

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Jure Možina

**Razvoj informacijske rešitve za planiranje in koordiniranje
prostorskih in drugih virov v Cankarjevem domu**

DIPLOMSKO DELO
NA VISOKOŠOLSLEM STROKOVNEM ŠTUDIJU

Mentor: prof. dr. Marko Bajec

Ljubljana, 2010



Št. naloge: 00501/2010

Datum: 15.03.2010

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

Kandidat: **JURE MOŽINA**

Naslov: **RAZVOJ INFORMACIJSKE REŠITVE ZA PLANIRANJE IN
KOORDINIRANJE PROSTORSKIH IN DRUGIH VIROV V
CANKARJEVEM DOMU**
**THE DEVELOPMENT OF A TOOL SUPPORT FOR RESOURCE
MANAGEMENT IN »CANKARJEV DOM«**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija

Tematika naloge:

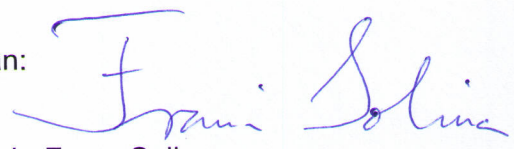
V okviru diplomske naloge razvijte programski paket za podporo kongresni dejavnosti. Pri zbiranju in opredelitvi zahtev se zgledujte po procesih, ki se za potrebe organizacije in upravljanja s kulturnimi in kongresnimi projekti izvajajo v Cankarjevem domu v Ljubljani. Za razvoj programskega paketa lahko uporabite poljubno arhitekturo, informacijske tehnologije in metodologije razvoja informacijskih sistemov, vendar vsak korak razvoja ustrezno dokumentirajte ter podajte razloge za posamezne odločitve.

Mentor:


prof. dr. Marko Bajec



Dekan:


prof. dr. Franc Solina

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani/-a JURE MOŽINA,

z vpisno številko 63040253,

sem avtor/-ica diplomskega dela z naslovom:

RAZVOJ INFORMACIJSKE REŠITVE ZA PLANIRANJE IN KOORDINIRANJE PROSTORSKIH IN DRUGIH VIROV V CANKARJEV DOMU

S svojim podpisom zagotavljam, da:

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

Prof. Dr. Marko Bajec

in somentorstvom (naziv, ime in priimek)

- 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 zbirki »Dela FRI«.

V Ljubljani, dne 15.6.2010

Podpis avtorja/-ice: _____

Zahvala

Svojemu mentorju Prof. Dr. Marku Bajcu se iskreno zahvaljujem za vso podporo, nasvete in praktično pomoč.

Prav tako se zahvaljujem mojemu nadrejenemu, Mensurju Hodžiću, dipl. org. man. za vodenje pri izdelavi diplomskega dela.

Za potrpežljivost in vzpodbudo se zahvaljujem svojim staršem, ki so mi omogočili študij in me vsa leta podpirali.

Seznam uporabljenih kratic

IR	-	Informacijska rešitev
IS	-	Informacijski sistem
CD	-	Cankarjev dom
ITK	-	Informacijske tehnologije in komunikacije
.NET	-	Skupek komponent podjetja Microsoft
MySQL-		Sistem za upravljanje s podatkovnimi bazami
eEpc diagram	-	Diagram za izdelavo podrobnega modela procesa
UML	-	Unified modeling Language – jezik za objektno modeliranje
RUP	-	Rational Unified Process – ogrodje objektnega procesnega modela
C#	-	Programski jezik
JavaScript	-	Objektni skriptni programski jezik
jQuery	-	Objektni skriptni programski jezik, ki temelji na knjižnicah javascript-a
CSS	-	Cascading Style Sheets – predloge, ki določajo izgled spletnih strani
HTML	-	Hyper Text Transfer Protocol – HTTP protokol

Kazalo

Povzetek	1
Abstract	2
1. Uvod	3
2. Življenski modeli razvoja IS.....	5
2.1. Zaporedni ali slapovni model	5
2.2. Iterativni model	6
2.3. Prototipni model	7
2.4. Inkrementalni model	8
2.5. Uporaba življenskih modelov	9
3. Metodologije	10
3.1. Kaj je metodologija.....	10
3.2. Vrste metodologij.....	10
3.3. Metodlogija v praktičnem primeru	10
4. Faze razvoja IS	12
4.1. Zajem in specifikacija zahtev	12
4.1.1. Kratek opis namena IR.....	12
4.1.2. Opis problemske domene oziroma funkcionalnih zahtev.....	13
4.1.3. Opis nefunkcionalnih zahtev	16
4.1.4. Opis vmesnikov	16
4.2. Analiza	17
4.2.1. Izdelava modela sistema	18
4.2.1.1. Podatkovni model - konceptualni model	18
4.2.1.2. Procesni model - eEpc	20
4.2.1.3. Model procesne logike – strukturiran jezik.....	21
4.2.1.4. UML diagram –Diagram primerov uporabe	23
4.2.2. Izdelava prototipov	31
4.2.2.1. Prototip št.1- Marketing (Rezervacije)	31
4.2.2.2. Prototip št.2. - Programski koordinator (Projekti)	33
4.2.2.3. Prototip št.3. - Vodja projekta (Dispozicije)	34
4.2.2.4. Prototip št.4. - Vodja izvedbe (Urniki, Dispozicije)	36
4.2.3. Opredelitev strategije testiranja	37

4.3. Načrtovanje	38
4.3.1. Načrt zgradbe sistema	38
4.3.2. Načrt podatkovne baze	38
4.3.3. Načrt programskih modulov.....	41
4.3.3.1. Opis programskih modulov za vmesnik »Rezervacije«	42
4.3.3.2. Opis programskih modulov za vmesnik »Projekti«	43
4.3.3.3. Opis programskih modulov za vmesnik »Dispozicije«.....	44
4.3.3.4. Opis programskih modulov za vmesnik »Urniki«	44
4.4. Implementacija	46
4.4.1. Implementacija vmesnika »Rezervacije«	46
4.4.2. Implementacija vmesnika »Projekti«	47
4.4.3. Implementacija vmesnika »Dispozicije«.....	48
4.4.4. Implementacija vmesnika »Urniki«	49
4.4.4.1. Grafični prikaz urnikov projektov	50
4.4.4.2. Grafični prikaz urnikov oseb	52
4.5. Testiranje	52
4.6. Vzdrževanje in nadzor	55
5. Zaključek	56
Slike.....	57
Tabele	58
Literatura	59

Povzetek

V Cankarjevem domu vsak dan potekajo kulturni in kongresni projekti. Za pripravo in izvedbo vsakega izmed njih je potrebna dobra organizacija, koordinacija, sodelovanje in interakcija med nastopajočimi v procesu. Za vsak projekt potekajo najrazličnejše aktivnosti oziroma urniki projekta, ki so opredeljeni po prostorskih virih. Znotraj teh urnikov so opredeljeni urniki izvedbenih delavcev. V celotnem procesu nastopa veliko število ljudi in drugih virov, ki na najrazličnejši način opravljajo svojo funkcijo. Pogoj, da tak zapleten sistem v okolju deluje mora biti seveda informacijsko podprt. Tako sem razvil informacijsko reešitev za planiranje in koordiniranje prostorskih in drugih virov. V diplomskem delu sem opisal metodologije ter življenske modele, ki nastopajo pri razvoju informacijskega sistema, njihove značilnosti ter uporabo pri razvoju IR.

Osrednji oziroma glavni del diplomskega dela obsega podroben opis razvoja informacijske rešitve. Pri razvoju srečamo ključne faze tega razvoja. Zajem zahtev, ki opredeljuje tako opis problemske domene, kot opis zahtev. V analizi so zbrane vse zahteve in predstavljene oziroma interpretirane z različnimi modeli. Za izdelavo modelov sem uporabil orodje Power designer 12. V fazi načrtovanja sem izdelal načrt podatkovne baze ter načrt programskih modulov s pomočjo orodja Edraw Max. V fazi implementacije sem predstavil celotno informacijsko rešitev sestavljeno iz štirih delov oziroma aplikacij in predstavil predvsem njihove ključne funkcionalnosti. Aplikacija »Rezervacije« za vnos projekta in samo rezervacijo s strani marketinga, aplikacija »Projekti« namenjena programskemu koordinatorju za urejanje podatkov projekta. Vodjem projektov je namenja aplikacija »dispozicije« za izdelavo in urejanje dispozicij. Med zanimivejšimi je aplikacija »Urniki«, ki med drugim omogoča grafičen pregled urnikov. Za izvedbo implementacije sem uporabil orodje Visual Studio 2008. Na koncu diplomskega dela sem opredelil še fazi testiranja in vzdrževanja informacijske rešitve.

KLJUČNE BESEDE:

razvoj informacijskega sistema, življenski modeli, Power designer 12, Edraw Max, Visual Studio 2008.

Abstract

Cultural and congress activities take place in Cankarjev dom on a daily basis. Preparing and executing each one of them needs very good organization, coordination, cooperation and interaction among workers involved in the process. Every project consists of many activities starting at different times and occurring in different spaces. Staff timetables are defined within these schedules. The entire process involves a lot of people and other resources performing various functions. Such a complex system needs to be supported by an information system in order to work. That is why I developed an information system for planning and coordinating space and other resources. The degree thesis describes the methodology and life models required for the development of an information system as well as their characteristics and application.

The main part of the thesis presents a detailed explanation of the development of the information system. Four main stages of development are described. Defining requirements and problem domain. The analysis covers all the requirements, together with the various model interpretations. The models were made using Power designer 12. During the planning stage, I designed a database plan as well as a program modules in Edraw Max. The implementation stage presents the complete information system, consisting of four parts (applications), showing, especially, their major functionalities. The application »Rezervacije« is used for project input and making reservations by the marketing department. The application »Projekti« is meant for the programme coordinator for managing project data. Project managers shall use the application »Dispozicije« which enables the preparation and management of dispositions. One of the most interesting applications is »Urniki«, which, among others, enables a graphical overview of schedules. Implementation was facilitated in Visual Studio 2008. The thesis concludes with a description of two additional stages, namely, testing and maintenance of the information system.

KEYWORDS:

Development of information system, life models, Power designer 12, Edraw Max, Visual Studio 2008.

1. Uvod

V Cankarjevem domu vsak dan potekajo najrazličnejše prireditve oziroma dogodki. Dejavnost v Cankarjevem domu je produkcija, koprodukcija, organizacija in posredovanje kulturno-umetniških, kongresnih, sejemskih, protokolarnih ter drugih prireditev, razstav in festivalov. Ker je predvsem kulturno središče, kulturi in umetnosti namenjajo letno več kot dve tretjini razpoložljivih terminov v dvoranah. Želijo ohraniti tudi vodilno vlogo kot osrednji slovenski kongresni center z največjim številom mednarodnih kongresov in mobilno skupino profesionalnih kongresnih organizatorjev na drugih slovenskih kongresnih prizoriščih. Na razpolago je približno 20 prostorov znotraj Cankarjevega doma. Poleg tega ima Cankarjev dom najete tudi zunanje prostore.

Za pripravo in izvedbo vsakega izmed njih je potrebna dobra organizacija, koordinacija, sodelovanje in interakcija med nastopajočimi v procesu. Za vsak dogodek potekajo najrazličnejše aktivnosti oziroma urniki prireditve (večji je dogodek, več je urnikov). Vsak urnik je v glavnem časovno, prostorsko in tipsko opredeljen. Urnik pa je opredeljen tudi z sodelujočimi oziroma izvedbenimi delavci ki so razporejeni s strani vodij posameznih oddelkov in nastopajo v okviru urnika projekta. Na leto je zabeleženih približno čez 800 dogodkov skupaj približno 8.000 urnikov, kar je v povprečju 600 urnikov oziroma aktivnosti na mesec!

Vsak kulturni ali kongresni projekt, ki je izveden v Cankarjevem domu se začne z interakcijo med marketingom in naročnikom. Ob uspešnem dogovoru marketing vpiše projekt v zbirko projektov. Gre za začetne okvirne podatke o projektu, ki jih pridobi od naročnika (še preden je projekt potrjen). Poleg tega pa lahko izvede tudi zahtevek za rezervacijo virov oziroma prostorov, katero kasneje programski koordinator potrdi ali ovrže. Nekateri projekti so zelo obsežni, zahtevni in posledično zahtevajo veliko napora. Glavni namen tega je, da na podlagi teh podatkov lahko ugotovimo ali je izvedba celotnega projekta možna in katere vire in storitve lahko zagotovimo ob zahtevah naročnika. Kasneje, če je projekt potrjen, se ga dokončno vpiše oziroma potrdi v zbirki projektov, ki bodo izvedeni v Cankarjevem domu.

Naslednjo vlogo v tem procesu ima programski koordinator. Njegova vloga je, da od marketinga pridobi potrjene projekte in jih dokončno vpiše v seznam projektov in zabeleži ostale podatke. Glavna naloga programskega koordinatorja pa je, da za vsak projekt specificira seznam okvirnih urnikov, ki bodo nastopali znotraj nekega projekta. Gre predvsem za predhodne rezervacije prostorskih virov, zgolj datumsko in prostorsko. Včasih pa le potrdi urnike rezervacij, saj jih vnese že marketing ob vpisu projekta.

Ko je projekt potrjen, in so okvirni urniki (rezervacije) definirani, naslednji, ki nastopa v tem procesu je vodja projekta. Vodja projekta mora biti tesno povezan tako z naročnikom kot z programskim koordinatorjem. Njegova naloga je, da podrobno določi urnike na podlagi

okvirnih urnikov (rezervacij) programskega koordinatorja in o tem obvesti programskega koordinatorja, ki razčlenjene urnike potrdi ali ovrže.

Vodja projekta od naročnika pridobiva nadaljne specifikacije in zahteve, ki se tekom procesa spreminjajo oziroma bolje definirajo in specifikirajo. Vse te podatke se posredno vpiše v dispozicijo. Dispozicija je dokument, ki nazorno opisuje vse podrobnosti projekta. V dispoziciji so zabeleženi tudi urniki projekta in njihove podrobnosti. V primeru ko nastopi sprememba s strani vodje projekta oziroma konkretno v dispoziciji, obvesti programskega koordinatorja o spremembi. Programski koordinator pregleda vpisane spremembe in jih potrdi ali ovrže in o tem obvesti vodjo projekta. Tako nastane nova verzija dispozicije. Tekom časa se dispozicija spreminja in tako dobiva nove verzije.

Naslednja vloga vodje projekta je da v dispoziciji zabeleži seznam virov ki bodo prisotni na licu mesta, oziroma izda zahtevek za seznam opreme. V glavnem pa so zapisane odgovornosti posameznih izvedbenih oddelkov, ki jih morajo izpolniti. To so odgovornosti vodje izvedbenih oddelkov (oddelek za postavitve prizorišč, oddelek za potek prireditev, oddelek za ozvočenje, oddelek za osvetljavo...), ki so naslednji, in nastopaju v procesu. Tudi s temi osebami mora biti vodja projekta tesno povezan, in osnova za komunikacijo je dispozicija.

Vodja projekta specifikira seznam opreme oziroma virov v dispoziciji, ki naj bi bila prisotna ob določenem času na določeni lokaciji upoštevajoč za posamezne oddelke. Naloga vodij izvedbnih oddelkov je, da iz dispozicije izvleče podatke in tako potrdi ali ovrže razpoložljivost virov za čas ki je določen v dispoziciji. Glavna vloga izvedbenih vodij pa je, da razporeja izvedbene delavce in določa njihove urnike v okviru izvedbe. Skrbi za zagotavljanje virov in opreme, ki je potrebna za določen projekt.

V središču procesa je vodja projekta, saj je v interakciji z večino udeleženi v samem sistemu. Sprejemati mora zahteve in dodatne specifikacije od naročnika tekom časa, vendar z omejitvami. Tesno mora biti povezan z programskim koordinatorjem in z vodjami izvedbe. Tudi vodje izvedbe se morajo hitro odzvati na zahteve vodje projekta in smotrno razporejati izvedbene delavce. Programski koordinator pa ima veliko dela z razporejevanjem okvirnih urnikov oziroma rezervacijami in potrjevanje spremenjenih oziroma razčlenjenih urnikov projektov s strani vodje projekta.

V samem procesu oziroma sistemu je udeleženi veliko ljudi in drugih virov ki na najrazličnejši način opravljajo svojo funkcijo in razpolagajo z množico podatkov. Ob tako veliki količini podatkov, ki so med seboj prepleteni in odvisni pa je samo obvladovanje, interpretacija in neko poizvedovanje dokaj naporno. Pogoji, da lahko tak zapleten sistem v okolju deluje, mora biti ob taki količini podatkov, sistem informacijsko podprt. Tako sem razvil IR za planiranje in koordiniranje prostorskih in drugih virov.

2. Življenski modeli razvoja IS

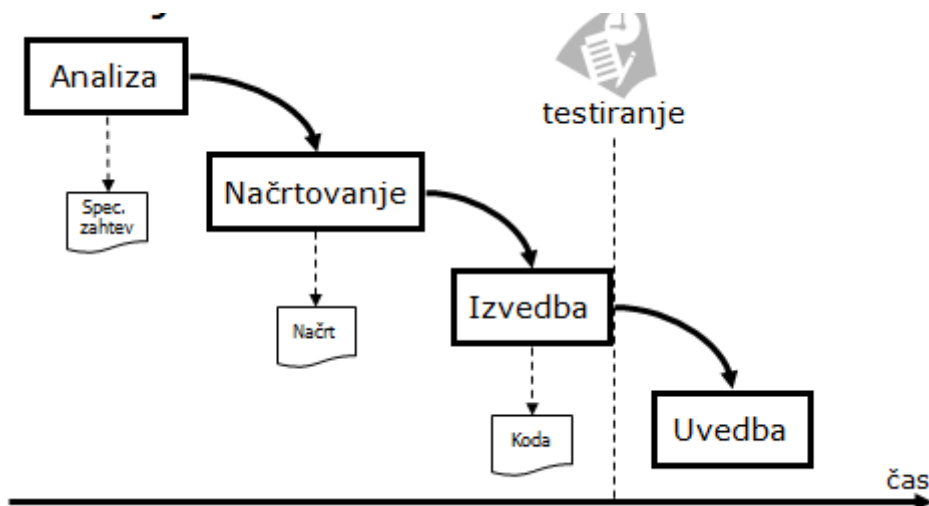
Proces nastajanja, razvijanja in delovanja informacijskega sistema običajno poimenujemo življenski cikel informacijskega sistema. Ciklus je sestavljen iz več faz in zajema v glavnem naslednje faze: analiza, načrtovanje, izvedba in vzdrževanje. Odvisno od pristopa problema in metodologije, je lahko več vmesnih faz. Celoten ciklus je odvisen od metodologije, ki jo uporabljamo pri razvoju IS. Življenski model razvoja informacijskega sistema torej pove, v kakšnem sledju in na kakšen način si v okviru razvoja IS sledijo posamezne faze.

Poznamo več vrst življenskih modelov:

- zaporedni ali slapovni
- iterativni
- prototipni
- inkrementalni

2.1. Zaporedni ali slapovni model

Zaporedni model je eden najstarejših modelov strukturnega razvoja IS. Pojavil se je v sedemdesetih letih. Po njem si faze razvoja sledijo popolnoma zaporedno in se ne prekrivajo. Vračanje nazaj ni mogoče. Naslednja faza se začne, ko se predhodna konča. Primeren je za relativno kompleksne projekte. V primeru da problematiko dobro razumemo in da se zahteve tekom razvoja ne bodo bistveno spreminjale. Zaporedni model je v literaturi večkrat označen z imenom "Slapovni model" (ang. Waterfall model), saj razvoj poteka iz ene faze v drugo, podobno kot se voda pretaka iz višjega na nižji nivo.



Slika 1: Zaporedni oziroma slapovni model

Prednosti modela:

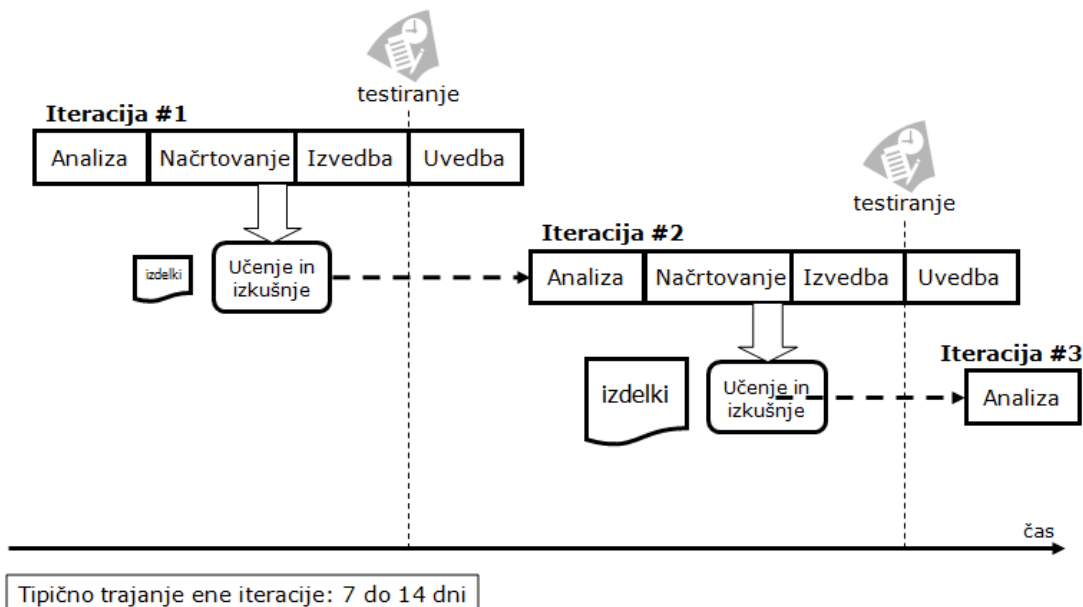
- natančno definira aktivnosti posameznih faz in izdelke ki so potrebni v vsaki fazi
- definira hrbtenico za razvoj kompleksnih aplikacij

Slabosti modela:

- ni fleksibilen. Vsaka sprememba zahteva veliko napora
- ne odraža resničnega razvojnega procesa (saj skoraj nikoli ne velja, da najprej popolnoma zaključimo eno fazo in nato naslednjo)
- ne omogoča paralelnega izvajanja faz

2.2. Iterativni model

Razvit zaradi pomankljivosti slapovnega pristopa. Vse faze izvajamo v večih korakih oziroma iteracijah. Iteracija je specifično zaporedje aktivnosti v okviru nekega načrta. Danes je poznanih več iterativnih modelov. Njihova glavna značilnost je postopen razvoj. Kar pomeni da faze ne zaključujemo v celoti, ampak delno, cel cikel pa ponavljamo. Glavna prednost je da model dovoljuje naraven razvojnega procesa, ki večkrat zahteva da se vrnemo v predhodne faze. S pojavom iterativnega modela so se pojavili tudi protipi ki igrajo v procesu veliko vlogo.



Slika 2: iterativni model

Lastnosti iteracij:

- trajanje od 7 do 14 dni
- vsaka iteracija gre skozi vse faze

- naslednja iteracija se lahko začne šele takrat ko se prejsnja konča

Prednosti modela:

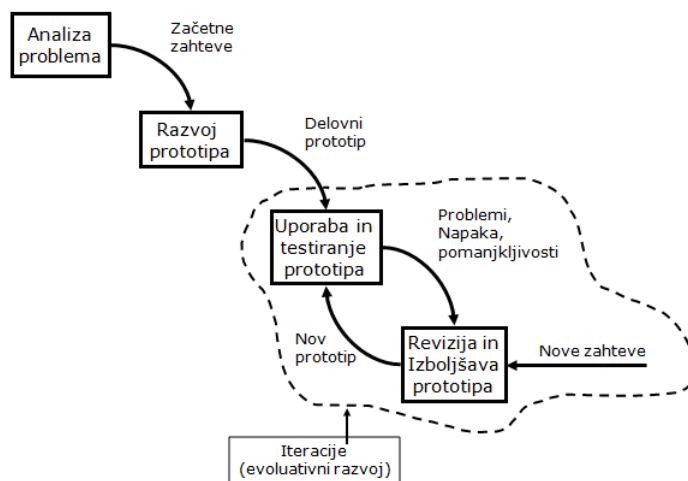
- možnost postopnega razvijanja v iteracijah
- najbolj tvegani deli razrešeni še preden postane investicija velika
- začetne iteracije omogočijo zgodnje povratne informacije s strani uporabnikov
- napredek merimo z ocenjevanjem izvedenega dela

Slabosti modela:

- ne omogoča dobrega načrtovanja poteka procesa
- vodenje projekta je zahtevno
- ni mogoče točno predvideti koliko bo iteracij

2.3. Prototipni model

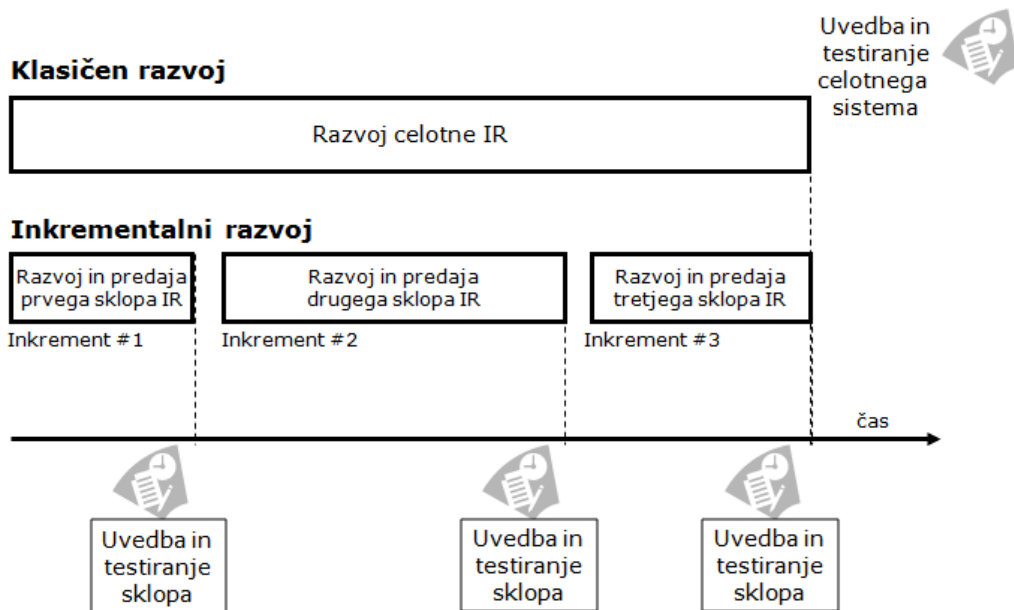
Je različica iterativnega modela. Prototipi se danes uporabljajo v več ali manj vseh razvojnih modelih. Model temelji na izdelavi prototipov, s katerimi označujemo predhodno izdelane in ponavadi nepopolne verzije sistema. Uporaba prototipov omogoča dobro komunikacijo z uporabnikom, zato jih uporabljajo v različnih fazah razvoja. Za izdelavo prototipov poznamo več vrst razvojnih okolij, ki omogočijo vizualno sestavljanje zaslonskih mask. Izdelava prototipov kot tehnika razvoja aplikacij temelji na tesnem sodelovanju med uporabnikom in analitikom. Analitik pripravi prvo verzijo prototipa, ki jo da uporabniku v uporabo oziroma testiranje. Nato uporabnik sproči analitiku kaj mu je všeč in kaj ne. Tako dobi analitik povratne informacije za izdelavo nove verzije prototipa.



Slika 3: Prototipni model

2.4. Inkrementalni model

Je med popularnejšimi modeli, pri katerem razdelimo celoten problem na podprobleme ali module, ki jih rešujemo posebej in neodvisno od ostalih. Aplikacijo tako razvijemo po delih, ki zajamejo le določeno funkcionalnost sistema. Vsak modul posebej do konca razvijemo, ga damo uporabniku nato začnemo z naslednjim. Ob predaji nov sklop povežemo z ostalimi sklopi. Inkremente oziroma module je možno razvijati tudi vzporedno. Rezultat razvoja po inkrementalnem modelu je IR sestavljena iz integriranih sklopov.



Slika 4: Inkrementalni model

Delitev projekta na podprojekte je velikokrat izvedena na osnovi določitve pomembnosti posameznih funkcionalnih zahtev, njihove uvrstitve v module ter določitve vrstnega reda izvedbe posameznih modulov. Priporočeno je določiti ustrezen raspored razvoja sklopov. Zaporedje določimo na različne načine:

- na podlagi pomembnosti sklopov
- na podlagi upoštevanja tveganja

Pri razporejanju sklopov, na osnovi pomembnosti, najprej razporedimo sklope z najpomembnejšo funkcionalnostjo. Na tak način funkcionalnost postopno nadgrajujemo do končnega sistema. Pri razporejanju sklopov glede na pomembnost lahko uporabimo analizo MOSCOW, kjer funkcionalnosti IR razdelimo na:

- funkcije, ki jih je nujno potrebno implementirati

- funkcije, ki jih bi bilo dobro implementirati
- funkcije, ki bi jih lahko implementirali
- funkcije, ki jih ne bomo implementirali

Prednosti modela:

- dele končnega produkata lahko že relativno zgodaj predamo v uporabo, kar poveča možnost za odkrivanje in odpravljanje morebitnih napak in pomankljivosti.
- je cenejši in manj tvegan
- naročnik lažje sledi napredovanju projekta

Slabosti:

- ni ga mogoče uporabiti pri vseh projektih
- IR moramo razdeliti po sklopih in predvideti odvisnosti med njimi
- tveganje, da bomo sklope nepravilno razdelili

2.5. Uporaba življenskih modelov

Za razvoj moje aplikacije sem uporabil inkrementalni model. Celotno problematiko IS sem razdelil na inkremente oziroma module in vsakega posebej razvil od začetne do kočne faze, neodvisno od ostalih. Celotno IR sem tako razvijal po delih, ki sem jih na koncu povezal med seboj. Različne module sem razvijal tudi vzporedno. Vrstni red sklopov sem razdelil po pomembnosti. Najprej sem se osredotočil na najbolj pomembne in zahtevne.

Znotraj vsakega inkrementa oziroma modula sem se nagibal k iterativnemu oziroma prototipnemu modelu. To je bilo skoraj neizogibno saj pri tem projektu gre za zahteve, ki se relativno veliko spreminjajo, niso točno dolečene in posameznih faz ni mogoče v celoti zaključiti in posledično je potrebno faze večkrat ponoviti. Veliko je bilo prisotnega prototipnega modeliranja. Izkazal se je kot eden izmed koristnejših. Na podlagi prototipov oziroma zaslonskih mask sem dobil veliko povratnih informacij s strani uporabnikov in tako sem bil bolj usmerjen v pravi razvoj in na ta način sem dobil tudi boljše in jasnejše informacije. Zaradi razvoja po delih, sem lahko nekatere module predčasno testiral in tako odkril napake in druge morebitne probleme, ki nastopijo. Edina težava je bila v tem, da nisem mogel predvideti koliko iteracij bo potrebnih za vsak modul in na koliko modulov naj razdelim celotno problematiko. Na koncu sem module povezal med seboj in ščasoma je IR postala celota.

3. Metodologije

3.1. Kaj je metodologija

Metodologijo opredelimo lahko kot skupek metod, postopkov in standardov, ki sestavljajo zaključeno celoto pri izvajanju inženirskih pristopov k razvoju produkta. V okviru razvoja informacijskega sistema pa gre za postopen način razvoja informacijskega sistema. Razvoj IS vključuje uporabo različnih tehnik in orodij.

Metodologija zajema vse, kar počnemo, da bi dosegli želen rezultat. Gre za:

- postopke, ki so neporedno povezani z razvojem
- podporne postopke
- način komunikacije med sodelujočimi
- pravila odločanja

Vsaka metodologija obsega tako formalne kot neformalne elemente. Formalni elementi so: postopki, smernice, napotki, standardi, orodja. Med neformalne štejemo; percepcija metodologije, izkušnje, načela, ideali in znanje, ki ga člani organizacije uporabljajo pri svojem delu.

3.2. Vrste metodologij

Obstaja več načinov za delitev metodologij. Delitev metodologij nam pomaga pri izbiri ustrezne metodologije oziroma obstoječo prilagodimo značilnostim projekta. Možne delitve metodologij glede na:

- tip metodologije (procesno, podatkovno procesno in objektno usmerjene)
- teža metodologije (teža je določena z obsegom in gostoto. Obseg metodologije je določen s številom različnih elementov, ki jih metodologija opisuje. Gostoto lahko definiramo kot zahtevan nivo podrobnosti oziroma formalnosti.)
- glede na utežitev metodologije (spredaj utežene ki dajejo povdarek načrtovanju in analizi, zadaj utežene, ki dajajo povdarek na kodiranju in testiranju in uravnotežene, kombinacija obeh.)

3.3. Metodologija v praktičnem primeru

Pri razvoju mojega praktičnega primera je bilo največ vpliva objektivno usmerjene metodologije. Posledica tega je uporaba inkrementalnega in iterativnega modela, kjer imajo glavno vlogo objekti, ki obsegajo tako procese kot podatke.

Z opredelitvijo po teži je bila v ospredju predvsem lahka metodologija, saj so bili udeleženi v procesu z relativno veliko izkušenj in znanja, za kar je značilna visoka prilagodljivost. Ni veliko poudarka na sami dokumentaciji in formalnosti, kar pomeni nizko optimiziranost. Glavni cilj je bil dokončen razvoj programske opreme.

Glede na utežitev metodologije, je bil pri razvoju poudarek na zadaj uteženi metodologiji saj je bilo največ poudarka v fazah kodiranja, testiranja in preizkuševanja. Polek tega je projekt opredeljen z malo skupino razvijalcev, s skupino, ki dobro pozna problematiko, vendar sistem s slabo določenimi zahtevami, ki se relativno spreminjajo in niso točno določene.

Za lažjo izbiro metodologije si lahko pomagamo na različne načine. Eden način je izbira glede na podprte življenske cikle.

Ocena	Kriteriji	Življenjski cikel	Slapovni razvoj	Modificiran slapovni razvoj	Iterativni razvoj	Inkrementalni razvoj	Evolucijsko prototipiranje	Kodiraj in popravi
4 -->	Deluje s slabo razumljenimi zahtevami		Slabo	Srednje do odlično	Odlično	Slabo do srednje	Srednje do odlično	Slabo
3 -->	Deluje s slabo razumljeno arhitekturo		Slabo	Srednje do odlično	Odlično	Slabo	Slabo	Slabo
4 -->	Rezultat je zanesljiv sistem		Odlično	Odlično	Odlično	Srednje do odlično	Srednje do odlično	Slabo
4 -->	Rezultat je nadgradljiv sistem		Odlično	Odlično	Odlično	Odlično	Odlično	Slabo do srednje
3 -->	Upravlja s tveganji		Slabo	Srednje	Odlično	Srednje do odlično	Srednje	Slabo
2 -->	Omogoča predvidljiv razvoj		Srednje	Srednje	Srednje	Odlično	Srednje	Slabo
3 -->	Potrebuje malo režije		Srednje	Odlično	Srednje	Srednje	Srednje	Odlično
4 -->	Dovoljuje spremembe med projektom		Slabo	Srednje	Srednje	Slabo do srednje	Srednje do odlično	Slabo do odlično
5 -->	Stranki omogoča nadzor napredovanja		Slabo	Srednje	Odlično	Srednje	Odlično	Srednje
4 -->	Vodstvu omogoča nadzor napredovanja		Srednje	Srednje do odlično	Odlično	Odlično	Odlično	Slabo
4 -->	Ne zahteva veliko izkušenj		Srednje	Slabo do srednje	Slabo	Slabo do srednje	Srednje	Odlično

Slika 5: Prikaz življenskih modelov z kriteriji

Na levi strani slike sem podal ocene oziroma uteži (od 1 do 5, kjer 1 pomeni najmanj pomembno) pomembnosti posameznim značilnostim mojega projekta pri razvoju. Na ta

način sem pridobil nekaj informacij, ki kažejo izbiro primernega življenjskega modela in s tem tudi metodologije pri razvoju praktičnega primera. Kot je razvidno iz ocen se izkaže iterativni in inkrementalni razvoj kot najbolj primeren pri razvoju, občasno pa tudi kodiraj in popravi model ter tudi modificiran slapovni model.

4. Faze razvoja IS

4.1. Zajem in specifikacija zahtev

Pri zajemu zahtev sem uporabil različne tehnike kot najbolj pogoste; razgovori in sestanki z udeleženi v procesu, opazovanje pri delu in z analiziranjem obstoječega dela. Na ta način sem pridobil bolj formalne in jedrate zahteve za opredelitev razvoja IR. Znotraj specifikacij zahtev sem opredelil kratek opis samega namena IR. Tako sem določil nefunkcionalne kot funkcionalne zahteve, na koncu pa tudi opredelil seznam modulov.

4.1.1. Kratek opis namena IR

Cankarjev dom želi informacijsko rešitev za vpis vseh dogodkov oziroma kulturnih in kongresnih projektov s strani marketinga, ki naj bi potekali v cankarjevem domu. Vsak projekt je določen z več lastnostmi, v glavnem pa projekt sestavlja množica urnikov. V nadaljevanju je vsak urnik prostorsko in časovno opredeljen. Začetni namen urnikov pa je predčasna rezervacija prostorov s strani programskega koordinatorja. Torej izdelati je potrebno aplikacijo ki bi programskemu koordinatorju omogočala dober vpogled že vnesenih urnikov rezervacij in samih urnikov nasploh in posledično tudi lažji vnos novih urnikov rezervacij.

Program naj bi omogočal vodjem projektov da podrobno definirajo urnike projekta in da jih v nadaljevanju programski koordinator tudi potrdi. Velik pomen zajema tudi dispozicija v kateri vodja projekta zapiše tudi seznam opreme, ki je potrebna na licu mesta, oziroma je časovno in prostorsko opredeljena, oziroma je opredeljena kar z urniki projekta.

Ko je dispozicija oddana, naj bi program omogočal vodjem izvedbe, da pregledajo dispozicijo in zagotovijo oziroma potrdijo ali zavržejo zahtevek za potrebno opremo, ki je opredeljena v dispoziciji. Na koncu, ko je dispozicija zaključena, lahko vodje izvedbe razporedijo zaposlene in določijo urnike zaposlenih. Za določanje urnikov zaposlenih so osnova urniki projekta. Izdelati bi bilo potrebno aplikacijo, ki omogoča dobro, hitro, sistematično in pregledno vnašanje urnikov izvedbenih delavcev in ki omogoča dober vpogled že vnešenih urnikov.

4.1.2. Opis problemske domene oziroma funkcionalnih zahtev

I.)

Na začetku marketing vpiše ključne oziroma primarne podatke o projektu (naziv, tuj naziv, status, vsebina, opombe, kraj prireditve, začetek prireditve, konec prireditve, št.držav, št.udeležencev). Ob teh podatkih morajo biti tudi podatki: kdo je vnesel projekt in kdaj je bil vnešen. Status je opredeljen z več vrednostmi: rezervacija, potrditev in storno (ob vnosu rezervacije je status označen kot rezervacija). Marketing naj ima tudi možnost vpisa rezevacij dvoran, zgolj datumsko in prostorsko (kasneje jih potrdi programski koordinator).

Za vsak projekt je določen tudi organizator (naročnik projekta). Naročnik je lahko tuj ali iz slovenije. Za vsak projekt je določen najmanj eden naročnik in največ dva. Lahko sta oba tuja ali oba iz iste države. Vsak naročnik je opredeljen z več podatki (naziv, naslov, naslov1, država, telefon, fax, mobitel, e-mail, opomba, tuj organizator). Polek tega naj bo omogočeno dodajanje novih orgnanzatorjev in spreminjanje podatkov obstoječih. Pomemben je tudi podatek kdo je vnesel v seznam novega organizatorja in datum novega zapisa. Torej ima marketing možnost urejanje šifrantov za organizatorje. Ob vnosu vsakega projekta oziroma rezervacije naj bo omogočen tudi opis tehničnih specifikacij naročnika, prav tako pa naj bo omogočeno zavesti ali bo prisotna pogostitev ob izvedbi projekta.

Projet določajo tudi trije kriteriji s točnim fiksnim naborom vrednosti. Projekt lahko v vsakem kriteriju zavzame le eno vrednost. Pomen kriterijev je tudi za izdelavo kode projekta.

Prvi kriterij (demografsko opredeljen):

- Mednarodna prireditve v CD
- Slovenske prireditve z mednarodno udeležbo v CD
- Slovenske prireditve v CD
- Mednarodne prireditve izven CD
- Slovenske prireditve z mednarodno udel. izven CD
- Slovenske prireditve izven CD

Drugi kriterij(tipsko opredeljen):

- Znanstvene in strokovne prireditve
- Samostojne družabne prireditve
- Prireditve gospodarsko-podjetniškega značaja
- Sejmi, komercialne razstave in festivali
- Prireditve medvladnih in političnih organizacij
- Sprejemi, kosila, zabave, ...

Tretji kriterij(drugo):

- PCO - Celotna organizacija
- Simultana aparatura in tolmačenje
- TIZ
- Posamezne kongresne storitve
- Posamezne tehnične storitve
- Posamezne finančne storitve

V primeru kulturnega projekta so navedeni naslednji kriteriji:

Prvi kriterij:

- Domača prireditve
- Tuja prireditve

Drugi kriterij:

- Glasbene prireditve
- Plesne prireditve
- Gledališke prireditve
- Film
- Humanistična vzgoja, predavanja, seminarji
- Prireditve drugih javnih zavodov
- Sejmi, razstave

II.)

Za vnos vseh ostalih podatkov, ko je projekt potrjen, bo skrbel projektni koordinator. Le-ta bo imel možnost vnosa vseh podatkov kot jih lahko marketing. Lahko bo vpisal še: kodo projekta, vodjo projekta, sezono, tip prireditve, status in barvo projekta. Sezona je opredeljena z enim letom. Sezona se začne s 1.septembrom in konca s 30.avgustom. Vsak projekt določa točno ena sezona. Isto velja za status projekta. Vsak projekt ima lahko v določenem trenutku le en status. Na splošno pa bo imel možnost spreminjanja vseh podatkov o projektu.

Glavno delo programskega koordinatorja pa bo vnašanje urnikov o projektih (rezervacije). Ko je projekt potrjen, je naslednja faza vnos urnikov rezervacij projekta. To so urniki, ki so opredeljeni zgolj datumsko in prostorsko in imajo status rezervacije. V primeru da urnike rezervacij vnese že marketing, dobi programski koordinato zahtevke o vnešenih rezervacijah in jih le-ta potrdi ali zavrže. Vsak projekt ima več urnikov, najmanj enega ali več. Kot celota ima urnik naslenje podatke: Začetek, konec, lokacija, aktivnost, dežurni, opomba, storno). Vsak urnik ima točno eno določeno aktivnost (kratica aktivnosti, naziv aktivnosti, opomba) in točno eno lokacijo (naziv, kratica, opomba, zap. št, domača lokacija). Zaporedna številka je

zaporedna številka nekega prostora, ki jo določi programski koordinator. Domača lokacija pa pove, ali je lokacija znotraj cankarjevega doma ali je zunanja, najeta lokacija. Programski koordinator ima tako možnost vnosa tudi novih aktivnosti in lokacij.

Želijo tudi hraniti podatke o osebah ki so zaposlene v cankarjevem domu (priimek, ime, username, e-mail, telefon, gsm, mpo, matična št.). Vsak zaposlen spada v en sektor, znotraj tega v oddelek in znotraj tega v pododdelek. Programski koordinator bo imel možnost vnosa novih oseb in spreminjanje obstoječih. Vnašal bo lahko tudi nove sektorje, oddelke in pododdelke. Poleg tega bo imel možnost vnašanja pravic dostopov oseb do določenih aplikacij.

III.)

Pomembno vlogo bo imela tudi dispozicija. Dispozicijo izdelava vodja projekta, ko so urniki projekta razčlenjeni. Je dokument ki je opredeljen z urniki projekta, v glavnem pa so v dispoziciji zabaležene zahteve tehnične opreme. Vsak projekt ima v osnovi eno dispozicijo lahko pa tudi več. Dispozicijo opredeljuje: Identifikacijska številka, projekt, datum vnosa dispozicije, opomba in ali je potrjena. Vsako dispozicijo sestavlja množico urnikov ki so razporejeni po lokacijah. Za vsak urnik ali množico urnikov je pa določena zahteva po tehničnih oddelkih (oddelek za luč, oddelek za ton, oddelek za video, oddelek za ITK, ...).

IV.)

Glavno vlogo bodo imeli tudi vodje posamezih pododdelkov v oddelku »tehnična izvedba programa«. Gre za oddelke kot so:

- oddelek za multimedijske tehnologije
- oddelek za odrsko-scenske postavitve
- oddelek za ozvočenje
- oddelek za postavitve prizorišč in splošno vzdrževanje
- oddelek za potek prireditve
- oddelek za scensko osvetljava
- oddelek za ITK, itd.

Naloga vodij teh oddelkov je razporejanje izvedbenih delavcev za opravljanje določenih aktivnosti v okviru nekaterih projektov. Osnova za to so urniki projektov, kot je omenjeno že zgoraj. Vsak urnik prireditve ima lahko nič ali več urnikov oseb oziroma sodelujočih. Sodelujoči so osebe oziroma izvedbeni delavci, ki so na licu mesta in opravljajo neko funkcijo glede na urnik prireditve. Vsak urnik osebe torej pripada urniku prireditve. Opredeljen je z: osebo, vlogo, začetkom, koncem, opombo. Vloga določa kakšno funkcijo bo imela oseba v okviru nekega urnika na samem mestu. Vsakemu urniku osebe pripada ena vloga. Dopolnjevanje in spreminjanje vlog bo delo programskega koordinatorja (naziv, opomba).

V Cankarjevem domu torej želijo imeti aplikacijo za vnos projektov in rezervacij, kasneje v primeru da je projekt potrjen aplikacijo za dober vpogled, nadzor in za samo dobro koordinacijo nad samimi urniki prireditev, saj le ti so osnova za nadaljno razporejanje in določevanje urnikov zaposlenih. Želijo aplikacijo za vodje projektov, za izdelavo dipozicij, za pregled nad obstoječimi, katere bi potrjevali in urejali tudi vodje izvedbenih oddelkov. Potrebujejo aplikacijo, prek katere se bi vnašalo in razporejalo izvedbene delavce, ob tem pa bi izhajali iz urnikov prireditev. Potrebujejo tudi sam izpis oziroma tiskanje urnikov zaposlenih za določen teden po oddelkih, poleg tega pa še aplikacijo namenjeno programskemu koordinatorju za urejanje vseh podatkov celotnega projekta.

4.1.3. Opis nefunkcionalnih zahtev

Celotna IR bo v glavnem sestavljena iz več web aplikacij oziroma modulov. Celotna rešitev bo temeljila na microsoftovi tehnologiji .NET. IR bo postavljena na spletnem strežniku znotraj cankarjevega doma. Ker bodo uporabniki do aplikacij dostopali preko portala Cankrarjevega doma, ne bo potrebne avtentikacije in avtorizacije, saj je le-ta zahtevana že preden se uporabnik prijavi na portal. Za dostop do aplikacij potrebuje uporabnik lokalni računalnik, domensko uporabniško ime in geslo ter brskalnik Microsoft Internet explorer (katerakoli verzija). Uporabnik lahko uporabi tudi druge brskalnike, vendar bodo aplikacije najboljše optimizirane za Internet explorer. Za razliko od ostalih bo imel programski koordinator nameščeno windows aplikacijo lokalno na svojem lokalnem računalniku. Podatkovna baza bo postavljena v tehnologiji Microsoft MySQL.

4.1.4. Opis vmesnikov

1.Vmesnik (Rezervacije)

Aplikacija »Rezervacije« bo spletna aplikacija namenjena oddelku za marketing. Namen te bo vpis projektov, vpis okvirnih začetnih podatkov in zahtev naročnika. Možnost bo tudi vpis predloge rezervacij dvoran. Marketing bo imel možnost urejanja vnešenih podatkov o projektu z omejitvami, poleg tega bo urejal tudi podatke o organizatorjih oziroma naročnikih

2.Vmesnik (Projekti)

»Projekti« bo windows aplikacija namenjena programskemu koordinatorju. Aplikacija bo omogočala pregled vseh projektov, potrjevanje projektov, urejanje le teh, vnos urnikov rezervacij ali potrjevanje le teh (v primeru da jih vnese že marketing), možnost vpisa vodje projekta, določanje kode projekta itd. Programski koordinator bo preko aplikacije potrjeval

tudi razčlenjene urnike s strani vodje projekta. Imel bo možnost dodajanja vseh šifrantov in tako tudi urejanje teh, prav tako bo vodil evidenco vseh oseb in zaposlenih in določeval pravice dostopov do aplikacij.

3.Vmesnik (Dispozicije)

Aplikacijo »Dispozicije« bodo uporabljali vodje projektov. Namen te aplikacije je kreiranje nove dispozicije za projekt. Pogoji za izdelavo dispozicije so potrjeni razčlenjeni urniki s strani vodje projektov zato bo aplikacija omogočala razčlenjevanje urnikov projekta, nato bo lahko vodja projekta izdelal dispozicijo. V dispoziciji bo navedel seznam tehnične specifikacije, možnost pripenjanja dokumentov in objava dispozicije oziroma oddaje dispozicije v predogled vodjem tehnično izvedbenih oddelkov.

4.Vmesnik (Urniki)

Aplikacija »Urniki« bo spletna aplikacija namenjena vodjem tehničnih izvedbenih oddelkov. Aplikacija bo prikazovala seznam vseh urnikov projektov, saj so le ta osnova za izdelavo urnikov izvedbenih delavcev. Aplikacija bo omogočala vnos novih urnikov izvedbenih delavcev in urejanje obstoječih urnikov z možnostjo tiskanja.

5.Vmesnik (Oprema) – opsijsko

Aplikacija »Oprema« bo spletna aplikacija namenjena vodjem tehničnih izvedbenih oddelkov. Gre zgolj za vpis in vodenje tehnične opreme po oddelkih. Za rezervacijo opreme, ki nastopa na projektih oziroma konkretno na dispozicijah.

4.2. Analiza

Glavni namen analize je izdelati razumljiv opis realnega sveta oziroma poslovnega okolja, na katerega se nanaša razvoj IS. Izdelamo model sistema, ki na formalen način opredeli potrebne podatkovne strukture in funkcije, ki te podatke uporabljajo.

Analiza služi kot:

- sredstvo za definicijo zahtev
- osnova za dogovor med naročnikom in izvajalcem
- osnova za kasnejše faze razvoja

Rezultat analize je model sistema. Na osnovi modelov se v nadaljnjih korakih izdelata podatkovna baza ter potrebni programski moduli. Poleg modela sistema v fazi analize izdelamo tudi predlog tehnične arhitekture sistema ter opsijsko prototipe komponente uporabniškega vmesnika.

Med izdelke analize sodi tudi strategija testiranja. Postopek analize izvajajo sistemski analitik, sistemski arhitekt in ključni uporabniki.

Postopek analize tipično zajema naslednje aktivnosti:

- izdelava modela sistema
- izdelava prototipov
- izdelava predloga tehnične arhitekture sistema
- opredelitev strategije testiranja
- predstavitev rezultatov analize naročniku

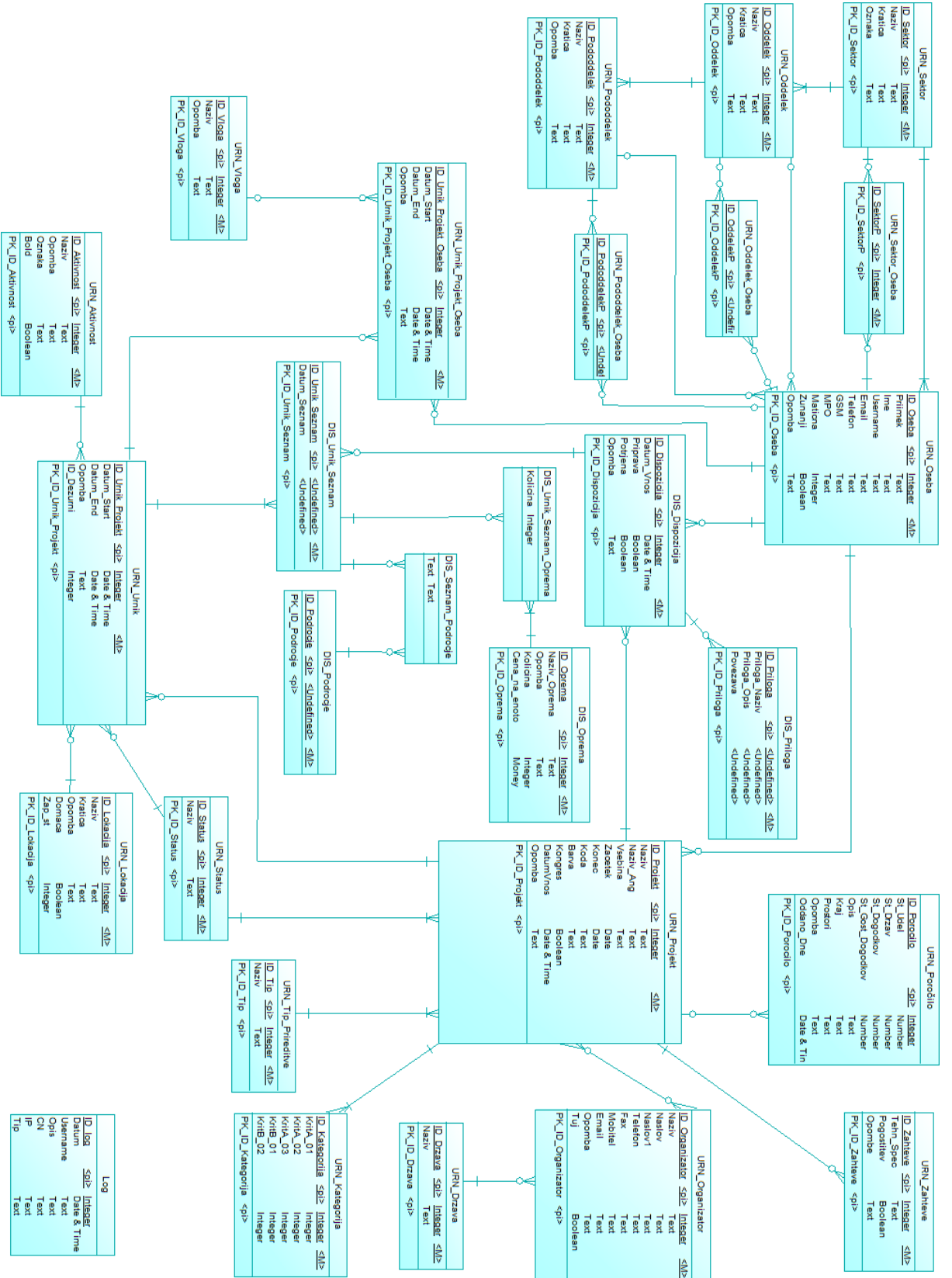
4.2.1. Izdelava modela sistema

Modeliranje je ključnega pomena, njegov namen je, da sistem bolje razumemo. Poudarimo pomen ključnih elementov in zanemarimo nepomembne. Modeliranje omogoča vizualizacijo sistema, prikaz statičnih in dinamičnih lastnosti in dokumentira sprejete odločitve, poleg tega predstavlja osnovo za nadaljno gradnjo sistema.

Izbiramo lahko med več modeli, najboljši so tisti, ki so dobro povezani z realnostjo. Izbira modela nekako določa kako bomo pristopili k problemu. Ponavadi izdelamo več modelov, saj eden sam ni ravno dovolj.

4.2.1.1. Podatkovni model - konceptualni model

prikazuje sistem s podatkovnega vidika tako, da opisuje podatkovne strukture, ki so potrebne za delovanje sistema. Poleg podatkovnih struktur zajema tudi vse povezave med njimi.



Slika 6: Prikaz podatkovnega modela - konceptualni diagram

Za tem nastopi vodja projekta, ki pregleda okvirne urnike. Na podlagi teh oblikuje podrobne urnike. Naslednjo nalogo ima spet programski koordinator, ki pogleda razčlenjene urnike vodje projekta in jih lahko: zavrže (v tem primeru mora vodja projekta urnike nekoliko spremeniti) ali potrdi in proces gre v naslednjo fazo. Ko so urniki projekta razčlenjeni in potrjeni, vodja projekta kreira dispozicijo. V njej določi seznam opreme, ki bo prisotna za določen urnik. Ko dokončno določi opremo, dispozicijo objavi.

Vodja izvedbenega oddelka pregleda izdano dispozicijo oziroma tisti del dispozicije, ki se nanaša nanj. Odobri ali zavrže seznam opreme, ki je navedena v dispoziciji. V primeru, da dispozicijo zavrže, mora vodja projekta seznam opreme spremeniti. V primeru potrditve vodja izvedbenega oddelka pripravi urnik izvedbe za izvedbene delavce. Kasneje, ko je dokončan, le-tega tudi tiska.

4.2.1.3. Model procesne logike – strukturiran jezik

Njegova značilnost je, da natančneje definira procese, definirane v procesnem modelu. Gre za kratke in jedrnate stavke sestavljene iz glagolskih in samostalniških oblik naravnega jezika.

Vnos projekta (Marketing):

//najprej preveri ali je projekt že vnešen
 vnos naziva projekta
 vnos opisa projekta
 vnos datuma začetka in konca projekta
 vnos morebitnih rezervacij prostorskih virov
 //izpiše se seznam naročnikov (organizatorjev)
 določi organizatorja (največ dva in najmanj eden)
 morebiti popravi kontaktne podatke organizatorja
 določi prisotnost pogostitve
 določi tehnično specifikacijo
 //izpisejo se kategorije projektov
 določi kategorijo projekta
 potrdi ali prekliči vnos projekta

Vnos ostalih informacij (referent)

//izpis vseh projektov
 izberi projekt
 //izpis vseh informacij o projektu
 vnos ostalih informacij

//izpis seznama vodij projektov
določi vodjo projekta
vnos urnikov projekta (rezervacij)
//prikaži seznam morebitnih že vnesenih urnikov
potrdi vnos

razčlenitev urnikov projekta (vodja projekta)

izberi projekt
//izpiše se seznam projektov
izberi urnik
//izpiše se seznam urnikov
določi uro in aktivnost
//izpiše se seznam aktivnosti
potrdi spremembo urnika

Kreiranje dispozicije (vodja projekta)

izberi projekt
//izpiše se seznam projektov
izberi dispozicijo
//izpiše se seznam dispozicij
izberi urnike
//izpiše se seznam urnikov
določi opremo za množico urnikov
//izpiše se seznam opreme
objavi dispozicijo
//dispozicija je objavljena

Preveri dispozicijo (vodja izvedbenega oddelka)

izberi projekt
//izpis seznama projektov
izberi dispozicijo
//izpis zahtev tehnične specifikacije dispozicije
potrdi in rezerviraj seznam opreme
//oprema se rezervira

Priprava urnikov izvedbe(Vodja izvedbenega oddelka)

izberi urnike
 //izpiše se seznam urnikov projekta
 izberi osebe
 //izpiše se seznam oseb
 določi vlogo osebe
 //izpiše se seznam vlog
 določi uro
 vnesi urnike oseb za urnike projektov

4.2.1.4. UML diagram –Diagram primerov uporabe

Primeri uporabe so ključni element procesa RUP. Z diagramom primerov uporabe prikažemo zakaj se sistem uporablja, kakšna je povezava med primeri uporabe in akterji. Diagram prikazuje zaporedje akcij, ki jih izvede sistem in dajo določenemu akterju nek rezultat. Akter pa lahko predstavlja osebo, napravo ali stvar izven sistema. Primer uporabe prikazuje pomembnejši način uporabe sistema za enega ali več akterjev, ki vplivajo na ta primer uporabe. Primer uporabe poimenujemo z vidika uporabnika.

Z jezikom UML smo zgradili diagram primerov uporabe. Akterji ki nastopajo v naši aplikaciji so:

- Marketing
- Programski koordinator
- Vodja projekta
- Vodja izvedbenih oddelkov

Primeri uporabe za Marketing

Primeri uporabe za oddelek za Marketing so naslednji:

- Prvi primer uporabe je **vnos projektov**, kjer marketing vnese projekt. Prej mora marketing preveriti ali ni morebiti projekt že vnesen. Vnese okvirne oziroma primarne podatke, nato vnese tudi naročnika, ki je soorganizator projekta in tudi njegove okvirne zahteve. Vnese lahko tudi okvirne rezervacije prostorov.

OSNOVNI TOK:

- 1.vnesi glavne podatke(Naziv, opis, opomba, začetek in konec projekta)
- 2.vnesi urnike rezervacij (opcijsko)
- 3.sistem prikaže seznam organizatorjev
- 4.izberi organizatorja
- 5.sistem prikaže polja za vnos okvirnih zahtev naročnika
- 6.vnos tehničnih in drugih zahtev
- 7.sistem prikaže seznam kriterijev

- 8.vnesi kriterije projekta
- 9.projekt vnešen

ALTERNATIVNI TOK:

3.sistem prikaže seznam organizatorja

3.1.na seznamu ni željenega organizatorja, sistem ponudi možnost vnosa novega organizatorja

- Drugi primer uporabe je **pregled vnešenih projektov**, kjer lahko marketing pregleduje že vnešene projekte in možnost spreminjanja obstoječih z omejitvami . Zadnja faza je **potrditev projekta**. Nato dobi referent obvestilo o potrditvi projekta. Po končanem projektu **odda poročilo** o izvršenem projektu.

OSNOVNI TOK:

1.prikaži seznam vnesenih projektov

2.sistem prikaže seznam projektov

3.izberi projekt

4.sistem prikaže opcijo za potrditev projekta

5.potrdi projekt

6.sistem obvesti programskega koordinatorja o potrditvi projekta

OSNOVNI TOK:

1.prikaži seznam vnesenih projektov

2.sistem prikaže seznam projektov

3.izberi projekt

4.sistem prikaže podatke za vnost poročila

5.vnos podatkov poročila

6.oddaja poročila

ALTERNATIVNI TOK:

5.vnos podatkov poročila

5.1.sistem ugotovi, da projekt še ni zaključen in zato poročila ni možno oddati

- Tretji primer je **pregled in urejanje organizatorjev**. Marketing ima možnost vnosa novih soorganizatorjev oziroma naročnikov in prav tako urejanje vseh informacij povzenih z organizatorji.

OSNOVNI TOK:

1.prikaži seznam organizatorjev

2.sistem prikaže seznam organizatorjev

3.vnesi novega organizatorja

4.sistem prikaže polja za vnos novega organizatorja

5.vnesi podatke organizatorja

- 6.potrdi vnos organizatorja
- 7.organizator vnešen

ALTERNATIVNI TOK:

- 3.vnesi novega organizatorja
- 3.1.organizator je že vnešen, zato sistem organizatorja ne vnese

Primeri uporabe za Programskega koordinatorja

Primeri uporabe za programskega koordinatorja so naslednji:

- prvi primer je **vnos ostalih informacij projekta**. Ob potrditvi projekta, referent vnese ostale informacije. Poleg tega ima možnost spreminjanja vseh ostalih informacij.

OSNOVNI TOK:

- 1.prikaži seznam projektov
- 2.sistem prikaže seznam projektov
- 3.izberi projekt
- 4.sistem prikaže vse podatke projekta
- 5.vnos ostalih podatkov o projektu
- 6.shrani spremembe

ALTERNATIVNI TOK:

- 3.izberi projekt
- 3.1.projekt je storniran in ga ni mogoče urejati

- Drugi primer je **vpis okvirnih urnikov oziroma rezervacije**. Po vnosu ostalih informacij programski koordinator začne z vpisovanjem okvirnih urnikov oziroma rezervacij prostorskih virov, ki so v nadaljevanju tudi kot osnova za oblikovanje podrobnih urnikov projekta.

OSNOVNI TOK:

- 1.prikaži seznam projektov
- 2.sistem prikaze seznam projektov
- 2.izberi projekt
- 3.sistem ponudi polja za vnos urnikov rezervacij
- 4.vnesi urnik
- 5.urnik vnešen

ALTERNATIVNI TOK:

- 4.izberi projekt
- 5.sistem prikaze seznam urnikov rezervacij (vnešeni s strani marketinga)

6.potrdi urnike

7.obvesti marketing o potrditvi rezervacij urnikov

- Tretji primer je **pregled razčlenjenih urnikov**. Ko vodja projekta razčleni oziroma podrobno definira urnike, obvesti programskega koordinatorja in njegova naloga je pregled razčlenjenih urnikov, ki jih lahko potrdi ali zavrže.

OSNOVNI TOK:

- 1.prikaži seznam projektov
- 2.sistem prikaže seznam projektov
- 3.izberi projekte z nepotrjenimi razčlenjenimi urniki
- 4.sistem prikaže seznam vseh projektov s tem kriterijem
- 5.izberi projekt
- 6.sistem prikaže seznam razčlenjenih urnikov
- 7.potrdi ali zavrne razčlenjene urnike
- 8.obvesti vodjo projekta

ALTERNATIVNI TOK:

- 3.izberi projekte z nepotrjenimi razčlenjenimi urniki
- 3.1.ni projektov z razčlenjenimi urniki

- Četrti primer uporabe je **vnos šifrantov**. Naloga programskega bo vnos vseh šifrantov, ki nastopajo v tem procesu. Z možnostjo brisanja, spreminjanja in vnašanja novih.

OSNOVNI TOK:

- 1.izberi šifranta
- 2.sistem ponudi vse šifrante
- 3.vnesi novega šifranta
- 4.sistem ponudi polja za vnos novega šifranta
- 5.vpiši podatke o šifrantu
- 6.shrani šifranta

Primeri uporabe za vodjo projekta

Primeri uporabe za vodjo projekta so naslednji:

- Prvi primer uporabe je **definiranje okvirnih urnikov projekta**, kjer vodja projekta na podlagi okvirnih urnikov(rezervacij) nazorno in podrobneje definira urnike projekta, ki so tudi osnova za dispozicijo.

OSNOVNI TOK:

- 1.prikaži seznam projektov
- 2.sistem prikaže seznam projektov
- 3.izberi projekt
- 4.sistem prikaže seznam okvirnih urnikov (rezervacij)
- 5.podrobno razčleni urnike
- 6.urniki razčlenjeni
- 7.sistem obvesti referenta za vpogled razčlenjenih urnikov

ALTERNATIVNI TOK

- 3.izberi projekt
- 3.1.okvirni urniki še niso definirani

- Drugi primer uporabe je **Izdelava dispozicije**. Ob uspešni potrditvi razčlenjenih urnikov s strani programskega koordinatorja lahko vodja projekta izdela dispozicijo. V dispoziciji definira seznam opreme, ki jo potrebuje za vsak urnik projekta po oddelkih.

OSNOVNI TOK:

- 1.prikaži seznam projektov
- 2.sistem prikaže seznam projektov
- 3.izberi projekt
- 4.sistem prikaže seznam tipov dispozicij
- 5.izberi tip dispozicije
- 6.kreiraj dispozicijo
- 7.dispozicija izdelana
- 8.sistem v dispoziciji prikaže seznam urnikov po dnevih in prostorih, z možnostjo vnosa zahtev po opremi
- 9.navedi seznam opreme za vsako množico urnikov po dvoranah in dnevih
- 10.shrani spremembe dispozicije

ALTERNATIVNI TOK:

- 6.kreiraj dispozicijo
- 6.1.sistem ne kreira dispozicije ker razčlenjeni urniki še niso potrjeni

- Tretji primer uporabe je **objava dispozicije**. Na koncu, ko vodja projekta približno izdela dispozicijo in navede približen seznam opreme, lahko dispozicijo objavi in jo da v pogled vodjem izvedbe.

OSNOVNI TOK:

- 1.prikaži seznam projektov
- 2.sistem prikaže seznam projektov
- 3.izberi projekt
- 4.sistem prikaže dispozicije
- 5.izberi dispozicijo
- 6.objavi dispozicijo

5.sistem pošlje obvestilo vodjem tehnične izvebe

ALTERNATIVNI TOK:

5.izberi dispozicijo

5.1.dispozicije ni mogoče objaviti saj je rok za objavo potekel

Primeri uporabe za vodjo izvedbe

Primeri uporabe za Vodjo izvedbe so naslednji:

- Prvi primer uporabe je **preveri dispozicijo**, kjer vodja izvedbenega oddelka na podlagi oddane dispozicije s strani vodje projekta preveri zahteve po opremi, ki je navedena v dispoziciji in le to potrdi ali ovrže.

OSNOVNI TOK:

1.prikaži seznam projektov

2.sistem prikaže seznam projektov

3.izberi projekt

4.sistem prikaže seznam dispozicij za projekt

5.izberi dispozicijo

6.sistem izpiše dispozicijo

7.potrdi dispozicijo

8.obvestilo vodji projekta o rezervaciji opreme in potrditvi dispozicije

ALTERNATIVNI TOK:

5.izberi dispozicijo

6.sistem izpiše dispozicijo

7.zavrnitev dispozicije

8.obvestilo vodji projekta o zavrnitvi dispozicije

- Drugi primer uporabe je **izdelava urnika izvedbe**. Ko je dispozicija izdelana in dokončno potrjena, vodja izvedbenih oddelkov razporeja izvedbene delavce in izdelava urnik izvedbnih delavcev na podlagi razčlenjenih urnikov projekta.

OSNOVNI TOK:

1.prikaži seznam razčlenjenih urnikov

2.sistem prikaže seznam urnikov

3.izberi razčlenjene urnike

4.prikaži seznam izvedbenih delavcev

5.sistem prikaže seznam izvedbenih delavcev

6.izberi izvedbene delavce

7.prikazi predogled seznam urnikov izvedbenih delavcev

8.vnos urnikov izvedbenih delavcev

9.urniki izvedbenih delavcev vnešeni

ALTERNATIVNI TOK:

4.prikaži seznam izvedbenih delavcev

4.1.nimate pravic za vnos urnikov izvedbenih delavcev

- tretji primer uporabe je **urejanje urnikov izvedbe**, kjer vodje izvedbenih delavcev pregledajo urnike izvedbe in jih morebiti popravijo, spremenijo in tako premeščajo zaposlene oziroma njihove urnike.

OSNOVNI TOK:

1.prikaži seznam izvedbenih urnikov

2.sistem prikaže seznam urnikov izvedbenih delavcev

3.izberi izvedbene delavce

4.prikaži seznam urnikov izbranih delavcev

5.sistem prikaže seznam urnikov

6.uredi urnike

7.vnos spremenjenih urnikov

ALTERNATIVNI TOK:

4.prikaži seznam urnikov izbranih delavcev

4.1.urniki za izbrane delavce niso definirani

- Četrty primer uporabe je **tiskanje urnikov izvedbe**, kjer vodja izvedbenega oddelka tiska urnike za svoje izvedbene delavce.

OSNOVNI TOK:

1.prikaži seznam oddelkov izvedbe

2.sistem prikaže oddelke

3.izberi izvedbeni oddelek

4.sistem prikaže seznam izvedbenih delavcev ki spadajo v ta oddelek

5.prikaži urnike izvedbenih delavcev

6.sistem prikaže urnike

7.formatiranje urnika

8.tiskanje urnika

ALTERNATIVNI TOK:

5.prikaži urnik izvedbenih delavcev

5.1.za izbrane delavce sistem ne prikaže urnikov saj le-ti niso definirani

4.2.2. Izdelava prototipov

Izdelava prototipov zaslonih mask je neobvezna. Dobra pa je za prikaz približnega izgleda uporabniškega vmesnika in njegove funkcionalnosti, še preden je ta razvit. Namen prototipiranja je iskanje informacij, s katerimi zmanjšamo tveganje napačnih odločitev.

Sama izdelava prototipov naj bi bila hitra in enostavna. Pogosto se uporabljajo za izdelavo prototipov druga orodja kot za samo izdelavo pravega uporabniškega vmesnika. Posledica tega je, da je lahko izdelava s pravimi orodji s katerimi izdelamo pravi uporabniški vmesnik preveč kompleksna in časovno potratna. V mojem primeru sem prototipe izdelal z istim orodjem kot samo implementacijo celotnega in končnega uporabniškega vmesnika, vendar sem podrobnosti zanemaril in se posvetil predvsem sami okvirni maski in prikaz glavnih funkcionalnosti. Prototipe lahko gradimo na podlagi primerov uporabe. Za izdelavo prototipov sem najprej naredil skico in šele nato samo izdelavo z programskim orodjem. Prototipe sem opredelil na podlagi akterjev.

Seznam prototipov:

- Prototip št. 1. - Marketing - Rezervacije (vnos projekta, seznam projektov, seznam organizatorjev)
- prototip št.2. - Programski koordinator - Projekti (vnos ostalih informacij, potrjevanje in razčlenjevanje urnikov)
- prototip št.3. - Vodja projekta - Dispozicije (razčlenjevanje urnikov, kreiranje dispozicij, seznam dispozicij)
- prototip št.4. - Vodja izvedbe - Urniki (pregled dispozicij, kreiranje urnikov izvedbe, potrjevanje dispozicij)

4.2.2.1. Prototip št.1- Marketing (Rezervacije)

Prvi del prototipa prikazuje zaslonko masko za vpis novega projekta. Na zaslonki maski so polja za vnos ključnih oziroma primarnih podatkov, kot so Naziv, Naziv ang, opis, opombe, status projekta (na začetku je izbrana vrednost rezervacija), datum začetka in konca projekta.

V nadaljevanju je možen vpis seznam urnikov za rezervacijo prostorov, ki so določeni prostorsko in datumsko. Na koncu obrazca so podatki za vnos organizatorjev, opis tehnične specifikacije in vnos morebitne prisotnosti pogostitve in kriteriji za opis projekta.

Drugi del prototipa prikazuje zaslonko masko za iskanje vnešenih projektov. Za

Slika 9: Prototip št.1 (Rezervacije) - obrazec za vnos rezervacije

lažje iskanje so na vrhu vmesnika filtri s katerimi zožimo nabor zadetkov. Filtiramo lahko po več vrednostih; Datumsko, z vpisom začetnega in končnega datuma, išemo lahko po kodi projekta, po organizatorju in po vodji projekta. Kot izhod se nam izpišejo vsi projekti pod izbranimi kriteriji v obliki seznama. Seznam lahko tudi uredimo po večih parametrih; Datum, Naziv, Koda, Status. Podatke projekta lahko tudi spreminjamo s klikom na le-tega v seznamu.

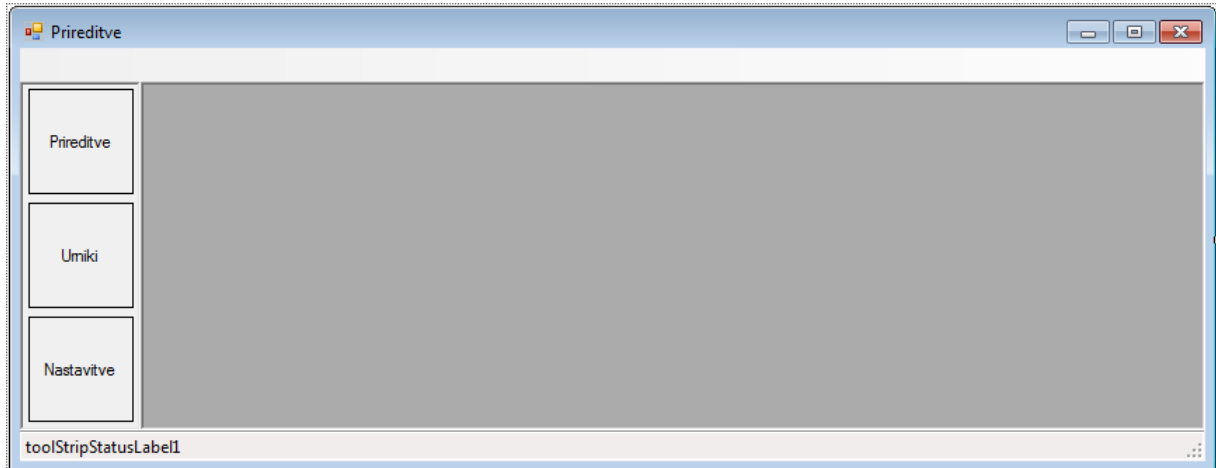
Slika 10: Prototip št.1 (Rezervacije) - seznam rezervacij

Tretji del prototipa je seznam organizatorjev. Namem tega programskega modula je urejanje podatkov organizatorjev, vnos novih in urejanje obstoječih. Vrednosti so prikazane v obliki seznama. Za lažje iskanje si pomagamo z uporabo filtrov; Domači organizator, tuj organizator in izpis vseh z možnostjo sortiranja po različnih parametrih: podjetje, kontakt, naslov in telefon organizatorja. Pri vnosu novega organizatorja so naslednji podatki: naziva, naslov1, naslov2, država, mednarodni/domači, telefon, kontakt, mobitel, faks in elektronski naslov.

Slika 11: Prototip št.1 (Rezervacije) - seznam organizatorjev

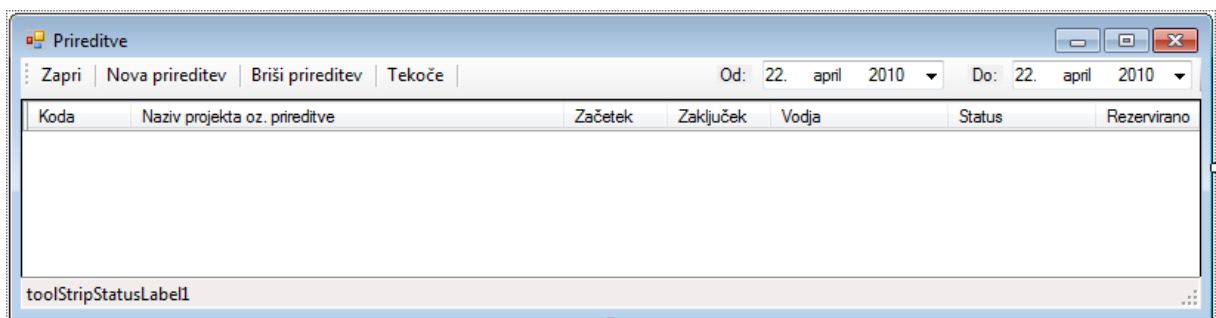
4.2.2.2. Prototip št.2. - Programski koordinator (Projekti)

Glavni del prototipa je glavna forma z možnostjo prikazovanja prireditev, urnikov in urejanje nastavitev. Tukaj gre zgolj za vpisovanje in urejanje podatkov o šifrantih ter določanje in urejanje pravic dostopov uporabnikov.



Slika 12: Prototip št.2 (Projekti) - glavna forma

Naslednji del prototipa so »projekti«, kjer so navedeni vsi projekti v obliki seznama. Izpis seznama lahko omejimo z datumskim filtrom. Seznam lahko poljubno urejamo. Omogočen je tudi izpis tekočih projektov (projekti, ki se dogajajo v trenutnem tednu), dodajanje novih projektov, brisanje projektov itd.

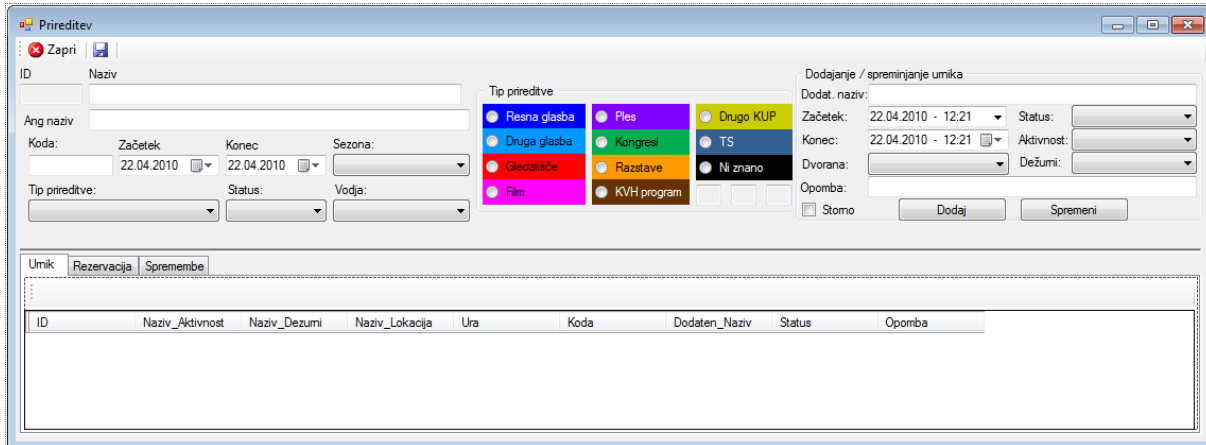


Slika 13: Prototip št.2 (Projekti) - seznam projektov

Z izbiro projekta iz zgornjega seznama se odpre naslednji del zaslonske maske, kjer se izpišejo vse podrobnosti projekta. Na zaslonski maski so polja za; naziv projekta, tuj naziv, koda projekta, začetek in konec projekta, sezona v katero sodi projekt, opis tipa projekta (od katere je tudi odvisna barva projekta), status projekta in vodja projekta.

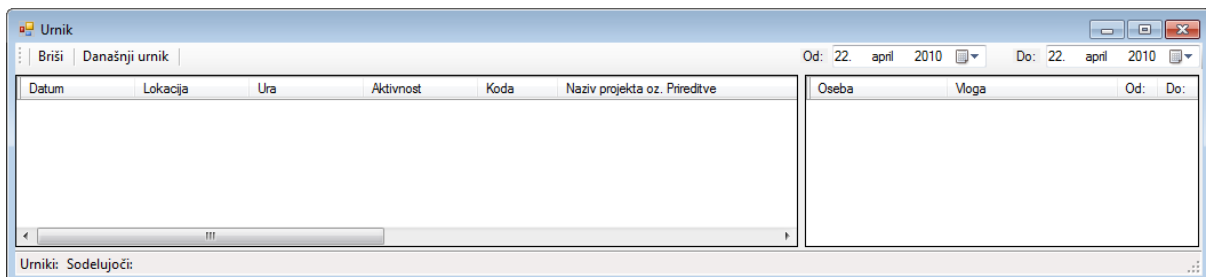
Na zaslonski maski bo na enem izmed zavihkov (Urniki) tudi seznam urnikov za izbran projekt z vsemi potrebnimi podatki. Omogočeno bo tudi spreminjanje urnikov, brisanje ter dodajanje le-teh izbranemu projektu. Za dodajanje novega urnika so vsa potrebna polja:

naziv, začetek in konec urnika, lokacija, status, aktivnost, dežurna oseba ter morebitna opomba. V naslednjem zavihku (Rezervacije) bodo vsi podatki katere je zavedel marketing ob vpisu projekta, ter zahteve naročnika oziroma organizatorja. V zadnjem zavihku (Spremembe) bodo zabeležne spremembe vseh podatkov projektov.



Slika 14: Prototip št.2 (Projekti) - projekt

Naslednji del prototipa so »Urniki«. Prikaz urnikov v obliki seznama z možnostjo filtriranja po datumu. Po privzetem bo seznam vseboval tekoče urnike oziroma urnike tekočega dne. Seznam bo prav tako možno urediti po različnih parametrih. Z izbiro nekega urnika projekta se bo na desni strani prikazal seznam sodelujočih oseb na tem urniku oziroma seznam izvedbenih delavcev z določeno vlogo.



Slika 15: Prototip št.2 (Projekti) - seznam urnikov

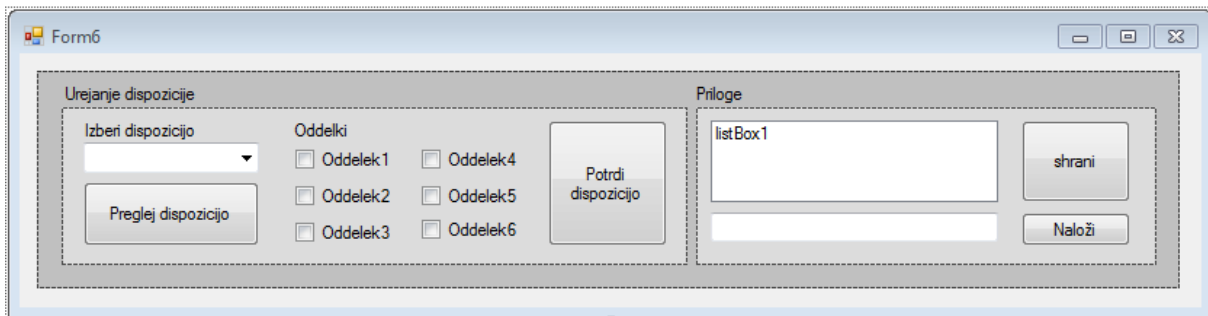
4.2.2.3. Prototip št.3. - Vodja projekta (Dispozicije)

Glavni del prototipa je seznam dispozicij. Ker pa so dispozicije vezane na projekt, bo iskanje dispozicij preko projektov (zato tudi zaslonska maska prikazuje seznam projektov). Iskanje bo mogoče omejiti z filtri na vrhu zaslonske maske: datum, projekt, vodja projekta in koda. Za lažje komuniciranje in sporazumavanje med vodji projektov in vodji izvedbenih oddelkov bo možno oddajanje krakih obvestil.

4.2.2.4. Prototip št.4. - Vodja izvedbe (Urniki, Dispozicije)

Glavni del zaslonske maske za iskanje in potrjevanje dispozicij je isti kot ga imajo vodje projekta (slika 16).

Z izbiro projekta iz seznama (slika 16), se odpre nova zaslonska maska oziroma drugi del prototipa (slika spodaj). Vmesnik omogoča pregled izbrane dispozicije v celoti. Omogočen je prikaz dispozicije z možnostjo izključitve posameznih oddelkov, potrjevanje dispozicije in oddajanje prilog k dispoziciji.



Slika 19: Prototip št.4 (Dispozicije) - zaslonska maska za kreiranje dispozicije s strani vodij izvedbe

Izgled pregleda dispozicije za vodje izvedbe prikazuje naslednja zaslonska maska (slika spodaj). Uporabniku se izpišejo vsi datumi, ki so navedeni za izbrano dispozicijo. Znotraj posameznega datuma so navedene lokacije, ki nastopajo za posamezen datum v dispoziciji. Znotraj lokacije so urniki, ki so opredeljeni za to lokacijo. Vsak vodja izvedbenega oddelka lahko potrdi le tisti del dispozicije oziroma tisti oddelok, za katerega nastopa kot vodja le-tega.

Datum			
Dvorana / Oddelki	<input type="checkbox"/> Potrdi	<input type="checkbox"/> Potrdi	<input type="checkbox"/> Potrdi
dvorana 1	oddelek1	oddelek2	oddelek3
05:00 - 7:00 urnik1 8:00 - 12:00 urnik2			
12:00 - 13:00 urnik3			
12:00 - 18:00 urnik4			

Slika 20: Prototip št.4 (Dispozicije) - primer dispozicije za potrjevanje le-te s strani vodij izvedbe

Tretji del prototipa prikazuje določanje urnikov izvedbe glede na urnike projekta. Na levi strani je seznam urnikov projektov. Na drugi strani je seznam oseb oziroma izvedbenih delavcev. Uporabnik iz levega seznama določi nabor urnikov projekta, iz desnega seznama določi osebe, ki bodo nastopale za izbrane urnike na levi. Pri osebah določi vlogo, ki jo bo oseba imela in uro, kdaj bo oseba nastopala znotraj urnika projekta (po privzetem je to ura urnika projekta). Uporabnik lahko zabeleži tudi morebitno opombo. Na koncu potrdi vnos z gumbom.

Urniki							Izvedbeni delavci					
IDU	Prireditev	Lokacija	Aktivnost	Datum	Ura	Označi	ID	Oseba	Vloga	Ura	Opomba	Označi
Databound	Databound	Databound	Databound	Databound	Databound	<input type="checkbox"/>	Databound	Databound		Dat: Dat:	Databound	<input type="checkbox"/>
Databound	Databound	Databound	Databound	Databound	Databound	<input type="checkbox"/>	Databound	Databound		Dat: Dat:	Databound	<input type="checkbox"/>
Databound	Databound	Databound	Databound	Databound	Databound	<input type="checkbox"/>	Databound	Databound		Dat: Dat:	Databound	<input type="checkbox"/>
Databound	Databound	Databound	Databound	Databound	Databound	<input type="checkbox"/>	Databound	Databound		Dat: Dat:	Databound	<input type="checkbox"/>
Databound	Databound	Databound	Databound	Databound	Databound	<input type="checkbox"/>	Databound	Databound		Dat: Dat:	Databound	<input type="checkbox"/>

Vnesi urnike

Slika 21: Prototip št.4 (Urniki) - prikaz seznama urnikov in seznama oseb za določanje urnikov izvedbe

Četrti del prototipa prikazuje prikaz in urejanje obstoječih urnikov izvedbe (slika spodaj). V seznamu so urniki izvedbe na osnovi urnikov projektov. Vmesnik omogoča brisanje urnikov ali urejanje obstoječih (oseba, vloga, datum, ura, opomba). Spremembe uporabnik potrdi z gumbom.

IDU	Prireditev	Lokacija	Aktivnost	Datum	Ura	Oseba	Vloga	Datum	Ura	Opomba	Označi
Databound	Databound	Databound	Databound	Databound	Databound						<input type="checkbox"/>
Databound	Databound	Databound	Databound	Databound	Databound						<input type="checkbox"/>
Databound	Databound	Databound	Databound	Databound	Databound						<input type="checkbox"/>
Databound	Databound	Databound	Databound	Databound	Databound						<input type="checkbox"/>
Databound	Databound	Databound	Databound	Databound	Databound						<input type="checkbox"/>

Shrani spremembe izbranih

Slika 22: Prototip št.4 (Urniki) - Seznam urnikov izvedbe

4.2.3. Opredelitev strategije testiranja

Testiranje bo potekalo za vsak modul oziroma vmesnik posebej:

- 1.vmesnik- Rezervacije
- 2.vmesnik - Projekti
- 3.vmesnik - Dispozicije
- 4.vmesnik - Urniki

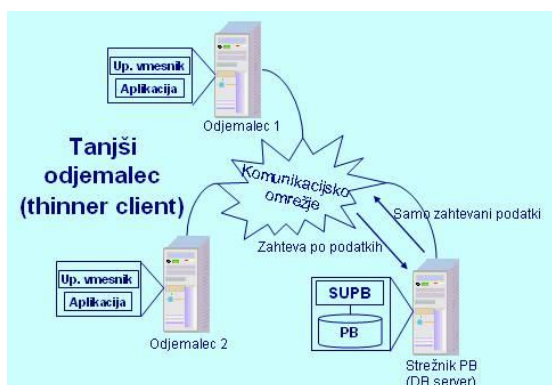
Vsak modul posebej bo najprej okvirno stestiral sam razvijalec v okviru razvojnega okolja, kasneje ga bo dokončno stestiral določen uporabnik. Testno okolje bo nameščeno tako pri izvajalcu kot uporabniku. Okvirno testiranje za razvijalca bo potekalo na lokalnem računalniku v operacijskem sistemu Windows 7 in prav tako tudi pri uporabniku. Testno

okolje se bo namestilo ob koncu razvoju vsakega modula posebej, potekalo bo tudi v več iteracijah. Na koncu, ko bodo vsi moduli dokončno razviti, se bo stestiralo vse module skupaj kot celoto.

4.3. Načrtovanje

Glavni namen načrtovanja je izdelati načrt zgradbe sistema na podlagi specifikacij zbranih v fazi analize. Osnovna rezultata načrtovanja sta načrt podatkovne baze in načrt programskih modulov, s katerima pripravimo vse potrebno za izdelavo podatkovnih in programskih komponent IR.

4.3.1. Načrt zgradbe sistema



Zgradba sistema je odvisna od funkcijskih in nefuncijskih zahtev zbranih v fazi analize. V našem primeru načrtovanja arhitekture sistema ni bilo potrebno, saj smo IR namestili na sistem ki je že vzpostavljen in je v delovanju. Arhitektura na obstoječem sistemu je dvonivojska, sestavljena iz več odjemalcev in podatkovnega oziroma aplikacijskega strežnika.

Slika 23: Dvonivojska arhitektura

4.3.2. Načrt podatkovne baze

Za hranjenje podatkov bomo uporabili relacijsko podatkovno bazo. Za izdelavo načrta podatkovne baze smo si pomagali z konceptualnim modelom izdelanim v fazi analize. Izdelava logičnega modela je bila preprosta, saj smo konceptualni model izdelali z orodjem Power Designer 12. V nadaljevanju omogoča izdelavo logičnega modela na podlagi konceptualnega, ter skripte za izdelavo fizičnega podatkovnega modela.

Podatkovni model sestavlja prek 20 tabel, med glavnimi spadajo naslednje z nekaterimi glavnimi atributi:

URN_Projekt – tabela z seznamom vseh atributov ki opisujejo projekt (naziv, datum, status, vodja, koda,...)

URN_Urnik Projekt – tabela z atributi ki opisuje urnik projekta (projekt, začetek, konec, status, lokacija,...)

URN_Urnik_Projekt_Oseba – tabela z atributi ki definira urnik osebe na podlagi urnika projekta (urnik, oseba, začetek, konec, vloga)

URN_Oseba – tabela šifrant z opisom oseb (ime, priimek, email, tel, sektor, oddelek, pododdelek)

DIS_Dispozicija – tabela z atributi, ki definira vsako dispozicijo(projekt,oseba, potrjena,datum)

DIS_Urnik_Seznam – tabela z množico urnikov projekta (dispozicija,urnik,datum)

Opis glavnih tabel podatkovnega modela:

URN Projekt je primarna tabela iz katere so povezane ostale tabele. Projekt določajo tudi šifranti; URN Status, URN Tip,URN Oseba, ki definirajo status projekta, tip projekta in vodjo projekta, ter kdo je projekt vnesel.

Vsak projekt ima nič ali več urnikov - URN urnik projekt. Tabela URN urnik projekt določa več šifrantov; URN Status, URN Lokacija in URN Aktivnost.

Vsak urnik projekta pa ima več urnikov oseb - URN Urnik Projekt Oseba, Katere šifranta sta URN Oseba in URN Vloga.

Vsak projekt ima nič ali več dispozicij – URN Dispozicija, katere šifrant je oseba - URN Oseba, ki je dispozicijo vnesla. Vsaka dispozicij pa ima nič ali več seznamov urnikov - URN Seznam Urnik, ki določajo množico urnikov projekta – URN Urnik Projekt.

Opis ostalih tabel podatkovnega modela:

Tabela URN Organizator je šifrant organizatorjev. Vsak organizator lahko nastopa na več projektih in prav tako ima lahko en projekt več organizatorjev. Za slednjo relacijo se vzpostavi vmesna tabela. Organizator vsebuje tudi šifranta URN Drzava. Tabela URN Porocilo vsebuje natanko en projekt in vsebuje okvirne attribute za opis, ko je projekt izveden. Tabela URN Zahteve vsebuje natanko en projekt in vsebuje okvirne attribute za opis zahtev naročnika. URN Kategorija je tabela za vpis kriterijev projekta in pripada natanko enem projektu. URN Sektor, URN Oddelek in URN Pododdelek so šifranti, ki se nanašajo na osebo oziroma oseba pripada v točno en sektor, v en ali nič oddelkov in v enega ali nič pododdelkov. Tabele URN Sektor Oseba, URN Oddelek Oseba in URN Pododdelek Oseba so vmesne tabele med osebo in šifranti. Njihov namen je vpisovanje pravic uporabnikov za dostop do aplikacij. DIS Priloga je tabela, ki pripada natanko eni dispoziciji z atributi za opis priloge dispozicij. DIS Seznam Podrocje in DIS Podrocje sta tabeli z atributi za opis zahtev vodij projektov. Tabela DIS Urnik Seznam Oprema je namenjena vodji izvedbe za rezervacijo opreme na podlagi prej omenjene zahteve. Oprema pa je zavedena v šifrantu DIS Oprema. Tabela Log je ločena tabela, ki ni v relaciji z nobeno drugo tabelo. Njen namen je vpisovanje dostopov uporabnikov do aplikacij.

4.3.3. Načrt programskih modulov

V okviru obravnavanega opravila je potrebno izdelati načrt programskih modulov, kar pomeni (s pomočjo v standardih načrtovanja programskih modulov opredeljenih diagramskih tehnik) izdelati funkcionalne specifikacije modulov in dokumentirati njihovo funkcionalnost.

V začetku je potrebno izdelati seznam vseh programskih modulov in sicer za vsak vmesnik opredeljen v fazi analize posebej. Za vsak modul posebej se opredeli njegov tip (zaslonska maska, izpis, paketna obdelava oziroma procedura) in morebiten komentar.

V nadaljevanju pa je potrebno za vsak vmesnik posebej prikazati strukturo aplikativnega sistema z vidika hierarhične strukture modulov. Arhitekturo aplikativnega sistema se najpogosteje predstavi hierarhično z uporabo diagramske tehnike strukturnih diagramov. Strukturni diagrami po eni strani prikazujejo hierarhijo modulov aplikativnega sistema, po drugi pa tudi njihovo strukturo. Strukturni diagrami so dekompozicijski diagrami, kar pomeni, da lahko celotno hierarhijo prikažemo z nekaj velikimi strukturnimi diagrami.

Namen strukturnih diagramov je grafična predstavitev odvisnosti med moduli aplikativnega sistema tako z vidika medsebojnega klicanja modulov, kot tudi z vidika prehajanja podatkov ter z vidika osnovnih komponent strukturnega programiranja: zaporedje, izbira in ponavljanje. Za strukturne diagrame so definirana tudi pravila, ki omogočajo prevedbo poteka in obdelave podatkov kot je prikazana v diagramu podatkovnih tokov v potek in obdelavo podatkov kot je prikazana v strukturnih diagramih.

V strukturnih diagramih nastopajo naslednji tipi modulov:

- **Menijski moduli:** Menijski moduli se nahajajo ne vrhu hierarhije. Njihova naloga je povezava modulov vseh tipov v menijsko strukturo, pri čemer je meni v aplikativnem sistemu lahko eden, lahko pa jih je tudi več. V primeru več menijev je za vsakega od njih potrebno narediti svoj strukturni diagram. Na vrhu hierarhije je vedno korenski menijski modul, ki kliče bodisi ostale menijske module, bodisi direktno zaslonske module.
- **Zaslonski moduli:** Zaslonski moduli predstavljajo module uporabniškega vmesnika, torej module, ki podpirajo interakcijo uporabnika z aplikativnim sistemom. Zaslonski moduli so lahko zaslonske maske ali izpisi. Kličejo lahko druge zaslonske module. Za svoje delovanje pa uporabljajo, kličejo programske module.
- **Programski moduli:** Programski moduli so procedure in funkcije, katerih naloga je implementirati bodisi procesno ali funkcijsko logiko, bodisi poslovno pravilo. Programski moduli se medsebojno kličejo in so zaradi preglednosti organizirani v hierarhije.

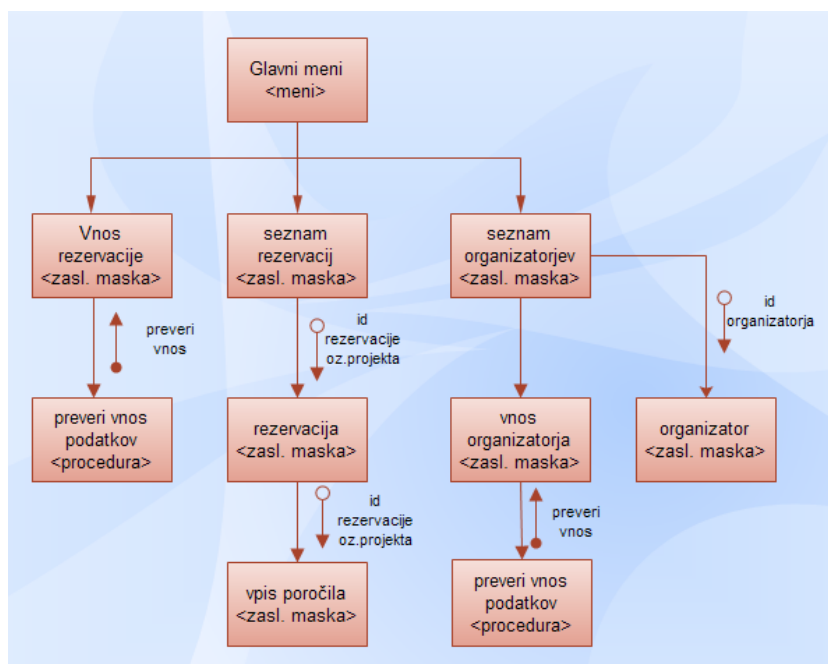
Na najvišjem nivoju strukturnega diagrama programskega modula se nahaja en glavni modul (koren), na naslednjem nivoju pa moduli, ki jih glavni modul kliče. Vsak modul, ki je podrejen korenskemu modulu, lahko kliče več modulov na naslednjem nivoju. Moduli na najnižjem nivoju ne kličejo drugih modulov, ampak le izvajajo specifične funkcije.

Moduli v strukturnem diagramu komunicirajo med seboj s pomočjo prenosa parametrov (nosilcev podatkov) in kontrolnih informacij (kontrolnih zastavic).

4.3.3.1. Opis programskih modulov za vmesnik »Rezervacije«

Oznaka modula	Tip modula	komentar
Glavni meni	Meni	Glavni korenski meni
Vnos rezervacije	Zaslonska maska	Vnos projekta z urniki rezervacij
Vnos organizatorja	Zaslonska maska	Šifrant
Vnos poročila rezervacije	Zaslonska maska	Vnos poročila
Rezervacija	Zaslonska maska	Pregled podrobnosti rezervacije
Organizator	Zaslonska maska	Pregled podrobnosti organizatorja
Seznam rezervacij	Zaslonska maska	Izpis seznama rezervacij oziroma projektov
Seznam organizatorjev	Zaslonska maska	Izpis seznama organizatorjev
Preveri vnos podatkov rez.	Procedura	
Preveri vnos podatkov org.	Procedura	

Tabela 1: Opis programskih modulov za vmesnik »Rezervacije«

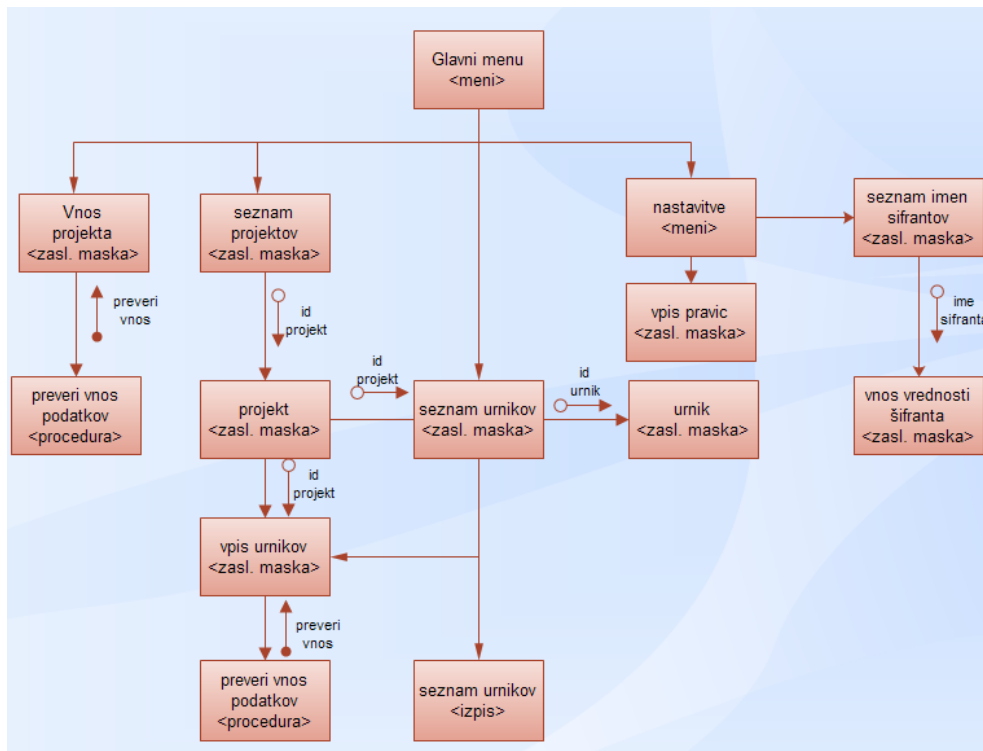


Slika 25: Predstavitev programskih modulov aplikacije "Rezervacije" s strukturnim diagramom

4.3.3.2. Opis programskih modulov za vmesnik »Projekti«

Oznaka modula	Tip modula	Komentar
Glavni meni	Meni	Glavni korenski meni
nastavitve	Meni	Podmeni za nastavitve
Vnos projektov	Zaslonska maska	Vnos in urejanje projektov
Vnos urnikov	Zaslonska maska	Vnos in potrjevanje urnikov
Vnos pravic	Zaslonska maska	Vnos pravic dostopov oseb
Vnos šifrantov	Zaslonska maska	Šifranti – oseba, tip_prireditve, status, lokacija, aktivnost, vloga, področje, sektor, oddelek, poddodelek
Projekt	Zaslonska maska	Izpis podatkov projekta
Urnik	Zaslonska maska	Izpis podatkov urnika
Seznam projektov	Zaslonska maska	Izpis projektov
Seznam urnikov	Zaslonska maska	Izpis urnikov
Seznam urnikov	Izpis	Tiskanje urnikov
Preveri vnos podatkov urnika	Procedura	
Preveri vnos podatkov projekta	Procedura	

Tabela 2: Opis programskih modulov za vmesnik »Projekti«

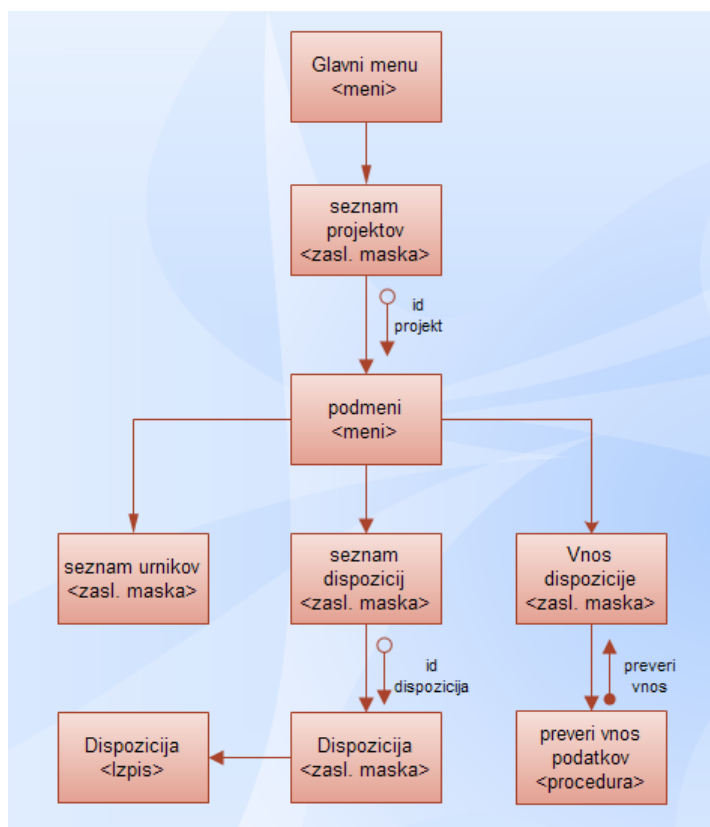


Slika 26: Predstavitev programskih modulov aplikacije "Projekti" s strukturalnim diagramom

4.3.3.3. Opis programskih modulov za vmesnik »Dispozicije«

Oznaka modula	Tip modula	Komentar
Glavni meni	Meni	Glavni korenski meni
Podmeni	Meni	podmeni
Seznam projektov	Zaslonska maska	Izpis projektov
Seznam urnikov	Zaslonska maska	Izpis urnikov projekta za razčlenjevanje
Dispozicija	Zaslonska maska	Izpis podatkov dispozicije
Seznam dispozicij	Zaslonska maska	Seznam dispozicij projekta
Vnos dispozicije	Zaslonska maska	Vnos dispozicije
Dispozicija	Izpis	Tiskanje dispozicije
Preveri vnos podatkov dispoz.	Procedura	

Tabela 3: Opis programskih modulov za vmesnik »Dispozicije«

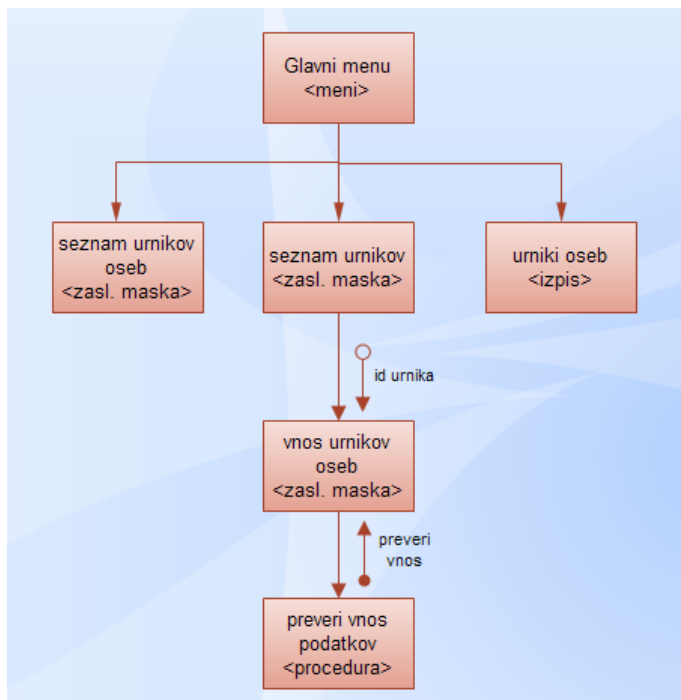


Slika 27: Predstavitev programskih modulov aplikacije "Dispozicije" s strukturalnim diagramom

4.3.3.4. Opis programskih modulov za vmesnik »Urniki«

Oznaka modula	Tip modula	komentar
Glavni meni	Meni	Glavni korenski meni
Seznam urnikov	Zaslonska maska	Izpis urnikov projekta
Seznam urnikov oseb	Zaslonska maska	Izpis urnikov oseb oziroma izvedbe
Vnos urnikov oseb	Zaslonska maska	Vnos urnikov
Urniki oseb	Izpis	Tiskanje urnikov
Preveri vnos podatkov urnikov o	Procedura	

Tabela 4: Opis programskih modulov za vmesnik »Urniki«



Slika 28: Predstavitev programskih modulov aplikacije "Urniki" s strukturalnim diagramom

4.4. Implementacija

Za implementacijo programske opreme je bilo uporabljeno programsko orodje Visual Studio 2008. Kot osnova pri programiranju je bil uporabljen jezik C#. Pri razvoju spletnih oziroma .NET aplikacij je bil uporabljen tudi skriptni jezik JavaScript in pa tudi jQuery. Za poenotenje stila spletnih aplikacij je uporabljena CSS tehnologija. Celotna informacijska rešitev je bila v osnovi razvita po sklopih oziroma po vmesnikih opredeljenih v prejšnji fazi analize in načrtovanja:

- rezervacije
- projekti
- dipozicije
- urniki

Za razvoj fizične podatkovne baze je bila uporabljena programska oprema MySQL server 2005. Zaradi lažjega poizvedovanja in interpretacije podatkov so v pomoč pogledi (ang. Views) shranjeni na podatkovnem strežniku. V Ideji pa je bilo da bi se čim več procesiranja podatkov preneslo na podatkovni strežnik in tako so v veliko pomoč shranjene podatkovne procedure (ang. stored procedure), ki so shranjene na podatkovnem strežniku. Za generiranje pogledov in podatkovnih procedur je bilo v pomoč orodje SQL Server management studio. Shranjevanje podatkovnih procedur na podatkovnem nivoju prinese veliko prednosti. Samo poizvedovanje je odzivnejše in hitrejše. Lažje je vzdrževanje in sistem je bolj sistematičen oziroma preglednejši.

4.4.1. Implementacija vmesnika »Rezervacije«

Implementacija vmesnika je potekala brez večjih zapletov. Zahteve so se tekom razvoja nekoliko spreminjale, vendar zaradi malega števila udeleženih v tem procesu je bila implementacija tega vmesnika med hitrejšimi in enostavnejšimi. Vmesnik je namenjen oddelku za marketing, ki obsega približno 4-7 osebe in je dosegljiv kot spletna aplikacija na portalu.

Aplikacija omogoča vnos rezervacij, pregled rezervacij z možnostjo filtriranja, spreminjanje obstoječih podatkov o rezervaciji in vnos poročila (v primeru potrditve in izvedbe projekta), pregled organizatorjev in vnos organizatorjev le-teh kot šifrantov.

Nova Rezervacija

| Osnovni podatki |

Slovenski naziv prireditve:
4. Festival Goriškega filma

Angleški naziv prireditve:

Status:
 Rezervacija
 Potrjeno
 Storno

Približna vsebina prireditve:

Opombe:

Kraj prireditve: Ljubljana, CD Datum: 06.06.2010 Lokacija: M4 Vnesi rezervacijo lokacije: Vnesi

Začetek prireditve: Tue, 01.06.2010 12:00

Konec prireditve: Sun, 06.06.2010 12:00

Št. udeležencev:

Št. držav: 1

ID	Datum	Lokacija	Aktivnost	Odstrani
6	tor, 01.06.2010	Klub CD	Rezervacija	-
9	sre, 02.06.2010	Klub CD	Rezervacija	-
10	čet, 03.06.2010	Klub CD	Rezervacija	-
11	čet, 03.06.2010	M1	Rezervacija	-
12	pet, 04.06.2010	M1	Rezervacija	-
13	sob, 05.06.2010	M4	Rezervacija	-
14	ned, 06.06.2010	M4	Rezervacija	-

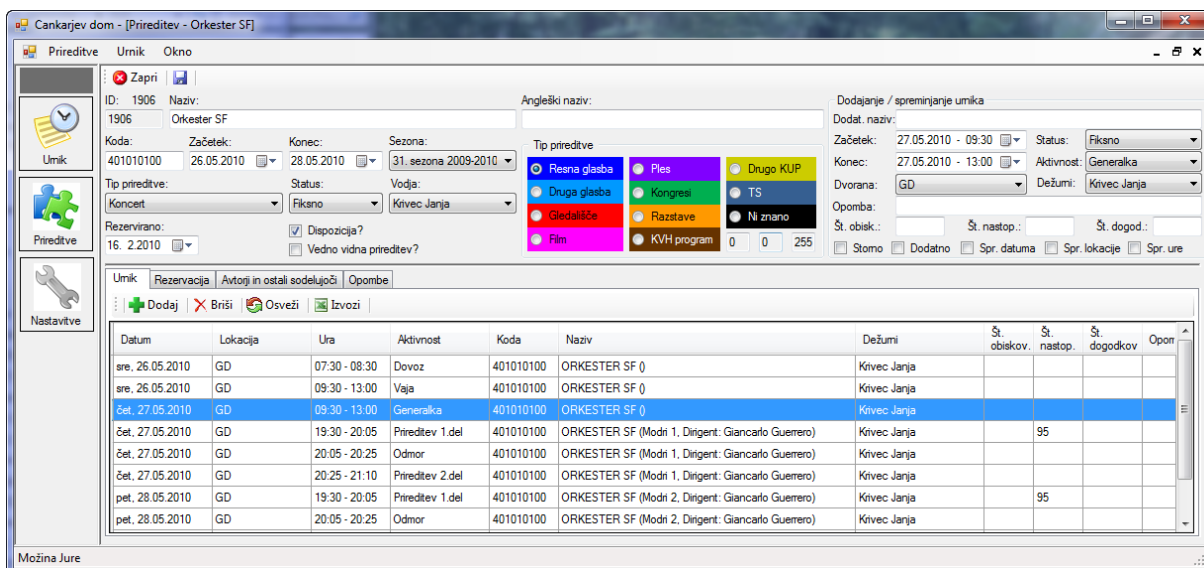
Nadaljuj

Slika 29: Prikaz delne zaslonske maske aplikacije »Rezervacije« za vnos nove rezervacije

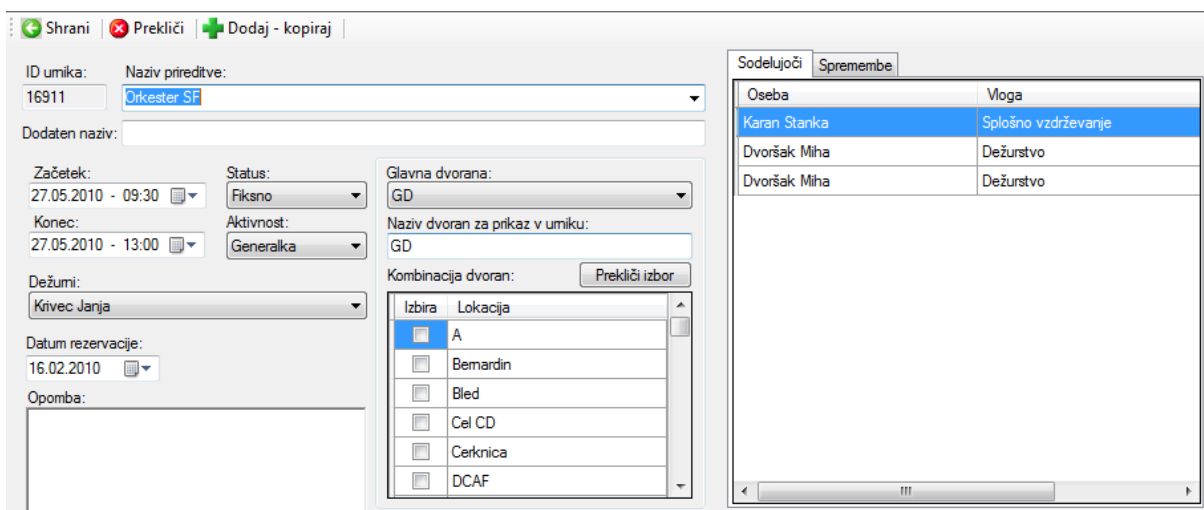
4.4.2. Implementacija vmesnika »Projekti«

Za implementacijo slednjega vmesnika je bilo potrebno več časa, saj je le-ta sestavljen iz več programskih modulov. Za razliko od ostalih vmesnikov, je opredeljen kot windows aplikacija in je nameščen lokalno na računalniku pri programskem koordinatorju. Z drugega vidika je bil razvoj le-tega tudi relativno nezahteven, saj je vmesnik namenjen le enemu udeležencu v procesu – programskemu koordinatorju, kar pomeni manj zahtev, manj spreminjajočih zahtev. Med drugim aplikacija vsebuje tudi druge programske module, ki niso v interesu tega diplomskega dela in zato niso zajeti.

Aplikacija omogoča upravljanje z skoraj vsemi oziroma večino podatkov zajetih v celotnem procesu. Omogoča vnašanje projektov, pregled le-teh in spreminjanje obstoječih projektov ter vnos vseh podatkov glede projekta: pregled urnikov projekta, spreminjanje in definiranje urnikov projekta, potrjevanje urnikov rezervacij ali vnos le-teh. Naloga programskega koordinatorja je tudi, da skrbi za vnos oziroma dodajanje šifrantov. Poleg tega skrbi tudi za pravice dostopov do aplikacij.



Slika 30: Prikaz zaslonske maske aplikacije »Projekti« za prikaz podatkov o projektu



Slika 31: prikaz zaslonske maske aplikacije »Projekti« za prikaz podatkov o urniku

4.4.3. Implementacija vmesnika »Dispozicije«

Za implementacijo tega vmesnika je bilo potrebno več časa. Aplikacija je namenjena večim osebam, udeleženi v procesu. Z ene strani vodjem projektov, z druge strani vodjem izvedbenih oddelkov (skupaj približno 15 oseb). Pri razvoju je bilo prisotno veliko zahtev in idej, ki jih je bilo potrebno poenotiti in porazdeliti. Za razliko od ostalih vmesnikov, je to edina aplikacija, ki je še v razvoju in pri kateri se pričakujejo še dodatne zahteve. Veliko časa je bilo potrebno za izdelavo interaktivnega spletnega obrazca oziroma dispozicije, na katero lahko vodje projektov zavedejo razčlenjene urnike, ki so časovno in prostorsko opredeljeni. V nadaljevanju za vsako množico urnikov tudi navedejo tehnično specifikacijo po izbranih oddelkih. Aplikacija je namenjena vodjem projektov, saj le-ti izdelujejo dispozicije in podajajo

tehnične zahteve, namenjena je pa tudi vodjem izvedbenega oddelka saj morajo dispozicijo pregledati in potrditi.

Aplikacija omogoča pregled dispozicij projektov, vnos novih dispozicij in urejanje le teh, potrjevanje dispozicij in objavo dispozicij. V aplikaciji vodje projektov tudi razčlenijo urnike projekta.

Projekt: TESTNA DISPOZICIJA

Uredi urnike | Izdelava Dispozicije | Uredi dispozicijo | Osveži

Uredi

Tip dispozicije: Tip1 | Izberi dispozicijo: Dispozicija1

Datum: čet, 01.04.2010

Uredi

ID	Področje	Izberi
1	Scena	<input checked="" type="checkbox"/>
2	Luč	<input checked="" type="checkbox"/>
3	Ton	<input checked="" type="checkbox"/>
4	Video	<input checked="" type="checkbox"/>
5	ITK	<input checked="" type="checkbox"/>
6	Inspekcija	<input type="checkbox"/>

Tiskanje: Tiskaj

Objava: Objavi

Priloge: Priloge

Izvozi: Objavljena? Browse...

Osveži

čet, 01.04.2010

Dvorana / Oddelki	Scena	Luč	Ton	Video	ITK	
+ Štišhova dvorana						
07:00 - 08:00 Aranžirka 10:00 - 12:00 Čiščenje kletke 12:00 - 16:00 Cocktail Uredi	1 x oder	2 x luč	3 x mikrofoni	3 x lcd 22"	3 rač	Shrani
14:00 - 16:00 Bralne vaje Uredi	1 x oder	2 x luč	3 x mikrofoni			
15:00 - 16:00 Class Uredi	1 x oder	2 x luč	3 x mikrofoni			
+ S1						
01:00 - 02:00 Ton Uredi	1 x oder					Shrani

Slika 32: prikaz zaslonske maske aplikacije »Dispozicije« za urejanje dispozicije z vidika vodje projekta

4.4.4. Implementacija vmesnika »Urniki«

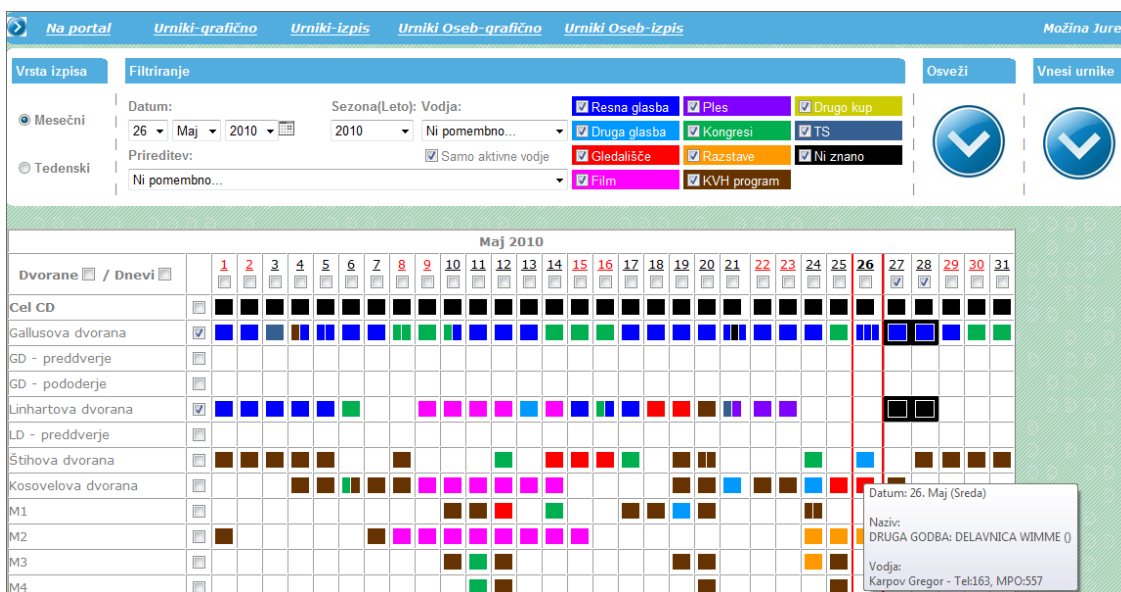
V Cankarjevem domu vsak dan potekajo najrazličnejše dogodki oziroma projekti. Na razpolago je približno 20 prostorov, znotraj Cankarjevega doma. Za vsakega izmed njih potekajo najrazličnejše aktivnosti oziroma urniki, ki so opisani z neko aktivnostjo. Na leto je zabeleženih približno čez 800 projektov skupaj približno 8.000 urnikov, kar je v povprečju približno 600 urnikov na mesec! Ob tako veliki količini podatkov pa je samo obvladovanje, interpretacija in neko poizvedovanje dokaj naporno. Zato je za dobro interpretacijo podatkov, grafična vizualizacija podatkov lahko odlična rešitev.

Implementacija vmesnika Urniki je bila najbolj obsežna, dolgotrajna, zahtevna, obenem pa tudi najbolj zanimiva. Ideja, ki bi grafično prikazovala urnike je bila prisotna že od začetka samega razvoja celotne informacijske rešitve. Aplikacija med drugim omogoča:

- Grafični prikaz urnikov projektov
- Grafični prikaz urnikov oseb, ki so v odvisnosti od urnikov projekta.

4.4.4.1. Grafični prikaz urnikov projektov

Aplikacija prikazuje potek urnikov po dvorinah, kjer so urniki prikazani z časovnimi enotami v obliki časovnih premic. Aplikacija omogoča tri načine pregledov (Mesečni, tedenski in dnevni). Pri mesečnem pregledu so časovne enote dnevi, pri tedenskem in dnevnem pa ure. Časovne enote imajo tudi ustrezno barvo, ki določajo tip prireditve oziroma projekta. Znotraj časovnih enot so tudi oznake, ki določajo tip aktivnosti oziroma urnika. Aplikacija omogoča tudi filtriranje po najrazličnejših parametrih (projekt, tip projekta, datum, vodja projekta, itd.).



Slika 33: prikaz zaslonske maske pri grafičnem izpisu oziroma prikazu urnikov projektov aplikacije »Urniki«

Na sliki zgoraj je prikazana množica urnikov po prostorskih virih in dnevih. Urniki so prikazani v obliki časovnih enot, ki so opredeljene z določeno barvo, ki definira tip projekta. Časovne enote so tudi interaktivne. S postavitvijo kurzorja se nam izpišejo določene informacije, ob kliku na dnevno časovno enoto pa se prikaže podroben opis časovne enote po urah (slika spodaj).



Slika 34: prikaz zaslonske maske aplikacije »urniki« pri grafičnem izpisu urnikov projektov (dnevni prikaz)

Aplikacija je namenjena praktično vsem udeleženi v procesu saj odlično prikazuje raspored urnikov po dvoranah in na nek sistematičen način dober vpogled zasedonosti prostorskih in časovnih virov, kar je za programskega koordinatorja velikega pomena, pri potrjevanju in določanju urnikov projektov. V veliko pomoč je tudi vsem vodjem projektov, pri razčlenjevanju urnikov in pri pregledu svojih projektov. Glavnega pomena je pa za vodje izvedbe saj preko te aplikacije razporejajo izvedbene delavce oziroma izdelajo urnike izvedbenih delavcev. Vodja izvedbe izbere prostorske in časovne vire in tako dobi seznam vseh urnikov projektov, ki so osnova za izdelavo urnikov izvedbenih delavcev. (slika spodaj).

		Maj 2010												
Dvorane / Dnevi		1	2	3	4	5	6	7	8	9	10	11	12	13
Cel CD	<input type="checkbox"/>	█	█	█	█	█	█	█	█	█	█	█	█	█
Gallusova dvorana	<input type="checkbox"/>	█	█	█	█	█	█	█	█	█	█	█	█	█
GD - preddverje	<input type="checkbox"/>													
GD - pododerje	<input type="checkbox"/>													
Linhartova dvorana	<input checked="" type="checkbox"/>	█	█	█	█	█	█	█	█	█	█	█	█	█
LD - preddverje	<input type="checkbox"/>													
Štihova dvorana	<input checked="" type="checkbox"/>	█	█	█	█	█	█	█	█	█	█	█	█	█
Kosovelova dvorana	<input type="checkbox"/>													

Slika 35: prikaz dela zaslonske maske aplikacije »urniki«. Primer izbire urnikov za vnos urnikov izvedbenih delavcev

Urniki							Sodelujoči						
IDU	Prireditel	Lokacija	Aktivnost	Datum	Ura	Oznaki	ID	Oseba	Vloga	Ura	Opomba	Oznaki	
20541	MARTIN IN ...	ŠD	Ton	pon, 03.05.2010	8:00-10:00	<input type="checkbox"/>	1	Avberšek Aleksander	Dežurstvo	0.00	0.00	<input type="checkbox"/>	
21238	Gledališki...	LD	Luč	pon, 03.05.2010	8:00-16:00	<input type="checkbox"/>	5	Bitežnik Gregor	ITK podpora	0.00	0.00	<input checked="" type="checkbox"/>	
11971	MARTIN IN ...	ŠD	Prireditel	pon, 03.05.2010	11:00-11:45	<input checked="" type="checkbox"/>	4	Bogataj Niko	Dežurstvo	0.00	0.00	<input type="checkbox"/>	
20542	MARTIN IN ...	ŠD	Prireditel	pon, 03.05.2010	12:30-13:15	<input checked="" type="checkbox"/>	6	Kolenko Zdravko	Dežurstvo	0.00	0.00	<input type="checkbox"/>	
14015	Gledališki...	LD	Priprava	pon, 03.05.2010	17:00-19:00	<input type="checkbox"/>	185	Možina Jure*	ITK podpora	0.00	0.00	<input checked="" type="checkbox"/>	
20151	Gledališki...	LD	Prireditel 1.del	pon, 03.05.2010	19:30-20:25	<input type="checkbox"/>							
20152	Gledališki...	LD	Odmor	pon, 03.05.2010	20:25-20:45	<input type="checkbox"/>							
20153	Gledališki...	LD	Prireditel 2.del	pon, 03.05.2010	20:45-21:55	<input type="checkbox"/>							

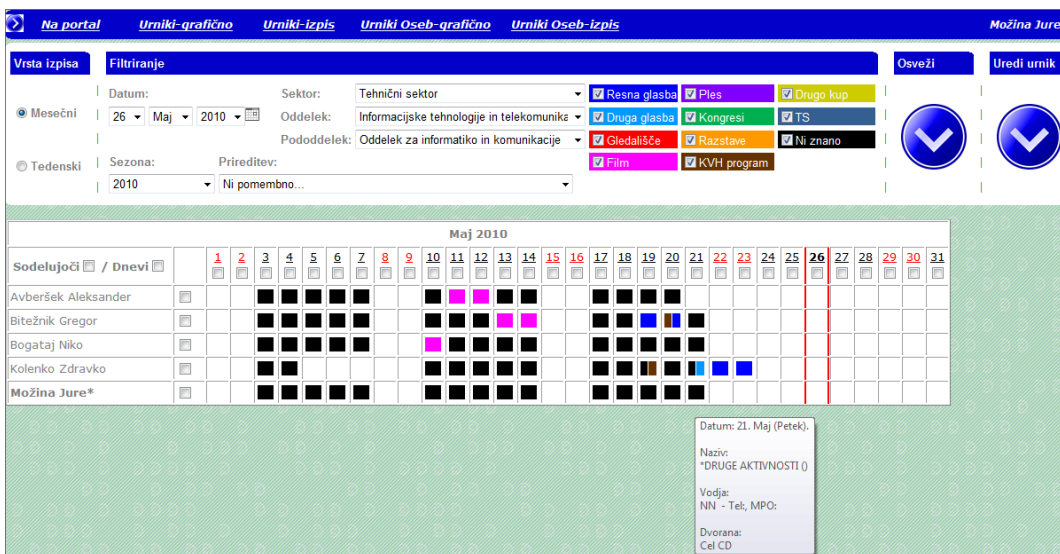
Urniki Sodelujočih											
IDU	Prireditel	Lokacija	Aktivnost	Datum	Ura	Oseba	Vloga	Datum	Ura	Opomba	Priloga
11971	MARTIN IN ...	ŠD	Prireditel	pon, 03.05.2010	11:00-11:45	Bitežnik Gregor	ITK podpora	03.05.2010	11.00	11.45	<input checked="" type="checkbox"/>
11971	MARTIN IN ...	ŠD	Prireditel	pon, 03.05.2010	11:00-11:45	Možina Jure*	ITK podpora	03.05.2010	11.00	11.45	<input checked="" type="checkbox"/>
20542	MARTIN IN ...	ŠD	Prireditel	pon, 03.05.2010	12:30-13:15	Bitežnik Gregor	ITK podpora	03.05.2010	12.30	13.15	<input checked="" type="checkbox"/>
20542	MARTIN IN ...	ŠD	Prireditel	pon, 03.05.2010	12:30-13:15	Možina Jure*	ITK podpora	03.05.2010	12.30	13.15	<input checked="" type="checkbox"/>

Slika 36: zaslonska maska aplikacije »Urniki«. Prikazuje rezultat izbire urnikov projektov z namenom vnosa urnikov izvedbe

Na sliki (slika zgoraj) se na levem seznamu nahaja nabor unikov projektov. Na desnem seznamu je seznam virov, s katerimi lahko vodja razporeja glede na urnike. Rezultat vnosa so urniki izvedbe.

4.4.4.2. Grafični prikaz urnikov oseb

Drugi del aplikacije deluje po istem principu kot prvi del, le da so tukaj prikazani drugi viri oziroma osebe namesto prostorskih virov. Prikazani so rezultati vnosa prvega dela aplikacije. Prikazani so urniki izvedbe. Prav tako so tudi tukaj časovne enote prikazane v obliki dnevov pri mesečnem pregledu in ure pri tedenskem in dnevnom pregledu. Vsaka časovna enota je barvno določena kar pomeni tip projekta na katirem oseba dela. Z filtri aplikacija omogoča izpis urnikov izvedbe po oddelkih oziroma sektorjih ali pa po projektih. Aplikacija odlično prikazuje zasedenost oziroma planiranje izvedbenih delavcev po dnevih. V prvi vrsti je namenjena vodjem izvedbenih oddelkov za planiranje in koordiniranje in pa tudi vsem ostalih, saj prikazuje sistematičen in zanimiv vpogled kdaj, kdo, kje in za koga dela.



Slika 37: prikaz zaslonske maske pri grafičnem izpisu oziroma prikazu urnikov izvedbe, aplikacije »Urniki«



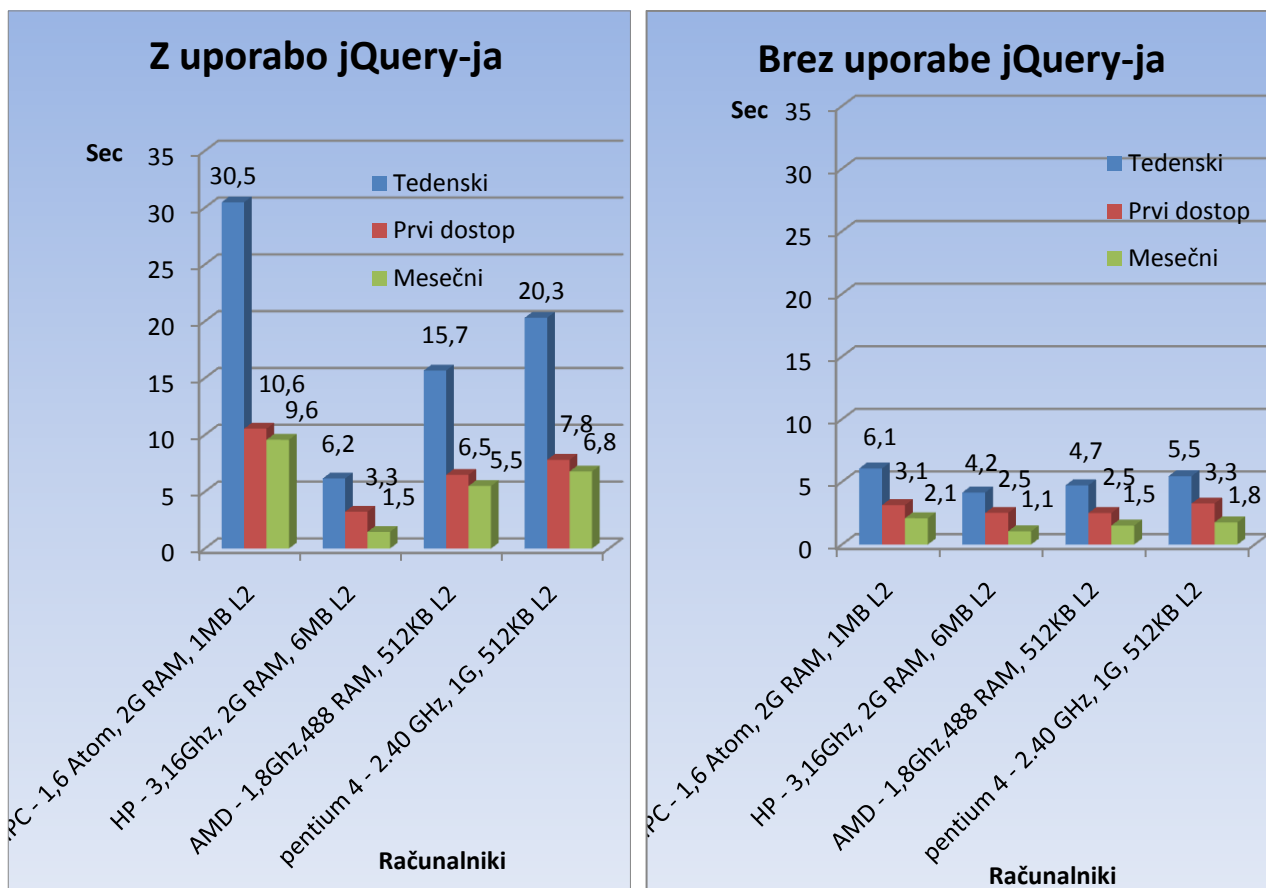
Slika 38: prikaz zaslonske maske aplikacije »urniki« pri grafičnem izpisu urnikov izvedbe(dnevni prikaz)

4.5. Testiranje

Testiranje je potekalo kot je bilo opredeljeno v fazi analize, za vsak modul oziroma vmesnik posebej. Nekatero napake so bile odpravljene že tekom razvoja s strani razvijalca, druge so bile odkrite kasneje s strani uporabnika. Ko je bil modul razvit in bil predan v uporabo uporabniku, so bili izvedeni pozitivni in negativni scenariji. Pozitivni scenariji so tisti, pri katerih napak ne simuliramo in tako mora aplikacija delovati brezhibno. Negativni scenariji so tisti pri katerih ustvarimo oziroma simuliramo napako. Negativni scenariji so bili izvedeni predvsem tam, kjer gre za vnos podatkov oziroma tam, kjer je v ozadju kontrola vnosa. Gre predvsem za vnos datumov in numeričnih podatkov. Namen tega je, da preverimo odziv spletne aplikacije. Odzvati se mora z ustreznim opozorilom.

Za vsako aplikacijo je bil izveden tudi test obremenitve in postavljena časovna analiza. Za razliko od ostalih aplikacij, pri katerih so bili pričakovani rezultati odzivnosti, je izstopala spletna aplikacija »Urniki«, ki omogoča grafičen pregled urnikov projektov in urnikov izvedbe. V začetku je bila odzivnost aplikacije zelo slaba. V nadaljevanju je bila izvedena optimizacija, ki je močno izboljšala odzivnost aplikacije. Uporabili smo shranjene sql procedure na strežniku, ki jih aplikacija kliče le, ko se prvič naloži v brskalniku pri odjemalcu in shrani rezultat poizvedb v statične podatkovne objekte tipa »DataView«. Poizvedbe s strani odjemalcev, ki uporabljajo ta podatkovni objekt pa se shranijo v seje. Na ta način smo pohitrili aplikacijo, s tem pa tudi razbremenili podatkovni strežnik. Tako je bila izvedena optimizacija na nivoju podatkovnega oziroma aplikacijskega strežnika.

Druga težava se je pojavila na strani odjemalca. Po zahtevku, ki ga dobi strežnik od odjemalca, vrne odjemalcu rezultat zahteve. Odjemalec pa mora rezultat prikazati v brskalniku. Pri aplikaciji, ki omogoča grafičen prikaz urnikov, odjemalec dobi na zahtevo od strežnika ogromno količino HTML kode, ki jo mora interpretirati. V nekaterih »worst-case-scenario« je računalnik klienta potreboval tudi 30 sekund, da je prikazal rezultat zahteve klienta v brskalniku. Po daljšem raziskovanju problema, smo tudi ta problem odpravili. Kot problem se je izkazala funkcija skriptnega jezika jQuery-ja, ki redefinira oziroma spremeni still vsakega objekta oziroma časovne enote, ki nastopa v tem izpisu. Na tem delu smo skriptni jezik oziroma celotno funkcijo umaknili in odzivni čas aplikacije se je možno izboljšal.



Slika 39: Grafa na sliki prikazujeta povprečni odzivni čas od zahteve klienta do časa ko brskalnik prikaže rezultat zahteve

Grafa na sliki prikazujeta povprečni odzivni čas od zahteve klienta do časa ko brskalnik prikaže rezultat zahteve. Levi graf prikazuje izmerjene čase z uporabo funkcije jQuery-ja, desni graf prikazuje izmerjene čase brez uporabe funkcije. Čas je prikazan v sekundah. Za merjenje smo uporabili več različno zmogljivih računalnikov. Za različne scenarije smo uporabili različne parametre pri vходу:

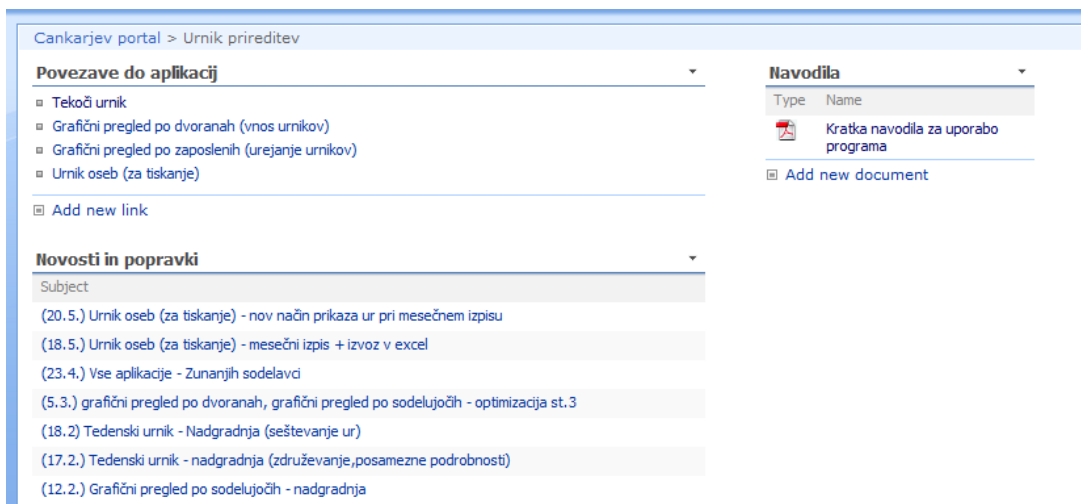
- Tedenski (prikaže tedenski izpis časovnih enot po urah, kar pomeni največ časovnih enot in posledično je računalnik v tem primeru najbolj obremenjen.)
- Prvi dostop (gre za scenarij ko se stran prvič naloži v brskalniku)
- Mesečni (prikaže mesečni izpis časovnih enot po dnevih za izbran mesec)

Za razliko od ostalih pri računalniku št.2 ni bilo velike razlike, saj je računalnik sam po sebi zelo zmogljiv, saj razpolaga z 6MB drugo nivojskega pomnilnika in z taktom procesorja 3.16 GHZ, kar je ključnega pomena.

4.6. Vzdrževanje in nadzor

Po pričetku redne uporabe informacijskega sistema je potrebno zagotoviti tudi njegovo vzdrževanje v smislu odpravljanja napak, odkritih s strani uporabnikov, kot tudi nadgradnja sistema z novimi funkcionalnostmi na nivoju aplikacij.

Seznam vseh povezav do spletnih aplikacij se nahaja na portalu. Portal temelji na microsoftovi tehnologiji Share point, ki omogoča obvladovanje velikih količin podatkov, dokumentov in drugih vsebin na enotnem mestu in predstavlja odlično rešitev za izmenjavo informacij v najrazličnejših oblikah med vsemi udeleženi v procesu. V našem primeru predstavlja dobro orodje za objavo povezav do spletnih aplikacij, popravkov in drugih dokumentov, ki zadevajo uporabnike.



Slika 40: prikaz zaslonske maske tehnologije Share Point

Za nadzor pri dostopu do aplikacij je postavljena preprosta spletna aplikacija, ki omogoča dnevni pregled dostopov do aplikacij s strani uporabnikov. Aplikacija je zgolj informativnega pomena, namenjena razvijalcem in drugim kot vpogled uporabnosti posameznih aplikacij.

Datum:		13.05.2010		Tip dostopa:		Ni pomembno					
ID	Datum	Username	Opis	CN	IP	Tip	Username	Dostopi	Zadnji_dostop		
2775	13.5.2010 6:53:50	jasna.mole	Tekoči urnik	RAKUN	192.168.0.28	Web	janez.kozlevcar	225	22.5.2010 12:08:20		
2776	13.5.2010 6:56:06	jasna.mole	Tekoči urnik	RAKUN	192.168.0.28	Web	mojca.vanelli	43	21.5.2010 16:14:03		
2777	13.5.2010 7:36:13	irena.verdinek	Zagon programa.	ps-irena	192.168.1.77	Win32	jasna.mole	93	21.5.2010 14:48:11		
2778	13.5.2010 7:38:22	irena.verdinek	Zagon programa.	ps-irena	192.168.1.77	Win32	mensur.hodzic	198	21.5.2010 14:28:31		
2779	13.5.2010 7:38:27	irena.verdinek	Izhod iz programa.	ps-irena	192.168.1.77	Win32	bostjan.zakrajsek	229	21.5.2010 14:23:44		
2780	13.5.2010 8:24:36	irena.verdinek	Grafični prikaz dvoran	RAKUN	192.168.0.28	Web	renata.sebez	12	21.5.2010 12:30:51		
2781	13.5.2010 8:29:29	jasna.mole	Tekoči urnik	RAKUN	192.168.0.28	Web	olga.pirc	267	21.5.2010 11:48:41		
2782	13.5.2010 8:51:51	aleksander.avbersek	Tekoči urnik	RAKUN	192.168.0.28	Web	tina.kramberger	212	21.5.2010 9:52:26		
2783	13.5.2010 8:51:54	aleksander.avbersek	Grafični prikaz sodelujočih	RAKUN	192.168.0.28	Web	gregor.biteznik	13	20.5.2010 20:49:34		
2784	13.5.2010 8:53:21	aleksander.avbersek	Zagon programa.	TS-Aleksander	192.168.0.15	Win32	suzana.sheppard	38	20.5.2010 16:12:40		

Slika 41: Prikaz aplikacije za beleženje dostopov do aplikacija

Aplikacija za nadzor dostopov prikazuje seznam dostop po dnevih, z opisom dostopa (čas dostopa, uporabnik, opis dostopa, kraj dostopa in tip dostopa). Omogoča tudi izpis števila dostopov po uporabnikih.

5. Zaključek

Cilj diplomskega dela je bil razvoj in opis celotne informacijske rešitve. Pri razvoju smo uporabili ključne faze razvoja informacijske rešitve. Prva faza, ki narekuje zajem zahtev je bila obsežna in naporna, saj je bilo potrebno povzeti zahteve na različne načine, jih poenotiti in porazdeliti. V fazi analize smo pripravili različne modele za boljše razumevanje celotne problematike. Faza načrtovanja obsega načrt podatkovne baze in definicijo programskih modulov. Sama faza implementacije pa je bila med glavnimi in najzahtevnejšimi. Celoten cikel faz se je večkrat tudi ponovil, saj so se zahteve in ideje tudi sprotoma pojavile.

Kot je razvidno, smo celotno rešitev razvili uspešno. Vsi moduli oziroma vse aplikacije so v glavnem končane, razen aplikacije »Dispozicije«, ki je še v razvoju in še ni v uporabi. Prav tako ni izdelan modul za vpis in vodenje tehnične in ostale opreme s strani vodij tehnične izvedbe. Tekom časa pa bodo potrebne posodobitve in nadgradnje ostalih aplikacij, saj se nove ideje in zamisli tudi sprotoma pojavijo. Pri opisu razvoja in sami implementaciji so nam bila v pomoč različna orodja.

Ideja o implementaciji opisane rešitve je bila prisotna veliko časa. Prav tako je tudi sam razvoj rešitve potekal že pred začetkom pisanja diplomskega dela. Cilj projekta je bil izdelati integriran sistem oziroma informacijsko rešitev, ki bo povezala celoten proces oziroma življensko pot projekta v cankarjevem domu in omogočala dober vpogled razporeda rezerviranih in potrjenih, prostorskih in drugih virov v cankarjevem domu.

Pri razvoju opisane rešitve sem pridobil veliko bogatih izkušenj. Utrdil sem znanje programiranja in razvoja informacijskih rešitev, ki sem ga pridobil na fakulteti. Utrdil sem znanje programiranja v skriptnih jezikih, v glavnem pa v objektnem jeziku C#. Pridobil sem veliko znanja za delo z orodjem Visual Studio 2008. Pri načrtovanju in postavitvi podatkovne baze, ter pisanja poizvedb SQL, sem pridobil izkušnje s področja podatkovnih baz. Pri razvoju celotnega projekta pa sem pridobil bogate izkušnje v smislu sodelovanja, dogovarjanja in tudi nesodelovanja z udeleženi v okolju. Postavljen sem bil tudi v vlogo pri sprejemanju pomembnih odločitev tekom razvoja in dobil sem vpogled kako dejansko v praksi poteka razvoj informacijske rešitve.

Slike

Slika 1: Zaporedni oziroma slapovni model	5
Slika 2: iterativni model.....	6
Slika 3: Prototipni model.....	7
Slika 4: Inkrementalni model.....	8
Slika 5: Prikaz življenskih modelov z kriteriji	11
Slika 6: Prikaz podatkovnega modela - konceptualni diagaram.....	19
Slika 7: Procesni diagram - eeEpc diagram. Prikaz celotnega procesa.....	20
Slika 8: UML diagram - diagram primerov uporabe.....	30
Slika 9: Prototip št.1 (Rezervacije) - obrazec za vnos rezervacije.....	31
Slika 10: Prototip št.1 (Rezervacije) - seznam rezervacij.....	32
Slika 11: Prototip št.1 (Rezervacije) - seznam organizatorjev	32
Slika 12:Prototip št.2 (Projekti) - glavna forma	33
Slika 13:Prototip št.2 (Projekti) - seznam projektov	33
Slika 14:Prototip št.2 (Projekti) - projekt.....	34
Slika 15:Prototip št.2 (Projekti) - seznam urnikov.....	34
Slika 16: Prototip št.3 (Dispozicije) - seznam projektov z dispozicijami.....	35
Slika 17: Prototip št.3 (Dispozicije) - zaslonska maska za kreiranje dispozicije s strani vodij projektov.....	35
Slika 18: Prototip št.3 (Dispozicije) - primer dispozicije za vnos zahtev.....	35
Slika 19: Prototip št.4 (Dispozicije) - zaslonska maska za kreiranje dispozicije s strani vodij izvedbe ..	36
Slika 20: Prototip št.4 (Dispozicije) - primer dispozicije za potrjevanje le-te s strani vodij izvedbe	36
Slika 21: Prototip št.4 (Urniki) - prikaz seznama urnikov in seznama oseb za določanje urnikov izvedbe	37
Slika 22: Prototip št.4 (Urniki) - Seznam urnikov izvedbe	37
Slika 23: Dvonivojska arhitektura.....	38
Slika 24: Podatkovni model - logični model	40
Slika 25: Predstavitev programskih modulov aplikacije "Rezervacije" s strukturnim diagramom	42
Slika 26: Predstavitev programskih modulov aplikacije "Projekti" s strukturnim diagramom	43
Slika 27: Predstavitev programskih modulov aplikacije "Dispozicije" s strukturnim diagramom.....	44
Slika 28: Predstavitev programskih modulov aplikacije "Urniki" s strukturnim diagramom.....	45
Slika 29: Prikaz delne zaslonske maske aplikacije »Rezervacije« za vnos nove rezervacije.....	47
Slika 30: Prikaz zaslonske maske aplikacije »Projekti« za prikaz podatkov o projektu.....	48
Slika 31: prikaz zaslonske maske aplikacije »Projekti« za prikaz podatkov o urniku.....	48
Slika 32: prikaz zaslonske maske aplikacije »Dispozicije« za urejanje dispozicije z vidika vodje projekta	49
Slika 33: prikaz zaslonske maske pri grafičnem izpisu oziroma prikazu urnikov projektov aplikacije »Urniki«	50
Slika 34: prikaz zaslonske maske aplikacije »urniki« pri grafičnem izpisu urnikov projektov (dnevni prikaz).....	50
Slika 35: prikaz dela zaslonske maske aplikacije »urniki«. Primer izbire urnikov za vnos urnikov izvedbenih delavcev	51
Slika 36: zaslonska maska aplikacije »Urniki«. Prikazuje rezultat izbire urnikov projektov z namenom vnosa urnikov izvedbe.....	51

Slika 37: prikaz zaslonske maske pri grafičnem izpisu oziroma prikazu urnikov izvedbe, aplikacije »Urniki«	52
Slika 38: prikaz zaslonske maske aplikacije »urniki« pri grafičnem izpisu urnikov izvedbe(dnevni prikaz)	52
Slika 39: Grafa na sliki prikazujeta povprečni odzivni čas od zahteve klienta do časa ko brskalnik prikaže rezultat zahteve	54
Slika 40: prikaz zaslonske maske tehnologije Share Point	55
Slika 41: Prikaz aplikacije za beleženje dostopov do aplikacija	55

Tabele

Tabela 1: Opis programskih modulov za vmesnik »Rezervacije«	42
Tabela 2: Opis programskih modulov za vmesnik »Projekti«	43
Tabela 3: Opis programskih modulov za vmesnik »Dispozicije«	44
Tabela 4: Opis programskih modulov za vmesnik »Urniki«	45

Literatura

- [1] (2010) Asp.net forum. Dostopno na:
<http://forums.asp.net/>
- [2] (2010) EMRIS - Enotna metodologija razvoja informacijskih sistemov. Dostopno na:
<http://www2.gov.si/mju/emris.nsf/>
- [3] (2010) Inforamcijski sistemi in metodologije razvoja. Dostopno na:
http://www.student-info.net/sis-mapa/skupina_doc/fri/knjiznica_datoteke/638500_emris_uvod.pdf
- [4] Microsoft Corporation, »*Introdution to Programming Microsoft .NET Applications with Microsoft Visual Studio 2005*«, 2007
- [5] (2010) Programska oprema Edraw Max. Dostopno na:
<http://www.edrawsoft.com/EPC-Diagrams.php>
- [6] (2010) Programska oprema Visual Studio 2008 - dokumentacija. Dostopno na:
[http://msdn.microsoft.com/en-us/library/bb166217\(v=VS.100\).aspx](http://msdn.microsoft.com/en-us/library/bb166217(v=VS.100).aspx)
- [7] (2010) Razvoj informacijskih sistemov. Dostopno na:
http://bajecm.fri.uni-lj.si/pg_ris.htm
- [8] (2010) Share Point 2007 tehnologija. Dostopno na:
<http://sharepoint.microsoft.com/en-us/Pages/default.aspx>
- [9] (2010) Tri-nivojska arhitektura. Dostopno na:
http://colos.fri.uni-lj.si/ERI/RACUNALNISTVO/PODATKOVNE_BAZE/trinivojska_arhitektura.html