



Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko

Smetanova ulica 17
2000 Maribor, Slovenija

Dominik Letnar

GRAFIČNI UPORABNIŠKI VMESNIK ZA ŠAHOVNICO DGT

Diplomsko delo

Maribor, september 2014

Diplomsko delo univerzitetnega študijskega programa

GRAFIČNI UPORABNIŠKI VMESNIK ZA ŠAHOVNICO DGT

Študent: Dominik Letnar
Študijski program: UN ŠP Računalništvo in informacijske tehnologije
Mentor: doc. dr. Borko Bošković
Somentor: red. prof. dr. Janez Brest
Lektorica: Teja Pušaver

Maribor, september 2014



Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko
Smetanova ulica 17
2000 Maribor, Slovenija

FERI

Številka: E1042674

Datum in kraj: 14. 08. 2013, Maribor

Na osnovi 330. člena Statuta Univerze v Mariboru (Ur. l. RS, št. 46/2012)
izdajam

SKLEP O DIPLOMSKEM DELU

1. **Dominiku Letnarju**, študentu univerzitetnega študijskega programa RAČUNALNIŠTVO IN INFORMACIJSKE TEHNOLOGIJE, se dovoljuje izdelati diplomsko delo pri predmetu Programski vzorci.
2. **MENTOR:** doc. dr. Borko Bošković
SOMENTOR: red. prof. dr. Janez Brest
3. **Naslov diplomskega dela:**
GRAFIČNI UPORABNIŠKI VMESNIK ZA ŠAHOVNICO DGT
4. **Naslov diplomskega dela v angleškem jeziku:**
GRAPHICAL USER INTERFACE FOR DGT CHESSBOARD
5. Diplomsko delo je potrebno izdelati skladno z "Navodili za izdelavo diplomskega dela". Skladno s 7. členom *Pravilnika o postopku priprave in zagovora diplomskega dela na dodiplomskem študiju*, je bilo odobreno podaljšanje roka za oddajo diplomskega dela do 30. 09. 2014. Diplomsko delo študent odda v treh izvodih (dva vezana izvoda in en v spiralo vezan izvod) ter en izvod elektronske verzije v referatu za študentske zadeve.

Pravni pouk: Zoper ta sklep je možna pritožba na senat članice v roku 3 delovnih dni.

Dekan:

red. prof. dr. Borut Žalik



Obvestiti:

- kandidata,
- mentorja,
- somentorja,
- odložiti v arhiv.

ZAHVALA

Zahvaljujem se mentorju doc. dr. Borku Boškoviću in somentorju red. prof. dr. Janezu Brestu, za priskrbljeno strojno opremo – šahovnico DGT – ter za pomoč in nepogrešljive nasvete pri opravljanju diplomskega dela.

Prav tako hvala asistentu Janiju Dugoniku za napotke in pomoč pri delu.

Posebna zahvala velja staršem in starim staršem, ki so mi omogočili študij na Univerzi v Mariboru ter me vseskozi podpirali in vzpodbujali.

GRAFIČNI UPORABNIŠKI VMESNIK ZA ŠAHOVNICO DGT

Ključne besede: elektronska šahovnica, programska knjižnica Qt, spletna storitev, grafični uporabniški vmesnik, spletna aplikacija

UDK: 004.42:794.1(043.2)

Povzetek

V diplomskem delu smo s pomočjo sodobnih konceptov programskih jezikov razvili program, ki bo omogočal spremljanje igre na šahovnici DGT preko svetovnega spleta. Diplomsko delo je sestavljeno iz namizne in spletne aplikacije ter spletne storitve. Namizna aplikacija prejme podatke od šahovnice in jih posreduje spletni storitvi, ta pa jih shrani v podatkovno zbirko. Spletna aplikacija prejme podatke od spletne storitve in tako prikaže stanje igre v spletnem grafičnem vmesniku različnih odjemalcev. Aplikaciji in spletno storitev smo razvili s pomočjo sodobne in zelo razširjene knjižnice Qt v programskem jeziku C++, razvojnega orodja Microsoft Visual Studio, programskega jezika C#, programskega jezika PHP in tehnologije AJAX.

GRAPHICAL USER INTERFACE FOR DGT CHESSBOARD

Keywords: electronic chessboard, Qt software library, web services, graphical user interface, web application

UDK: 004.42:794.1(043.2)

Abstract

In this thesis we developed a program which will display the game of chess on the DGT chess board via the World Wide Web by using concepts of modern programming languages. The thesis consists of the desktop and web application, and the web service. The desktop application receives data from the chess board and forwards them to the web service, which stores them in a database. The web application receives data from the web service and shows the state of the game in the web graphical user interface (GUI) to different clients. Applications and the web service are developed with the help of modern and very extensive Qt library in the C++ programming language, Microsoft Visual Studio development tools, C# programming language, PHP programming language and AJAX technology.

VSEBINA

1	UVOD	1
2	SORODNA DELA	3
2.1	XBoard	3
2.2	Arena	4
2.3	CheesX	5
3	UPORABLJENE TEHNOLOGIJE	7
3.1	Gonilnik šahovnice DGT	7
3.2	Programska knjižnica Qt	8
3.3	Microsoft Visual Studio	9
3.4	Spletna storitev	9
3.5	Protokol SOAP	9
3.6	Programski jezik PHP	10
3.7	Tehnologija AJAX	10
4	OPIS IMPLEMENTIRANE APLIKACIJE	11
4.1	Povezava s šahovnico DGT	12
4.2	Grafični vmesnik	14
4.3	Spletna storitev	16
4.4	Povezava namizne aplikacije s spletno storitvijo	19
4.5	Spletna aplikacija in povezava s spletno storitvijo	20
5	PREIZKUŠANJE	22
5.1	Testiranje ob prekinitvi povezave	24
6	SKLEP	25
	VIRI	26

KAZALO SLIK

Slika 2.1: Grafični uporabniški vmesnik XBoard [26].....	4
Slika 2.2: Grafični uporabniški vmesnik Arena [5].....	5
Slika 2.3: Grafični uporabniški vmesnik ChessX [9].....	6
Slika 3.1: Uporaba gradnika, signala in reže v okolju Qt.....	8
Slika 4.1: Potek komunikacije.....	11
Slika 4.2: Povezava šahovnice z računalnikom.....	12
Slika 4.3: Povezava ure s šahovnico.....	13
Slika 4.4: Podatki, ki jih vrača šahovnica.....	13
Slika 4.5: Premik figure – prikaz v konzoli.....	14
Slika 4.6: Premik figure – prikaz v grafičnem vmesniku.....	15
Slika 4.7: Nevidne informacije.....	16
Slika 4.8: Metode spletne storitve.....	16
Slika 4.9: Načrt spletne storitve.....	17
Slika 4.10: Struktura datoteke WSDL – metode get.....	18
Slika 4.11: Struktura datoteke WSDL – metode set.....	19
Slika 4.12: Povezava namizne aplikacije in spletne storitve.....	20
Slika 4.13: Povezava spletne aplikacije in spletne storitve.....	20
Slika 4.14: Klic metode spletne storitve iz spletne aplikacije.....	21
Slika 4.15: Grafični vmesnik spletne aplikacije.....	21
Slika 5.1: Grafični prikaz namizne in spletne aplikacije.....	22
Slika 5.2: Vmesno stanje, ko je na potezi igralec z belimi figurami.....	23
Slika 5.3: Premik in izguba figure.....	23

KAZALO TABEL

Tabela 4.1: Simboli in njihova razlaga	15
Tabela 5.1: Rezultati testiranja.....	24

SEZNAM KRATIC

AJAX – Asynchronous JavaScript and XML
ASP – Active Server Pages
CLI – Common Language Infrastructure
DGT – Digital Game Timer
GNU – GNU's Not Unix
GUI – Graphical User Interface
HTTP – HyperText Transfer Protocol
HTTPS – HyperText Transfer Protocol Secure
LGPL – Lesser General Public License
PGN – Portable Game Notation
PHP – HyperText Preprocessor
POSIX – Portable Operating System Interface
RPC – Remote Procedure Calls
SMTP – Simple Mail Transfer Protocol
SOAP – Simple Object Access Protocol
TCP – Transmission Control Protocol
UCI – Universal Chess Interface
USB – Universal Serial Bus
WSDL – Web Services Description Language
XML – eXtensible Markup Language

1 UVOD

V današnjem času je popularno razvijanje namiznih, spletnih in mobilnih aplikacij. Vedno več je novih stvari, vedno večje so spremembe, tehnologija gre z ekstremno hitrostjo naprej. Ljudje imajo na voljo vedno več raznovrstnih sistemov, ki jih z veseljem uporabljajo, predstavlja jim nov izziv ter nudi zadovoljstvo.

Namen diplomskega dela je izdelati namizno in spletno aplikacijo za spremljanje šaha preko svetovnega spleta in neodvisno od lokacije. Spremljanje šaha je lahko velik problem, ker igra poteka za eno mizo ter sprejme le peščico gledalcev. Neodvisnost od lokacije, prihranek denarja in časa, spremljanje partije šaha preko svetovnega spleta in neomejeno število gledalcev so prednosti, ki jih lahko povzamemo v celoto. Posledično lahko takšna rešitev prinese zadovoljstvo ljubiteljev šaha ter igralcem omogoči mirnejšo igro, saj gledalci niso prisotni. Ker ni gledalcev, ni dodatnih morebitnih vplivov na samo igro.

Uporabili bomo elektronsko šahovnico DGT, ki nam pošlje podatke preko povezave USB. Namizna aplikacija prejete podatke shrani in jih posreduje spletni storitvi SOAP. Namen spletne storitve je, da prejete podatke od namizne aplikacije shrani v podatkovno zbirko. S pomočjo spletne storitve bomo podatke iz podatkovne zbirke posredovali spletni aplikaciji, ki prejete podatke prikaže različnim odjemalcem.

Cilj diplomske naloge je, da razpoznamo podatke dobljene od elektronske šahovnice DGT in jih prikažemo v namizni ter v spletni aplikaciji. Namizna in spletna aplikacija bosta uporabljali spletno storitev SOAP, s pomočjo katere si bomo iz namizne aplikacije posredovali podatke v spletno aplikacijo. Podatke bomo zaradi večje preglednosti prikazovali v grafičnem uporabniškem vmesniku različnim odjemalcem.

V drugem poglavju bomo predstavili nekaj sorodnih del, ter izpostavili njihove prednosti. V tretjem poglavju bomo predstavili uporabljene tehnologije, ki smo jih uporabili pri našem delu in izpostavili značilnosti ter odločitev, zakaj smo jih izbrali. V četrtem poglavju bomo

predstavili, kako smo implementirali našo aplikacijo, dotaknili se bomo povezave s šahovnico, implementacijo namizne in spletne aplikacije ter spletne storitve. Predstavili bomo še, kako smo povezali aplikaciji s spletno storitvijo. V petem poglavju bomo predstavili rezultate preizkušanja. V šestem poglavju bomo zaključili naše diplomsko delo ter predstavili možnosti za nadaljnje delo.

2 SORODNA DELA

Grafični uporabniški vmesniki za spremljanje šaha nam omogočajo različne funkcije, kot so shranjevanje in nalaganje igre, analiziranje, igranje proti računalniku ter spremljanje iger. V nadaljevanju bomo podrobneje opisali nekaj izbranih sorodnih del, in sicer XBoard, Arena ter ChessX.

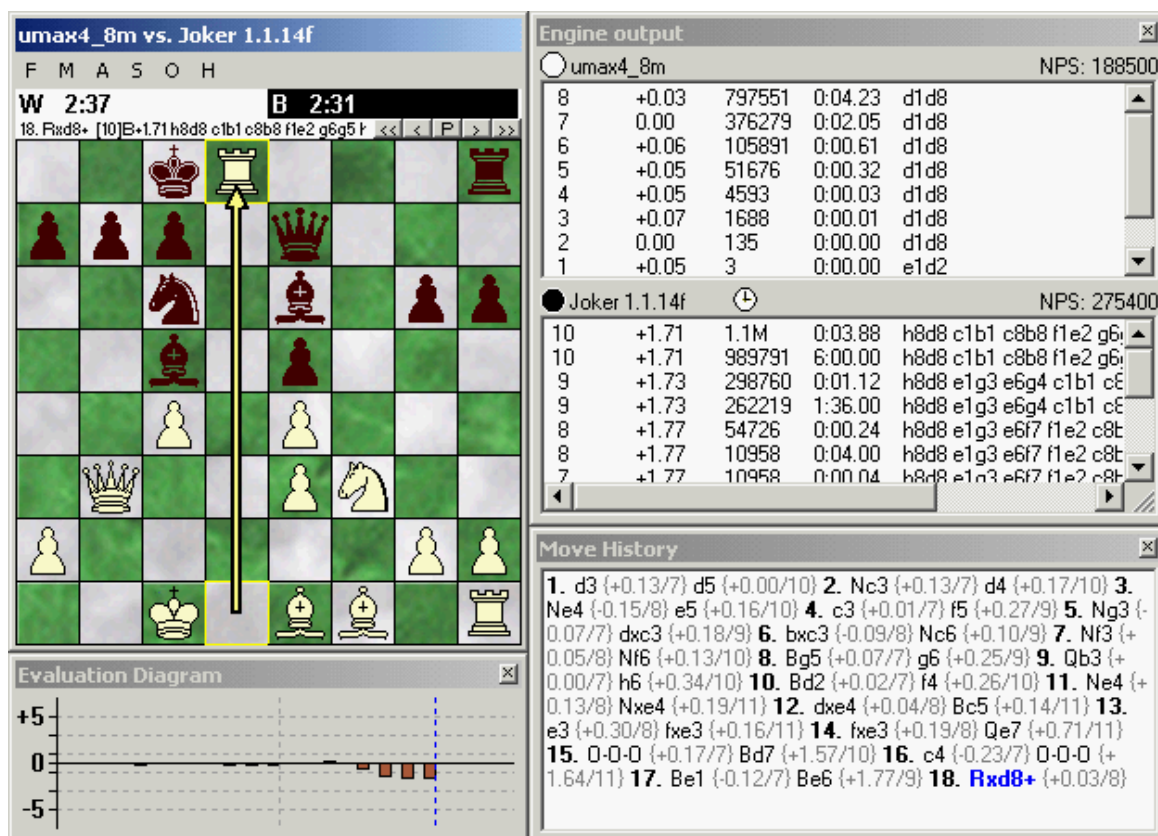
2.1 XBoard

XBoard [25] je grafični uporabniški vmesnik za šah v vseh njegovih pomembnejših oblikah, kot so mednarodni šah, kitajski šah (xiangqi), japonski šah (shogi) in Makruk, poleg manjših različic, kot so Losers Chess, Crazyhouse, Chess960 in Capablanca Chess. Prikaže nam celotno šahovnico na zaslonu, omogoča premik figur z miško ter shranjevanje in nalaganje igre v formatu PGN (*angl. Portable Game Notation*). Grafični uporabniški vmesnik XBoard je prikazan na sliki 2.1, ki prikazuje igralno polje, zgodovino potez in nekaj ostalih podatkov.

Uporabniški vmesnik XBoard je kompatibilen in deluje na sistemih UNIX in UNIX podobnim sistemih, ki uporabljajo sisteme X Windows. Nudi podporo za šahovnico DGT (*angl. Digital Game Timer*).

Uporabniški vmesnik XBoard vključuje tudi:

- protokol za povezovanje s pogoni. Pogon (*angl. chess engine*) je program, ki vsebuje umetno inteligenco za igranje igre. Šahovski igralni pogon omogoča igranje in analiziranje igre na naši napravi (računalniku),
- povezavo na šahovski strežnik, ki omogoča povezavo z drugimi igralci po celem svetu, prav tako lahko gledamo igre drugih igralcev ter omogoča klepet z drugimi uporabniki,
- program CMail samodejno sporoči vsako odigrano potezo našega nasprotnika na naš elektronski naslov. Nasprotnik je na enak način obveščen ob naši odigrani potezi.



Slika 2.1: Grafični uporabniški vmesnik XBoard [26]

2.2 Arena

Arena [4] je brezplačen grafični uporabniški vmesnik za igranje šaha, ki nam omogoča analizo igre in igranje šaha, prav tako omogoča testiranje pogonov. Deluje na sistemih družine Windows ali pa na sistemih Linux s programom Wine.

Grafični uporabniški vmesnik, ki je prikazan na sliki 2.2, lahko prilagodimo lastnim potrebam, hkrati pa je enostaven za uporabo. Obstaja približno 250 pogonov, ki jih podpira grafični uporabniški vmesnik Arena in omogoča spreminjanje težavnosti igre (lahka, srednja, težka,...). Vmesnik podaja podrobne informacije o šahovski analizi, ki mu jo posredujejo pogoni. Prav tako omogoča igranje z drugimi igralci preko interneta, omogoča tudi samodejno analizo iger in napredno pomoč. Podpira tudi shranjevanje in nalaganje igre v formatu PGN in nudi podporo uporabe šahovnice DGT.

Grafični uporabniški vmesnik Arena je kompatibilen s protokoli Winboard in UCI (*angl. Universal Chess Interface*), prav tako podpira Chess960 in še mnogo več.



Slika 2.2: Grafični uporabniški vmesnik Arena [5]

2.3 CheesX

ChessX [1] je odprtokodna šahovska podatkovna zbirka, ki ne nudi podpore za šahovnico DGT. Z njo lahko upravljamo našo zbirko šahovskih iger v različnih načinih, kot je brskanje, spreminjanje, organiziranje, analiziranje in podobno. Deluje tudi na različnih platformah, kot so Windows, Linux in Mac Os X. Podpira shranjevanje in nalaganje iger v formatu PGN, deluje tudi na več podatkovnih zbirkah istočasno, vstavljanje potez, spremembe in komentarje le-teh, analiza poteka z uporabo pogonov UCI in Winboard. Grafični uporabniški vmesnik ChessX je prikazan na sliki 2.3, kjer je prikazano igralno polje, zgodovina potez in še mnogo več.



Slika 2.3: Grafični uporabniški vmesnik ChessX [9]

3 UPORABLJENE TEHNOLOGIJE

Razvijalcem je na voljo vedno več različnih razvojnih okolij in programskih orodij. Mi smo izbrali okolje Qt, s pomočjo katerega smo ustvarili namizno aplikacijo, pri kateri smo pridobivali podatke iz elektronske šahovnice DGT. Velika prednost okolja Qt pred ostalimi je, da omogoča prenos naše izvorne kode na več različnih platform, ne da bi spreminjali programsko kodo. Pri komunikaciji z elektronsko šahovnico DGT potrebujemo dodatni gonilnik, ki omogoča komunikacijo s šahovnico ter uro, ki je dodatno priključena na šahovnico. Mi smo uporabili gonilnik dgtnix. Pri razvoju aplikacije smo uporabili okolje Microsoft Visual Studio zaradi lažje implementacije in nadaljnjega dela s spletno storitvijo. Spletno storitev SOAP (*angl. Simple Object Access Protocol*) smo implementirali v programskem jeziku C#. Spletno aplikacijo smo razvili s pomočjo programskega jezika PHP (*angl. Hypertext Preprocessor*), ker je zelo razširjen in omogoča enostavno upravljanje s podatki, ki jih imamo shranjene v podatkovni zbirki. Da smo izboljšali uporabniško izkušnjo ter interaktivnost spletne aplikacije, smo uporabili tehnologijo AJAX (*angl. Asynchronous JavaScript and XML*), ki omogoča osveževanje podatkov v ozadju. S pomočjo tehnologije AJAX ni potrebno vedno znova nalagati celotne spletne strani. Posodobi se le tisti del strani, ki je potreben glede na dobljene podatke.

3.1 Gonilnik šahovnice DGT

Računalnik in elektronsko šahovnico DGT povežemo s kablom USB (*angl. Universal Serial Bus*). Za povezavo naše aplikacije s šahovnico DGT potrebujemo gonilnike dgtnix [11]. Ti gonilniki podpirajo standard POSIX (*angl. Portable Operating System Interface*) in so objavljeni pod licenco LGPL (*angl. Lesser General Public Licence*). Gonilniki skrbijo za povezavo med našo aplikacijo in šahovnico DGT. Trenutna verzija gonilnika je 1.9.1. in ima že precej izboljšav, kot so jih imele verzije na začetku.

3.2 Programska knjižnica Qt

Namizno aplikacijo smo izdelali v programskem okolju Qt [23][24]. Uporabili smo orodje Qt Creator, ki podpira programski jezik C++ in ga uporablja več kot 500.000 razvijalcev po celem svetu. Njegova največja prednost je ta, da omogoča razvoj celotne aplikacije z manj programske kode, ki jo nato lahko prevedemo in namestimo na različne platforme. Tako lahko eno izvorno kodo uporabimo za več platform, ne da bi razvijali aplikacijo na novo. Samo s prevajanjem dosežemo prenosljivost aplikacij med različnimi platformami.

Okolje Qt vsebuje tudi značilne elemente [6][18], kot so gradniki (*angl. widgets*), s pomočjo katerih zgradimo vizualni vmesnik aplikacije ter signale (*angl. signals*) in reže (*angl. slots*). Reže so navadne metode objekta, ki se prav tako odzovejo na signale. Signali pa so na drugi strani poslani s strani objektov. Signal, ki izhaja iz objekta, je lahko povezan na eno ali več rež enega samega objekta, ki je sprejemnik, ali na več različnih objektov, ki so sprejemniki. Če objekt pošlje signal, so vse reže povezane s signalom. Če ni ujemajoče povezave, se ne zgodi nič.

Gradniki lahko vsebujejo druge gradnike. Aplikacijsko okno je gradnik, ki velikokrat vsebuje QMenuBar, QToolBars, QStatusBar in še kaj drugega. Veliko aplikacij za aplikacijsko okno uporablja QMainWindow ali QDialog. Ker je okolje Qt zelo prilagodljivo, se lahko za okno uporabi kateri koli gradnik.

Primer uporabe signala, reže in gradnika je na sliki 3.1, ki prikazuje gumb z besedilom končaj in v primeru klika na gumb se aplikacija zapre.

```
QPushButton gumb("končaj");  
gumb.show();  
QObject::connect(&gumb, SIGNAL(clicked()), &a, SLOT(quit()));
```

Slika 3.1: Uporaba gradnika, signala in reže v okolju Qt

Splošno namenski programski jezik C++ [10] se je razvil iz jezika C, ki so mu med drugim dodali razrede, virtualne funkcije, večkratno dedovanje, šablone in izjeme. C++ se prevede v strojni jezik, kar pa mu omogoča, da se izvaja bistveno hitreje kot jeziki, ki se prevajajo v vmesni jezik (Java, C#, PHP,...).

3.3 Microsoft Visual Studio

Da smo na lažji način povezali spletno in namizno aplikacijo, ki uporabljata različni tehnologiji, smo uporabili spletno storitev in podatkovno zbirko MySQL [19]. Spletno storitev smo lahko klicali iz namizne aplikacije in spletne aplikacije ter skrbeli za shranjevanje ali posredovanje podatkov. Pri tem smo uporabili razvijalno okolje Microsoft Visual Studio Ultimate 2012. Projekt, ki smo ga ustvarili smo pisali v programskem jeziku C#.

Programski jezik C# [7] je objektno orientiran jezik in deluje na .NET platformi. C# je preprost, modernejši in splošno namenski jezik, zgrajen je na osnovi standarda CLI (*angl. Common Language Infrastructure*). C# je naslednik programskega jezika C++ in je zelo podoben programskemu jeziku Java, vendar pa ima tudi nekaj svojih značilnosti.

3.4 Spletna storitev

Spletne storitve (*angl. web services*) [12] so programske storitve, ki so sestavljene iz različnih funkcij (metod) in jih lahko oddaljeno kličemo preko spleta. Spletna storitev je neodvisna od implementacije programskega jezika ter računalnika, na katerem se izvaja. S pomočjo spletnih storitev dosežemo povezljivost med informacijskimi sistemi (aplikacijami) preko spleta. Mi smo se odločili za spletno storitev, ki temelji na protokolu SOAP.

3.5 Protokol SOAP

SOAP [16] je protokol za izmenjavo informacij, zasnovan na standardu XML (*angl. eXtensible Markup Language*) in se pogosto uporablja v spletnih storitvah. Prav tako je SOAP ena izmed RPC (*angl. Remote Procedure Calls*) tehnologij, ki omogoča izvajanje oddaljenih metod preko interneta. Z uporabo protokola HTTP (*angl. HyperText Transfer Protocol*) in standarda XML omogoča izvajanje kode z minimalnim naborom omejitev preko svetovnega spleta. SOAP lahko uporabljamo tudi pri drugih transportnih protokolih kot so npr.: SMTP (*angl. Simple Mail Transfer Protocol*), TCP (*angl. Transmission Control Protocol*), HTTPS (*angl. HyperText Transfer Protocol Secure*).

3.6 Programski jezik PHP

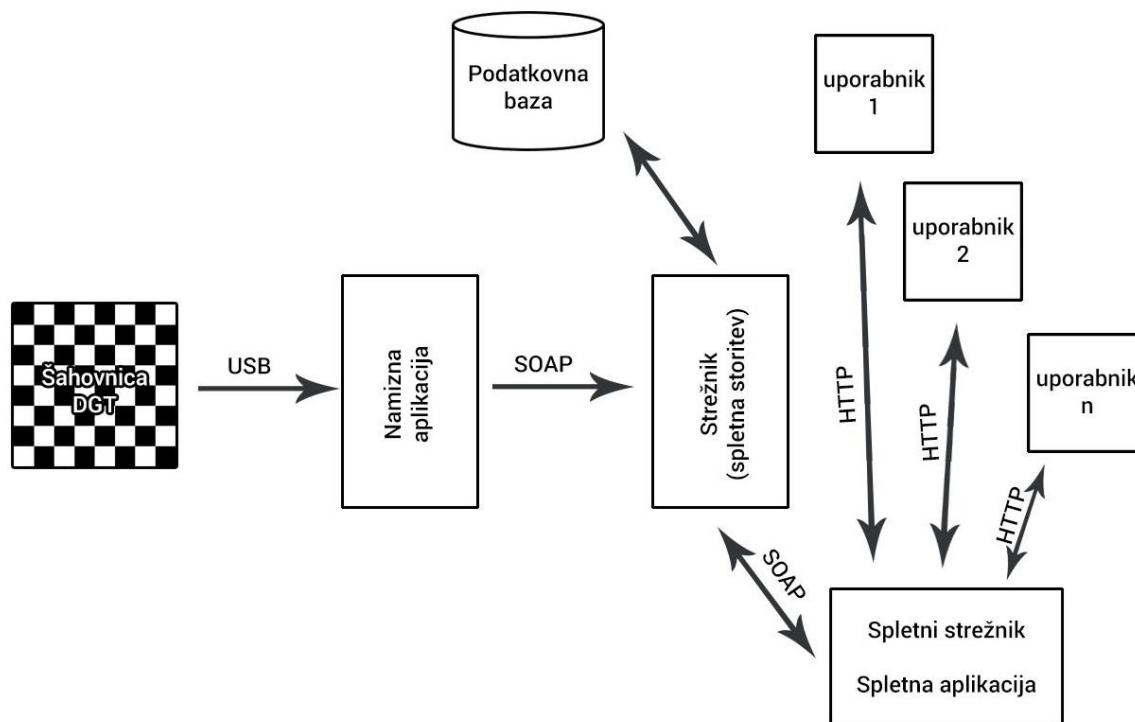
Splošno namenski programski jezik PHP [17][20][21] je zelo priljubljen in razširjen skriptni in odprtokodni programski jezik. Prvotno je bil zasnovan za ustvarjanje dinamične spletne vsebine in še vedno je najbolj primeren za to nalogo. Odlikujejo ga hitrost, prilagodljivost in pragmatičnost. Podobnost običajno najdemo s strukturiranimi programskimi jeziki, kot sta C in Perl, najbolj izkušenim programerjem pa dovoljuje razvijanje zahtevnejših modulov in aplikacij brez dolgotrajnega učenja.

3.7 Tehnologija AJAX

Asinhroni JavaScript in XML (AJAX) [2][3] ni programski jezik, je nova pot uporabe že obstoječih standardov. AJAX je skupina medsebojno povezanih spletnih razvojnih tehnik, ki se uporabljajo za ustvarjanje interaktivnih spletnih aplikacij. S tehnologijo AJAX lahko podatke med spletno aplikacijo in strežnikom izmenjujemo asinhrono v ozadju in ni potrebe po novem nalaganju (osveževanju) strani. Uporabniško izkušnjo s pomočjo tehnologije AJAX zagotovo izboljšamo, saj lahko hitreje in tekoče spreminjamo vsebine v spletni aplikaciji. V primerih, ko uporabimo tehnologijo AJAX, se izognemo dejanju, ko strežnik vedno znova vrne kot odgovor celotno spletno stran in se morajo vsi podatki naložiti na novo.

4 OPIS IMPLEMENTIRANE APLIKACIJE

Namizna aplikacija sprejema podatke, ki nam jih posreduje šahovnica preko protokola USB, jih razpozna in prikaže tudi v grafičnem uporabniškem vmesniku. Namizna aplikacija s pomočjo spletne storitve shranjuje podatke v podatkovno bazo. S pomočjo spletne aplikacije lahko istočasno spremljamo igro. V nadaljevanju bomo podrobneje opisali samo implementacijo in rešitev, s katero smo dosegli želene rezultate. Arhitektura aplikacije in potek komunikacije prikazuje slika 4.1.



Slika 4.1: Potek komunikacije

4.1 Povezava s šahovnico DGT

Elektronska šahovnica DGT je zelo podobna običajnim šahovnicam, saj omogoča igranje šaha s pravimi figurami, hkrati pa ima tudi veliko prednost, ker omogoča posredovanje stanja igre v računalnik. Vse to dosežemo s pomočjo protokola USB. Povezava šahovnice z računalnikom ter povezava ure s šahovnico je prikazana na sliki 4.2.



Slika 4.2: Povezava šahovnice z računalnikom

Elektronska šahovnica DGT omogoča tudi priklop ure, kot je prikazano na sliki 4.3. Ura je obvezen dodatek pri igranju šaha, saj so igre časovno omejene. Šahovnica v primeru povezave z uro posreduje tudi informacije, kateri igralec je na potezi in čas, ki ga imata igralca še na voljo.



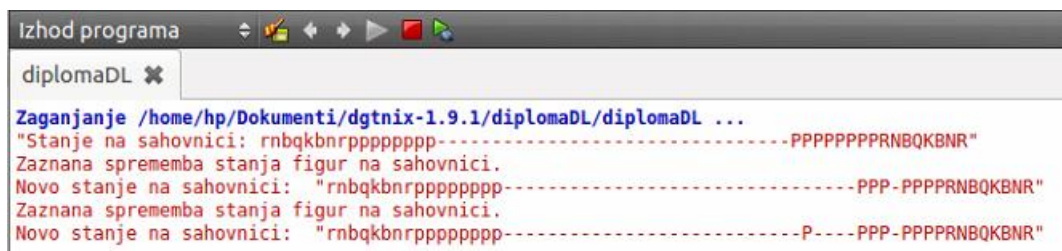
Slika 4.3: Povezava ure s šahovnico

Po uspešni povezavi nam šahovnica pošlje stanje figur na šahovnici, kot je prikazano na sliki 4.4. Figure smo postavili na ustrezna polja za začetek igre in na računalnik sprejeli podatke, ki jih je posredovala šahovnica.

```
Izhod programa
diplomaDL x
Zaganjanje /home/hp/Dokumenti/dgtnix-1.9.1/diplomaDL/diplomaDL ...
"Stanje na sahovnici: rnbqkbnrpppppppp          Ppppppprnbqkbnr"
```

Slika 4.4: Podatki, ki jih vrača šahovnica

Zaradi enostavnejšega pregleda nad poljem, kjer ni figur, smo dodali poseben znak "-". Po premiku figure šahovnica zazna spremembo in le-to posreduje naši aplikaciji kot je prikazano na sliki 4.5.



```
Izhod programa
diplomaDL ✖
Zaganjanje /home/hp/Dokumenti/dgtnix-1.9.1/diplomaDL/diplomaDL ...
"Stanje na sahovnici: rnbqkbnrppppppp-----PPPPPPPRNBQKBNR"
Zaznana sprememba stanja figur na sahovnici.
Novo stanje na sahovnici: "rnbqkbnrppppppp-----PPP-PPPRNBQKBNR"
Zaznana sprememba stanja figur na sahovnici.
Novo stanje na sahovnici: "rnbqkbnrppppppp-----P-----PPP-PPPRNBQKBNR"
```


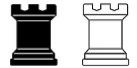

Slika 4.5: Premik figure – prikaz v konzoli

4.2 Grafični vmesnik

S pomočjo grafičnega vmesnika je spremljanje igre lažje in razločnejše kot v primeru na sliki 4.5. Grafični vmesnik poskrbi, da je uporabniku naša aplikacija prijaznejša. Grafični vmesnik prikazuje šahovnico in nevidne informacije (čas igralca z belimi in s črnimi figurami ter informacijo o igralcu, ki je na vrsti). Šahovnica je sestavljena iz 64 oznak (*angl. labels*). Vsaki oznaki smo nastavili ozadje na ustrezno barvo igralnega polja in tako dobili šahovnico.

Oznake, ki jih naši namizni aplikaciji posreduje šahovnica DGT, so oznake figur [8][22], kot prikazuje tabela 4.1 v nadaljevanju. Napisali smo funkcijo, ki za določeno črko prikaže sliko, tako smo ustvarili grafični prikaz figur na šahovnici.

Tabela 4.1: Simboli in njihova razlaga

Simbol [13]	Oznaka [črni / beli]	Ime figur v angleščini	Ime figur v slovenščini
	K / k	King	Kralj
	Q / q	Queen	Dama
	R / r	Rook	Trdnjava
	B / b	Bishop	Lovec
	N / n	Knight	Skakač
	P / p	Pawn	Kmet

Na sliki 4.6 je grafični prikaz premika bele figure tipa kmet za eno polje naprej.



Slika 4.6: Premik figure – prikaz v grafičnem vmesniku

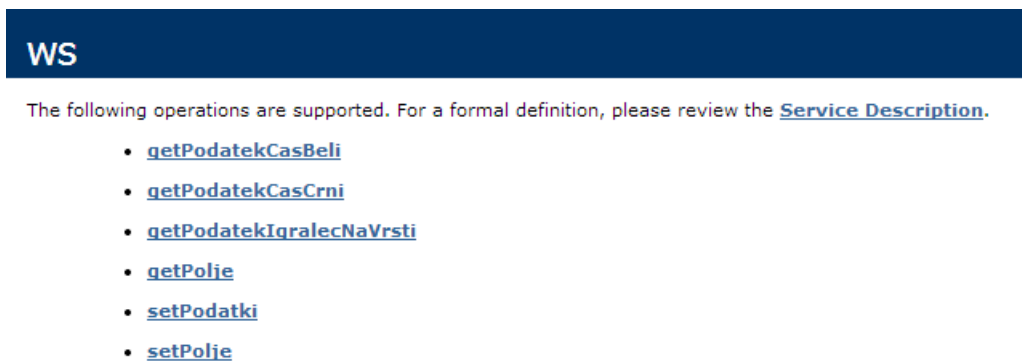
Grafični vmesnik na dosedanjih slikah prikazuje le, šahovnico, figure in kako je s premiki figur. S šahovnico smo povezali tudi uro, ki nam vrne naslednje podatke: čas igralca z belimi figurami, čas igralca s črnimi figurami in igralca, ki je trenutno na vrsti. Grafični vmesnik nam kaže tudi nevidne informacije, kot je prikazano na sliki 4.7.



Slika 4.7: Nevidne informacije

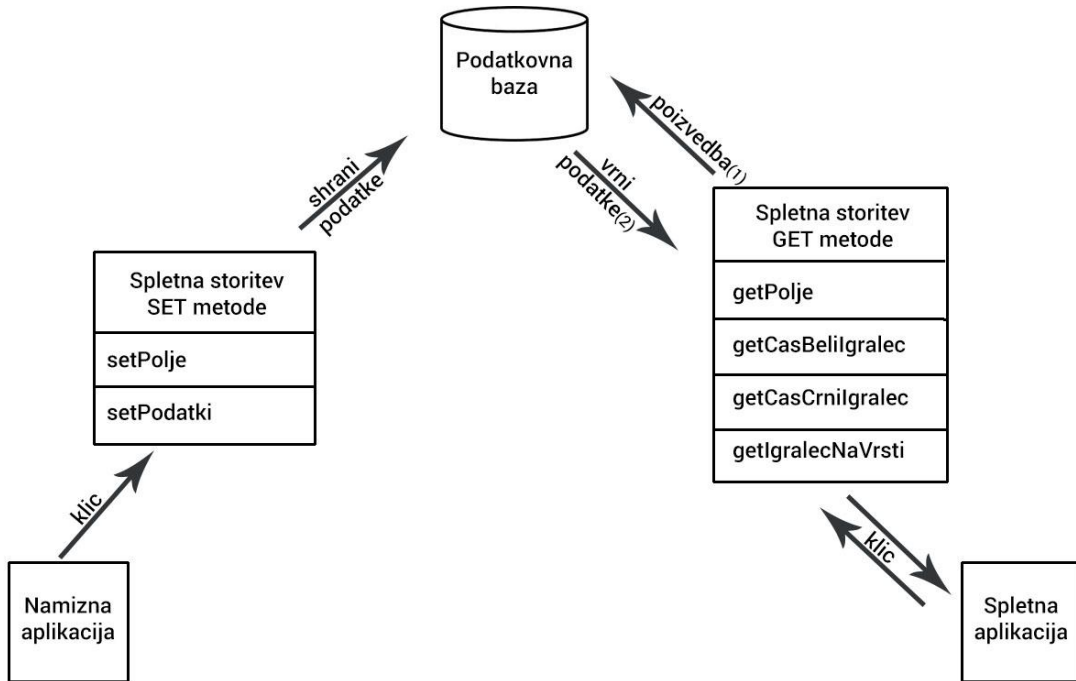
4.3 Spletna storitev

Spletna storitev nam omogoča, da lahko podatke shranimo, ali jih preberemo iz podatkovne zbirke MySQL. Naša spletna storitev vsebuje več metod, ki so prikazane na sliki 4.8. Nekaj metod skrbi za shranjevanje podatkov in jih kličemo iz namizne aplikacije. Metode, ki skrbijo za branje podatkov iz podatkovne zbirke MySQL, kličemo iz spletne aplikacije.



Slika 4.8: Metode spletne storitve

Spletna storitev vsebuje štiri metode, ki vračajo podatke in dve metodi, ki shranjujeta podatke. Načrt spletne storitve, ki smo jo zasnovali, prikazuje slika 4.9. Vse o teh metodah lahko najdemo v datoteki WSDL (*angl. Web Services Description Language*), ki se ustvari samodejno. Datoteka WSDL opisuje spletno storitev, njeno lokacijo in poda operacije, ki jih spletna storitev omogoča [15].



Slika 4.9: Načrt spletne storitve

Metode tipa *get*:

- *getPodatekCasBeli* – vrne čas igralca z belimi figurami,
- *getPodatekCasCrni* – vrne čas igralca s črnimi figurami,
- *getPodatekIgralecNaVrsti* – vrne igralca, ki je na potezi in
- *getPolje* – vrne stanje figur na igralni plošči kot znakovni niz.

V datoteki WSDL najdemo opis *get* metod, kot je prikazano na sliki 4.10. V tej datoteki je zapisano, kaj metoda vrne kot rezultat in kakšnega podatkovnega tipa je.

```

▼<s:element name="getPolje">
  <s:complexType/>
</s:element>
▼<s:element name="getPoljeResponse">
  ▼<s:complexType>
    ▼<s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="getPoljeResult" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
▼<s:element name="getPodatekCasBeli">
  <s:complexType/>
</s:element>
▼<s:element name="getPodatekCasBeliResponse">
  ▼<s:complexType>
    ▼<s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="getPodatekCasBeliResult" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
▼<s:element name="getPodatekCasCrni">
  <s:complexType/>
</s:element>
▼<s:element name="getPodatekCasCrniResponse">
  ▼<s:complexType>
    ▼<s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="getPodatekCasCrniResult" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
▼<s:element name="getPodatekIgralecNaVrsti">
  <s:complexType/>
</s:element>
▼<s:element name="getPodatekIgralecNaVrstiResponse">
  ▼<s:complexType>
    ▼<s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="getPodatekIgralecNaVrstiResult" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="string" nillable="true" type="s:string"/>

```

Slika 4.10: Struktura datoteke WSDL – metode get

Opis metod tipa *set* prav tako najdemo v datoteki WSDL, kot je prikazano na sliki 4.11. V tej datoteki je zapisano, kakšen parameter sprejme metoda ter kakšnega podatkovnega tipa je.

Metodi tipa *set*:

- *setPodatki* – shrani čas igralca z belimi figurami, čas igralca s črnimi figurami ter igralca, ki je na vrsti in
- *setPolje* – shrani položaj figur na šahovnici.

```
▼<s:element name="setPolje">
  ▼<s:complexType>
    ▼<s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="polje" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
▼<s:element name="setPoljeResponse">
  <s:complexType/>
</s:element>
▼<s:element name="setPodatki">
  ▼<s:complexType>
    ▼<s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="casBeli" type="s:string"/>
      <s:element minOccurs="0" maxOccurs="1" name="casCrni" type="s:string"/>
      <s:element minOccurs="0" maxOccurs="1" name="igralecNaVrsti" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
▼<s:element name="setPodatkiResponse">
  <s:complexType/>
</s:element>
```

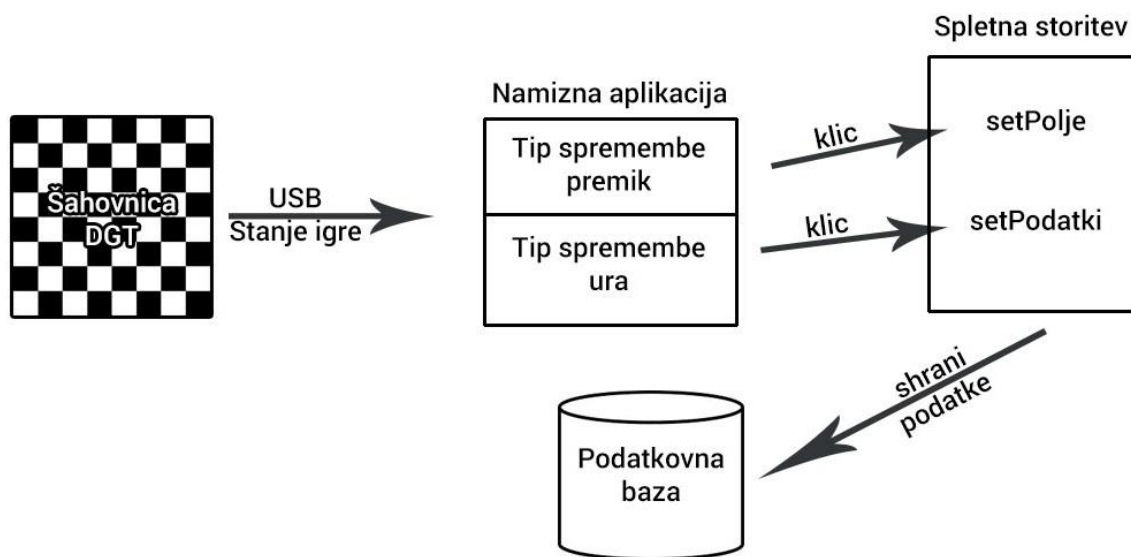
Slika 4.11: Struktura datoteke WSDL – metode set

4.4 Povezava namizne aplikacije s spletno storitvijo

Da vzpostavimo povezavo aplikacije Qt in spletne storitve SOAP, smo v okolju Qt namestili programsko knjižnico gSOAP [14], ki s pomočjo programskega jezika C in C++ omogoča uporabo spletnih storitev SOAP. Orodje analizira datoteko WSDL in zgradi metode, ki so potrebne za uporabo spletne storitve.

V namizni aplikaciji smo implementirali metodo setPolje za pošiljanje niza znakov, ki ga vrača elektronska šahovnica in metodo setPodatki za pošiljanje podatkov za uro (čas igralca z belimi figurami, čas igralca s črnimi figurami in podatek, kateri igralec je na vrsti, da odigra potezo).

Način povezave namizne aplikacije in spletne storitve je prikazan na sliki 4.12.

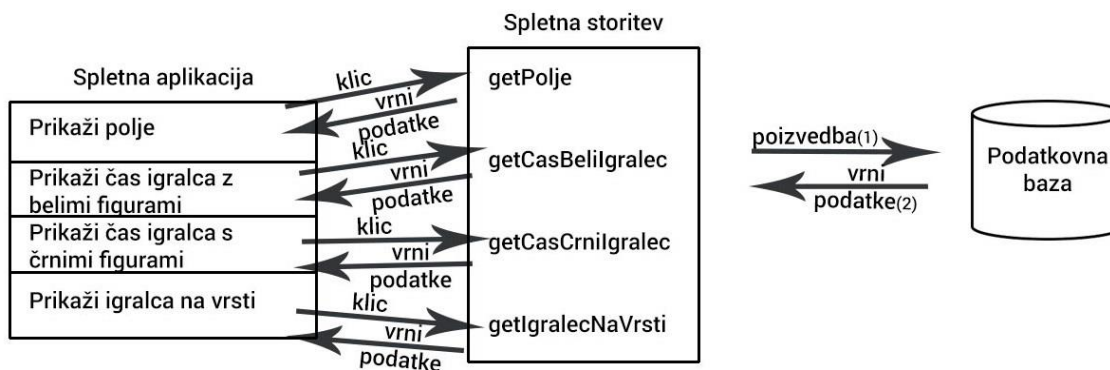


Slika 4.12: Povezava namizne aplikacije in spletne storitve

Ko se na šahovnici spremeni stanje (dvig, premik figure, ipd.) ali čas ure, šahovnica pošlje namizni aplikaciji spremembo. Aplikacija nato ugotovi za kakšno spremembo gre (stanje polja ali sprememba podatkov) in glede na spremembo kliče ustrezno metodo spletne storitve, ki nato podatke shrani v podatkovno zbirko.

4.5 Spletna aplikacija in povezava s spletno storitvijo

Spletna aplikacija kliče ustrezno metodo spletne storitve. Ob klicu metode se naredi poizvedba na podatkovno zbirko. Spletna storitev nato vrne podatke spletni aplikaciji. Način povezave med spletno aplikacijo in spletno metodo je prikazan na sliki 4.13.



Slika 4.13: Povezava spletne aplikacije in spletne storitve

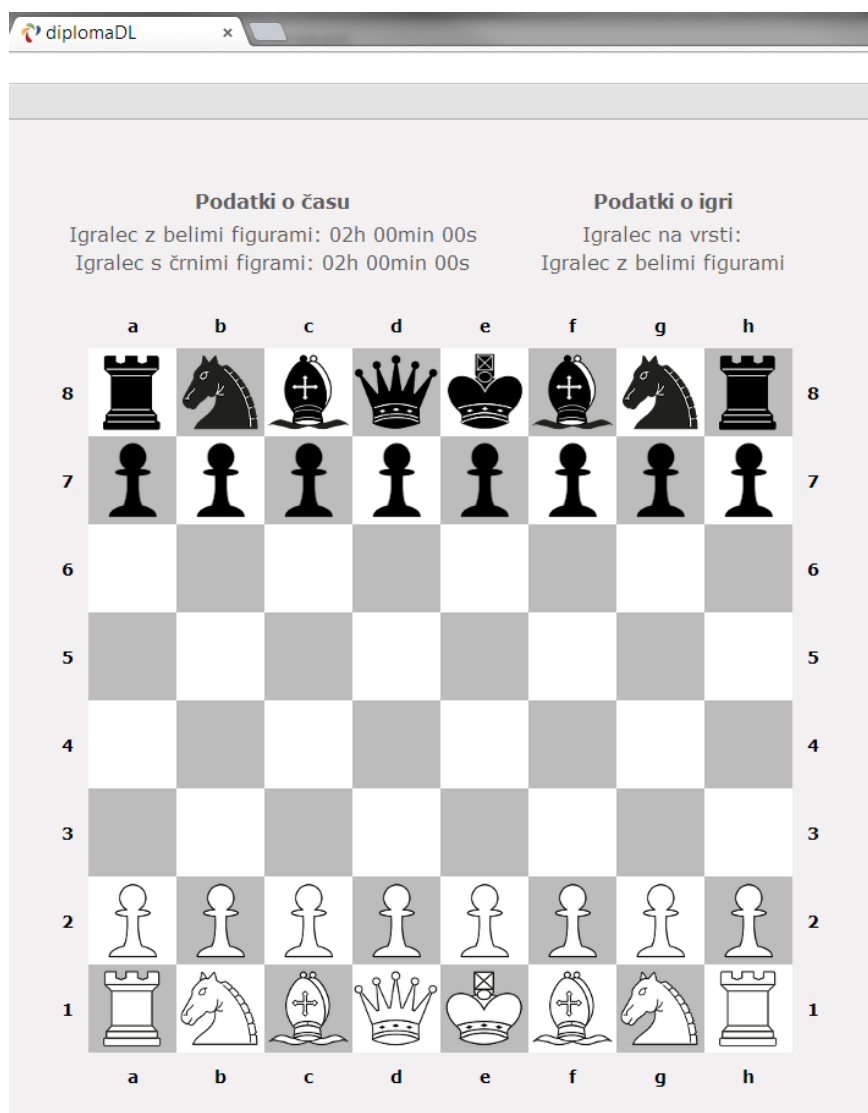
Programski jezik PHP vsebuje odjemalca SOAP (*angl. SoapClient*), ki mu nastavimo samo spletni naslov do datoteke WSDL. Klic metode v programskem jeziku PHP prikazuje slika 4.14.

```

$client = new SoapClient("http://bl4cky-hp:9192/ws.asmx?wsdl");
$stanjeSahovnica = $client->getPolje()->getPoljeResult;
    
```

Slika 4.14: Klic metode spletne storitve iz spletne aplikacije

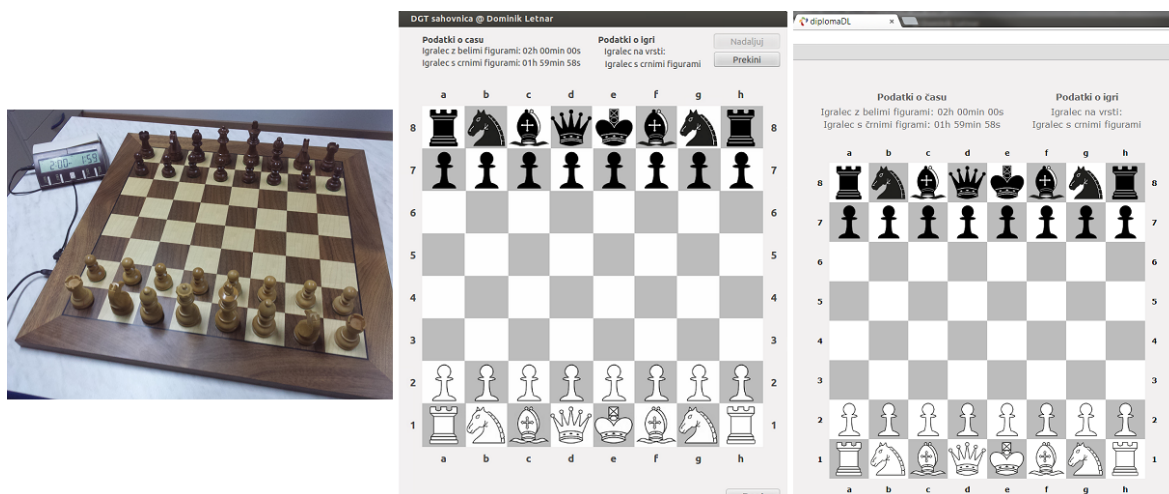
Grafični vmesnik spletne aplikacije je prikazan na sliki 4.15. Prikazuje stanje figur na šahovnici DGT in nevidne informacije.



Slika 4.15: Grafični vmesnik spletne aplikacije

5 PREIZKUŠANJE

Preizkušanje naše celotne aplikacije vključuje povezavo elektronske šahovnice DGT z računalnikom preko protokola USB, zagon namizne aplikacije, zagon spletne storitve in zagon spletne aplikacije. Ko smo se povezali s šahovnico, je namizna aplikacija prejela trenutno stanje igre. Podatki so bili prikazani v grafičnem vmesniku namizne aplikacije in prav tako tudi na spletni aplikaciji, kot prikazuje slika 5.1.



Slika 5.1: Grafični prikaz namizne in spletne aplikacije

Povezava je bila uspešna, podatki so bili osveženi, na tej točki smo lahko bili prepričani, da vse skupaj deluje. Ob spremembi sekund na uri šahovnice smo hkrati spremljali namizno aplikacijo, ki je prikazovala odštevanje sekund. Preverjali smo še, če odštevanje sekund pravilno deluje tudi na spletni aplikaciji. Namizna in spletna aplikacija sta prikazovali odštevanje sekund, kot se je to dogajalo na sami uri, priključeni na šahovnico.

Odigrali smo nekaj potez, preklapljali uro in ugotovili, da osveževanje ter prikazovanje deluje. Slika 5.2 prikazuje vmesno stanje na šahovnici, ko je bil na vrsti igralec z belimi figurami.



Slika 5.2: Vmesno stanje, ko je na potezi igralec z belimi figurami

Slika 5.3 prikazuje stanje, ko smo izvedli premik lovca in jemanje skakača.



Slika 5.3: Premik in izguba figure

Preizkušanje smo zaključili po večkratnem testiranju in tudi več odigranih partijah, kjer so uporabniki spremljali partijo na spletni aplikaciji. Med samim preizkušanjem nismo naleteli na nobeno težavo. Vse je delovalo, kot je bilo potrebno in kot je bilo načrtovano, zato smo preizkušanje uspešno zaključili.

5.1 Testiranje ob prekinitvi povezave

Testirali smo tudi, kaj se zgodi z našo aplikacijo ob prekinitvi internetne povezave, saj v tem primeru namizna aplikacija nima dostopa do spletne storitve. Ta način testiranja smo izvedli tako, da smo med premikanjem figur izklopili usmerjevalnik (*angl. router*). Po izklopu usmerjevalnika smo izgubili internetno in tudi lokalno povezavo. Po uspešni prekinitvi smo nato usmerjevalnik priklopili in počakali, da se je ponovno vzpostavila lokalna in internetna povezava.

Pri tej obliki testiranja smo imeli dva možna izida – neuspešno ter uspešno. Neuspešno smo označili izid, kjer je prišlo do napake pri izvajanju aplikacije. Uspešno pa smo označili izid, ko se je po ponovni vzpostavitvi povezave stanje igre na šahovnici uspešno posodobilo tudi na spletni aplikaciji. Večkrat smo ponovili ta način testiranja, rezultate bomo predstavili v tabeli 5.1.

Tabela 5.1: Rezultati testiranja

Izklop (x-krat)	Neuspešno	Uspešno
20	16	4

V 80 % aplikacija ni uspešno obnovila stanje igre oz. kar koli se je že zgodilo. Program se je zaustavil in pojavila se je napaka, ki je nismo pričakovali in za njo nismo ustrezno poskrbeli. V 20 % je aplikacija uspešno nadaljevala izvajanje brez težav in posodobil podatke, kot smo pričakovali.

Po prekinitvi internetne povezave namizna aplikacija ni mogla posredovati podatkov spletni storitvi in se je program zaustavil, ker nismo obravnavali vseh izjem. To napako smo odpravili tako, da smo ulovili izjeme in preprečili zaustavitev programa. Ob vsaki spremembi stanja igre ali časa ure, šahovnica posreduje podatke namizni aplikaciji in ta poskuša ponovno posredovati podatke spletni storitvi. Ob vzpostavitvi omrežja nato namizna aplikacija uspešno posreduje podatke spletni storitvi in aplikacija deluje pravilno ter nemoteno naprej. Po ponovnem testiranju se je aplikacija v 100% uspešno nadaljevala.

6 SKLEP

Cilj diplomske naloge, ki smo si ga zadali, smo uspešno dosegli. Implementirali smo namizno in spletno aplikacijo ter spletno storitev. Menimo, da je izdelek dober in uporaben. Namizna in spletna aplikacija delujeta odlično in odzivno, saj ob premiku oz. spremembi na šahovnici takoj opazimo spremembo tudi v namizni in spletni aplikaciji. Ljubitelji šaha lahko spremljajo partije na oddaljeni lokaciji in tukaj prihranijo denar, saj nimajo nobenih stroškov s potjo, z namestitvijo in podobno, hkrati pa prihranijo čas, saj ni potrebno nikamor potovati.

V nadaljnjem delu bi lahko dodali še beleženje potez, prikaz izgubljenih figur igralca, nadgradili bi grafični uporabniški vmesnik. Naredili bi ga še privlačnejšega in uporabnejšega. S temi posodobitvami bi naredili izdelek, ki bi bil na višjem nivoju in bi lahko zelo resno konkurirali že obstoječim izdelkom.

V prihodnosti bi lahko celotno idejo ter aplikacijo razširili tako, da bi ustvarili še mobilno aplikacijo, ki bi prav tako dostopala do spletne storitve in bi sledili konceptu, ki smo ga že uporabili.

Prepričani smo, da smo naredili kvalitetno namizno in spletno aplikacijo ter spletno storitev. S tem smo dosegli, da je naša aplikacija zelo uporabna za ljubitelje šaha. Tako smo dosegli želene in zastavljene cilje.

VIRI

- [1] About ChessX. Dostopno na: <http://chessx.sourceforge.net/node/5> (17. 7. 2014).
- [2] AJAX Tutorial. Dostopno na: <http://www.w3schools.com/ajax/default.ASP> (21. 7. 2014).
- [3] AJAX (programiranje) – Wikipedija, prosta enciklopedija. Dostopno na: [http://sl.wikipedia.org/wiki/Ajax_\(programiranje\)](http://sl.wikipedia.org/wiki/Ajax_(programiranje)) (21. 7. 2014).
- [4] Arena Chess GUI 3.5 – Welcome to Arena. Dostopno na: http://www.playwitharena.com/?Welcome_to_Arena (17. 7. 2014).
- [5] a206_1280.png (1280x978). Dostopno na: http://www.playwitharena.com/common/pic/a206_1280.png (17. 7. 2014).
- [6] Blanchette J., Summerfield M. *C++ GUI Programming with Qt 4, druga izdaja*. United States of America: Trolltech, 2006.
- [7] C Sharp (programming language) – Wikipedia, the free encyclopedia. Dostopno na: http://en.wikipedia.org/wiki/C_Sharp_%28programming_language%29 (21. 7. 2014).
- [8] Chess piece – Wikipedia, the free encyclopedia. Dostopno na: http://en.wikipedia.org/wiki/Chess_piece (23. 7. 2014).
- [9] chessx06-linux.png (848x725). Dostopno na: <http://chessx.sourceforge.net/images/chessx06-linux.png> (17. 7. 2014).
- [10] C++ - Wikipedija, prosta enciklopedija. Dostopno na: <http://sl.wikipedia.org/wiki/C%2B%2B> (20. 7. 2014).
- [11] dgtnix project homepage. Dostopno na: <http://dgtnix.sourceforge.net> (19. 7. 2014).
- [12] eDavki – Pogosta vprašanja. Dostopno na: <https://edavki.durs.si/OpenPortal/Pages/Faq/FaqPreview.aspx?id=29> (21. 7. 2014).
- [13] Free Vector Graphic: Chess, Chess ICons, Shah of Persian – Free Image on Pixabay – 335030. Dostopno na: <http://pixabay.com/en/chess-chess-icons-shah-of-persian-335030> (23. 7. 2014).
- [14] gSOAP – Wikipedia, the free encyclopedia. Dostopno na: <http://en.wikipedia.org/wiki/GSOAP> (25. 7. 2014).
- [15] IBM Knowledge Center – Generiranje datoteke WSDL (Web Services Description Language). Dostopno na: http://www-01.ibm.com/support/knowledgecenter/SSALK7_7.5.1/com.ibm.mbs.doc/wsregistry/c_wsd_genaratie.html?cp=SSALK7_7.5.1%2F0-15-2-3-4&lang=sl (24. 7. 2014).

- [16] Kaj je SOAP, Presentia d.o.o. baza znanja. Dostopno na: <http://www.presentia.si/baza-znanja-helpdesk/2010/kaj-je-soap> (21. 7. 2014).
- [17] Lerdorf R., MacIntyre P., Tatroe K. *Programming PHP, tretja izdaja*. United States of America: Gravenstein Highway North, Sebastopol, O'Reilly Media, 2013.
- [18] Molkenitin D. *The Book of Qt 4, The Art of Building Qt Applications*. Munich, Germany: Open Source Press GmbH, 2006.
- [19] MySQL – Wikipedija, prosta enciklopedija. Dostopno na: <http://sl.wikipedia.org/wiki/MySQL> (24. 7. 2014).
- [20] PHP: Hypertext Preprocessor. Dostopno na: <http://php.net> (21. 7. 2014).
- [21] PHP – Wikipedija, prosta enciklopedija. Dostopno na: <http://sl.wikipedia.org/wiki/PHP> (21. 7. 2014).
- [22] Šahovska figura – Wikipedija, prosta enciklopedija. Dostopno na: http://sl.wikipedia.org/wiki/%C5%A0ahovska_figura (23. 7. 2014).
- [23] Qt – Cross-platform application and UI development framework. Dostopno na: <http://qt.digia.com> (19. 7. 2014).
- [24] Qt – Qt application development and device creation. Dostopno na: <http://qt.digia.com/Why-Qt> (19. 7. 2014).
- [25] XBoard – GNU Project – Free Software Foundation. Dostopno na: <http://www.gnu.org/software/xboard> (17. 7. 2014).
- [26] xboard-4.4.0-petite.png (626x449). Dostopno na: <http://www.gnu.org/software/xboard/graphics/xboard-4.4.0-petite.png> (17. 7. 2014).



Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko
Smetanova ulica 17
2000 Maribor, Slovenija



**IZJAVA O ISTOVETNOSTI TISKANE IN ELEKTRONSKE VERZIJE ZAKLJUČNEGA
DELA IN OBJAVI OSEBNIH PODATKOV DIPLOMANTOV**

Ime in priimek avtorja-ice: DOMINIK LETNAR
Vpisna številka: E1042674
Študijski program: UN ŠP RAČUNALNIŠTVO IN INFORMACIJSKE TEHNOLOGIJE
Naslov zaključnega dela: GRAFIČNI UPORABNIŠKI VMESNIK
ZA ŠAHOVNICO DGT
Mentor: DOC. DR. BORKO BOŠKOVIC
Somentor: RED. PROF. DR. JANEZ BREST

Podpisani-a DOMINIK LETNAR izjavljam, da sem za potrebe arhiviranja oddal elektronsko verzijo zaključnega dela v Digitalno knjižnico Univerze v Mariboru. Zaključno delo sem izdelal-a sam-a ob pomoči mentorja. V skladu s 1. odstavkom 21. člena Zakona o avtorskih in sorodnih pravicah dovoljujem, da se zgoraj navedeno zaključno delo objavi na portalu Digitalne knjižnice Univerze v Mariboru.

Tiskana verzija zaključnega dela je istovetna z elektronsko verzijo elektronski verziji, ki sem jo oddal za objavo v Digitalno knjižnico Univerze v Mariboru.

Zaključno delo zaradi zagotavljanja konkurenčne prednosti, varstva industrijske lastnine ali tajnosti podatkov naročnika: _____ ne sme biti javno dostopno do _____ (datum odloga javne objave ne sme biti daljši kot 3 leta od zagovora dela).

Podpisani izjavljam, da dovoljujem objavo osebnih podatkov, vezanih na zaključek študija (ime, priimek, leto in kraj rojstva, datum zaključka študija, naslov zaključnega dela), na spletnih straneh in v publikacijah UM.

Datum in kraj: MARIBOR, 27.08.2014 Podpis avtorja-ice: Letnar Dominik

Podpis mentorja: _____
(samo v primeru, če delo ne sme biti javno dostopno)

Podpis odgovorne osebe naročnika in žig: _____
(samo v primeru, če delo ne sme biti javno dostopno)



Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko
Smetanova ulica 17
2000 Maribor, Slovenija



IZJAVA O USTREZNOSTI ZAKLJUČNEGA DELA

Podpisani mentor :

Borko Boškovič

(ime in priimek mentorja)

in somentor (eden ali več, če obstajata):

Janez Brest

(ime in priimek somentorja)

Izjavljam (-va), da je študent

Ime in priimek: Dominik Letnar

Št. indeksa: E1042674

Na programu: UN ŠP RAČUNALNIŠTVO IN
INFORMACIJSKE TEHNOLOGIJE

izdelal zaključno delo z naslovom:

GRAFIČNI UPORABNIŠKI VMESNIK ZA ŠAHOVNICO DGT

(naslov zaključnega dela v slovenskem in angleškem jeziku)

GRAPHICAL USER INTERFACE FOR DGT CHESSBOARD

v skladu z odobreno temo zaključnega dela, Navodilih o pripravi zaključnih del in mojimi (najinimi oziroma našimi) navodili.

Preveril (-a, -i) in pregledal (-a, -i) sem (sva, smo) poročilo o plagiatstvu.

Datum in kraj: 27. 08. 2014

Podpis mentorja: Borko

Datum in kraj: 27. 08. 2014

Podpis somentorja (če obstaja):

Janez Brest



Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko
Smetanova ulica 17
2000 Maribor, Slovenija



IZJAVA O AVTORSTVU

Spodaj podpisani/-a DOMINIK LETNAR

z vpisno številko E1042674

sem avtor/-ica diplomskega dela z naslovom: _____

GRAFIČNI UPORABNIŠKI VMESNIK

ZA ŠAHOVNICO DGT

(naslov diplomskega dela)

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom (naziv, ime in priimek)

DOC. DR. BORKO BOŠKOVIC

in somentorstvom (naziv, ime in priimek)

RED. PROF. DR. JANEZ BREST

- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela.
- soglašam z javno objavo elektronske oblike diplomskega dela v DKUM.

V Mariboru, dne 27.08.2014

Podpis avtorja/-ice:

Letnar Dominik