prikazanem



Poročilo za osebe

Splošno poročilo za obdobje 01-01-2018 - 01-12-2018

Mesec	Število vseh vključenih uporabnikov	Povprečno število ur aktivne* vključenosti enega uporabnika v obdobju
Januar	19	61.92
Februar	20	51.18
Marec	20	67.73
April	20	58.98
Maj	20	66.50
Junij	20	63.65
Julij	20	68.78
Avgust	20	71.03
September	20	62.60
Oktober	20	76.00

Osebe ki so bile prisotne

#	Ime in priimek
1	Ana Zupancic
2	Marja Horvat
3	Nina Krajnc
4	Maja Makica
5	Barbara Blagonja
6	Jožef Kralj
7	Taja Jerebica
8	Katarina Ojo
9	Martina Apotekar
10	Katja Vidmar
11	Tim Mlakar

Shrani Prekliči

Slika 7.22: Prikaz okna splošnega poročil oseb.

mesecu in izbranem obdobju. Izpis poročila je prikazan na sliki 7.24.

Tudo to poročilo se lahko izvozi v PDF dokument. Ta kreira dokument, kjer je vsak mesec prikazan na svoji strani, saj tako poskrbimo za večjo preglednost.

Če podamo dodatni parameter, za katerega udeleženca želi odgovorna oseba pridobiti poročilo, se poročilo izpiše na enak način, le da so tiste celice, ki prikazujejo datum, kjer je podana oseba imela izvedeno individualno delo, obarvane. Prav tako se prikaže število individualnih del v izbranem obdobju in v prikazanem mesecu, vendar tu številke veljajo le za izbranega udeleženca.



# Poglavje 8

## GDPR

### 8.1 Splošno o uredbi

GDPR določa nova pravila glede varstva osebnih podatkov. V EU se je uredba začela uporabljati 25. maja 2018. Vsaka obdelava osebnih podatkov v okviru dejavnosti sedeža upravljavca ali obdelovalca v EU bi morala biti izvedena skladno z GDPR, ne glede na to, ali sama obdelava poteka v EU ali drugje. Namen je zagotoviti varstvo podatkov in omogočiti prebivalcem EU nadzor nad njihovimi osebnimi podatki, ki se nahajajo pri obdelovalcih in upravljavcih. Uredba je neodvisna od tehnologije, torej velja za elektronsko ali ročno obdelovanje podatkov. Izpušča posameznike, ki osebne podatke obdelujejo iz osebnih razlogov ali v namen domače dejavnosti, ki ni povezana s poklicno ali komercialno dejavnostjo. Prav tako so osebni podatki umrlih in pravnih oseb izpuščeni iz uredbe. Določena so tudi ukrepanja v primeru kršitev, in sicer od opozorila, opomina in prepovedi zbiranja podatkov, do denarne kazni, ki lahko znese do 20 milijonov EUR ali 4 % letnega prometa [25, 7, 27, 19].

V nadaljevanju bomo opredelili osnovne pojme, ki jih navaja GDPR.

**Osebni podatek** je katera koli informacija o posamezniku, ki jo lahko razberemo posredno ali neposredno.

**Obdelava** pomeni vsako dejanje, ki se izvaja v zvezi z osebnimi podatki z

avtomatiziranimi sredstvi ali brez njih.

**Upravljalec** pomeni fizično ali pravno osebo, javni organ, agencijo ali drugo telo, ki samo ali skupaj z drugimi določa namene in sredstva obdelave.

**Obdelovalec** pomeni fizično ali pravno osebo, javni organ, agencijo ali drugo telo, ki obdeluje osebne podatke v imenu upravljavca.

**Uporabnik** pomeni fizično ali pravno osebo, javni organ, agencijo ali drugo telo, ki so mu bili osebni podatki razkriti, ne glede na to, ali je tretja oseba ali ne.

**Tretja oseba** pomeni fizično ali pravno osebo, javni organ, agencijo ali telo, ki ni posameznik, na katerega se nanašajo osebni podatki, upravljaavec, obdelovalec in osebe, ki so pooblašcene za obdelavo osebnih podatkov pod neposrednim vodstvom upravljavca ali obdelovalca [19].

GDPR določa, da se podatki obdelujejo na način, da iz podatkov, ki so rezultat obdelave, brez dodatnih informacij ni več možno pripisati posamezniku. Dodatne informacije pa se morajo hraniti ločeno, z nadzorom in omejitvijo dostopa. Opisano predstavlja psevdonimizacijo. Definira tudi anonimizacijo, kar je postopek, s katerim pretvorimo osebne podatke v anonimizirane, ki se jih kljub dodatnim informacijam ne da pripisati posamezniku [25].

GDPR daje posamezniku veliko pravic. Obdelovalec mora posamezniku v roku od enega do treh mesecev ugoditi na prošnjo za uveljavljanje pravic. Upravljalec mora posamezniku zagotoviti pregled obdelave njegovih osebnih podatkov ter razložiti namen uporabe. Vsak posameznik lahko zahteva pridobitev vseh podatkov, ki se navezujejo nanj. Torej moramo zagotoviti izvoz teh podatkov ob takem zahtevku, kot tudi pridobiti informacije o namenu obdelave teh podatkov, kako so se podatki pridobili in kako dolgo se bodo hranili. Posamezniku moramo podati možnost upravljanja njegovih osebnih podatkov. Olajšati mu moramo prenos, shranjevanje, kopiranje in ponovno

uporabo podatkov. Upravljalec mora osebo, ki so jim bili osebni podatki razkriti, obvestiti o popravkih. Posameznik ima pravico dopolniti in popraviti podatke, ki se nanj nanašajo. Posameznik lahko tudi zahteva, da se njegovi podatki izbrišejo iz sistema. Tu pride do izjem, v primeru da upravljalec in obdelovalec še imata upravičen razlog za obdelovanje posameznikovih podatkov, je ta zahteva zavrnjena. Posameznik lahko doseže, da upravljalec po zaključeni obdelavi, ko teh podatkov ne potrebuje več, obdelavo omeji. To pomeni, da se podatki le hranijo. Vsak uporabnik lahko ugovarja obdelavi zaradi posebnih razlogov, čeprav je obdelava zakonita [25, 7, 27].

### **8.1.1 Načela pri zbiranju in obdelavi osebnih podatkov**

#### **8.1.1.1 Zakonitost, pravičnost, preglednost**

Vsaka obdelava osebnih podatkov bi morala biti zakonita in potekati na pošten način. Za obdelavo osebnih podatkov mora obstajati ustrezna pravna podlaga, med drugim moramo pridobiti privolitev posameznika. Ta mora biti dana z jasnim pritrdilnim dejanjem [7, 27, 19].

#### **8.1.1.2 Omejitev namena zbiranja in nadaljnje obdelave**

Namen obdelave mora biti vnaprej določen in skladen z zakonodajo. Zbiranje za določen in zakonit namen, pri čemer ni dovoljena nadaljnja obdelava na način, ki ni združljiv s tem namenom. Izjema je nadaljnja obdelava v namene arhiviranja v javnem interesu, v znanstveno ali zgodovinsko-raziskovalne namene ali statistične namene. Ta bi morala veljati za združljivo zakonito obdelavo [7, 19].

#### **8.1.1.3 Najmanjši obseg podatkov**

Zbira se le tiste podatke, ki so ustrezni, relativni in potrebni za obdelovanje oziroma za izpolnitev namena. Cilj je preprečevanje zbiranja podatkov „na zalogo“. Po uredbi je treba utemeljiti razlog za potrebo po vseh osebnih podatkih, ki se jih pridobiva od posameznikov, ter kako so ti podatki

obdelani [25, 7, 19].

#### **8.1.1.4 Točnost**

Podatki morajo biti točni in posodobljeni. Vse netočne osebne podatke je treba izbrisati ali popraviti ob upoštevanju namenov, za katere se obdelujejo [7, 19].

#### **8.1.1.5 Omejitev shranjevanja**

Podatki morajo biti hranjeni v obliki, ki dopušča identifikacijo posameznikov, na katere se osebni podatki nanašajo, vendar le toliko časa, kolikor je potrebno za namene, za katere se ti obdelujejo. Tu ponovno pride do izjeme za namene arhiviranja, kot smo to že opisali v razdelku omejitve namena zbiranja in nadaljnje obdelave [7, 19].

#### **8.1.1.6 Celovitost in zaupnost**

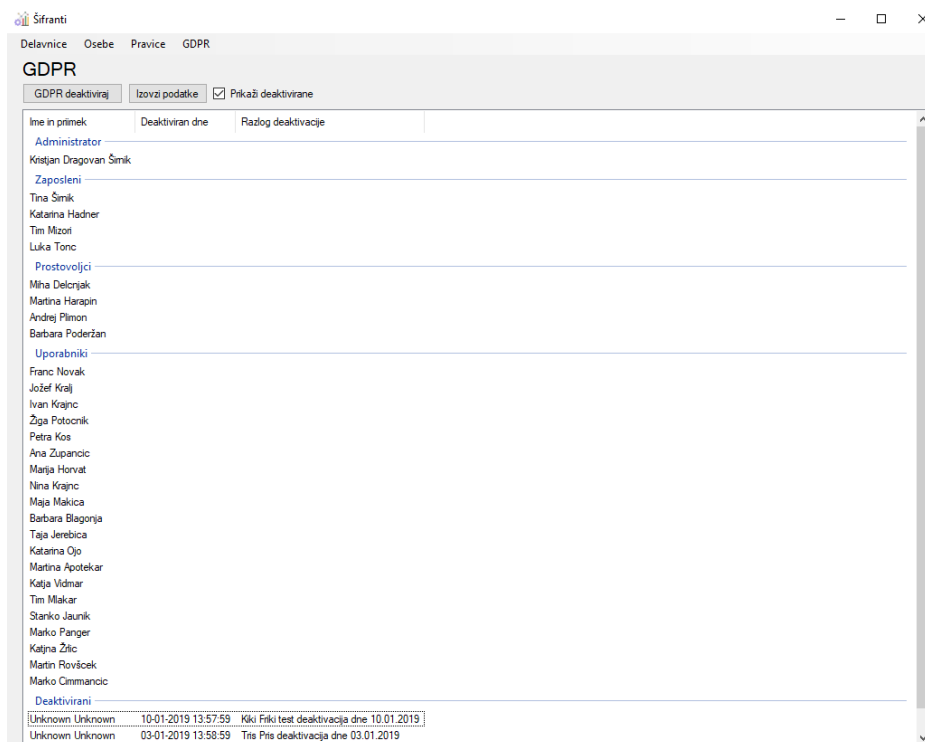
Podatki se obdelujejo na način, ki zagotavlja ustrežno varnost osebnih podatkov. Zagotovitev varnosti se doseže s tehničnimi in organizacijskimi ukrepi. Podatke je treba zaščititi pred nedovoljeno ali nezakonito obdelavo, kot tudi pred nenamerno izgubo ali uničenjem [25, 19].

#### **8.1.1.7 Odgovornost**

Upravljalca je odgovoren za skladnost z vsemi do sedaj navedenimi načeli. Zahtevano je preventivno in proaktivno ravnanje [7, 19].

## **8.2 Rešitev v programu**

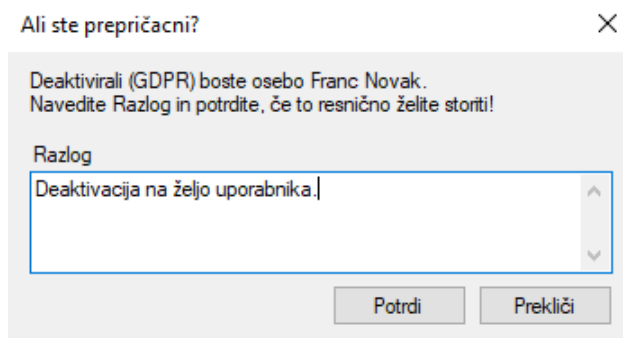
Pod modulom šifrantov najdemo razdelek GDPR. Tam se odgovornim osebam prikaže okno, kot ga vidimo na sliki 8.1. Naštete so vse osebe, ki so vnešene v sistem. Posamezno osebo lahko deaktiviramo ali izvozimo vse njegove podatke.



Slika 8.1: Prikaz GDPR okna.

### 8.2.1 Deaktivacija osebe

Ob deaktivaciji je obvezno vnesti razlog v potrditveno okno, prikazano na sliki 8.2. Pri tem se vsi osebni podatki v PB za izbrano osebo izbrišejo, oziroma polja, ki so obvezna, se nastavijo na neznane vrednosti. Dnevni prisotnosti, prisotnosti na delavnicah, individualnih delih, ciljev in njihovih evalvacij ter vnešenih donacij izbrane osebe pa ne smemo izbrisati, saj potem poročila ne bi bila točna. Deaktivirane osebe ne prikazujemo več na formah, v primeru urejanja že obstoječe izvedbe delavnice oziroma individualnega dela pa se na dnu izpiše še število oseb, ki so bili prisotni, a so deaktivirani. Osebo se kasneje ne da ponovno aktivirati.



Slika 8.2: Prikaz okna pri deaktivaciji osebe.

## 8.2.2 Izvoz podatkov o osebi

Vse podatke osebe lahko tudi izvozimo. To lahko stori odgovorna oseba za vse osebe, ki niso bile deaktivirane. Če je oseba že podala željo deaktivacije in odstranitve osebnih podatkov iz PB, izvoz podatkov ni več mogoč. Pri izvozu se ustvari datoteka v JSON formatu, ki vsebuje:

- osebne podatke osebe:
  - identifikator osebe,
  - ime,
  - priimek,
  - datum rojstva,
  - tip osebe,
  - email,
- izvedbe delavnic, na katerih je bila oseba prisotna:
  - identifikator izvedbe,
  - datum izvedbe,
  - opombe,
  - podatki o delavnici:

- \* šifra delavnice,
- \* ime delavnice,
- individualna dela osebe:
  - identifikator individualnega dela,
  - datum individualna dela,
  - zaznamek,
- dnevne prisotnosti osebe:
  - identifikator dnevne prisotnosti,
  - datum prisotnosti,
  - ure,
- cilji osebe:
  - identifikator cilja,
  - datum ustvaritve,
  - cilj,
  - datum zaključka cilja, v primeru, da je zaključen,
  - evalvacije:
    - \* identifikator evalvacije,
    - \* evalvacija,
    - \* datum evalvacije.

### 8.2.3 Spremljanje poteka programa

Vsak dostop v program je nadzorovan in zabeležen. V PB smo s tem namenom ustvarili posebno tabelo, imenovano „Log“, ki ni povezana z nobeno drugo tabelo. Do podatkov v tabeli lahko dostopa le skrbnik PB. V tabeli hranimo datum in čas dostopa, identifikator osebe, ki je dostopala, modul,

do katerega je dostopala, kratek opis in opis napake, v primeru, da je do te prišlo. Od osebe hranimo le identifikator, zato tabela ni povezana s tabelo oseb, saj bi drugače morali ob brisanju izbrisati tudi vse zapise v „Log“ tabeli, kar pokvari namen omenjene tabele. V primeru izbrisa osebe bomo še vedno imeli zapisano, da je neka oseba dostopala do določenih modulov, oziroma spreminjala in dodajala podatke v PB ter brisala le-te. Tabela tudi nam - razvijalcem programa - olajša iskanje in razreševanje napak v primeru, da bi do teh prišlo.

## Poglavje 9

# Sklepne ugotovitve

S programom, ki smo ga razvili, društvo lahko elektronsko evidentira svoje delo in pripravlja zahtevana poročila. Delo društva v zvezi z navedenim je tako vsekakor enostavneje, hitreje in preglednejše, kot tudi bolj ekonomično, kar je bil tudi namen tega programa. Pri uporabi programa so pričakovani manjši zapleti zaradi pomanjkljivosti pri evidentiranju dela s strani zaposlenih (človeški faktor), ki jih bo na začetku uporabe programa več. Toleranca, ki je upoštevana pri generiranju poročil, je vnaprej določena po prošnjah s strani zaposlenih. Po testiranju programa bi lahko po želji naročnika ustvarili dodaten modul, kjer bi toleranco za posamezno časovno kategorijo in modul definirali in spreminjali sami. Kot smo že omenili v poglavju 4, bi lahko glede na potrebe in način uporabe dodali prožilce, ki bi se izvedli ob vsakem spreminjanju tabele. Ti bi ob pravilni uporabi koristili hitrejši razvrstitvi delavnic in oseb v časovne kategorije. V izogib počasnosti delovanja programa zaradi velikega obsega podatkov bi lahko dodali strani s številčno omejitvijo izpisa podatkov in možnost listanja med različnimi stranmi.

Ustvarjeni program uspešno odgovarja na potrebe društva pri evidentiranju dela in pripravi poročil, kar je bil tudi njegov prvotni namen.



# Literatura

- [1] Programski jezik C sharp. Dosegljivo: [https://sl.wikipedia.org/wiki/Programski\\_jezik\\_C\\_sharp](https://sl.wikipedia.org/wiki/Programski_jezik_C_sharp), 2017. [Dostopano: 24. 1. 2019].
- [2] C Sharp (programming language). Dosegljivo: [https://en.wikipedia.org/wiki/C\\_Sharp\\_\(programming\\_language\)#Common\\_type\\_system](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#Common_type_system), 2019. [Dostopano: 24. 1. 2019].
- [3] CREATE TRIGGER (Transact-SQL). Dosegljivo: <https://docs.microsoft.com/en-us/sql/t-sql/statements/create-trigger-transact-sql?view=sql-server-2017>. [Dostopano: 13. 2. 2019].
- [4] Database trigger. Dosegljivo: [https://en.wikipedia.org/wiki/Database\\_trigger](https://en.wikipedia.org/wiki/Database_trigger), 2018. [Dostopano: 13. 2. 2019].
- [5] draw.io. Dosegljivo: <https://about.draw.io/>. [Dostopano: 25. 1. 2019].
- [6] Indeksi. Dosegljivo: [http://colos.fri.uni-lj.si/ERI/RACUNALNISTVO/PODATKOVNE\\_BAZE/indeksi.html](http://colos.fri.uni-lj.si/ERI/RACUNALNISTVO/PODATKOVNE_BAZE/indeksi.html). [Dostopano: 7. 2. 2019].
- [7] Informacijski pooblaščenec. Dosegljivo: <https://www.ip-rs.si/>. [Dostopano: 15. 2. 2019].
- [8] Introducing JSON. Dosegljivo: <https://www.json.org/>. [Dostopano: 7. 2. 2019].

- [9] JSON. Dosegljivo: <https://en.wikipedia.org/wiki/JSON/>, 2019. [Dostopano: 7. 2. 2019].
- [10] Language Integrated Query. Dosegljivo: [https://en.wikipedia.org/wiki/Language\\_Integrated\\_Query](https://en.wikipedia.org/wiki/Language_Integrated_Query), 2018. [Dostopano: 25. 1. 2019].
- [11] Language Integrated Query (LINQ). Dosegljivo: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/>. [Dostopano: 25. 1. 2019].
- [12] Microsoft Azure. Dosegljivo: [https://en.wikipedia.org/wiki/Microsoft\\_Azure](https://en.wikipedia.org/wiki/Microsoft_Azure), 2019. [Dostopano: 14. 2. 2019].
- [13] Microsoft .NET. Dosegljivo: [https://sl.wikipedia.org/wiki/Microsoft\\_.NET](https://sl.wikipedia.org/wiki/Microsoft_.NET), 2017. [Dostopano: 24. 1. 2019].
- [14] Microsoft SQL Server. Dosegljivo: [https://en.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](https://en.wikipedia.org/wiki/Microsoft_SQL_Server), 2019. [Dostopano: 24. 1. 2019].
- [15] Ministrstvo za delo, družino, socialne zadeve in enake možnosti. Dosegljivo: <http://www.mddsz.gov.si/si/>. [Dostopano: 25. 1. 2019].
- [16] MySQL Workbench. Dosegljivo: [https://en.wikipedia.org/wiki/MySQL\\_Workbench](https://en.wikipedia.org/wiki/MySQL_Workbench), 2018. [Dostopano: 25. 1. 2019].
- [17] .NET Framework. Dosegljivo: [https://en.wikipedia.org/wiki/.NET\\_Framework#Architecture](https://en.wikipedia.org/wiki/.NET_Framework#Architecture), 2019. [Dostopano: 24. 1. 2019].
- [18] Poenoteni jezik modeliranja. Dosegljivo: [https://sl.wikipedia.org/wiki/Poenoteni\\_jezik\\_modeliranja](https://sl.wikipedia.org/wiki/Poenoteni_jezik_modeliranja), 2018. [Dostopano: 28. 2. 2019].
- [19] Lea Polak. Varstvo osebnih podatkov, izobraževanje za zaposlene, 2017. [Dostopano: 15. 2. 2019].
- [20] PowerDesigner. Dosegljivo: <https://en.wikipedia.org/wiki/PowerDesigner>, 2018. [Dostopano: 25. 1. 2019].

- 
- [21] Refleksija (računalništvo). Dosegljivo: <https://sl.wikipedia.org/wiki/Refleksija>, 2016. [Dostopano: 24. 1. 2019].
- [22] Stored procedure. Dosegljivo: [https://en.wikipedia.org/wiki/Stored\\_procedure](https://en.wikipedia.org/wiki/Stored_procedure), 2018. [Dostopano: 7. 2. 2019].
- [23] Table (database). Dosegljivo: [https://en.wikipedia.org/wiki/Table\\_\(database\)](https://en.wikipedia.org/wiki/Table_(database)), 2019. [Dostopano: 7. 2. 2019].
- [24] Transact-SQL. Dosegljivo: <https://en.wikipedia.org/wiki/Transact-SQL>, 2018. [Dostopano: 24. 1. 2019].
- [25] Maja Umek. Splošna uredba o varstvu podatkov (GDPR) in podatkovno intenzivne mobilne aplikacije. Diplomsko delo, Fakulteta za računalništvo in informatiko, Univerza v Ljubljani, 2018.
- [26] Use the Azure storage emulator for development and testing. Dosegljivo: <https://docs.microsoft.com/en-us/azure/storage/common/storage-use-emulator>. [Dostopano: 14. 2. 2019].
- [27] Vse o uredbi GDPR. Dosegljivo: <https://www.monitor.si/clanek/vse-o-uredbi-gdpr/182630/>, 2017. [Dostopano: 15. 2. 2019].
- [28] Maja Zafran. Prototip mobilne aplikacije mStudis. Diplomsko delo, Fakulteta za računalništvo in informatiko, Univerza v Ljubljani, 2018.