

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Borut Furlan

**Spletna aplikacija za vzdrževanje
distribucijskega sistema**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM
PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Luka Čehovin Zajc

SOMENTOR: mag. Marko Zupančič

Ljubljana, 2018

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^AT_EX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V okviru diplomske naloge izdelajte novi spletni uporabniški vmesnik za vzdrževanje distribucijskega sistema, ki bo temeljil na obstoječi spletni aplikaciji, ki ste jo spoznali med prakso. Prenova aplikacije naj temelji na sodobnih spletnih tehnologijah, izboljšavi uporabniške izkušnje z mislijo omogočiti odzivni uporabniški vmesnik s podporo novejših brskalnikov in možnostjo dela na mobilnih napravah. Nova aplikacija naj ohrani funkcionalne uporabniške zahteve starejše aplikacije, hkrati naj z novo tehnologijo izboljša obstoječe in doda nove funkcionalnosti, obenem pa naj stremi k dodatni podpori brezpapirnega delovanja vzdrževalnega procesa.

Zahvaljujem se vsem, ki so po svojih najboljših močeh prispevali k izdelavi diplomske naloge in me pri tem spodbujali. Posebno pa se zahvaljujem mentorju doc. dr. Luka Čehovin Zajc, ki me je po najboljših močeh usmerjal pri izdelovanju diplomske naloge. Zahvaljujem se tudi vsem v podjetju Mikrografija d.o.o., ki so mi stali ob strani in me usmerjali.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Opis problema	3
2.1	Potek dela pri naročniku pred uvedbo nove aplikacije	3
2.2	Opis delovnega procesa pri naročniku	4
2.3	Vse funkcionalnosti spletne aplikacije	6
3	Zasova spletne aplikacije	11
3.1	Uporabljene tehnologije, orodja in predloga pri razvoju spletne aplikacije	11
3.2	Podatkovni model	19
4	Razvoj spletne aplikacije	23
4.1	Opis razvoja spletne aplikacije	23
4.2	Opis delovanja spletne aplikacije	36
5	Testiranje in evaluacija	47
6	Sklepne ugotovitve	53
	Literatura	57

Seznam uporabljenih kratic

kratica	angleško	slovensko
MVC	model-view-controlle	model-pogled-krmilnik
AJAX	asynchronous JavaScript and XML	asinhroni JavaScript in XML
DAO	Database Access Object	objekti za dostopanje do podatkovne baze
HTML	Hyper Text Markup Language	jezik za označevanje nadbese-dila
CSS	Cascading Style Sheets	kaskadne stilske predloge
SQL	Structured Query Language	strukturni poizvedovalni jezik
IDE	Integrated development environment	integrirano razvojno okolje
UI	User Interface	uporabniški vmesnik

Povzetek

Naslov: Spletna aplikacija za vzdrževanje distribucijskega sistema

Avtor: Borut Furlan

Diplomsko delo opisuje razvijanje spletne aplikacije za vzdrževanje distribucijskega sistema, ki se je ponovno razvila s pomočjo novejših tehnologij in prenosom funkcionalnosti stare spletne aplikacije na podlagi obstoječega podatkovnega modela. Stara spletna aplikacija je bila razvita s pomočjo tehnologij, ki so zdaj že zastarele, zaradi nevezdrževanja so se pojavile pomankljivosti v funkcionalnostih, nekatere so celo prenehale delovati, kar je delo v naročnikovem podjetju le oteževalo. Z uporabo stare spletne aplikacije je delo potekalo tudi ročno z izpolnjevanjem obrazcev v papirnati obliki. V na novo razviti spletni aplikaciji smo na podlagi delovnega procesa dodali tudi funkcionalnosti, s katerimi smo izpodrinili ročno delo in s tem olajšali delo zaposlenih. Opisane so vse tehnologije in orodja, uporabljena v postopku razvoja. Prav tako je opisan podatkovni model, na podlagi katerega smo razvijali spletno aplikacijo in vse vsebovane funkcionalnosti. Sledi opis delovanja spletne aplikacije, na podlagi katerega smo izvedli testiranje in evalvacijo testiranja uporabniških vmesnikov stare in nove spletne aplikacije, ter sklepne ugotovitve, do katerih smo prišli med razvojem.

Ključne besede: yii2, spletna aplikacija, uporabniški vmesnik, Spletna aplikacija za vzdrževanje distribucijskega sistema.

Abstract

Title: Web application for maintenance of a distribution system

Author: Borut Furlan

The graduation thesis describes the development of a web application for the maintenance of the distribution system, which was re-developed with the help of newer technologies and the transfer of the functionality of the old web application based on the existing data model. The old web application was developed with the help of technologies which are already outdated, due to lack of maintenance, there were some weaknesses in the functionalities, some even stopped working, which made the parts in the subscriber's company more difficult. Using the old web application, the work was also carried out manually by filling out forms in paper form. In the newly developed web application, we also added functionality based on the work process, with this functionalitis we displaced manual work and thus facilitated the work of employees. All the technologies and tools that have been used during the development are described. We also describe the data model on the basis of which we developed the web application and all the functionality contained. The following is a description of the operation of the web application, based on which we tested and evaluated the testing of user interfaces of the old and new web applications, and the conclusions reached during the development process.

Keywords: yii2, web application, user interface, Web application for maintenance of a distribution system.

Poglavje 1

Uvod

Ideja diplomske naloge je nastala v sodelovanju zunanjega naročnika spletne aplikacije in podjetja Mikrografija d.o.o. Podjetje Mikrografija je zelo prilagodljivo in se ukvarja z skeniranjem, obdelovanjem, hrambo in arhiviranjem papirnatih dokumentacij vseh oblik in formatov, v sodelovanju z domačimi in tujimi podjetji. Poleg glavne dejavnosti delujejo tudi na področjih spremljevalnih dejavnosti, kot so urejanje arhivov, masovno skeniranje dokumentacij različnih vrst, zajem oziroma prepoznavanje vsebin dokumentacije, shranjevanje dokumentacij v elektronski obliki, uničenje dokumentacijskega gradiva, odbiranje itd. Spremljevalne dejavnosti spremljajo spremljevalne storitve, razvite v podjetju Mikrografija d.o.o., kot so mDocs, mScan, mSef, mZajem itd.

V sodelovanju podjetja Mikrografija d.o.o. in podjetja naročnika je prišlo do ideje, da se na podlagi stare spletne aplikacije z uporabno novih tehnologij in orodij ponovno razvije spletna aplikacija, kajti stara je zaradi zastarelih tehnologij in pomankanja vzdrževanja funkcionalnosti zastarala, nekatere funkcionalnosti so prenehale delovati, spet druge niso delovale pravilno.

V diplomskem delu si bomo pogledali razvoj spletne aplikacije za vzdrževanje distribucijskega sistema, ki je potekal v podjetju Mikrografija za zunanjega naročnika. Spletna aplikacija, ki smo jo razvili na novo, vsebuje vse funkcionalnosti stare naročnikove aplikacije, dodali smo še nekaj dodatnih funk-

cionalnosti, ki so jih predhodno uporabljali s pomočjo drugih aplikacij, in omogočili, da se prejšnje ročno delo zdaj opravlja s pomočjo spletne aplikacije in se tako olajša delo zaposlenih.

Cilj na novo razvite spletne aplikacije je razviti spletno aplikacijo z novimi tehnologijami z bolj intuitivnim uporabniškim vmesnikom in možnostno prilagoditve funkcionalnosti potrebam drugih naročnikov.

Poglavje 2

Opis problema

Preden pričnemo z razvojem spletne aplikacije, je treba dobro poznati potek dela pri naročniku aplikacije. Glede na to, da se je aplikacija razvila na podlagi starega podatkovnega modela, se delovni proces ni veliko spremenil, v grobem je celo ostal enak.

V naslednjem poglavju je opisan potek dela in delovni proces pri naročniku pred uvedbo nove spletne aplikacije. Delovni proces pri naročniku spletne aplikacije je ostal enak. Opisane so tudi vse potrebne funkcionalnosti, ki morajo biti vsebovane v novi spletni aplikaciji naročnika.

2.1 Potek dela pri naročniku pred uvedbo nove aplikacije

Glavna naloga naročnika so vzdrževalna dela na njihovih objektih, opremi in povezavah med objekti, katere dodeljujejo, opravljajo, beležijo in zaključujejo s pomočjo delovnih nalogov in opravil, ki jih vodijo s pomočjo stare spletne aplikacije za vzdrževanje distribucijskega sistema.

Naročnikova spletna aplikacija je namenjena obvladovanju delovnih nalogov in pripadajočih opravil, kar se je do zdaj izvajalo s pomočjo stare spletne aplikacije naročnika, ki je zastarela in ni bila vzdrževana, zato so nekatere funkcionalnosti prenehale delovati ali niso delovale pravilno. Spletna aplika-

cija naročnika se je v veliki meri uporabljala za vodenje evidence delovnih nalogov in opravil s pomočjo vpisovanja v podatkovni model in preglede nekaterih poročil v obliki statistike.

Dela so se pri naročniku izvajala počasi, saj so zahtevala veliko nepotrebnega dodatnega dela. Terensko delo poleg vzdrževalnih del vsebuje tudi popise vzdrževalnih del na vzdrževanih objektih in povezavah med objekti; popise so izvajali ročno s pomočjo natisnjenih obrazcev. Popisovanje na takšen način je počasno in neučinkovito, obrazce so hranili v papirnati obliki kot fizično dokumentacijo. Slabost takšne hrambe je, da se lahko kakšen obrazec založi ali celo izgubi. V primeru izgubljenih obrazcev opravljeno delo nima pomena, saj ga je treba še enkrat opraviti.

2.2 Opis delovnega procesa pri naročniku

Poenostavljen diagram poteka delovnega procesa pri naročniku spletne aplikacije predstavlja Slika 2.1

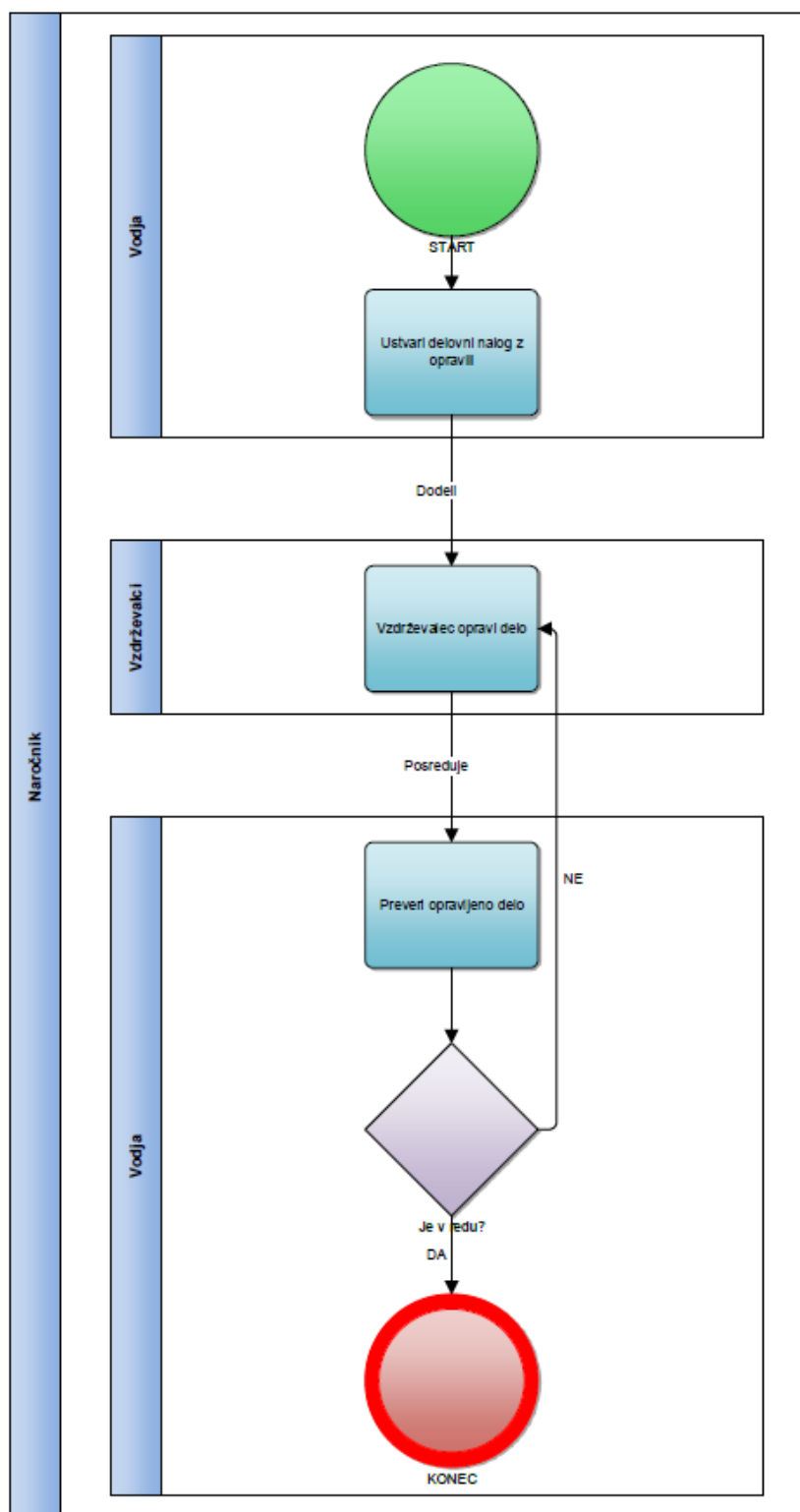
Delovni proces pri naročniku spletne aplikacije za urejanje distribucijskega omrežja se deli na več nalog, ki jih izvajajo zaposleni, ki jim pripada določena vloga.

Vloge zaposlenih na podlagi stare spletne aplikacije so:

- **Vodja** – Dodeljuje in preverja vzdrževalna dela.
- **Vzdrževalec** – Od vodje prejme vzdrževalno delo, ga opravi in posreduje nazaj vodji.

Da se delovni proces začne, mora Vodja sestaviti delovne naloge, nanje doda vsa potrebna opravila in jim nato dodeli Vzdrževalce distribucijskega sistema.

Vzdrževalci od Vodje prejmejo delovne naloge z opravili, ki jih morajo opraviti na določenih objektih ali povezavah med objekti. Ko vzdrževalec izpolni vsa opravila na delovnem nalogu, to posreduje Vodji.



Slika 2.1: Diagram poteka naročnikovega delovnega procesa.

Vodja na koncu preveri, ali so opravila na delovnem nalogu opravljena ustrezno. Če so, zaključi vsa opravila delovnega naloga in sam delovni nalog. V nasprotnem primeru, če torej nekatera opravila niso ustrezno opravljena, teh opravil ne zaključi in delovni nalog posreduje nazaj Vzdrževalcu. Ko so vsa opravila delovnega naloga in delovni nalog zaključeni, se konča tudi delovni proces.

2.3 Vse funkcionalnosti spletne aplikacije

Funkcionalnosti, ki morajo biti vsebovane v naročnikovi spletni aplikaciji za vzdrževanje distribucijskega sistema, smo razdelili v tri skupine.

V prvo skupino sodijo funkcionalnosti, ki so bile razvite v predhodni spletni aplikaciji pri naročniku in morajo po enakem principu delovati tudi v novi spletni aplikaciji. Nekatere funkcionalnosti so v prejšnji spletni aplikaciji delovale nepravilno ali so imele pomankljivosti. Te funkcionalnosti smo v novi spletni aplikaciji popravili in dopolnili.

Funkcionalnosti druge skupine v predhodni spletni aplikaciji niso bile razvite, v nadaljevanju poglavja si jih bomo podrobneje ogledali.

Imamo še tretjo skupino funkcionalnosti, v katero sodijo na novo razvite funkcionalnosti, katerih do zdaj niso poznali v drugačni obliki, kot je ročno delo. Ročno delo je delo na papir, to so razni popisi objektov in opreme, in se je izvajalo s pomočjo različnih obrazcev, ki so bili odvisni od vrste vzdrževalnega dela na posameznih opravilih. Izpolnjene obrazce so fizično hranili kot fizično dokumentacijo. V spletni aplikaciji naročnika smo obrazce razvili tako, da jih lahko zdaj izpolnjujejo na terenu s pomočjo spletne aplikacije, do katere dostopajo s pomočjo tabličnega računalnika ali pametnega telefona, in razširili podatkovni model tako, da omogoča hranjenje vseh dodatnih obrazcev.

2.3.1 Prva skupina funkcionalnosti

Pregled funkcionalnosti prve skupine

- ***Delovni nalogi***

- Ustvarjanje delovnega naloga
- Urejanje delovnega naloga
- Pregled delovnega naloga
- Brisanje delovnega naloga
- Koledarski pregled delovnih nalogov
- Premik delovnega naloga/ov na drug datum
- Kopiranje enega ali več delovnih nalogov

- ***Opravila***

- Ustvarjanje opravila
- Urejanje opravila
- Pregled opravila
- Brisanje opravila
- Koledarski pregled opravil
- Premik enega ali več opravil na drug delovni nalog
- Spreminjanje statusa opravil
- Spreminjanje periodike opravil
- Dodajanje opreme na opravilo
- Pregled opreme na opravilu

- ***Poročila***

- ***Šifranti***

Predhodno naštete funkcionalnosti so bile vsebovane že v stari spletni aplikaciji. Delovne naloge lahko ustvarjamo, urejamo, pregledujemo ali izbrisemo; če so na delovnem nalogu opravila, se ravno tako izbrišejo s predhodnim opozorilom. Koledarski pregled delovnih nalogov omogoča lažje iskanje

delovnih nalogov in prikazuje število delovnih nalogov po posameznih datumih. Delovne naloge z vsemi opravili lahko tudi premaknemo na drug datum. Lahko pa jih kopiramo in ob kopiranju spremenimo nastavitve opravil. Kopiramo lahko enega ali več delovnih nalogov hkrati, ob tem se kopirajo tudi vsa pripadajoča opravila posameznega delovnega naloga.

Opravila lahko ravno tako kot delovne naloge ustvarjamo na novo ali dodamo že ustvarjeno planirano opravilo, lahko jih urejamo, brišemo in pregledujemo. Prav tako kot pri delovnih nalogih imamo koledarski pregled, ki deluje po enakem principu kot koledarski pregled delovnega naloga, le da ne prikaže števila opravil na posameznem datumu. Opravila lahko premaknemo z enega delovnega naloga na drugega, lahko premaknemo eno ali več opravil hkrati. Spreminjamo lahko status opravila glede na to, ali je opravilo že zaključeno ali je v katerem drugem stanju, spreminjamo lahko nastavitve periodičnega ponavljanja delovnih nalogov in opravil na enem ali več hkrati. Na posamezna opravila lahko dodajamo opremo iz šifranta oprema, ki jo prav tako lahko urejamo, pregledujemo in brišemo.

Poročila prikazujejo statistike vzdrževalnih del, zaključenih del, opravljenih opravil itd. Nekatera poročila v stari spletni aplikaciji so zaradi zastarelosti prenehala delovati.

Šifrantov je veliko različnih in so namenjeni zgolj vpisovanju podatkov, ki se uporabljajo na delovnih nalogih in opravilih, kjer lahko posamezne šifrante pregledujemo, dodajamo, urejamo ali brišemo.

2.3.2 Druga skupina funkcionalnosti

V drugo skupino funkcionalnosti nove spletne aplikacije smo uvrstili funkcionalnosti, ki so bile dodatno razvite in jih predhodno pri naročniku niso uporabljali. Dodatno smo razvili dodajanje slik pri šifrantu opreme in objektov. Dodajamo lahko že shranjene slike, lahko pa jih dodamo s pomočjo mobilnega telefona ali tablice tako, da fotografijo direktno posnamemo s pomočjo aplikacije za zajem fotografij, ki je nameščena na napravah.

Dodana je bila možnost tiskanja enega ali več delovnih nalogov hkrati.

Tiskanje delovnih nalogov poteka na podlagi predhodno zgrajene predloge za tiskanje delovnih nalogov. Predloga za tisk delovnega naloga je zgrajena s pomočjo programa Microsoft Word. Delovni nalog natisnemo po naslednjih korakih: najprej naložimo predlogo za tisk delovnih nalogov s pomočjo na novo razvite funkcionalnosti. Nato izberemo delovne naloge, ki jih želimo natisniti, in izberemo želeno predlogo za tiskanje.

2.3.3 Tretja skupina funkcionalnosti

V to skupino sodijo funkcionalnosti, ki smo jih smo razvili tako, da nadomestijo ročno delo na papir. Relacijski podatkovni model smo razširili z dodatnimi tabelami in povezavami, tako da omogoča shranjevanje zapisov obrazcev. Za vsak posamezen obrazec je kreirana tabela z vsemi potrebnimi povezavami v relacijskem podatkovnem modelu. V spletni aplikaciji naročnika smo razvili obrazce, ki so na voljo na opravilih v primeru, da je izbrano ustrezno vzdrževalno delo. Če je na opravilu izbrano vzdrževalno delo, kateremu pripada določen obrazec, se odpre možnost za dodajanje tega obrazca, ki ga lahko kasneje tudi pregledujemo, urejamo ali izbrišemo. Nekaterim vzdrževalnim delom pripada več obrazcev hkrati.

Poglavje 3

Zasova spletne aplikacije

Vsaka spletna aplikacija za svoj razvoj potrebuje tehnologije in orodja. Uporabljene tehnologije in orodja novo razvite spletne aplikacije smo opisali v nadaljevanju poglavja. Predstavili bomo tudi podatkovni model in njegove razširitve, ki smo jih ustvarjali ob razvoju spletne aplikacije in si s tem olajšali delo ali omogočili delo na dodatnih funkcionalnostih. Predstavljena je tudi predloga, na podlagi katere se je razvijala spletna aplikacija za vzdrževanje distribucijskega sistema.

3.1 Uporabljene tehnologije, orodja in predloga pri razvoju spletne aplikacije

Brez tehnologij in orodij je delo na razvoju spletne aplikacije oteženo in zamudno, zato smo za razvoj spletne aplikacije določili naslednje tehnologije in orodja, s katerimi so se zaposleni v podjetju že srečali in s pomočjo katerih bomo razvili funkcionalnosti iz poglavja 2.3. Delo na videzu uporabniškega vmesnika smo poenostavili z uporabo predhodno zgrajene predloge.

3.1.1 Tehnologije

Sledijo opisi tehnologij, uporabljenih pri razvoju spletne aplikacije za vzdrževanje distribucijskega sistema.

Yii2 Ogrodje

Yii 2 ogrodje [14] ali na kratko Yii smo izbrali za razvoj aplikacije za vzdrževanje distribucijskega sistema, ker se je veliko razvijalcev v podjetju s tem orodjem že srečalo in ga dovolj dobro poznajo. Yii je moderno visoko zmogljivo odprtokodno ogrodje PHP za razvijanje spletnih aplikacij in aplikacijskih programskih vmesnikov API (ang. Application Programming Interface). Ogrodje omogoča hiter razvoj kompleksnih aplikacij po vzorcu DRY (ang. Don't Repeat Yourself), saj omogoča učinkovit, razširljiv in trajen razvoj spletnih aplikacij. Ogrodje Yii 2 je zelo hitro ogrodje, saj je svoj sloves doseglo z optimizacijo za hitrost in je namenjeno poslovnim aplikacijam, prav tako pa tudi vsem drugim tipom aplikacij.

Ogrodje Yii 2 vsebuje tudi svoja orodja, s pomočjo katerih omogoča hiter razvoj vseh tipov aplikacij. Med ta orodja spadajo princip delovanja MVC (ang. Model-View-Controller), lažje dostopanje do podatkovne baze s pomočjo DAO (ang. Data Access Object), omogoča validacijo vnosnih polj, prav tako lahko uporabljamo asinhroni JavaScript in XML AJAX (ang. Asynchronous JavaScript and XMLHttpRequest), omogoča tudi poročanje napak s pomočjo brskalnika, avtentikacijo, uporabljamo lahko različne teme, spletne storitve, varnost, avtomatsko generiranje kode za hiter razvoj in še veliko več zelo uporabnih orodij. Zelo velika prednost orodja je, da ima podarek na počasnem nalaganju (ang. Lazy Loading), zaradi tega tudi slovi po optimizaciji na področju hitrosti. Primer počasnega nalaganja je, ko se objekt kreira šele ob klicu, dostop do podatkovne baze se izvede šele, ko je to potrebno.

PHP

Hypertext Preprocessor ali kratko PHP [10] je skriptni jezik, ki se uporablja v spletnem razvoju, lahko pa se uporablja kot splošni programski jezik. Koda se, preden prispe do odjemalca, preprocesira na strežniku, ki jo obdela PHP tolmač (ang. PHP interpreter), ki se izvaja kot modul na strežniku. Tako se na strežniku zadržujejo rezultati interpretirane in izvedene PHP kode, ki je sestavljena iz različnih vrst podatkov in se servira na spletu. Za razvoj spletne aplikacije za vzdrževanje distribucijskega sistema smo uporabili PHP verzije 7.2, ki se od prejšnjih verzij predvsem razlikuje v hitrosti procesiranja PHP kode s pomočjo PHP tolmača.

Strežnik Microsoft SQL

Strežnik Microsoft SQL ali na kratko MS SQL [9] je sistem za upravljanje relacijske podatkovne baze, ki teče na strežniku in podpira različne aplikacije za obdelovanje transakcij in analitike podjetniških okolij IT. Spada med tri vodilne tehnologije za podatkovne baze. MS SQL je zgrajen na podlagi SQL-a, standardiziranega poizvedovalnega jezika, ki se uporablja za upravljanje podatkov v podatkovni bazi. Strežnik SQL pa temelji na Transact-SQL, ki je implementacija SQL-a iz Microsofta, saj so standardnemu SQL-u dodali niz svojih razširitev, zato se tudi imenuje strežnik Microsoft SQL.

Izvorna koda strežnika SQL se je razvijala v osemdesetih letih prejšnjega stoletja pod okriljem Sybase Inc, zdaj pa je v lasti SAP. Koda je bila prvotno zgrajena tako, da se je zagon izvedel v sistemu Unix. V sodelovanju Microsofta in Ashton-Tate Corp je nastala prva različica strežnika Microsoft SQL za OS/2, ki je bil izdan leta 1989. Do danes je izšlo veliko različic strežnika MS SQL, ki so ga preoblikovale v relacijsko podatkovno bazo, kot ga poznamo danes, in dodale nekaj orodij za upravljanje s podatkovno bazo.

HTML

HTML (ang. Hyper Text Markup Language) [16] je jezik za označevanje nadbesedila, ki je namenjen izdelovanju spletnih strani. S pomočjo HTML-ja se spletna stran prikaže v brskalniku, določa pa tudi zgradbo same spletne strani. HTML sestoji iz elementov HTML, ki jih predstavljajo pari značk. Par značk je sestavljen iz začetne in končne značke, posamezno značko obdajajo špičasti oklepaji. Med posamezne značke lahko pišemo poljubno besedilo, ki se prikaže na spletni strani, lahko pa vstavimo še druge značke in tako gnezdimo elemente HTML. HTML lahko zgradimo s pomočjo vsakega še tako preprostega urejevalnika besedila.

CSS

CSS (ang. Cascading Style Sheets) [16] je kaskadna stilska predloga, predstavljena s preprostim slogovnim jezikom, ki ga lahko urejamo v katerem koli urejevalniku besedila. CSS skrbi za sam videz spletne strani tako, da definiramo stil posameznega elementa HTML. Elementom HTML lahko določamo barve, velikosti, odmike, poravnave in še mnogo drugih atributov, lahko nadziramo tudi akcije, ki jih uporabnik izvaja nad elementi HTML, primer: ko uporabnik izvede akcijo in z miško prekrije povezavo, se ta obarva drugače.

Bistvo uporabe CSS je ločiti strukturo in predstavitev strani, tako imamo HTML in CSS, in nam omogoča preprosto dodajanje novih stilskih predlog. Množica strani lahko uporablja iste predloge CSS, kar preprečuje oziroma zmanjšuje ponavljanje kode.

JavaScript

JavaScript [15] je objektni skriptni jezik, s pomočjo katerega so spletne strani postale interaktivne. JavaScript se je razvil ločeno od jave, vendar so se nekatere lastnosti prenesle, razvit je tako, da sodeluje s kodo HTML in tako naredi spletno stran dinamično. Sama sintaksa JavaScripta sledi sintaksi programskega jezika C, saj noben izmed teh dveh jezikov nima vgrajenih vhodno-

izhodnih funkcij, zato se izvajata v odvisnosti od gostitelja. JavaScript se uporablja za ustvarjanje dimaničnih spletnih strani, ki jih ni mogoče zgraditi samo s HTML-jem. JavaScript se vključi v HTML in opravlja naloge, ki niso statične, npr. odpiranje novih oken, preverjanje pravilnosti vnesenih podatkov in še mnogo drugih nalog. Samo kodo JavaScripta je treba prilagoditi tako, da je podprta na vseh brskalnikih, kar se izvede s pisanjem dodatnih funkcij za posamezne brskalnike. Uporablja se tudi v različnih orodjih, ki niso na spletu, Adobe Acrobat in Adobe Reader na primer uporabljata JavaScript v datotekah PDF.

JavaScript je odprtokodni objektni skriptni programski jezik, ki je podprt s strani programskih podjetji in je kot odprt jezik na voljo vsakemu uporabniku za uporabo brez licene. Podprt je tudi v vseh novejših spletnih brskalnikih.

Uporabili smo tudi nekaj dodatnih JavaScript knjižnic, kot sta jQuery [6], ki je namenjen za lažjo manipulacijo, in Bootstrap [1], ki skrbi za lepo postavitvev elementov.

3.1.2 Orodja

Pri razvoju spletne aplikacije za vzdrževanje distribucijskega omrežja smo si pomagali z različnimi orodji, ki so nam olajšala sam razvoj spletne aplikacije in pisanje kode.

IntelliJ IDEA

IntelliJ IDEA [5] je večkrat nadgrajeno integrirano razvojno orodje ali na kratko IDE (ang. Integrated Development Environment), razvili so ga v podjetju JetBrains. Orodje je predvsem namenjeno razvijanju programske opreme v Javi. Orodje je na voljo v zastonski verziji, ki izhaja z odprtokodno licenco, posledično lahko pridobimo tudi izvorno kodo samega orodja, in plačljivi verziji. V obeh primerih se je za razvojno okolje treba prijaviti, da ga nato lahko uporabljamo.

Razvojno okolje IntelliJ IDEA je zelo prilagodljivo z razširitvenimi moduli, tako si lahko vsak razvijalec svoje razvojno okolje s pomočjo zelenih modulov uredi tako, kot mu ustreza. IntelliJ IDEA podpira zelo veliko programskih jezikov, tehnologij in okolij, tako pa pridobimo veliko prilagodljivost.

SQL Server Management Studio

SQL Server Management Studio [11] je programska oprema, ki se je prvič začela uporabljati leta 2005 s strežnikom Microsoft SQL. Namenjena je bila konfiguriranju in upravljanju vseh komponent znotraj strežnika Microsoft SQL. Orodje vsebuje urejevalnike skript, grafično okolje in brskalnik, s pomočjo katerega lahko uporabnik brska med predmeti na strežniku. SQL Server Management Studio je za povezovanje s primerki strežnikov SQL sprva uporabljal ločeno verzijo Expressa, v kasnejše verzije pa je Express že vgrajen v SQL Server Management Studio. Pri Microsoftu so se prav tako odločili, da vključijo združljivost za starejše strežnike SQL, kar pomeni, da se novejše verzije lahko povezujejo na starejše verzije strežnikov SQL.

XAMPP

XAMPP [12] je skupek odprtokodne programske kode, s pomočjo katere lahko tvori delujoč spletni strežnik, na katerem lahko gostimo dinamične spletne strani. XAMPP je sestavljen tako, da se namestitev spletnega strežnika izvede po najhitrejši poti. Zaženemo XAMPP in v nekaj minutah imamo že postavljen delujoč spletni strežnik, to je veliko hitreje kot pa poganjanje zaporedja ukazov, ki iz izvirne kode zgradi aplikacijo. XAMPP so razvili Apache Friends in je sestavljen iz strežnika Apache, podatkovne baze MariaDB in tolmačev (ang. Interpreters) za skripte v jezikih PHP in Perl, od tod tudi kratica XAMPP (Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P)). Apache predstavlja skupek vse programske opreme, ki je nujno potrebna za delovanje lokalnega strežnika, s pomočjo katerega aplikacijo razvijamo in testiramo. XAMPP deluje na vseh platformah, saj za

ustvarjanje spletnega strežnika potrebujejo iste komponente pri Windowsu, Macu in Linuxu.

Gii

Yii2 vključuje orodje, imenovano Gii [3], s pomočjo katerega lahko hitro in na enostaven način generiramo pogosto uporabljene delčke kode. Gii je spletni vmesnik, s pomočjo katerega lahko generiramo zeleno kodo. Generiramo lahko kodo modelov, pogledov in krmilnikov. Pri našem razvoju smo orodje Gii s pridom izkoriščali za generiranje modelov na podlagi podatkovne baze in pogledov, s pomočjo katerih smo naše podatke iz podatkovne baze prikazovali in urejali.

Xdebug

Xdebug [13] je razširitev PHP, ki omogoča odpravljanje napak. Za odpravljanje napak ali razhroščevanje uporablja protokol DBGp. Xdebug smo vključili v naše razvojno orodje IDE 3.1.2 in nam je bil v pomoč pri razvijanju kode, ki smo jo lahko razhroščevali korak za korakom.

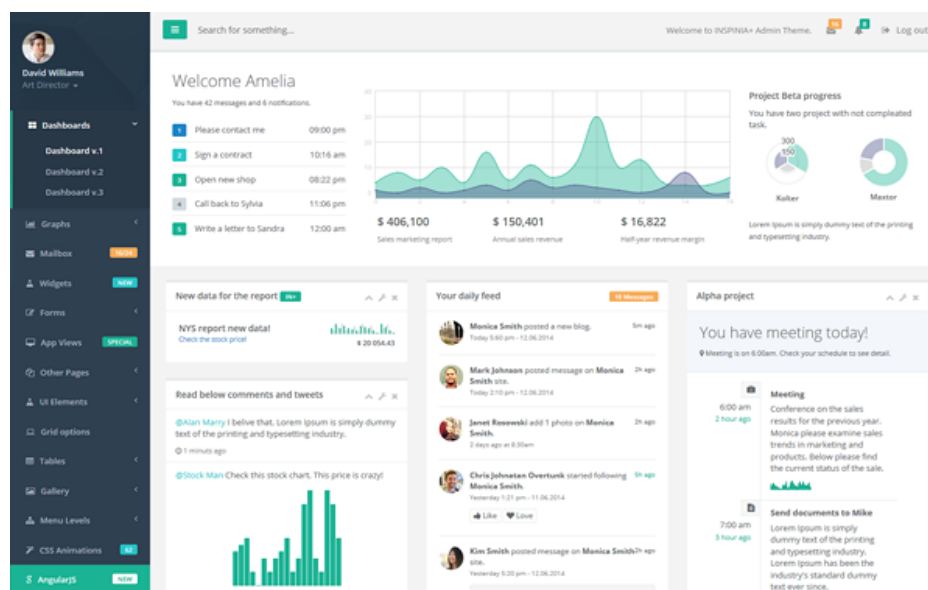
Composer

Composer [2] je orodje za upravljanje knjižnic na ravni aplikacij. Zagotavlja standardni format za upravljanje odvisnosti knjižnic in programske opreme PHP. Razvila sta ga Nils Adermann in Jordi Boggiano leta 2011, prvič pa je izšel leta 2012.

V svoji spletni aplikaciji za vzdrževanje distribucijskega sistema smo composer uporabljali za nameščanje in odstranjevanje knjižnic, kar je delo zelo olajšalo in pohitrilo. S Composerjem samo poženemo ustrezen ukaz in se nam zelena knjižnica samodejno namesti.

3.1.3 Predloga spletne aplikacije

Sodobne spletne aplikacije, ki jih najdemo na spletu, uporabljajo nekatere tehnologije, ki smo jih opisali v poglavju 3.1.1, najbolj osnovne tehnologije so HTML, CSS in JavaScript. Z uporabo teh tehnologij je spletna aplikacija primerna za uporabo na namiznem računalniku, prav tako kot stara aplikacija za vzdrževanje distribucijskega omrežja. Na željo naročnika smo za novo spletno aplikacijo za vzdrževanje distribucijskega sistema razširili spekter uporabljenih tehnologij, ena izmed njih je tudi Bootstrap [1]. Bootstrap je bil v začetku razvoja v verziji v4, ki pa še ni bila dokončno izpopolnjena, zato smo se odločili za verzijo v3, ki je bila že izpopolnjena. S pomočjo knjižnice Bootstrap smo prilagodili spletno aplikacijo tako, da je primerna za uporabo na vseh napravah, kot so namizni računalniki, prenosniki, tablice in pametni telefoni. Celotno spletno aplikacijo pa smo izdelali na podlagi teme INSPINIA [4], ki jo prikazuje slika 3.1. INSPINIA je plačljiva tema, ki uporablja knjižnico Bootstrap, s pomočjo katere je prilagodljiva za različne naprave, stilske predloge pa so spremenjene tako, da smo dosegli želeni videz naše spletne aplikacije. Poleg integrirane knjižnice Bootstrap ima tema INSPINIA integriranih še nekaj drugih knjižnic, ki so nam služile kot zgled pri razvoju. Tema INSPINIA omogoča tudi izbiro barve, za spletno aplikacijo za vzdrževanje distribucijskega sistema smo izbrali standardno barvo teme (ang. default-skin), ki se lepo ujema z barvami podjetja pri naročniku.



Slika 3.1: Primer spletne predloge INSPINIA

3.2 Podatkovni model

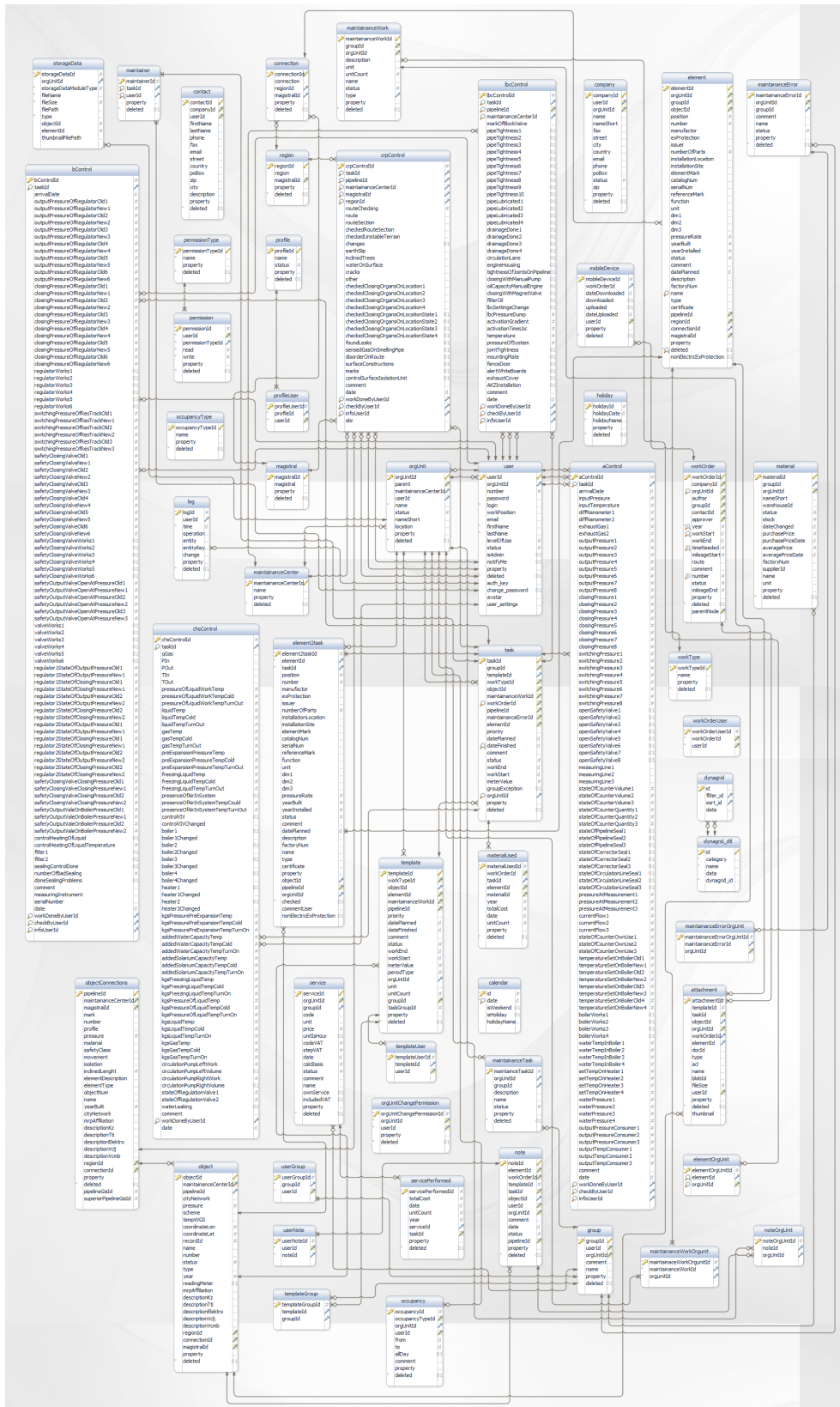
Pred razvojem vsake aplikacije, ki zahteva podatkovni model, je tega treba dobro zasnovati in razviti tako, da bo razvoj aplikacije potekal hitro in brez težav. Pri razvoju spletne aplikacije za vzdrževanje distribucijskega sistema žal ni bilo tako. Podatkovni model je na željo naročnika spletne aplikacije ostal nespremenjen, z minimalnim spremembami in dodelavami, in sicer predvsem zaradi hitrejšega razvijanja spletne aplikacije. Razvoj spletne aplikacije je potekal v iteracijah, zato je med razvojem prišlo do minimalnih sprememb že obstoječega podatkovnega modela, vanj smo dodali tudi nekaj dodatnih tabel, indeksov in povezav, ki so nam služile za hitrejše delovanje in razvoj dodatnih funkcionalnosti, kot so na primer obrazci. Velikokrat je bilo treba v podatkovni model dodati kakšen atribut ali popraviti podatkovni tip in zgraditi dodatne tabele, to smo na enostaven način izvajali s pomočjo migracij, da je razvoj spletne aplikacije potekal kar se da hitro. Na sliki 3.2 je predstavljen podatkovni model z vsemi tabelami, ki smo jih uporabili pri razvoju spletne aplikacije za vzdrževanje distribucijskega sistema. Podatkovni model

vsebuje zelo veliko tabel, zato smo jih združili v skupine in opisali samo nekaj najpomembnejših.

- ***workOrder*** – Tabela, ki hrani zapise delovnih nalogov in vse attribute, potrebne za zapis delovnega naloga. Ena vrstica ali en zapis v tabeli predstavlja natanko en delovni nalog.
- ***task*** – Tabela, ki hrani zapise opravil (ta lahko stojijo samostojno ali na delovnem nalogu) in vse attribute, ki predstavljajo opravilo. Vsak zapis v tabeli predstavlja natanko eno opravilo.
- Tabele šifrantov: ***company, contact, orgUnit, user, group, object-Connections, object, element, material, services, maintenanceWork, maintenanceError, holiday, template, magistral, regionaln, connections in storageData*** – V vsaki tabeli so attribute, namenjeni opisu šifranta, ki mu pripada tabela. Ena vrstica v tabeli predstavlja en zapis šifranta v tabeli. Tabela *storageData* pa je namenjena hrambi podatkov o naloženih predlogah, slikah in priponkah, ravno tako ena vrstica v tabeli predstavlja eno priponko, sliko ali predlogo.
- ***orgUnitChangePermission, prermission in prermissionType*** – Tabele, ki vsebujejo vse potrebne attribute, s katerimi opisujejo pravice menjave oddelkov, šifrantov in pravic do raznih akcij, ki jih lahko uporabnik izvaja v aplikaciji. Ena vrstica v tabeli predstavlja en zapis pravic.
- Tabele obrazcev: ***aControl, bControl, chsControl, lbcControl in crpControl*** – Tabele, namenjene za hranjenje obrazcev, ena vrstica v tabeli predstavlja en obrazec. Tabele so bile kreirane dodatno s pomočjo migracij. Obrazce smo razvijali kot dodatne funkcionalnosti, zato smo tudi sami dodali našete tabele.
- Tabele poročil: ***materialUsed in servicePerformed*** – Ti dve tabeli hranita zapise o poročilih za obračun materialov in storitev. Preostala

poročila pa so zgrajena s pomočjo več povezanih tabel.

- Tabele vzdrževalcev: *workOrderUser*, *userGroup*, *templateUser in maintainer* – Služijo kot vmesne tabele pri povezavah mnogo proti mnogo (ang. many to many) in jih uporabljamo za zapis vzdrževalcev ali uporabnikov skupine, delovnega naloga, periodike in opravil. Ena vrstica v tabeli z vsemi potrebnimi atributi pomeni enega vzdrževalca, ki pripada opravi, delovnemu nalogu, skupini ali periodiki.
- *Ostale tabele* – Služijo za shranjevanje zgodovine sprememb v spletni aplikaciji in nastavitve posameznih uporabnikov, ki jih uporabniki lahko sami spreminjajo s pomočjo spletne aplikacije na vsaki strani, na kateri je to omogočeno.



Slika 3.2: Celotni podatkovni model, uporabljen v Spletni aplikaciji za vzdrževanje distribucijskega sistema.

Poglavje 4

Razvoj spletne aplikacije

Poglavje opisuje posamezne korake razvoja spletne aplikacije s pomočjo agilne prakse ter samo delovanje, pri katerem smo opisali le najbolj uporabljene funkcionalnosti spletne aplikacije za vzdrževanje distribucijskega sistema.

4.1 Opis razvoja spletne aplikacije

Razvoj spletne aplikacije za vzdrževanje distribucijskega sistema je potekal na podlagi starega podatkovnega modela, funkcionalnosti stare aplikacije za vzdrževanje distribucijskega sistema in tesno sodelovanje podjetja Mikrografija in naročnika. Slednje je pripomoglo k temu, da se je spletna aplikacija razvijala po željah naročnika, da jim bo delo kar se da olajšala. Za lepši videz aplikacije smo izbrali temo INSPIANA, ki smo jo opisali v poglavju 3.1.3, in sam videz teme preuredili po željah naročnika. Večji poudarek pa je bil na razvoju funkcionalnosti same spletne aplikacije, saj je bilo nekatere že obstoječe funkcionalnosti treba popraviti ali jih na novo razviti, ker niso delovale pravilno.

Ob vzpostavitvi razvojnega okolja je spletna aplikacija že vsebovala delujočo prijavo in registracijo, ki smo jo odstranili, ker so uporabniki aplikacije lahko samo zaposleni in jim profile ustvarijo nadrejeni, ki imajo dodeljene ustrezne pravice.

Podatkovnega modela nismo ustvarjali znova, ampak smo uporabili star podatkovni model, ki je bil uporabljen v predhodni spletni aplikaciji. Podatkovni model je na željo naročnika ostal nespremenjen, prihranili smo tudi na času vzpostavitve podatkovnega modela, kar je omogočalo hitrejši razvoj spletne aplikacije za vzdrževanje distribucijskega sistema. Za uvoz podatkovnega modela smo uporabili orodja SQL Server Management Studio, ki smo ga predhodno opisali v poglavju 3.1.2, in ga nismo obdelovali ročno, saj je bilo to nepotrebno na začetku razvoja spletne aplikacije. V času razvoja so se v poznejših fazah dodajale potrebne tabele ali smo jih popravili s pomočjo ogrodja Yii2, ki podpira migracijske skripte.

Migracijska skripta vsebuje dve funkciji, prva je *function safeUp()*, v kateri napišemo kodo, ki preoblikuje podatkovni model, kot je potrebno, druga funkcija pa je *function safeDown()*, v kateri spišemo kodo, ki stori ravno obratno in podatkovni model vrne v stanje pred izvedbo migracijske skripte v primeru, da je funkcija za povratek podatkovnega modela spisana pravilno. Migracijske skripte poganjamo v konzoli s pomočjo ukazov yii migrate/up in yii migrate/down, ki sprožita zeleno funkcijo migracijske skripte.

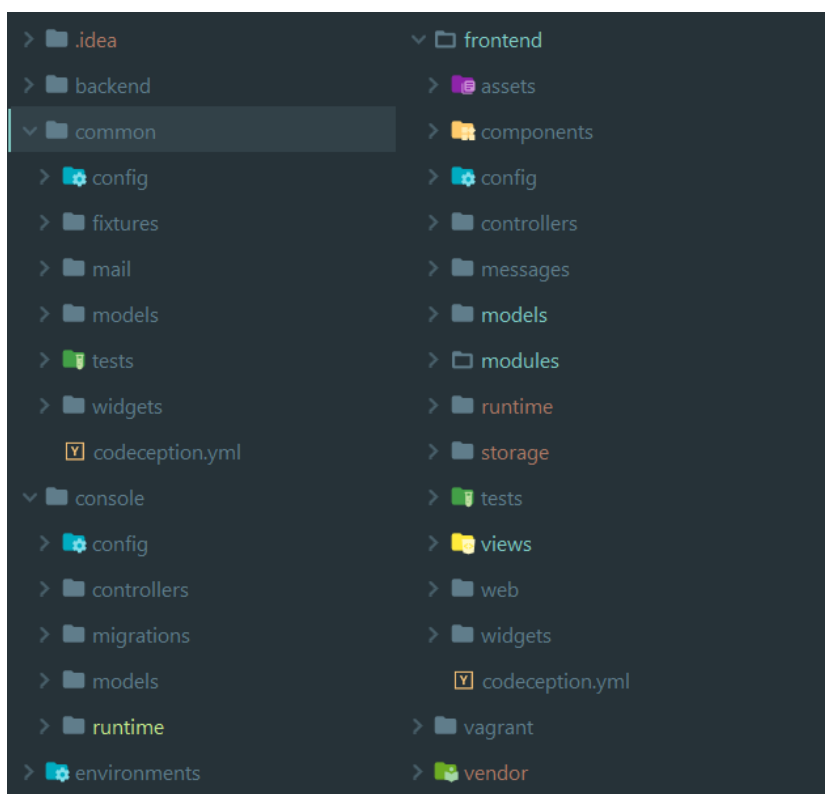
V nadaljevanju sledi primer funkcij *function safeUp()* in *function safeDown()*, s pomočjo katerih smo v tabelah opreme dodali stolpec za ne-električno opremo.

```
public function safeUp()
{
    $this->addColumn('element', 'nonElectricExProtection',
        $this->boolean()->defaultValue(0));
    $this->addColumn('element2task', 'nonElectricExProtection',
        $this->boolean()->defaultValue(0));
    echo Yii::$app->db->createCommand("
        UPDATE element
        SET nonElectricExProtection =
        CASE WHEN element.description
        like '%Neelektri%' THEN 1 ELSE 0 END
```

```
        ")->execute();
    }

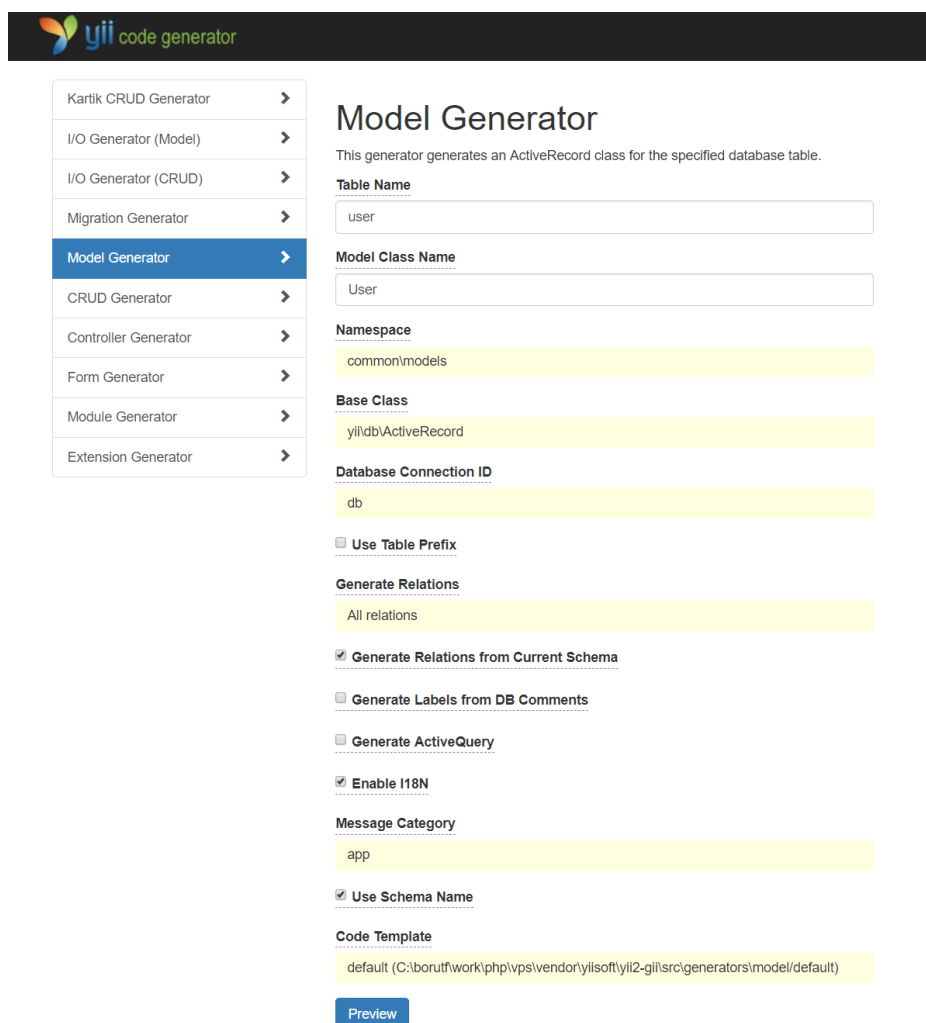
public function safeDown()
{
    $this->dropColumn('element', 'nonElectricExProtection');
    $this->dropColumn('element2task', 'nonElectricExProtection');
}
```

Sam razvoj spletne aplikacije za vzdrževanje distribucijskega sistema smo



Slika 4.1: Struktura projekta spletne aplikacije za vzdrževanje distribucijskega sistema.

si olajšali in pohitrili z uporabo agilne prakse, pri kateri se naročniku v periodi 14 dni dokončane dele spletne aplikacije s pomočjo orodja Gii, ki ga podpira ogrodje Yii2 in je opisano v poglavju 3.1.2, dostavi v testiranje. Agilna metoda Scrumban pripomore k večjemu vključevanju naročnika v sam razvoj in s tem k boljšemu produktu. S pomočjo orodja Gii z možnostjo ustvarjanja modelov (ang. Model Generator) smo najpej ustvarili Modele, to so razredi, s pomočjo katerih hitro dostopamo do podatkov podatkovne baze. Slika 4.2 prikazuje ustvarjanje modela za tabelo User, ki predstavlja uporabnika. Vse modele, ustvarjene s pomočjo orodja Gii, smo shranili v mapo *common/models*, kot je razvidno iz slike 4.1. Ob morebitnih spremembah tabele v podatkovni bazi lahko model ponovno ustvarimo in povežimo obstoječi model, orodje Gii nas o tem predhodno obvesti in nam prepusti odločitev, ali želimo model povežiti z novo ustvarjenim modelom ali ne. Ko model znova ustvarimo in povežimo obstoječega, lahko izgubimo kodo, spisano v obstoječem modelu. Zato smo modele oziroma razrede razširili v mapah *frontend/models* in *frontend/moduls/inventory/models*, ki ju lahko vidimo na sliki 4.1. S tem smo preprečili izgubo ročno spisane kode, ki ostaja v teh dveh mapah in je z novim ustvarjanjem modelov ne moremo povežiti.



yii code generator

- Kartik CRUD Generator >
- I/O Generator (Model) >
- I/O Generator (CRUD) >
- Migration Generator >
- Model Generator >**
- CRUD Generator >
- Controller Generator >
- Form Generator >
- Module Generator >
- Extension Generator >

Model Generator

This generator generates an ActiveRecord class for the specified database table.

Table Name
user

Model Class Name
User

Namespace
common\models

Base Class
yii\db\ActiveRecord

Database Connection ID
db

Use Table Prefix

Generate Relations
All relations

Generate Relations from Current Schema

Generate Labels from DB Comments

Generate ActiveQuery

Enable I18N

Message Category
app

Use Schema Name

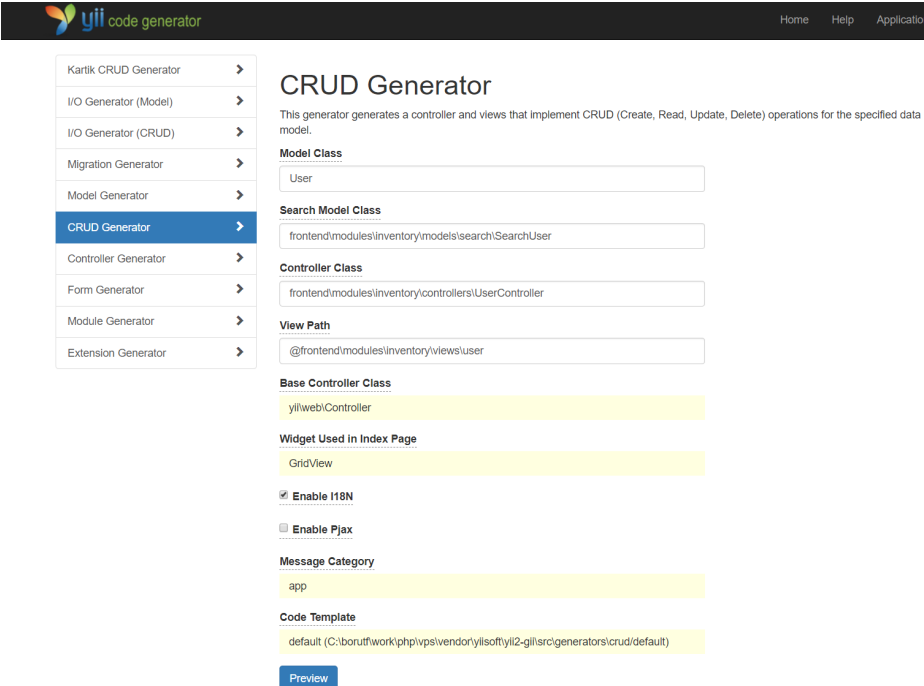
Code Template
default (C:\borutf\work\php\vps\vendor\yii2-gii\src\generators\model\default)

Preview

Slika 4.2: Primer ustvarjanja modela User z Gii možnostjo Model Generator.

Po končanem ustvarjanju modelov smo pričeli z razvijanjem funkcionalnosti in vseh potrebnih pogledov spletne aplikacije. Za lažje delo smo se vrnili k orodju Gii, opisanem v poglavju 3.1.2, in možnosti, ki jo ponuja za ustvarjanje operacij CRUD (ang. CRUD Generator), kot prikazuje slika 4.3. Operacije CRUD so razvidne iz same besede, ki sestoji iz začetnic posamezne operacije C-Ustvari, R-Beri, U-Uredi, D-Briši (ang. C-Create, R-Read, U-

Update, D-Delete). S pomočjo generatorja CRUD smo ustvarili vse operacije CRUD za delovne naloge, opravila in šifrante; ostale poglede smo ustvarili ročno, ker nismo potrebovali vseh operacij CRUD.



The screenshot shows the YII Code Generator web interface. On the left is a sidebar menu with various generator options, with 'CRUD Generator' selected. The main content area is titled 'CRUD Generator' and contains a description: 'This generator generates a controller and views that implement CRUD (Create, Read, Update, Delete) operations for the specified data model.' Below this are several configuration fields:

- Model Class:** User
- Search Model Class:** frontend\modules\inventory\models\search\SearchUser
- Controller Class:** frontend\modules\inventory\controllers\UserController
- View Path:** @frontend\modules\inventory\views\user
- Base Controller Class:** yii\web\Controller
- Widget Used in Index Page:** GridView
- Enable If&N:**
- Enable Pjax:**
- Message Category:** app
- Code Template:** default (C:\borutfwork\php\ipvs\vendor\yiisoft\yii2-gii\src\generators\crud\default)

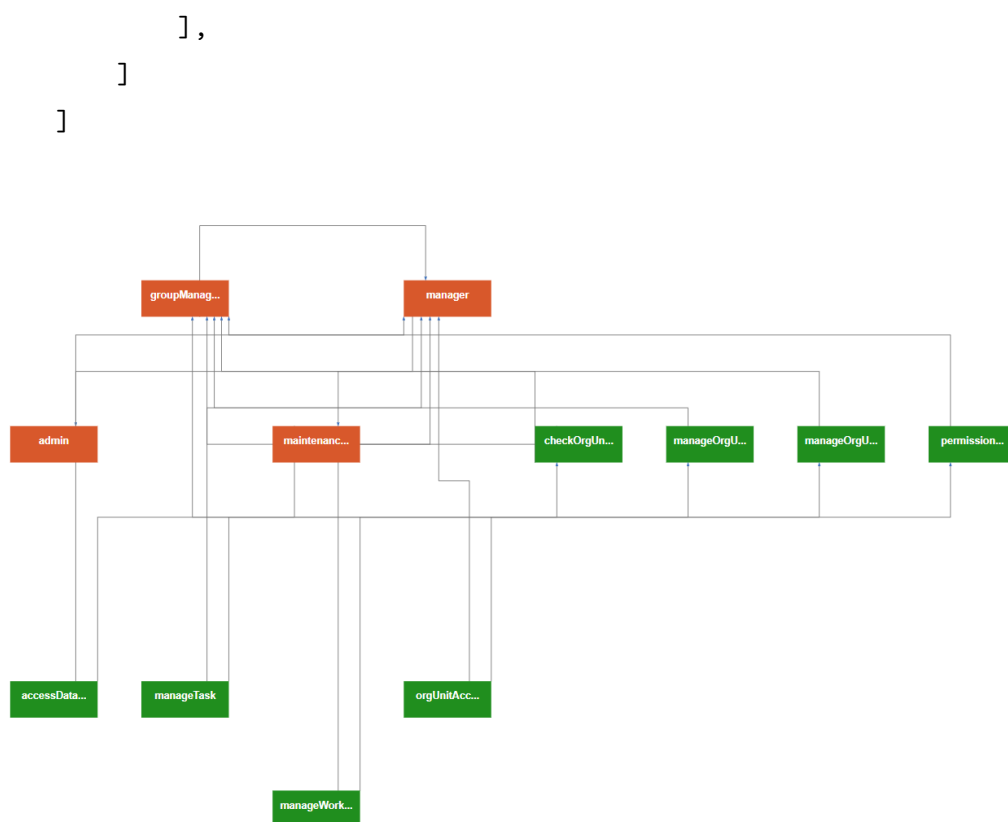
A 'Preview' button is located at the bottom of the configuration area.

Slika 4.3: Primer ustvarjanja pogleda za uporabnike z generatorjem CRUD.

Ko smo ustvarili vse potrebne modele in operacije CRUD, smo pričeli z razvijanjem funkcionalnosti v novi spletni aplikaciji za vzdrževanje distribucijskega sistema, ki jih je bilo treba razviti tako, da so delovale kot v stari aplikaciji. Nekatere funkcionalnosti je bilo treba spremeniti za pravilno delovanje, spet druge dopolniti, ker niso bile popolne ali sploh niso delovale. Funkcionalnosti prijave delujejo enako kot v stari spletni aplikaciji, le da smo odstranili možnost registracije, kajti uporabnike ustvarijo drugi uporabniki z ustreznimi pravicami. Tako se uporabniki lahko samo prijavljajo, kajti spletna aplikacija za vzdrževanje distribucijskega sistema je zaprtega tipa in je namenjena samo zaposlenim v podjetju pri naročniku te spletne aplikacije. Uporabnikom smo dodali tudi pravice do akcij, ki jih lahko izvajajo, to so pravice za dostop do šifrantov, med katerimi sta tudi šifrant uporabnikov in dostop do oddelkov. Pravice smo uporabnikom dodali s pomočjo knjižnice *yii2mod/yii2-rbac*. Po končni konfiguraciji knjižnice smo uporabnikom lahko dodelili pravice avtorizacije in jim določili uporabniške vloge.

Uporabniške vloge lahko dodajamo, urejamo in brišemo ročno s pomočjo orodja RBAC, ki ga ponuja knjižnica *yii2mod/yii2-rbac* in ga prikazuje slika 4.4, in s pomočjo migracijskih skript, ki omogočajo lažjo prenosljivost projekta in sprememb na podatkovnem modelu. Uporabniške vloge smo dodali s pomočjo migracij, v krmilnikih (ang. Controller) je bilo treba napisati, katere uporabniške vloge lahko dostopajo do katerih akcij, to smo določili z naslednjo kodo, ki smo jo dodali v krmilnik uporabnika (ang. UserController).

```
'access' => [
    'class' => AccessControl::className(),
    'rules' => [
        [
            'allow' => true,
            'actions' => [ 'update', 'view', 'create',
                'index', 'index-update', 'delete'],
            'roles' => ['accessDataRecords'],
            'roleParams' => ['user']
```




Slika 4.4: Pregled uporabniških vlog z orodjem RBAC.

Uporabniške vloge:

1. **Admin** – Administrator ali skrbnik spletne aplikacije je prvi na lestvici in ima vse pravice, to pomeni pravice vseh uporabnikov, ki so na lestvici pod njim.
2. **Manager** – Napredni uporabnik lahko izvaja vse akcije vseh oddelkov, nima pravic administratorja, ima pa vse pravice, ki jih imajo osnovni uporabniki in vzdrževalci.
3. **Group manager** – Osnovni uporabniki lahko izvajajo le del akcij iz določene skupine oddelkov, nimajo pravic naprednih uporabnikov in administratorjev, imajo pa pravice vzdrževalcev.

4. *Maintenance worker* – Vzdrževalci so najnižje na lestvici in imajo s tem najmanj pravic, uporabljajo lahko samo akcije, ki so jim dodeljene s strani nadrejenih, enako velja za oddelke.

Sam razvoj spletne aplikacije za vzdrževanje distribucijskega sistema je obsegal tudi razvijanje novih dodatnih funkcionalnosti po željah naročnika spletne aplikacije. Ena izmed teh funkcionalnosti je tudi tiskanje delovnega naloga na podlagi podane predloge za tiskanje, funkcionalnost je v nadaljevanju poglavja podrobneje opisana.



DELOVNI NALOG ŠT. $\{number-year\}$

<u>Oddelek</u>	$\{orgUnit\}$	<u>Datum začetek</u>	$\{workStart\}$
<u>Vzdrževalci</u>	$\{maintainers\}$	<u>Datum konec</u>	$\{workEnd\}$
<u>Izvajalec</u>	$\{company\}$	<u>Sestavil</u>	$\{author\}$
<u>Datum</u>	$\{date\}$	<u>Odhod</u>	
<u>Relacija</u>	$\{woRelation\}$	<u>Prihod</u>	
<u>Začetek km</u>	$\{woStartKm\}$		
<u>Konec km</u>	$\{woEndKm\}$		

	$\{objectNumber\}$	$\{objectName\}$		
<u>S(task)</u>	<u>Vrsta dela</u>	<u>Vzdrževalno delo</u>	<u>Opomba</u>	
	$\{workName\}$	$\{maintenanceName\}$	$\{comment\}$	
	<u>Vrsta napake:</u>	$\{maintenanceErrorName\}$		
	<u>Oprema:</u>	$\{elementName\}$		
	<u>Material:</u>	$\{materialList\}$		

Odgovorni izvajalec: $\{contact\}$
Stran 1
Odobril: $\{approver\}$

Slika 4.5: Primer predloge za tiskanje delovnega naloga.

Razvoj same funkcionalnosti za tisk delovnega naloga smo pričeli s kreiranjem predloge, s pomočjo katere lahko natisnemo delovni nalog z vsemi pripadajočimi opravili in opremo, ki je potrebna za opravljanje posameznega opravila. S pomočjo programa Microsoft Office Word smo zgradili predlogo, ki vsebuje ogrodje in spremenljivke, do katerih lahko dostopamo in jih spreminjamo. Predlogo za tiskanje enega ali več delovnih nalogov prikazuje slika 4.5.

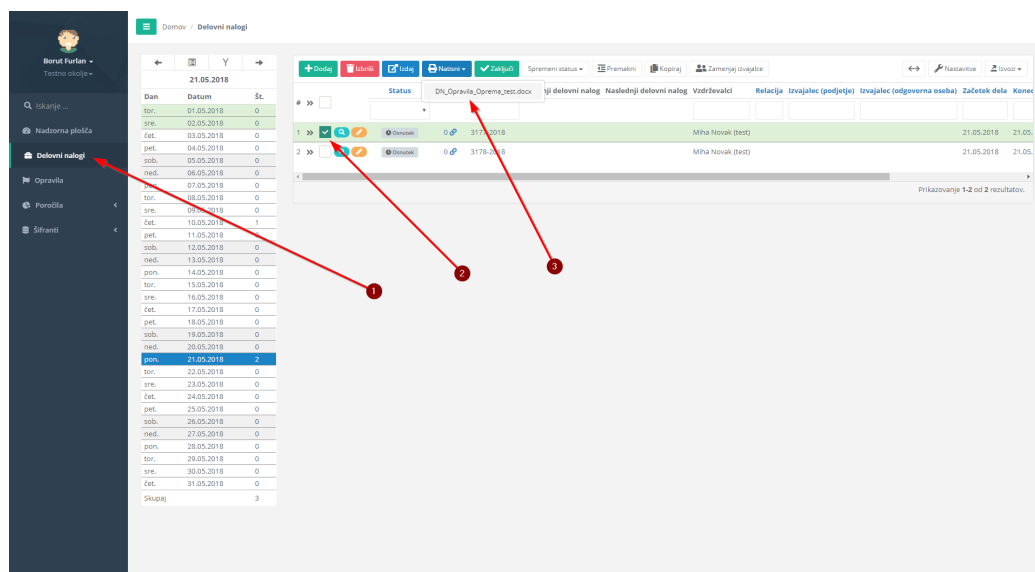
Pred vnašanjem podatkov v predlogo za tisk je treba pridobiti vse potrebne podatke. To so podatki o delovnih nalogih, podatki o vseh pripadajočih opravilih in opremi, ki se uporablja za izvajanje teh opravila. Za zbiranje zelenih podatkov je zadostovalo označevanje delovnih nalogov, ki jih želimo natisniti, s čimer smo pridobili tudi podatke o pripadajočih opravilih in opremi. Izbiranje delovnih nalogov prikazuje 2. korak na sliki 4.6, ki smo ga implementirali s pomočjo stolpca `CheckboxColumn`, ki ga ponuja razširitev za ogrodje `Yii2 Kartik – GridView` [8]. S pomočjo skriptnega jezika `JavaScript` smo podatke o delovnih nalogih posredovali v krmilnik pogleda delovnega naloga za nadaljnjo obdelavo.

```
$tempStoragePath = Yii::getAlias('@runtime') .  
    DIRECTORY_SEPARATOR . 'tmp';  
if (file_exists($tempStoragePath)) {  
    $files = glob($tempStoragePath . DIRECTORY_SEPARATOR . '*');  
    foreach ($files as $file) {  
        if (is_file($file))  
            unlink($file);  
    }  
} else {  
    FileHelper::createDirectory($tempStoragePath);  
}
```

V krmilniku pogleda delovnega naloga na podlagi ID-jev delovnih nalogov zapisanih v `JSON` formatu, ki so bili posredovani iz pogleda v krmilnik, zgradi tabelo ID-jev delovnih nalogov, s pomočjo katerih lažje dostopamo do delovnih nalogov. Nato s predhodno navedeno kodo krmilnika pogleda delovnega naloga preverimo, ali že obstaja kakšna začasna datoteka za shranjeva-

nje; če obstaja, jo pobrišemo in ustvarimo novo datoteko za shranjevanje. V to datoteko nato zgradimo eno ali več predlog za tiskanje delovnih nalogov, kar prikazuje slika 4.6. Predloge napolnimo tako, da na podlagi posredovanih ID-jev poiščemo ustrezne delovne naloge, po katerih se s pomočjo zanke sprehodimo, in za vsak delovni nalog zgradimo svojo predlogo.

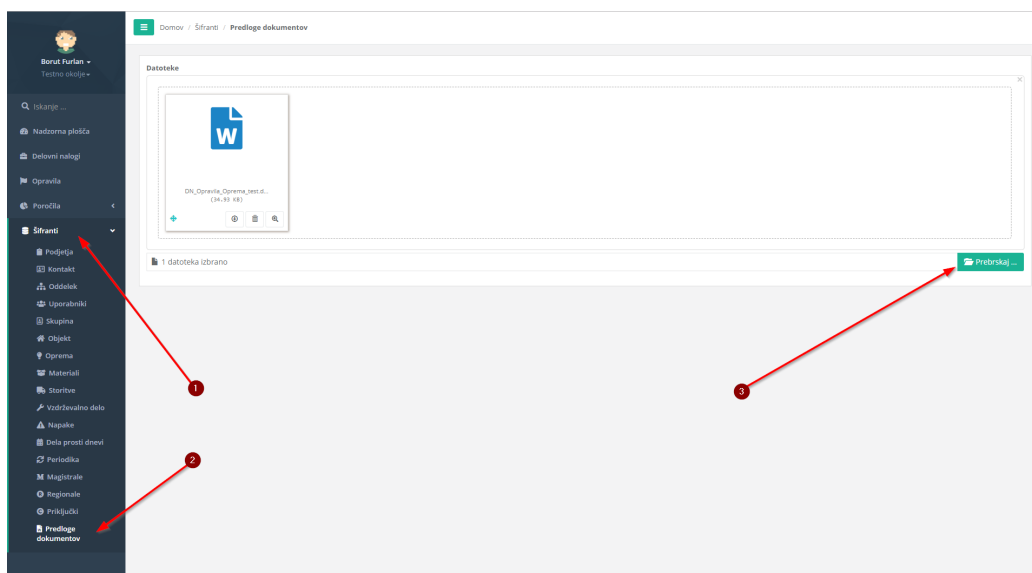
Nato pregleđamo vsa pripadajoča opravila delovnega naloga in ustrezno dopolnimo predlogo; če je opravil na enem delovnem nalogu več, se polje za vpisovanje opravila kopira tolikokrat, da so vpisana vsa opravila, ki pripadajo delovnemu nalogu na predlogi za tisk. Predloga za tisk delovnega naloga je tako skoraj končana, sprehodimo se le še čez vso opremo, ki pripada opravi- lom, in to izpolnimo v predlogi. Tako imamo zgrajene predloge shranjene v začasni datoteki, število zgrajenih predlog je enako številu izbranih delovnih nalogov. Predloge na koncu le še združimo v eno predlogo, ki se prenese na našo napravo.



Slika 4.6: Primer tiskanja delovnega naloga na podlagi predhodno naložene predloge.

Del tiska delovnega naloga je tudi funkcionalnost za nalaganje predloge, kar prikazuje slika 4.7. Za nalaganje datotek, v našem primeru za nalaganje predloge, smo uporabili razširitev ogrodja Yii2 Kartik FileInput Widget [7]. Razširitev smo konfigurirali tako, da omogoča nalaganje samo .doc in .docx, in s tem preprečili morebitne težave z drugimi tipi datotek, ki se naložijo kot predlogo za tisk delovnega naloga.

V krmilniku smo spisali kodo PHP, ki naloženo predlogo shrani v podatkovno bazo še z nekaterimi drugimi podatki, kot sta oddelek in tip shranjene datoteke. Razširitev FileInput omogoča tudi prenašanje datoteke na napravo iz spletne aplikacije, zato je bilo treba določiti tudi pot shranjevanja in tip prenešene predloge.



Slika 4.7: Primer nalaganja predloge za tisk.

Funkcionalnost za tiskanje delovnega naloga v spletni aplikaciji za vzdrževanje distribucijskega sistema deluje po korakih, ki jih prikazujeta sliki 4.7 in 4.6. Predno želimo natisniti enega ali več delovnih nalogov, moramo naložiti predlogo po korakih, razvidnih iz slike 4.7.

1. **Šifranti** – S klikom na šifrante se nam odpre razdelek šifrantov, ki nam omogoča izbiranje zelenega šifranta.
2. **Predloge dokumentov** – Predloge dokumentov, ki nam omogočajo nalaganje predloge za tiskanje delovnega naloga.
3. **Izbiranje predloge** – Uporabimo gumb "Prebrskaj", s pomočjo katerega se nam odpre okno za brskanje med datotekami, izberemo zeleno predlogo in svojo izbiro potrdimo.
4. **Nalaganje predloge** – Ko izberemo zeleno predlogo, se nam ob gumbu "Prebrskaj" prikaže še gumb "Naloži", s katerim izbrane predloge naložimo v spletno aplikacijo in jih shranimo v podatkovni model.

Tako naložimo pravilno zgrajeno predlogo za tisk delovnega naloga, ki mora biti v ustreznem formatu. Če format ni pravilen, nas spletna aplikacija ob nalaganju o tem opozori.

Nato sledita tiskanje delovnega naloga in uporaba predhodno naložene predloge za tiskanje delovnega naloga. Korake tiskanja delovnega naloga prikazuje slika 4.6.

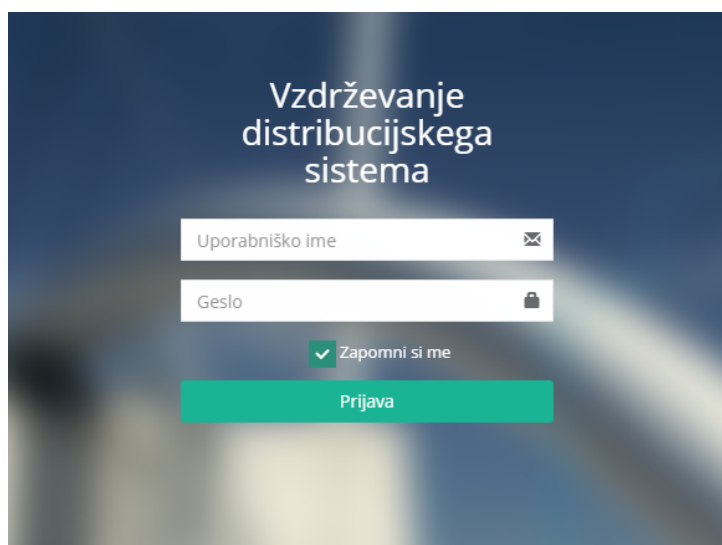
1. **Delovni nalogi** – S klikom na razdelek Delovni nalogi se nam odpre koledarski pregled delovnih nalogov, s pomočjo katerega poiščemo delovne naloge, ki jih želimo natisniti.
2. **Izbiranje** – Izbrane delovne naloge označimo, kot to prikazuje drugi korak. Omogočeno je tudi izbiranje vseh delovnih nalogov hkrati tako, da izberemo kvadrateg ob imenih stolpcev.
3. **Izbiranje predloge** – Z gumbom "Natisniše" nam odpre spustni meni naloženih predlog oddelka, na katerem trenutno smo. Ob kliku na izbrano predlogo se nam zgradi dokument .doc in se prenese na našo napravo.

4. *Tiskanje* – Odpremo prenešen dokument, ki ga lahko ob morebitnih nepravilnih predhodnih izbirah še ustrezno ročno popravimo, in ga natisnemo.

4.2 Opis delovanja spletne aplikacije

V nadaljevanju podpoglavja je s pomočjo slik in opisov predstavljeno delovanje spletne aplikacije za vzdrževanje distribucijskega sistema. Predstavljeno je delovanje funkcionalnosti, ki se največ uporabljajo pri delu v podjetju naročnika spletne aplikacije. Zaradi obsežnosti spletne aplikacije bomo predstavili le nekaj najbolj pogosto uporabljenih pogledov, s pomočjo katerih bo razumevanje delovanja spletne aplikacije lažje. Nekateri pogledi vsebujejo tudi podatke, ki so občutljivi za javnost in last naročnikovega podjetja, zato smo jih zameglili, kar pa ne vpliva na razumevanje delovanja spletne aplikacije. Delovanje spletne aplikacije je predstavljeno s stališča administratorskega uporabniškega računa, v katerem so na voljo vse funkcionalnosti aplikacije.

Pred vstopom v spletno aplikacijo za vzdrževanje distribucijskega sistema se uporabniku prikaže okno za vpis v spletno aplikacijo, ki ga prikazuje slika 4.8. Uporabniki se v spletno aplikacijo lahko samo prijavljajo, ne morejo pa se sami registrirati. Spletna aplikacija je namenjena samo zaposlenim v podjetju pri naročniku spletne aplikacije, zato uporabniške račune ustvarijo zaposleni, ki imajo pravice za ustvarjanje uporabniških računov za novozaposlene člane. Uporabniki se v svoje uporabniške račune prijavijo z uporabniškim imenom in geslom.



Slika 4.8: Vpis v spletno aplikacijo za vzdrževanje distribucijskega sistema.

Po uspešni prijavi se uporabniku prikaže koledarski pregled delovnih nalogov, kot prikazuje slika 4.9. Na tem pogledu je veliko funkcionalnosti, s katerimi upravljamo z delovnimi nalogi. Najbolj osnovna in najbolj uporabljana funkcionalnost je ustvarjanje novega delovnega naloga, opis te sledi v nadaljevanju tega podpoglavja. Vsak delovni nalog v tem pogledu ima možnost razširitve, s katero se pokaže pregled opravil tega delovnega naloga, razširimo lahko tudi vse delovne naloge hkrati.

Naslednje funkcionalnosti, ki so na voljo za upravljanje delovnih nalogov, lahko delujejo na enem ali na več delovnih nalogov hkrati, kar izvedemo tako, da izberemo in označimo zelene delovne naloge, nad katerimi želimo izvesti določeno operacijo, ki je na voljo. Delovne naloge lahko brišemo, lahko jih izdamo in s tem se delovni nalogi prikažejo vzdrževalcem. Delovne naloge lahko tudi zaključimo, če so ustrezno opravljeni. Delovnim nalogom lahko tudi spreminjamo status, s katerim označujemo, v kakšnem stanju je nalog, status delovnega naloga je lahko osnutek, izdan, v odobritvi ali zaključen. S statusom osnutek označimo delovni nalog, ki je bil ustvarjen na novo. Status izdan označuje delovne naloge, izdane vzdrževalcem. Delovni nalog s

#	Status	Opravljeni delovni nalog	Prejeto delovni nalog	Naslednji delovni nalog	Začetek dela	Konec dela	Sestavi	Odobri	Skupina	Odelek
1	✓	251-2018	3001-2018	01.05.2018	01.05.2018					SV (VC MRL)
2	✓	252-2018	3002-2018	01.05.2018	01.05.2018					SV (VC MRL)
3	✓	253-2018	3003-2018	01.05.2018	01.05.2018					SV (VC MRL)
4	✓	290-2018	2900-2018	3003-2018, 3004-2018	01.05.2018	01.05.2018				SV (VC MRL)
5	✓	291-2018	2910-2018	3005-2018, 191-2019	01.05.2018	01.05.2018				SV (VC MRL)
6	✓	292-2018	3006-2018		01.05.2018	01.05.2018				SV (VC MRL)
7	✓	3143-2018	2933-2017		01.05.2018	01.05.2018				SV (VC MRL)
8	✓	3007-2018			03.05.2018	03.05.2018				SV (VC MRL)
9	✓	3003-2018	2950-2018	3020-2018	04.05.2018	04.05.2018				SV (VC MRL)
10	✓	3005-2018	2951-2018	3021-2018	04.05.2018	04.05.2018				SV (VC MRL)
11	✓	2583-2018		3019-2018	04.05.2018	04.05.2018				SV (VC MRL)
12	✓	2579-2018			04.05.2018	04.05.2018				SV (VC MRL)
13	✓	2580-2018		3018-2018	04.05.2018	04.05.2018				SV (VC MRL)
14	✓	2540-2018		3015-2018	05.05.2018	05.05.2018				SV (VC MRL)
15	✓	3000-2018		3019-2018	05.05.2018	05.05.2018				SV (VC MRL)
16	✓	3000-2018	3003-2018	3017-2018	05.05.2018	05.05.2018				SV (VC MRL)
17	✓	3001-2018	3005-2018	3038-2018	05.05.2018	05.05.2018				SV (VC MRL)
18	✓	2544-2018		3036-2018	05.05.2018	05.05.2018				SV (VC MRL)
19	✓	255-2018		3055-2018	06.05.2018	06.05.2018				SV (VC MRL)
20	✓	2948-2018	2825-2018	3050-2018, 3051-2018	06.05.2018	06.05.2018				SV (VC MRL)
21	✓	3007-2018	3010-2018	3045-2018	06.05.2018	06.05.2018				SV (VC MRL)
22	✓	3008-2018	3021-2018	3052-2018	06.05.2018	06.05.2018				SV (VC MRL)
23	✓	3040-2018		3056-2018	06.05.2018	06.05.2018				SV (VC MRL)
24	✓	257-2018			07.05.2018	07.05.2018				SV (VC MRL)
25	✓	3052-2018	3038-2018	3056-2018, 194-2019	07.05.2018	07.05.2018				SV (VC MRL)

Slika 4.9: Koledarski pregled delovnih nalogov.

statusom v odobritvi pomeni, da je delovni nalog opravljen in čaka, da ga nadrejeni potrdi in zaključi.

Izbrane delovne naloge lahko premaknemo na drug datum s pomočjo obrazca, ki se prikaže ob kliku na gumb "Premakni", pri čemer izberemo ustrezeni datum, na katerega želimo premakniti delovne naloge. Pogled delovnega naloga omogoča tudi kopiranje delovnih nalogov s pomočjo obrazca, do katerega dostopamo s klikom na gumb "Kopiraj" in ustrezno popravimo vzdrževalna dela in vrsto dela na delovnih nalogih, ki so bili prekopirani. Prav tako lahko na izbranih delovnih nalogih zamenjamo vzdrževalce delovnih nalogov z obrazcem, ki se prikaže s klikom na gumb "Zamenjaj vzdrževalce".

Sledi ustvarjanje novega delovnega naloga s pomočjo obrazca, ki ga prikazuje slika 4.10, do obrazca dostopamo s klikom na gumb "Dodaj" na pogledu, ki ga prikazuje slika 4.9. Obrazec izpolnimo s pomočjo spustnih seznamov, katerih podatki se napolnijo glede na podatke šifrantov. Nekatera polja so zaklenjena in se izpolnijo samodejno glede na vnesene podatke, nekatera pa se izpolnijo, ko na delovni nalog dodamo opravila. Vnesene podatke shranimo in tako ustvarimo delovni nalog s pogleda na sliki 4.11.

The screenshot shows a web application interface for creating a new work order. The interface is in Slovenian and includes a sidebar with navigation options like 'Nadomestna plošča', 'Delovni nalogi', 'Opravila', 'Poročila', and 'Sifranti'. The main content area is titled 'Ustvari delovni nalog' and contains several input fields and dropdown menus for defining the work order details.

Kilometri (začetek)	Kilometri (konec)	Kilometri (praznika)
1352	1453	101

Slika 4.10: Ustvarjanje novega delovnega naloga.

Na tem pogledu podatkov ne moramo več urejati, morebitne napake lahko uredimo in popravimo v urejevalnem pogledu delovnega naloga, do katerega dostopamo z gumbom "Uredi". Delovni nalog lahko v tem koraku natisnemo, zaključimo ali izdamo vzdrževalcem, vendar sam delovni nalog brez opravil nima pomena. Opravila lahko dodamo ročno s pomočjo obrazca za dodajanje opravil, ki ga prikazuje slika 4.12, pri čemer podatke opravila določimo ročno ali opravilo dodamo iz pregleda planiranih opravil, ki ga vidimo na sliki 4.13.

The screenshot shows a web application interface for managing work orders. The interface is in Slovenian. On the left, there is a dark sidebar with navigation options: 'Delovni nalogi' (Work orders), 'Oprava' (Maintenance), 'Poročila' (Reports), and 'Štatsi' (Statistics). The main content area is a form for creating or editing a work order. The form includes the following fields and sections:

- Header:** 'Delovni nalog [3196-2018] (Dovrniti)'. Buttons for 'Dodaj opremo', '+ Nova oprema', 'Zaloga', 'Naloga', and 'Zaloga'.
- Location and Group:** 'Izvajalec (podjetje)', 'Skupina', and 'Izvajalec (odgovorna oseba)'. A dropdown menu for 'Odbitek' with 'Tehno okoliš' selected.
- Executor:** 'Vzdrževalci' dropdown menu with 'Mihla Novak (best)' selected.
- Dates and Time:** 'Začetek dela' (14.06.2018 07:00), 'Konec dela' (14.06.2018 15:00), and 'Porabljen čas (h:mm)' (08:00).
- Setup and Location:** 'Sestavi' dropdown menu with 'Janez Novak (best)' selected, and 'Relacija' dropdown menu with 'LjNM' selected.
- Kilometers:** 'Kilometri (začetek)' (1302), 'Kilometri (konec)' (1453), and 'Kilometri (razlika)' (101).

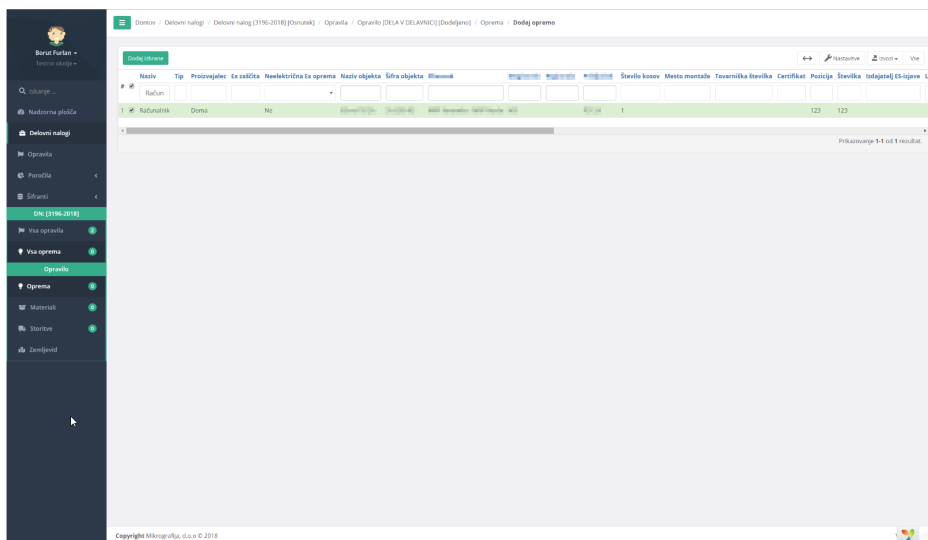
At the bottom of the page, there is a copyright notice: 'Copyright Mikropolija, d.o.o © 2018'.

Slika 4.11: Pregled delovnega naloga.

Za ročno dodajanje opravil uporabimo pogled s slike 4.12 in podatke vnesemo s pomočjo izbir spustnih menijev, enako kot na delovnem nalogu. Delovni nalog lahko vsebuje več opravil, ki se prikazujejo v pregledu pod obrazcem za vnos podatkov opravila. Opravila lahko shranjujemo z gumbom "Šhrani" in z gumbom "Dodaj opremo", ki opravilo shrani in odpre obrazec za dodajanje opreme na shranjeno opravilo delovnega naloga. Opravila so lahko različna glede na vzdrževalna dela, ki so lahko periodična, kar pomeni, da se ponavljajo glede na časovno periodo, ki jo določimo z ročnim vnosom opravil. Periodična opravila se ustvarijo samodejno z delovnim nalogom ob zaključku delovnega naloga.

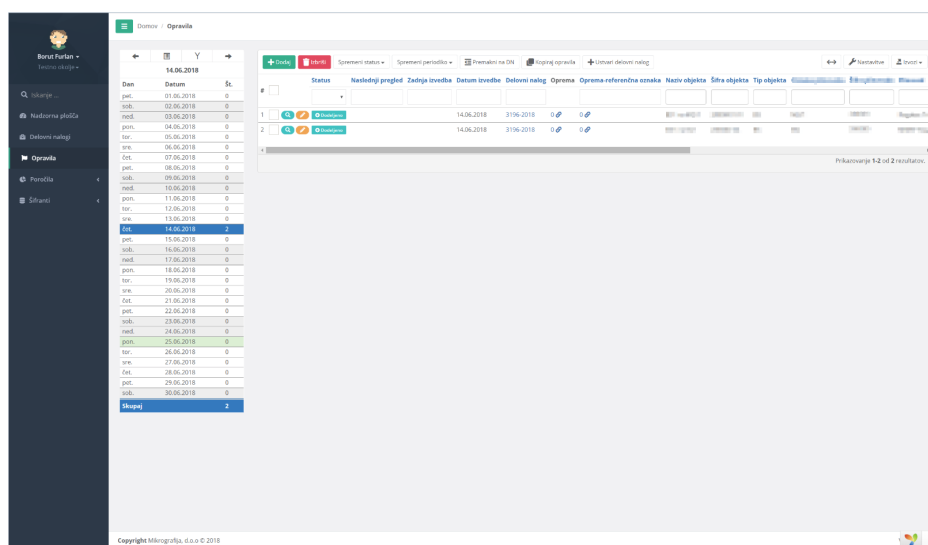
Po uspešnem ustvarjanju delovnega naloga in dodajanju opravil sledi dodajanje opreme, materiala in storitev na opravila delovnega naloga. Dodajanje opreme je zelo podobno dodajanju materiala in storitev, zato je v nadaljevanu bolj podrobno opisano dodajanje opreme na opravila delovnih nalogov. Do pogleda, ki ga prikazuje slika 4.14, lahko dostopamo na več načinov. Opremo lahko na opravilo delovnega naloga dodamo ob kreiranju opravila, po potrebi jo lahko dodamo tudi pozneje s klikom na gumb "Oprema" v razdelku levega menija, ki se nam prikaže, če se nahajamo na opravilu. Opremo dodamo tako, da označimo vso opremo, ki jo želimo dodati na opravilo, in svojo izbiro potrdimo z gumbom "Dodaj izbrane". Nato se nam odpre pogled za pregled opreme, ki smo jo dodali na opravilo delovnega naloga. To opremo lahko tudi izberemo in zamenjamo z drugo opremo.

Opravila lahko ustvarjamo tudi brez delovnih nalogov; opravila brez delovnih nalogov se nahajajo med planiranimi opravili, vsa opravila pa lahko pregledujemo in obvladujemo v koledarskem pogledu s slike 4.15. Koledarski pogled opravil ponuja možnosti za spreminjanje statusa opravil, ki je lahko planirano, dodeljeno, opravljeno, ne bo opravljeno, neopravljeno in za-



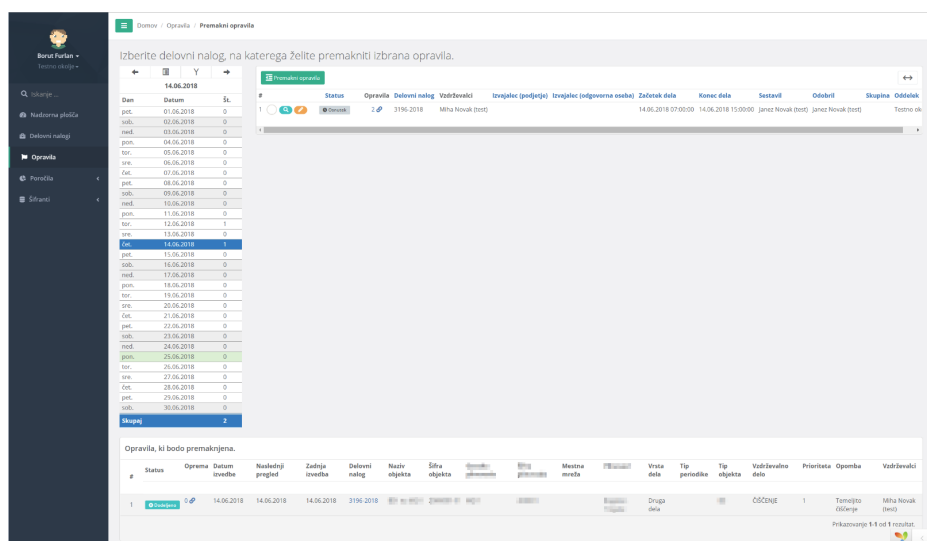
Slika 4.14: Dodajanje opreme na opravila delovnega naloga.

ključeno. Spremenimo lahko tudi periodiko opravila, ki je lahko LIFO, FIFO in Vedno isti, kar pomeni, kako se bodo razporejali vzdrževalci za periodična opravila. LIFO pomeni, da zadnji vzdrževalec prvi opravlja opravilo, FIFO pomeni, da prvi vzdrževalec prvi opravlja opravilo, Vedno isti pa označuje vzdrževalce, ki bodo vedno opravljali to opravilo. Opravila lahko kopiramo tudi na isti datum kot kopirano opravilo.



Slika 4.15: Koledarski pregled opravil.

Opravila lahko izbiramo in označujemo, kar nam omogoča, da iz označenih opravil ustvarimo delovni nalog, prikaže pa se nam pogled za ustvarjanje delovnega naloga s slike 4.10. Označena opravila lahko tudi premaknemo na delovni nalog, ki ga s pomočjo obrazca na sliki 4.16 izberemo in potrdimo.



Slika 4.16: Pogled za izbiranje delovnega naloga, na katerega bomo premaknili opravila.

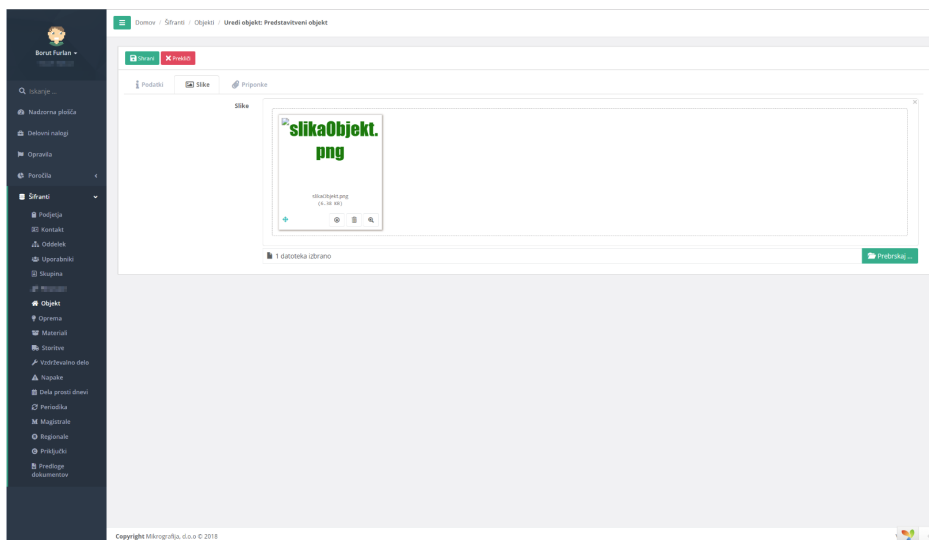
V nadaljevanju je opisan pregled objektov, ki ga prikazuje slika 4.17, in dodajanje šifrant objekt. Na pregledu objektov so na voljo samo operacije CRUD, ki smo jih opisali v poglavju 4.1, in sicer so to operacije za branje, urejanje, brisanje in ustvarjanje. Šifrant objektov ustvarimo enako kot vse druge šifrante spletne aplikacije. Dodajanje šifrantov smo razdelili v tri zavihke, to so zavihek za podatke, ki ga prikazuje slika 4.18, zavihek za dodajanje slik na sliki 4.19 in zavihek za dodajanje predlog, ki ga prikazuje slika 4.20. V obrazec za vnos šifrant objektov podatke vnesemo s pomočjo spustnih seznamov ali jih vpišemo ročno; če smo vnesli podatek napačnega tipa, nas aplikacija o tem opozori. Nalaganje slik in priponk deluje enako, z uporabo gumba "Prebrskaj" se nam odpre okno za brskanje med datotekami, med katerimi izberemo zeleno priponko ali sliko, ki jo želimo naložiti na šifrant objektov, izbiro potrdimo. Nato s pomočjo gumba »Naloži« svojo izbiro naložimo na šifrant objektov. Tako smo vnesli vse zelene podatke, ki jih šifrant objektov potrebuje. Nov objekt se v šifrantu objektov ustvari ob shranjevanju.

#	Šifra objekta	Ime objekta	Tip	Šifra mesta	Šifra občine	Šifra države	Šifra projekta	Šifra dela	Šifra mesta	Vzdrževalni center	Titul	Leto izdelave
1	00000001	00000001	0000	000000	0000	0000	0000	0000	0000	0000	0000	2018
2	00000002	00000002	0000	000000	0000	0000	0000	0000	0000	0000	0000	2018
3	00000003	00000003	0000	000000	0000	0000	0000	0000	0000	0000	0000	2018
4	00000004	00000004	0000	000000	0000	0000	0000	0000	0000	0000	0000	2018
5	00000005	00000005	0000	000000	0000	0000	0000	0000	0000	0000	0000	2018
6	00000006	00000006	0000	000000	0000	0000	0000	0000	0000	0000	0000	2018
7	00000007	00000007	0000	000000	0000	0000	0000	0000	0000	0000	0000	2018
8	00000008	00000008	0000	000000	0000	0000	0000	0000	0000	0000	0000	2018
9	00000009	00000009	0000	000000	0000	0000	0000	0000	0000	0000	0000	2018
10	00000010	00000010	0000	000000	0000	0000	0000	0000	0000	0000	0000	2018
11	00000011	00000011	0000	000000	0000	0000	0000	0000	0000	0000	0000	2018
12	00000012	00000012	0000	000000	0000	0000	0000	0000	0000	0000	0000	2018
13	00000013	00000013	0000	000000	0000	0000	0000	0000	0000	0000	0000	2018
14	00000014	00000014	0000	000000	0000	0000	0000	0000	0000	0000	0000	2018
15	00000015	00000015	0000	000000	0000	0000	0000	0000	0000	0000	0000	2018
16	00000016	00000016	0000	000000	0000	0000	0000	0000	0000	0000	0000	2018
17	00000017	00000017	0000	000000	0000	0000	0000	0000	0000	0000	0000	2018
18	00000018	00000018	0000	000000	0000	0000	0000	0000	0000	0000	0000	2018
19	00000019	00000019	0000	000000	0000	0000	0000	0000	0000	0000	0000	2018
20	00000020	00000020	0000	000000	0000	0000	0000	0000	0000	0000	0000	2018
21	00000021	00000021	0000	000000	0000	0000	0000	0000	0000	0000	0000	2018
22	00000022	00000022	0000	000000	0000	0000	0000	0000	0000	0000	0000	2018
23	00000023	00000023	0000	000000	0000	0000	0000	0000	0000	0000	0000	2018
24	00000024	00000024	0000	000000	0000	0000	0000	0000	0000	0000	0000	2018

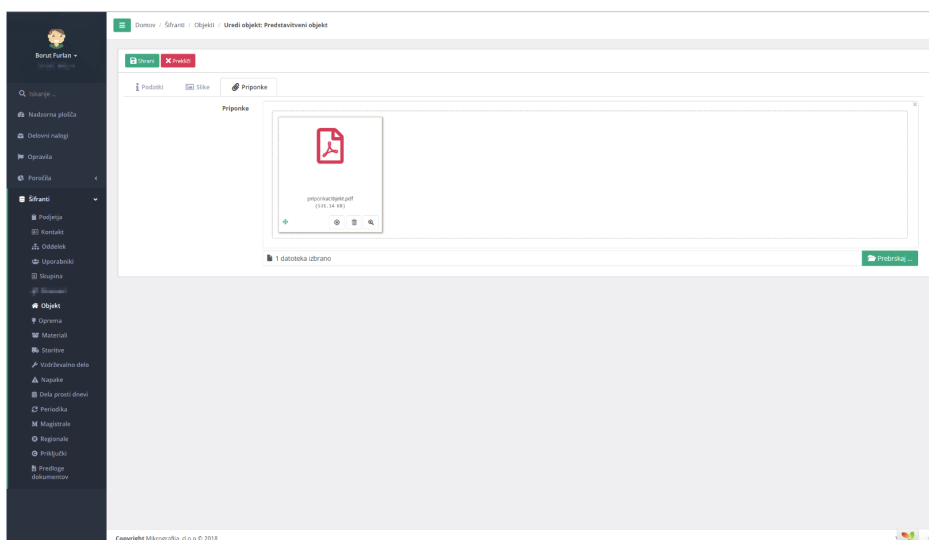
Slika 4.17: Pogled šifrantu objektov.

Šifra objekta: 123
 Ime objekta: Predstavitveni objekt
 Tip: predstavitev
 Mesto: Ljubljana
 Vzdrževalni center: Ljubljana
 Titul: 10
 Leto izdelave: 2018
 Zemljepisna dolžina: 14.5057515
 Zemljepisna širina: 46.0568405
 Stanje: Aktiven
 Skupina: 0000
 Kategorija: 0000
 Prilagoditev BMP: Ljubljana
 Opis KZ:
 Opis TT1:
 Opis Elektro:
 Opis VCU:
 Opis YCMB:

Slika 4.18: Ustvarjanje šifrantu objekt vpis podatkov.



Slika 4.19: Ustvarjanje šifranta objekt nalaganje slik.



Slika 4.20: Ustvarjanje šifranta objekt nalaganje prilopk.

Poglavje 5

Testiranje in evaluacija

Glavni namen razvoja spletne aplikacije za vzdrževanje distribucijskega sistema je bil razvoj novega uporabniškega vmesnika v novih tehnologijah na podlagi funkcionalnosti predhodne spletne aplikacije. Ob zaključku razvoja spletne aplikacije za vzdrževanje distribucijskega sistema smo izvedli nekaj testov na primerih vsakodneвне uporabe spletne aplikacije. Testirali smo staro in na novo razvito spletno aplikacijo. S pomočjo izvedenih testov bomo v nadaljevanju poglavja primerjali uporabniški vmesnik predhodne spletne aplikacije in uporabniški vmesnik novo razvite spletne aplikacije za vzdrževanje distribucijskega sistema in na podlagi porabljenih časov za izvedbo testnih nalog primerjali, katera spletna aplikacija za vzdrževanje distribucijskega sistema ima boljši uporabniški vmesnik.

Testiranje uporabniških vmesnikov spletnih aplikacij za vzdrževanje distribucijskega sistema je potekalo s pomočjo naslednjih testnih nalog. Testne naloge so izvajale osebe, ki so poznale obe spletni aplikaciji, in osebe, ki so se s spletno aplikacijo srečale prvič.

1. **korak** – Preverimo, ali smo na oddelku "Testno okolje", če ne, se ustrezno prestavimo na oddelek "Testno okolje".
2. **korak** – V šifrantih ustvarimo novo skupino z zelenimi vzdrževalci na oddelku "Testno okolje".

3. **korak** – Na oddelku "Testno okolje" ustvarimo novo planirano opravilo dela v delavnici, na katerega dodamo predhodno ustvarjeno skupino in današnji datum. Opravilo mora biti periodična naloga s periodo enega dne.
4. **korak** – Na oddelku "Testno okolje" ustvarimo nov delovni nalog s predhodno ustvarjeno skupino in opravilom, ki ga dodamo iz planiranih opravil.
5. **korak** – Ustvarjeni delovni nalog premaknemo za dva dneva naprej.
6. **korak** – Ustvarjeni delovni nalog zaključimo, poiščemo delovni nalog, ki ga je spletna aplikacija za vzdrževanje distribucijskega sistema ustvarila sama na podlagi periodike.
7. **korak** – Delovni nalog, ki ga je spletna aplikacija za vzdrževanje distribucijskega sistema ustvarila sama na podlagi periodike, nato natisnemo.

Pri testiranju obeh spletnih aplikacij je sodelovalo 10 anonimnih oseb. Testne osebe so opravljale iste testne naloge, razdelili smo jih v dve skupini na podlagi tega, ali so s spletno aplikacijo seznanjeni ali so se z njo srečali prvič. Osebe za testiranje spletne aplikacije za vzdrževanje distribucijskega sistema so v testiranju sodelovale prostovoljno in anonimno.

Rezultati testiranja spletne aplikacije za vzdrževanje distribucijskega sistema so izmerjeni časi, ki so jih testne osebe porabile za izvajanje posameznih testnih nalog, in so zbrani v dveh tabelah. Vsaka tabela rezultatov je namenjena svoji testni skupini glede na to, ali so s spletno aplikacijo seznanjeni ali ne.

Po končanem testiranju smo primerjali rezultate, zbrane v tabelah 5.1 in 5.2, primerjali smo rezultate obeh spletnih aplikacij v obsegu znotraj ene testne skupine. Primerjave v posamerni testni skupini so pokazale, da so časi, porabljeni za opravljanje testnih nalog v novem uporabniškem vmesniku, krajši kakor v stari spletni aplikaciji.

Stara Spletna aplikacija	Oseba 1	Oseba 2	Oseba 3	Oseba 4	Oseba 5
1. korak	12,8	15,1	19,4	19,1	19,5
2. korak	20,2	21,5	28,3	26,3	25,5
3. korak	35,3	36,3	45,6	42,4	38,5
4. korak	40,7	43,6	53,7	47,1	46,2
5. korak	11,5	12,5	12,9	14,6	16,4
6. korak	5,2	5,9	6,7	6,5	6,1
7. korak	9,8	9,3	10,2	9,9	11,5
Skupaj	135,5	144,2	176,8	165,9	163,7
Nova spletna aplikacija	Oseba 1	Oseba 2	Oseba 3	Oseba 4	Oseba 5
1. korak	4,8	4,2	5,7	8,3	5,3
2. korak	16,0	17,9	19,2	20,8	18,6
3. korak	26,3	27,0	24,1	28,6	29,1
4. korak	30,2	37,2	34,6	38,1	35,4
5. korak	7,3	8,2	6,9	9,4	8,9
6. korak	8,2	8,6	10,2	11,0	9,8
7. korak	4,4	5,2	5,6	6,5	6,9
Skupaj	97,2	108,3	106,3	122,7	114

Tabela 5.1: Izmerjeni časi v sekundah seznanjenih testnih oseb.

Stara Spletna aplikacija	Oseba 1	Oseba 2	Oseba 3	Oseba 4	Oseba 5
1. korak	27,4	25,5	34,2	35,3	34,7
2. korak	54,9	52,8	56,6	66,3	68,1
3. korak	67,1	70,8	68,2	70,1	73,4
4. korak	89,6	88,6	93,0	91,2	95,2
5. korak	28,9	38,8	36,1	28,8	34,5
6. korak	24,3	34,7	29,4	28,4	34,1
7. korak	22,2	23,4	24,8	23,7	25,4
Skupaj	314,4	334,6	342,3	273,7	362,1
Nova spletna aplikacija	Oseba 1	Oseba 2	Oseba 3	Oseba 4	Oseba 5
1. korak	12,2	19,1	15,4	16,3	12,1
2. korak	36,6	39,8	39,1	42,9	45,2
3. korak	72,1	89,5	69,9	75,2	78,4
4. korak	66,9	71,0	75,3	68,2	69,5
5. korak	41,7	50,4	54,1	45,8	48,5
6. korak	22,8	25,5	25,8	29,1	32,5
7. korak	13,9	21,9	14,2	11,5	16,8
Skupaj	266,2	317,2	293,8	289	303

Tabela 5.2: Izmerjeni časi v sekundah neseznanih testnih oseb.

Rezultati testiranja spletnih aplikacij za vzdrževanje distribucijskega sistema so pokazali, da so časi, porabljeni za testne naloge, odvisni od predhodne seznanitve s spletno aplikacijo, kajti testna skupina, ki je bila s spletno aplikacijo predhodno seznanjena, je v povprečju dosegla za polovico boljše rezultate ne glede na spletno aplikacijo. So pa primerjave razlik izmerjenih časov znotraj posamezne skupine pokazale, da je na novo razvit uporabniški vmesnik bolj intuitiven, saj so enake testne naloge udeleženci v novem uporabniškem vmesniku opravili hitreje.

Na podlagi rezultatov tabel 5.1 in 5.2, ki smo jih pridobili s pomočjo testnih oseb in primerjav samih rezultatov, smo prišli do ugotovitve, da je uporabniški vmesnik nove spletne aplikacije za vzdrževanje distribucijskega sistema bolj intuitiven, na kar kažejo krajši izmerjeni časi ob opravljanju testnih nalog obeh skupin v novi spletni aplikaciji za vzdrževanje distribucijskega sistema. Eden izmed glavnih ciljev razvoja spletne aplikacije za vzdrževanje distribucijskega sistema in njenega novega uporabniškega vmesnika, je bil razviti boljši, bolj učinkovit in bolj intuitiven uporabniški vmesnik. Glede na izvedene testne naloge in primerjave izmerjenih časov je cilj dosežen.

Poglavje 6

Sklepne ugotovitve

V času študentskega dela se je razvila ideja za ponoven razvoj spletne aplikacije za vzdrževanje distribucijskega sistema zaradi zastarelosti in pomanjkanja vzdrževanja predhodne spletne aplikacije. Idejo smo v okviru diplomske naloge realizirali v podjetju Mikrografija d.o.o. V tesnem sodelovanju z naročnikom smo razvijali novo, boljšo spletno aplikacijo. En glavnih ciljev razvoja spletne aplikacije je bil ustvariti boljši in bolj intuitiven uporabniški vmesnik.

Sam razvoj spletne aplikacije smo pričeli z dogovorom, kakšna bo, kako bo videti in katere funkcionalnosti bo vsebovala. Dogovorili smo se tudi glede podatkovnega modela, ki je na željo naročnika in v želji po prihranku dragocenega časa na tem področju, ostal enak z nekaj popravki in dodelavami. Podatkovni model smo v času razvoja spletne aplikacije tudi dodelali z dodatnimi tabelami za beleženje zgodovine in hranjenje obrazcev.

V sodelovanju in dogovarjanju z razvijalci smo v podjetju Mikrografija izbrali najbolj ustrezne in poznane tehnologije in orodja, s katerimi so se razvijalci v podjetju že srečali. Vzpostavili smo delovno okolje, uvozili podatkovno bazo ter izbrali predlogo Bootstrap naše spletne aplikacije in jo prilagodili barvam naročnikovega podjetja. Ob sami vzpostavitvi delovnega okolja smo že imeli delujočo aplikacijo s prijavo in registracijo, ki smo jo odstranili, saj je spletna aplikacija za vzdrževanje distribucijskega sistema

zaprtega tipa in uporabnikom ne omogoča registracije, ampak samo prijavo v račune, ki jih ustvarijo zaposleni v naročnikovem podjetju z ustreznimi pravicami. Velika večina funkcionalnosti stare spletne aplikacije se je prenesla na novo spletno aplikacijo, nekatere funkcionalnosti je bilo treba dodelati, spet druge spremeniti, kajti v stari aplikaciji niso delovale zaradi neupravljanosti, nekaj funkcionalnosti zaradi neuporabe nismo razvili.

V času razvoja spletne aplikacije smo naleteli na veliko težav, ki smo jih smo brez naporov reševali v sodelovanju in komunikaciji z naročnikom spletne aplikacije. Samo sodelovanje pa je pripeljalo do razvijanja še dodatnih funkcionalnosti spletne aplikacije, kot je na primer tiskanje delovnega naloga, ki poleg opravi natisne vse potrebne materiale, opremo in storitve, ki služijo za opravljanje opravi delovnih nalogov.

Spletna aplikacija za vzdrževanje distribucijskega sistema se je dokončno razvila, vendar se v sodelovanju z naročnikom spletne aplikacije na njegove želje nadgrajuje in ustvarja nove funkcionalnosti, ki bodo omogočale lažje delo.

Ob koncu razvoja smo spletno aplikacijo tudi pretestirali s pomočjo testnih oseb, ki so tako razvijalci iz podjetja kot prijatelji in znanci. Razvijalci so s spletno aplikacijo seznanjeni, zato so dosegali boljše rezultate kot pa prijatelji in znanci, ki so se s spletno aplikacijo srečali prvič. Vendar so bili rezultati testiranja boljši na spletni aplikaciji z novim uporabniškim vmesnikom, kar pomeni, da smo svoj cilj o izboljšanju intuitivnosti uporabniškega vmesnika dosegli.

Spletna aplikacija ima še veliko možnosti izboljšav in dodatnih funkcionalnosti, ki se bodo razvijale po željah in potrebah naročnika. V spletni aplikaciji je že deloma razvita funkcionalnost zemljevida, ki je le prikaz koordinat posameznih opravi, ki jih je treba opraviti. Kot dodatno funkcionalnost bi se lahko razvilo še usmerjanje po zemljevidu od trenutne lokacije do lokacije opravi, ki se nahaja na opravi.

Tako spletna aplikacija za vzdrževanje distribucijskega sistema čaka na nove funkcionalnosti in nove naročnike, za katere jo bomo lahko preuredili in

prilagodili njihovim potrebam.

Literatura

- [1] Bootstrap. Dosegljivo: <https://getbootstrap.com/docs/3.3/>. [Dostopano: 30. 8. 2018].
- [2] Composer. Dosegljivo: <https://getcomposer.org/>. [Dostopano: 30. 8. 2018].
- [3] Gii. Dosegljivo: <https://yii2-framework.readthedocs.io/en/stable/guide/tool-gii/>. [Dostopano: 30. 8. 2018].
- [4] Inspinia. Dosegljivo: <https://wrapbootstrap.com/theme/inspinia-responsive-admin-theme-WB0R5L90S>. [Dostopano: 30. 8. 2018].
- [5] Intellij idea. Dosegljivo: https://sl.wikipedia.org/wiki/IntelliJ_IDEA. [Dostopano: 30. 8. 2018].
- [6] jquery. Dosegljivo: <https://jquery.com/>. [Dostopano: 30. 8. 2018].
- [7] Kartik - fileinput widget. Dosegljivo: <http://demos.krajee.com/widget-details/fileinput>. [Dostopano: 8. 9. 2018].
- [8] Kartik - gridview. Dosegljivo: <http://demos.krajee.com/grid>. [Dostopano: 8. 9. 2018].
- [9] Microsoft sql strežnik. Dosegljivo: <https://searchsqlserver.techtarget.com/definition/SQL-Server>. [Dostopano: 27. 8. 2018].
- [10] Php. Dosegljivo: <http://php.net/>. [Dostopano: 27. 8. 2018].

-
- [11] Sql server management studio. Dosegljivo: https://en.wikipedia.org/wiki/SQL_Server_Management_Studio. [Dostopano: 30. 8. 2018].
- [12] Xampp. Dosegljivo: <https://sl.wikipedia.org/wiki/XAMPP>. [Dostopano: 30. 8. 2018].
- [13] Xdebug. Dosegljivo: <https://en.wikipedia.org/wiki/Xdebug>. [Dostopano: 30. 8. 2018].
- [14] Yii2 ogrodje. Dosegljivo: <https://www.yiiframework.com/doc/guide/2.0/en/intro-yii>. [Dostopano: 27. 8. 2018].
- [15] Goodman Danny, Morrison Michael, Novitski Paul, and Rayl Tia, Gustaff. *JavaScript Bible*. Wiley Publishing, Inc., 2010.
- [16] Duckett Jon. *HTML and CSS Design and build websites*. John Wiley and Sons, Inc., 2011.