

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKA

Blaž Cegnar

**APLIKACIJA ZA UPRAVLJANJE INVENTARJA
RAČUNALNIŠKE OPREME**

DIPLOMSKO DELO
VISOKOŠOLSKEGA STROKOVNEGA ŠTUDIJA

Mentor: viš. pred. dr. Igor Rožanc

Ljubljana, 2011



Št. naloge: 00161/2011

Datum: 05.09.2011

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **BLAŽ CEGNAR**

Naslov: **APLIKACIJA ZA UPRAVLJANJE INVENTARJA RAČUNALNIŠKE
OPREME**
**APPLICATION FOR MANAGEMENT OF THE COMPUTER EQUIPMENT
INVENTORY**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

V diplomski nalogi predstavite razvoj učinkovite večuporabniške spletne aplikacije za upravljanje inventarja računalniške opreme v podjetju. Pri tem najprej analizirajte nekaj tipičnih aplikacij s tega področja, definirajte zahteve in sistematično razvijte prilagodljivo spletno aplikacijo, ki omogoča tudi hranjenje in obdelavo poljubnih podatkov. Nalogo zaključite z opisom in uporabo rešitve.

Mentor:


viš. pred. dr. Igor Rožanc

Dekan:


prof. dr. Nikolaj Zimic



Zahvala

Za podporo in pomoč bi se zahvalil mojemu mentorju, viš. pred. dr. Igorju Rožancu, prav tako pa uni. dipl. inž. Antonu Zvonetu Žličarju ter univ. dipl inž. Mitji Novaku in univ. dipl. inž. Urški Lah za idejo naloge in pomoč pri izdelavi.

Kazalo

Kazalo slik	5
Povzetek	8
Summary.....	9
1. Uvod	10
2. Opis problema	11
3. Analiza obstoječih rešitev.....	11
3.1. Program Hardware Asset Tracker	11
3.2. Program Asset Manager	12
3.3. Program Computer Admin Pro 5.....	13
3.4. Program Total Network Inventory	14
4.Zasnova in izdelava lastne rešitve	15
4.1. Definicija zahtev	15
4.1.1. Prijava in vrste uporabnikov.....	16
4.1.2. Dodajanje, urejanje in pregled tabel ter zapisov.....	17
4.2. Načrtovanje rešitve	19
4.2.1. Izbira in opis ustreznih metodologij in orodij	19
4.2.2. Entitetno-relacijski diagram	19
4.2.3. Jezik HyperText Markup Language (HTML)	19
4.2.4. Jezik Hypertext Preprocessor (PHP)	20
4.2.5. My Structured Query Language (MySQL)	20

4.2.6. Jezik Extensible Markup Language (XML).....	20
4.2.7. Jezik Cascading Style Sheets (CSS)	20
4.2.8. WampServer razvojno okolje.....	20
4.2.9. Zasnova podatkovne baze	21
4.2.10. Zasnova uporabniškega vmesnika	22
4.3. Opis izdelave.....	23
4.3.1. Modularno programiranje programskih funkcij.....	24
4.3.2. Prijavna stran.....	24
4.3.3. Glavno okno	25
4.3.4. Urejanje uporabnikov aplikacije	26
4.3.5. Dodajanje osnovnih tabel in njihovih funkcij	27
4.3.6. Povezave med tabelami.....	29
4.3.7. Razširitev funkcij na poljubne tabele.....	29
4.3.8. Dodajanje nove tabele v seznam, nastavitve, odstranjevanje	30
4.3.9. Dodajanje zapisa v tabelo	32
4.3.10. Pregledovanje tabel in zapisov.....	32
4.3.11. Beleženje zgodovine sprememb.....	33
4.3.12. Dinamično povezovanje tabel.....	33
4.3.13. Izvoz tabel, iskanje, sortiranje.....	35
4.3.14. Vstavek za izpis predlogov pri vnašanju podatkov.....	36
4.3.15. Urejanje izgleda z jezikom CSS.....	37

5. Namestitvev in vzdrževanje	38
6. Opis uporabe programa	38
7. Sklepne ugotovitve	47
8. Viri.....	48

Kazalo slik

Slika 1: Program Hardware Asset Tracker	12
Slika 2: Program Asset Manager	13
Slika 3: Program Computer Admin Pro	14
Slika 4: Program Total Network Inventory	15
Slika 5: Definicija osnovnih tabel v podatkovni bazi.....	21
Slika 6: Povezave tabel izrisane s Crow's foot notacijo	22
Slika 7: Zasnova uporabniškega vmesnika (glavno okno)	23
Slika 8: koda HTML prijavnne strani.....	24
Slika 9: Struktura tabele UPORABNIKI	25
Slika 10: del kode ob prijavi v glavno okno.....	25
Slika 11: Povezava s podatkovno bazo.....	26
Slika 12: Sortiranje tabel	28
Slika 13: Primer povezovalne tabele	29
Slika 14: Primer povezave med zapisi (vmesna tabela)	29

Slika 15: Obrazec za nastavitve tabele.....	30
Slika 16: Tabela prikazneTabele z nastavitvami prikaznih tabel.....	30
Slika 17: Branje in dešifriranje zapisa pravic	31
Slika 18: Primer beleženja zgodovine.....	33
Slika 19: Dodajanje in izpis povezav med tabelami	34
Slika 20: Poenostavljen diagram kode za izpis tabele s povezavami	35
Slika 21: Izpis predlogov za vnos	36
Slika 22: Del kode funkcije za dodajanje vrstice predlogov.....	36
Slika 23: Funkcija za dodajanje izpisa predlogov.....	37
Slika 24: koda CSS za izpis tabele.....	37
Slika 25: Osnovna stran WampServer-ja	38
Slika 26: Prijavna stran	39
Slika 27: Glavna stran aplikacije	39
Slika 28: Dodajanje računalnika	40
Slika 29: Obrazec za dodajanje računalnika	40
Slika 30: Izpis baze računalnikov	41
Slika 31: Možnosti dodajanja konfiguracije	41
Slika 32: Izpis podrobnosti računalnika.....	42
Slika 33: Obrazec za dodajanje inštalacije.....	43
Slika 34: Dodajanje tabele v seznam	43
Slika 35: Izpis dodanih tabel.....	44

Slika 36: Povezovanje tabel.....	44
Slika 37: Nastavitve povezav	45
Slika 38: Izpis povezav med tabelami	45
Slika 39: Izpis povezane tabele	46
Slika 40: Primer izpisanega poročila v PDF dokument.....	47

Uporabljene kratice

CSS	Cascading Style Sheets – Stilske predloge
CSV	Comma Separated Values – Podatki ločeni z vejico
DOM	Document Object Model – Dokumentni objektni model
ER diagram	Entitetno relacijski diagram
FPDF	Free PDF – Prosti PDF
HTML	HyperText Markup Language – Jezik za označevanje nadbesedila
PDF	Portable Document Format – Prenosni dokumentni format
SQL	Structured Query Language – Strukturirani povpraševalni jezik
XML	Extensible Markup Language – Razširljiv označevalni jezik

Povzetek

Namen diplomske naloge je bil izdelati spletno aplikacijo, s katero bo mogoče enostavno in pregledno shranjevati, pregledovati in obdelovati podatke o računalniški in informacijski opremi v podjetju ter tudi o njihovih uporabnikih, pri čemer bo mogoče tabele med seboj dinamično povezovati. Področje naloge zajema izdelavo in programiranje spletne strani in podatkovne baze na podlagi preučitve podobnih, že obstoječih aplikacij ter podanih zahtev glede na potrebe podjetja ter vrsto informacij, ki jih želimo shranjevati. Pri tem se uporablja razvojno okolje WampServer z Apache strežnikom, kjer se za razvoj spletne aplikacije uporabljata predvsem programska jezika HTML in PHP, ter jezik SQL za delo s podatkovnimi bazami. Po definiciji zahtev se funkcije aplikacije izdelajo na podlagi osnovnih definiranih tabel, nato pa se njihova uporabnost razširi in prilagodi za poljubne tabele v podatkovni bazi, ki bi jih morda želeli uporabljati za shranjevanje podatkov naknadno.

Za izdelavo spletne aplikacije je uporabljena agilna, modularna metoda razvoja, kjer se funkcije dodaja postopoma za lažje programiranje in večjo fleksibilnost. Pri izdelavi podatkovne baze pa se uporabi metodo entitetno-relacijskih diagramov s Crow's foot notacijo pri opisu relacij med tabelami.

Rezultat naloge je osnovna dokončana spletna aplikacija z vsemi zastavljenimi funkcijami za vpis in urejanje uporabnikov ter delo s tabelami, kar vključuje vpisovanje podatkov, urejanje, iskanje, sortiranje, izpisovanje poročil, beleženje zgodovine, izvoz podatkov ter dodajanje tabel in njihovo dinamično povezovanje, pri čemer je možno določati pravice izvajanja funkcij glede na vrsto uporabnika. Predstavlja orodje za preprost način mrežnega dostopa do baze podatkov ter izvajanja funkcij nad njo, pri čemer se uporabnike lahko omeji glede na njihove pravice.

Ob zaključku so bili izvedeni vsi zastavljeni cilji, poleg tega pa se je že med razvojem porodilo tudi več idej za možne izboljšave in nadgradnje, kar je z nekaj dodatnega dela mogoče enostavno realizirati glede na samo zasnovano aplikacijo.

Ključne besede: Spletna-mrežna aplikacija, podatkovne baze, inventar računalniške opreme, dinamično povezovanje tabel.

Summary

The goal of this project was to design an online application through which one will be able to store, browse and work with data about computer and information equipment in a company and also its users, in a way where data tables can also be dynamically connected to each other. The field of this project includes the design and programming of an online application and creation of a database, based on given basic requirements and the prior examination of similar applications, considering the nature of information that will be stored and used. For this the WampServer web development environment is used together with Apache server. The main programming languages for the web application are HTML and PHP, where for the database management SQL language is used. After the basic requirements are clearly defined, application functions are developed based on the basic required data tables. Afterwards the functionality of these functions is expanded and adjusted to work with any table in the database that may be created in a later time according to the emerged needs.

In development of this online application a method of agile software development is used where functions are designed in steps and as modules. In this way the programming is simplified and made more flexible. In development of the database the entity-relationship model is used as defined by Crow's foot notation for description of the relationships between the data tables.

The result of this project is a basic finished online application that includes all the goal functions defined in the beginning, such as user management and tools for working with data tables, which includes entering data, editing, searching, sorting, printing of report, noting change history, exporting of data, adding of new tables and creating dynamic table connections, where the rights for the use of functions can be set according to the category of users. The application represents a simple to use tool for online database access and data table management with possible user access restrictions.

In conclusion all the set goals of the project were achieved and also some new ideas were already created during and after the development for possible improvements and upgrades that could be simply implemented in the future, considering the modular design of the application.

Key words: Online-network application, database, computer equipment inventory, dynamic table connections.

1. Uvod

Računalniška oprema se danes uporablja že na vseh možnih področjih z različnimi cilji in potrebami. Še zmeraj pa je osnovna dejavnost računalnikov shranjevanje in obdelava podatkov in informacij. V podjetjih in drugih ustanovah se rokuje z ogromnim številom letih, ki jih je potrebno učinkovito shranjevati, urejati in pregledovati, da se lahko na njihovi podlagi učinkovito odločamo in zagotavljamo nemoteno delovanje vseh delov ustanove.

Samo shranjevanje podatkov v računalnik je precej enostavno. Težave pa nastopijo ob njihovi obdelavi in rokovanju, saj želimo, da bi to bilo čimbolj enostavno, hitro in učinkovito. Pri tem moramo poiskati odgovore na vprašanja kot so: kako poenotiti vnos informacij, v kakšni obliki naj bodo podatki shranjeni, na kakšen način bomo do njih dostopali, kako podatke smiselno povezati, na kakšen način jih želimo obdelovati, itd. Odgovore na ta vprašanja smo iskali tudi v sklopu te diplomske naloge, ki vključuje reševanje problema shranjevanja in obdelave podatkov o informacijski opremi in njihovih uporabnikih v podjetju, za lažje vodenje evidenc in sledenje spremembam, ki se dogajajo v povezavi z njimi.

Bistvo te diplomske naloge je torej prikazati postopek zasnove in izdelave rešitve problema shranjevanja in obdelave podatkov v zvezi z informacijsko opremo in njihovimi uporabniki v podjetju.

Pri tem smo začeli pri iskanju že obstoječih podobnih rešitev v obliki neplačljivih programov. Na njihovih prednostih in slabostih smo zasnovali svojo, ki bo za naš konkreten primer najbolj primerna in bo zadostovala potrebam podjetja. Pred začetkom izdelave smo natančneje določili zahteve in funkcije aplikacije ter določili formate podatkov, ki jih želimo vnašati in obdelovati, pri čemer smo si pomagali tudi z metodologijo entitetno-relacijskih diagramov. Nato smo izbrali ustrezna orodja in tehnologije, s katerimi bomo učinkovito dosegli cilj, pri čemer smo upoštevali enostavnost izvedbe.

Samo izdelavo naloge smo želeli zastaviti modularno, saj nam to omogoča postopno dodajanje funkcij, česar prednost je delujoča aplikacija tudi v primeru, če vanjo še ne bi mogli integrirati vseh funkcij v časovnem roku diplomske naloge. Med izdelavo smo bili pozorni predvsem na uporabnost in praktičnost aplikacije, temu prilagojen in poenostavljen pa je tudi izgled.

V zaključku naloge je še opis aplikacije in njenih funkcij ter primer uporabe.

2. Opis problema

Izdelati želimo aplikacijo, preko katere bomo lahko na čimbolj enostaven in praktičen način shranjevali in urejali podatke o informacijski opreми v podjetju in o njenih uporabnikih. S tem bi olajšali vodenje inventarja in evidence pozameznih naprav, njihovih stanj in sprememb, s čimer bi dosegli hitrejši dostop do podatkov, večjo natančnost in lažji pregled.

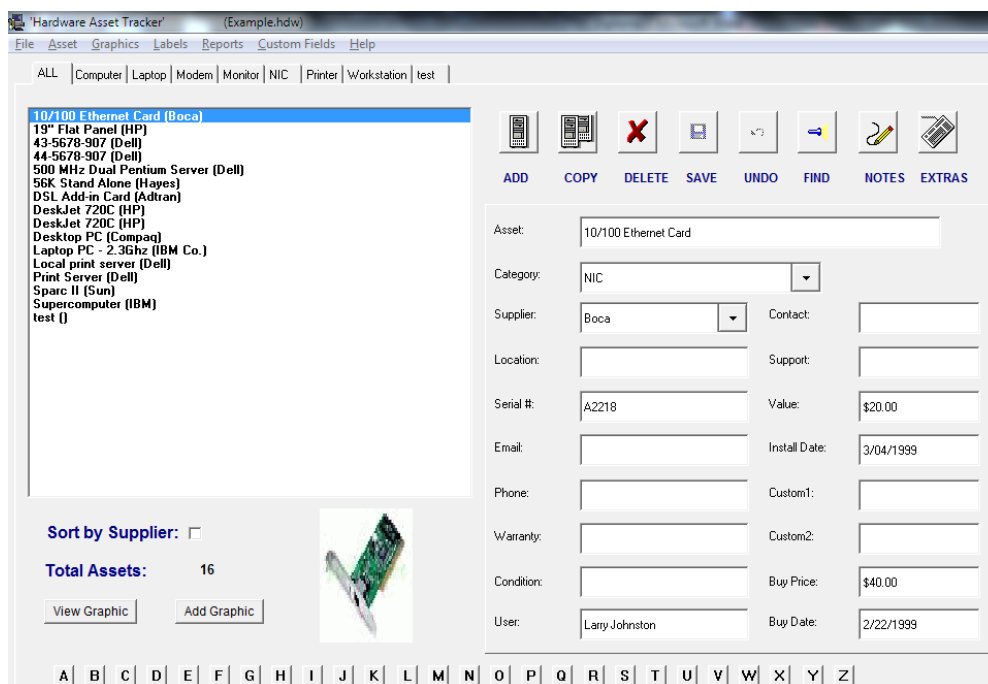
3. Analiza obstoječih rešitev

V času hitrega razvoja računalništva in programske opreme je zelo verjetno, da se je z našim problemom kdo že ukvarjal in zanj izdelal tudi rešitev. Zato smo najprej pregledali splet, kjer smo iskali že izdelane rešitve, predvsem v obliki brezplačnih aplikacij. Med raziskovanjem smo našli več programov, ki se v osnovi ukvarjajo z reševanjem našega problema. Med najdenimi smo podrobneje pregledali naslednje štiri:

- Hardware Asset Tracker (brezplačen) [4]
- Asset Manager (plačljiv) [5]
- Computer Admin Pro 5 (plačljiv) [6]
- Total Network Inventory (plačljiv) [7]

3.1. Program Hardware Asset Tracker

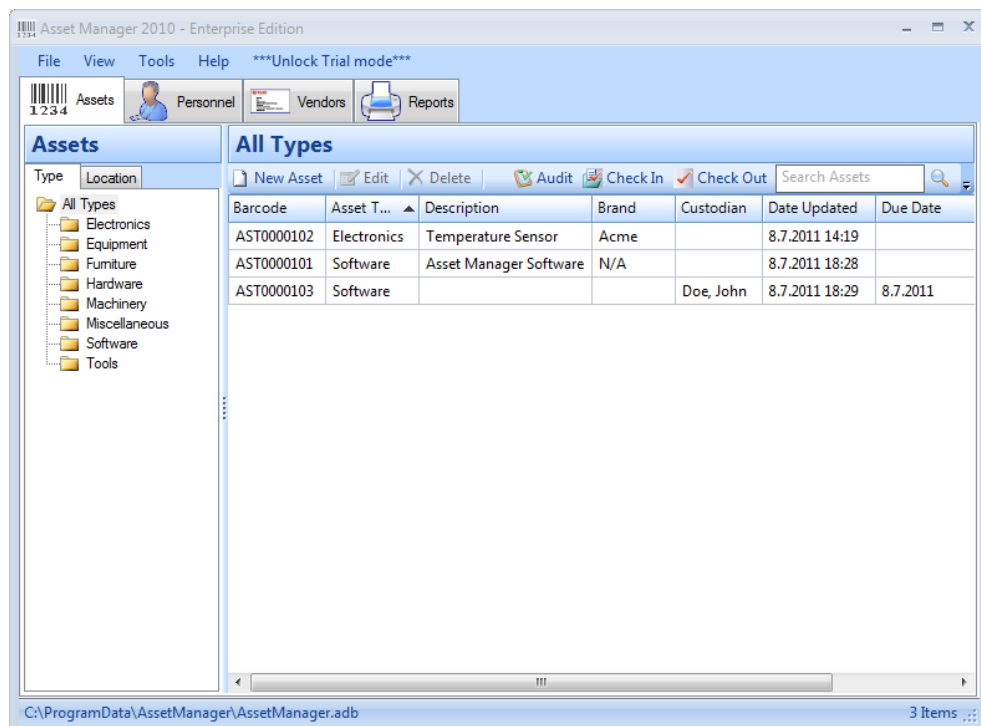
Ta program je izmed naštetih najbolj preprost in edini brezplačen [4]. Je samostojna aplikacija namenjena shranjevanju podatkov o posameznih računalnikih in strojni opreми. Večina funkcij je razvidnih s slike 1. Po želji lahko spremenimo imena polj v šabloni za vnos podatkov, s čimer lahko vnašalni obrazec priredimo na primer tudi za vnašanje oseb, vendar smo pri vsakem obrazcu omejeni na 18 polj. Poleg tega program ne omogoča povezovanja posameznih vnosov med seboj. Za našo rešitev bi želeli tudi več vrst uporabnikov z različnimi pravicami urejanja, kar tu ni mogoče. Program pa omogoča izpis poročil po različnih parametrih v obliki formata prenosnega dokumenta (ang. Portable Document Format - PDF), kar je zelo uporabno. Prav tako omogoča izvoz baze v obliko vejčno ločenih vrednosti (ang. Comma-Separated Values - CSV). Te funkcije bi želeli v naši rešitvi, poleg tega pa bi potrebovali tudi beleženje zgodovine. Zaradi omenjenih pomanjkljivosti omenjeni program za naš primer ni najbolj primeren.



Slika 1: Program Hardware Asset Tracker

3.2. Program Asset Manager

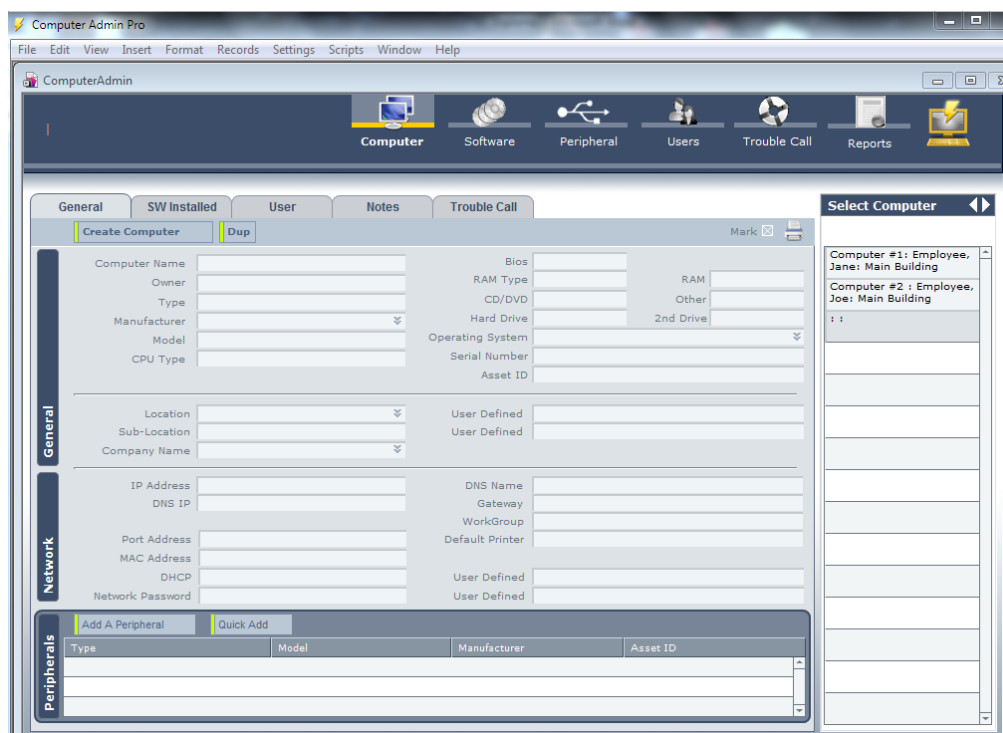
Asset Manager [5] je plačljiva rešitev in je zato tudi bolj dodelan od prej omenjenega programa. V bazo lahko shranjujemo podatke o osebju, prodajalcih in predmetih, pri čemer smo pri ustvarjanju novih šablon (novih kategorij predmetov za vnašanje) omejeni na standardna vnosna polja, ki so tako za vse predmete (kot tudi osebe in prodajalce) že vnaprej določena in enaka, zato lahko vnašamo le nekajosnovnih podatkov. Pri tem posameznih kategorij ne moremo povezovati med seboj. Ostale funkcije med drugim vključujejo izpis poročil, izvoz in uvoz podatkov, beleženje zgodovine, branje in vnašanje črtne kode predmetov in določanje statusa predmetov preko funkcij 'sprejem' in 'predaja'. Izgled je prikazan na spodnji sliki. Program tako vsebuje skoraj vse funkcije, ki bi jih mi želeli, manjkajo pa mu nadzor dostopa do podatkov glede na različno vrsto uporabnikov, povezovanje tabel (kategorij) med seboj in večja svoboda pri ustvarjanju novih obrazcev (kategorij) za vnašanje podatkov. Poleg tega je tudi plačljiv.



Slika 2: Program Asset Manager

3.3. Program Computer Admin Pro 5

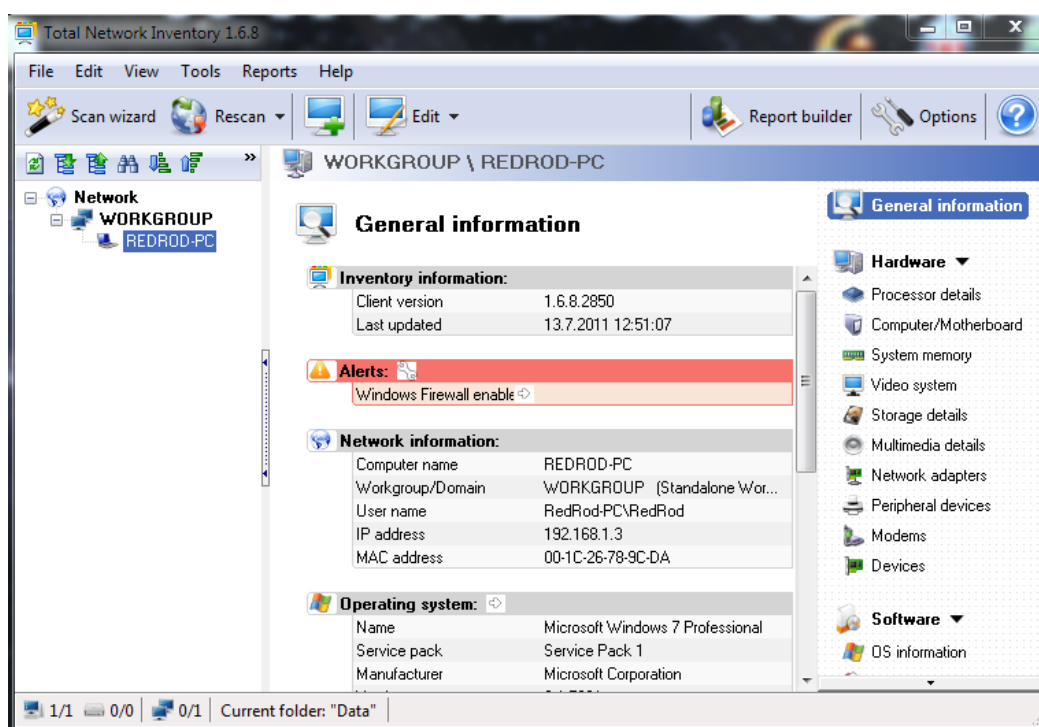
Sledeči program [6] (slika 3) je zasnovan podobno kot Asset Manager, le da je ta še nekoliko bolj dodelan in obsežen, kot želi nakazati že ime z oznako 'Professional'. Posebej lahko vnašamo podatke za računalnike, programsko opremo, periferijo in uporabnike, pri čemer imamo v obrazcih na voljo precej večje število polj, a smo še vedno omejeni na že izdelane obrazce, ki pa so med seboj povezljivi. Program vsebuje tudi nekaj dodatnih funkcij, kot na primer 'Trouble Call', ki predstavlja zahtevo po odpravi težave, ki se je pojavila na določeni periferiji ali računalniku. Pomankljivosti programa v smislu rešitve, ki bi jo želeli za naš problem, pa so podobne zgoraj omenjenim.



Slika 3: Program Computer Admin Pro

3.4. Program Total Network Inventory

Zadnji izmed preizkušenih programov [7] je zasnovan nekoliko drugače od prejšnjih. Deluje preko lokalnega omrežja in omogoča pregled ter urejanje informacij o posameznih napravah, ki so na omrežje priključene. To omogoča hiter in enostaven zajem podatkov o posameznih napravah, pri čemer pa izostanejo vse ostale funkcije, ki jih vsebujejo že omenjeni programi in ki bi si jih želeli v naši rešitvi. Ta program je zato najmanj primeren kot rešitev, kljub temu pa bi si želeli podobne možnosti hitrega dostopa do podatkov o posameznih napravah (predmetih) pri vnašanju v obrazce, kjer je to za našo rešitev smiselno in primerno.



Slika 4: Program Total Network Inventory

4.Zasnova in izdelava lastne rešitve

Po pregledu obstoječih programskih rešitev s spleta smo se odločili za izdelavo svoje aplikacije, saj noben izmed pregledanih programov ni bil popolnoma primerna, poleg tega pa tudi nismo želeli plačljivih rešitev. Na podlagi že videnih funkcij v pregledanih programih, predvsem pa na lastnih zastavljenih zahtevah za aplikacijo, smo sestavili seznam funkcij, ki naj jih naš program vsebuje.

4.1. Definicija zahtev

Program naj bo zasnovan kot spletna aplikacija, da se bo lahko do nje dostopalo s poljubnega računalnika v omrežju, pri čemer naj bo dostop omejen glede na vrsto uporabnika. Omogoča naj vpisovanje, pregled in urejanje podatkov o računalniški in drugi informacijski opremitvi v podjetju, njihovih komponentah, programski opremitvi in uporabnikih. Pri tem naj obstaja možnost beleženja zgodovine sprememb. Aplikacija naj omogoča dodajanje novih tabel glede na potrebe shranjevanja podatkov, pri čemer naj omogoča dinamično povezovanje med njimi, kjer je to smiselno. Omogoča naj izpis tabel in njihovo sortiranje glede na vnosna polja ter iskanje zapisov. Prav tako naj omogoča izpis podatkov v obliki poročil PDF ter v formata XML in CSV.

Uporabniški vmesnik naj bo enostaven za uporabo in zasnovan s poudarkom na preglednosti. Z vidika varnosti naj bo omejen dostop do aplikacije in podatkov s sistemom prijave, kjer so

uporabniki omejeni pri spreminjanju podatkov v podatkovni bazi glede na svojo vlogo. Program naj bo zaščitem proti neželenemu brisanju podatkov tako, da uporabniki lahko spreminjajo le posamezne zapise in ne tabel samih. Pri izpisu poročil in izvozu podatkov naj se ti samodejno shranijo v ustrezno mapo na strežniku. Aplikacija naj se primerno odziva tudi ob več prijavljenih uporabnikih. Aplikacija ni predvidena za delo z ogromnim številom podatkov, zato v našem primeru ni potrebno posvečati posebne pozornosti zagotavljanju hitrega delovanja ob večjih količinah podatkov.

Sledi podroben opis posameznih funkcij programa.

4.1.1. Prijava in vrste uporabnikov

Program bo dostopen komurkoli znotraj omrežja, zato morajo biti posamezni uporabniki omejeni glede na njihove pravice, kar bo imelo vpliv na njihovo stopnjo dovoljenega poseganja v podatkovno bazo. Zato smo se odločili, da uporabnike razdelimo na tri vrste (vloge uporabnikov):

- Administrator
- Uporabnik
- Gost

Administrator:

To je vrsta uporabnikov z najvišjo stopnjo pravic, ki lahko brez omejitev uporabljajo vse funkcije programa. Kot skrbniki aplikacije in podatkovne baze lahko urejajo in dodeljujejo pravice drugih uporabnikov. Prav tako lahko dodajajo v seznam na novo ustvarjene tabele v podatkovni bazi, urejajo povezave med njimi in spreminjajo njihove nastavitve. Imajo tudi pregled nad zgodovino sprememb.

Uporabnik:

Uporabniki niso skrbniki aplikacije in podatkovne baze, zato teh funkcij ne potrebujejo. Na voljo so jim vse osnovne funkcije programa, kot so iskanje, sortiranje, pregledovanje in dodajanje novih zapisov v podatkovno bazo ter urejajo že obstoječih. Ne morejo pa izvažati zapisov in izpisovati poročil, prav tako pa ne morejo brisati že obstoječih zapisov v bazi.

Gost:

To so uporabniki z najmanj pravicami in lahko v osnovi le preglejujejo zapise v podatkovni bazi, kjer jim je to omogočeno.

Če želimo dostopati do funkcij aplikacije se moramo tako najprej vanjo prijaviti s predhodno dodeljenim uporabniškim imenom in geslom, s čimer dobimo ustrezne pravice. Brez prijave

dostop ni mogoč. Administratorji imajo za to na voljo funkcijo za dodajanje in urejanje uporabnikov aplikacije.

4.1.2. Dodajanje, urejanje in pregled tabel ter zapisov

Osrednji del aplikacije sta dodajanje in urejanje tabel in zapisov v podatkovno bazo ter njihovo medsebojno povezovanje. Dodajanje tabele in urejanje povezav je na voljo le administratorjem.

Dodajanje tabele v seznam:

Pri dodajanju nove tabele v seznam je to potrebno najprej ustvariti v podatkovni bazi. Nato se tabelo doda v seznam k ostalim tabelam v aplikacijo, pri čemer lahko administrator nastavi pravice za tabelo glede na uporabnika, s čimer določi vidnost tabele posamezni vrsti uporabnikov oziroma njihovo stopnjo dovoljenega posega v zapise v podatkovni bazi za dodano tabelo.

Dinamično povezovanje tabel:

Ta funkcija administratorju omogoča, da dve tabeli poveže med seboj, pri čemer aplikacija ustvari vmesno tabelo v podatkovni bazi. Tako lahko nekemu zapisu iz prve tabele določimo enega ali več zapisov iz druge tabele, odvisno od nastavljene relacije med izbranimi tabelama. Ob nastavitvi povezave se izbere tudi prikazno polje za obe tabeli. To polje se prikaže ob že obstoječih poljih pri izpisu ene izmed povezanih tabel in služi kot referenca do zapisa v drugi povezani tabeli. Tako lažje in hitreje razberemo, s katerim zapisom (ali več njih) iz druge tabele je povezan nek zapis trenutno izpisane tabele.

Izpis tabele:

Ob kliku na ustrezno tabelo, ki je bila pred tem dodana v seznam, se ta tabela izpiše skupaj z vsemi funkcijami, ki jih lahko izvajamo nad tabelo. Če je tabela povezana s katero drugo tabelo, se poleg stolpcev omenjene tabele izpiše še prikazni stolpec iz povezane tabele (ali več njih). Ob vsaki izpisani vrstici izpisane tabele se tako izpiše še vrednost prikaznega polja dodeljenega zapisa iz druge tabele. Če zapisu iz trenutne tabele še ni dodeljen noben zapis iz povezane tabele je to posebej poudarjeno.

Dodajanje zapisov v tabelo:

Ob izpisu posamezne tabele iz podatkovne baze se pojavi možnost za dodajanje novega zapisa v tabelo, ustrezno s pravicami uporabnika. Pri dodajanju se izpiše obrazec za vnos podatkov, pri čemer moramo vpisati vsaj obvezna polja. S potrditvijo dodamo nov zapis v ustrezno tabelo.

Urejanje zapisov tabele:

Ob vsakem izpisanem zapisu iz tabele se pojavi gumb za urejanje, skladno s pravicami trenutnega uporabnika. Ob kliku nanj se odpre enak obrazec kot pri dodajanju novega zapisa, le da so tokrat že vnešeni ustrezni podatki, ki jih lahko spreminjamo in popravljamo. Ob spremembi podatkov se to zabeleži v zgodovino tabele v kolikor je ta funkcija vključena.

Brisanje zapisov iz tabele:

Tako kot pri urejanju zapisov se poleg vsakega zapisa tabele pojavi gumb za izbris zapisa iz tabele in podatkovne baze. Če je omenjeni zapis povezan še s katerim drugim preko povezave tabel, nas aplikacija o tem še enkrat obvesti pred izbrisanjem.

Izpis podrobnosti posameznega zapisa tabele:

Če je izpisana tabela povezana še s katero drugo, se ob posameznem zapisu tabele pojavi možnost za izpis podrobnosti. Ob kliku na podrobnosti se izpišejo vsa polja omenjenega zapisa, prav tako pa vsa polja vseh dodeljenih zapisov iz povezanih tabel. Izpišejo se tudi ustrezne funkcije za dodajanje in odstranjevanje dodeljenih zapisov iz povezanih tabel ter za izpis njihovih podrobnosti.

Sortiranje izpisa tabele:

Izpis tabele lahko sortiramo po abecedi oziroma od najmanjše do največje vrednosti glede na katerikoli izbrani stolpec v tabeli, razen tistih, ki predstavljajo referenčna prikazna polja iz drugih povezanih tabel.

Iskanje po tabeli:

Po vsaki tabeli lahko iščemo zapise s funkcijo iskanja po geslih.

Izvoz tabele:

Administrator lahko zapise vsake tabele izvozi v obliko XML ali CSV. Prav tako lahko za vsako tabelo izpiše poročilo v PDF obliki, kjer lahko izbere še izpis zgodovine, če je ta funkcija vklopljena za izbrano tabelo.

Izpis zgodovine tabele:

Pri dodajanju nove tabele v seznam lahko administrator vključi beleženje zgodovine za tabelo. V tem primer se vsakič, ko administrator izpiše takšno tabelo, pod njo izpiše še zgodovina tabele. Zgodovina ni vidna drugim vrstam uporabnikov. Ob izpisu poročila lahko administrator po želji doda še izpis zgodovine sprememb za izbrano tabelo.

4.2. Načrtovanje rešitve

Pred začetkom izdelave rešitve je potrebno izbrati ustrezna orodja s katerimi bomo najlažje in učinkovito prišli do rešitve. Ob tem so prav tako pomembne metodologije s katerimi si pomagamo osnovati in udejaniti naše cilje tako, da bomo uspešno zadostili vsem zahtevam.

4.2.1. Izbira in opis ustreznih metodologij in orodij

Glede na zahteve in željo po čimbolj enostavni izvedbi smo se odločili za realizacijo aplikacije z uporabo spletnih jezikov HTML in PHP v povezavi z jezikom za urejanje podatkovnih baz: SQL oziroma MySQL [2]. Ker bo aplikacija delovala preko spleta, potrebujemo še primerno okolje za izdelavo, ki omogoča navidezni strežnik. Odločili smo se za uporabo razvojnega okolja WampServer [8], saj omogoča enostavno ustvarjanje spletnih aplikacij in že vključuje Apache server [14], jezik PHP in MySQL podatkovne baze. Prav tako nudi vmsenik za lažje urejanje podatkovnih baz. V naslednjih poglavjih so opisana omenjena orodja in metodologije, ki smo jih uporabili.

Izvedbe se bomo lotili z agilnimi metodami razvoja [9]. Izbrana orodja omogočajo modularno izdelavo programa, saj bo aplikacija zasnovana kot spletna stran, kjer bomo lahko postopno in po potrebi dodajali željene funkcije. To omogoča lažjo izvedbo in programiranje, prav tako pa omogoča enostavno nadgradnjo in dodajanje novih funkcij.

4.2.2. Entitetno-relacijski diagram

Entitetno-relacijski (krajše ER) diagram [3] ali model je orodje za abstrakten prikaz informacij in njihovih medsebojnih relacij ali povezav. Predvsem je uporaben za grafični prikaz modela podatkovne baze. Z njim lahko prikažemo potrebe informacijskega sistema in določimo vrsto informacij in način, kako so te povezane znotraj podatkovne baze. Poznamo več načinov izrisa entitet in določanja kardinalnosti, v naših diagramih pa bomo uporabili Crow's foot notacijo, kjer entitete izrišemo kot pravokotnike, relacije pa kot črte, ki jih povezujejo.

4.2.3. Jezik HyperText Markup Language (HTML)

Jezik HTML [1, 10] je najpogosteje uporabljan jezik za definiranje izgleda in sestave spletnih strani. Sestavljen je iz oznak, ki so zapisane med znaka '<' in '>', te pa so vstavljene med vsebino spletne strani, kar uporabniku omogoča definicijo izgleda, vstavljanje slik, tabel in podobnih elementov. Spletni brskalniki to kodo interpretirajo in stran prikažejo v željeni obliki. Med kodo HTML lahko vstavljamo tudi kodo drugih jezikov in s tem še natančneje definiramo izgled ali nanjo dodamo nove interaktivne elemente.

Jezik HTML bo služil za osnovo in izgled spletne strani.

4.2.4. Jezik Hypertext Preprocessor (PHP)

Jezik PHP [2, 11] je splošen jezik namenjen predvsem delovanju na serverjih. V osnovi je namenjen izdelavi dinamičnih spletnih strani, njegovo kodo pa vnašamo med kodo HTML dokumenta, ki ga interpretira spletni strežnik, kjer se generiral dokument spletne strani. Uporablja se ga lahko brezplačno na večina spletnih strežnikov in operacijskih sistemov, zato je zelo priljubljen.

Uporabljali ga bomo za izdelavo prijavnih sej, obrazcev za vnašanje podatkov in še pri nekaterih drugih interaktivnih funkcijah spletne strani.

4.2.5. My Structured Query Language (MySQL)

MySQL [2] je relacijski sistem urejanja podatkovnih baz in deluje kot strežnik, ki zagotavlja dostop do podatkovnih baz večim uporabnikom. Potrebujemo ga za osnivanje in urejanje naše podatkovne baze, kar predstavlja osrednji del aplikacije, saj bi brez tega težko shranjevali večje količine podatkov. Urejanje podatkov poteka preko programskega jezika SQL, s katerim podajamo ukaze in poizvedbe nad podatkovno bazo in posameznimi zapisi v njej.

4.2.6. Jezik Extensible Markup Language (XML)

Jezik XML [12] predstavlja skupek pravil za kodiran zapis podatkov in dokumentov v obliki, ki je računalniško enostavno berljiva. Za zapis se uporablja tekstovni format. Z njegovo pomočjo lahko definiramo drevesne podatkovne strukture z različnimi atributi, kar omogoča urejanje zapisov podatkov, do katerih lahko enostavno in hitro dostopamo.

Uporabili ga bomo predvsem kot možni format za izvoz podatkov iz podatkovne baze.

4.2.7. Jezik Cascading Style Sheets (CSS)

Jezik CSS [1] je stilski jezik za opis predstavitvene semantike oz. izgleda dokumenta zapisanega v jeziku kot je HTML. Namenjen je predvsem ločevanju vsebine dokumenta od definicije njenega izgleda, kar poveča pregled in dostopnost ter omogoča lažje delo, saj izgled urejamo na enem mestu, prav tako pa si lahko več strani deli enak format izgleda.

4.2.8. WampServer razvojno okolje

WampServer [8] je razvojno okolje namenjeno izdelavi spletnih strani in aplikacij. Narejen je tako, da povezuje program Apache za simulacijo spletnega strežnika, vstavek PHP za izdelavo in testiranje spletnih strani z omenjeno kodo ter prav tako vstavek MySQL za delo s podatkovnimi bazami. Ker je brezplačen predstavlja idealno platformo za razvoj spletne aplikacije. Vsebuje tudi uporabniški vmesnik, ki celotno uporabo zelo poenostavi.

Paket WampServer smo najprej prenesli na računalnik z uradne spletne strani in ga namestili s čarovnikom za nameščanje. Ob zagonu se nam v vrstici z ikonami za obveščanje pojavi še ikona WampServerja, od koder enostavno dostopamo do funkcij. S klikom nanjo se nam odpre seznam od koder zaženemo vse procese strežnika. Za uspešno delovanje slednjega morajo biti prosta vrata 80, s čimer smo imeli na začetku nekaj težav, saj je ta vrata že zasedal program Sykpe. Po odpravi težave smo lahko pričeli z uporabo programa, tako da smo odprli brskalnik in v vrstico za naslov vpisali ukaz za dostop do lokalnega strežnika (ang. localhost). S tem smo dobili dostop do vseh funkcij razvojnega okolja WampServer, od koder smo se najprej lotili izdelave podatkovne baze.

4.2.9. Zasnova podatkovne baze

Pred začetkom izdelave podatkovne baze je potrebno definirati podatkovne strukture in njihove relacije, glede na potrebe zajemanja podatkov. Zato smo se najprej osredotočili na osnivanje ustreznih tabel in definicijo podatkovnih polj, ki jih bodo te vsebovale. Aplikacija bo namenjena predvsem za shranjevanje podatkov o informacijski opremi in njenih uporabnikih, zato smo najprej zasnovali naslednje tabele:

- OSEBE
- RAČUNALNIK
- KONFIGURACIJA
- INŠTALACIJA

Natančna definicija je razvidna s spodnjega slike.

Osebe			Računalnik			Konfiguracija			Inštalacija		
PK	id	INTEGER	PK	id	INTEGER	PK	id	CHAR(10)	PK	id	INTEGER
	ime	CHAR(15)		ime	CHAR(15)		oznaka	CHAR(20)		oznaka	VARCHAR(20)
	priimek	CHAR(20)		inventarna	CHAR(13)		procesor	CHAR(20)		OSkoda	CHAR(30)
	oddelek	CHAR(4)		datumVnosa	DATETIME		hitrostProcesorja	CHAR(10)		verzija	CHAR(10)
	funkcija	TEXT(50)		datumDobave	DATETIME		ram	CHAR(10)		32bit/64bit	CHAR(7)
	uporabniškoIme	CHAR(10)		dobavitelj	CHAR(10)		diski	CHAR(100)		servicePack	CHAR(4)
	geslo	CHAR(10)		garancija	INTEGER		raidPolje	BIT		Slnastavitve	BIT
	imeZaTiskalnik	CHAR(10)		številkaRačuna	CHAR(15)		kapacitetaDiskov	CHAR(30)		windowsUpdate	CHAR(10)
	gesloZaTiskalnik	CHAR(4)		opombe	TEXT(10)		grafičnaKartica	CHAR(25)		lokalnoAdminGeslo	BIT
	predalZaSken	CHAR(2)		zgodovina	LONGTEXT					prijavaVdomeno	BIT
	ePoštniNaslov	TEXT(50)								uporabnikKotAdmin	BIT
	zgodovina	LONGTEXT								AVzaščitaPošte	BIT
										officeVerzija	CHAR(16)
										officeKoda	CHAR(30)
										officeVerzija	CHAR(10)
										združljivostniPaket	BIT
										formatiZaMSO	BIT
										tiskalnikPDF	CHAR(11)
										tiskalnik	BIT
										servisnaOrodja	BIT
										programi	LONGTEXT

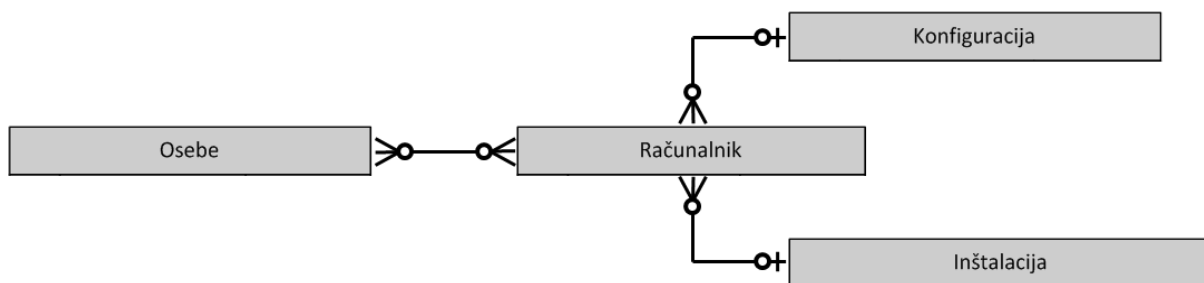
Slika 5: Definicija osnovnih tabel v podatkovni bazi

V tabeli OSEBE bomo shranjevali podatke o uporabnikih informacijske opreme – o zaposlenih. V naslednji tabeli RAČUNALNIK shranjujemo podatke o računalnikih, v tabeli KONFIGURACIJA pa bodo shranjene strojne konfiguracije računalnikov. Podobno bodo v

tabeli INŠTALACIJA shranjeni podatki o programski opremi, ki jo posamezni računalniki vsebujejo. V podatkovni bazi moramo omenjene tabele ustrezno povezati, pri čemer mora veljati:

Oseba lahko uporablja več računalnikov ali pa nobenega in obratno.
 Računalniku lahko dodelimo eno konfiguracijo ali pa nobene.
 Konfiguracija je lahko dodeljena več računalnikom ali pa nobenemu.
 Računalnik ima lahko dodeljeno eno inštalacijo ali pa nobene.
 Inštalacija je lahko dodeljena večim računalnikom ali pa nobenemu.

Za lažjo predstavo smo tabele izrisali še z ER diagramom in jih povezali po pravilih Crow's foot notacije, kot kaže spodnja slika.



Slika 6: Povezave tabel izrisane s Crow's foot notacijo

Aplikacija bo poleg zgornjih osnovnih tabel shranjevala tudi podatke o uporabnikih aplikacije, kjer se bodo shranjevala njihova uporabniška imena in gesla. Za ta namen smo v bazi ustvarili tabelo uporabniki, ki vsebuje naslednja polja:

- ID
- Uporabnik
- Geslo
- Vrsta

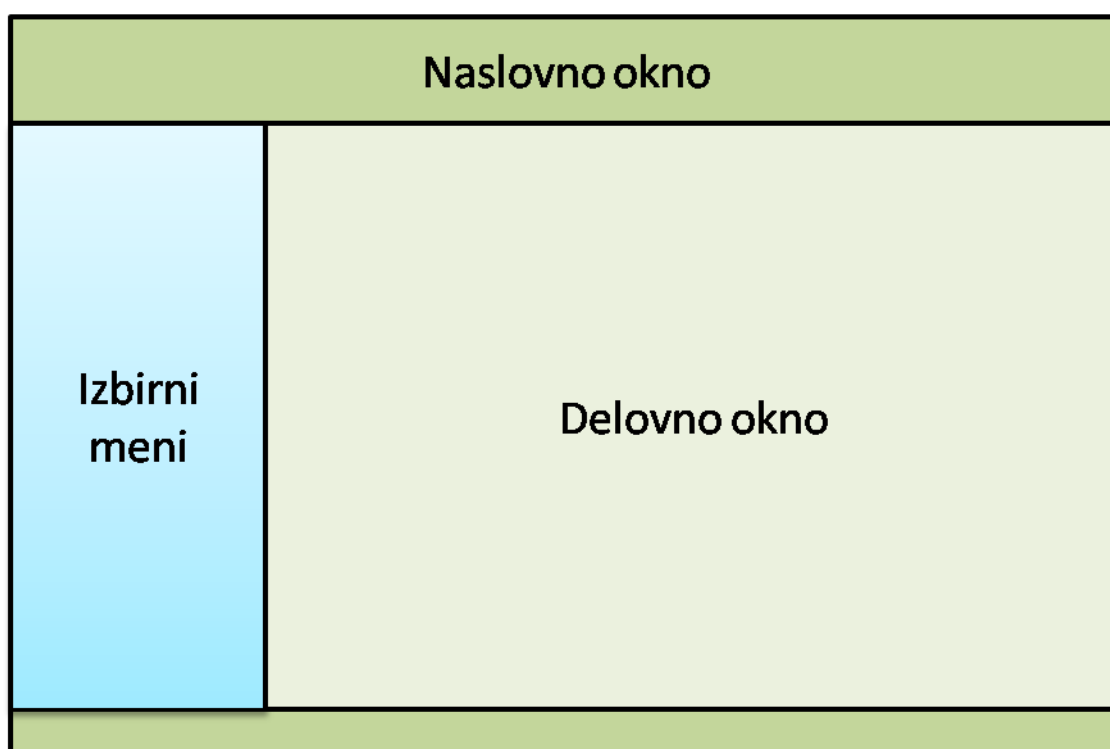
To je osnovna definicija podatkovne baze, ki jo bomo ustvarili s prvo različico programa. S tem bomo izdelali ogrodje z osnovnimi funkcija, katerega lahko po potrebi popravimo in dodelamo glede na potrebe. Nato pa bomo dodajali še ostale funkcije.

4.2.10. Zasnova uporabniškega vmesnika

Za izvedbo uporabniškega vmesnika bomo uporabili jezik HTML skupaj z jezikom PHP, ki omogočata izdelavo spletnih strani in aplikacij. Izgled bo tako prilagojen formatu spletnih strani, saj bomo do aplikacije dostopali preko spletnega brskalnika. Osnovno vodilo pri izdelavi bosta praktičnost in preglednost, saj želimo do podatkov hitro dostopati in z njimi ravnati na urejen in pregleden način.

Uporabnik bo ob vsaki prvi interakciji z aplikacijo naletel na prijavno stran, kjer mora vpisati svoje uporabniško ime in geslo. Z uspešno prijavo mu bo omogočen dostop do glavne spletne strani, od koder bo dostopal do vseh ustreznih funkcij.

Najbolj pomembne (osnovne) funkcije programa morajo biti uporabniku vedno na voljo in zato vedno prikazane, da uporabnik pri njihovi pogosti uporabi ni oviran. Med te funkcije spada odjava in prikaz podatkov o prijavljenem uporabniku, prikaz tabel v podatkovni bazi do katerih lahko kot uporabnik dostopamo in posebne administratorske funkcije, ki bodo vidne le administratorjem. Ob izbiri posamezne funkcije se bodo ustrezne informacije prikazale v temu namenjenem delovnem oknu, kjer bomo izbrane funkcije dejansko opravljali. Spodnja slika prikazuje ogrodje uporabniškega vmesnika.



Slika 7: Zasnova uporabniškega vmesnika (glavno okno)

4.3. Opis izdelave

Aplikacijo smo programirali postopno in ji dodajali funkcije v smiselnem zaporedju tako, da smo slednje programirali v posamezne datoteke oziroma spletne strani, ki so povezane preko osnovne spletne strani. Ta je po strukturi in izgledu enaka zgornji sliki in predstavlja osnovni uporabniški vmesnik. Izgled spletnih strani smo urejali z jezikom HTML, za ostale funkcije pa smo poskrbeli z vstavki kode PHP, pri čemer smo za urejanje kode uporabljali program Notepad++ [13]. Podatkovno bazo smo ustvarjali preko uporabniškega vmesnika WampServer-ja z uporabo funkcije phpMyAdmin [8]. To orodje zelo poenostavi kreiranje

novih tabel in določanje ostalih parametrov podatkovne baze, ker temelji na vizualnem uporabniškem vmesniku in ne zahteva tipkanja ukazov.

4.3.1. Modularno programiranje programskih funkcij

Pred vsakim dostopom se moramo v aplikacijo prijaviti, zato smo najprej začeli z izdelavo prijavnne strani, čemur je sledila izdelava glavnega okna. Ta bo prikazan ves čas uporabe aplikacije, razen ob odjavi, ko se zopet vrnemo na prijavno stran.

4.3.2. Prijavna stran

Prijavna stran se mora prva prikazati ob dostopu preko brskalnika, zato smo najprej ustvarili datoteko `index.html`, ki uporabnika takoj preusmeri na prijavno stran. Tu se prikaže preprosto prijavno okno za vpis uporabniškega imena in gesla. Na sliki 8 je prikazana koda za HTML del.

```
<html>
<head>
  <title>IT inventar prijava</title>
</head>
<body bgcolor='#DAF5FE' onLoad="document.getElementById('username').focus()">
<h1 align="center">IT inventar</h1>
<table width="300" border="0" align="center" cellpadding="0" cellspacing="1" bgcolor="#CCCCCC">
  <tr>
    <form name="prijava" method="post" action="login.php">
      <td>
        <table width="100%" border="0" cellpadding="3" cellspacing="1" bgcolor="#F2FAFF">
          <tr>
            <td colspan="3"><strong>Prijava uporabnika</strong></td>
          </tr><tr>
            <td width="78">Uporabnik</td>
            <td width="6">:</td>
            <td width="294"><input name="username" type="text" id="username" maxLength="20" size="20"></td>
          </tr><tr>
            <td>Geslo</td>
            <td>:</td>
            <td><input name="password" type="password" id="password" maxLength="20" size="20"></td>
          </tr><tr>
            <td>&nbsp;</td> <!--non breaking space-->
            <td>&nbsp;</td>
            <td align="right"><input type="submit" name="submit" value="Prijava"></td>
          </tr>
        </table>
      </td>
    </form>
  </tr>
</table>
</body>
</html>
```

Slika 8: koda HTML prijavnne strani

Elementi so na stran umeščeni z uporabo tabele (oznaka `table`), ki stran razdeli na polja. Z oznakami `form` ter `input` določimo polja obrazca za vnos podatkov in kam se ti pošljejo. V določitev telesa dokumenta (`body`) smo vstavili še vrstico kode JavaScript, ki samodejno nastavi fokus kurzorja na prvo vnosno polje z imenom `username`, kar olajša prijavo, saj lahko takoj po odprtju strani začnemo vnašati uporabniško ime. Ob kliku na gumb `prijava` se podatki iz vnosnih polj pošljejo v dokument s kodo PHP, kjer preverimo njihovo veljavnost. Če se vpisano uporabniško ime in geslo ujemata s tistima v podatkovni bazi, uporabniku omogočimo dostop do aplikacije glede na njegovo funkcijo oziroma vrsto. To dosežemo s PHP ukazom `session_start()`, ki ustvari novo sejo s podatki uporabnika.

Tako ob dostopu do glavne strani preverimo, če je bila seja ustvarjena in omogočimo ustrezne funkcije. V nasprotnem primeru uporabnika vrnemo na stran za prijavo.

Za delovanje prijave smo morali ustvariti tabelo v podatkovni bazi, kjer bomo hranili naslednje podatke o uporabnikih:

Field	Type	Collation	Attributes	Null	Default	Extra
<u>id</u>	tinyint(4)			No	None	AUTO_INCREMENT
uporabnik	varchar(20)	utf32_slovenian_ci		No	None	
geslo	varchar(20)	utf32_slovenian_ci		No	None	
vrsta	int(1)			No	None	

Slika 9: Struktura tabele UPORABNIKI

Glede na to, da aplikacija še ne vsebuje funkcije za dodajanje uporabnikov, ročno vstavimo podatke v tabelo za testno prijavo preko orodja phpMyAdmin, s katerim smo tabelo tudi ustvarili.

4.3.3. Glavno okno

V glavno okno smo preusmerjeni ob uspešni potrditvi vnešenih podatkov za prijavo, pri čemer se je ustvarila nova seja. Zato najprej preverimo veljavnost seje. Kodo za to smo zapisali v svojo datoteko in jo dodali na začetek datoteke glavnega okna (slika 10, vrstica 9) z ukazom *require*. Tako lahko z enakim ukazom preverimo veljavnost seje kjerkoli.

```

9     require("checklogin.php");
10
11     $vrsta = $_SESSION['vrsta'];
12
13     switch($vrsta) {
14         case 1: $vloga = "Administrator"; break;
15         case 2: $vloga = "Uporabnik"; break;
16         case 3: $vloga = "Gost"; break;
17         default: $vloga = "<font color='red'>NEZNANA</font>"; break;
18     };
19
20     if (!isset($_GET['stran'])) {
21         $_GET['stran'] = "undefined";
22     }
23
24     $stran = $_GET['stran']; //dobimo ime strani za prikaz
25
26     switch($stran) {
27
28         case "editupor":
29             $file = "editupor.php";
30             break;

```

Slika 10: del kode ob prijavi v glavno okno

Ob veljavni seji preverimo vlogo vpisanega uporabnika (slika 10, vrstice 11-18). Slednja je v tabeli zapisana s številom 1, 2 ali 3. Po tem se ravnamo, ko izpisujemo ustrezne funkcije v izbirnem meniju (glej sliko 7) in drugje, kjer so te omejene glede na vrsto uporabnika.

Nato preverimo, kaj je potrebno izpisati v delovnem oknu (slika 10, vrstica 24). Ob prvi prijavi to še ni določeno, zato bomo izpisali privzeto stran. Do sedaj še nismo ničesar dejansko izpisali na zaslon, saj smo morali najprej pridobiti ustrezne informacije.

V nadaljevanju dokumenta glavnega okna se prepletata kodi HTML in PHP, kjer izpišemo stran glede na zasnovo uporabniškega vmesnika. Izpis strani po razdelkih smo dosegli tako, da smo, podobno kot na strani za prijavo, slednjo razdelili na polja z uporabo tabele. Tako smo dosegli, da se denimo polje *izbirni meni* ne spreminja ob izbiri ene izmed v njem prikazanih funkcij, spremni se le vsebina delovnega okna.

Na vrhu izbirnega menija smo izpisali uporabniško ime in vlogo uporabnika poleg povezave za odjavo. Ob kliku nanjo se seja zaključí. Pod tem bomo izpisali seznam tabel, pod njimi pa še administratorske funkcije, ki so vidne le administratorjem. Ena izmed teh je urejanje uporabnikov aplikacije, ki jo želimo dodati najprej.

4.3.4. Urejanje uporabnikov aplikacije

Tabelo s podatki o uporabnikih smo že ustvarili za potrebe testne prijave. Sedaj bomo med administratorske funkcije dodali povezavo *Uporabniki*, s klikom na katero bomo v delovnem oknu izpisali omenjeno tabelo uporabnikov. Poleg nje bomo dodali gumb za njihovo dodajanje, prav tako bomo poleg vsake izpisane vrstice tabele oziroma vsakega uporabnika dodali gumba za njegovo brisanje in urejanje. Kodo za vse naštetu bomo zapisali v novo datoteko (modul), tako kot za vse ostale funkcije.

Za vsako funkcijo, ki se izpiše v delovnem oknu, najprej na vrhu le-tega izpišemo njeno ime in ločilno črto za boljši pregled. Nato se s kodo PHP povežemo v podatkovno bazo (koda na spodnji sliki) in izberemo željene podatke.

```

16     require("DBinfo.php");
17
18     //povezava s podatkovno bazo
19     $db = mysql_connect("$dbHost", "$dbUser", "$dbPass") or die ("Error connecting to database.");
20     mysql_select_db("$dbDatabase", $db) or die ("Couldn't select the database.");
21     $result = mysql_query("SELECT * FROM uporabniki", $db) or die(mysql_error());
22     $numrows = mysql_num_rows($result);

```

Slika 11: Povezava s podatkovno bazo

Podatke o bazi smo zapisali v svojo datoteko, da nam jih ob njihovi spremembi ni potrebno spreminjati za vsak dostop posebej. Popravimo le datoteko *DBinfo*. S kodo, ki sledi zgoraj prikazani, izpišemo ustrezne podatke in funkcije.

Nato moramo ustvariti še datoteke za vsako funkcijo, ki jo lahko izberemo. Pri kliku na `Nov uporabnik` se nam bo prikazala stran z obrazcem za vnos ustreznih podatkov. Ob kliku na `uredi` podobno, le da bodo podatki o izbranem uporabniku že vpisani. Pri kliku na `izbriši` pa se bo zapis izbrisal iz tabele.

4.3.5. Dodajanje osnovnih tabel in njihovih funkcij

V naslednjem koraku bomo ustvarili osnovne tabele (opisane v poglavju *zasnova podatkovne baze*), ki jih želimo uporabljati za shranjevanje podatkov. Na njihovi podlagi bomo razvili vse željene funkcije in jih kasneje prilagodili poljubnim tabelam, ki bi jih želeli dodati v bazo naknadno.

Po kreaciji osnovnih štirih tabel v podatkovni bazi z orodjem `phpMyAdmin` dodamo njihove povezave v izbirni meni. Ob kliku na eno izmed njih želimo izpisati ustrezno tabelo s podatki v delovnem oknu, skupaj z naslednjimi funkcijami:

- Dodajanje zapisa v tabelo
- Sortiranje tabele po izbranem stolpcu
- Iskanje zapisa
- Izvoz tabele v eno izmed formatov (PDF, CSV, XML)

Dodajanje zapisa v tabelo

Za vsako tabelo smo ustvarili svojo datoteko za dodajanje, ki izpiše ustrezen obrazec za vnos novega zapisa. Tako smo lahko popolnoma prilagodili obrazec za vsako tabelo, kar omogoča optimalno postavitve vnašalnih polj, a je precej zamudno. Prav tako je koda primerna le za specifičen obrazec.

Pri dodajanju novega zapisa v konfiguracijo ali inštalacijo smo dodali možnost dodajanja po šabloni. Ob gumbu za nov vnos je polje, v katerega lahko vnesemo ID številko že obstoječega zapisa, pri čemer se nam bodo v obrazec že vnesli podatki tega zapisa.

Za dodajanje nove konfiguracije smo dodali možnost uvoza podatkov iz datoteke XML, ki jo pred tem ustvarimo s programom `Speccy` [15]. To je program, ki ga zaženemo na poljubnem računalniku, nakar ta pridobi različne podatke o njegovi opremi in stanju. To lahko shranimo v obliko XML dokumenta. S tem si lahko zelo olajšamo vnos podatkov o konfiguraciji računalnika. Ob izbiri možnosti za vnos konfiguracije iz XML datoteke se nam prikaže okno za njen izbor. S potrditvijo se datoteka naloži v mapo `uploads` na serverju, nakar se iz nje preberejo potrebni podatki. To smo dosegli z uporabo dokumentnega objektnega modela (ang. `Document Object Model - DOM`) [16], s katerim ustvarimo dokument z drevesno strukturo objektov, po katerih se lahko nato pomikamo in beremo željene vrednosti. Nato se izpiše obrazec za vnos nove konfiguracije, kjer so vsa polja že ustrezno izpolnjena. S potrditvijo zapis dodamo, po želji pa lahko XML dokument pustimo v mapi na serverju ali pa potrdimo njegov izbris.

Sortiranje tabele po izbranem stolpcu

Nad vsako izpisano tabelo se prikaže funkcija za sortiranje po stolpcih. Ob kliku na gumb se tabela ponovno izpiše, le da so tokrat zapisi urejeni po abecedi glede na izbrano stolpec. Izvedba funkcije je precej enostavna, saj lahko uporabimo SQL ukaz, ki željene vrednosti predhodno sortira. Del kode je prikazan na spodnji sliki.

```
48 $sortpogoj = "SELECT * FROM racunalniki";
49
50 if(isset($_POST['sortiraj'])) $sortpogoj .= " ORDER BY " . $_POST['pogoj'];
```

Slika 12: Sortiranje tabel

Iskanje zapisa

Iskanje zapisa je nekoliko podobno zgornji funkciji, le da tokrat iščemo po tabeli zapise z vnešenim nizom. Nato izpišemo tabelo vseh zapisov, ki pogoju ustrezajo.

Izvoz tabele

Ta funkcija je dostopna le administratorjem. Ob kliku nanjo se nam odpre stran za vnos imena datoteke, v katero želimo shraniti tabelo. Njeno mesto je predhodno določeno v za to posebej ustvarjeni datoteki *poročila* na strežniku. Tako podatke dodatno zasčitimo, saj lahko do strežnika dostopa le administrator.

Pretvorbi v zapis CSV in XML sta precej podobni. Pri zapisu CSV moramo vrednosti iz vsake vrstice ločiti z vejico. V jeziku PHP smo uporabili ukaz `fputcsv`, ki zapise iz polja ustrezno pretvori in zapiše v datoteko. Zato smo vse vrednosti pred tem še zapisali v polje.

Pri pretvorbi v XML smo vsako vrednost oziroma vrstico ročno opremili z ustreznimi oznakami in vrednosti sproti shranjevali v spremenljivko `$XML`. Na koncu smo niz še zapisali v datoteko.

Pretvorba v PDF dokument je precej bolj zapletena. Zato smo pri ustvarjanju dokumenta uporabili odprtokodni vstavek FPDF [17], ki omogoča kreacijo takšnega dokumenta v jeziku PHP, tako da uporabljamo prilagojene ukaze za postavitev posameznih elementov v dokument. To nam omogoča precejšnjo svobodo pri določanju izgleda.

4.3.6. Povezave med tabelami

Field	Type	Collation	Attributes	Null	Default	Extra
<u>id</u>	int(11)			No	None	AUTO_INCREMENT
idUporabnika	int(11)			No	None	
idRacunalnika	int(11)			No	None	

Slika 13: Primer povezovalne tabele

Tabele so med seboj povezane, kot to prikazuje slika 6. V podatkovni bazi smo morali zato ustvariti vmesne tabele, ki povezujejo med seboj zapise dveh tabel (primer na zgornji sliki). V njih bomo hranili primarna ključa dveh povezanih zapisov iz ustreznih tabel. Tako dosežemo, da je lahko nek zapis iz ene izmed tabel povezan s poljubnim številom zapisov iz druge tabele (slika 14). Povezave vplivajo tudi na obrazce vnašanja novega zapisa, saj moramo na primer pri dodajanju novega računalnika določiti tudi njegovo konfiguracijo, ali pa izbrati brez.

id	idUporabnika	idRacunalnika
98	34	45
99	35	45

Slika 14: Primer povezave med zapisi (vmesna tabela)

Vsak prikazan zapis iz tabele računalniki in osebe vsebuje tudi gumb podrobnosti, s klikom na katerega se nam odpre stran s podrobnostmi izbranega zapisa. To pomeni, da se nam prikažejo vsa polja, ki morda zaradi preglednosti niso bila izpisana v osnovni tabeli, prav tako pa se nam posebej izpišejo tabele z vsemi polji povezanih zapisov. Na primer, če odpremo podrobnosti enega izmed računalnikov, se nam izpišejo še podrobnosti dodeljene konfiguracije in inštalacije.

Vsakemu zapisu iz tabele osebe in računalniki smo dodali beleženje zgodovine. V ta namen smo vsakemu zapisu v obeh tabelah dodali še polje zgodovina. V njega shranimo vsako spremembo, ki je povezana z ustreznim zapisom. Posamezne spremembe smo razdelili po barvah, da jih lažje razberemo. Okno z zgodovino za vsak zapis se prikaže na strani s podrobnostmi. Privzeto se izpiše le pet vrstic, ob kliku na gumb razširi pa se okno poveča za večjo preglednost.

4.3.7. Razširitev funkcij na poljubne tabele

Sedaj smo ustvarili osnovno delujočo aplikacijo, ki vsebuje osnovne funkcije. Tako smo lažje dodelali in preizkusili funkcije, ki jih želimo nad tabelami izvajati, vendar smo zaenkrat še omejeni na predoločeno podatkovno bazo. V nadaljevanju bomo zato dodali funkcije za dodajanje poljubnih tabel v seznam, pri čemer želimo ohraniti tudi vse funkcije, ki jih lahko nad njimi opravljamo.

4.3.8. Dodajanje nove tabele v seznam, nastavitve, odstranjevanje

Najprej smo med administratorske funkcije dodali možnost dodajanja nove tabele v seznam. To storimo tako, da vpišemo ime tabele, ki smo jo pred tem ustvarili in definirali z orodjem phpMyAdmin. Odločili smo se, da funkcije za definicijo nove tabele ne bomo vgradili v aplikacijo, saj je ta že na voljo v orodju phpMyAdmin. Naša rešitev bi bila kvečjemu enaka, verjetno pa manj natančna. V vsakem primeru bodo do nje dostopali le administratorji.

Po vnosu imena tabele se izpiše struktura le-te (če ta obstaja) skupaj s podatki, ki so morebiti vanjo že vnešeni. Tako lažje preverimo, da res vstavljamo željeno tabelo. Nato se nam izpiše obrazec za nastavitve tabele.

Izberi nastavitve:	uporabnik	gost
Tabelo lahko preglejuje:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Zapise lahko dodaja:	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Zapise lahko ureja in briše:	<input type="checkbox"/>	<input type="checkbox"/>
Prikazno ime tabele:	<input type="text" value="Vozila"/>	
Beleži zgodovino tabele:	<input checked="" type="checkbox"/>	
<input type="button" value="Potrdi"/>		

Slika 15: Obrazec za nastavitve tabele

V njej določimo pravice uporabnikov do omenjene tabele, njeno prikazno ime in beleženje zgodovine. Te nastavitve moramo shraniti za vsako tabelo posebej, zato smo v podatkovni bazi ustvarili novo tabelo.

id	imeTabele	prikaznoIme	pravice
73	ekipe	Ekipe	1;0;0; 1;0;0; 1;0;0; 0
74	programi	Programi	1;0;0; 1;0;0; 1;0;0; 0
76	vozniki	vozniki	1;0;0; 1;0;0; 1;0;0; 1
77	vozila	Vozila	1;2;3; 1;2;0; 1;0;0; 1

Slika 16: Tabela prikazneTabele z nastavitvami prikaznih tabel

V njej so shranjeni ustrezni parametri, pri čemer so pravice za posamezno skupino uporabnikov shranjene na naslednji način:

Prvi razdelek treh števil (ločen z navpično črto: ' | ') predstavlja prvo vrstico v obrazcu za nastavitve, pri čemer vsako število predstavlja skupino uporabnikov. Če je število različno od nič, je ta možnost omogočena. Primer za tabelo `vozila` iz tabele na sliki 16:

Prvi razdelek v stolpcu `pravice` v zadnji vrstici (`1;2;3;|`) se nanaša na možnost 'tabelo lahko pregleduje' v obrazcu nastavitvev. Drugi razdelek na drugo in tretji na tretjo možnost. Števila od 1 do 3 predstavljajo administratorje, uporabnike in goste, pri čemer je na prvem mestu vedno 1 (ker so administratorjem vedno omogočene vse možnosti); Na drugem 0 ali 2 ter na tretjem 0 ali 3. Tako isto število v kodi vedno predstavlja isto skupino uporabnikov, spreminja se le kontekst uporabe pri posamezni nastavitvi. Zadnji razdelek vsebuje le število 0 ali 1, kar označuje nastavitve beleženja zgodovine.

S potrditvijo se nastavitve shranijo v tabelo nastavitvev in tabela se doda v seznam tabel v izbirnem meniju.

V kodi aplikacije se moramo na ustreznem mestu (na primer pred izpisom tabele v delovnem oknu) ravnati glede na nastavitve določene tabele. Zato moramo nastavitve prebrati in dešifrirati. To storimo tako, da pravice zapišemo v seznam (`$pravice_array[]`) in jih razdelimo glede na razdelke oziroma posamezno nastavitveno možnost, nakar vsako polje primerjamo s skupino trenutno prijavljenega uporabnika (`$vrsta`). Če se zapisa ujemata, ustrezno funkcijo omogočimo. Nastavitvev za beleženje zgodovine zgolj preberemo. (Primer kode spodaj.)

```

14 if(isset($_POST['prikaziTabele'])) {
15     $imeTabele = $_POST['imeTabele'];
16     $prikaznoImeTabele = $_POST['prikaznoImeTabele'];
17     $idTabele = $_POST['idTabele'];
18
19     require("DBinfo.php");
20
21     $db = mysql_connect("$dbHost", "$dbUser", "$dbPass") or die ("Error connecting
22     mysql_select_db("$dbDatabase", $db) or die ("Couldn't select the database.");
23     $result = mysql_query("SELECT * FROM $imeTabele", $db) or die("Ta tabela ne ob
24     $numfields = mysql_num_fields($result);
25     $numrows = mysql_num_rows($result);
26
27     $result1 = mysql_query("SELECT * FROM prikazneTabele WHERE id=$idTabele", $db)
28     $row1 = mysql_fetch_array($result1);
29     $pravice = $row1['pravice'];
30     $pravice_array[] = explode("|", $pravice);
31
32     $pregled = $pravice_array[0][2]; // pravice za urejanje in brisanje;
33     $pravice_array1[] = explode(";", $pregled);
34     $urediBrisi = FALSE;
35     foreach($pravice_array1[0] as $value) {
36         if(strcmp($value, $vrsta) == 0) $urediBrisi = TRUE;
37     }

```

Slika 17: Branje in dešifriranje zapisa pravic

Zraven gumba vsake tabele iz seznama v izbirnem meniju dodamo še funkcijo za izbris tabele iz seznama ter gumb za dostop do nastavitev. Te dve funkciji sta dostopni le administratorjem.

Pri brisanju tabele iz seznama se izbriše le ustrezen zapis v tabeli `prikazneTabele`. Tako sama tabela ostane v bazi, le da ta ni prikazana v seznamu tabel v izbirnem meniju aplikacije. S tem se tudi zavarujemo pred neželenim izbrisom podatkov, saj lahko tabelo kadarkoli spet dodamo v seznam. Za dokončen izbris pa mora administrator tabelo izbrisati z orodjem `phpMyAdmin`.

4.3.9. Dodajanje zapisa v tabelo

Ob dodajanju novega zapisa se nam prikaže obrazec za vnos podatkov glede na polja iz tabele. Pri osnovnih tabelah smo že vnaprej vedeli, katera polja moramo vnesti, zato smo lahko obrazec prilagodili za vsak specifičen primer. Pri poljubnih tabelah pa ne moremo vnaprej vedeti, katera polja bomo vnašali. Izpis obrazca smo zato zasnovali tako, da se polja izpisujejo eden pod drugim po vrsti, tako kot so navedena v tabeli. Vsa polja smo izpisali v obliki polja za tekstovni vnos (razen povezovalnih), pri čemer se na njihovi levi strani izpiše ime polja, na desni strani pa podatkovni tip in največjo dovoljeno dolžino vnosa, po čemer se ravnamo pri vnašanju podatkov.

Kadar je tabela povezana s katero drugo (ali več njih), se nazadnje izpišejo še ustrezna (prikazna) polja za dodeljevanje zapisa iz druge tabele v obliki spustnega seznama, pri čemer lahko polje pustimo nedefinirano. Ob pridržanju kurzorja na enem izmed možnosti iz spustnega seznama se nam prikaže vrstica s podatki vseh temu zapisu pripadajočih polj. Tako natančno vemo kateri zapis izbiramo.

4.3.10. Pregledovanje tabel in zapisov

Ob kliku na eno izmed dodanih tabel v izbirnem meniju se ta izpiše v delovnem oknu z vsemi pripadajočimi funkcijami, podobno kot pri osnovnih tabelah. Razlika je v tem, da se administratorjem pod tabelo izpiše še okno z zgodovino, če je njeno beleženje vključeno. Ob vsakem zapisu tabele, ki z drugimi ni povezana, se izpišeta gumba za izbris vrstice in njeno urejanje. Ob vsaki vrstici povezanih tabel pa se izpišejo tudi ustrezna prikazna polja vseh povezanih tabel in gumb za izbris vrstice. Namesto gumba za urejanje pa se izpiše gumb za podrobnosti. Ob kliku nanj se nam prikaže stran z vsemi podrobnostmi izbranega zapisa, skupaj z gumboma za brisanje in urejanje, prav tako pa podrobnosti vseh dodeljenih zapisov iz drugih tabel. Na tem mestu lahko te zapise tudi odstranjujemo in dodajamo, glede na omejitve povezave med ustreznima tabelama.

4.3.11. Beleženje zgodovine sprememb

Vsaki tabeli lahko nastavimo beleženje zgodovine sprememb, ko jo dodamo v seznam. Kasneje te možnost ne moremo spreminjati, razen če tabelo najprej odstranimo iz seznama in jo nato zopet dodamo, pri čemer se odstranitev zabeleži. Za ta namen smo v podatkovni bazi ustvarili novo tabelo *zgodovinaTabel*. Vanjo bomo shranjevali spremembe, ki se zgodijo v določeni tabeli, kadar je ta možnost vključena. Spremembe so razdeljene po barvah:

- Spremembe osnovnih podatkov zapisov (črna)
- Dodajanja zapisa, nova povezava (zelena)
- Odvzemanje zapisa, izbris povezave (rdeča)

Vsaka sprememba je v oknu z zgodovino zapisana v svojo vrstico, pri čemer je njena struktura naslednja: datum, čas (uporabnik); sprememba.

Okno z zgodovino je prikazano administratorjem pod ustrezno tabelo. Pri osnovnih tabelah smo jo prikazali zgolj na strani s podrobnostmi posameznega zapisa. Sedaj smo to posplošili na celotno tabelo, saj zgodovino ne shranjujemo več za vsak zapis posebej.

```
if($beleziZgodovino == 1) {
    $result = mysql_query("SELECT * FROM zgodovinatabel WHERE imeTabele='$imeTabele'", $db)
    $row = mysql_fetch_array($result);
    $zgodovina = $row['zgodovina'];
    $zgodovina = addslashes($zgodovina) . "<font color='#4AA02C'>$date ($username); Dodan
    zapis ID=$idVnosa.</font><br>";
    $result = mysql_query("UPDATE zgodovinatabel SET zgodovina='$zgodovina' WHERE
    imeTabele='$imeTabele'", $db) or die(mysql_error());
}
```

Slika 18: Primer beleženja zgodovine

Zgoraj je primer kode za dodajanje vrstice v zgodovino tabele ob vstavljanju novega zapisa vanjo.

4.3.12. Dinamično povezovanje tabel

Osnovne tabele smo fiksno povezali med seboj na logičen način, kar se je odražalo pri njihovem izpisu, dodajanju novih zapisov in beležnju zgodovine. Sedaj želimo to funkcijo vpeljati med poljubne tabele iz seznama. V izbirnem meniju smo tako dodali med administratorske funkcije možnost povezovanja tabel. Ob kliku nanjo se nam prikaže obrazec, kjer lahko izbiramo med poljubnima dvema tabelama iz podatkovne baze, ki ju želimo povezati. Pod obrazcem se izpiše seznam že obstoječih povezav, kjer lahko slednje tudi izbrišemo.

Podatke o povezanih tabelah bomo shranjevali v za to posebej pripravljeno tabelo `povezaneTabele` v podatkovni bazi. Vanjo bomo shranjevali naslednje parametre:

- ime prve tabele
- prikazno polje prve tabele
- relacija prve tabele do druge tabele
- ime druge tabele
- prikazno polje druge tabele
- relacija druge tabele do prve tabele
- ime povezovalne tabele

V prikazno polje bomo shranili ime polja, ki smo ga določili kot prikaznega. To polje prikažemo poleg osnovnih polj nekega zapisa v tabeli, ki mu je dodeljen nek zapis iz druge tabele. Tako nam poleg osnovnih polj zapisa ni potrebno izpisati vseh polj njemu dodeljenega zapisa iz druge tabele, kar skrajšuje izpis in hkrati omogoča smiselno referenco.

Ob ustvarjanju nove povezave tako izberemo vse željene parametre, nakar se povezava doda v seznam povezav (parametri se shranijo v tabelo `povezaneTabele`), med tabelama pa se v podatkovni bazi ustvari še vmesna tabela, podobno kot to prikazuje slika 14. V to tabelo se shranjujejo ID vrednosti povezanih zapisov, s tem pa omogočimo, da lahko ima nek zapis iz prve tabele več dodeljenih zapisov iz druge. Spodnja slika prikazuje osnovni obrazec za dodajanje povezav in simboličen izpis že obstoječih s Crow's foot notacijo, pri čemer je na desni strani puščice (`-->`) ime povezovalne tabele.

Ustvari povezovalno tabelo med naslednjima že obstoječima:

Izberi prvo tabelo:

Izberi drugo tabelo:

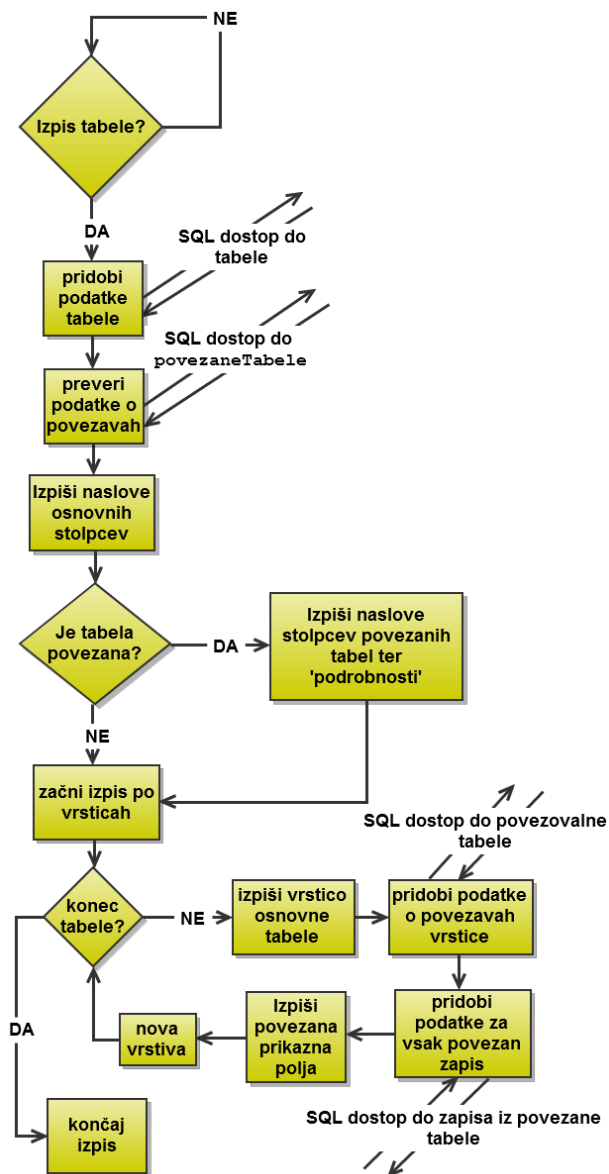
Že povezane tabele:

ekipe >0—0< vozila —> ekipe_vozila	<input type="button" value="Briši povezavo"/>
ekipe +0—0< vozniki —> ekipe_vozniki	<input type="button" value="Briši povezavo"/>

Slika 19: Dodajanje in izpis povezav med tabelami

Pri pisanju kode smo bili pozorni na to, da smo onemogočili povezovanje dveh že povezanih tabel in iste tabele s seboj. Prav tako smo upoštevali dolžino imen tabel, saj je ta omejena, zato je pri ustvarjanju imena povezovalne tabele (ki je v osnovi enako imenu obeh tabel, ločenih s podčrtajem) potrebno v primeru predolgh imen te ustrezno skrajšati.

Naslednji diagram (slika 20) predstavlja poenostavljen potek kode za izpis tabele s povezavami, iz katerega je tudi razvidno, do katerih tabel v podatkovni bazi pri tem dostopamo.



Slika 20: Poenostavljen diagram kode za izpis tabele s povezavami

4.3.13. Izvoz tabel, iskanje, sortiranje

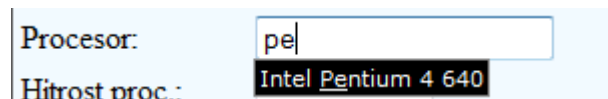
Izvoz tabel deluje podobno kot pri osnovnih tabelah, le da je koda prilagojena dejstvu, da moramo poleg podatkov tabel pridobiti tudi imena polj. Za izvoz v format PDF smo ustvarili poseben gumb poleg izpisa imena tabele, saj je ta format namenjen izpisu poročil in ne izvozu

podatkov. Ob gumb smo dodali možnost za izpis poročila skupaj z zgodovino, če je njeno beleženje za to tabelo vključeno.

Funkciji iskanja in sortiranja smo prav tako prilagodili za poljubno tabelo in delujeta enako kot prej.

4.3.14. Vstavek za izpis predlogov pri vnašanju podatkov

Pri vnašanju podatkov v obrazce je pomembno, da iste elemente vedno označimo enako, da ne pride do zmešnjave in nepreglednosti. V ta namen smo v obrazce namestili vstavek, ki ob tipkanju samodejno izpiše predloge za vnos, glede na že vnešene zapise nekega polja iz podatkovne baze. Tako preprečimo, da bi enak element dobil različne oznake.



Slika 21: Izpis predlogov za vnos

Vstavek [19] deluje na osnovi jezika JavaScript [18] in je odprtokoden. Za delovanje smo ga prilagodili naši aplikaciji. Na sliki 22 je prikazan del kode s funkcijo za dodajanje vrstice predlogov k vnašalnemu polju.

```

1  <script language="javascript" type="text/javascript" src="actb.js"></script>
2  <script language="javascript" type="text/javascript" src="common.js"></script>
3
4  <?php
5  function addSuggestionLabel($imePolja, $imeTabele) {
6      require("DBinfo.php");
7      $db = mysql_connect("$dbHost", "$dbUser", "$dbPass") or die ("Error connecting t
8      mysql_select_db("$dbDatabase", $db) or die ("Couldn't select the database.");
9      $resultsug = mysql_query("SELECT $imePolja FROM $imeTabele", $db);
10
11     $i=0;
12     while($row = mysql_fetch_array($resultsug)) {
13         $suggest[$i] = $row[$imePolja];
14         $i++;
15     }
16     if(isset($suggest)) {
17         $suggest = array_unique($suggest);
18         $suggest = array_values($suggest);
19     }
20 }
  
```

Slika 22: Del kode funkcije za dodajanje vrstice predlogov

K posameznemu vnosnemu polju dodamo vrstico predlogov z ustrežno funkcijo `addSuggestionLabel()`, kot prikazuje slika 23. Prvi argument predstavlja ime polja, drugi pa ime tabele, iz katere se predlogi berejo.

```

<td>
    Processor:
</td>
<td>
    <input name="procesor" id="procesor" type="text" size="20" maxlength="20" va
    <?php addSuggestionLabel("procesor", "konfiguracija"); ?>
</td>

```

Slika 23: Funkcija za dodajanje izpisa predlogov

4.3.15. Urejanje izgleda z jezikom CSS

Z jezikom CSS smo definirali enoten izgled za vse izpisane tabele. Spodaj je prikazana koda CSS datoteke.

```

1 <style type="text/css">
2
3 table.dataTable {
4     border-width: 2px;
5     border-spacing: 1px;
6     border-style: inset;
7     border-color: gray;
8     border-collapse: collapse;
9     background-color: rgb(255, 245, 238);
10 }
11 table.dataTable th {
12     border-width: 1px;
13     padding: 3px;
14     border-style: inset;
15     border-color: gray;
16     background-color: #D6E4E1;
17     -moz-border-radius: ;
18 }
19 table.dataTable td {
20     border-width: 1px;
21     padding: 3px;
22     border-style: inset;
23     border-color: gray;
24     background-color: #EDFFFF;
25     -moz-border-radius: ;
26 }
27
28 </style>

```

Slika 24: koda CSS za izpis tabele

Omenjeni izgled tabeli dodelimo enostavno tako, da v definicijo tabele v kodi HTML dodamo kodo `class='dataTable'`.

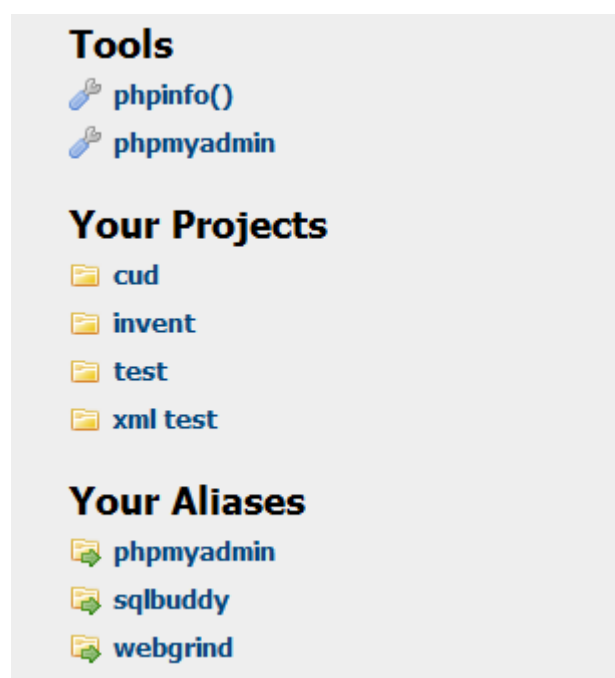
5. Namestitev in vzdrževanje

Aplikacijo moramo za pravilno delovanje najprej namestiti na izbrani stražnik, od koder bo dostopna tudi drugim računalnikom v omrežju. Zato na server najprej namestimo WampServer. V ustrezno mapo (.../wamp/www/) nato skopiramo datoteke aplikacije, podatkovno bazo pa najprej izvozimo in nato ponovno uvozimo z orodjem phpMyAdmin.

Za vzdrževanje aplikacije so odgovorni administratorji v smislu urejanje tabel, njihovih relacij ter dodeljevanja pravic uporabnikom. Aplikacija je tudi precej fleksibilna, zato ji lahko po potrebi dodajamo ali izboljšujemo funkcije z vstavki in popravki kode PHP in HTML.

6. Opis uporabe programa

Za lažjo predstavo delovanja programa bomo prikazali primer uporabe v vlogi administratorja. Ker do aplikacije dostopamo iz serverja, lahko v naslovno vrstico brskalnika vpišemo kar localhost. Odpre se nam osnovna stran programa WampServer, kjer lahko dostopamo do raznih orodij in projektov, med katerimi je tudi naša aplikacija.



Slika 25: Osnovna stran WampServer-ja

Odpremo jo s klikom na datoteko cud, kjer se nahajajo vse datoteke s kodo aplikacije. Prikaže se nam vstopna stran, kjer za prijavo vpišemo naše uporabniško ime in geslo.

The screenshot shows a light blue background with the title 'IT inventar' in bold black text at the top center. Below the title is a white rectangular box with a thin border. Inside this box, the text 'Prijava uporabnika' is displayed in bold. There are two input fields: the first is labeled 'Uporabnik :' and contains the text 'admin'; the second is labeled 'Geslo :' and contains five black dots. A 'Prijava' button is located at the bottom right of the box.

Slika 26: Prijavna stran

Sedaj se nam prikaže glavno okno aplikacije z vsemi funkcijami (slika spodaj).

The screenshot shows the main interface of the 'IT inventar' application. At the top, there is a light blue header with the title 'IT inventar' in bold black text. Below the header, the page is divided into two main sections. On the left, there is a sidebar with a light blue background. It contains the following text: 'Pozdrav, admin!', 'Vloga: Administrator', and a blue underlined link 'Odjava'. Below this is the text 'Izberi:' followed by a horizontal line. Under the line are three blue underlined links: 'Osebe', '>Računalniki', '>>Konfiguracije', and '>>Inštalacije'. Below these links is another horizontal line, followed by the text 'Admin. funkcije:' and three blue underlined links: 'Uporabniki', 'Dodaj tabelo', and 'Povezave tabel'. On the right side of the page, there is a white area. At the top right, it says 'Stran naložena: 13.8. 2011 09:26'. Below this is the title 'Glavna stran' in bold black text. Underneath is a horizontal line, followed by the text 'Izberi možnost v meniju.'. At the bottom of the page, there is a light blue footer with the text '| © Blaž Cegnar|'.

Slika 27: Glavna stran aplikacije

V podatkovno bazo bomo dodali nov računalnik in ga dodelili enemu izmed uporabnikov. V izbirnem meniju kliknemo na možnost računalniki. V delovnem oknu na desni se nam izpiše tabela računalnikov, kjer izberemo možnost Dodaj računalnik.

Izberi:

[Dodaj računalnik](#) Sortiraj po: ID ▼ [Sortiraj](#) Išči:

[Osebe](#)
 >[Računalniki](#)
 >>[Konfiguracije](#)
 >>[Inštalacije](#)

ID	Ime	Uporabnik(i)	Inventarna	Dobavljeno	Dobavitelj	Gar.	
45	janezPC	Janez Gornik	1012201101	06.11.2011	Avtera	3	8:

Slika 28: Dodajanje računalnika

Izpiše se nam obrazec za vnos parametrov (spodaj), pri čemer so obvezna polja obarvana rdeče.

Izberi:

[Osebe](#)
 >[Računalniki](#)
 >>[Konfiguracije](#)
 >>[Inštalacije](#)

Admin. funkcije:
[Uporabniki](#)
[Dodaj tabelo](#)
[Povezave tabel](#)

Dodaj računalnik

Datum vnosa: 13.08.2011

Ime: JanezPC2

Uporabnik: brez ▼

Inventarna: brez
 Janez Gornik

Datum dobave: 13 ▼ 8 ▼ 2011 ▼

Dobavitelj: 34 • Janez • Gornik • prog • programer • janez.gornik@mail.je

Garancija: 3 ▼

Št. računa: 329725982652738

Konfiguracija: brez ▼

Inštalacija: brez ▼

Opombe:

[Shrani](#) [Prekliči](#)

Slika 29: Obrazec za dodajanje računalnika

Datum vnosa se izpolni samodejno in ga ne moremo popravljati. Ostale parametre ustrezno izpolnimo. Ker že vemo, komu bomo dodali računalnik in ker je ta oseba že vnešena v bazo, jo lahko izberemo v spustnem seznamu uporabnikov. Ob pridrzanju kurzorja se nam izpišejo vsa pripadajoča polja. Konfiguracijo in inštalacijo za ta računalnik še nismo vnesli v bazo, zato ti dve polji pustimo nedefinirani. Vnos potrdimo s klimom na gumb shrani. S tem se vrnemo na prejšnjo stran, kjer je med izpis dodan naš računalnik (JanezPC2).

Baza računalnikov

Dodaj računalnik Sortiraj po: ID Sortiraj Išči: Išči Izvozi izpis v: XML Izvozi

ID	Ime	Uporabnik(i)	Inventarna	Dobavljeno	Dobavitelj	Gar.	Št. računa	Konfiguracija	Inštalacija	Vnešeno	Briši	Podrobnosti
45	janezPC	Janez Gornik	1012201101	06.11.2011	Avtera	3	812789210461378	RIP-CAE2.xml	Win7 64 MSOFF 2010	25.06.2011	Briši	Podrobnosti
46	JanezPC2	Janez Gornik	1308201104	13.08.2011	Avtera	3	329725982652738	BREZ	BREZ	13.08.2011	Briši	Podrobnosti

Slika 30: Izpis baze računalnikov

Sedaj mu želimo dodati še ustrezno konfiguracijo in inštalacijo. Podobno kot prej v izbirnem meniju izberemo Konfiguracije, le da bomo tokrat dodali konfiguracijo iz datoteke XML, ki smo jo pred tem ustvarili na ustreznem računalniku.

Baza konfiguracij

ID konf. za šablono: Dodaj konfiguracijo Dodaj konf. iz XML

Izvozi izpis v: XML Izvozi

ID	Oznaka	Procesor	Hitrost proc.	RAM	Disk
27	RIP-CAE2.xml	Intel Core 2 Duo	3166 MHz	8192	Unknov manufac

Slika 31: Možnosti dodajanja konfiguracije

Izberemo možnost Dodaj konf. Iz XML, kjer izberemo ustrezno datoteko. Obrazec se nam samodejno izpolni, konfiguracijo le še shranimo. Nato izberemo Računalniki in poleg našega računalnika (lahko ga poiščemo s funkcijo iskanja) izberemo možnost Podrobnosti. Izpišejo se nam vsi podatki in relacije računalnika (slika 31). Sedaj mu lahko dodamo prej vnešeno konfiguracijo s klikom na Dodaj konfiguracijo, kjer iz izbirnega menija izberemo željeno in vnos potrdimo. S tem se vrnemo na podrobnosti, kjer se pod dodeljena konfiguracija izpišejo ustrezni podatki. Postopek ponovimo še pri dodajanju inštalacije.

Podrobnosti računalnika: JanezPC2 z zgodovino

ID	Ime	Uporabnik(i)	Inventarna	Dobavljeno	Dobavitelj	Garancija	Št. računa	Vnešeno	Podatki računalnika	
46	JanezPC2	<input type="button" value="Janez Gornik"/>	1308201104	13.08.2011	Avtera	3	329725982652738	13.08.2011	<input type="button" value="Briši"/>	<input type="button" value="Uredi"/>

Opombe:

Dodeljena konfiguracija:
BREZ

Dodeljena inštalacija:
BREZ

Zgodovina: *Zadnji popravek spodaj.

13.08.2011 9:47:55 (admin); dodan upor.: ID: 34, Janez Gornik |

Legenda barv: sprememba osnovnih podatkov / dodajanje / odvzemanje / sprememba konf. / sprememba inšt.

Slika 32: Izpis podrobnosti računalnika

Obrazec za dodajanje inštalacije je nekoliko drugačen, kot je razvidno iz spodnje slike. Prilagojen je zahtevanim vnosnim poljem. Nameščene programe izbiramo iz seznam tako, da držimo tipko *ctrl*. Če želimo dodati še kakšne, ki jih ni v seznamu, jih vpišemo v polje Dodaj programe in ločimo s podpičjem. Ob potrditvi vnosa s kliom na Shrani se bodo samodejno dodali v seznam pri naslednjem vnosu. Nato računalniku enako kot prej dodamo še inštalacijo. Tako smo posamezne zapise ustrezno povezali med seboj, kar vpliva na njihove spremembe in izpise ter tudi beleženje zgodovine.

Dodaj inštalacijo

Oznaka:

OS koda:

verzija:

32/64 bit:

Service Pack:

Windows update:

Open/MS Office:

Office koda:

Office verzija:

Tiskalnik PDF:

Opombe:

SI nastavitve:

Formati za Office:

Lokalno admin geslo:

Prijava v domeno:

Uporabnik kot admin:

Servisna orodja:

Zaščita pošte:

Tiskalnik:

Internet:

Združljivostni paket:

Prijava v del. skupino:

Nameščeni programi:

- PPview
- JAVA
- EMO ohranjevalnik
- 7-Zip
- PDFreader
- BS Player

Dodaj programe:

Slika 33: Obrazec za dodajanje inštalacije

Zgoraj omenjene tabele so fiksne in se jih ne da spreminjati. Lahko pa v seznam tabel dodamo tudi nove, poljubne. To storimo tako, da v izbirnem meniju pod Admin. Funkcije izberemo možnost Dodaj tabelo. Pred vnosom jo moramo ustvariti v podatkovni bazi. Nato vnesemo njeno ime, nakar se njena struktura izpiše skupaj z nastavitvami, ki jih lahko določimo (slika 33).

Dodaj novo tabelo v seznam

Vnesite ime tabele, ki ste jo ustvarili v podatkovni bazi:

vozniki:

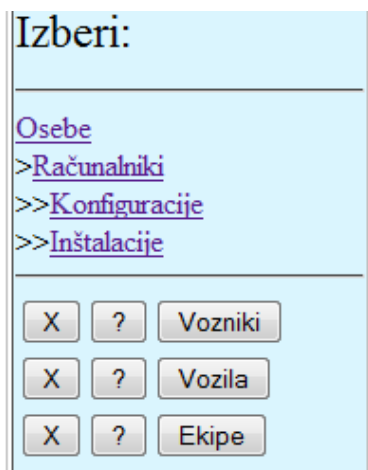
id	ime	priimek	starost	datum_rojstva	naslov	telefon
1	Luka	Novak	22	1988-12-01	Ljubljanska c. 5	768534442
2	Miha	Prašnički	23	1987-12-12	Na Otoku 10	123423513

Izberi nastavitve:	uporabnik	gost
Tabelo lahko preglejuje:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Zapise lahko dodaja:	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Zapise lahko ureja in briše:	<input type="checkbox"/>	<input type="checkbox"/>
Prikazno ime tabele:	<input type="text" value="Vozniki"/>	
Beleži zgodovino tabele:	<input checked="" type="checkbox"/>	

Dodam zgornjo tabelo v seznam?

Slika 34: Dodajanje tabele v seznam

Ob potrditvi se tabela doda v seznam v izbirni meni. Po enakem postopku bomo dodali še dve tabeli, tako bomo v seznamu tabel imeli tri: Vozniki, Vozila in Ekipe.



Slika 35: Izpis dodanih tabel

Sedaj bomo tabele logično (dinamično) povezali med seboj. Izberemo možnost Povezave tabel pod Admin. Funkcije (gles liko 28).



Slika 36: Povezovanje tabel

Izberemo tabeli, ki jih želimo povezati (slika 35), nato pa še ustrezne nastavitve povezave (slika 36).

Izberite prikazno povezovalno polje za obe tabeli:

Prikazno polje za tabelo **vozniki**:

Prikazno polje za tabelo **ekipe**:

Nastavite možnost povezovanja:

Enemu zapisu iz tabele **vozniki** lahko dodelimo več različnih zapisov iz tabele **ekipe**?

Enemu zapisu iz tabele **ekipe** lahko dodelimo več različnih zapisov iz tabele **vozniki**?

Slika 37: Nastavitve povezav

Enako ponovimo še za povezavo med tabelama `vozila` in `ekipe`. V povezavah tabel se nam izpišejo povezave, kot prikazuje spodnja slika.

Že povezane tabele:

<code>vozniki</code> >0 — 0+ <code>ekipe</code> → <code>vozniki_ekipe</code>	<input type="button" value="Briši povezavo"/>
<code>vozila</code> >0 — 0+ <code>ekipe</code> → <code>vozila_ekipe</code>	<input type="button" value="Briši povezavo"/>

Slika 38: Izpis povezav med tabelami

Ob kliku na tabelo `Ekipe` se nam prikaže ustrezen izpis (slika 38).

Izpis tabele 'Ekipe' z zgodovino

Sortiraj po:

ekipe:

id	ime_ekipe	vozniki	vozila	Briši	Podrobnosti
38	Porsche Racing	BREZ	BREZ	<input type="button" value="Briši"/>	<input type="button" value="Podrobnosti"/>
39	Peugeot ekipa	BREZ	BREZ	<input type="button" value="Briši"/>	<input type="button" value="Podrobnosti"/>

Zgodovina: *Zadnji popravek spodaj.

13.08.2011 10:32:28 (admin); Tabela dodana v seznam. Vključeno beležer
 13.08.2011 10:37:57 (admin); Tabela ni več povezana s tabelo: vozila.
 13.08.2011 10:37:59 (admin); Tabela ni več povezana s tabelo: vozniki.
 13.08.2011 10:43:11 (admin); Tabela povezana s tabelo: vozniki.
 13.08.2011 10:43:29 (admin); Tabela povezana s tabelo: vozila.

Legenda barv: sprememba osnovnih podatkov / dodajanje / odzemanje

Slika 39: Izpis povezane tabele

Da je tabela `ekipe` povezana s tabelama `vozniki` in `vozila` je razvidno iz samega izpisa tabele (slika 38), saj se imeni teh dve tabel ustrezno izpišeta v naslovni vrstici dveh stolpcev, ki sta izpisana desno od dvojne navpičen črte, med katerima so polja tabele prazna. V samih poljih teh stolpcev se izpiše ustrezna vrednost iz povezanega zapisa, ki predstavlja referenčno prikazno polje povezane tabele, kot smo ga določili ob definiciji povezave. V primeru iz slike 38 še zapisom tabele `ekipe` nismo določili nobenega zapisa iz tabel `vozniki` oziroma `vozila`, zato je v poljih izpisana vrednost BREZ. Ko pa bi denimo ekipi `Porsche Racing` določili nekega voznika, bi se priimek le-tega izpisal v ustreznem polju v stolpcu `vozniki`, ker smo priimek določili kot referenčno prikazno polje tabele `vozniki`.

S klikom na `Podrobnosti` se izpišejo vsi parametri, kjer lahko tudi (v tem primeru) dodajamo voznike in vozila glede na pravila povezave.

Za konec bomo izpisali še PDF dokument zgornje tabele urejene po stolpcu `ime_ekipe`, skupaj z njeno zgodovino, po tem ko smo določili še voznike in vozila. Pri tem moramo izbrati ime datoteke, ki se samodejno shrani v mapo `Poročila` na strežniku. (Primer PDF poročila na sliki 39.)

Izpis tabele: ekipe

Datum: 13.8. 2011, 10:56h

Skupaj izpisov: 2

id	ime_ekipe	vozniki	vozila
39	Peugeot ekipa	Prašnički;	Ferrari; Porsche;
38	Porsche Racing	Novak;	Peugeot;

Zgodovina: (Zadnji popravek spodaj.)

20.06.2011 13:48:05 (admin); Tabela dodana v seznam. Vključeno beleženje zgodovine.
20.06.2011 13:54:05 (admin); Tabela odstranjena s seznama.
20.06.2011 13:54:11 (admin); Tabela dodana v seznam. Vključeno beleženje zgodovine.

Slika 40: Primer izpisanega poročila v PDF dokument

7. Sklepne ugotovitve

Glede na zastavljene cilje smo projekt uspešno izdelali, že med samo izdelavo pa smo dobili tudi precej idej za možne izboljšave in nadgradnje. Osebnostno mi je pri nalogi največji izziv predstavljalo povezovanje vseh uporabljenih programskih jezikov, saj sem se prvič lotil projekta v takšnem obsegu, zato sem se sproti tudi veliko naučil in si precej razširil razumevanje tematike, ki jo diplomska naloga obsega.

Sama aplikacija ob zaključku naloge vsebuje vse funkcije, ki smo si jih zamislili ob njeni zasnovi. Z namenom smo jo razvijali po modulih, tako da se lahko na enostaven način izboljša in doda funkcionalnosti. Za začetek bi lahko dodelali vnašalni obrazec za vnos podatkov tako, da se univerzalno vnosno polje ustrezno spremeni glede na tip podatka, ki ga vnašamo. Prav tako bi lahko pravice uporabnikov natančneje definirali tako, da bi lahko skupina z uporabniškimi pravicami popravljala svoje vnose, drugih pa ne, saj so sedaj omejeni le na izbiro, kjer ti lahko spreminjajo vse vnose v neki tabeli ali pa nobenega. Idej je torej še veliko in za vse nam je zmanjkalo časa, vseeno pa smo v predvidenem časovnem okviru uspešno izdelali vse predvidene funkcije.

Program je uporabne za pregledno shranjevanje in obdelavo vseh vrst informacij, ne le računalniške opreme, in vsebuje precej pripomočkov za lažje delo. Od podobnih rešitev pa se razlikuje predvsem v tem, da je zasnovan za delovanje preko mreže oziroma spleta, od koder lahko do podatkov dostopa večje število uporabnikov, ki jim lahko pri tem omejimo pravice izvajanja določenih funkcij. Prav tako pa zaradi modularne zasnove omogoča enostavne nadgradnje in popravke glede na potrebe uporabnika.

8. Viri

[1] M. Kaltenekar, Oblikovanje spletnih strani, HTML, CSS in JavaScript: Hitri vodnik, Ljubljana 2006.

[2] M. Štrancar, S. Klemen, PHP in MySQL na spletnem strežniku Apache, Ljubljana 2005.

[3] (2011) Entity-relationship model. Dostopno na:
http://en.wikipedia.org/wiki/Entity-relationship_model

[4] (2011) Hardware Asset Tracker. Dostopno na:
<http://www.ducksoftware.com/hardware-organizer.html>

[5] (2011) Asset Manager Software User Guide. Dostopno na:
<http://www.kzsoftware.com/products/asset-management-software/>

[6] (2011) Computer Admin Pro. Dostopno na:
<http://www.computeradminpro.com/>

[7] (2011) Total Network Inventory Documentation. Dostopno na:
<http://www.softinventive.com/products/total-network-inventory/>

[8] (2011) WampServer. Dostopno na:
<http://www.wampserver.com/en/>

[9] (2011) Wikipedija, Agilne metode razvoja programske opreme. Dostopno na:
http://sl.wikipedia.org/wiki/Agilne_metode_razvoja_programske_opreme

[10] (2011) Wikipedija, HTML. Dostopno na:
<http://sl.wikipedia.org/wiki/Html>

[11] (2011) Wikipedija, PHP. Dostopno na:
<http://sl.wikipedia.org/wiki/Php>

[12] (2011) Wikipedija, XML. Dostopno na:
<http://sl.wikipedia.org/wiki/Xml>

[13] (2011) About Notepad++. Dostopno na:
<http://notepad-plus-plus.org/>

[14] (2011) Apache HTTP Server Project. Dostopno na:
<http://httpd.apache.org/>

[15] (2011) Speccy. Dostopno na:
<http://www.piriform.com/speccy>

[16] (2011) Wikipedia, Document Object Model. Dostopno na:
http://en.wikipedia.org/wiki/Document_Object_Model

[17] (2011) Free PDF. Dostopno na:
<http://www.fpdf.org/>

[18] (2011) Wikipedia, JavaScript. Dostopno na:
<http://en.wikipedia.org/wiki/JavaScript>

[19] (2011) Code Project - Articles, Auto-complete Control. Dostopno na:
<http://www.codeproject.com/KB/scripting/jsactb.aspx>