

**UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO**

Marko Klopčič

**RAZVOJ SISTEMA ZA ZAJEM REGISTRACIJ
DELOVNEGA ČASA**

DIPLOMSKO DELO UNIVERZITETNEGA ŠTUDIJA

Mentor: prof. dr. Marjan Krisper

Ljubljana, 2008

ZAHVALA

Zahvaljujem se mentorju prof. dr. Marjanu Krisperju za njegovo strokovno pomoč in nasvete pri izdelavi diplomskega dela.

Zahvaljujem se Simonu in Davidu Sakelšku iz podjetja Špica International d. o. o., ki sta mi pomagala pri razumevanju delovanja tehnologije ter njeni integraciji v sistem.

Zahvaljujem se svoji družini za podporo v času mojega študija.

Zahvaljujem se lektorici Katji Pirnat ter vsem, ki so kakorkoli pripomogli k temu, da sem študij ter diplomsko delo uspešno dokončal.

Kazalo

Povzetek	1
Abstract.....	2
1 Uvod	3
2 Tehnologija komunikacije v bližnjem polju	4
2.1 Kaj je komunikacija v bližnjem polju	4
2.2 Delovanje	4
2.3 NFC Forum	4
2.3.1 O organizaciji NFC Forum.....	4
2.3.2 Smernice razvoja tehnologije KBP	5
2.4 Načini komunikacije.....	6
2.5 Varnost tehnologije KBP	7
2.5.1 Prisluskovanje.....	7
2.5.2 Uničenje podatkov	7
1.5.3 Spreminjanje podatkov.....	7
2.5.4 Vstavljanje podatkov.....	7
2.5.5 Napad z vmesnim členom	8
2.5.6 Rešitev z vzpostavitvijo varnega kanala.....	8
2.6 Razširjenost tehnologije med proizvajalci GSM aparatov	8
3 Registracija delovnega časa.....	9
3.1 Sistem za registracijo delovnega časa Time&Space.....	9
3.2 Standardni pristop k zajemu registracije delovnega časa.....	9
4 Integracijska rešitev registracije delovnega časa z uporabo tehnologije komunikacije v bližnjem polju	11
4.1 Ideja rešitve	11
4.2 Umestitev rešitve v sistem Time&Space	12
5 Metodologija objektnega razvoja ter jezik UML	13
5.1 Kaj je metodologija.....	13
5.2 Objektna tehnologija.....	13
5.3 Orodja CASE.....	14
5.4 Izbira procesnega modela.....	14

5.5	Objektni proces razvoja	15
5.5.1	Dinamični vidik	15
5.5.2	Statični vidik.....	16
5.6	Jezik UML	17
6	Razvoj sistema za zajem registracij z uporabo objektne metodologije	18
6.1	Zajemanje zahtev.....	18
6.1.1	Opredelevitev zahtev	18
6.1.2	Izdelava diagramov primerov uporabe.....	21
6.1.3	Določanje prioritete primerov uporabe.....	24
6.1.4	Izgradnja prototipa uporabniškega vmesnika in dopolnitev primerov uporabe.....	25
6.2	Analiza.....	27
6.2.1	Izdelava razrednega diagrama obravnavane domene.....	27
6.2.2	Analiza primerov uporabe	30
6.2.3	Razvoj razrednega diagrama analize.....	32
6.2.4	Specificiranje operacij sistema	33
6.2.5	Pregled modela analize.....	33
6.3	Arhitekturno načrtovanje	34
6.3.1	Organiziranje razredov v komponente	34
6.3.2	Izbira arhitekturnih stilov	35
6.3.3	Izgradnja logične arhitekture	35
6.3.4	Izgradnja fizične arhitekture.....	36
6.4	Načrtovanje objektov	37
6.4.1	Razvoj razrednega diagrama načrtovanja.....	37
6.4.2	Načrtovanje algoritmov.....	38
6.4.3	Določanje dostopa in optimizacija.....	40
6.4.4	Določanje začetnega stanja objektov ter pregled skladnosti	41
6.5	Implementacija	42
6.5.1	Generiranje tehnoloških razredov	42
6.5.2	Razvoj kode razredov.....	42
6.5.3	Analiza zmogljivosti	43
6.5.4	Pregled kode	44
6.6	Testiranje	45
6.6.1	Planiranje testiranja.....	45

6.6.2	Oblikovanje testnih primerov	46
6.6.3	Implementacija testnih primerov	47
6.6.4	Izvajanje testiranja	47
6.6.5	Vrednotenje testiranja	48
7	Zaključek	49
8	Priloge.....	50
8.1	Seznam slik	50
8.2	Seznam preglednic.....	50
9	Viri in literatura	51
	Izjava o avtorstvu.....	52

Slovar kratic

Bluetooth	–	brezžična tehnologija
ECMA	–	organizacija za standardizacijo na področju telekomunikacij
EMRIS	–	enotna metodologija razvoja programske opreme
ETSI	–	Evropski telekomunikacijski inštitut za standardizacijo
Event Collector	–	strežnik za zajem registracij
GPRS	–	paketni prenos podatkov
HSPDA	–	protokol mobilne telekomunikacije (tretja generacija)
ISO	–	Mednarodna organizacija za standardizacijo
IEC	–	Organizacija za standardizacijo na področju elektronike, elektrotehnike in drugih sorodnih tehnologij
IMEI	–	Mednarodna identična številka mobilne naprave
Java ME	–	razvojno okolje za izdelavo mobilnih aplikacij v jeziku Java
KBP	–	komunikacija v bližnjem polju (Near Field Communication - NFC)
MS SQL	–	strežnik podjetja Microsoft
naprava KBP	–	naprava, ki nudi podporo tehnologiji KBP (GSM aparat ...)
NXP	–	veliko podjetje, katerega ustanovitelj je podjetje Philips
Oznaka KBP	–	pasivna naprava KBP; spominska enota, ki lahko drži podatke aktivna naprava KBP, do tega podatka lahko dostopa (NFC tag)
registracija	–	zajem podatkov o lokaciji ter času določene osebe
RFID	–	metoda avtomatske identifikacije
Time&Space	–	produkt podjetja Špica International d. o. o. za zajem ter obdelavo registracij delovnega časa
točka	–	lokacija v prostoru, katero želimo spremljati oziroma nadzorovati v sistemu
UML	–	Unified modeling Language – jezik za objektno modeliranje
UMTS	–	univerzalni mobilni telekomunikacijski sistem (tretja generacija)
URI, URL	–	povezava na določen naslov na internetu
WI-FI	–	brezžična tehnologija
.NET	–	skupek komponent podjetja Microsoft

Povzetek

V diplomskem delu je predstavljen informacijski sistem za zajem registracije delovnega časa z uporabo tehnologije komunikacije v bližnjem polju. Podrobno je opisana sama tehnologija KBP, predstavljene so njene prednosti, lastnosti ter napovedi za njeno uporabnost v prihodnosti. Poleg tehnologije KBP sta predstavljena tudi princip registracije delovnega časa in trenutne rešitve na tem področju.

V nadaljevanju je podana ideja informacijske rešitve za zajem registracije z uporabo tehnologije KBP. Predstavljene so prednosti rešitve, njene omejitve ter zahteve. Sledi predstavitev objektne metodologije razvoja informacijskega sistema ter razvoj rešitve. Razvoj rešitve poteka po aktivnostih objektne metodologije, katere so sestavljene iz več opravil. V diplomskem delu so predstavljeni ključni diagrami ter opisi, ki nastanejo kot izdelek posamezne aktivnosti in vodijo do izdelave informacijskega sistema. Diagramska tehnika, ki se pri tem uporablja, je jezik za objektno modeliranje UML in je implementirana v orodju PowerDesigner.

V zaključku diplomske naloge so predstavljene možne izboljšave zgrajenega informacijskega sistema, njegove možnosti za uporabo v realnosti, podani pa so tudi nekateri drugi možni pristopi k rešitvi zajema registracije z uporabo tehnologije KBP.

Ključne besede: registracija delovnega časa, tehnologija KBP, UML, objekt, razred, diagram, registracija.

Abstract

The following work presents information system for handling time attendance using NFC – Near-Field Communication. The first stage describes NFC technology, its details, benefits and predictions for the future use. The second stage describes the concept of time attendance and current available solutions.

The main part of the work describes a solution of information system for handling time attendance using NFC technology. It presents benefits of the solution, its obstacles and requirements. The introduction into the object-oriented methodology and development of the solution follow then. The development is divided into six activities that capture the complete process of the realized information system. The following work presents all important diagrams and descriptions which are elaborated in several activities. They lead to the complete solution. Diagrams are made with UML – Unified Modeling Language and they are implemented in PowerDesigner.

In the conclusion of the work, some possible future improvements of the information system are mentioned and the estimation about using the solution in real world is given. Also some other possible solutions about integrating time attendance with near-field communication are mentioned.

Key words: Time attendance, NFC, UML, object, Class, diagram, registration.

1 Uvod

Sprotno beleženje registracij (prihodov, odhodov, ...) na delovnem mestu poteka že od nastanka prvih podjetij. Sam način beleženja podatkov pa se je skozi čas precej spreminjal. V preteklosti se je beleženje registracij izvajalo na različne načine. Beleženje je potekalo z ročnimi vpisi časa registracije vsakega posameznika ali z uporabo avtomata, ki je čas registracije zabeležil na uporabnikov karton. Z informatizacijo podjetij se je tudi na tem področju naredil korak naprej. Postopek registracije je v današnjih časih postal precej hitrejši, enostavnejši ter prijaznejši do samega uporabnika. Sistemi, ki so namenjeni zajemu ter upravljanju z registracijami, ponujajo hitra ter vsebinsko bogata poročila tako za posameznika, kot za skupino. Najbolj pogost način registracije v današnjem času je tako postal koncept z uporabo kartice. Uporabnik s svojo kartico pride do terminala, kjer se nato z njo registrira. Zelo primeren način za zajem registracije je tudi registracija z zajemom prstnega odtisa uporabnika ter registracija s prepoznavanjem obraza. Oba načina sta zelo napredna in kakovostna, poleg tega pa uporabnik za registracijo praktično ne potrebuje ničesar. Razloga za manjšo razširjenost takega načina registracije sta predvsem v ceni rešitve ter nasprotovanju Evropske unije, katera nasprotuje shranjevanju biometričnih podatkov.

Podjetje Špica International d. o. o. že več let ponuja informacijske rešitve s področja kontrole pristopa ter registracije delovnega časa. Produkt Time&Space zajema velik nabor orodij, aplikacij, terminalov, kartic ter drugih komponent, ki sodijo v koncept zajema ter obdelave registracij. Prav tako nudi podporo raznim komponentam za zajem biometričnih podatkov, kot so prstni odtis, obrazna mimika ter očesna šarenica. Podjetje je realiziralo tudi preprosto rešitev za registracijo delovnega časa z uporabo tehnologije komunikacije v bližnjem polju, s katero so prikazali, kako je moč to tehnologijo izkoristiti tudi v ta namen. Za to idejo je podjetje prejelo nagrado za inovacijo.

V pričujoči diplomski nalogi bom predstavil ter z objektivno metodologijo izdelal informacijsko rešitev, ki bo prav tako uporabljala tehnologijo KBP, poleg tega bo tudi omogočala interakcijo uporabnika s samim sistemom. Predstavljena rešitev bo tako uporabniku ponujala precej večji nabor možnosti pri samem zajemu registracije, kar dani rešitvi poleg inovativnosti daje še dodatno dodano vrednost.

2 Tehnologija komunikacije v bližnjem polju

2.1 Kaj je komunikacija v bližnjem polju

Komunikacija v bližnjem polju (v nadaljevanju KBP) ali NFC (Near Field Communication) je tehnologija brezžične povezave. Le-ta predstavlja kombinacijo različnih brezkontaktnih identifikacijskih in omrežnih tehnologij. KBP v osnovi omogoča enostavno brezkontaktno komunikacijo kratkega dometa med različnimi elektronskimi napravami. Je izjemno enostavna za uporabo, hitra ter relativno varna. Njeno uporabnost pričakujejo tudi v aplikacijah, ki se ukvarjajo z elektronskim bančništvom.

Tehnološka osnova temelji na standardih ISO, ETSI ter ECMA. NFC je bil leta 2003 potrjen s strani ISO ter IEC organizacije kot uraden standard ISO/IEC 18092. Leto kasneje so podjetja NXP, Sony ter Nokia ustanovili NFC Forum, ki skrbi za nadaljnji razvoj tehnologije ter standardov. [1]

2.2 Delovanje

KBP deluje v bližnjem magnetnem polju pri frekvenci 13.56 MHz in lahko prenaša podatke s hitrostjo do 424 kbit/s. Komunikacija med dvema KBP napravama se vzpostavi, ko se le-ti nahajata na razdalji nekaj centimetrov. Že rahel zamah ali dotik omogoči, da se podatki med napravama začno izmenjevati.

Katerakoli od NFC naprav lahko nudi združljivost s tehnologijami kot so Wi-Fi, Bluetooth ter paketna GSM omrežja (GPRS, UMTS, HSPDA). NFC tako lahko deluje kot začetna ali končna točka integracije omrežij dolgega, srednjega ali kratkega dometa. [1, 2]

2.3 NFC Forum

2.3.1 O organizaciji NFC Forum

NFC Forum je neodvisna organizacija, ustanovljena leta 2004, katere namen je pospeševati uporabo tehnologije komunikacije v bližnjem polju. Člani NFC Foruma so proizvajalci, prodajalci, ponudniki storitev in vsa ostala podjetja, ki se tako ali drugače srečujejo s tehnologijo KBP v mobilnih telefonih, računalnikih in ostalih elektronskih napravah široke potrošnje (MasterCard, Nokia, NXP, Microsoft, ...). Zelo pomembna dejavnost NFC Foruma je tudi izobraževanje končnih uporabnikov, saj se zavedajo, da bo morda strah pred pomanjkanjem varnosti glavni zaviralec vpeljave tehnologije, zato naj bi v obliki konferenc in člankov ljudem omogočali lahek dostop do zanesljivih objektivnih informacij. [11]

2.3.2 Smernice razvoja tehnologije KBP

Od ustanovitve do danes je NFC Forum razvil devet specifikacij, ki so bile tako rekoč nujne za uglasen nadaljnji razvoj tehnologije KBP. Prvih pet specifikacij je vezanih na format podatkovnih zapisov [12]. Te so naslednje:

- **Format za izmenjavo podatkov KBP (FIP)**
Specifikacija definira osnovni format sporočil, ki se prenašajo med dvema napravama KBP. Je primarna specifikacija, katere lastnosti podedujejo vse ostale specifikacije formatov. Definira strukturo formata za izmenjavo podatkov ter glavo sporočila, iz katere je moč razbrati lastnosti samega zapisa. Konkretni tipi zapisov so definirani v DTZ specifikaciji.
- **Definicija tipov zapisa KBP (DTZ)**
DTZ definira format in pravila za zapis standardnih tipov zapisov, ki se uporabljajo v NFC Forum aplikacijah. Le-te temeljijo na NDEF formatu. Bistvena lastnost DTZ je odprtost do razvoja novih tipov zapisov, ki se bodo lahko uporabljali v novih aplikacijah. Specifikacija v osnovi tipe zapisov razdeli na NFC Forum poznane tipe, ter na tipe, ki jih po svojih potrebah definirajo posamezne organizacije.
- **Definicija tekstovnih tipov zapisa KBP (TEKST DTZ)**
Tekst DTZ definira pravila za zapis tekstovnih nizov v več jezikih z uporabo DTZ pravil in FIP formata zapisa.
- **Definicija URL tipov zapisa KBP (URL DTZ)**
Eden izmed znanih NFC Forum tipov se imenuje pametni plakat («Smart Poster»). Le-ta omogoča zapis URL, SMS ali telefonske številke na NFC oznako ali njihov prenos med napravami. Temelji na vseh ostalih prej definiranih specifikacijah, saj uporablja osnovni FIP format, omogoča pa tako zapis URI identifikatorja, kot običajnega teksta. Namenjen je uporabi v aplikacijah tako imenovanih pametnih plakatov, kjer je na plakatu oznaka KBP, preko nje pa lahko uporabnik dobi dodatne informacije o vsebini, ki jo plakat opisuje. Pametni plakat je trenutno tudi edina aplikacija, ki jo je definirala NFC Forum.

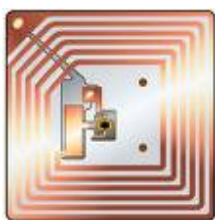
Preostale štiri specifikacije definirajo tehnične lastnosti oznak KBP, ki zagotavljajo združljivost med različnimi proizvajalci oznak KBP ter naprav KBP:

- **Oznaka KBP - Tip 1**
Oznaka prvega tipa temelji na ISO standardu 14443A, ki definira bralno/pisalno oznako (uporabnik jo lahko tudi nastavi samo na bralno) s 96 do 2048 bajti uporabniškega spomina ter hitrostjo komunikacije 106 kbit/s.
- **Oznaka KBP - Tip 2**
Tudi drugi tip temelji na ISO 14443A. Razlika glede na tip 1 je, da imajo lahko oznake tipa 2 tudi samo 48 bajtov spomina.
- **Oznaka KBP - Tip 3**

Oznaka tretjega tipa temelji na japonskem industrijskem standardu X 6319-4, bolj poznanem kot FeliCa. Oznake se pri proizvodnji nastavijo bodisi na bralno/pisalni ali pa na bralni način. Zgornja meja spomina je 1 MB, možni pa sta dve komunikacijski hitrosti: 212 ter 424 kbit/s.

- **Oznaka KBP - Tip 4**

Oznaka tipa 4 je v popolnosti združljiva z ISO standardoma 14443A in 14443B. Bralno/pisalne zmožnosti oznake so nastavljene pri proizvodnji, največja velikost spomina je 32 kB. Komunikacijska hitrost je 424 kbit/s.



Slika 1: Oznaka KBP

2.4 Načini komunikacije

Tehnološka osnova komunikacije v bližnjem polju je RFID komunikacija na podlagi magnetne indukcije. Anteni oznake ali naprave KBP, ki deluje kot bralna narava, sta torej zanki, ki kadar se nahajata dovolj blizu skupaj, tvorita magnetni sklop. Domet komunikacije v bližnjem polju je do približno 20 cm. Tipično se uporablja na razdaljah, manjših od 10 cm. S standardi podprte hitrosti so 106, 212 in 424 kbit/s. [2, 4] Obstajata dva načina komunikacije:

- Pasivna komunikacija
Komunikacijo začne aktivna naprava. Pasivna naprava modulira signal, ki ga oddaja aktivna naprava.
- Aktivna komunikacija
Obe napravi izmenično oddajata svoja signala, kar pomeni, da imata lastni vir napajanja.

2.5 Varnost tehnologije KBP

Zagotovljena določena mera varnosti je za tehnologijo, kot je komunikacija v bližnjem polju življenjskega pomena, saj se lahko uporablja tudi v aplikacijah z denarnimi transakcijami. Kljub temu, da je pričakovan domet uporabe KBP nekaj centimetrov, obstaja cela vrsta napadov, na katere naprave KBP niso imune same po sebi. [5, 14]

V nadaljevanju si bomo pogledali nekaj možnih načinov neavtoriziranih vdorov v komunikacijo v bližnjem polju:

2.5.1 Prisluškovanje

Komunikacija v bližnjem polju ne nudi nikakršne zaščite proti prisluškovanju (izjema je le izjemno kratek domet, kar pa ne predstavlja neke zanesljivosti). Aktivna naprava oddaja močnejši signal, zato je teoretična razdalja prisluškovanja lahko tudi 10 metrov. Zanesljiva rešitev za preprečevanje prisluškovanja je vzpostavitev varnega kanala.

2.5.2 Uničenje podatkov

S primerno opremo lahko posameznik izvede obliko napada, s katerim uniči del podatkov na eni od NFC naprav, ki sodeluje pri komunikaciji. S tem lahko prepreči izvedbo storitve. Tak način napada je izjemno enostavno izvesti, saj se potrebno opremo lahko kupi preko interneta. Zaščita pred takim napadom ni možna. Lahko pa tak napad zaznamo, saj je napetost v polju pri teh napadih občutno večja od napetostih pri komunikaciji v bližnjem polju. Poleg tega te naprave povzročajo trk, ki jo naprava KBP lahko zazna.

1.5.3 Spreminjanje podatkov

Pri tej obliki napada želi napadalec doseči, da prejemnik dobi veljavne, vendar spremenjene podatke. Tak napad je izvesti precej težje. Pri Millerjevem kodiranju signala je tak napad možen samo za določene bite, nasprotno pa je pri Manchester kodiranju signala napad vedno možen. Poznamo več vrst obramb pred takšno obliko napada. Prva je uporaba 106 kHz frekvence. S tem onemogočimo zunanji napravi, da spreminja podatke. Slabost te rešitve je v tem, da komunikacija pri taki frekvenci lahko poteka samo takrat, kadar gre za aktivno komunikacijo. Druga rešitev, ki se ponuja, je, da aktivna naprava preverja polje ter poskuša zaznati napadalca. Tretja, verjetno najboljša, rešitev pa je vpeljava varnega kanala.

2.5.4 Vstavljanje podatkov

Napadalec pri tej obliki napada pošlje podatke čakajoči napravi še preden ta dobi odgovor od prave naprave. Tak napad je možno izvesti takrat, ko naprava, ki pošilja odgovor potrebuje več časa. V primeru, da dasta hkrati odgovor tako napadalec kot pravi oddajnik, pride do neveljavnih podatkov. Zopet se ponujajo tri možne rešitve. Prva je takojšnji odgovor prave naprave. V

najslabšem primeru lahko pride edino do trka z napadalcem. Druga rešitev je, da naprava, ki pripravlja podatke za pošiljanje posluša, kaj se dogaja v komunikacijskem kanalu. Ko zazna, da nekdo odgovarja namesto nje, se primerno odzove. Tretja, najprimernejša, je z vpeljavo varnega kanala.

2.5.5 Napad z vmesnim členom

Pri tem napadu se v komunikacijo vmeša napadalec tako, da vsa sporočila potujejo preko njega. V primeru, da se napravi, ki sta skušali vzpostaviti komunikacijo, tega ne zavedata, lahko ta oseba poljubno spreminja podatke, ki se pošiljajo. Pri tehnologiji komunikacije v bližnjem polju je tak napad težko izvedljiv. Rešitev za obrambo pred takim napadom je vzpostavitev pasivne povezave, pri kateri se pričakuje, da je naenkrat aktivna samo ena naprava. Naprava potem poskuša zaznati, ali je v bližini še katera druga naprava. Le-te namreč ne bi smelo biti.

2.5.6 Rešitev z vzpostavitvijo varnega kanala

Če privzamemo dejstvo, da je pri komunikaciji v bližnjem polju napad z vmesnim členom praktično nemogoč, lahko vidimo, da je najboljša rešitev za ostale napade vzpostavitev varnega kanala. Najbolj primerna rešitev je z uporabo standardnega Diffie-Hellmanovega protokola. Pri tem algoritmu poteka komunikacija tako, da oba pošiljatelja zakodirata svoja sporočila s ključi, po prispetju sporočila pa jih znata dekodirati le onadva.

2.6 Razširjenost tehnologije med proizvajalci GSM aparatov

Trenutna podpora komunikaciji v bližnjem polju s strani proizvajalcev GSM aparatov še ni na visokem nivoju, se pa za vgraditev te tehnologije v aparate odloča vedno več ponudnikov. Trenutno je najbolj razširjen telefon s podporo KBP, izdelek podjetja Nokia, model 6131, ki ga je možno dobiti tudi na slovenskem trgu. Med bolj znanimi ponudniki sta tudi Panasonic ter Benq. V prihodnosti pričakujemo aparate s podporo za KBP tudi od ostalih ponudnikov, kot sta Sony Ericsson ter Siemens.



Slika 2: GSM aparat s podporo KBP – Nokia 6131

3 Registracija delovnega časa

Vsako podjetje, ki ima željo po izboljševanju organizacije delovnih procesov ter s tem posledično zmanjševanje stroškov, potrebuje sistem za registracijo delovnega časa. Cilj registracije delovnega časa je natančno beleženje prihodov ter odhodov zaposlenih na oziroma iz delovnega mesta ter še mnogo drugih podatkov, kot so na primer koriščenje dopusta, bolniške, čas, porabljen za malico, nadure, kršitve, privatne odsotnosti ter drugo. Z množico zbranih podatkov ter dobro organizacijo dela, lahko registracija delovnega časa prinese veliko prednosti pri upravljanju s človeškimi viri ter posledično tudi velike prihranke.

3.1 Sistem za registracijo delovnega časa Time&Space

Podjetje Špica International je razvilo produkt za registracijo delovnega časa ter kontrolo pristopa – Time&Space. Time&Space je integriran sistem za obvladovanje delovnega časa in prostora. Celoten sistem sestavlja družina programov za upravljanje ter pregledovanje, sistemi terminalov, ter kartice.

Informacije o delovnem času ter gibanju ljudi skozi poslovne prostore prihajajo iz registracijskih terminalov. Uporabniki registrirajo svoj dogodek (prihod, odhod, malico, ...) na terminalu za zbiranje registracij. Identificira se lahko na več načinov (kartica, biometrija, ...). Registrirani dogodki se nato stekajo v skupno podatkovno bazo, do katere imajo nato dostop vse aplikacije, ki spadajo v Time&Space paket.

Za prenos podatkov iz terminalov v podatkovno bazo skrbi strežnik Event Collector. Poleg te primarne funkcije opravlja še vrsto drugih funkcij, kot so na primer beleženje napak pri komunikaciji, terminalom pošilja podatke, ki so bili vneseni v aplikacijah sistema Time&Space ter še mnogo drugega. Če povzamemo, je strežnik Event Collector most med strojno ter programsko opremo sistema Time&Space.

Informacijski sistem Time&Space omogoča kontrolo pristopa, krmiljenje alarmnih naprav, sproten vpogled v evidenco delovnega časa ter evidenco obiskovalcev. Sistem je modularno zasnovan ter s tem prinaša sodoben pristop k reševanju kompleksnih problemov.

3.2 Standardni pristop k zajemu registracije delovnega časa

Poglejmo si primer klasičnega pristopa k zajemu registracije delovnega časa. Oseba pristopi do vhoda, kjer je nameščen terminal za zajem registracije delovnega časa. Oseba se z identifikatorjem (kartica, prstni odtis, očesna šarenica, ...) identificira na terminalu. Če sistem prepozna osebo, zabeleži njeno registracijo ter ji odpre vrata. V nasprotnem primeru zavrne registracijo, izpiše obvestilo na terminalu ter vrat ne odpre.

Tak primer registracije je najbolj pogost tako v svetu kot pri nas. Prednost take rešitve je zanesljivost ter učinkovitost. Cena takega sistema zajema tri elemente. Identifikator (kartica), bralna naprava (terminal) ter sistem za upravljanje ter evidenco. [3, 8, 13]

4 Integracijska rešitev registracije delovnega časa z uporabo tehnologije komunikacije v bližnjem polju

Sedaj, ko podrobneje poznamo komunikacijo v bližnjem polju ter sistem registracije delovnega časa, si pogledajmo idejo informacijske rešitve za zajem registracij z uporabo tehnologije KBP.

Predpostavim, da ima v vseh razvitih družbah današnjega sveta GSM aparat tako rekoč vsakdo. Uporaba GSM aparatov je v nekaj letih izjemno narasla, s tem pa se je tudi cena naprave zelo znižala. Tehnologija KBP je trenutno podprta pri le nekaj aparatih proizvajalcev, vendar je glede na optimistične napovedi pričakovati porast tehnologije v aparatih. Cena aparatov s podporo komunikaciji v bližnjem polju bo tako dosegljiva za posameznika kot za podjetje.

4.1 Ideja rešitve

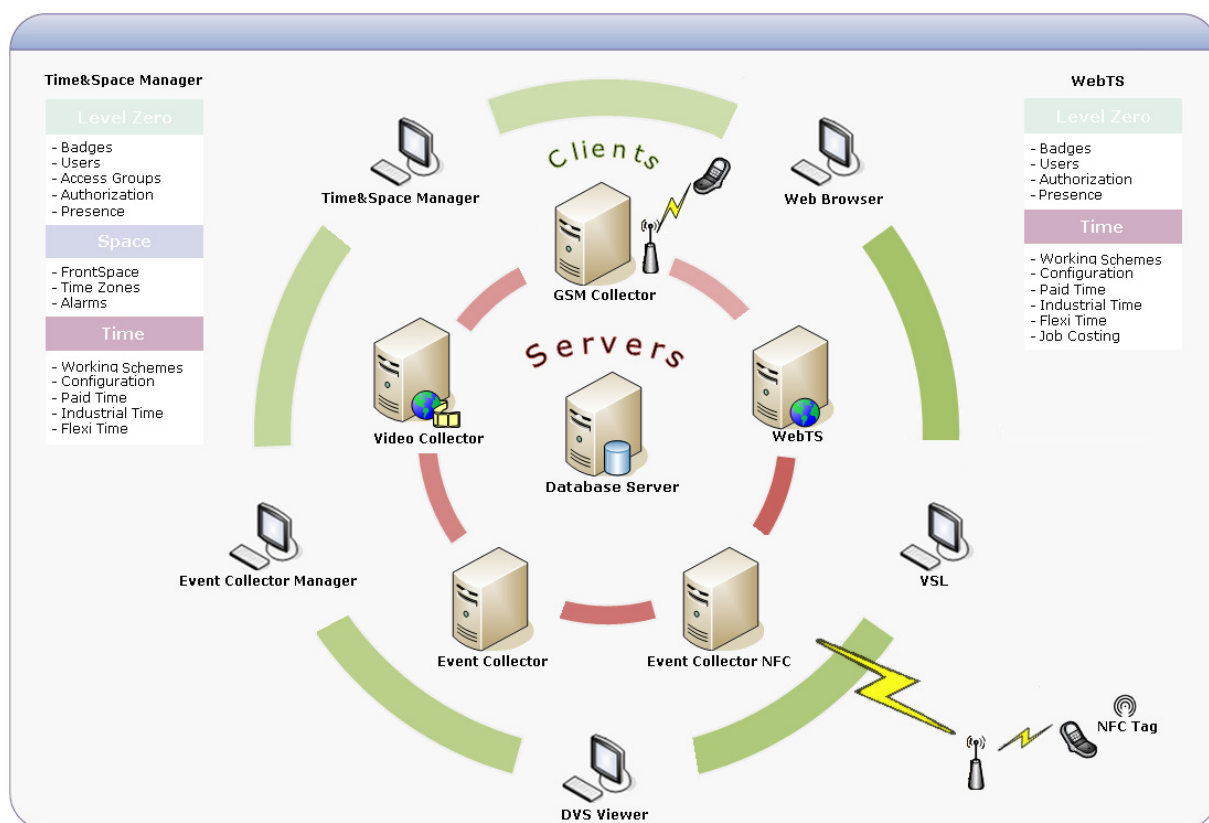
Ideja rešitve je obrniti trenuten potek registracije delovnega časa. Namesto postavitve terminalov za zbiranje ter pošiljanje dogodkov na vseh prehodih, kjer želimo beležiti podatke, sedaj namestimo oznake KBP. Registracijo izvedemo preko GSM aparata, ki podpira komunikacijo v bližnjem polju. Aparat ob zaznavi oznake KBP zažene aplikacijo, s katero enostavno izvedemo registracijo.

Aplikacija preko paketnega prenosa pošlje podatke o identifikaciji (v primeru take rešitve je to številka telefona ali pa številka GSM aparata), točki registracije ter tipu dogodka, ki ga želimo registrirati. Podatki se pošljejo strežniku KBP, ki registracijo shrani v podatkovno bazo ter po potrebi sproži še kakšno drugo operacijo (povratna informacija o saldu ur, dopusta, ...).

Rešitev je elegantna, tehnološko zelo napredna, poleg tega se podjetja tako lahko izognejo terminalom na prehodih, saj le-ti pogosto kvarijo zunanji izgled objekta. Oznake KBP so izjemno majhne in jih lahko napravimo skoraj neopazne. Pri podjetjih, ki želijo nadzirati večje število prehodov, je velika prednost tudi cena take rešitve. Namesto večjega števila terminalov, ki so relativno dragi, sedaj potrebujejo le večje število oznak KBP, ki pa so zaradi svoje enostavnosti izredno poceni. Poleg tega se s to rešitvijo podjetja izognejo tudi nakupu kartic, saj je identifikacija skrita že v samem telefonu.

4.2 Umestitev rešitve v sistem Time&Space

Rešitev zajema registracije s podporo tehnologije komunikacije v bližnjem polju lahko preprosto umestimo v sistem Time&Space. Uporabnik na svojem GSM aparatu potrebuje aplikacijo, s katero lahko nastavi ter pošlje podatke o registraciji. Na strani sistema Time&Space je potrebno pripraviti aplikacijo, ki bo prejete podatke znala obdelati ter jih pravilno zapisati v podatkovno bazo, da bodo lahko na voljo tudi drugim modulom sistema. Rešitev tako predstavlja dodatno funkcionalnost pri zajemu registracij ter je popolnoma transparentna za ostale komponente sistema Time&Space.



Slika 3: Umestitev rešitve zajema registracij z uporabo KBP v celostno rešitev sistema za obdelavo registracij delovnega časa produkta Time&Space

5 Metodologija objektnega razvoja ter jezik UML

Enotna metodologija razvoja informacijskih sistemov (v nadaljevanju EMRIS) priporoča tri možne pristope k razvoju informacijskega sistema. Četrty zvezek EMRIS opisuje objektni razvoj informacijskih sistemov, katerega bom uporabil za izdelavo informacijskega sistema za zajem registracij z uporabo tehnologije v bližnjem polju.

5.1 Kaj je metodologija

Metodologija zajema vse, kar redno počnemo, da bi dosegli želen rezultat, torej izdelek ali storitev, ki je cilj našega dela. V primeru razvoja programske opreme to ne pomeni zgolj postopkov, ki so neposredno povezani z razvojem, temveč zajema tudi podporne postopke, načine komunikacije med sodelujočimi, pravila odločanja in podobno. V tem oziru tehnologijo lahko opredelimo kot množico dogovorov, s katerimi se projektna organizacija strinja.

Metodologija je prežeta z idejami ter načeli organizacije ter njenih članov, kar še posebej poudari njeno sociološko komponento. Ne more nastati neodvisno od ljudi oziroma organizacije. Kljub temu, da so nekatere metodologije formalno določene, si jih posamezne organizacije oziroma ljudje prilagajajo po svojem merilu, tako da ustrezajo njihovem načinu dela ter njihovi domeni. Z uporabo metodologije si uporabniki pridobivajo izkušnje, s čimer se bogati tudi metodologija sama. [10]

5.2 Objektna tehnologija

Objektna tehnologija je vodilna paradigma razvoja informacijskih sistemov v praksi. Vzrok temu je zagotavljanje učinkovitih mehanizmov za doseganje temeljnih ciljev informatike, kot so visoka stopnja ponovne uporabe, kakovost ter večja produktivnost.

Objektni pristop je močno povezan inkrementalno-iterativnim razvojem, kar je posledica uporabe istih konceptov v vseh fazah razvoja. Prednost objektnega razvoja je namreč v tem, da se koncept ponovne uporabe ne nanaša le na izvirno kodo, pač pa tudi na modele, izdelane v zgodnejših fazah, ki jih ponovno lahko uporabimo v nadaljevanju razvoja. Osnovna ideja objektna tehnologije je torej izgradnja sistemov na osnovi obstoječih, že preizkušenih gradnikov. [9]

5.3 Orodja CASE

Veliko težo pri uveljavitvi objektne tehnologije imajo tudi orodja za podporo razvoja programske opreme (CASE orodja). Orodja nudijo podporo pri modeliranju, načrtovanju ter generiranju kode. Poleg tega, da podpirajo razvojne metode, nudijo tudi podporo pri generiranju testnih primerov. Pri razvoju sistema za zajem registracije z uporabo tehnologije KBP, bom uporabil orodje PowerDesigner podjetja Sybase. Orodje nudi podporo v vseh fazah razvoja, poudarek pri objektnem načrtovanju je na jeziku za modeliranje UML (Unified Modeling Language), ki je de facto standard pri objektnem razvoju. [9]

5.4 Izbira procesnega modela

Glede na tip projekta je potrebno izbrati procesni model. Glede na izbiro modela nato lažje določimo smernice izdelovanja programske opreme. [9] Poznamo več vrst procesnih modelov:

- prvi te vrste,
- izpeljanka izvedenega,
- preureditev obstoječega,
- kreiranje ponovno uporabnih komponent,
- izboljšave obstoječega in vzdrževanje.

Za projekt razvoja sistema za zajem registracije delovnega časa sem izbral procesni model »prvi te vrste«. Razlogi za izbiro tičijo v tehnologiji KBP, saj je razvoj aplikacije, ki bo delovala s to tehnologijo, popolnoma nova izkušnja.

5.5 Objektni proces razvoja

Na objektni proces razvoja lahko gledamo skozi dva vidika. To sta dinamični ter statični. Dinamični vidik razdeli projekt na faze ter tako odraža življenjski cikel procesa, medtem ko statični vidik razdeli procese na aktivnosti ter prikazuje njegove sestavine. [9]

5.5.1 Dinamični vidik

Razvoj programske opreme razdelimo na štiri faze:

- Začetna faza
V tej fazi definiramo projekt s poslovnega stališča ter določimo njegov obseg. V ta namen poiščemo vse zunanje entitete ter način sodelovanja. Poiskati moramo vse primere uporabe ter glavne med njimi tudi opisati. Gledano s poslovnega stališča je v tej fazi potrebno določiti kriterije uspeha, tveganja ter časovne mejnike.
- Faza zbiranja informacij
Potrebno je analizirati problemsko domeno, postaviti osnovno ogrodje arhitekture, izdelati projektni plan ter identificirati najpomembnejše elemente sistema. Opisati je potrebno vse primere uporabe, izbrati arhitekturo ter jo ovrednotiti. Na koncu pregledamo cilje sistema, njegov obseg, arhitekturo ter morebitna tveganja.
- Faza konstrukcije
V tej fazi izdelamo celoten proizvod. Faza zajema načrtovanje, implementacijo ter testiranje. Ob koncu faze moramo preveriti pripravljenost programske opreme ter njenih uporabnikov za opravljanje vsakodnevnih nalog.
- Faza prevzema
V tej fazi predamo programsko opremo uporabniku. Faza zajema popravke napak, ki se pojavijo z uporabo, dopolnitev funkcionalnosti za dele sistema, ki niso bili implementirani. Na koncu je potrebno oceniti, ali bo potrebno začeti razvijati novejšo verzijo, ter kako izboljšati obstoječi proces na podlagi pridobljenih izkušenj.

5.5.2 Statični vidik

Proces razvoja lahko razdelimo na dve skupini aktivnosti, tehnične ter podporne. Pri tehničnih aktivnosti so osredotočimo na opravila, ki so izrazito tehnične narave. Podporne aktivnosti so namenjene vodenju tehničnih aktivnosti. V nalogi se bomo osredotočili predvsem na sam razvoj informacijskega sistema, zato si bomo podrobneje ogledali tehnične aktivnosti.

- Aktivnost zajemanja zahtev
Cilj aktivnosti je pridobitev uporabnikovih zahtev, izdelati primere uporabe ter določiti nefunkcionalne zahteve, ki jih mora sistem izpopolnjevati.
- Aktivnost analize
Definirati je potrebno operacije ter objekte, ki bodo v sistemu nastopali. V tej fazi se ne oziramo na to, kako bodo te operacije in objekti izdelani.
- Aktivnost arhitekturno načrtovanje
Aktivnost določa strukturo sistema v smislu komponent in povezav med njimi. Rezultat aktivnosti je fizična arhitektura sistema, ki je izdelana na podlagi logične arhitekture.
- Aktivnost načrtovanja objektov
Objektom dodelimo operacije, določimo njihovo vidljivost ter sprejmemo odločitve glede predstavitve povezav. Rezultat aktivnosti je razredni diagram načrtovanja, diagram sodelovanja objektov ter posodobljen arhitekturni diagram.
- Aktivnost implementacije
Rezultate načrtovanja implementiramo v izbranem programskem jeziku. Upoštevati je potrebno arhitekturo sistema ter nefunkcionalne zahteve, ki so bile zajete.
- Aktivnost testiranja
Testirati je potrebno tako posamezne module kot sistem v celoti. Izdelati je potrebno tudi testne primere, ki jih lahko pri ponavljanju testiranja ponovno uporabimo. Za testiranje kode lahko izdelamo testne razrede.

5.6 Jezik UML

Jezik UML (Unified Modeling Language) ali drugače jezik za modeliranje se uporablja za načrtovanje programske opreme. Vsebuje množico diagramskih tehnik, s katerimi nato ustvarimo abstraktni model izbrane domene. Razvit je bil kot grafični jezik za vizualizacijo, specifikacijo ter dokumentacijo programske domene. Nudi podporo tako konceptualni predstavitvi rešitve, kot tudi natančnemu načrtovanju ter sami implementaciji. Jezik UML je standardno orodje pri objektivnem razvoju. [15]

Najbolj znani diagrami jezika UML so:

- razredni diagram,
- diagram komponent,
- postavitveni diagram,
- objektni diagram,
- diagram aktivnosti,
- diagram primerov uporabe,
- diagram stanj,
- diagram zaporedja.

6 Razvoj sistema za zajem registracij z uporabo objektne metodologije

Idejo o sistemu za zajem registracij delovnega časa z uporabo tehnologije KBP bomo realizirali po priporočilih EMRIS. Uporabili bomo objektni pristop k razvoju informacijskega sistema. Razvoj aplikacije bo potekal postopoma po aktivnostih ter opravilih, ki jih določa EMRIS. Aktivnosti potekajo skozi več faz razvoja in se pri tem dopolnjujejo. [9]

6.1 Zajemanje zahtev

Večina opravil aktivnosti zbiranja zahtev poteka v začetni fazi, fazi zbiranja informacij ter fazi konstrukcije.

6.1.1 Opredelitev zahtev

Pri opredelitvi zahtev je najprej potrebno identificirati vse vpletene osebe, ki bodo pri tem projektu sodelovale, ter jih razporedimo po fazah razvoja projekta, kjer bodo sodelovali. Pri tem moramo zajeti tako naročnike, investitorje, načrtovalce kot uporabnike. [9]

Pri razvoju sistema za zajem registracij je shema vpletenih oseb predstavljena v spodnji preglednici.

Faza	Vpletene osebe
Začetna faza:	<ul style="list-style-type: none"> strokovnjaki problemskega področja podjetja Špica d. o. o., uvozniki mobilnih telefonov, ki dobavljajo ustrezne aparate.
Faza zbiranja informacij:	<ul style="list-style-type: none"> prodajalci podjetja, kateri so zadolženi za iskanje potencialnih kupcev, razvijalec – tehnolog, ki preuči novo tehnologijo ter orodja, ki tej tehnologiji nudijo podporo.
Faza konstrukcije:	<ul style="list-style-type: none"> Vodja projekta pri podjetju, ki je zadolžen za izdelavo plana razvoja, zagotavljanje primernih kadrov, ter je zadolžen za vodenje. Načrtovalec, ki zagotavlja skladnost proizvodov posameznih aktivnosti. Ekipo programerjev, katerih naloga je implementirati rešitev na podlagi izdelanih načrtov. Testni inženir pripravi testni plan ter testne primere. Testna ekipa, ki izvaja plan testiranja.
Faza prevzema:	<ul style="list-style-type: none"> Uporabniki sistema, ki začnejo sistem uporabljati. Stranka, ki je sistem kupila. Serviserji podjetja Špica d. o. o., ki bodo pri stranki namestili sistem. Podpora podjetja Špica d. o. o., ki je zadolžena za pomoč pri težavah, ki se pojavijo po prevzemu sistema. Administratorji sistema, ki bodo zadolženi za konfiguracijo ter zaščito točk.

Preglednica 1: Seznam oseb, ki so vključene pri projektu razvoja IS za zajem registracij z uporabo KBP

Naslednji korak je opredelitev funkcionalnih zahtev. Sem spadajo želene ter potrebne lastnosti sistema, lahko pa tudi zahteve, ki jih določimo na podlagi konkurenčnih produktov.

V nadaljevanju so predstavljene funkcionalne zahteve našega sistema.

Seznam funkcionalnih zahtev informacijskega sistema:

- **Zajem registracije z uporabo tehnologije komunikacije v bližnjem polju**
Uporabnik sistema pride do KBP oznake. S svojim GSM aparatom se aplikaciji približa ter izbere dogodek, ki ga želi registrirati (prihod, odhod, malica, ...). Podatek se nato preko paketnega prenosa pošlje v sistem za obdelavo registracij delovnega časa Time&Space.
- **Omogočena izbira dogodka na izbrani točki**
Pri registraciji ima uporabnik možnost izbrati poljuben dogodek. Sistem naj uporabniku ponudi možnost, da želen dogodek izbere s seznama dogodkov, ki so na voljo. Dogodek naj bo podan s številko ter opisom. V kolikor seznama ni mogoče pridobiti, naj omogoči vnos proste izbire številke dogodka.

- **Možnost nastavitve podatkov o točki**
Potrebno je omogočiti nastavitve identitete posamezni točki. Ta podatek se bo nato ob registraciji prebral ter nato poslal v sistem za obdelavo registracij delovnega časa.
- **Nastavitve zaščite točke**
Omogočiti je potrebno možnost, da posamezno točko lahko tudi zaščitimo pred spreminjanjem. Točko lahko nato spreminja le oseba, ki pozna zaščitno geslo.
- **Omogočiti zajem »off-line« registracij**
V primeru, ko storitev za zajem registracij ni na voljo, naj se registracija, ki jo hočemo izvesti, shrani v pomnilnik telefona. Registracijo se kasneje lahko poizkusi poslati ponovno.

Poleg zajema funkcionalnih zahtev sistema, je potrebno opredeliti tudi nefunkcionalne zahteve, katerim mora sistem zadoščati. Nefunkcionalne zahteve so za končnega uporabnika enako pomembne kot funkcionalne.

Seznam nefunkcionalnih zahtev sistema:

- **Zagotoviti oznake KBP v različnih barvah**
KBP oznake bi bilo moč zagotoviti v različnih barvah, kar bi doprineslo k veji dodani vrednosti. Stranka bi tako lahko sama izbrala barvo oznake, ki bo najboljše sovpadala z njenimi prostori.
- **Izvedba registracije v realnem času**
Ko uporabnik želi izvesti registracijo, mora biti stik s točko kar najkrajši. Ne moremo si privoščiti, da bi uporabnik potreboval veliko časa za registracijo, saj so z registracijo na določeni točki lahko povezane še druge storitve (odpiranje vrat, ...). Poleg tega se v industrijskih podjetjih naenkrat želi registrirati večje število oseb, zato je hitrost pretoka na določeni točki zelo pomembna.
- **Razširljivost sistema**
Sistem za zajem registracij z uporabo komunikacije v bližnjem polju je prvi te vrste, zato so spremembe ter dodatki v prihodnosti pričakovani. Sistem mora biti zasnovan tako, da dodajanje funkcionalnosti ter spreminjanje sistema ne bosta zapletena.
- **Združljivost s sistemom za registracijo delovnega časa Time&Space**
Sistem za zajem registracij s tehnologijo komunikacije v bližnjem polju mora poskrbeti, da so podatki, ki se vpišejo v sistem Time&Space, pravilni in obračunani tako, da ne pride do anomalij pri nadaljnjem upravljanju sistema.

6.1.2 Izdelava diagramov primerov uporabe

Diagrami primerov uporabe povzamejo funkcionalnost sistema z akterji ter primeri uporabe. Cilj opravila je poiskati vse akterje ter primere uporabe, ki jih dobimo na podlagi vpletenih oseb ter funkcionalnih zahtev sistema. [9]

V sistemu, ki ga razvijamo obstajata dva akterja:

- Akter uporabnik, ki sistem vsakodnevno uporablja kot sredstvo za registracijo ob prihodu, odhodu ali kakem drugem dogodku.
- Akter administrator, katerega naloga je zagotoviti nastavitve posameznih točk na želene vrednosti in poskrbeti za njihovo zaščito pred spreminjanjem.

Poleg akterjev je potrebno v sistemu poiskati ter opisati še primere uporabe. Identificirani primeri uporabe v sistemu so naslednji:

- **Registracija uporabnika z uporabo tehnologije KBP**

Cilj uporabe sistema je izvesti registracijo na določeni točki. Za uspešno izvedbo registracije mora delovati storitev, ki zna prejete podatke zapisati v sistem Time&Space. Poleg tega mora biti na določeni lokaciji prisotna tudi KBP oznaka. Uporabnik, ki želi izvesti registracijo, mora imeti GSM aparat, ki podpira tehnologijo KBP.

Za izvedbo registracije mora uporabnik približati svoj GSM aparat k oznaki KBP. Tehnologija na aparatu zažene našo aplikacijo, ter prebere številko lokacije z oznake KBP. Številko lokacije pošlje strežniku KBP, kateri vrne seznam veljavnih dogodkov. V kolikor to ni mogoče, je uporabniku na voljo možnost, da številko dogodka vnese sam. Aplikacija nato podatek pošlje preko paketnega prenosa strežniku KBP, le-ta pa podatek zapiše v sistem Time&Space. Po uspešni izvedbi registracije dobi uporabnik povratno informacijo. V primeru, ko NFC strežnik ne pošlje odgovora, se registracija shrani v pomnilnik telefona, aplikacija pa bo ob naslednjem zagonu ponudila ponovno možnost pošiljanja dogodka

- **Nastavitev identitete KBP oznake na želeno vrednost**

Cilj primera uporabe je določiti identiteto posamezni točki, ki bo vezana na sistem Time&Space. Za izvedbo nastavitve na točki (oznaka KBP) potrebujemo telefon, ki podpira tehnologijo KBP.

Za izvedbo nastavitve mora administrator zagnati administrativno aplikacijo. GSM aparat nato približa oznaki KBP, nato pa nastavi novo vrednost točke. Po potrditvi se nova vrednost zapiše na oznako. V primeru, da ima oznaka nastavljeno zaščito pred spreminjanjem, mora administrator najprej vnesti zaščitno geslo.

- **Zaščita spreminjanja KBP oznake**

Oznako KBP lahko zaščitimo pred neavtoriziranim spreminjanjem. V primeru, da je oznaka zaščitena, moramo preko aplikacije najprej vnesti trenutno geslo zaščite. Po potrditvi lahko vnesemo novo geslo zaščite.

- **Pregled salda ur ter salda dopusta**

Uporabnik v svoji aplikaciji lahko izbere funkcijo za pregled salda ur ter salda dopusta. Aplikacija priskrbi podatke na podlagi uporabnikove identitete in mu jih prikaže na zaslonu GSM aparata.

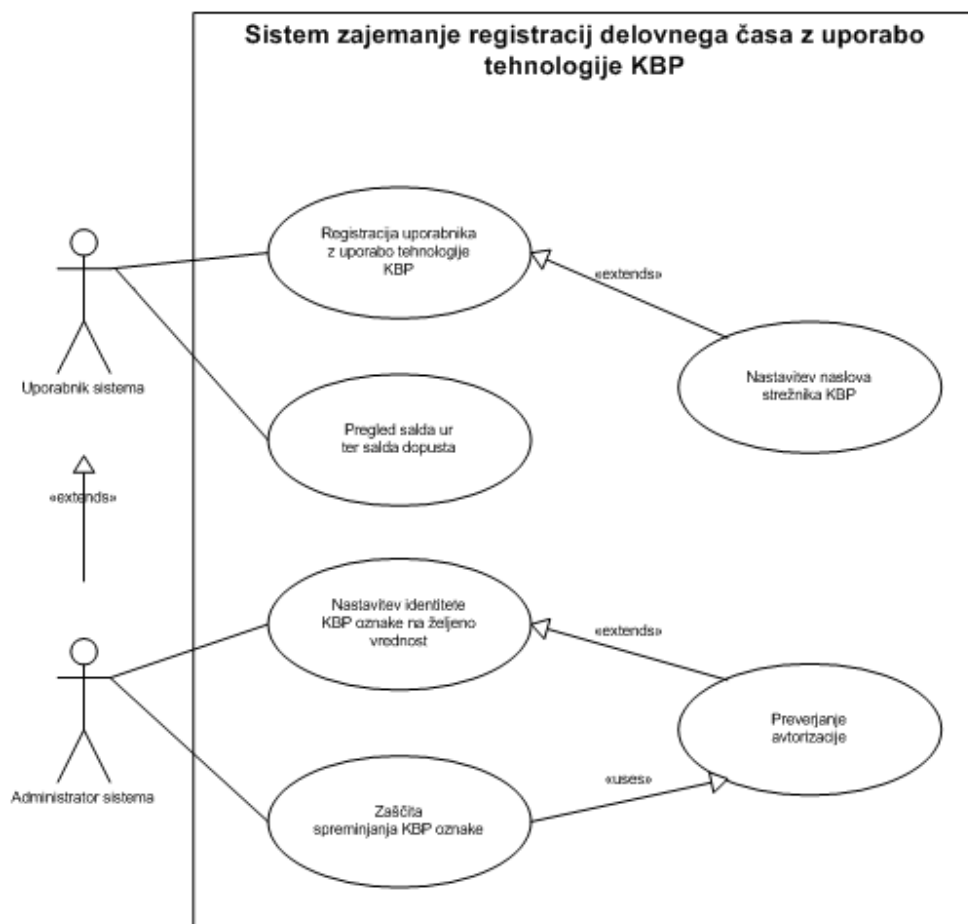
- **Nastavitev naslova strežnika KBP**

Uporabnik mora imeti v svoji aplikaciji možnost nastaviti naslov strežnika, kamor se registracije pošiljajo. Akcijo mora izvesti pred prvo uporabo oziroma po vsaki spremembi naslova.

- **Preverjanje avtorizacije**

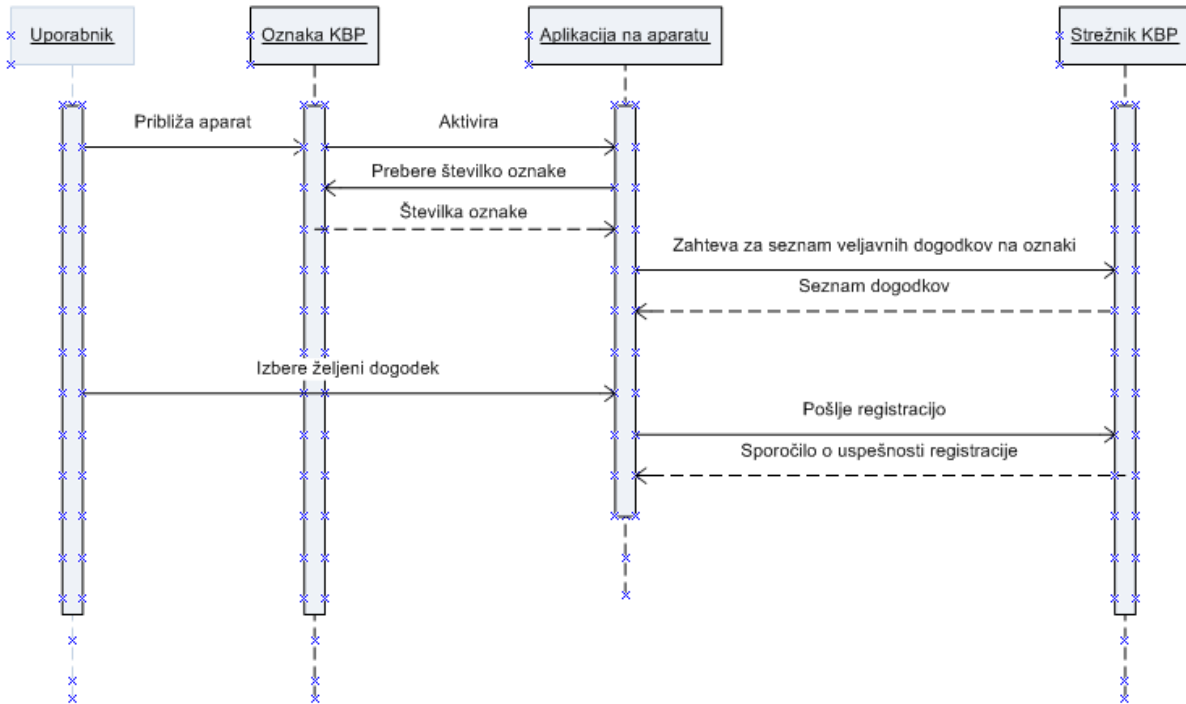
Določeni primeri uporabe pri svoji izvedbi zahtevajo preverjanje avtorizacije, na podlagi katere se omogoči izvedba le-tega.

Na podlagi identificiranih akterjev ter primerov uporabe sedaj lahko zgradimo diagram primerov uporabe za naš sistem.



Slika 4: Diagram primerov uporabe za sistem zajema registracij z uporabo tehnologije KBP

Za pomembnejše primere uporabe je potrebno izdelati tudi diagram zaporedja, s katerim natančno predstavimo primer uporabe. Najbolj pomemben primer uporabe v našem sistemu je zajem registracije, zato v ta namen izdelamo diagram zaporedja.



Slika 5: Diagram zaporedja za primer uporabe »registracija uporabnika s tehnologijo KBP«

6.1.3 Določanje prioritet primerov uporabe

Izmed vseh primerov uporabe je potrebno določiti tiste, ki so za sistem najbolj pomembni. V našem sistemu sta najpomembnejša dva primera uporabe. To sta:

- registracija uporabnika z uporabo tehnologije komunikacije v bližnjem polju in
- nastavitev podatkov KBP oznake na zelene vrednosti.

Brez implementacije teh dveh primerov uporabe sistem nima funkcionalnosti. Za ostale primere uporabe pa določimo nižjo stopnjo pomembnosti. Brez implementacije ostalih primerov uporabe bo sistem še vedno deloval, ne bo pa zadoščeno nekaterim njegovim zahtevam, kot je na primer varnost.

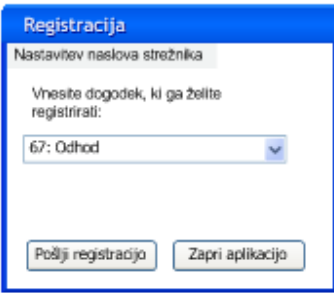
Zaradi relativne majhnosti sistema, bi razvoj le-tega postavil v eno samo iteracijo.

6.1.4 Izgradnja prototipa uporabniškega vmesnika in dopolnitev primerov uporabe

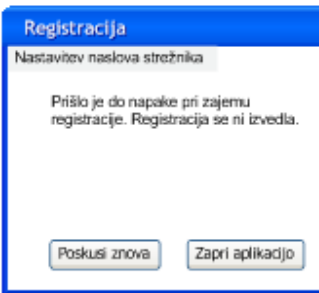
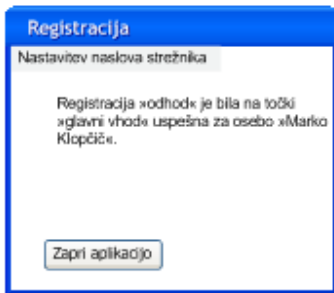
V aktivnosti zajema zahtev je potrebno predstaviti tudi primere zaslonских mask, s katerimi še dodatno pojasnimo sistem ter njegovo delovanje. V našem sistemu imamo dve aplikaciji. Prva je namenjena uporabniku za zajem registracij, druga pa administratorju za nastavljanje identitete točki. V ta namen izdelamo dva prototipa uporabniškega vmesnika, ki ga pred nadaljevanjem potrdi naročnik. Pri izdelavi prototipov lahko odkrijemo tudi nove primere uporabe, na katere prej morda nismo pomislili. V tem primeru dopolnimo predhodno izdelane diagrame ter opise. [9]

Uporabniški vmesnik akterja uporabnik

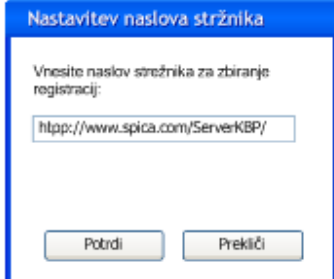
Uporabnik se približa KBP oznaki. Zažene se aplikacija za zajem registracije, kjer uporabnik vnese željen dogodek.



Po oddaji registracije, uporabnik dobi povratni informacijo o uspešnosti le-te.



V meniju lahko izbere tudi opcijo nastavitve strežnika. V tem dialogu vnese strežnik, kateremu se registracija pošlje in ki poskrbi za njeno obdelavo.



Slika 6: prototip uporabniškega vmesnika uporabniške aplikacije

Uporabniški vmesnik akterja Administrator

Administrator se najprej prijavi v aplikacijo s svojim geslom.

Avtorizacija

Vnesite geslo, ki omogoča spreminjanje nastavitve točk:

Potrdi Prekliči

Ob uspešni prijavi se mu odpre okno za nastavitve nove številke točke. Po vnosu in potrditvi dobi povratno informacijo o uspešnosti nastavitve.

Spreminjanje številke točke

Sprememba gesla...

Vnesite novo referenčno številko, ki jo želite nastaviti točki:

Nastavi Prekliči

Spreminjanje številke točke

Sprememba gesla...

Referenčna številka točke je bila uspešno nastavljena (57).

Zapri

V meniju je možno izbrati tudi spreminjanje gesla. Ob izbiri te akcije se odpre dialog za spreminjanje gesla. Ob potrditvi se novo geslo shrani.

Spreminjanje avtorizacije

Vnesite obstoječe geslo:

Vnesite novo geslo:

Potrdite novo geslo:

Potrdi Prekliči

Slika 7: Prototip uporabniškega vmesnika administratorske aplikacije

Vsi modeli, opisi ter diagrami, ki smo jih izdelali v aktivnosti zajema zahtev, tvorijo skupaj model zahtev. Model zahtev je glavni proizvod te aktivnosti, zato ga je potrebno pred nadaljevanjem razvoja preveriti skupaj z naročnikom.

6.2 Analiza

Aktivnost analize poteka skozi vse faze razvoja programske opreme. Največji poudarek aktivnosti, pa je v fazi zbiranja zahtev. V aktivnost določimo razrede, povezave med njimi, izdelamo razredni diagram, ter več drugih diagramov, s katerimi natančno pojasnimo ter opišemo delovanje sistema. Glavni izdelek te aktivnosti je model analize, kateri sestoji iz proizvodov posameznih opravil aktivnosti. [9]

6.2.1 Izdelava razrednega diagrama obravnavane domene

Pri izdelavi razrednega diagrama je potrebno najprej določiti koncepte sistema. Koncepte poskusimo določiti na podlagi opisov primerov uporabe. V našem sistemu so koncepti domene naslednji:

- uporabnik,
- administrator,
- registracija,
- dogodek,
- točka,
- tehnologija komunikacije v bližnjem polju,
- strežnik KBP,
- sistem za upravljanje z registracijo delovnega časa,
- paketni prenos.

Na podlagi konceptov sistema je potrebno določiti logične povezave med njimi. V našem primeru so logične povezave naslednje:

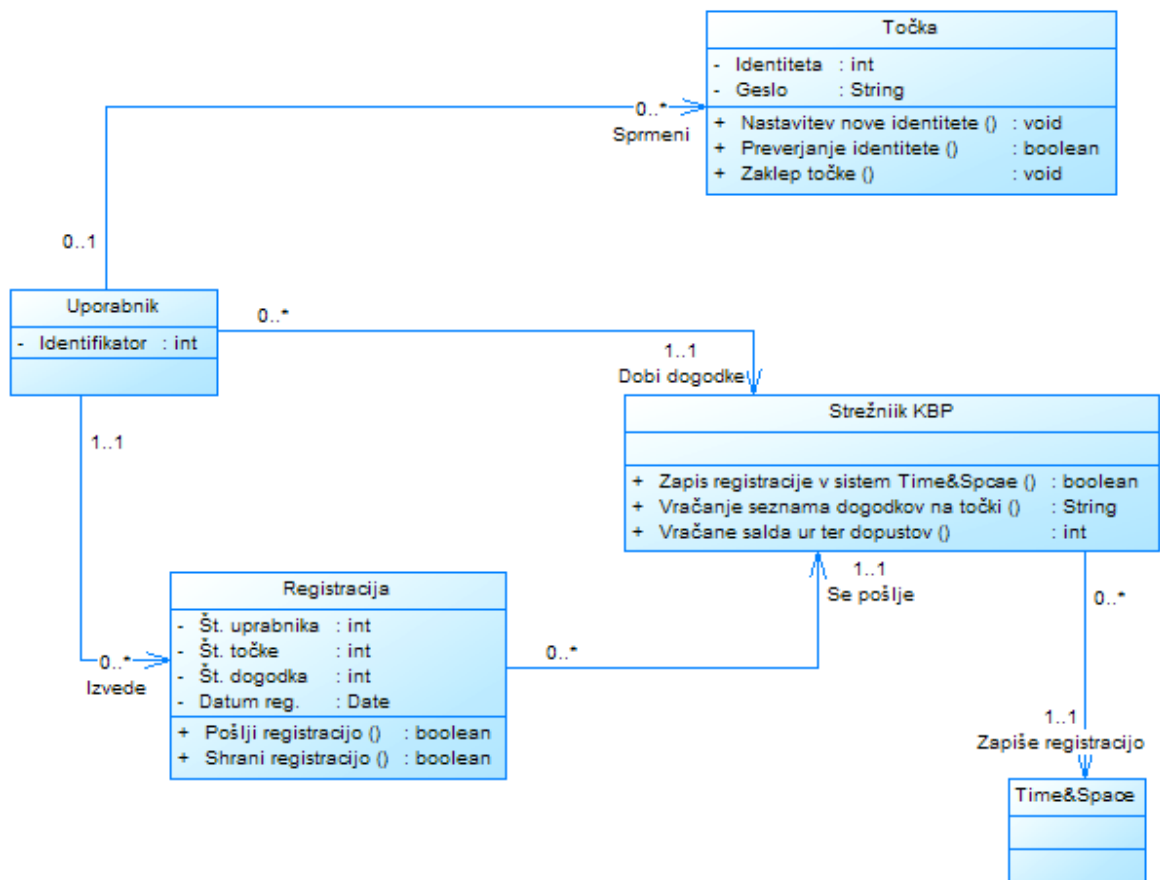
- Uporabnik *izvede* registracijo.
- Registracijo *sestavlja* podatek o identiteti uporabnika.
- Registracijo *sestavlja* podatek o točki.
- Registracijo *sestavlja* podatek o dogodku.
- Registracijo *pošljemo* preko paketnega prenosa podatkov strežniku KBP.
- Podatek o točki *preberemo* z uporabo tehnologije komunikacije v bližnjem polju.
- Strežnik KBP *zapiše* registracije v sistem za upravljanje z registracijami delovnega časa.
- Administrator *nastavi* novo vrednost točke s tehnologijo komunikacije v bližnjem polju.
- Administrator z uporabo komunikacije v bližnjem polju lahko *spremeni* zaščito oznake KBP.

Poleg logičnih povezav je potrebno konceptom domene določiti njihove osnovne operacije ter attribute, ki so pomembni z vidika informacijske domene:

Koncept	Operacije ter atributi
Uporabnik:	<ul style="list-style-type: none"> • identiteta – IMEI.
Registracija:	<ul style="list-style-type: none"> • številka uporabnika, • številka točke, • številka dogodka, • čas registracije: <ul style="list-style-type: none"> ➤ pošiljanje registracije, ➤ shranjevanje registracije, ➤ pošiljanje shranjenih registracij.
Tehnologija KBP:	<ul style="list-style-type: none"> ➤ vzpostavitev komunikacije, ➤ pošiljanje podatkov, ➤ prejemanje podatkov.
Strežnik KBP:	<ul style="list-style-type: none"> ➤ zapis registracije v sistem Time&Space, ➤ vračanje seznama dogodkov na točki, ➤ vračanje salda ur, ➤ vračanje salda dopusta.
Točka:	<ul style="list-style-type: none"> • identiteta točke, • zaščitno geslo: <ul style="list-style-type: none"> ➤ nastavitve nove identitete točki, ➤ zaklep točke, ➤ preverjanje avtorizacije na točki.
Paketni prenos:	<ul style="list-style-type: none"> ➤ vzpostavitev povezave, ➤ pošiljanje podatkov, ➤ prejemanje podatkov.

Preglednica 2: Atributi ter operacije nad posameznimi koncepti domene

S tem, ko smo določili koncepte ter njihove operacije ter attribute, lahko nato konstruiramo konceptualni razredni diagram. Pri tem lahko uporabimo tudi metodo generalizacije, v kolikor se le-ta v zgradbi sistema pojavi.



Slika 8: Konceptualni razredni diagram sistema

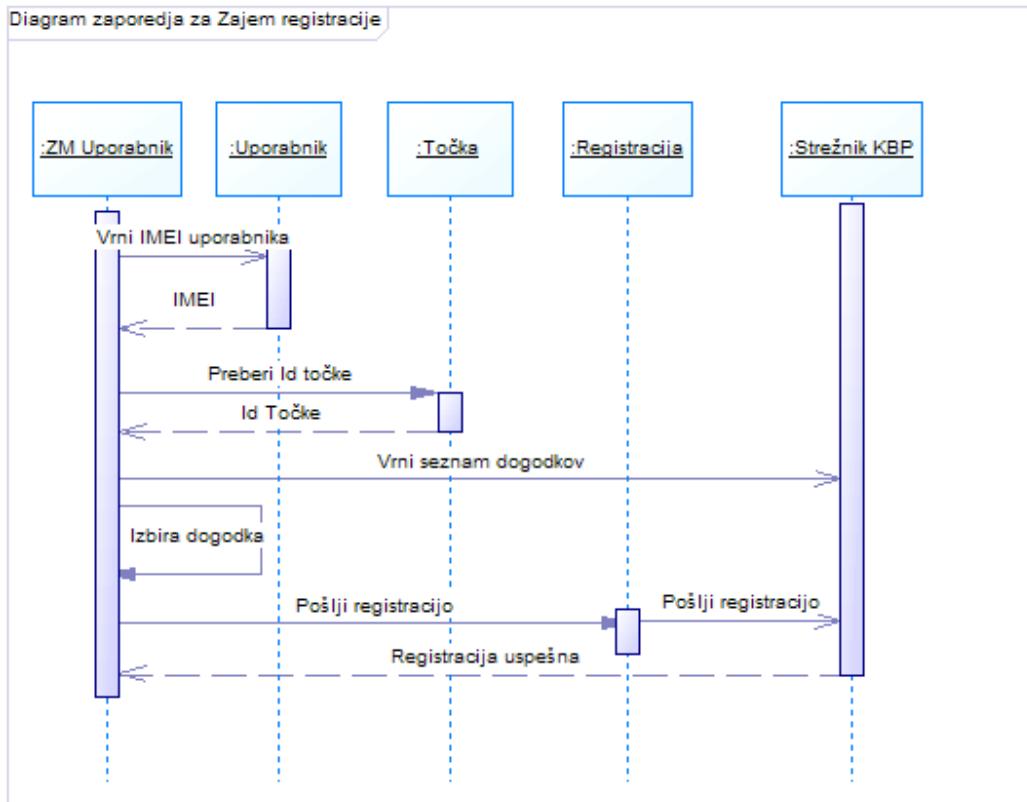
6.2.2 Analiza primerov uporabe

Opravilo analiza primerov uporabe zajema dopolnitev le-teh na podlagi novih spoznanj. Posamezne primere uporabe podrobneje razdelamo ter poizkusimo obnašanje sistema opisati z odgovornostmi. Na podlagi identificiranih primerov uporabe moramo najprej določiti njihove realizacijske razrede [9]. Razdelitev pri našem sistemu je predstavljena v spodnji preglednici:

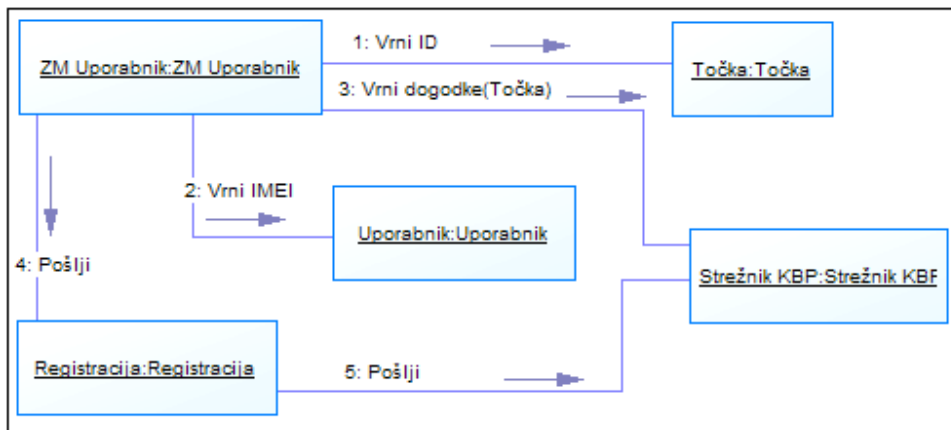
Primer uporabe	Realizacijski razredi
Registracija uporabnika z uporabo tehnologije KBP:	<ul style="list-style-type: none">• ZM Uporabnik,• Registracija,• Točka,• Strežnik KBP,• Uporabnik.
Zaščita spreminjanja KBP oznake:	<ul style="list-style-type: none">• ZM Administrator,• Točka,• Uporabnik.
Pregled salda ur ter salda dopusta:	<ul style="list-style-type: none">• ZM Uporabnik,• Uporabnik,• Strežnik KBP.
Nastavitev naslova strežnika KBP:	<ul style="list-style-type: none">• ZM Uporabnik.
Preverjanje avtorizacije:	<ul style="list-style-type: none">• ZM Administrator,• Točka,• Uporabnik.

Preglednica 3: Realizacijski razredi za posamezne primere uporabe sistema za zajem registracij

Naslednji korak opravi izdelava diagramov interakcij za posamezne primere uporabe. Diagrami interakcij prikazujejo, kako realizacijski sistemi med seboj sodelujejo ter njihovo povezanost. Praviloma diagrame izdelamo samo za pozitivne scenarije primerov uporabe. Poznamo dva tipa diagramov interakcij. To sta diagram zaporedja, ki kaže dinamiko primera uporabe skozi čas, ter diagram sodelovanja, ki prikazuje statičen pogled na razrede, ki sodelujejo v posameznem primeru uporabe. [9]



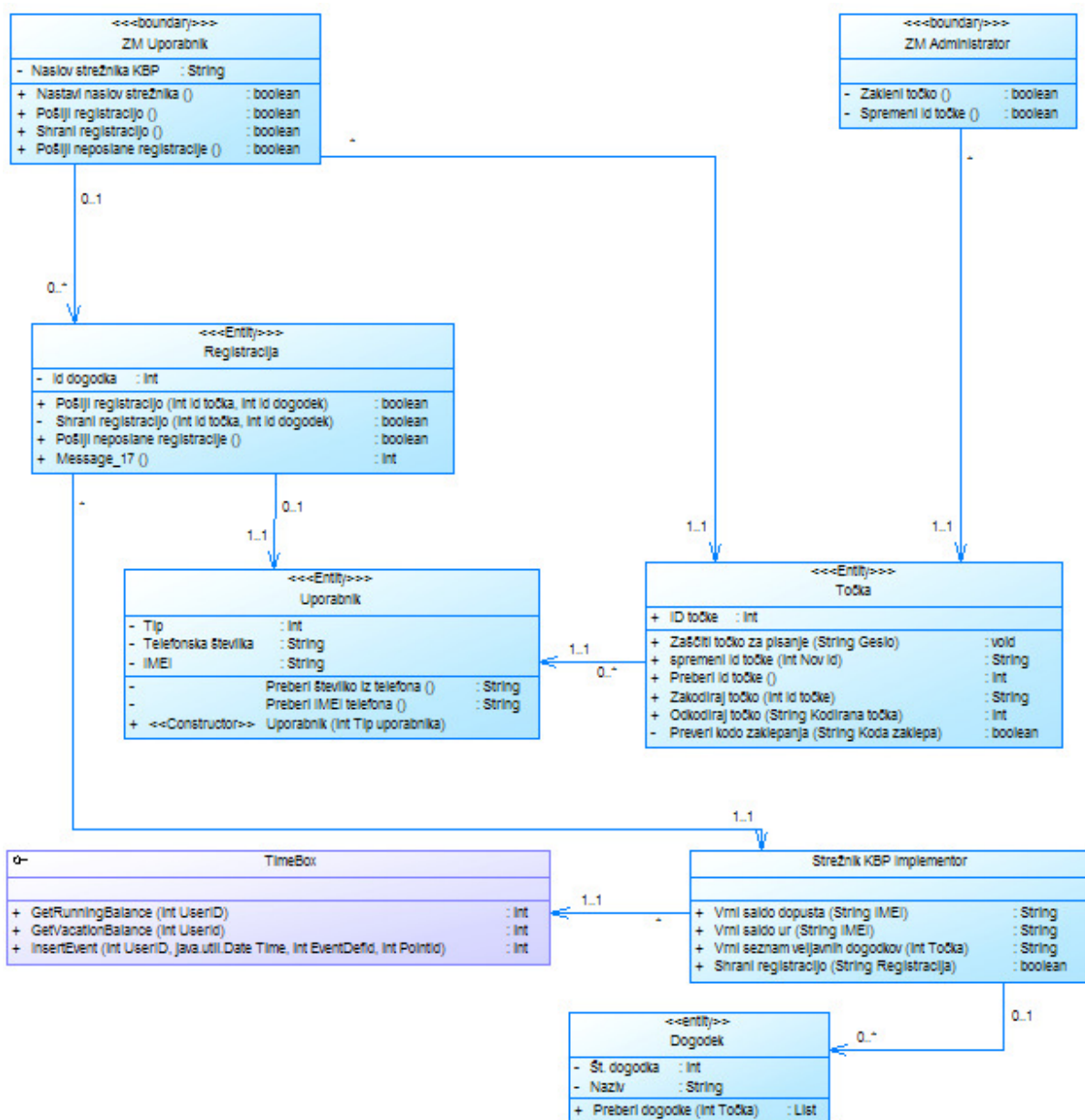
Slika 9 Diagram zaporedja za primer uporabe »Zajem registracij«



Slika 10: Diagram sodelovanja za primer uporabe »Zajem registracij«

6.2.3 Razvoj razrednega diagrama analize

Konceptualni razredni diagram dopolnimo z rezultati natančnega pregleda primerov uporabe. Določimo parametre posameznih operacij razredov ter iz diagrama odstranimo zunanje razrede. Za razrede, ki izražajo pomembno dinamično obnašanje, izdelamo diagrame stanj. Le-te konstruiramo na podlagi opisa ter vseh diagramov interakcij razreda. Razrede nato na podlagi njihove funkcionalnosti poizkusimo združiti v komponente. [9]



Slika 11: Razredni diagram sistema za zajem registracij delovnega časa z uporabo KBP

6.2.4 Specificiranje operacij sistema

Na podlagi modela primerov uporabe ter zajetih funkcionalnih zahtev, določimo glavne operacije sistema ter za njih določimo predpogoje, popogoje, objekte, ki jih spreminjamo ter dogodke, ki se pri tem pošiljajo [9]. Za sistem pošiljanja registracij so ključne operacije, njihove lastnosti ter razredi, ki operacijo implementirajo in so definirani v spodnji preglednici.

Operacija	Lastnosti operacije	Realizacijski razredi
Pošiljanje registracije	<ul style="list-style-type: none">• Predpogoji:<ul style="list-style-type: none">○ Zajeti podatki o točki, dogodku, identiteti uporabnika ter času.○ Delovanje strežnika za obdelavo registracij.• Operacija vrne podatek, če je bila registracija uspešna.	<ul style="list-style-type: none">• Točka• Uporabnik• Registracija• Strežnik KBP
Nastavitev identitete točki	<ul style="list-style-type: none">• Predpogoji:<ul style="list-style-type: none">○ Točka ne potrebuje gesla oziroma vnesemo pravilno geslo.• Povratna informacija o uspešni nastavitvi.	<ul style="list-style-type: none">• Točka
Zaščita točke	<ul style="list-style-type: none">• Predpogoji:<ul style="list-style-type: none">○ V primeru predhodne zaščite je potrebno vnesti staro geslo.	<ul style="list-style-type: none">• Točka
Pridobivanje podatka o saldirh	<ul style="list-style-type: none">• Predpogoji:<ul style="list-style-type: none">○ Delovanje strežnika za obdelavo registracij.	<ul style="list-style-type: none">• Uporabnik• Strežnik KBP

Preglednica 4: Operacije sistema, njihove lastnosti ter razredi, ki jih implementirajo

6.2.5 Pregled modela analize

Pri pregledu modela analize moramo dobiti odgovor na vprašanje, ali smo primere uporabe pravilno realizirali v skladu z njihovimi specifikacijami. Pomembno je, da se primeri uporabe ter modeli analize med seboj skladajo. Poleg tega se morajo med seboj skladati tudi specifikacije operacij sistema ter razredni diagram analize. V primeru, da prihaja do razlik, je potrebno podatke dopolniti. [9]

6.3 Arhitekturno načrtovanje

Arhitekturo razumemo kot specifikacijo sistema v smislu komponent ter povezav med njimi. Na arhitekturo vplivajo primeri uporabe, nefunkcionalne zahteve, izkušnje razvojne skupine ter druge omejitve (obstoječi sistem ...). Primarni rezultat aktivnosti je arhitektura z opisom sodelovanja med komponentami. Večina aktivnosti poteka v fazi zajema zahtev.

Z arhitekturnim opisom pridobimo večje razumevanje sistema, dobimo informacijo, kako organizirati razvoj ter spodbujamo ponovno uporabnost.

6.3.1 Organiziranje razredov v komponente

Komponenta je skupek razredov, ki imajo kot celota nizko stopnjo zunanje povezanosti. V sistemu moramo tako poiskati razrede, ki bi jih med seboj povezali ter združili v komponento. Komponenti sta tudi uporabniški vmesnik ter podatkovna baza. Preveriti je potrebno tudi nefunkcionalne zahteve, ki imajo lahko zahteve tudi po posebnih komponentah (dodaten strežnik, ...). [9]

V sistemu za zajem registracije z uporabo KBP identificiramo naslednje komponente:

- uporabniška vmesnika za uporabnika ter administratorja,
- komponenta registrator KBP, ki zna s tehnologijo KBP brati, pisati ter pošiljati podatke,
- podatkovna baza, v katero zapišemo podatke,
- komponenta Timebox, ki jo uporabimo za obračun podatkov,
- Time&Space KBP, ki skrbi za komunikacijo mobilnih aplikacij s sistemom Time&Space.

Komponenta	Opis	Realizacijski razredi
Uporabniški vmesnik uporabnika ter administratorja	Uporabniška vmesnika skrbita za interakcijo med akterjema uporabnik in administrator ter samim sistemom za zajem registracij delovnega časa.	<ul style="list-style-type: none">• ZM Uporabnik• ZM Administrator
Komponenta registrator KBP	Komponenta združuje vso funkcionalnost zajema registracije ter drugih operacij, ki se izvajajo s tehnologijo komunikacije v bližnjem polju. Omogoča branje ter pisanje na točko, zajem registracije ter njeno shranjevanje v primeru napake strežnika.	<ul style="list-style-type: none">• Točka• Registracija• Uporabnik
Time&Space KBP	Vsebuje logiko za pridobivanje ter zapisovanje podatkov v sistem Time&Space.	<ul style="list-style-type: none">• Dogodek• Strežnik KBP
TimeBox	Komponenta sistema Time&Space, ki se uporablja za obračun podatkov o uporabnikih ter vrača zelene podatke iz sistema Time&Space.	<ul style="list-style-type: none">• TimeBox (interface)
Podatkovna baza	Podatkovna baza sistema Time&Space, v katero se zapisujejo poslane registracije.	

Preglednica 5: Opisi komponent ter njihovi realizacijski razredi

6.3.2 Izbira arhitekturnih stilov

Ključna naloga aktivnosti je zgraditi fizično arhitekturo. V ta namen je potrebno izbrati primerne arhitekturne stile, ki bodo zadostili nefunkcionalnim zahtevam. V našem sistemu je za izbiro arhitekturnega stila pomembna zahteva »Zajem registracije v realnem času.« Le-ta od nas zahteva, da se registracija zapiše v sistem takoj, ko je to mogoče. Pri tem je potrebno upoštevati, da je registracija lahko zajeta na različnih lokacijah. Najprimernejši arhitekturni stil, ki se prilega želeni funkcionalnosti, je princip odjemalec – strežnik.

Arhitektura Odjemalec – Strežnik

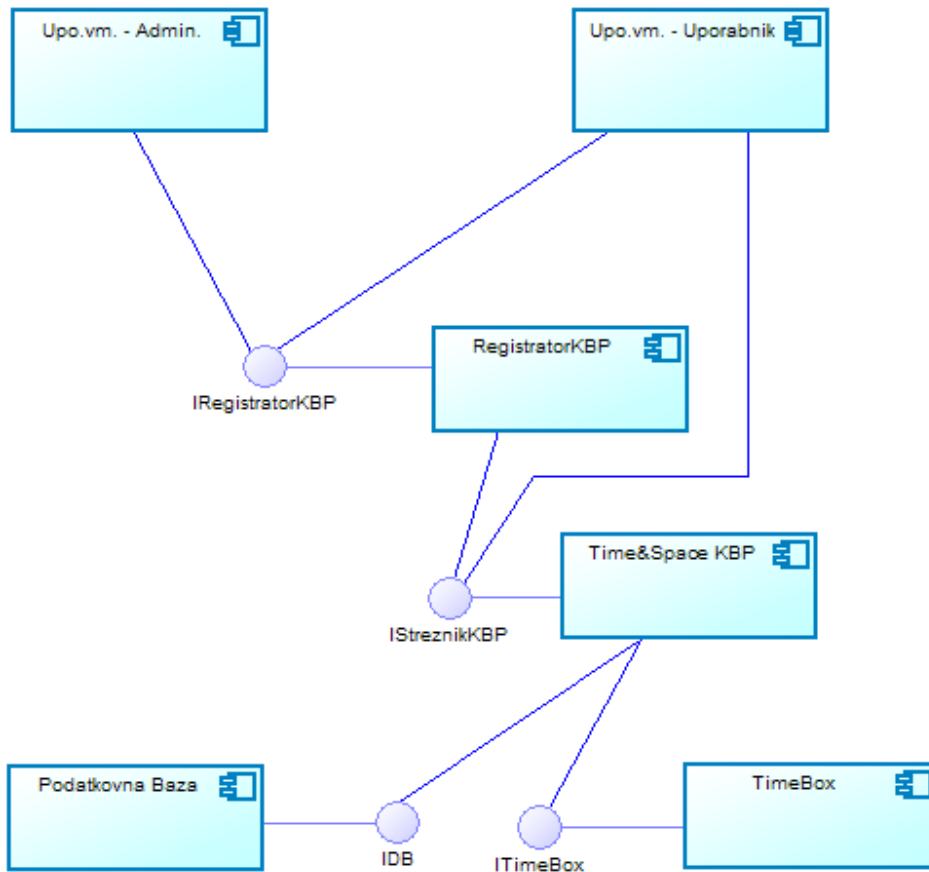
Arhitektura odjemalec – strežnik opisuje odnos med dvema aplikacijama. Aplikacija odjemalec pošilja zahteve strežniški aplikaciji, le-ta pa te zahteve obdela ter vrne zelene rezultate. Prednosti izbrane arhitekture so naslednje:

- Medsebojna povezljivost (interoperability) - ključne komponente delujejo neodvisno po skupnem protokolu.
- Skalabilnost (scalability) - zaradi spremembe obsega obdelave lahko vsako od ključnih komponent kadarkoli zamenjamo z bolj ali manj zmogljivo, brez večjih sprememb na ostalih komponentah.
- Prilagodljivost (adaptability) - novo tehnologijo brez težav vključimo v obstoječi sistem.
- Dosegljivost (accessability) - podatki so enostavno dosegljivi s pomočjo računalniškega omrežja in aplikacij odjemalca.
- Učinkovitost (performance) - učinkovitost se lahko optimizira z optimizacijo strojne opreme in procesov.
- Varovanje (security) - podatkovna varnost je centralizirana na strežniku.
- Redundantnost (redundancy) - z vgradnjo redundantnih komponent lahko omogočimo delovanje sistema kljub izpadu določenih komponent.
- Uravnoteženje (balancing) - sistem lahko enostavno uravnotežimo. Preobremenjene poti razbremenimo tako, da podatke preusmerjamo na manj obremenjene poti (routing).

6.3.3 Izgradnja logične arhitekture

Za arhitekturni stil smo se odločili na podlagi narave rešitve in z izbiro http protokola kot komunikacijskega medija. Izbrana arhitektura nam že v osnovi zagotavlja možnost obdelave velikega števila zahtev. Http protokol, ki ga uporabljamo kot komunikacijsko sredstvo, je že v osnovi zasnovan po modelu odjemalec - strežnik in je kot tak odličen temelj za izgradnjo tovrstnih aplikacij.

Na podlagi diagramov primerov uporabe lahko izdelamo diagrame interakcij med komponentami ter tako ugotovimo povezave med komponentami sistema.



Slika 12: Diagram komponent ter njihovo sodelovanje

Ogrodje arhitekture nato na podlagi uporabnikovih zahtev ter diagrama primerov uporabe preverimo.

6.3.4 Izgradnja fizične arhitekture

Arhitekturo, ki smo jo definirali na logičnem nivoju je sedaj potrebno prenesti še na fizični nivo. Potrebno je določiti tehnološko osnovo sistema. Podatkovna baza sistema je MS SQL. Strežniška aplikacija se bo izvajala kot spletna storitev na strežniku. Izdelana bo v Microsoft Visual Studio 2005. Uporabljala bo NET razvojno okolje verzije 3.5.

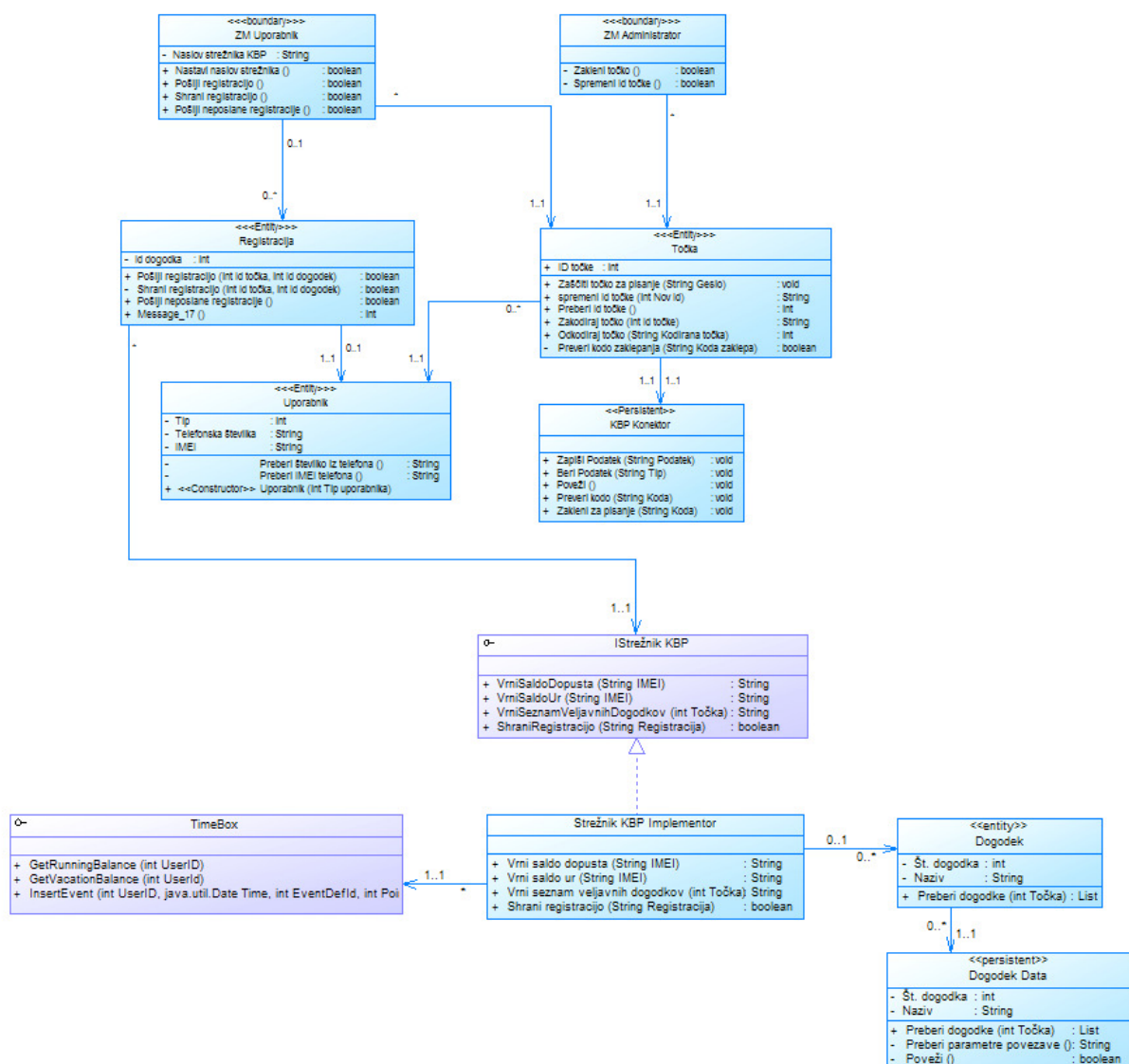
Aplikacija, ki bo tekla na GSM aparatu, bo izdelana v razvojnem okolju Java ME, saj ga podpira večina aparatov. Sama komunikacija med aplikacija bo potekala po protokolu http.

6.4 Načrtovanje objektov

Večina opravil aktivnosti načrtovanja objektov se opravi v fazah zbiranja informacij ter fazi konstrukcije. Aktivnost zajema natančno definiranje razredov, njihove interakcije ter načrtovanje zahtevnih algoritmov, ki se bodo v sistemu izvajali. [9]

6.4.1 Razvoj razrednega diagrama načrtovanja

Na podlagi komponent sistema določimo interno strukturo posameznih komponent. Pri tem moramo upoštevati izbrano arhitekturo sistema ter funkcionalnost posameznih komponent. Izdelati je potrebno razredne diagrame, iz katerih je razvidno lastništvo med razredi, njihovo sodelovanje ter metode, ki jih izvajajo. [6, 9]



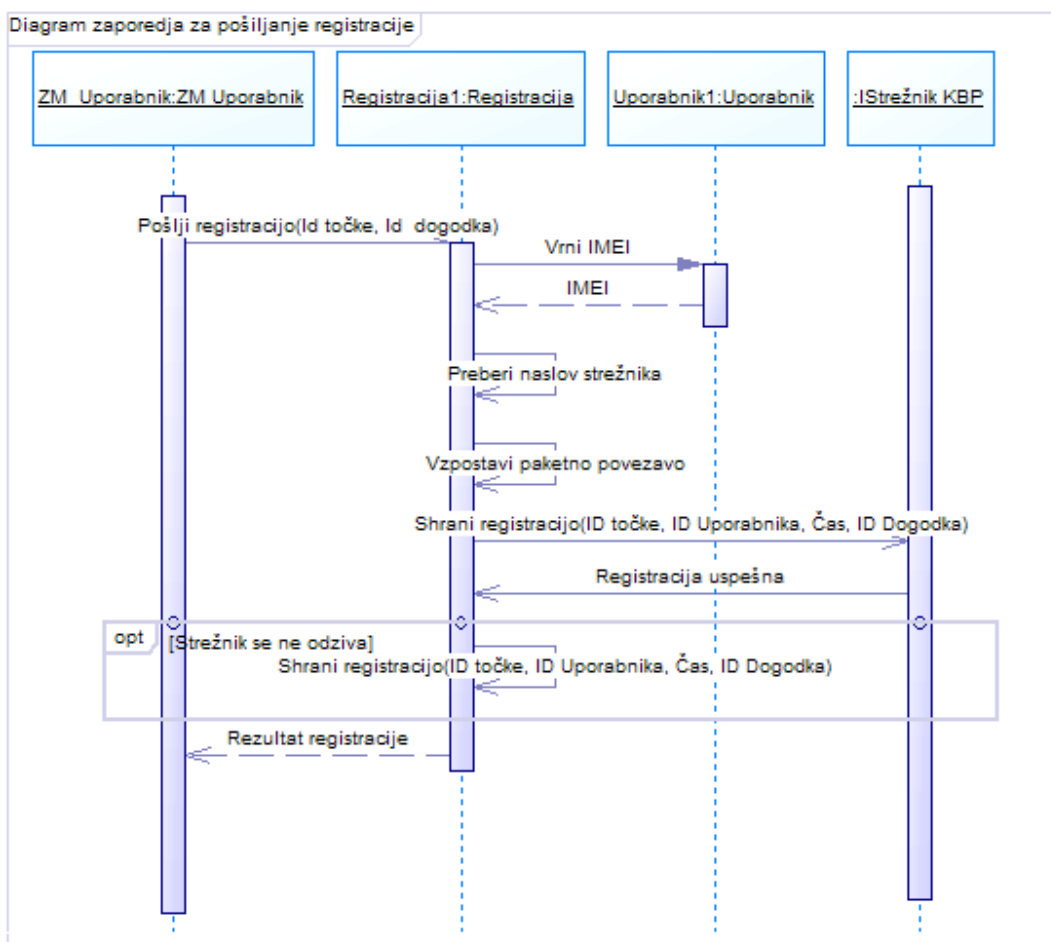
Slika 13: Razredni diagram aktivnosti načrtovanja objektov

6.4.2 Načrtovanje algoritmov

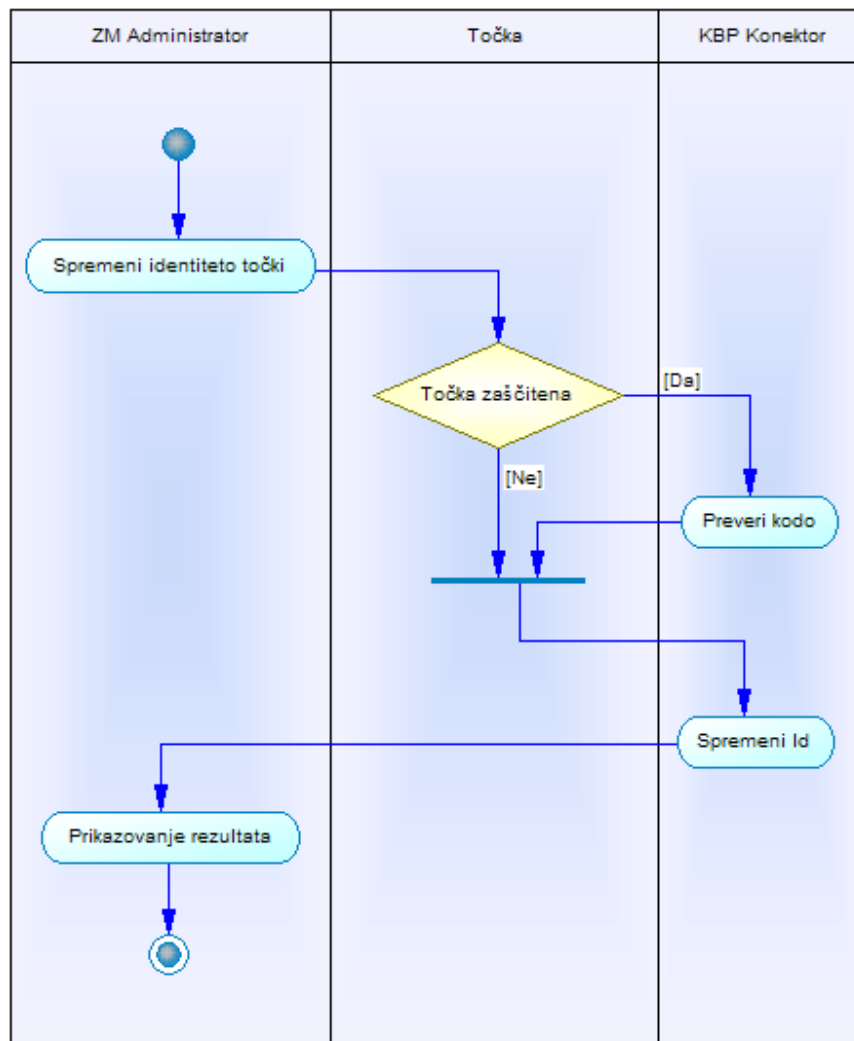
Na podlagi operacij sistema, ki smo jih določili v fazi analize, identificiramo kompleksne algoritme, katere bi bilo pred implementacijo dobro natančneje razdelati. V sistemu za zajem registracij z uporabo tehnologije KBP je najbolj kompleksna operacija »pošiljanje zajete registracije«, saj je v primeru da pošiljanje ni mogoče, potrebno poskrbeti za njeno začasno shranjevanje.

Algoritem za pošiljanje registracij

Pri pošiljanju registracije se poleg vnese številke dogodka uporabnika ter prebrane identitete točke prebereta še sistemski datum in čas ter nato IMEI telefona, katerega dobimo iz objekta uporabnik. Ko dobimo IMEI objekta, uporabnika ne potrebujemo več. Objekt registracija nato vzpostavi sistemsko povezavo. Na naslov strežnika, ki ga prebere iz podatkov, nato pošlje vse štiri podatke o registraciji v tekstovni obliki. Po poslani registraciji čaka na odgovor s strežnika. V primeru, da odgovora ne dobi, registracijo shrani v pomnilnik telefona. V nasprotnem primeru odgovor strežniške aplikacije vrne uporabniškemu vmesniku.



Slika 14: Zaporedni diagram izvajanja algoritma za pošiljanje registracij.



Slika 15: Diagram aktivnosti algoritma za spreminjanje identitete točki

Ostale operacije sistema so relativno enostavne za samo implementacijo. Algoritmčno niso zahtevne, so pa nekatere izmed njih težavne za implementacijo zaradi slabe tehnične podprtosti. To se nanaša predvsem na knjižnice jezika Java, ki nudijo podporo tehnologiji KBP, ki je še v razvoju in še ni popolnoma izpopolnjena.

6.4.3 Določanje dostopa in optimizacija

Določiti je potrebno dostope med objekti odjemalci ter objekti strežniki. Potrebno je ugotoviti vrsto dostopa, čas trajanja interakcije ter dostop med njimi optimizirati. [9]

Objekt odjemalec	Objekt strežnik	Vrsta dostopa ter čas obstoja
strežnik KBP Implementator	TimeBox	Do objekta TimeBox dostopamo preko njegovega interface-a. Objekt živi ves čas obstoja objekta odjemalec.
strežnik KBP Implementator	dogodek	Objekt dogodek se kreira ob klicu metode »Vrni dogodke«. Ob izhodu iz metode se objekt uniči.
registracija, točka	uporabnik	Objekt uporabnik se kreira, ko imamo zahtevo po podatku o identiteti. Ko podatek dobimo, instanco objekta sprostimo.
ZM uporabnik	registracija	Objekt kreiramo, ko želimo izvesti registracijo. Ta nato živi, dokler registracija ni izvedena oziroma dokler obstaja odjemalec.
ZM uporabnik, ZM administrator	točka	Objekt točka se kreira ob kreiranju odjemalčevega objekta. Ta do njega dostopa preko reference privatne spremenljivke.
registracija	strežnik KBP Implementator	Do objekta dostopamo preko vmesnika.
točka	KBP komunikator	KBP komunikator se kreira ob kreiranju objekta točka. Točka do njega dostopa preko reference.

Preglednica 6: Dostopi ter čas trajanje povezav

6.4.4 Določanje začetnega stanja objektov ter pregled skladnosti

Objektom je potrebno določiti začetna stanja. Potrebno je ugotoviti, kdaj se določen objekt kreira ter v kakšnem stanju bo. Na podlagi začetnega stanja atributov posameznega objekta, bo komponenta kasneje rasla. [9]

Objekt	Atribut	Začetno stanje
točka	id	Id točke se nastavi, ko se vzpostavi prva komunikacija s točko preko KBP. Do takrat je id nastavljen na predefinirano vrednost 0.
ZM uporabnik	naslov strežnika	Nastavi se ob kreiranju objekta. Naslov se prebere iz datoteke nastavitvev. V primeru, da podatka ni, je vrednost atributa prazna.
registracija	id dogodka	Vrednost se nastavi ob klicu metode pošiljanja registracije. Do takrat je predefinirana vrednost 0.
uporabnik	IMEI	Vrednost se nastavi ob branju tega podatka. Ob prvem klicu se pokliče funkcija, ki prebere podatek iz telefona.

Preglednica 7: Določitev začetnega stanja atributom.

Ob zaključku aktivnosti načrtovanja objektov morajo načrtovalci preveriti skladnost modela analize z modelom načrtovanja objektov. V primeru neskladnosti je potrebno izvesti korektivne ukrepe, ki so potrebni za doseg skladnosti. Skladnost modelov je pomembna zaradi potencialnega nadaljnega razvoja sistema. Ker je glavna značilnost objektnega razvoja iterativno-inkrementalni pristop, je skladnost med modeli potrebno zagotoviti zaradi prihodnjih iteracij. [9]

6.5 Implementacija

S sodobnimi orodji za načrtovanje in razvoj programske opreme je aktivnost implementacije precej enostavnejša. Večino zahtevnih odločitev smo analizirali ter sprejeli že v aktivnosti načrtovanja arhitekture ter načrtovanja objektov. [9]

6.5.1 Generiranje tehnoloških razredov

Tehnološki razredi so razredi, ki se nanašajo na samo tehnologijo, ki jo uporabljamo v našem sistemu, ne vsebujejo pa poslovne logike našega informacijskega sistema. Pri sistemu zajema registracije z uporabo KBP imamo dva tehnološka razreda. To sta razred »Dogodek Data«, katerega naloga je zapisovanje ter branje podatkov iz podatkovne baze in razred »KBP komunikator«, katerega naloga je interakcija z oznako KBP. Z orodjem PowerDesigner generiramo ogrodje tehnoloških razredov v izbranem jeziku. Generirana razreda že vsebujeta attribute ter metode, ki smo jih definirali v predhodnih aktivnostih. V generiranih razredih je nato potrebno implementirati metode, ki bodo izvajale zahtevano funkcionalnost. Pri implementaciji razreda KBP komunikator bomo uporabili knjižnico podjetja Nokia, ki je narejena v jeziku Java in podpira funkcionalnost, potrebno za komunikacijo z napravo KBP.

6.5.2 Razvoj kode razredov

Z orodji CASE se ogrodje razredov generira avtomatsko. Atributi se preslikajo v spremenljivke razreda, operacije iz faze načrtovanja pa postanejo metode razreda. Na podlagi specifikacij operacije implementiramo posamezne metode. Posebno pozornost moramo nameniti tudi obravnavi napak ter poskrbeti, da se obnašanje komponente pri vseh napakah obnaša enotno [9].

Potrebno je postaviti tudi določila za pisanje kode. Kodo bo verjetno pisalo več programerjev, zato je smiselno, da se vzpostavijo osnovna določila pisanja. To namreč zelo poenostavi samo sodelovanje med programerji, poleg tega pa je tudi koda razumljivejša ter njeno vzdrževanje bistveno lažje. V našem primeru so pravila, katerim je potrebno zadostiti pri pisanju naslednja:

- V kodi ne uporabljamo kratic. Vsaka spremenljivka ali metoda naj bo poimenovana tako, da je njen namen natančno določen.
- Če je ime metode ali spremenljivke sestavljeno iz več besed, naj bo prva črka vsake besede napisana z veliko začetnico, ostale črke pa z malo.
- Privatne spremenljivke razreda naj uporabljajo predpono F.
- Spremenljivke, ki pridejo v metodo preko parametrov, naj imajo predpono A.
- Pri pisanju metod je potrebno uporabljati zamike. Uporablja naj se standardni zamik.

Na podlagi pravil pisanja kode se lotimo implementacije razredov. Izberemo najbolj primerno metodo za implementacijo posamezne asociacije, vendar moramo pri tem paziti, da se osnovni vmesnik razreda ne spremeni.

Ko je opravilo pisanja kode zaključeno, se lotimo pregledovanja razredov. Preveriti moramo, če je implementacija razredov skladna z njihovo specifikacijo. Preverimo tudi, ali smo pokrili vso funkcionalnost, kot je bila zahtevana.

6.5.3 Analiza zmogljivosti

Z analizo zmogljivosti smo se predhodno ukvarjali že v fazi analize ter fazi načrtovanja. Zmogljivost pa se dejansko lahko preveri šele, ko je sistem implementiran. Za sistem določimo kritične točke, ki jih moramo preučiti. Ugotoviti moramo, če je sistem dovolj zmogljiv, da ustreza pogojem, ki so bili predhodno zavedeni. V primeru, da sistem pogojem ne ustreza, je potrebno poiskati rešitev, ki bo zadostila navedenim pogojem. Optimizacijo sistema izvajamo nad operacijami, ki se v sistemu izvedejo velikokrat in ne nad tistimi, ki se izvedejo le redko [9]. Kritične točke sistema, ki ga razvijamo ter plan analize zmogljivosti, so navedene v spodnji preglednici:

Kritične točke	Plan analize zmogljivosti
čakanje na vzpostavitev komunikacije s točko	Preveriti je potrebno, koliko časa traja od takrat, ko približamo GSM aparat k oznaki KBP, dokler aplikacija ni naložena. V primeru, da sama akcija traja več kot dve sekundi, je potrebno preveriti, katera izvajanja operacij bi lahko prestavili na kasnejši čas.
čas trajanja pošiljanja zajete registracije	Po izbiri oziroma vnosu dogodka, pošljemo registracijo strežniku KBP. Potrebno je preveriti čas trajanja samega pošiljanja registracije. Samo pošiljanje registracije je sestavljeno iz časa povezovanja, časa pošiljanja ter časa čakanja na odgovor strežnika o uspešnosti poslane registracije. Pošiljanje ne sme trajati več kot 10 sekund. V primeru, da traja sama izvedba dalj časa, je potrebno vzpostaviti povezavo s strežnikom že pred samim pošiljanjem.
prejemanje veljavnih dogodkov na točki	Potrebno je preveriti, koliko časa traja prejemanje podatkov o dogodkih na izbrani točki. Ker podatke pridobimo od strežnika, je pred tem potrebno vzpostaviti komunikacijo. To pa lahko povzroči zakasnitev. Potrebno je ugotoviti, koliko časa traja akcija ter v primeru predolгих časov omogočiti, da se uporabniku kot privzeto ponudi samodejni vnos številke dogodka.
pošiljanje večjega števila registracij ob istem času (prepustnost sistema)	Z večjim številom aparatov je potrebno preizkusiti samo prepustnost sistema. V kolikor večjega števila aparatov ne bo na voljo, je potrebno izdelati orodje, s katerim bomo lahko simulirali pošiljanje večjega števila registracij istočasno. Tako bomo lahko preverili samo odzivnost sistema tudi za primer industrijskega okolja, kjer registracije nastopajo v serijah. Potrebno je določiti kritično mejo sistema ter določiti, koliko odjemalcev naenkrat še lahko dostopa do strežnika. V primeru večjih zahtev je potrebno postaviti dve strežniški aplikaciji.

Preglednica 8: Kritične točke sistema ter plan analize zmogljivosti.

6.5.4 Pregled kode

Po zaključku implementacije sistema se lotimo še pregleda kode. Sam pregled kode je zelo koristen, saj se z njim izognemo potencialnim nepravilnostim delovanja sistema. Poleg tega pa tudi ugotovimo napake pri samem načinu razvoja. S tem izboljšamo tako sistem, kot tudi samo implementacijo razvijalca. Poznamo dve vrsti pregleda kode [9]:

- Skupni pregled kode
Cilj skupnega pregleda kode je iskanje napak ter potencialno nevarnih delov kode. So učinkovit način pregleda kode, saj oseba, ki je kodo implementirala, potencialne težave svoje kode težje odkrije.
- Testiranje
S testiranjem preverjamo pravilnost ter delovanje posameznih delov kode. V ta namen lahko izdelamo testne primere, ki jih nato preizkusimo nad implementirano kodo.

6.6 Testiranje

Cilj aktivnosti testiranja je odkriti napake v sistemu ter ugotoviti morebitne pomanjkljivosti ter težave, ki se z uporabo sistema lahko pojavijo. [9]

6.6.1 Planiranje testiranja

Pri testiranju se bomo osredotočili na nekaj glavnih parametrov kakovosti, ki so zahtevani za sistem zajema registracij z uporabo KBP. Pri testiranju je potrebno zajeti:

- **Pravilnost delovanja**
Cilj testiranja pravilnosti delovanja je preveriti delovanje sistema pri različnih naborih vhodnih parametrov. Preveriti je potrebno delovanje funkcionalnosti sistema tako pri pozitivnih scenarijih, kot tudi pri alternativni. Potrebno je preizkusiti odzivnost sistema pri več sočasnih registracijah, tako na različnih točkah, kot na eni sami.
- **Hitrost**
V večini podjetij ljudje z delovnega mesta odhajajo ali prihajajo ob istem času. Zato je odzivnost sistema pri zajemu registracije zelo pomembna. Hitrost je tako eden glavnih parametrov kakovosti. Pri testiranju je poseben poudarek namenjen času posamezne registracije, času več zaporednih registracij in času obdelave več sočasnih registracij z ene ali več točk.
- **Robustnost sistema**
Sistem za zajem registracij mora biti robusten ter odporen na napake, katere se v sistemu lahko pojavijo. Ne sme se zgoditi, da bi določena nepravilnost povzročila izpad sistema za daljši čas. V primeru izpada sistema je potrebno preveriti tudi ali se registracija začasno shrani ter ali jo je možno poslati kasneje.
- **Zaščita**
Varnost in zaščita sta postala pomembna parametra informacijskih sistemov. Preveriti je potrebno delovanje zaklepanja točke za pisanje ter zaščito pri napadih, ki so mogoči pri tehnologiji komunikacije v bližnjem polju.

Za sistem, ki ga razvijamo, bo testiranje izvedla testna ekipa podjetja Špica International d.o.o. Plan testiranja bo zajemal tako testiranje posameznih modulov sistema, kot tudi testiranje celotnega sistema. Test posameznih komponent sistema se bo izvedel takoj, ko bo zaključena aktivnost implementacije. Ko bo sistem izdelan v celoti, se bo izvedel še potrditveni test celotnega sistema.

6.6.2 Oblikovanje testnih primerov

Z modela primerov uporabe izberemo funkcionalnost sistema, ki jo želimo testirati. Za vsako od izbranih funkcionalnosti, katero želimo testirati, izdelamo plan ter model testiranja. Plan testiranja poda zadolžitve ter časovne okvire izvedbe testiranja. Plan testiranja lahko zelo nazorno predstavimo z gantovim diagramom. V modelu testiranja zajamemo podatke o tem, kaj želimo testirati, kakšni so želeni podatki ter kakšni so vhodni pogoji. Podrobni opis scenarija testiranja se ponavadi odda na posebnem listu. [9]

Oznaka modula:	zajem registracije.
Oznaka scenarija:	ZA-1.
Opis scenarija:	zajem registracije z uporabo tehnologije KBP.
Kriteriji pravilnosti delovanja modula:	<ol style="list-style-type: none">1. Ob zagonu aplikacije se avtomatsko prebere tudi številka točke.2. Možno je izbrati dogodke s seznama.3. Registracija se zapiše v sistem.4. Ko strežnik ne deluje, se registracija shrani v pomnilnik telefona.
Opis množice vhodnih podatkov:	<ul style="list-style-type: none">• Naslov strežnika je nastavljen na http://www.spica.com/Test_NFC/.• Številka točke 15.• Številka dogodka 66 – prihod.• Uporabnik je v sistemu povezan z uporabnikom številka 780.

Preglednica 9: Model testiranja za scenarij zajema registracije

Podroben opis scenarija zajema registracije

Oznaka scenarija: ZA-1.

Opis scenarija: Zajem registracije z uporabo tehnologije KBP.

Ko telefon z uporabniško aplikacijo približamo točki KBP, se mora zagnati uporabniška aplikacija. Ob zagonu se mora prebrati tudi številka točke.

Na dialogu, ki se odpre, je treba preveriti:

- Ali se dogodki na točki dobijo iz strežnika ter so prikazani v drsečem seznamu.
- Če naslov strežnika ni nastavljen ali pa je strežnik nedosegljiv, je možnost ročnega vnosa številke dogodka.

Nastavimo vrednost dogodka in pritisnemo tipko pošlji. Nato preverimo:

- Če je povratna informacija o poslani registraciji potrjena. Preverimo v sistemu Time&Space, ali je registracija resnično zabeležena.
- Če je povratna informacija negativna, preverimo, ali se je registracija shranila. Pri pravilno poslani registraciji preverimo, ali se je zabeležila k pravi osebi, ali je bil zabeležen pravi dogodek, prava točka ter pravi čas dogodka.

Vse moteče opazke, predolge čase čakanja, pripombe na izgled dialoga zapišite kot komentar k testu.

Kriterij uspešnosti:

Vse postavke v scenariju se morajo izvesti pravilno.

Slika 16: Podroben opis scenarija zajema registracije

Oznaka modula:	Spreminjanje oznake KBP.
Oznaka scenarija:	SP-1.
Opis scenarija:	Spreminjanje identitete oznaki KBP, pri predhodno nastavljeni zaščiti.
Kriteriji pravilnosti delovanja modula:	<ol style="list-style-type: none"> 1. Nastavimo nov id oznake in pritisnemo »nastavi«. Prikaže se dialog za vnos gesla, nastavljenega na točki. 2. Pravilen vnos gesla. Identiteta se spremeni, izpiše se »Spremenjeno«. 3. Nepravilni vnos gesla, izpiše se »Napačno geslo«.
Opis množice vhodnih podatkov:	<ul style="list-style-type: none"> • oznaka KBP, zaščiten z geslom, • izbrano geslo, • nov id oznake 15.

Preglednica 10: Model testiranja za scenarij Spreminjanje identitete oznaki KBP

6.6.3 Implementacija testnih primerov

Tako jezik Java (JUnit), kot tudi .NET (NUnit), nudita podporo za izdelavo testnih razredov. Z njimi uvedemo postopek avtomatizacije testiranja, saj z njimi hitro ter enostavno lahko izvedemo teste nad kodo, katero testni razredi pokrivajo. Tako se lahko izognemo napakam, ki pridejo z evolucijo razvoja razredov ter metod. Željeno je, da napišemo čim večje število testnih razredov, saj tako z njimi pokrijemo večji del kode ter tako naredimo korak naprej k zanesljivosti ter pravilnosti delovanja sistema. Pri izdelavi testnih primerov moramo poleg testiranja splošnih primerov izvesti tudi testiranje robnih pogojev posameznih metod.

6.6.4 Izvajanje testiranja

Testiranje izvedemo po predhodno dogovorjenem načrtu. Pri sistemu za zajem registracij z uporabo tehnologije KBP bo test izvedla testna ekipa oddelka Time&Space. Testiranje z uporabo testnih razredov bodo predhodno izvedli razvijalci sami ter ga ponovili ob vsakih nadaljnjih popravkih.

Testerji morajo na svojih računalnikih naprej postaviti potrebno testno okolje za samo izvedbo testa. Namestiti si morajo aplikacijo Time&Space, postaviti testno bazo MS SQL ter v sam sistem vnesti podatke, ki določeno osebo iz sistema za registracijo delovnega časa poveže z osebo, ki bo izvajala registracije preko mobilnega aparata. Postaviti je potrebno tudi strežniško aplikacijo Time&Space KBP, ki bo skrbela za vnos podatkov v sam sistem ter iz njega pridobivala podatke. Na GSM aparat Nokia 6131 je potrebno naložiti tako uporabniško kot administratorsko napravo. Na aparatu je potrebno nastaviti, da je privzeta aplikacija ob zaznani napravi KBP uporabniška aplikacija. Potrebno je priskrbeti oznake KBP, katere bomo pri testu uporabljali.

Na podlagi testnega plana testerji razberejo, katere teste morajo izvesti. Scenariji testiranja se nahajajo na platformi »Test«. Vse rezultate testiranja nato primerno zavedejo na platformi ter ob zaključku testiranja posameznega testa označijo test kot zaključen.

Ko so vsi testni primeri uspešno zaključeni, testna ekipa izvede še prevzemno testiranja. To testiranje zajema preverjanje izpolnjevanja prevzemnih kriterijih, ki so bili predhodno zajeti. V skladu s tem pripravimo testne primere.

6.6.5 Vrednotenje testiranja

Razvojna ekipa skupaj pregleda teste, ki imajo status »zaključeno«. V primeru, da do bile v testu ugotovljene nepravilnosti oziroma pomanjkljivosti, razvojna ekipa le-te analizira. Vsa odstopanja je potrebno dokumentirati ter zanje navesti razlog. V razvojnem oddelku podjetja Špica d. o. o., se ugotovljene nepravilnosti beležijo na platformo »bug«. Ekipa nato za vse ugotovljene neskladnosti poišče vzrok ter za njih določi korektivne ukrepe, ali pa jih v primeru, da je nepravilnost zanemarljiva, samo zavede v dokument [9].

7 Zaključek

V diplomski nalogi sem predstavil informacijsko rešitev za zajem registracije delovnega časa z uporabo tehnologije v bližnjem polju. Razvoj informacijskega sistema sem izvajal po aktivnostih, ki jih narekuje enotna metodologija razvoja informacijskega sistema – EMRIS. Skozi sam razvoj sem ugotovil, da se določene zahteve, ki vplivajo na samo rešitev, dejansko pokažejo šele pri načrtovanju. Pri dobro načrtovanem sistemu te zahteve relativno enostavno vključimo v sistem in ne povzročijo težav pri sami izgradnji sistema. Poleg tega imamo ob zaključku sistema zelo dobro dokumentiran sistem, kar omogoča precej lažje vzdrževanje v prihodnosti.

Rešitev, ki sem jo v diplomski nalogi razvil, predstavlja nov način za registracijo zaposlenih v podjetju. Registracija na izbrani način uporablja sodobne tehnologije in je izvedena na preprost način. Poleg tega pa uporabniku tudi nudi pridobivanje zelenih informacij iz sistema. Sistem nudi še precej prostora za razširitve. Prvo področje, ki bi se mu bilo pri nadaljnjem razvoju informacijskega sistema potrebno posvetiti, je varnost in zaščita. Potrebno bi bilo kar najbolje zaščititi sistem pred nevarnostmi, ki grozijo ob uporabi tehnologije KBP. Poleg tega bi preko GSM aparata uporabnik lahko pregledoval bistveno večjo količino informacij, ki se hranijo v sistemu. Preko aparata bi mu lahko omogočili tudi možnost vnosa napovedi dopusta.

Možnost dejanske uporabe rešitve v realnem svetu je po mojem mnenju trenutno zelo malo verjetna. Razlog za tako oceno tiči predvsem v omejenosti tehnologije v današnjem času. V svetu je trenutno na voljo le nekaj telefonov, ki podpirajo tehnologijo KBP. Ta omejitev vodi do zahteve, da morajo imeti vsi delavci nekega podjetja enak telefon. To pa je praktično nemogoča zahteva. Če upoštevamo napovedi o širitvi tehnologije v večje število aparatov, ima razvita informacijska rešitev večje možnosti za uporabo v prihodnosti.

Rešitev z uporabo KBP, ki bi bila lahko uporabna tudi v trenutnem času, vidim v obratni postavitvi sistema. Vsak uporabnik bi imel pri sebi oznako KBP, na kateri bi bil zapisan njegov identifikator. Ob vходу bi bil pritrjen aparat s podporo tehnologiji KBP. Uporabnik bi oznako približal aparatu, ta pa bi z nje prebral njegovo identiteto ter mu ponudil dogodke, ki jih lahko registrira. Uporabnik bi dogodek izbral, registracija pa bi se nato poslala v sistem. Tako bi se izognili omejitvi, ki od uporabnika zahteva, da ima točno določen GSM aparat.

8 Priloge

8.1 Seznam slik

Slika 1: Oznaka KBP	6
Slika 2: GSM aparat s podporo KBP – Nokia 6131	8
Slika 3: Umestitev rešitve zajema registracij z uporabo KBP v celostno rešitev sistema za obdelavo registracij delovnega časa produkta Time&Space	12
Slika 4: Diagram primerov uporabe za sistem zajema registracij z uporabo tehnologije KBP.....	23
Slika 5: Diagram zaporedja za primer uporabe »registracija uporabnika s tehnologijo KBP«.....	24
Slika 6: prototip uporabniškega vmesnika uporabniške aplikacije	25
Slika 7: Prototip uporabniškega vmesnika administratorske aplikacije.....	26
Slika 8: Konceptualni razredni diagram sistema.....	29
Slika 9 Diagram zaporedja za primer uporabe »Zajem registracij«.....	31
Slika 10: Diagram sodelovanja za primer uporabe »Zajem registracij«	31
Slika 11: Razredni diagram sistema za zajem registracij delovnega časa z uporabo KBP	32
Slika 12: Diagram komponent ter njihovo sodelovanje	36
Slika 13: Razredni diagram aktivnosti načrtovanja objektov	37
Slika 14: Zaporedni diagram izvajanja algoritma za pošiljanje registracij.	38
Slika 15: Diagram aktivnosti algoritma za spreminjanje identitete točki	39
Slika 16: Podroben opis scenarija zajema registracije	46

8.2 Seznam preglednic

Preglednica 1: Seznam oseb, ki so vključene pri projektu razvoja IS za zajem registracij z uporabo KBP	19
Preglednica 2: Atributi ter operacije nad posameznimi koncepti domene.....	28
Preglednica 3: Realizacijski razredi za posamezne primere uporabe sistema za zajem registracij.....	30
Preglednica 4: Operacije sistema, njihove lastnosti ter razredi, ki jih implementirajo	33
Preglednica 5: Opisi komponent ter njihovi realizacijski razredi	34
Preglednica 6: Dostopi ter čas trajanje povezav	40
Preglednica 7: Določitev začetnega stanja atributom.	41
Preglednica 8: Kritične točke sistema ter plan analize zmogljivosti.....	43
Preglednica 9: Model testiranja za scenarij zajema registracije.....	46
Preglednica 10: Model testiranja za scenarij Spreminjanje identitete oznaki KBP	47

9 Viri in literatura

- [1] About NFC. Dostopno na: <http://www.nfc-forum.org/aboutnfc/>
- [2] About NFC technology. Dostopno na: <http://www.nfc-forum.org/resources/faqs#headTechnology>
- [3] Access Control and Time Attendance. Dostopno na: <http://www.spica.com>
- [4] Data transmission rates. Dostopno na: <http://www.nfc-forum.org/resources/faqs#data>
- [5] Diffie-Hellman key exchange. Dostopno na: <http://en.wikipedia.org/wiki/Diffie-Hellman>
- [6] M. Fowler, »Patterns of Enterprise Application Architecture«, Pearson, 2002
- [7] How does NFC technology work. Dostopno na: <http://www.nfc-forum.org/resources/faqs#howwork>
- [8] U. Kočevar, *Uporabniška dokumentacija Sistema Time & Space*, Ljubljana: Špica International, d.o.o., 2007
- [9] M. Krisper, R. Rupnik, M. Bajec, ..., »Enotna metodologija razvoja informacijskih sistemov«, 4. Zvezek, Ljubljana, 2000
- [10] M. Krisper, R. Rupnik, M. Bajec, ..., »Uvod, Enotna metodologija razvoja informacijskih sistemov«, druga izdaja, december 2003
- [11] NFC Forum. Dostopno na: <http://www.nfc-forum.org/aboutus/>
- [12] NFC Specifications. Dostopno na: <http://www.nfc-forum.org/specs/>
- [13] Registracija delovnega časa. Dostopno na: <http://www.jantar.si/site.php?str=sol&pg=02&ref=lm>
- [14] Security in Near Field Communication (NFC). Dostopno na: <http://events.iaik.tugraz.at/RFIDSec06/Program/papers/002%20-%20Security%20in%20NFC.pdf>
- [15] Unified Modeling Language. Dostopno na: http://en.wikipedia.org/wiki/Unified_Modeling_Language

Izjava o avtorstvu

Izjavljam, da sem diplomsko delo izdelal samostojno pod vodstvom mentorja prof. dr. Marjana Krisperja. Izkazano pomoč drugih sodelavcev sem navedel v zahvali.

Prevoje, september 2008

Marko Klopčič