

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Branko Pinter

RAZVOJ APLIKACIJE ZA POZICIONIRANJE  
INŠTRUMENTOV NA PLOŠČE ULTRALAHKEGA  
LETALA

DIPLOMSKO DELO UNIVERZITETNEGA ŠTUDIJA

Ljubljana, 2008

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Branko Pinter

RAZVOJ APLIKACIJE ZA POZICIONIRANJE  
INŠTRUMENTOV NA PLOŠČE ULTRALAHKEGA  
LETALA

DIPLOMSKO DELO UNIVERZITETNEGA ŠTUDIJA

Mentor:  
doc. dr. Marko Bajec

Ljubljana, 2008

## **Zahvala**

Zahvaljujem se mentorju, doc. dr. Marku Bajcu za vodenje, vse nasvete, pripombe in pregled diplomskega dela. Posebna zahvala gre družini in vsem bližnjim, ki so me tekom študija spodbujali in mi stali ob strani.

Zahvaljujem se tudi podjetju Pipistrel iz Ajdovščine. Hvala, da smo lahko sodelovali skupaj.

# Kazalo

1. Uvod.....	3
2. Pipistrel.....	4
2.1. Proizvodnja letal.....	5
2.2. Pristop k problemu.....	7
3. Analiza in načrtovanje aplikacije.....	9
3.1. Poslovni procesi .....	9
3.1.1. Urejanje šifrantov plošč .....	10
3.1.2. Urejanje šifrantov skupine inštrumentov .....	11
3.1.3. Urejanje šifrantov inštrumentov .....	12
3.1.4. Urejanje šifrantov predlog.....	13
3.2. Model primerov uporabe .....	14
3.2.1. Diagram primerov uporabe.....	14
3.2.2. Opisi primerov uporabe.....	15
3.2.2.1. Urejanje šifrantov plošč .....	15
3.2.2.2. Urejanje šifrantov skupine inštrumentov .....	18
3.2.2.3. Urejanje šifrantov inštrumentov .....	20
3.2.2.4. Urejanje šifrantov predlog.....	22
3.3. Načrtovanje podatkovne baze .....	26
3.3.1. Konceptualni podatkovni model .....	27
3.3.2. Logični podatkovni model .....	29
3.3.2.1. Relacija plate.....	29
3.3.2.2. Relacija type.....	29
3.3.2.3. Relacija shape .....	30
3.3.2.4. Relacija instrument .....	30
3.3.2.5. Relacija pos_on_plate.....	31
3.4. Načrtovanje uporabniškega vmesnika .....	32
3.4.1. Dizajn glavnega menija .....	32
3.4.2. Dizajn urejanja šifrantov plošč .....	33
3.4.3. Dizajn urejanja šifrantov skupin inštrumentov .....	33
3.4.4. Dizajn urejanja inštrumentov .....	34
3.4.5. Dizajn urejanja predlog .....	35
4. Aplikacija .....	37
4.1. Splošno o aplikaciji.....	37
4.1.1. Inštalacija aplikacije .....	37
4.1.2. Prenosljivost aplikacije .....	37
4.2. Opis posameznih zaslonskih mask .....	37
4.2.1. Zaslonska maska glavnega menija.....	37
4.2.2. Zaslonska maska plošče .....	38
4.2.3. Zaslonska maska skupine inštrumentov .....	39
4.2.4. Zaslonska maska inštrumenti.....	40
4.2.5. Zaslonska maska predloge.....	42
4.3. Izzivi pri izdelavi aplikacije.....	45
4.3.1. Izrisovanje vektorske slike na ploščo.....	45
4.3.2. Določanje razmerij .....	47
5. Sklepne ugotovitve.....	49

<b>6. Priloge.....</b>	<b>50</b>
<b>6.1. Inštrumenti.....</b>	<b>50</b>
<b>6.1.1. Brauniger.....</b>	<b>50</b>
<b>6.1.2. Radijska postaja .....</b>	<b>51</b>
<b>6.1.3. Ostali inštrumenti.....</b>	<b>51</b>
<b>7. Viri .....</b>	<b>53</b>

## **Seznam uporabljenih kratic**

WMF – windows metafile, format vektorske slike

PNG – format bitne slike

CNC stroj – računalniško voden stroj, ki ga lahko programiramo

.NET – Microsoft .NET Framework

ZM – zaslonska maska

## **Povzetek**

Diplomsko delo obravnava analizo, načrtovanje in izdelavo aplikacije za podjetje Pipistrel, ki se ukvarja z izdelavo ultralahkih letal. Namen aplikacije je poenostaviti, pospešiti in s tem posledično tudi zmanjšati stroške izdelave letal. Aplikacija za izboljšanje se nanaša le na del proizvodnega procesa, kjer postavljajo letalske inštrumente (v nadaljevanju inštrumenti) na armaturno ploščo letala (v nadaljevanju plošča).

Ker so v podjetju ta del proizvodnega procesa izvajali povsem ročno in niso uporabljali nobene aplikacije, je bila izbira programskega jezika in razvojnega okolja prepuščena meni. Glede na današnjo izbiro tehnologij in orodij za razvoj aplikacij sem se odločil za precej razširjeno razvojno okolje Microsoft Visual Studio 2008. Za kodiranje aplikacije sem imel na voljo več programskih jezikov, a sem se odločil za C#, ker se dandanes veliko uporablja in sem se v preteklosti že soočil z njim. Za shranjevanje podatkov sem uporabil podatkovno bazo Microsoft Access.

Uvodni del diplomskega dela vsebuje kratko predstavitev podjetja, s katerim smo sodelovali in opis pristopa k njihovem problemu. Opisal sem analizo in načrtovanje aplikativnega sistema. V ta del spadajo poslovni procesi, diagrami primerov uporabe z opisi posameznih primerov uporabe, načrtovanje podatkovne baze in načrtovanje uporabniškega vmesnika. Četrto poglavje vsebuje rezultate dela, kjer so prikazane zaslonske maske aplikacije in problemi, na katere sem naletel pri razvoju. Sledijo še sklepne ugotovitve z opisom možnega nadaljnjega razvoja aplikacije in opisi nekaterih inštrumentov.

## **Ključne besede**

Razvoj aplikacije, ultralahka letala, model primerov uporabe, poslovni proces, konceptualni podatkovni model.

## **Abstract**

The thesis deals with the analysis, planning and development of an application for company Pipistrel, which makes ultra-light planes. The application is to simplify, speed up and consequently reduce the expenses of the production process. It however aims to improve only the part of the production process where the aircraft instruments (henceforth instruments) are being placed on armature plates (henceforth plates).

This part of the production process was done manually so far and no application was used, so the choice of a program language and development environment was entirely mine. Considering the offer of the current technology and application development environment, I opted for a widely used development tool Microsoft Visual Studio 2008. I also had several programming languages to choose from, and decided to use C#, because it is often used today and I already worked with it in the past. For saving data I chose the Microsoft Access database.

The introductory part of the thesis includes a short presentation of the company I cooperated with and a description of our approach to the problem. I described the analysis and the planning of the application system. This includes business processes, diagrams illustrating the use with descriptions of individual examples of use, database planning and user interface planning. Chapter 4 shows the results of my work - the application forms and the problems I came across during the development of the application. The final part of the thesis features my conclusions and a description of a possible further development of the application and descriptions of specific instruments.

## **Key words**

Application development, ultra-light planes, use case model, business processes, conceptual data model.

## 1. Uvod

Predvidevam, da je že vsak izmed vas kdaj videl ploščo ultralahkega letala. Letala imajo na armaturni plošči veliko inštrumentov, ki pilotu posredujejo informacije, katere so ključne za varno letenje. Inštrumenti, ki prikazujejo osnovne podatke o stanju letala, kot je na primer hitrost, višina dviganja/spuščanja, količina goriva, temperatura motorja in podobno, so obvezni za vsako letalo. Obstaja pa tudi množica manj pomembnih inštrumentov, ki so lahko v letalu po želji in izbiri pilota.

Inštrumenti morajo biti razporejeni pregledno, da pilot takoj dobi ustrezno informacijo. Delijo se v več skupin. Letalni, navigacijski, komunikacijski in drugi.

Vsaka skupina inštrumentov ima priporočeno lokacijo na plošči. Kljub temu, da se tega ne prakticira, si pilot lahko sam izbere lokacije inštrumentov.

V podjetju Pipistrel so do sedaj inštrumente postavljali ročno. Ob nakupu letala se je stranka odločala pri izbiri postavitev. Inštrumente je postavilo podjetje ali pa se je stranka sama odločila kako naj bi plošča izgledala. Če je razporejalo podjetje so stranki ponudili že izdelane primere, v nasprotnem primeru pa so se dogovorili za srečanje in skupaj izbrali pozicije in inštrumente. Ker ima vsako letalo drugačno ploščo, mora stranka najprej povedati kakšno letalo želi. Na izbrano ploščo so fizično postavljali inštrumente, jih risali na papir in potem glede na željo stranke in priporočila strokovnjakov, na papirju oziroma fizično popravljali pozicije inštrumentov. Ko so prišli do primerne postavitve, se je lahko začel proces same proizvodnje. Če so želeli narediti ploščo, so morali dobiti točne koordinate posameznega elementa na plošči, kar pa je precej zahtevno, če upoštevamo, da se večina inštrumentov razlikuje po velikosti. Inštrumente je bilo potrebno vedno znova postavljati na prazno ploščo in zaradi tega je bilo veliko preračunavanja koordinat. Vedeti so morali v katero skupino spada kakšen inštrument ter kakšna je priporočljiva lokacija za posamezno skupino. Ker je bilo delo zamudno in nerodno, so se odločili za razvoj aplikacije, ki bi simulirala postavitev inštrumentov in jim s tem olajšala delo. Poenostavila naj bi postavljanje inštrumentov in pridobitev koordinat brez veliko preračunavanja.

Na začetku je potrebno v aplikacijo vnesti vse inštrumente in plošče ter njihove podatke, kar predstavlja nekaj dodatnega dela, vendar to odtehta veliko lažje postavljanje inštrumentov. Prednost aplikacije je, da prodajalcu ni potrebno vedno znova postavljati inštrumentov od začetka, ampak lahko uporabi že vnaprej pripravljene predloge in jih spreminja. Po končanem postavljanju imamo določene vse koordinate inštrumentov na plošči, zato ni potrebnega dodatnega računanja. Takoj se lahko lotimo izdelovanja plošče. Z aplikacijo lahko elegantno rešimo probleme in prihranimo veliko časa.

## 2. Pipistrel

Beseda Pipistrel izvira iz latinščine in pomeni netopir. Pipistrel je podjetje s sedežem v Ajdovščini, ki se ukvarja z izdelavo ultralahkih letal. Zgodovina podjetja sega v leto 1982, ko sta pilota Ivo Boscarol in Bojan Sajovic začela iskati nove načine letenja. Stopila sta v kontakt s tedaj znanim pilotom, Salvatore Stefanom. Navdušil ju je za alternativne načine letenja, kot so letenje z zmaji, ultralahkimi letali in podobno. Nadaljnjih nekaj let, sta Ivo Boscarol in Bojan Sajovic raziskovala nove načine letenja in izdelala tudi nekaj poskusnih letal.

Leta 1987 je Ivo Boscarol uradno dobil licenco za izdelovanje manjših letal in s tem postal prvi Jugoslovanski proizvajalec ultralahkih letal.

1988 leta je dobil licenco za pilota zmajev in pilota jadralnih letal. Prva serijska proizvodnja zmajev z imenom »Ajdovščina«, se je začela leta 1989 in se kasneje preimenovala v »Basic«. Izdelanih je bilo 32 zmajev in prvi izmed njih je bil izvožen v Italijo. Leto kasneje so izdelali zmaja »Plus«, katerega glavni konstruktor je bil Bojan. Do leta 2000 so prodali preko 200 zmajev »Plus«. Izdelujejo jih še danes. Leta 1990 je bil prvi zmaj prodan v Afriko. Istega leta je v Ajdovščini podjetje Pipistrel organiziralo prvo tekmovanje z zmaji, imenovano »Pipistrel cup«. Udeleženih je bilo 18 tekmovalcev. Pipistrel je prejel tudi nagrado za inovacije v tem letu.

Leta 1991 je bil izdelan prvi vodni zmaj, ki je lahko vzletal in pristajal le na vodi. Teh vrst zmajev je bilo izdelanih zelo malo, ker je bil glavni trg Hrvaška in se je prodaja zaradi takratne vojne ustavila.

Izdelali so nekaj posebnih zmajev namenjenih za hrvaško vojsko. To so bila prva tovrstna bojna letala v tej državi. Začeli so s proizvodnjo tri in štiri krakih propelerjev.

V letu 1992 so začeli izdelovati izboljšano različico zmajev. Začeli so proizvajati ultralahka jadralna letala. Podjetje Pipistrel d.o.o., ki sta ga ustanovila Ivo Boscarol in Vida Lorbek je bilo uradno ustanovljeno leta 1992.



Slika 1: Glavni vhod podjetja Pipistrel

Leta 1994 so začeli z izdelovanjem prvih ultralahkih letal imenovanih »Sinus«. Javno je bil predstavljen leto zatem v Nemčiji. Zanj je bilo veliko zanimanja. Prodaja zmajev se je vsako leto širila v vedno več evropskih držav.

Leta 1997 so začeli z razvojem nove serije ultralahkih letal imenovanih »Virus«. Dve leti zatem je poletelo prvo tovrstno letalo, ki ga je preizkusil večkratni svetovni prvak v letenju, Philippe Zen. Nad letalom je bil zelo navdušen, zato se je odločil, da bo zastopal prodajo le teh v Franciji. Leto je bilo uspešno, saj je bilo prodanih 32 »Virusov«.

Leta 2000 so začeli s serijsko proizvodnjo »Virusov« in hkrati tudi z razvojem novih ultralahkih letal, ki so lahko dosegali hitrost do 380km/h. Ker se je povpraševanje po vseh vrstah letal zelo povečalo, so bili prisiljeni povečati proizvodnjo. Začeli so z gradnjo novih prostorov na letališču v Ajdovščini. V naslednjem letu so začeli z razvojem revolucionarnega jadralnega letala »Taurus«. Razpon kril je meril 15 metrov, vanj pa sta lahko sedli dve osebi. Posebnost tega jadralnega letala je bila, da je lahko sam vzletel. To jadralno letalo je vzletelo prvič v letu 2004. Takrat je podjetje začelo tudi s prodajo letal v Ameriko, Avstralijo, Tajvan in še nekatere oddaljene države. Tega leta je z letalom Sinus 912, Matevž Lenarčič poletel okoli sveta.

Leta 2006 so začeli z razvojem prvega električnega Taurusa na svetu, ki ga poganja elektromotor.

V letu 2007 in 2008 je letalo Virus 912 dvakrat zapored zmagalo na mednarodnem tekmovanju v organizaciji Nase. Letalo je premagalo vso ameriško in evropsko konkurenco, v osmih od dvanajstih kategorij. V vseh letih od začetka je podjetje prejelo veliko priznanj in nagrad. Podjetje še danes zelo uspešno posluje.

## 2.1. Proizvodnja letal

Proizvodnja letal v podjetju se začne z razvojem. Za tem je potrebno pripraviti kalupe kril in trupov, ki jih dostavijo kooperantom. Nato kooperanti iz kalupov izdelajo krila, trupe in druge plastične dele letal. Vse dele, ki jih podjetje dobi od kooperantov je potrebno tudi prebarvati. Barvanje se vrši v posebnem prostoru.



Slika 2: Prostor za barvanje delov letala

Ko imajo pripravljene vse dele, postopoma pričnejo s sestavljanjem letala. Da je letalo čim hitreje sestavljeno, imajo proizvodno linijo, kjer lahko sestavljajo več letal hkrati. Tako pridobijo na času.



**Slika 3: Proizvodna linija**

Med procesom proizvodnje je potrebno pripraviti armaturno ploščo, ki jo vgradijo v letalo. Načrtujejo postavitev inštrumentov na armaturno ploščo in ročno izrežejo luknje ter vstavijo želene inštrumente v njo. V prihodnosti bodo v podjetju Pipistrel uporabljali aplikacijo, ki sem jo razvil, za postavitev teh inštrumentov na ploščo. S tem bo ta proces bolj optimiziran.



**Slika 4: Armaturna plošča letala**

Ko je letalo sestavljeno je potrebno preveriti brezhibnost. Če zadosti vsem pogojem je pripravljeno za dostavo do stranke. Pri tem je potrebno še upoštevati, da mora biti pripravljena vsa dokumentacija.



Slika 5: Izdelano letalo

## 2.2. Pristop k problemu

Z mentorjem sva obiskala podjetje v Ajdovščini. Razkazali so nama svoje prostore in predstavili nekatere proizvodne postopke. Za tem smo se lotili problema in s pogovorom izmenjali mnenja, ideje in želje.

Njihova težava je bila pri postavljanju inštrumentov na ploščo letala. Pokazali so nama ploščo in inštrumente ter kako poteka postavljanje sedaj. Postopek, ki so ga uporabljali je bil takšen, da so po dogovoru s stranko dobili seznam inštrumentov, ki jih je stranka želela na letalu. Nato so naredili primer postavitve in to pokazali stranki. Inštrumente so postavljali ročno in tako poskušali dobiti čim boljše pozicijo inštrumenta. Inštrumenti morajo biti postavljeni tako, da pilot takoj dobi dober pregled nad informacijami, ki jih ponujajo. Upoštevati je treba fizične mere inštrumentov in plošč. Vse mora biti urejeno in estetsko. Postavljanje je vzelo veliko časa, ker je bilo težko pridobiti točne koordinate. Potrebno je bilo veliko preračunavanja, da so dobili prave koordinate inštrumenta. Upoštevati so morali velikosti inštrumentov, minimalen razmak med njimi, velikost plošče in še druge pogoje. Tako smo prišli do ideje, da bi izdelali tako aplikacijo, da bi stranka izbrala inštrumente, ki jih želi na letalu. Aplikacija bi s pomočjo optimizacije sama izračunala koordinate in izrisala določeno postavitev.

Najprej bi izdelali zaslonsko masko, ki bi omogočala izbor letala (plošče). Nato bi iz seznama izbrali inštrumente. Za tem bi sledil algoritem, ki bi preučil vse pogoje in izrisal eno izmed možnih postavitvev. V nadaljevanju bi bilo omogočeno naknadno popraviljanje pozicije posameznega inštrumenta. Ko bi bili zadovoljni s postavitvijo, bi shranili postavitev kot predlogo.

Aplikacija bi dobila vektorsko sliko, s tem pa bi pridobili obliko plošče in vse njene mere. Velikosti posameznih inštrumentov bi algoritem prebral iz podatkovne baze. Preden bi

algoritem podal točno lokacijo posameznega inštrumenta, bi moral upoštevati več pogojev. Vsak inštrument spada v določeno skupino inštrumentov, kot so letalni, komunikacijski in drugi. Vsaka od teh skupin ima priporočeno lokacijo na plošči. Algoritem bi moral upoštevati priporočeno lokacijo inštrumenta in hkrati tudi strankino željo. Da bi na podlagi teh pogojev dobili primerno lokacijo, bi bilo potrebno utežiti vsak pogoj posebej in na podlagi le teh, bi dobili koordinate položaja inštrumenta. Najboljša utežitev pogoja bi se lahko hitro ugotovila, vendar pa je vprašanje, kako dobri bi bili rezultati. Boljša rešitev bi bila, da bi določili utež za vsak inštrument, za vsak pogoj posebej. S tem pa se že pojavi problem, kako poiskati optimalne uteži, da bo postavitve čim boljše. Ko bi enkrat imeli uteži, bi algoritem preveril ali so koordinate inštrumentov znotraj površine plošče in če se inštrumenti prekrivajo med sabo. Če kateri izmed teh dveh pogojev ne bi ustrezal, bi bilo potrebno popravljanje lokacije in zopet se pojavi vprašanje, kateri inštrument premakniti in v katero smer. Ko bi dobili popravljene koordinate, bi bilo na koncu potrebno upoštevati tudi estetiko postavljenih inštrumentov. Ko vse te pogoje združimo, ugotovimo, da je tovrstna postavitve zahtevna. Uspešnost te postavitve je v veliki meri odvisna od izbranih uteži.

Ker je ta način postavitve inštrumentov zahteven, nas zanima ali bo algoritem dajal zelene rezultate in ali je smiselno razvijati v tej smeri. Če algoritem ne bi dajal ustreznih koordinat, bi na koncu morali ogromno popravljati. Prišli smo do ideje, da bi lahko izdelali program, ki ne bi imel implementiranega tega algoritma, temveč bi imel na voljo izbiro plošče, na katero bi sami postavljali inštrumente. V tej verziji bi bil poudarek na prijaznem grafičnem vmesniku, ki bi omogočal enostavno postavljanje inštrumentov.

Ugotovili smo, da je najboljša rešitev maska za izris plošče in inštrumentov, z naprednim grafičnim vmesnikom. Inštrumente lahko računalniško postavlja prodajalec in prav tako kupec.

Nadaljnji razvoj aplikacije je potekal v tej smeri.

### **3. Analiza in načrtovanje aplikacije**

V fazi analize je potrebno natančno opredeliti funkcionalne in nefunkcionalne zahteve, ki jih bo aplikacija podpirala. Izdelati je potrebno podrobne procesne in podatkovne modele. Pomembno pri tem je, da uporabniki sodelujejo pri razvoju in jasno izrazijo svoje želje in zahteve.

Nato je v fazi načrtovanja potrebno izdelati načrt aplikativnega sistema, na podlagi modelov narejenih v fazi analize. Potrebno je narediti načrt podatkovne baze in načrt programskih modulov. V tej fazi dobimo odgovor na vprašanje, kako bo aplikativni sistem deloval.

#### **3.1. Poslovni procesi**

V tem poglavju so prikazani poslovni procesi, ki jih podpira aplikacija. Za prikaz procesov je uporabljena diagramska tehnika »Event-Driven Process Chain« (EPC).

Glavna funkcionalnost aplikacije bo urejanje šifrantov predlog, ki bo omogočala postavljanje inštrumentov na ploščo. Po opravljenem postavljanju bo možno predlogo shraniti ali pa izvoziti v PNG format bitne slike. Če želimo postavljati inštrumente na plošče, moramo imeti bazo, v kateri so shranjeni vsi podatki o inštrumentih in ploščah, zaradi tega pa tudi možnost urejanja šifrantov plošč in inštrumentov. Ker vsak inštrument spada v določeno skupino, potrebujemo tudi možnost urejanja šifrantov skupin.

### 3.1.1. Urejanje šifrantov plošč

Proces opisuje funkcionalnosti urejanja šifrantov plošč. Omogoča dodajanje, spreminjanje in brisanje posameznih plošč.

Najprej je potrebno odpreti zaslonsko masko, ki omogoča urejanje šifrantov plošč. Za tem izberemo želeno operacijo (dodajanje, spreminjanje, brisanje). Odpre se ustrezen obrazec, kamor je potrebno vnesti podatke. Ko so podatki vneseni, je potrebno preveriti, ali so vneseni podatki pravilni oziroma veljavni za določeno polje. Če podatek v polju ni ustrezen, se pojavi opozorilo. Imamo možnost odpravljanja napake, v nasprotnem primeru pa se pojavi obvestilo o uspešno opravljeni operaciji.



Slika 6: Proces urejanje šifrantov plošč

### 3.1.2. Urejanje šifrantov skupine inštrumentov

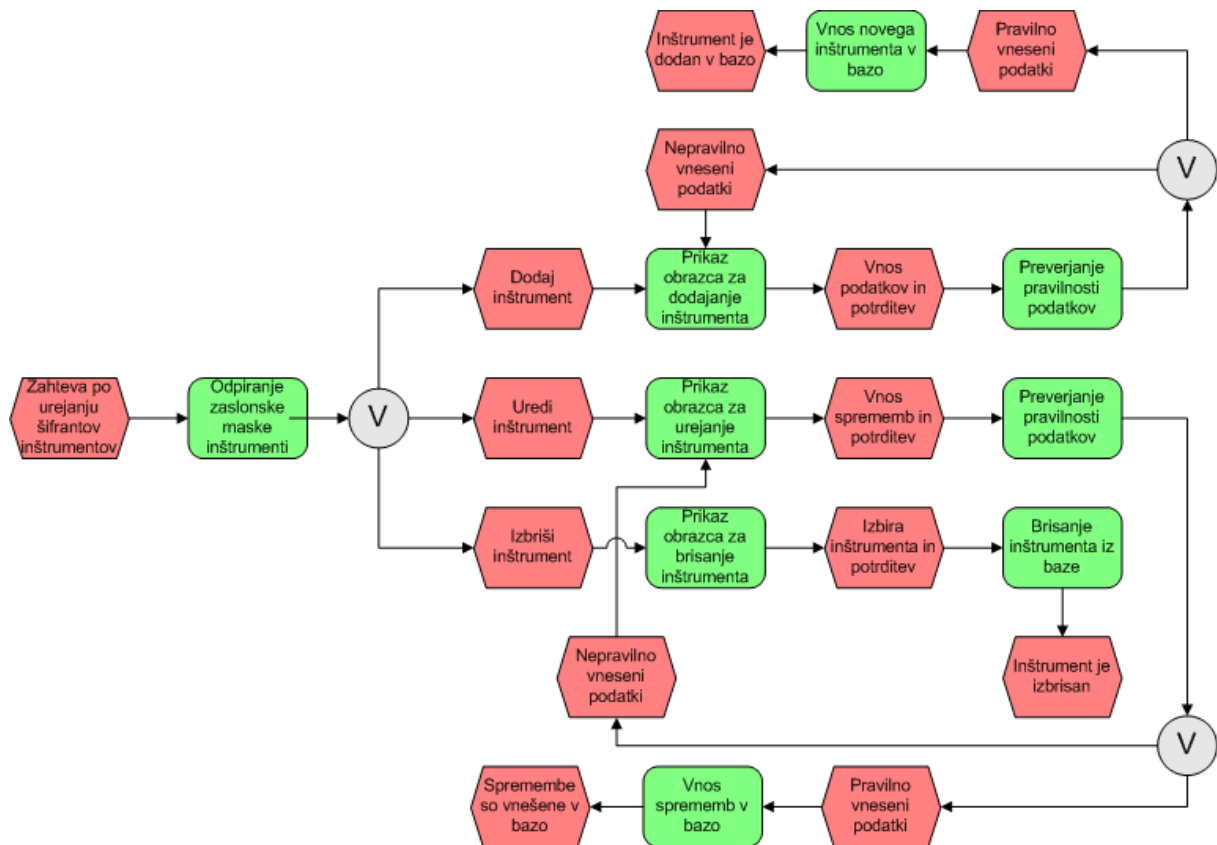
Proces opisuje postopek urejanja šifrantov skupine inštrumentov. Ko se pojavi zahteva po urejanju skupine inštrumentov odpremo zaslonsko masko. Nato izberemo, glede na našo zahtevo, dodajanje, spreminjanje ali brisanje skupine inštrumentov. Pri dodajanju in spreminjanju se pojavi ustrezen obrazec, v katerega se vnesejo podatki. Po zaključku se preveri pravilnost vnesenih podatkov. V primeru napak imamo možnost, da jih odpravimo, sicer pa se operacija uspešno izvrši. Pri brisanju skupine inštrumentov se odpre obrazec iz katerega se izbere skupina inštrumentov, ki se bo brisala. Če v bazi obstaja inštrument z izbrano skupino inštrumentov, se bo brisanje izvršilo, sicer pa brisanje skupine ni mogoče.



Slika 7: Proces urejanje šifrantov skupine inštrumentov

### 3.1.3. Urejanje šifrantov inštrumentov

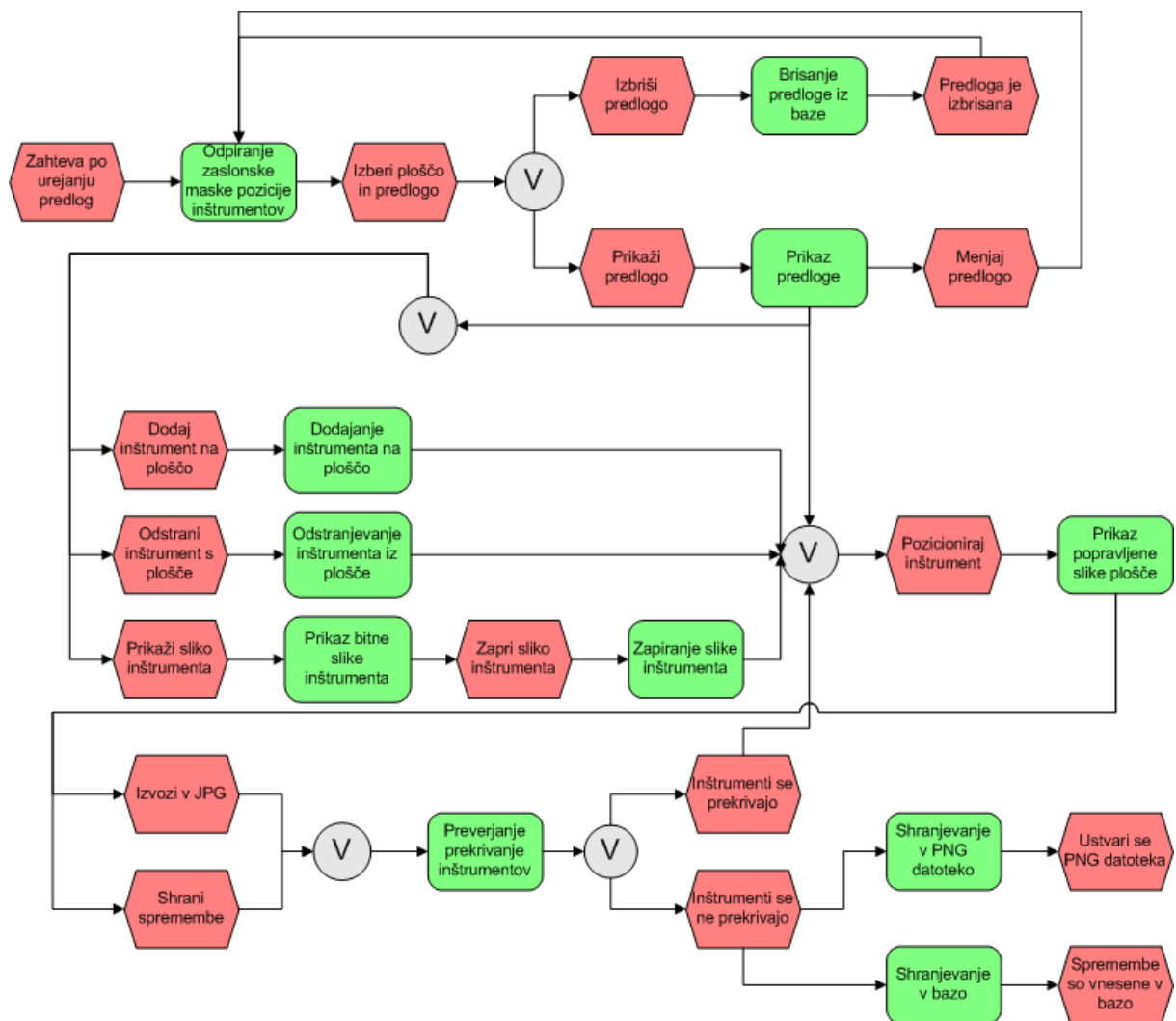
Proces opisuje postopek urejanja šifrantov inštrumentov. V začetku procesa izberemo ali želimo dodati nov inštrument, spremeniti obstoječega ali izbrisati že vnesen inštrument. Glede na našo izbiro se prikaže ustrezen obrazec z vnosnimi polji. Izpolnimo polja in potrdimo našo odločitev. Aplikacija preveri pravilnost vnesenih podatkov in v primeru, da ni napak pri podatkih, vnese spremembe v bazo. V nasprotnem primeru nam omogoči, da popravimo napake pri vnosu in zopet potrdimo. Ob pravilnem vnosu se operacija izvrši in spremembe se shranijo v podatkovno bazo.



Slika 8: Proces urejanje šifrantov inštrumentov

### 3.1.4. Urejanje šifrantov predlog

Proces opisuje postopek urejanja šifrantov predlog. Prične se z odprtjem zaslonske maske. Preko nje lahko izbrišemo obstoječe predloge, izdelamo novo ali pa spremenimo že vnesene predloge v bazi. Najprej izberemo ploščo in predlogo. Za tem imamo možnost brisanja ali spreminjanja predloge. Na voljo je tudi prikaz prazne predloge. Ko imamo predlogo na zaslonu, lahko poljubno dodajamo in odstranjujemo inštrumente iz plošče. Lahko tudi prikažemo bitno sliko inštrumenta. Ko končamo s postavljanjem, lahko shranimo spremembe v podatkovno bazo ali pa izvozimo sliko aplikacije v PNG datoteko. Po tem se preveri prekrivanje inštrumentov. Če se inštrumenti prekrivajo, se prikaže obvestilo. Za uspešen zaključek procesa moramo popraviti pozicije inštrumentov, v nasprotnem primeru pa se proces zaključi brez sprememb v podatkovni bazi.



Slika 9: Proces urejanje šifrantov predlog

## 3.2. Model primerov uporabe

Model primerov uporabe je sestavljen iz diagrama primerov uporabe ter njihovih opisov.

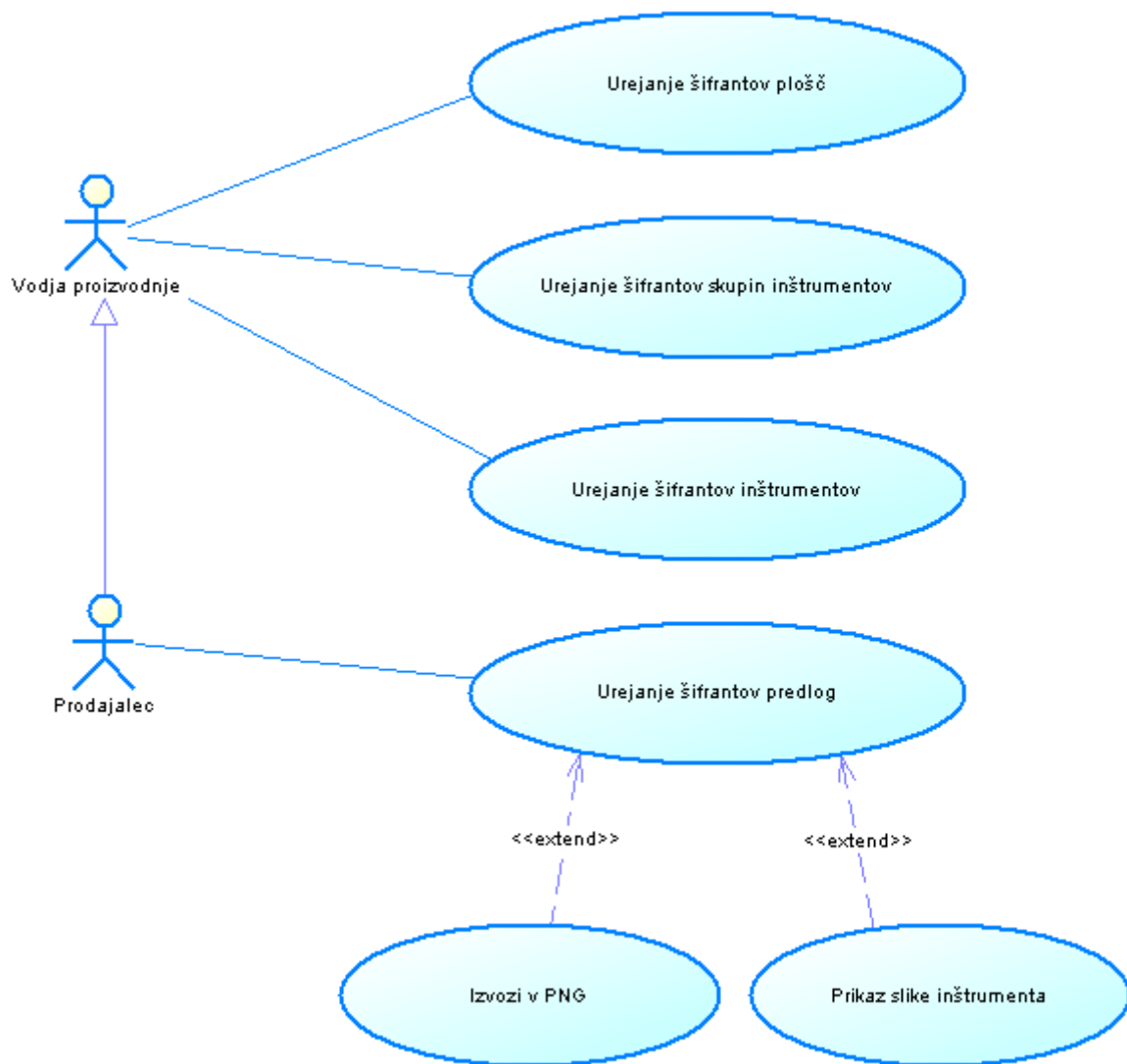
### 3.2.1. Diagram primerov uporabe

Za izdelavo diagrama primerov uporabe moramo identificirati in opisati akterje ter primere uporabe. Akterja, ki bosta sodelovala v tej aplikaciji sta vodja proizvodnje in prodajalec. Vodja proizvodnje je oseba, ki vnaša podatke o šifrantih plošč, inštrumentov in skupin inštrumentov. Prodajalec pa je zadolžen za ustvarjanje in popravljanje predlog na podlagi želja strank. Ko imamo določene akterje je potrebno definirati še primere uporabe. Iz poslovnih procesov definiramo primere uporabe. Iz procesa urejanja šifrantov plošč dobimo tri primere uporabe in sicer dodajanje plošč, urejanje plošč in brisanje plošč. Prav tako dobimo podobne primere uporabe iz procesov urejanje šifrantov inštrumentov in urejanje šifrantov skupine inštrumentov. Iz prvega dobimo primere uporabe, dodajanje inštrumentov, spreminjanje inštrumentov, brisanje inštrumentov. Iz drugega procesa pa dodajanje, spreminjanje in brisanje skupin inštrumentov. Zaradi večje preglednosti sem v celotnem diagramu primerov združil primere uporabe, kot so:

- dodaj, spremeni in izbriši plošče v primer uporabe urejanje šifrantov plošč
- dodaj, spremeni in izbriši inštrumente v urejanje šifrantov inštrumentov
- dodaj, spremeni in izbriši skupino inštrumentov v urejanje šifrantov skupine inštrumentov.

Vse omenjene funkcionalnosti opravlja vodja proizvodnje. Kadar se pojavi nov produkt v proizvodnji, mora vnesti podatke o njem in kadar kakšen produkt izločijo iz proizvodnje ga je potrebno izbrisati.

Prodajalec, ki je v kontaktu s strankami mora skrbeti za predloge. Na voljo ima izdelavo novih predlog, brisanje predlog ter spreminjanje obstoječih predlog. Tako dobimo nove primere uporabe, kot so dodaj, spremeni in izbriši predlogo, ki sem jih prav tako združil v skupen primer uporabe z imenom urejanje šifrantov predlog. Poleg samega urejanja je omogočen tudi izvoz predloge v PNG datoteko ali ažuriranje sprememb v bazi. Če upoštevamo vse zgoraj navedene akterje in primere uporabe dobimo diagram primerov uporabe, ki je prikazan na spodnji sliki.



Slika 10: Diagram primerov uporabe

### 3.2.2. Opisi primerov uporabe

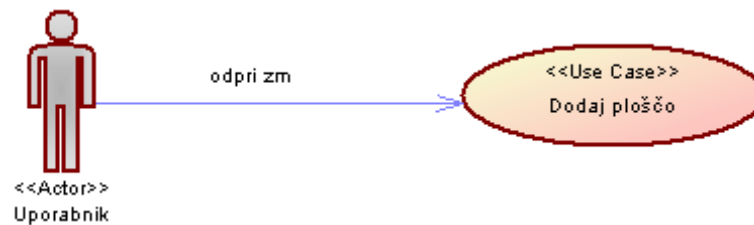
V tem poglavju bom natančno opisal vse primere uporabe in podal možne scenarije za posamezen primer uporabe ter opisal njihove pogoje in posledice. Pri vseh primerih uporabe, razen pri urejanju šifrantov predlog je kot akter naveden uporabnik, ki je lahko bodisi vodja proizvodnje, bodisi prodajalec.

#### 3.2.2.1. Urejanje šifrantov plošč

Urejanje šifrantov plošč vsebuje tri primere uporabe. Ti primeri uporabe so dodaj ploščo, uredi ploščo in izbrisi ploščo.

##### 3.2.2.1.1. Primer uporabe dodaj ploščo

Primer uporabe omogoča, dodajanje nove plošče v podatkovno bazo. Podvajanje plošč v podatkovni bazi ni dovoljeno.



Slika 11: Primer uporabe dodaj ploščo

**Osnovni tok dogodkov:**

Primer uporabe se začne z zahtevo po vnosu nove plošče.

1. uporabnik odpre zaslonsko masko plošče
2. izbere dodaj ploščo
3. vnese ustrezne podatke v obrazec in potrdi
4. sistem preveri pravilnost vnesenih podatkov
5. sistem prikaže obvestilo o uspešnem vnosu v bazo

**Alternativni tokovi dogodkov:**

- Če je katero izmed obveznih vnosnih polj prazno oziroma nepravilno izpolnjeno, sistem vrne obvestilo o napaki in ponovno prikaže obrazec
- Če ime plošče že obstaja v bazi, sistem vrne obvestilo o napaki in ponovno prikaže obrazec

**Pogoji**

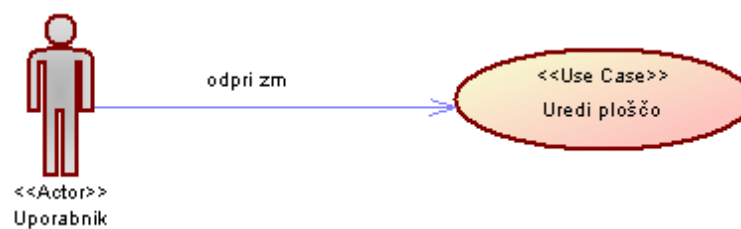
Edini pogoj je, da v bazi ne sme obstajati plošča z istim imenom, kot jo vnašamo.

**Posledice**

V podatkovni bazi je dodan zapis o plošči.

**3.2.2.1.2. Primer uporabe uredi ploščo**

Primer uporabe omogoča popravljanje podatkov o že obstoječi plošči iz podatkovne baze.



Slika 12: Primer uporabe uredi ploščo

**Osnovni tok dogodkov:**

Primer uporabe se začne z zahtevo po spremembi podatkov o plošči.

1. uporabnik odpre zaslonsko masko plošče
2. izbere uredi ploščo
3. izbere ploščo, ki jo želi urejati
4. spremeni podatke na obrazcu v ustreznih poljih in potrdi
5. sistem preveri pravilnost vnesenih podatkov
6. sistem prikaže obvestilo o uspešno opravljenih spremembah v bazi

**Alternativni tokovi dogodkov:**

- Če je katero izmed vnosnih polj prazno oziroma nepravilno izpolnjeno, sistem vrne obvestilo o napaki in ponovno prikaže obrazec
- Če je ime plošče drugačno od imena plošče, ki ga spreminjamo in že obstaja v bazi, sistem vrne obvestilo o napaki in ponovno prikaže obrazec

**Pogoji**

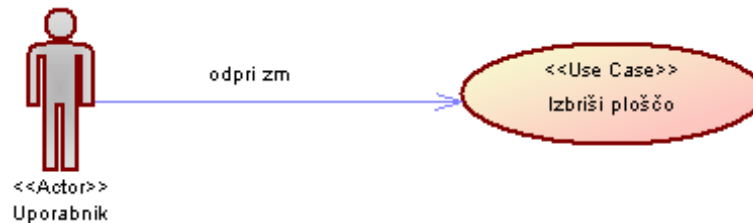
V podatkovni bazi mora obstajati vsaj en zapis o plošči.

**Posledice**

Spremenjeni podatki o plošči so vneseni v podatkovno bazo.

**3.2.2.1.3. Primer uporabe izbriši ploščo**

Primer uporabe omogoča brisanje plošče iz podatkovne baze. Ob brisanju izbriše tudi vse predloge s to ploščo.



Slika 13: Primer uporabe izbriši ploščo

**Osnovni tok dogodkov:**

Primer uporabe se začne z zahtevo po brisanju plošče.

1. uporabnik odpre zaslonsko masko plošče
2. izbere izbriši ploščo
3. izbere ploščo, ki jo želi izbrisati
4. potrdi brisanje
5. sistem izbriše izbrano ploščo iz baze
6. sistem prikaže obvestilo o uspešnem brisanju

**Alternativni tok dogodkov:**

- Če nimamo vnesene nobene plošče, sistem ne naredi ničesar ob potrditvi brisanja
- Če obstajajo predloge iz plošče, ki jo želimo brisati, nas sistem o tem obvesti in nam ponudi na izbiro: brisanje ali prekinitvev brisanja plošče

**Pogoji**

V podatkovni bazi mora obstajati vsaj en zapis o plošči.

**Posledice**

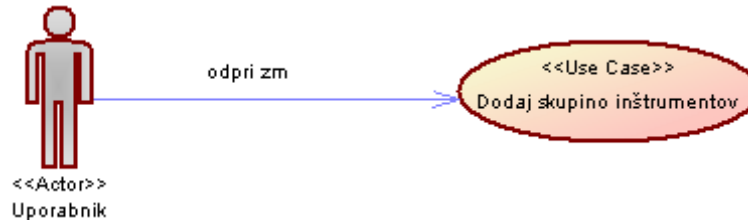
Zapis plošče je izbrisan iz podatkovne baze. Poleg tega se izbrišejo tudi vse predloge, ki so bile narejene iz te plošče.

### 3.2.2.2. Urejanje šifrantov skupine inštrumentov

Urejanje šifrantov skupine inštrumentov vsebuje tri primere uporabe. Ti primeri uporabe so dodaj skupino inštrumentov, uredi skupino inštrumentov in izbriši skupino inštrumentov.

#### 3.2.2.2.1. Primer uporabe dodaj skupino inštrumentov

Primer uporabe omogoča dodajanje novih skupin inštrumentov, ki jih še ni v bazi. Z izbiro kvadrantov se izberejo priporočene lokacije za določeno skupino inštrumentov. Podvajanje skupin z istim imenom ni dovoljeno.



Slika 14: Primer uporabe dodaj skupino inštrumentov

#### Osnovni tok dogodkov:

Primer uporabe se začne z zahtevo za vnos nove skupine inštrumentov.

1. uporabnik odpre zaslonsko masko skupine inštrumentov
2. izbere dodaj skupino
3. vnese ustrezne podatke in potrdi
4. sistem preveri pravilnost vnesenih podatkov
5. sistem vrne obvestilo o uspešnem vnosu v bazo

#### Alternativni tokovi dogodkov:

- Če je vnosno polje ime skupine prazno, sistem javi napako in ponovno prikaže obrazec
- Če skupina z enakim imenom že obstaja v bazi sistem javi napako

#### Pogoji

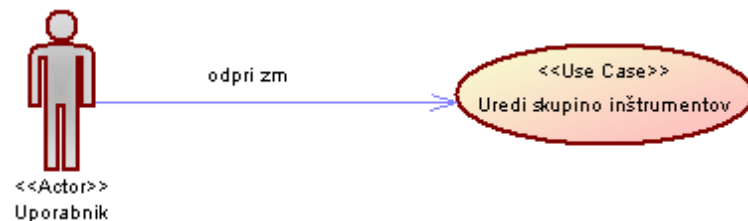
V podatkovni bazi ne sme obstajati skupina inštrumentov z istim imenom kot jo vnašamo.

#### Posledice

V podatkovno bazo je dodan zapis o novi skupini inštrumentov

#### 3.2.2.2.2. Primer uporabe uredi skupino inštrumentov

Primer uporabe omogoča spreminjanje imena skupine in priporočene pozicije skupin.



Slika 15: Primer uporabe uredi skupino inštrumentov

**Osnovni tok dogodkov:**

Primer uporabe se začne z zahtevo za spremembo podatkov o skupini inštrumentov.

1. uporabnik odpre zaslonsko masko skupine inštrumentov
2. izbere uredi skupino
3. izbere skupino inštrumentov, ki jo želi urejati
4. spremeni podatke v ustreznih poljih in potrdi
5. sistem preveri pravilnost vnesenih podatkov
6. sistem prikaže obvestilo o uspešno opravljenih spremembah v bazi

**Alternativni tok dogodkov**

- Če spremenimo ime skupine v takšno, ki že obstaja v podatkovni bazi sistem javi napako in ponovno prikaže obrazec za urejanje skupine inštrumentov

**Pogoji**

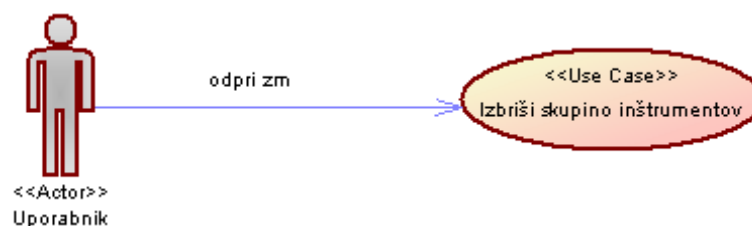
V podatkovni bazi mora obstajati vsaj ena skupina inštrumentov.

**Posledice**

Spremenjeni podatki o skupini inštrumentov so vneseni v podatkovno bazo.

**3.2.2.2.3. Primer uporabe izbriši skupino inštrumentov**

Omogoča brisanje izbrane skupine inštrumentov iz baze. Brisanje skupine inštrumentov iz baze ni mogoče v primeru, da imamo v bazi inštrumente, ki spadajo v skupino, ki jo želimo izbrisati.



Slika 16: Primer uporabe izbriši skupino inštrumentov

**Osnovni tok dogodkov:**

Primer uporabe se začne z zahtevo po brisanju skupine inštrumentov

1. uporabnik odpre zaslonsko masko skupine inštrumentov
2. izbere izbriši skupino
3. izbere skupino inštrumentov, ki jo želi izbrisati
4. potrdi brisanje
5. sistem izbriše izbrano skupino inštrumentov iz baze
6. sistem vrne obvestilo o uspešnem brisanju

**Alternativni tok dogodkov:**

- Če obstajajo v bazi inštrumenti, ki spadajo v skupino, ki jo želimo brisati, sistem javi napako ter skupine ne izbriše

**Pogoji**

V podatkovni bazi ne smejo obstajati inštrumenti, ki se sklicujejo na brisano skupino inštrumentov.

## Posledice

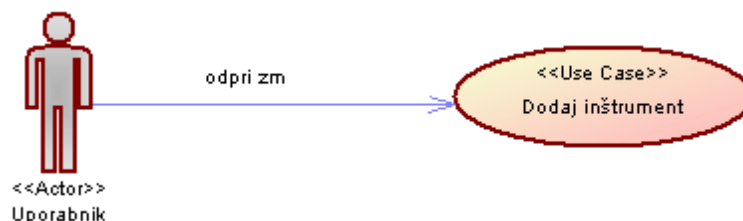
Skupina inštrumentov je izbrisana iz podatkovne baze.

### 3.2.2.3. Urejanje šifrantov inštrumentov

Urejanje šifrantov inštrumentov vsebuje tri primere uporabe. Ti primeri uporabe so dodaj inštrument, uredi inštrument in izbriši inštrument.

#### 3.2.2.3.1. Primer uporabe dodaj inštrument

Primer uporabe omogoča dodajanje novih inštrumentov ob izpolnjenem pogoju, da imamo vneseno vsaj eno skupino inštrumentov.



Slika 17: Primer uporabe dodaj inštrument

#### Osnovni tok dogodkov:

Primer uporabe se začne z zahtevo po vnosu novega inštrumenta.

1. uporabnik odpre zaslonsko masko inštrumenti
2. izbere dodaj inštrument
3. vnese ustrezne podatke v obrazec in potrdi
4. sistem preveri pravilnost vnesenih podatkov
5. sistem prikaže obvestilo o uspešnem vnosu v bazo

#### Alternativni tokovi dogodkov:

- Če je katero izmed vnosnih polj prazno oziroma nepravilno izpolnjeno, sistem vrne obvestilo o napaki in ponovno prikaže obrazec
- Če ime inštrumenta že obstaja v bazi sistem vrne obvestilo o napaki in ponovno prikaže obrazec
- Če nimamo vnesene nobene skupine inštrumentov, sistem ne shrani inštrumenta in ponovno prikaže obrazec

#### Pogoji

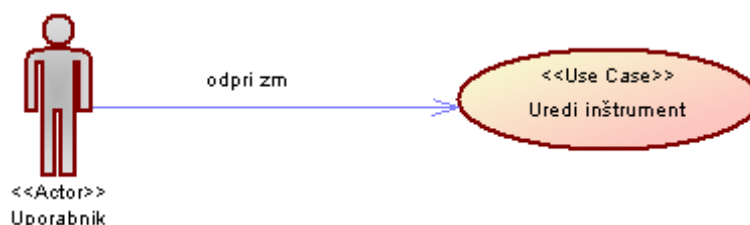
V podatkovni bazi moramo imeti vsaj eno skupino inštrumentov in ne sme obstajati inštrument z enakim imenom.

#### Posledice

V podatkovni bazi je dodan nov zapis o vnesenem inštrumentu.

#### 3.2.2.3.2. Primer uporabe uredi inštrument

Primer uporabe omogoča spreminjanje podatkov o posameznem inštrumentu.



Slika 18: Primer uporabe uredi inštrument

**Osnovni tok dogodkov:**

Primer uporabe se začne z zahtevo po spremembi podatkov o inštrumentu.

1. uporabnik odpre zaslonsko masko inštrumenti
2. izbere uredi inštrument
3. izbere inštrument, ki ga želi urejati
4. spremeni podatke na obrazcu v ustreznih poljih in potrdi
5. sistem preveri pravilnost vnesenih podatkov
6. sistem prikaže obvestilo o uspešno opravljenih spremembah v bazi

**Alternativni tok dogodkov**

- Če je katero izmed vnosnih polj prazno oz. nepravilno izpolnjeno sistem vrne obvestilo o napaki in ponovno prikaže obrazec
- Če je ime inštrumenta drugačno od imena inštrumenta, ki ga spreminjamo in že obstaja v bazi, sistem vrne obvestilo o napaki in ponovno prikaže obrazec

**Pogoji**

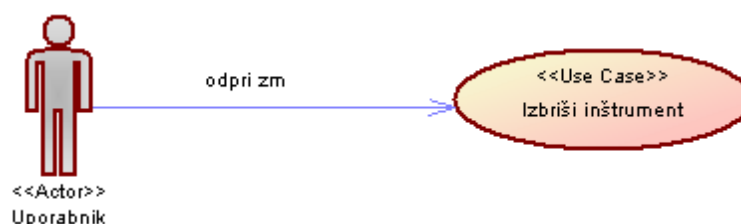
V podatkovni bazi mora obstajati vsaj en inštrument.

**Posledice**

Spremembe o inštrumentu so vnesene v podatkovno bazo.

**3.2.2.3.3. Primer uporabe izbriši inštrument**

Primer uporabe omogoča brisanje inštrumenta iz baze podatkov. Ob brisanju se izbriše inštrument tudi iz vseh plošč, na katere je bil postavljen inštrument.



Slika 19: Primer uporabe izbriši inštrument

**Osnovni tok dogodkov:**

Primer uporabe se začne z zahtevo po brisanju inštrumenta iz podatkovne baze.

1. uporabnik odpre zaslonsko masko inštrumenti
2. izbere izbriši inštrument

3. izbere inštrument, ki ga želi izbrisati
4. potrdi brisanje
5. sistem izbriše izbrani inštrument iz baze
6. sistem pošlje obvestilo o uspešnem brisanju

#### Alternativni tokovi dogodkov:

- Če v bazi nimamo nobenega inštrumenta, sistem ne naredi sprememb v njej
- Če imamo inštrument na kateri izmed predlog, nas sistem o tem obvesti in ponudi možnost brisanja inštrumenta iz vseh predlog ali pa prekinitve brisanja

#### Pogoji

V podatkovni bazi mora obstajati vsaj en zapis o inštrumentu.

#### Posledice

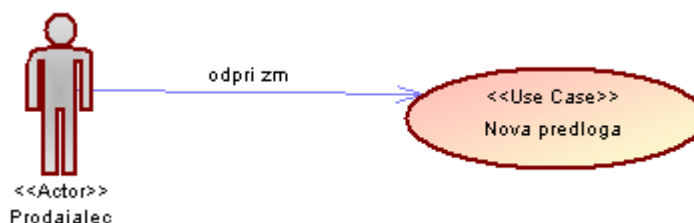
Inštrument je izbrisan iz podatkovne baze. Poleg tega se izbrisani inštrument izbriše tudi iz vseh predlog, na katere je bil postavljen.

#### 3.2.2.4. Urejanje šifrantov predlog

Urejanje šifrantov predlog omogoča brisanje predlog, ustvarjanje novih ter spreminjanje obstoječih. Ko je predloga pripravljena jo lahko izvozimo v PNG datoteko ali pa shranimo predlogo v podatkovno bazo. Prodajalec je akter, ki lahko uporablja omenjene primere uporabe.

##### 3.2.2.4.1. Primer uporabe ustvari novo predlogo

Primer uporabe omogoča, da ustvarimo novo predlogo od začetka ali pa uporabimo obstoječo predlogo in iz nje ustvarimo novo predlogo.



Slika 20: Primer uporabe ustvari novo predlogo

#### Osnovni tok dogodkov:

Primer uporabe se začne z zahtevo po novi predlogi.

1. prodajalec odpre zaslonsko masko predloge
2. izbere ploščo in prazno predlogo
3. sistem prikaže prazno predlogo iz izbrane plošče
4. prodajalec izbere ime predloge
5. postavi inštrumente ustrezno na ploščo
6. shrani predlogo v podatkovno bazo
7. sistem preveri pravilnost postavljenih inštrumentov
8. prikaže se obvestilo o uspešno shranjeni predlogi

**Alternativni tokovi dogodkov:**

- Prodajalec ustvari novo predlogo iz obstoječe predloge
- Če inštrumenti niso pravilno postavljeni sistem javi napako. Pod pravilno postavljene inštrumente se smatra, da se inštrumenti med seboj ne prekrivajo

**Pogoji**

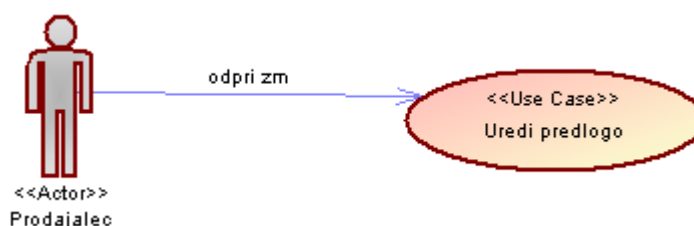
V podatkovni bazi mora biti shranjena vsaj ena plošča.

**Posledice**

Ustvarjen je zapis o novi predlogi v podatkovni bazi.

**3.2.2.4.2. Primer uporabe uredi predlogo**

Primer uporabe omogoča, da spremenimo postavitev inštrumentov, na že obstoječi predlogi.



Slika 21: Primer uporabi uredi predlogo

**Osnovni tok dogodkov:**

Primer uporabe se začne z zahtevo po urejanju predloge.

1. prodajalec odpre zaslonsko masko predloge
2. izbere ploščo
3. sistem prikaže vse predloge iz izbrane plošče
4. prodajalec izbere predlogo, ki jo želi urejati
5. sistem prikaže predlogo
6. prodajalec uredi predlogo in shrani spremembe
7. sistem preveri pravilnost postavljenih inštrumentov
8. prikaže se obvestilo o uspešno spremenjeni predlogi

**Alternativni tokovi dogodkov:**

- Če se inštrumenti na plošči med sabo prekrivajo sistem javi napako.

**Pogoji**

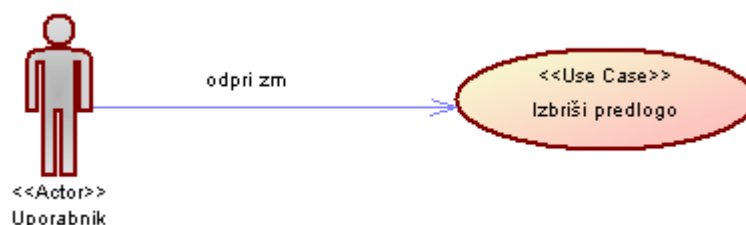
V podatkovni bazi mora obstajati vsaj ena predloga.

**Posledice**

Spremembe predloge so shranjene v podatkovno bazo.

**3.2.2.4.3. Primer uporabe izbriši predlogo**

Primer uporabe omogoča brisanje predloge iz podatkovne baze.



Slika 22: Primer uporabe izbriši predlogo

**Osnovni tok dogodkov:**

Primer uporabe se začne z zahtevo po brisanju predloge.

1. prodajalec odpre zaslonsko masko predloge
2. prodajalec izbere ploščo
3. sistem prikaže vse ustvarjene predloge iz izbrane plošče
4. prodajalec izbere predlogo in potrdi brisanje
5. sistem zahteva potrditev o brisanju predloge
6. prodajalec potrdi okence za brisanje
7. sistem vrne sporočilo o uspešno izbrisani predlogi

**Alternativni tokovi dogodkov:**

- Če se prodajalec pri okencu za potrditev brisanja odloči, da ne bo izbrisal predloge, je sistem ne izbriše.
- Če v bazi nimamo nobene predloge, sistem ne naredi sprememb.

**Pogoji**

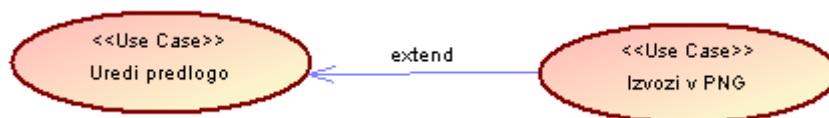
V podatkovni bazi mora obstajati vsaj ena predloga.

**Posledice**

Predloga je izbrisana iz podatkovne baze.

**3.2.2.4.4. Primer uporabe izvozi v PNG**

Primer uporabe razširja primer uporabe urejanja šifrantov predlog. Omogoča nam, da ustvarjeno predlogo izvozimo v PNG datoteko, ki jo lahko kasneje uporabimo za tiskanje na papir, pošiljanje preko e-pošte ali kakšne druge namene.



Slika 23: Primer uporabe izvozi v PNG

**Osnovni tok dogodkov:**

Primer uporabe se začne z zahtevo po izvozu predloge v bitno sliko. Prodajalec lahko odpre predlogo po postopku za ustvarjanje nove ali urejanje obstoječe predloge.

1. prodajalec pripravi predlogo za izvoz
2. pritisne gumb izvozi v PNG
3. sistem preveri, če se inštrumenti prekrivajo
4. prodajalec izbere pot in ime, kamor bo shranil sliko
5. sistem ustvari datoteko

**Alternativni tokovi dogodkov:**

- Če se inštrumenti prekrivajo med seboj, nam sistem vrne obvestilo o prekrivanju.

**Pogoji**

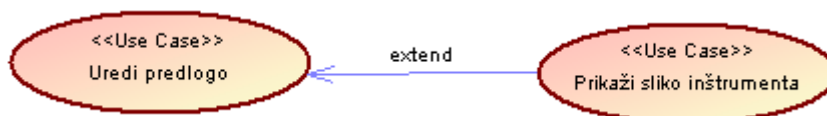
Inštrumenti se ne smejo prekrivati med seboj.

**Posledice**

Ustvarjena je datoteka s končnico PNG.

**3.2.2.4.5. Primer uporabe prikaži sliko inštrumenta**

Primer uporabe razširja primer uporabe urejanje šifrantov predlog. Omogoča nam, da pogledamo bitno sliko izbranega inštrumenta.



Slika 24: Primer uporabe prikaži sliko inštrumenta

**Osnovni tok dogodkov:**

Primer uporabe se prične z zahtevo po prikazu slike inštrumenta. Predhodno mora prodajalec imeti odprto predlogo, ki jo lahko odpre po postopku za ustvarjanje nove ali urejanje obstoječe predloge.

1. prodajalec odprto predlogo
2. izbere inštrument in pritisne gumb prikaži inštrument
3. sistem preveri pot do bitne slike
4. sistem prikaže sliko inštrumenta

**Alternativni tokovi dogodkov:**

- Če pot do bitne slike ni pravilna, se slika ne prikaže.

**Pogoji**

Inštrument mora biti izbran in pot do bitne slike mora biti pravilna.

**Posledice**

Odpre se novo okno z bitno sliko izbranega inštrumenta.

### 3.3. Načrtovanje podatkovne baze

Podatkovna baza je strukturirana zbirka podatkov, ki jo upravlja sistem za upravljanje podatkovnih baz in po svoji vsebini predstavlja model oziroma podatkovno sliko določenega dela realnega sveta[3]. Sestavljena je iz fizične in logične plasti. Fizično plast v veliki meri vnaprej določa sistem za upravljanje podatkovnih baz, tako da je potrebno upravljelju podatkovne baze definirati le manjši del notranje sheme. S tem je poglavitni del načrtovanja podatkovne baze vezan le na načrtovanje logične sheme.

Do logične sheme podatkovne baze lahko pridemo na dva načina. Eden način je z direktnim načrtovanjem relacijskih shem, pri katerem načrtovalec najprej definira izhodiščne relacijske sheme, ki jih nato s postopkom normalizacije pretvori v končno množico relacijskih shem, imenovano logična shema podatkovne baze. Drugi način pa je preko vmesnega – konceptualnega modela. Konceptualni model temelji običajno na enem izmed semantičnih podatkovnih modelov, ki obsega več modelirnih konceptov in je zaradi tega preprostejši. Prednost teh modelov je, da obsegajo tudi grafične predstavitve, kar pomeni, da so preprostejši za razumevanje.

Logično shemo podatkovne baze sem izdelal preko načrta konceptualnega podatkovnega modela, ki sem ga nato preslikal v relacijski model. Konceptualni podatkovni model je včasih poimenovan kot entitetni model, podrobni podatkovni model ali pa kar podatkovni model [4]. Glavni koncepti konceptualnega podatkovnega modela so entiteta, atribut, povezave med entitetami in entitetni identifikator.

Entitete so posamezne stvari, ki obstajajo v svetu, ki ga modeliramo. Atributi predstavljajo lastnosti entitet, tako da identificirajo, poimenujejo ali opisujejo primerke entitet. Vsak atribut pripada določenemu podatkovnemu tipu. Najbolj pogosti podatkovni tipi so znakovni in numerični, katerim moramo določiti tudi dolžino. Entitetni identifikator je lahko sestavljen iz enega ali več atributov in enolično ter nedvoumno identificira vsak primerke entitete. Konceptualni podatkovni model dobimo z združevanjem konceptualnih podatkovnih podmodelov. Ker gre v našem primeru za manjši sistem, imamo le en podmodel, ki je hkrati tudi konceptualni podatkovni model.

Postopek izdelave modela je sledeč:

- določitev entitet in njihovih atributov
- določitev povezav med entitetami
- določitev enoličnih identifikatorjev entitet
- opisovanje gradnikov entitetnih diagramov: entitet, njihovih atributov in povezav med entitetami

### 3.3.1. Konceptualni podatkovni model

Iz postopka izdelave dobimo za naš primer pet entitet. Te so INSTRUMENT, PLATE, TYPE, SHAPE in POS\_ON\_PLATE.

#### Opisi entitet, njihovih atributov in povezav med entitetami:

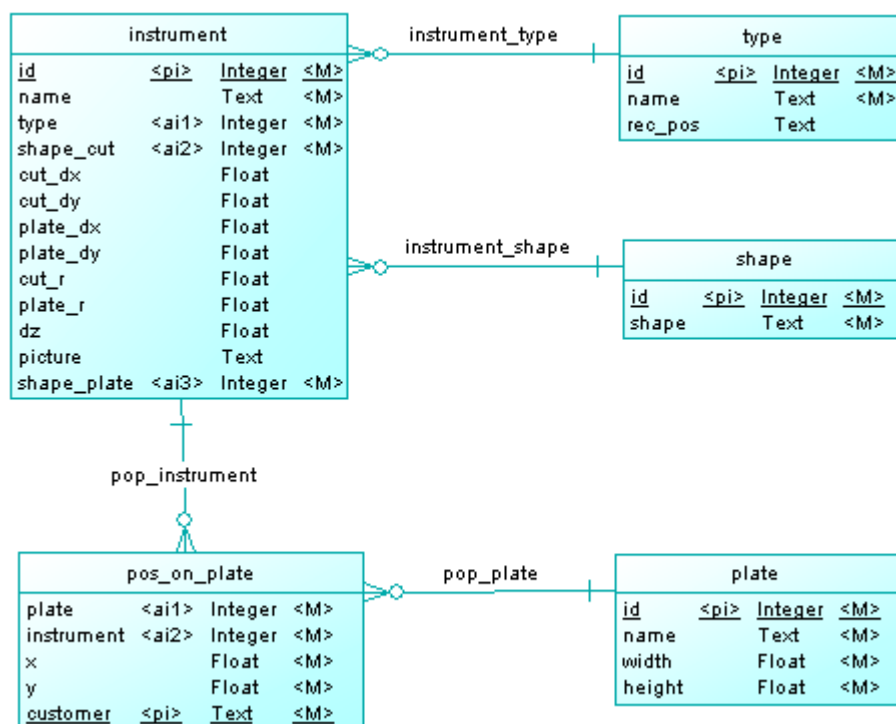
Entiteta: INSTRUMENT  
 Opis: vsebuje podatke o inštrumentih.  
 Atributi: ID – enolični identifikator  
 NAME – ime inštrumenta  
 TYPE – skupina inštrumenta v katero spada  
 SHAPE\_CUT – predstavlja obliko, ki jo moramo izrezati na plošči, da lahko vstavimo inštrument  
 SHAPE\_PLATE – predstavlja obliko, ki je vidna na plošči, ko je inštrument vstavljen v ploščo.  
 CUT\_DX – potrebna širina izreza v ploščo, da lahko inštrument vstavimo  
 CUT\_DY – potrebna višina izreza v ploščo, da lahko inštrument vstavimo  
 PLATE\_DX – širina inštrumenta na površini plošče  
 PLATE\_DY – višina inštrumenta na površini plošče  
 CUT\_R – polmer izreza v ploščo, da lahko inštrument vstavimo  
 PLATE\_R – polmer inštrumenta na površini plošče  
 DZ – globina inštrumenta  
 PICTURE – pot do bitne slike inštrumenta

Entiteta: PLATE  
 Opis: vsebuje podatke o ploščah  
 Atributi: ID – enolični identifikator  
 NAME – ime plošče  
 WIDTH – širina plošče  
 HEIGHT – višina plošče  
 V\_PICTURE – pot do vektorske slike plošče

Entiteta: SHAPE  
 Opis: podatki o možnih oblikah inštrumentov  
 Atributi: ID – enolični identifikator  
 SHAPE – ime oblike

Entiteta: TYPE  
 Opis: podatki o skupinah inštrumentov  
 Atributi: ID – enolični identifikator  
 NAME – ime skupine inštrumentov  
 REC\_POS – priporočljiva lokacija skupine inštrumentov, shranjena v obliki števil, ki predstavljajo kvadrante

Entiteta: POS\_ON\_PLATE  
 Opis: podatki o predlogah, ki hranijo koordinate inštrumentov na plošči  
 Atributi: PLATE – enolični identifikator plošče  
 INSTRUMENT – enolični identifikator inštrumenta  
 X – koordinata x od inštrumenta  
 Y – koordinata y od inštrumenta  
 CUSTOMER – ime predloge



Slika 25: Konceptualni podatkovni model

### Povezave med entitetami:

*instrument\_type* – povezava tipa 1:N med entitetama instrument in type. Vsakemu primerku entitete instrument pripada natanko eden primerek entitete type. Vsakemu primerku entitete type pripada nič ali več primerkov entitete instrument.

*instrument\_shape* - povezava tipa 1:N med entitetama instrument in shape. Vsakemu primerku entitete instrument pripada natanko eden primerek entitete shape. Vsakemu primerku entitete shape pripada nič ali več primerkov entitete instrument.

*pop\_instrument* – povezava tipa 1:N med entitetama pos\_on\_plate in instrument. Vsakemu primerku entitete instrument pripada nič ali več primerkov entitete pos\_on\_plate. Vsakemu primerku entitete pos\_on\_plate pripada natanko eden primerek entitete instrument.

*pop\_plate* – povezava tipa 1:N med entitetama pos\_on\_plate in plate. Vsakemu primerku entitete plate pripada nič ali več primerkov entitete pos\_on\_plate. Vsakemu primerku entitete pos\_on\_plate pripada natanko eden primerek entitete plate.

### 3.3.2. Logični podatkovni model

Logični podatkovni model opisuje logično strukturo baze podatkov in je izdelan na podlagi postopka, ki ustreza relacijskemu modelu. Podlaga za izdelavo relacijskega modela je konceptualni podatkovni model. Pretvorba poteka tako, da se najprej vsaka entiteta konceptualnega podatkovnega modela pretvori v relacijo v relacijskem modelu. Pri tem se enolični identifikator pretvori v primarni ključ relacije. V naslednjem koraku je potrebno pretvoriti vse povezave konceptualnega podatkovnega modela v relacijskem modelu. Povezava je lahko predstavljena, tako da primarni ključ ene relacije postane zunanji ključ v drugi relaciji ali pa je za predstavitev povezave kreirana posebna relacija. V zadnjem koraku je potrebno relacije, ki so nastale predhodno normalizirati, da preprečimo anomalije.

#### 3.3.2.1. Relacija plate

Tabela služi hranjenju podatkov o ploščah. Podatke v njej lahko spreminjamo preko zaslonske maske urejanje šifrantov plošč.

Ime atributa	Podatkovni tip
<u>ID</u>	Number(autoincrement)
NAME	Varchar(50)
WIDTH	Float
HEIGHT	Float
V_PICTURE	Varchar(200)

Tabela 1: Relacija plate

#### Atributi in njihov opis:

- ID – je primarni ključ tabele in je samoštevilo, ki se samodejno poveča ob novem zapisu.
- NAME – ime plošče, ki je lahko dolgo do 50 znakov. Podvojena imena niso dovoljena.
- WIDTH – predstavlja širino plošče v milimetrih in je racionalno število. Podatkovni tip je racionalno število.
- HEIGHT – predstavlja višino plošče v milimetrih in je racionalno število. Podatkovni tip je racionalno število.
- V\_PICTURE – predstavlja pot do vektorske slike plošče in je lahko dolgo največ 200 znakov.

#### 3.3.2.2. Relacija type

Tabela hrani podatke o skupini inštrumentov. Spreminja se jo lahko preko zaslonske maske urejanje šifrantov skupin inštrumentov.

Ime atributa	Podatkovni tip
<u>ID</u>	Number(autoincrement)
NAME	Varchar(50)
REC_POS	Varchar(50)

Tabela 2: Relacija type

**Atributi in njihov opis:**

- ID – je primarni ključ tabele in je samoštevilo, ki se samodejno poveča ob novem zapisu.
- NAME – predstavlja ime skupine inštrumentov, ki je lahko dolgo največ 50 znakov. Podvojene skupine niso dovoljene.
- REC\_POS – predstavlja priporočeno lokacijo za določeno skupino inštrumentov. V polju so shranjene številke kvadrantov (npr.: 1,4,7). Izbiramo lahko med devetimi kvadranti.

**3.3.2.3. Relacija shape**

Tabela hrani podatke o možnih oblikah inštrumentov. Tabela je fiksno določena in je ni možno urejati preko aplikacije. Za spremembo je potreben poseg v podatkovno zbirko, preko programa Microsoft Access. Vendar to ni priporočljivo, ker s tem pokvarimo delovanje aplikacije. Tabela je narejena tako, ker je večina inštrumentov dveh oblik, in sicer pravokotne in okrogle. Izdelal sem jo, ker v primeru pojava novih oblik v prihodnosti ni potrebno spreminjati strukture podatkovne baze, ampak je potrebno spremeniti le program.

Ime atributa	Podatkovni tip
ID	Number(autoincrement)
NAME	Varchar(50)

Tabela 3: Relacija shape

**Atributi in njihov opis:**

- ID – je primarni ključ tabele in je samoštevilo, ki se samodejno poveča ob novem zapisu.
- NAME – predstavlja ime oblike, ki je lahko dolgo največ 50 znakov. Podvojena imena niso dovoljena.

**3.3.2.4. Relacija instrument**

Tabela instrument hrani podatke o inštrumentih v podatkovni bazi. Spreminja se jih lahko preko zaslonske maske urejanje šifrantov inštrumentov.

Ime atributa	Podatkovni tip
ID	Number(autoincrement)
NAME	Varchar(50)
TYPE	Integer
SHAPE_CUT	Integer
SHAPE_PLATE	Integer
CUT_DX	Float
CUT_DY	Float
PLATE_DX	Float
PLATE_DY	Float
CUT_R	Float
PLATE_R	Float
DZ	Float
PICTURE	Varchar(200)

Tabela 4: Relacija instrument

**Atributi in njihov opis:**

- ID – je primarni ključ tabele in je samoštevilo, ki se samodejno poveča ob novem zapisu.
- NAME – ime inštrumenta, ki je lahko dolgo 50 znakov. Podvojena imena niso dovoljena.
- TYPE – je zunanji ključ relacije type in predstavlja skupino inštrumenta v katero spada.
- SHAPE\_CUT – je zunanji ključ relacije shape in predstavlja obliko izreza inštrumenta.
- SHAPE\_PLATE – je zunanji ključ relacije shape in predstavlja obliko inštrumenta na plošči.
- CUT\_DX – predstavlja širino izreza na plošči v milimetrih v primeru, da je oblika izreza pravokotnik. Podatkovni tip je racionalno število.
- CUT\_DY – predstavlja višino izreza na plošči v milimetrih v primeru, da je oblika izreza pravokotnik. Podatkovni tip je racionalno število.
- PLATE\_DX – predstavlja širino inštrumenta na plošči v milimetrih v primeru, da je oblika na plošči pravokotnik. Podatkovni tip je racionalno število.
- PLATE\_DY – predstavlja višino inštrumenta na plošči v milimetrih v primeru, da je oblika na plošči pravokotnik. Podatkovni tip je racionalno število.
- CUT\_R – predstavlja polmer izreza na plošči v milimetrih v primeru, da je oblika izreza krog. Podatkovni tip je racionalno število.
- PLATE\_R – predstavlja polmer inštrumenta na plošči v milimetrih v primeru, da je oblika izreza krog. Podatkovni tip je racionalno število.
- DZ – predstavlja globino inštrumenta v milimetrih. Podatkovni tip je racionalno število.
- PICTURE – predstavlja pot do bitne slike inštrumenta. Največja dovoljena dolžina je 200 znakov.

Kadar je izbrana oblika izreza (SHAPE\_CUT) pravokotnik, se podatki iz obrazca shranijo v CUT\_DX in CUT\_DY, kadar pa krog, se podatki shranijo v CUT\_R. Enako velja za obliko na plošči (SHAPE\_PLATE).

**3.3.2.5. Relacija pos\_on\_plate**

Tabela hrani podatke o predlogah. Spreminja se jih lahko preko zaslonske maske urejanje šifrantov predlog.

Ime atributa	Podatkovni tip
<u>PLATE</u>	Integer
<u>INSTRUMENT</u>	Integer
X	Float
Y	Float
<u>CUSTOMER</u>	Varchar(50)

Tabela 5: Rlacija pos\_on\_plate

### Atributi in njihov opis:

- PLATE – je zunanji ključ relacije plate in predstavlja ploščo, ki je uporabljena pri predlogi.
- INSTRUMENT – je zunanji ključ relacije instrument in predstavlja inštrument, ki je na določeni predlogi.
- X – predstavlja koordinato x, za določen inštrument, na določeni plošči in predlogi.
- Y – predstavlja koordinato y, za določen inštrument, na določeni plošči in predlogi.
- CUSTOMER – je primarni ključ in predstavlja ime predloge, ki je lahko dolgo največ 50 znakov. Podvojena imena predlog niso dovoljena.

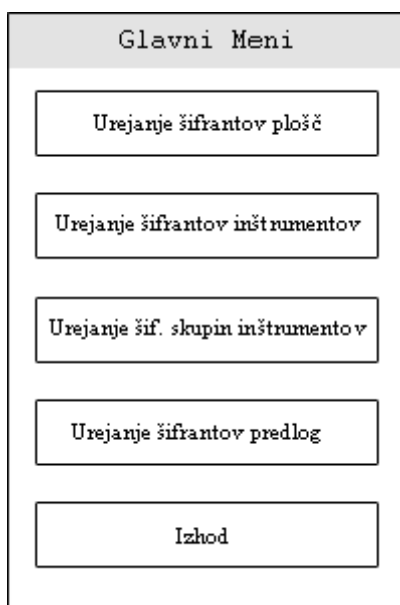
## 3.4. Načrtovanje uporabniškega vmesnika

Načrtovanje uporabniškega vmesnika in izpisa je pomembna aktivnost, saj uporabniki pogosto ocenjujejo kvaliteto aplikacije po izgledu in jo zaradi tega tudi raje in hitreje sprejmejo. Zaradi tega je potrebno uporabniški vmesnik čimbolj oblikovati. Mora pa biti tudi funkcionalen, da ga uporabniki z veseljem uporabljajo.

Najprej je potrebno ugotoviti, koliko zaslonkih mask potrebujemo. Ena zaslonska maska služi kot glavni meni, iz katerega je mogoče dostopati do ostalih menijev. Ostale zaslonske maske, ki jih potrebujemo so urejanje šifrantov plošč, urejanje šifrantov inštrumentov, urejanje šifrantov skupin inštrumentov ter urejanje šifrantov predlog.

### 3.4.1. Dizajn glavnega menija

Iz glavnega menija moramo imeti dostop do vseh pomembnejših zaslonkih mask. Iz njega lahko dostopamo do šifrantov plošč, inštrumentov, skupin inštrumentov in predlog. Omogočati mora tudi izhod iz aplikacije. Iz glavnega menija je mogoče odpreti največ eno zaslonsko masko hkrati. Ko zaključimo z delom odprte zaslonske maske, lahko odpremo naslednjo.



Slika 26: Skica glavnega menija

### 3.4.2. Dizajn urejanja šifrantov plošč

Ko odpremo zaslonsko masko urejanje šifrantov plošč, najprej izberemo dodaj, uredi ali izbriši ploščo. Privzeto je nastavljeno, da se ob prikazu zaslonske maske odpre obrazec za dodajanje plošč. Naknadno lahko spremenimo v uredi ali izbriši ploščo. Na obrazcu za dodajanje imamo polja kot so ime ploščice, širina ploščice, višina ploščice in vektorska slika. Vsa ta polja so tekstovna polja, v katera vpisujemo ustrezne podatke.

Pri urejanju ploščice se pojavi dodatno polje izberi ploščo, s katerim izberemo ploščo, ki jo bomo urejali.

Obrazec izbriši ploščo ima vsa polja enaka kot uredi ploščo, le da ima vsa polja onemogočena za vnos, razen polja izberi ploščo, ki ima enako vlogo kot pri urejanju. Z gumbom potrdi shranimo spremembe v bazo, gumb zapri pa nas vrne na zaslonsko masko glavnega menija. Spodnja slika prikazuje skico obrazca za urejanje plošč.



Urejanje šifrantov plošč

Dodaj     Uredi     Izbriši

Izberi ploščo:

Ime ploščice:

Širina ploščice:

Višina ploščice:

Vektorska slika:

Potrdi    Zapri

Slika 27: Skica urejanja šifrantov plošč

### 3.4.3. Dizajn urejanja šifrantov skupin inštrumentov

Ko odpremo urejanje šifrantov skupin inštrumentov najprej izberemo ali bomo dodajali, urejali ali brisali skupine inštrumentov. Privzeto je, da se odpre obrazec za dodajanje skupine, lahko pa sami spremenimo na uredi oziroma izbriši skupino inštrumentov.

Pri dodajanju skupine inštrumentov imamo na obrazcu polje, ki se imenuje ime skupine in omogoča vnos znakov. Poleg tega se nam prikaže devet kvadrantov, ki določajo katera lokacija na plošči je priporočena za določeno skupino inštrumentov. Kvadrante lahko označimo ali odznačimo.

Pri urejanju se nam pojavi dodatno polje, izberi skupino, ki omogoča, da izberemo iz seznama skupino, ki jo želimo spreminjati. Prav tako imamo na voljo devet kvadrantov, ki jih lahko označimo ali odznačimo.

Obrazec za brisanje skupin ima onemogočena vsa polja, razen polja za izbiro skupine.

Z gumbom potrdi, se spremembe vnesejo v podatkovno bazo. Gumb zapri pa nas vrne na glavni meni. Spodnja slika prikazuje skico obrazca za urejanje skupin inštrumentov.

Urejanje šifrantov skupin inštrumentov

Dodaj   
  Uredi   
  Izbriši

Izberi skupino:

Ime skupine:

<input checked="" type="checkbox"/> 1.	<input checked="" type="checkbox"/> 2.	<input checked="" type="checkbox"/> 3.
<input type="checkbox"/> 4.	<input type="checkbox"/> 5.	<input type="checkbox"/> 6.
<input type="checkbox"/> 7.	<input type="checkbox"/> 8.	<input type="checkbox"/> 9.

Slika 28: Skica urejanja šifrantov skupin inštrumentov

#### 3.4.4. Dizajn urejanja inštrumentov

Ko odpremo zaslonsko masko urejanje inštrumentov, se prikaže obrazec za dodajanje inštrumentov. Po želji lahko z ustrezno izbiro urejamo ali brišemo inštrumente iz baze. Na obrazcu za dodajanje inštrumentov imamo polja, kot so ime inštrumenta, izrez dx, izrez dy, plošča dy, plošča dy, globina inštrumenta in slika inštrumenta, ki so tekstovna polja za vnos. Poleg imamo še izbirna polja kot so skupina inštrumenta, oblika izreza in oblika na plošči pri katerih izbiramo vrednosti iz seznama.

Pri urejanju inštrumentov se nam pojavi dodatno polje izberi inštrument, s katerim izberemo iz seznama inštrument, ki ga želimo spremeniti.

Pri brisanju inštrumentov je enak obrazec, kot pri urejanju, le da so vsa polja, razen polja izberi inštrument, onemogočena za spreminjanje.

Z gumbom potrdi izvršimo spremembe, gumb zapri pa zapre zaslonsko masko in nas vrne v glavni meni. Spodnja skica prikazuje obrazec za urejanje inštrumentov.

Urejanje šifrantov inštrumentov

o Dodaj   ● Uredi   o Izbriši

Izberi inštrument:

Ime inštrumenta:

Skupina inštrumenta:

Oblika izreza:

Izrez dx:

Izrez dy:

Oblika na plošči:

Plošča dx:

Plošča dy:

Globina inštrumenta:

Slika inštrumenta:

Potrdi   Zapri

Slika 29: Skica urejanja šifrantov inštrumentov

### 3.4.5. Dizajn urejanja predlog

Urejanje šifrantov predlog poteka v dveh delih. Ko odpremo zaslonsko masko, se odpre obrazec, kjer izberemo ploščo in predlogo. Polja izberi ploščo in izberi predlogo sta seznama iz katerih se izbere želene vrednosti. Ko sta polji izbrani, lahko izbrišemo predlogo ali pa nadaljujemo z gumbom pokaži, kjer lahko postavljamo inštrumente na ploščo.

Urejanje šifrantov predlog

Izberi ploščo:

Izberi predlogo:

Zbriši predlogo   Pokaži predlogo

Zapri

Slika 30: Skica izbiranja predloge

Ko zaključimo z izbiranjem plošče in predloge in pritisnemo na gumb pokaži predlogo se nam odpre nov obrazec, ki omogoča postavljanje inštrumentov na ploščo. Gumbi s številkami od 1 do 7 služijo raznovrstnim poravnavam inštrumentov. Na desni strani je risalna površina, kjer se izriše izbrana plošča in inštrumenti postavljeni na ploščo. Tekstovni polji, izbrana plošča in izbrana predloga sta onemogočeni za vnos in prikazujeta imena izbranih vrednosti iz predhodne zaslonske maske. Gumb menjaj predlogo omogoča zamenjavo predloge. Gumba

dodaj na ploščo in odstrani s plošče služita za dodajanje in odstranjevanje inštrumentov iz plošče. Polje shrani predlogo je tekstovno polje, v katerega vnesemo ime predloge. Pod njim imamo še seznam inštrumentov v bazi, ki jih je možno dodajati na ploščo. Polje inštrumenti na plošči je več vrstično polje s seznamom inštrumentov, ki se trenutno nahajajo na plošči. Iz tega seznama lahko izbiramo posamezne inštrumente. Z gumbom shrani, shranimo spremembe v podatkovno bazo, z izvozi v PNG pa ustvarimo novo bitno datoteko, ki prikazuje postavitev inštrumentov na plošči.

Urejanje šifrantov predlog

Izbrana plošča:  1 2 3 4 5 6 7

Izbrana predloga:

Shrani predlogo:

Izberi inštrument

Inštrumenti na plošči:

Prezrcali čez y os

Koordinate: X

Y

risalna površina

Slika 31: Skica urejanja predloge

## 4. Aplikacija

### 4.1. Splošno o aplikaciji

Za razvoj aplikacije sem uporabil programsko okolje Visual studio 2008. Za shranjevanje podatkov sem uporabil podatkovno bazo Microsoft Access. Za kodiranje programa pa sem uporabil programski jezik C#.

#### 4.1.1. Inštalacija aplikacije

Aplikacijo sestavljata dve datoteki. Prva je *Pipistrel.exe*, ki je izvršljiva datoteka, ki omogoča zagon aplikacije in jo lahko po želji tudi preimenujemo. Druga je podatkovna baza z imenom *baza.mdb*, ki omogoča shranjevanje podatkov in je ni dovoljeno preimenovali. Obe datoteki prekopiramo v mapo, ki je skupna obema datotekama in si po želji ustvarimo bližnjico do izvršljive datoteke. S tem je aplikacija inštalirana. Lahko jo začnemo uporabljati.

Predpogoj za delovanje aplikacije je nameščen Microsoft Windows (XP, 2000, 2003, Vista) operacijski sistem z nameščenim programom .NET 3.0 različico ali višjo. Minimalne strojne zahteve so odvisne od nameščenega operacijskega sistema in so takšne, kot jih zahteva nameščen operacijski sistem za zagon.

#### 4.1.2. Prenosljivost aplikacije

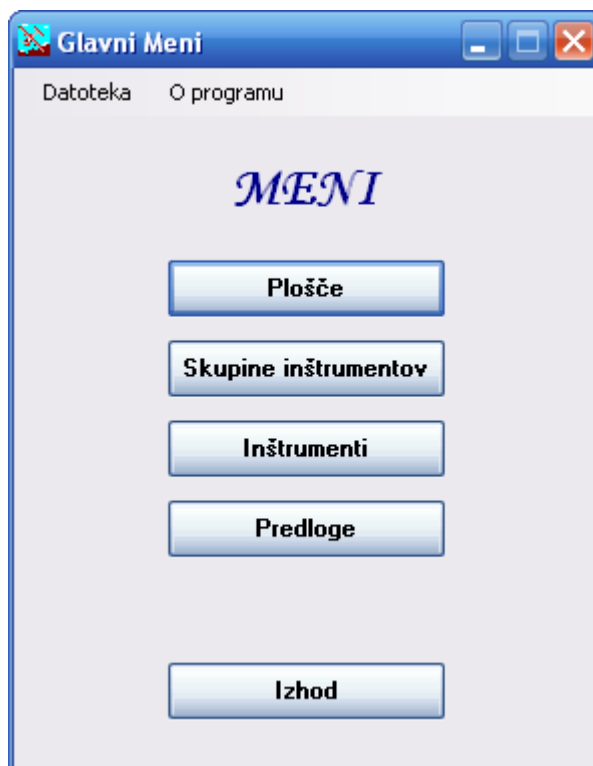
Pogoj za popolno prenosljivost aplikacije je, da imamo vse slike na katere se sklicujemo v aplikaciji, shranjene v eni ali več podmapah aplikacije. Praviloma lahko podamo pri sklicu na sliko tudi sliko, ki se ne nahaja v podmapi aplikacije, vendar v tem primeru ob prenosu aplikacije na drug računalnik ni nujno, da bodo slike pravilno prikazane, oziroma, da bo aplikacija pravilno delovala.

Če upoštevamo zgornji pogoj za popolno prenosljivost aplikacije, jo lahko prenesemo na drug računalnik tako, da prenesemo celotno mapo, kjer se nahaja aplikacija v drug računalnik ali pa v drugo mapo na istem računalniku.

## 4.2. Opis posameznih zaslonkih mask

### 4.2.1. Zaslonska maska glavnega menija

Aplikacija ob zagonu prikaže glavni meni, ki omogoča izbor zaslonkih mask za urejanje plošč, skupin inštrumentov, inštrumentov in predlog. Prve tri zaslonske maske so si po funkcionalnostih precej podobne, saj vse omogočajo dodajanje, spreminjanje in brisanje iz podatkovne baze. Zaslonska maska predloge je nekoliko bolj kompleksna in omogoča postavljanje inštrumentov na plošče. Na vrhu aplikacije imamo meni, ki omogoča identično izbiranje, kot če bi uporabili gube na zaslonski maski. Poleg imamo še informacije o programu.



Slika 32: Zaslonska maska glavnega menija

#### 4.2.2. Zaslonska maska plošče

Ob prikazu zaslonske maske imamo na izbiro tri funkcionalnosti. Dodaj ploščo, uredi ploščo in izbrisi ploščo. Privzeto je, da se odpre obrazec dodaj ploščo, naknadno pa lahko z ustrezno izbiro izbirnega gumba na zaslonski maski prikažemo obrazca za urejanje in brisanje plošč. To zaslonsko masko uporabljajo bodisi vodja proizvodnje bodisi prodajalec.

Elementi, ki sestavljajo zaslonsko masko plošče:

##### **Polje s seznamom:**

*Izberi ploščo* – polje za izbiro plošče iz seznama

##### **Vnosna polja:**

*Ime plošče* – tekstovno polje, ki vsebuje ime plošče

*Širina plošče* – tekstovno polje, ki sprejema numerične vrednosti širine plošče v milimetrih

*Višina plošče* – tekstovno polje, ki sprejema numerične vrednosti višine plošče v milimetrih

*Vektorska slika* – tekstovno polje, ki vsebuje pot do vektorske slike

##### **Gumbi:**

*Prebrskaj* – omogoča enostavno vnašanje poti vektorske slike

*Potrdi* – omogoča ažuriranje podatkov v podatkovni bazi

*Zapri* – zapre zaslonsko masko in nas vrne v glavni meni

Obrazec za dodajanje inštrumentov vsebuje vsa polja, razen polja s seznamom izberi ploščo. Vsa polja na tem obrazcu morajo biti izpolnjena pravilno za ažuriranje podatkov v podatkovni

bazi. V primeru, da je katero polje prazno oziroma je nepravilno izpolnjeno (npr. širina podana z črkami), se prikaže obvestilo o napaki. Pri tem se podatki ne shranijo v podatkovno bazo.

Slika 33: Zaslonska maska z obrazcem za dodajanje plošče

Za urejanje plošč moramo izbrati uredi ploščo na zaslonski maski. Pojavi se obrazec za urejanje plošče, ki ima identična vnosna polja in gube kot obrazec za dodajanje inštrumentov. Za vse elemente veljajo enaka pravila, kot pri dodajanju plošč. Dodano imamo izbirno polje izberi ploščo. Polje omogoča izbiro plošče iz seznama.

Brisanje plošče je po prikazu identično urejanju plošče, s to razliko, da so vsa vnosna polja, razen izberi ploščo onemogočena za pisanje.

#### 4.2.3. Zaslonska maska skupine inštrumentov

Zaslonska maska omogoča podobno, kot zaslonska maska plošče. Dodajanje, urejanje in brisanje skupin inštrumentov. Privzeto se odpre obrazec za dodajanje, lahko pa z izbirnim gumbom prikažemo obrazec za urejanje ali brisanje skupin. Vsaka skupina inštrumentov ima določeno priporočeno lokacijo na plošči, ki jo določimo z izbiro kvadrantov. Zaslonsko masko lahko uporablja vodja proizvodnje ali prodajalec.

Elementi, ki sestavljajo zaslonsko masko skupine inštrumentov:

**Polje s seznamom:**

*Izberi skupino* – polje za izbiro skupine inštrumentov iz seznama

**Vnosno polje:**

*Ime skupine* – tekstovno polje, ki vsebuje ime skupine

**Potrditvena polja:**

*1. – 9. kvadrant* – polja, ki omogočajo izbiro posameznega območja z da/ne

**Gumbi:**

*Potrdi* – omogoča ažuriranje sprememb v podatkovni bazi

*Zapri* – zapre zaslonsko masko in nas vrne na glavni meni

Obrazec za dodajanje skupine inštrumentov je sestavljen iz vseh zgoraj navedenih elementov, razen polja s seznamom. Obvezen je vnos imena skupine, izbira kvadrantov pa je poljubna. V primeru napačnega vnosa imena skupine se podatki ne shranijo v bazo. Prikaže se obvestilo o napaki.

Slika 34: Zaslonska maska z obrazcem za urejanje skupine inštrumenta

Izbirni gumb uredi skupino, nam prikaže obrazec za urejanje skupine inštrumentov. Obrazec je sestavljen iz vseh elementov, ki jih vsebuje obrazec za dodajanje s to razliko, da ima dodano polje s seznamom izberi skupino, s katerim izberemo skupino, ki jo želimo urejati.

Obrazec za brisanje je po obliki identičen obrazcu za urejanje. Razlika je v tem, da so vnosno polje in potrditvena polja onemogočeni za vpisovanje. Brisanje skupine ni možno v primeru, da imamo v podatkovni bazi inštrument, ki spada v skupino, ki jo brišemo. Najprej je potrebno pobrisati oz. spremeniti inštrumente.

#### 4.2.4. Zaslonska maska inštrumenti

Zaslonska maska omogoča dodajanje, urejanje in brisanje inštrumentov. Privzeto se prikaže obrazec za dodajanje, lahko pa z izbirnim gumbom prikažemo obrazec za urejanje ali brisanje inštrumentov. Obvezno mora biti vnesena vsaj ena skupina inštrumentov v podatkovni bazi, da lahko uporabljamo to zaslonsko masko. Uporablja jo lahko vodja proizvodnje ali prodajalec.

## Elementi, ki sestavljajo zaslonsko masko inštrumenti:

### **Polja s seznamom:**

*Izberi inštrumenta* – omogoča izbiro inštrumenta iz seznama

*Skupina inštrumenta* – omogoča izbiro skupine v katero spada inštrument iz seznama

*Oblika izreza* – omogoča izbiro oblike izreza iz seznama

*Oblika na plošči* – omogoča izbiro oblike inštrumenta na plošči iz seznama

### **Vnosna polja:**

*Ime inštrumenta* – tekstovno polje, ki vsebuje ime inštrumenta

*Izrez dx* – numerično polje, ki vsebuje širino izreza na plošči v milimetrih za inštrument v primeru, da je oblika izreza pravokotnik

*Izred dy* – numerično polje, ki vsebuje višino izreza na plošči v milimetrih za inštrument v primeru, da je oblika izreza pravokotnik

*Izrez radij* – numerično polje, ki vsebuje radij izreza na plošči v milimetrih za inštrument v primeru, da je oblika izreza krog

*Plošča dx* – numerično polje, ki vsebuje širino inštrumenta na plošči v milimetrih v primeru, da je oblika izreza pravokotnik

*Plošča dy* – numerično polje, ki vsebuje višino inštrumenta na plošči v milimetrih v primeru, da je oblika izreza pravokotnik

*Plošča radij* – numerično polje, ki vsebuje radij izreza na plošči v milimetrih za inštrument v primeru, da je oblika izreza krog

*Globina inštrumenta* – numerično polje, ki vsebuje globino inštrumenta

*Slika inštrumenta* – numerično polje, ki vsebuje pot do bitne slike inštrumenta

### **Gumbi:**

*Prebrskaj* – omogoča enostavno vnašanje poti bitne slike inštrumenta

*Potrdi* – omogoča ažuriranje podatkov v podatkovni bazi

*Zapri* – zapre zaslonsko masko in nas vrne v glavni meni

Obrazci za dodajanje, urejanje in brisanje so po obliki skoraj enaki. Razlika med obrazcema dodaj in uredi je, da ima obrazec dodaj inštrument skrito polje s seznamom izberi inštrument. Obrazca dodaj in uredi zahtevata od uporabnika, da izpolni vsa polja, razen slike inštrumenta. Obrazec izbriši ima onemogočena vsa polja, razen polja izberi inštrument. Gumb prebrskaj prav tako ni viden na tem obrazcu.

Pri vseh obrazcih se glede na izbiro oblike izreza prikazujejo vnosna polja izrez dx, izrez dy in izrez radij. Če je izbrana oblika izreza pravokotnik, se prikažeta polji izrez dx in izrez dy, če pa je krog se prikaže polje izrez radij. Enako velja za polja plošča dy, plošča dy in plošča radij.

Gumb potrdi, shrani spremembe v podatkovno bazo, pri pogoju, da so vsa polja pravilno izpolnjena. Gumb zapri, zapre zaslonsko masko in nas vrne na glavni meni.

Slika 35: Zaslonska maska z obrazcem za brisanje inštrumenta

#### 4.2.5. Zaslonska maska predloge

Ta zaslonska maska je sestavljena iz dveh delov. V prvem izberemo ploščo in predlogo, ki jo želimo uporabiti, v drugem pa postavljamo inštrumente na ploščo.

Elementi, ki sestavljajo prvi del zaslonske maske predloge:

##### **Polja s seznamom:**

*Izberi ploščo* – omogoča izbiro plošče iz seznama

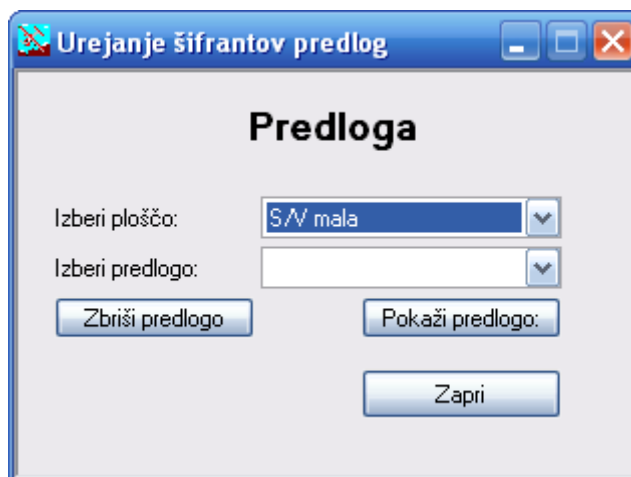
*Izberi predlogo* – omogoča izbiro predloge iz seznama

##### **Gumbi:**

*Zbriši predlogo* – izbriše izbrano predlogo iz podatkovne baze

*Pokaži predlogo* – prikaže izbrano predlogo

V prvem delu imamo možnost izbire nove (prazna ploščo) ali obstoječe predloge. Če želimo izbrati novo predlogo moramo pustiti polje izberi predlogo prazno. Imamo tudi možnost brisanja predloge. Za prikaz predloge je obvezno, da imamo vneseno vsaj eno ploščo, sicer ne moremo nadaljevati.



Slika 36: Prvi del zaslonske maske predloge

Elementi, ki sestavljajo drugi del zaslonske maske za urejanje predloge:

**Polja s seznamom:**

*Izbrana plošča* – onemogočeno polje za izbiranje, ki prikazuje iz katere plošče je predloga ustvarjena

*Izbrana predloga* – onemogočeno polje za izbiranje, ki prikazuje iz katere predloge izhajamo

*Izberi inštrument* – omogoča izbiro inštrumentov iz seznama

**Vnosna polja:**

*Shrani predlogo* – tekstovno polje, ki vsebuje ime, pod katerega želimo shraniti ustvarjeno predlogo

*Koordinata x* – omogoča določanje koordinate x za izbrani inštrument

*Koordinata y* – omogoča določanje koordinate y za izbrani inštrument

**Potrditveno polje:**

*Prezrcali čez y os* – omogoča zrcaljenje celotne predloge čez y os

**Gumbi:**

*Menjaj predlogo* – omogoča menjavo izbrane predloge

*Slika inštrumenta* – omogoča prikaz bitne slike izbranega inštrumenta iz seznama izberi inštrument oziroma iz inštrumenta na plošči

*Dodaj na ploščo* – doda inštrument, ki je izbran v polju izberi inštrument na ploščo

*Odstrani s plošče* – odstrani označen inštrument iz plošče

*Izvozi v PNG* – ustvari PNG datoteko

*Shrani* – shrani podatke o koordinatah v podatkovno bazo

*Zapri* – zapre zaslonsko masko in prikaže glavni meni

Ko izberemo ploščo lahko nadaljujemo z urejanjem predloge. V primeru, da smo izbrali napačno predlogo, jo lahko zamenjamo z gumbom menjaj predlogo. Sicer pa lahko določimo ime nove predloge ali pa prepisemo že obstoječo. Za vsako predlogo posebej najprej postavimo zelene inštrumente na ploščo in jih nato premikamo po želji. Imamo tudi možnost pogleda bitne slike posameznega inštrumenta. Ko dodamo inštrument na ploščo, se le ta postavi v zgornji levi kot in ga lahko nato prestavljamo.

Inštrumente lahko na plošči premikamo na naslednje načine:

- z miško
- s tipkovnico

### Določanje pozicije z miško

Če želimo prestaviti inštrument, se postavimo z miško v območje le tega in držimo levi gumb na miški ter premikamo inštrument po risalni površini. Če držimo poleg še tipko *Ctrl* na tipkovnici, lahko izberemo več inštrumentov in jih hkrati premikamo po risalni površini. Težava pri tem načinu postavljanja je, da ne moremo vedno postaviti inštrumenta na točno določeno točko, ampak je običajno nekaj odstopanja.













### Določanje pozicije s tipkovnico

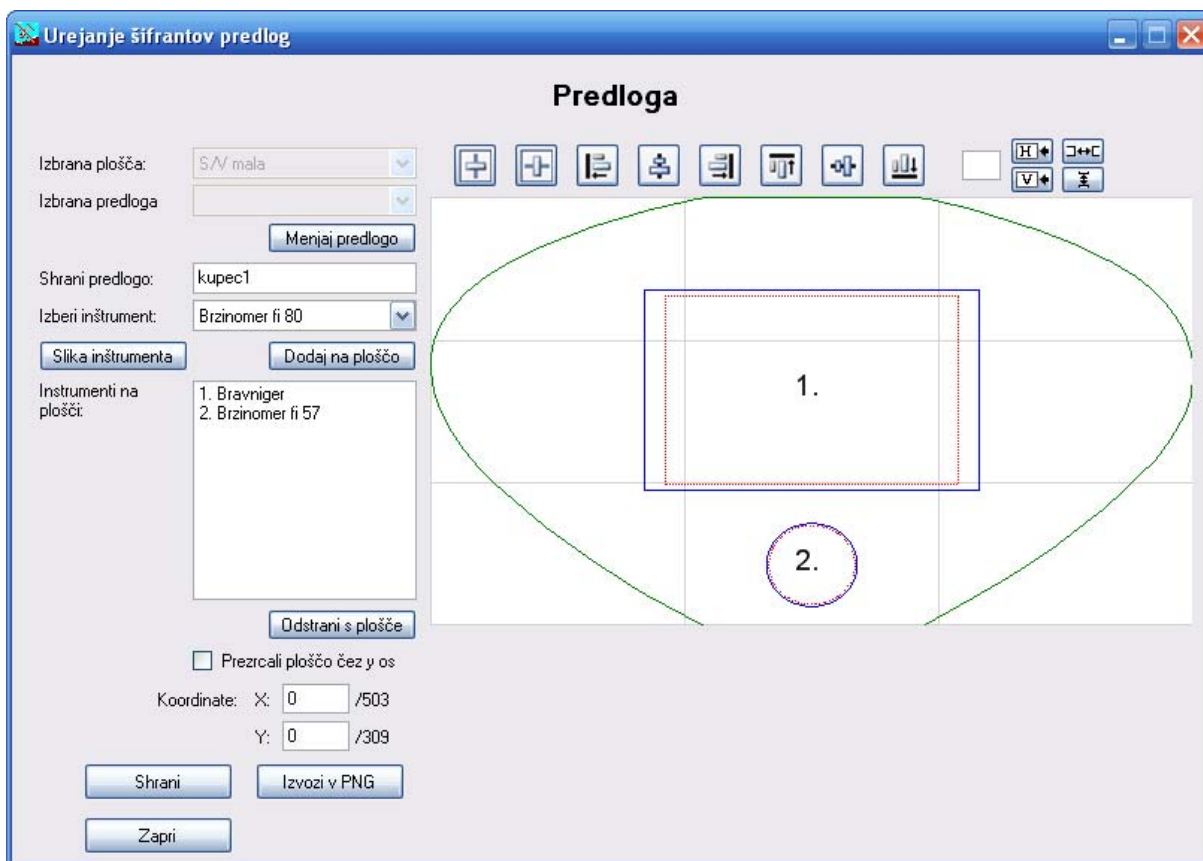
S pomočjo tabulatorja, smernih tipk in še nekaterih dodatnih tipk je mogoče samo s tipkovnico postavljati inštrumente. Ker danes praktično ni več računalnika brez miške, bom opisal le nekatere možnosti premikanja s tipkovnico. Včasih se zgodi, da je z miško težko do milimetra natančno določiti pozicijo, zato lahko uporabimo tipkovnico. Ko imamo izbrane inštrumente, jih lahko pomikamo za eno enoto v vse smeri s pomočjo kombinacije tipk *Ctrl* + *smerna tipka*. Druga, nekoliko bolj posebna možnost premikanja pa je kombinacija tipk *Ctrl* + *Shift* + *smerna tipka*, ki omogoča premikanje v določeni smeri za 10 enot.

Kadar nam ta način premikanja ne zadošča, lahko postavimo inštrument tudi tako, da vnesemo koordinate x in y, do ene decimalke natančno.

Da je postavljanje bolj enostavno, je dodanih še nekaj gumbov za več vrst poravnav. Kadar imamo izbranih več inštrumentov, je vedno eden izmed njih označen z rumeno barvo, kar nam pove, da je ta inštrument sidrni, ter da se vse poravnave vršijo glede na ta inštrument.

Imamo naslednje načine poravnav:

-  centriraj horizontalno sredinsko
-  centriraj vertikalno sredinsko
-  poravnaj z levim robom inštrumenta
-  poravnaj vertikalno s sredino inštrumenta
-  poravnaj z desnim robom inštrumenta
-  poravnaj z zgornjim robom inštrumenta
-  poravnaj horizontalno s sredino inštrumenta
-  poravnaj s spodnjim robom inštrumenta
-  pridobi horizontalni razmik med dvema izbranimi inštrumentoma
-  naredi horizontalni razmik med izbranimi inštrumenti
-  pridobi vertikalni razmik med dvema izbranimi inštrumentoma
-  naredi vertikalni razmik med izbranimi inštrumenti



Slika 37: Drugi del zaslonske maske predloge

Ko zaključimo s postavljanjem, lahko celotno aplikacijo izvozimo v PNG format ali pa shranimo spremembe v podatkovno bazo. Preden se podatki ažurirajo v podatkovni bazi oziroma se ustvari PNG datoteka, se izvede preverjanje prekrivanja inštrumentov. Če se inštrumenti prekrivajo ali so si preblizu, se nam prikaže obvestilo in spremembe v bazi se ne ažurirajo.

### 4.3. Izzivi pri izdelavi aplikacije

Kot pri večini projektov je tudi pri tem prišlo do nekaterih problemov oziroma izzivov. Pomembnejši izzivi, ki jih bom podrobneje opisal so izrisovanje vektorske slike plošče na risalno površino ter določanje razmerij med risalno površino, realno velikostjo plošče ter vektorsko sliko plošče.

#### 4.3.1. Izrisovanje vektorske slike na ploščo

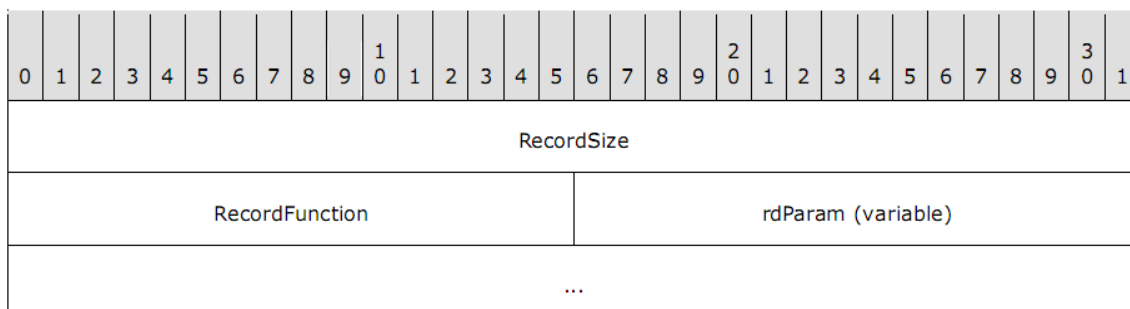
Od podjetja sem dobil vektorske slike plošč v formatu WMF. WMF datoteka je množica različnih zapisov, tako imenovanih WMF zapisi, ki vsebujejo ukaze za risanje grafike, definicije objektov in lastnosti [1]. Vsaka WMF datoteka vsebuje na začetku, tako imenovano glavo, ki vsebuje verzijo WMF-ja, njegovo velikost in število objektov definiranih v njej.

Za uspešen izris vektorske slike v formatu WMF je bilo potrebno preučiti dokumentacijo o formatu. Ob napačni interpretaciji lahko namreč dobimo popolnoma drugačno sliko, kot bi jo pričakovali, zato je pomembno, da upoštevamo strukturo WMF formata. Podrobneje je bilo

potrebno preučiti WMF glavo ter WMF zapise. WMF zapise lahko združimo v nekaj glavnih tipov zapisov:

- *bitni zapisi* – upravljajo z bitno sliko
- *kontrolni zapisi* – določajo začetek in konec WMF datoteke
- *risalni zapisi* – določajo vrstni red risanja grafike
- *zapisi objektov* – ustvarja in upravlja grafične objekte
- *zapisi stanj* – določa in upravlja z konfiguracijo grafike

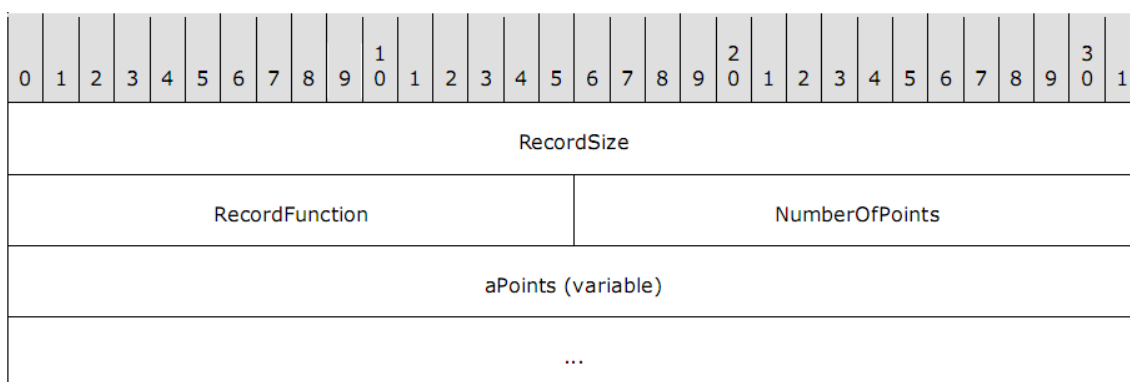
Vsi WMF zapisi, razen kontrolnih zapisov imajo enako strukturo. Zapisi vsebujejo podatke o velikosti zapisa (RecordSize - 32 bitov), funkciji zapisa (RecordFunction - 16 bitov) in parametri zapisa (rdParam - 16 bitov).



Slika 38: WMF zapis

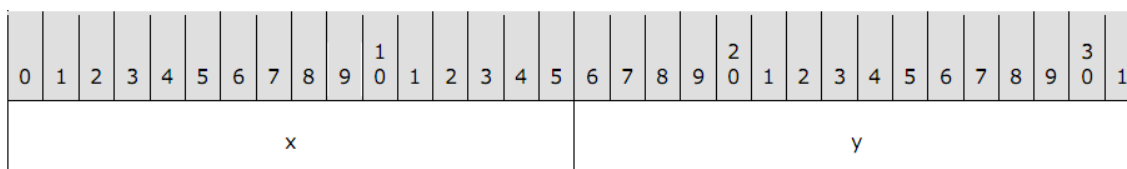
Risalni zapisi določajo like (kvadrat, elipsa, zaobljen kvadrat, črta, množica povezanih črt in drugi), ki omogočajo izris grafike. Zaradi obsežnosti bom podrobneje opisal samo risalni zapis za množico povezanih črt, imenovan meta\_polyline. Če želi bralec izvedeti več tudi o ostalih zapisih, lahko najde podatke na spletu[2].

#### Natančnejši opis risalnega zapisa meta\_polyline:



Slika 39: Risalni zapis meta\_polyline

Zapis je sestavljen iz velikosti zapisa (RecordSize – 32 bitov), funkcije zapisa (RecordFunction – 16 bitov), števila povezanih točk (NumberOfPoints – 16 bitov) ter koordinatami posamezne točke (aPoints(variable) – 32 bitov). Če želimo dobiti koordinate točke, moramo poznati obliko zapisa (aPoints), ki je lepo razvidna iz naslednje slike:



Slika 40: Zgradba zapisa točke(aPoints)

Prvih 16 bitov nam poda koordinato x, drugih 16 bitov pa koordinato y. Ko smo vse to prebrali, smo dobili množico povezanih točk, ki jih nato lahko narišemo na zeleno površino. Pri tem je pomembno, da upoštevamo dolžino zapisa, kajti ob napačnem podatku o dolžini, se lahko zgodi, da zapisa ne preberemo v celoti ali pa naslednji zapis interpretiramo kot del trenutnega zapisa, kar pa je napačno in se lahko zgodi, da dobimo napako pri branju.

### 4.3.2. Določanje razmerij

Aplikacija ima določeno širino in višino risalne površine. Ker ima vsaka plošča običajno drugačne mere in s tem tudi drugačno razmerje stranic, je potrebno upoštevati to razmerje pri risanju. Pojavi pa se še problem vektorske slike. Lahko se zgodi, da ima vektorska slika drugačno razmerje stranic, kot jo ima plošča v realnosti. Sedaj se pojavi vprašanje, kako uskladiti razmerja med risalno površino, dejanskimi merami, vektorsko sliko in inštrumenti, ki jih postavljamo na risalno površino.

Spremenljivke, ki jih bom uporabil pri prikazu izračunov:

- $w_{rp}$  – širina risalne površine
- $h_{rp}$  – višina risalne površine
- $w_{vs}$  – širina vektorske slike
- $h_{vs}$  – višina vektorske slike
- $w_{pl}$  – širina plošče
- $h_{pl}$  – višina plošče
- $w_{in}$  – širina inštrumenta
- $h_{in}$  – višina inštrumenta

Kot osnovo sem vzel razmerje vektorske slike, zato je bilo potrebno najprej prilagoditi razmerje risalne površine tako, da ustreza razmerju vektorske slike. Risalna površina ima določeno maksimalno širino in višino, glede na velikost okna. Vektorsko sliko ustrezno pomanjšamo tako, da je ena stranica enaka maksimalni velikosti stranice na risalni površini, druga stranica risalne površine pa se prilagodi tako, da ohranimo isto razmerje kot ga ima vektorska slika. Najprej sem izračunal razmerje širine in višine stranic vektorske slike in risalne površine.

$$r_{vs} = \frac{w_{vs}}{h_{vs}} \quad (1)$$

$$r_{rp} = \frac{w_{rp}}{h_{rp}} \quad (2)$$

Ko imamo izračunana razmerja preverim:

- če je:  $w_{rp}=w_{rp}$  ter  $h_{rp}=h_{rp}$

Za tem izračunamo razmerje med širino plošče in širino risalne površine ter med višino plošče in višino risalne površine.

$$r_w = \frac{w_{pl}}{w_{rp}} \quad (3)$$

$$r_h = \frac{h_{pl}}{h_{rp}} \quad (4)$$

S tem smo dobili koeficiente raztegnjenosti v širino in višino. To pomeni, da nam predstavlja eden piksel na zaslonu  $r_w$  (3) milimetrov v širino in  $r_h$  (4) milimetrov v višino. Na tem mestu bi omenil še to, da kadar pomikamo inštrumente z miško, jih lahko prestavljamo za najmanj eden piksel in ker eden piksel na zaslonu pomeni  $r_w$  oz.  $r_h$  milimetrov v realnosti, se pri koordinatah  $x$  in  $y$  izpisujejo decimalna števila. Ker imamo v bazi podano velikost inštrumentov v milimetrih, je potrebno tudi pri risanju le teh upoštevati koeficiente zato dobimo, da je:

$$w_{in} = \frac{w_{in}}{r_w} \quad (5)$$

$$h_{in} = \frac{h_{in}}{r_h} \quad (6)$$

Kadar premikamo inštrument z miško, se koordinata  $x$  in  $y$  izračuna v milimetre, tako da vzamemo koordinate miške na risalni površini v pikslih in jih pomnožimo z  $r_w$  za  $x$  oz. z  $r_h$  za  $y$  koordinato.

S temi izračuni imamo vse potrebno za risanje na risalni površini, tako, da se vektorska slika, velikost inštrumentov in velikost plošče skladajo, ter se pravilno izpisujejo koordinate  $x$  in  $y$  pri posameznem inštrumentu.

## 5. Sklepne ugotovitve

V sklopu diplomske naloge so bile preučene želje in potrebe podjetja in na podlagi le teh je bila razvita aplikacija, ki omogoča urejanje šifrantov plošč, inštrumentov in skupin inštrumentov ter postavljanje inštrumentov na plošče. Izdelan je bil prijazen uporabniški vmesnik. Implementirana je možnost shranjevanja posameznih predlog v bazo oziroma izvoz aplikacije v PNG datoteko, ki jo lahko posredujemo strankam ali uporabimo za kakšne druge namene.

Nadaljnji razvoj aplikacije lahko poteka v smeri izboljšanja uporabniškega vmesnika. Tu mislim predvsem na možnost vnašanja, izrisovanja in izpisovanja dodatnih parametrov, kot so koordinate lukenj za pritrditev posameznih inštrumentov ter kakšnih drugih specifičnih lukenj ali oblik. Ta funkcionalnost bo zelo dobrodošla, ko bodo plošče izrezovali s CNC stroji. Zaenkrat v podjetju izrezujejo plošče ročno, vendar načrtujejo posodobitev s CNC stroji, ki bodo ta postopek avtomatizirali. CNC stroj deluje tako, da prebere iz datoteke vse izreze, ki jih mora narediti, zato je potrebno dostaviti pravilen format datoteke. Ker zaenkrat v podjetju še nimajo CNC stroja in tudi ne vemo kakšen format datoteke bo stroj potreboval, ta funkcionalnost ni bila implementirana, je pa to vsekakor možno pričakovati v prihodnosti.

Danes, ko je internet zelo razširjen in ga ima že skoraj vsako gospodinjstvo, bi bilo dobro v bodoče načrtovati spletno aplikacijo, ki bi imela možnost izbiranja plošč in inštrumentov, ter izdelovanja predlog. S tem bi omogočili, da bi si lahko potencialne stranke že same pripravile svojo predlogo ali več predlog in jo posredovali podjetju kar po elektronski pošti.

## 6. Priloge

### 6.1. Inštrumenti

Inštrument je naprava, ki pilotu prikazuje informacije potrebne za letenje. Obstajajo inštrumenti, ki so bolj mehanski in nimajo vgrajene kakšnih elektronskih delov in inštrumenti, ki so popolnoma elektronski ter delujejo podobno kot računalnik.

Zaradi lažje predstave bom prikazal najpogosteje uporabljene inštrumente in jih na kratko opisal.

#### 6.1.1. Brauniger

Brauniger je zelo dovršen inštrument, ki ga izdelujejo že več kot 25 let. Je eden najpomembnejših in tudi obveznih inštrumentov na letalu in je povsem elektronski inštrument, brez katerega praktično ni mogoče leteti. Brauniger je sestavljen iz procesorja, pomnilnika in programske opreme, ki je naložena na njem [5]. Ob vklopu se izvrši avtomatsko samotestiranje inštrumenta, ki takoj javi morebitne napake. Kadarkoli ga je možno nadgraditi z novejšo programsko opremo, brez dodatnih stroškov.

Inštrument prikazuje na zaslonu le najpomembnejše informacije pri letenju, kot so hitrost letenja, višina letenja, hitrost padanja oz. dviganja letala, količina goriva, temperatura motornega olja in vode. Poleg teh informacij ima še mnogo drugih podatkov, ki so dostopni preko gumbov. Zaradi obsežnosti podatkov ima strukturiran meni za hiter dostop do informacij in LCD prikazovalnik za enostavno odčitavanje le teh. Brauniger deluje kot črna skrinjica na letalu, saj shranjuje poročila o zadnjih nekaj deset letih. Poročilo vsebuje vse informacije, o dogajanjih na letalu.



Slika 41: Brauniger

Ta inštrument je tako izpopolnjen, da ne potrebujemo nobenega dodatnega inštrumenta za pilotiranje. Vendar se v praksi izkaže, da je dobro imeti še druge inštrumente.

### 6.1.2. Radijska postaja

Radijska postaja je zelo uporaben inštrument, ki omogoča, da pilot komunicira z letališčem vendar ni obvezna. Če želimo potovati z letalom po drugih letališčih je potrebno zaprositi za dovoljenje za pristane, kar se lahko stori s pomočjo tega inštrumenta. Služi lahko tudi za komunikacijo med letali v zraku.



Slika 42: Radijska postaja

### 6.1.3. Ostali inštrumenti

Pogosto si piloti naročijo še dodatno inštrumente kot so brzinomer, variometer in podobno. Brzinomer prikazuje hitrost letenja, variometer pa hitrost padanja oziroma dviganja letala. Kljub temu, da nam vse te informacije prikazuje brauniger, piloti naročijo te inštrumente zaradi dodatne varnosti. Brauniger je en sam inštrument, ki nam prikazuje vse informacije in v primeru okvare le tega ostanemo brez vsega, zato je priporočljivo imeti vgrajene tudi mehanske inštrumente. Za elektronske inštrumente obstaja večja možnost okvare, kot pri mehanskih inštrumentih.



Slika 43: Brzinomer



**Slika 44: Variometer**

Za konec bom predstavil še sliko plošče letala z vgrajenimi inštrumenti, ki prikazuje inštrumente med delovanjem.



**Slika 45: Plošča z delujočimi inštrumenti**

## 7. Viri

[1] Windows metafile format specification. Dostopno na:  
<http://msdn.microsoft.com/en-us/library/cc215212.aspx>

[2] Drawing Record Types. Dostopno na:  
<http://msdn.microsoft.com/en-us/library/cc215405.aspx>

[3] T. Mohorič, *Načrtovanje relacijskih podatkovnih baz*, Ljubljana:BI-TIM d.o.o., 1997

[4] M. Krisper, R. Rupnik, A. Zrnec, M. Bajec, R. Osojnik, *EMRIS*, Ljubljana: Vlada Republike Slovenije, 2004, Strukturni razvoj

[5] Brauniger flugelectronic. Dostopno na:  
<http://www.brauniger.com/english/products/alphamfd/introduction.html>

## **Izjava o samostojnosti dela**

Izjavljam, da sem diplomsko delo izdelal samostojno pod vodstvom mentorja doc. dr. Marka Bajca. Izkazano pomoč drugih sodelavcev sem v celoti navedel v zahvali.

Branko Pinter