

UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN
INFORMACIJSKE TEHNOLOGIJE

ZAKLJUČNA NALOGA
RAZVOJ SAP FIORI APLIKACIJE V IZBRANEM
PODJETJU COMCOM D.O.O.

TADEJ MAJNIK

UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN
INFORMACIJSKE TEHNOLOGIJE

Zaključna naloga

Razvoj SAP Fiori aplikacije v izbranem podjetju Comcom d.o.o.

(SAP Fiori application development in the selected company Comcom d.o.o.)

Ime in priimek: Tadej Majnik

Študijski program: Računalništvo in informatika

Mentor: doc. Dr. Peter Rogelj

Delovni Somentor: univ. dipl. inž. rač. in inf. Miha Strel

Koper, maj 2022

Ključna dokumentacijska informacija

Ime in PRIIMEK: Tadej MAJNIK

Naslov zaključne naloge: Razvoj SAP Fiori aplikacije v izbranem podjetju Comcom d.o.o.

Kraj: Koper

Leto: 2022

Število listov: 45 Število slik: 23

Število referenc: 15

Mentor: doc. Dr. Peter Rogelj

Delovni somentor: univ. dipl. inž. rač. in inf. Miha Strel

Ključne besede: SAP, Fiori, aplikacija, HTML5, ABAP, JavaScript, plačilne liste, uporabniški vmesnik, Programsko inženirstvo

Izvleček:

Cilj te zaključne naloge je opis celotnega programskega procesa pri razvoju SAPUI5 Fiori aplikacije, vključujoč faze analize, načrtovanja, izvedbe, integracije in testiranja. Fiori aplikacija se bo imenovala "Pregled plačilnih list". Namen aplikacije je pregled uporabnikovih plačilnih list glede na izbrano časovno obdobje z enostavnim, a hkrati uporabnim in intuitivnim uporabniškim vmesnikom. Aplikacija bo imela možnost predogleda dokumenta v brskalniku, možnost pošiljanja dokumenta na e-poštni naslov in možnost oddaje pritožbe na izbrano plačilno listo.

Key document information

Name and SURNAME: Tadej MAJNIK

Title of the final project paper: SAP Fiori application development in the selected company Comcom d.o.o.

Place: Koper

Year: 2022

Number of pages: 45 Number of figures: 23

Number of references: 15

Mentor: Assistant Professor Peter Rogelj, PhD

Work Co-Mentor: univ. dipl. inž. rač. in inf. Miha Strel

Keywords: SAP, Fiori, application, HTML5, ABAP, JavaScript, payslips, user interface, Software engineering

Abstract:

In my thesis, I will present the complete process of developing an SAP Fiori application, including the phase of analysis, planning, implementation, integration and testing. The application will be named »Pregled plačilnih list« which stands for »Overview of payslips«. The aim of this application is an overview of employee's own payslips in the selected time period with a simple, yet useful and intuitive user interface. It will have the option to preview the document in browser, send it over e-mail and submit a complaint about the selected payslip document.

ZAHVALA

Posebna zahvala gre mentorju doc. Dr. Petru Roglju in delovnemu somentorju univ. dipl. inž. Rač. in inf. Mihi Strelu. Prav tako bi se rad zahvalil podjetju Comcom d.o.o. za omogočanje dela zaključne naloge ter uporabo njihovih SAP sistemov in druge programske opreme. Zahvala gre tudi Zofiji Majnik za pomoč pri iskanju slovničnih in pravopisnih napak.

KAZALO VSEBINE

1 UVOD.....	1
2 DEFINICIJA PROBLEMA.....	2
3 SLABOSTI VMESNIKA SAP GUI IN PRIDOBITEV Z UPORABO SAP FIORI APLIKACIJE	3
3.1 Slabosti	3
3.2 Pridobitve	3
4 ŠTUDIJA IZVEDLJIVOSTI.....	4
4.1 Tehnični vidik.....	4
4.1.1 Podprte uporabniške naprave.....	4
4.1.2 Omejitve uporabniškega vmesnika	4
4.1.3 Dovoljenja in dostop do strojne opreme	4
4.2 ORGANIZACIJSKI VIDIK	5
4.2.1 Profesionalni deležniki moje fiori aplikacije.....	5
4.2.2 Pogodba EULA	5
4.3 EKONOMSKI VIDIK.....	6
4.3.1 Prednosti za podjetje	6
5 ANALIZA, DEFINIRANJE IN MODELIRANJE ZAHTEV	7
5.1 Namen aplikacije	7
5.2 Pregled plačilnih list	8
5.3 Predogled dokumenta v brskalniku	9
5.4 Pošiljanje na e-poštni naslov	10
5.5 Oddaja pritožbe na plačilno listo	11
5.6 Nefunkcijske zahteve.....	12
5.6.1 Varnost uporabnikovih podatkov.....	12
5.6.2 Razpoložljivost.....	12
5.6.3 Tehnične zahteve kode in implementacije.....	12
5.6.4 Uporabnost in preglednost vmesnika	12
5.6.5 Učinkovitost delovanja	12
5.6.6 Opcija izbire jezika.....	13
5.6.7 Podpora mobilnih naprav	13

6 NAČRTOVANJE SISTEMA	14
6.1 Definiranje načrtovalskih ciljev.....	14
6.1.1 Učinkovitost delovanja	14
6.1.2 Zanesljivost.....	14
6.1.3 Vzdrževanje	15
6.1.4 Uporabniške zahteve.....	15
6.2 Dekompozicija sistema na podsisteme/komponente	16
6.3 Izbira že izdelanih komponent.....	17
6.4 Upravljanje s perzistentnimi podatki.....	17
6.5 Določitev politike nadzora dostopa	18
7 IZVEDBA.....	19
7.1 Izbira programskih jezikov in razvojnih orodij	19
7.2 Program za pripravo plačilnih list	19
7.2.1 Nabiranje podatkov plačilne liste v transakciji HRFORMS	19
7.2.2 Izdelava logike in oblikovanja obrazca za plačilne liste.....	20
7.2.3 Prilagoditev razreda CL_HRFORMS_HRF02.....	20
7.2.4 Priprava arhiviranja na Comcom zalednem SAP sistemu.....	20
7.2.5 Kreiranje tabele metapodatkov	21
7.2.6 Prevodi plačilnih list	21
7.2.7 Arhiviranje plačilnih list.....	22
7.3 Izdelava oData storitve	23
7.3.1 Definiranje oData storitev	24
7.3.2 Branje seznama plačilnih list	24
7.3.3 Prenos dejanskega PDF dokumenta v oblični sistem	24
7.3.4 Pošiljanje plačilne liste zaposlenemu prek e-pošte.....	25
7.3.5 Pošiljanje pritožbe v kadrovske službo.....	25
7.4 Registracija oData storitve in dodajanje avtorizacij	26
7.5 SAPUI5 Fiori aplikacija	26
7.5.1 Osnovna struktura aplikacije.....	26
7.5.2 Glavna stran aplikacije	27

7.5.3 Branje seznama plačilnih list	27
7.5.4 Prenos dokumenta na oblični sistem in njegov predogled	28
7.5.5 Prikazovanje obvestil o uspešnem izvajanju aplikacije	28
7.5.6 Prevodi uporabniškega vmesnika aplikacije.....	28
8 TESTIRANJE APLIKACIJE	29
8.1 Pregled plačilnih list	29
8.2 Pošiljanje dokumenta na e-poštni naslov	31
8.3 Predogled dokumenta v brskalniku	32
8.4 Oddajanje pritožbe na plačilno listo	32
9 MOŽNE KASNEJŠE DOPOLNITVE IN RAZŠIRITVE APLIKACIJE	33
10 ZAKLJUČEK	34
11 VIRI IN LITERATURA.....	35

KAZALO SLIK IN GRAFIKONOV

Slika 1: Primer začetnega ekrana uporabniškega vmesnika SAP GUI	2
Slika 2: Diagram primerov uporabe aplikacije.....	7
Slika 3: Skica glavne strani aplikacije.....	8
Slika 4: Skica pregleda plačilne liste ter izgled same plačilne liste	9
Slika 5: Primer prejetega e-poštnega sporočila	10
Slika 6: Skica pogovornega okna za oddajo pritožbe.....	11
Slika 7: Skica aplikacije na mobilni napravi	13
Slika 8: Skica okna za oddajo pritožbe na mobilni napravi	13
Slika 9: Diagram postavitve Fiori aplikacije	16
Slika 10: Prikaz Fiori portala in ploščice za zagon aplikacije.....	18
Slika 11: Struktura tabele metapodatkov.....	21
Slika 12: Primer selekcijskega ekrana za pravilno arhiviranje.....	22
Slika 13: Struktura entitetnega tipa za seznam plačilnih list.....	23
Slika 14: Struktura entitetnega tipa za prenos datoteke.....	23
Slika 15: Struktura entitetnega tipa za pošiljanje e-poštnega sporočila	24
Slika 16: Primer standardnega teksta v transakciji SO10.....	25
Slika 17: Testiranje pregleda plačilnih list za kadrovske 956.....	29
Slika 18: Testiranje pregleda plačilnih list z izbiro časovnega obdobja	30
Slika 19: Testiranje pregleda plačilnih list pri manjkajočih podatkih.....	31
Slika 20: Obvestilo o uspešno poslanem e-poštnem sporočilu.....	31
Slika 21: Obvestilo o napaki pri pošiljanju e-poštnega sporočila	32
Slika 22: Obvestilo o uspešno vloženi pritožbi	32
Slika 23: Obvestilo o napaki pri vlaganju pritožbe	32

SLOVARČEK TUJK IN KRATIC

- ABAP (Advanced business application programming language) → Programski jezik, namenjen razvoju naprednih poslovnih aplikacij.
- JavaScript → Visoko-nivojski spletni programski jezik
- XML (Extensible markup language) → Označevalni jezik, ki opredeljuje niz opravil za kodiranje dokumentov v obliki, ki je lahko berljiva za ljudi.
- SAP (System Applications and Products in Data Processing) → Sistemske aplikacije in produkti za procesiranje podatkov
- ERP (Enterprise resource planning) → Sistem za načrtovanje virov podjetja
- GUI (Graphical user interface) → Grafični uporabniški vmesnik
- HR (Human resources) → Kadrovska služba
- VPN (Virtual private network) → virtualno zasebno omrežje
- EULA (End user licence agreement) → Licenčna pogodba za končnega uporabnika
- GDPR (General Data Protection Regulation) → Splošna uredba o varstvu podatkov
- HTML5 → Najnovejša različica označevalnega jezika HTML
- SAPUI5 → Ogradje za izdelavo spletnih aplikacij, ki temelji na HTML5
- SAP Web IDE (integrated development environment) → Spletno integrirano razvojno okolje
- oData (Open Data Protocol) → odprti podatkovni protokol
- UML (Unified Modeling Language) → Računalniški standard za modeliranje računalniških sistemov in procesov
- PDF (Portable Document Format) → Odprt standard za izmenjavo elektronskih dokumentov
- Transakcija → posamezen logični proces v SAP sistemu

1 UVOD

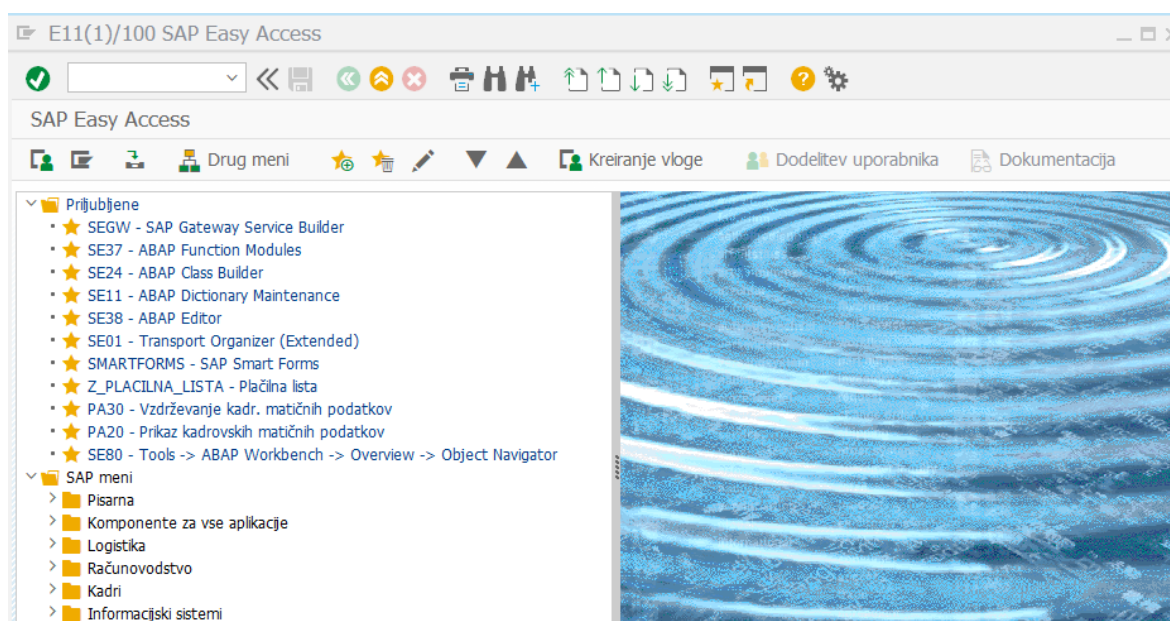
SAP[11] je eden največjih ERP (Enterprise resource planning) sistemov na svetu, ki pokriva vsa področja poslovnih procesov, od upravljanja s človeškimi viri pa do planiranja, načrtovanja ter upravljanja z materialom in časom. Razvilo ga je Nemško podjetje Systemanalyse Programmentwicklung, kasneje skrajšano na SAP. Začetki podjetja segajo vse do leta 1972. Podjetje ja danes znano pod imenom SAP SE. Dandanes podjetje zaposluje več kot 105.000 ljudi po celem svetu. Tak ERP sistem je zelo koristen, še posebej pri velikih podjetjih, ki imajo veliko število zaposlenih. Za razvijanje programov v SAP-ju se uporablja programski jezik ABAP. ABAP[9] je visoko-nivojski programski jezik, namenjen programiranju naprednih poslovnih aplikacij izključno v SAP sistemih.

SAP Fiori[10] je spletna platforma, ki predstavlja novejši in modernejši uporabniški vmesnik s SAP sistemom, hkrati pa je enostavnejši in zato lažji za uporabo. Fiori platforma poleg osebnih računalnikov podpira tudi dostop z drugih naprav, kot so tablice in pametni telefoni. Poudarek je na podpori naprav z upravljanjem na dotik. Poznamo tri tipe Fiori aplikacij: transakcijske in analitične aplikacije ter informativne liste (v izvorniku Transactional, Analytical, fact sheets).

V tej zaključni nalogi bom prikazal implementacijo transakcijske aplikacije. Namen aplikacije bo pregled uporabnikovih plačilnih list. Za izbiro take teme sem se odločil, ker trenutno delam pri podjetju Comcom, ki specializira v SAP svetovanju ter z ekipo izkušenih svetovalcev in razvijalcev sodeluje pri implementaciji integriranega poslovnega informacijskega sistema SAP.

2 DEFINICIJA PROBLEMA

Standardni uporabniški vmesnik sistema SAP je SAP GUI[13], ki pa je že zastarel, okoren, uporabniku neprijazen na pogled, zaradi prenasičenosti ikon in gumbov pa je lahko za nove uporabnike precej zapleten, zato je potrebno precej časa, preden se ga naučijo učinkovito uporabljati. Platforma SAP Fiori pa predstavlja nov, modernejši in uporabniku bolj prijazen uporabniški vmesnik. Čas, ki je potreben za učenje vmesnika SAP Fiori aplikacije, je občutno krajši kot pri standardnem SAP vmesniku.



Slika 1: Primer začetnega ekrana uporabniškega vmesnika SAP GUI

Menim, da bi z uporabo Fiori platforme ter odzivne in uporabniku prijazne aplikacije poenostavili in pohitrili nekatera opravila kot npr. pregled in pošiljanje plačilnih list uporabnika. Uporabnik bo lahko imel na enem mestu pregled nad vsemi svojimi plačilnimi listami. Že sama platforma Fiori daje velik poudarek na preprostosti vmesnika, a hkrati tudi na uporabnosti za končnega uporabnika.

Cilj te zaključne naloge je opis celotnega programskega procesa pri razvoju SAP Fiori aplikacije, vključujoč faze analize, načrtovanja, izvedbe, integracije in testiranja. Namen aplikacije je prikaz posameznikovih plačilnih list, ki so že obračunane v SAP sistemu. Če so prikazane plačilne liste že arhivirane, si jih lahko uporabnik tudi ogleda, pošlje na svoj e-poštni naslov ali pa nanje vloži pritožbo.

3 SLABOSTI VMESNIKA SAP GUI IN PRIDOBITEV Z UPORABO SAP FIORI APLIKACIJE

3.1 Slabosti

Standardni uporabniški vmesnik SAP GUI je že zelo zastarel, okoren za uporabo in za uporabnika na videz neprivlačen. Zaradi prenasičenosti ikon, prikazanih na vmesniku, je velikokrat po sistemu težko krmariti, sploh za nove uporabnike. Zapletenost vmesnika SAP GUI zato za učinkovito uporabo zahteva daljši čas izobraževanja uporabnikov. Uporabniški vmesnik SAP GUI predstavlja manjšo mobilnost, ker lahko na njem delamo samo preko osebnega računalnika.

3.2 Pridobitve

SAPUI5[10] Fiori aplikacija, vsebuje nov, modernejši in poenostavljen uporabniški vmesnik, ki je lepši na pogled, preglednejši in zato tudi lažji za uporabo za končnega uporabnika. S tem lahko pohitrimo nekatera opravila. Z uporabo modernega programskega jezika JavaScript[4] in modernih HTML5[8] brskalnikov je izdelava uporabniškega vmesnika take aplikacije zelo preprosta. Zaradi preprostejšega vmesnika se čas izobraževanja uporabnikov za uporabo aplikacije občutno skrajša. Aplikacije na SAP Fiori platformi in sama platforma predstavljajo večjo mobilnost, saj jih lahko poleg osebnih računalnikov uporabljamo tudi na mobilnih telefonih in tablicah. Fiori platforma je tudi optimizirana za uporabo na napravah z upravljanjem na dotik.

4 ŠTUDIJA IZVEDLJIVOSTI

4.1 Tehnični vidik

4.1.1 Podprte uporabniške naprave

Moja Fiori aplikacija za pregled plačilnih list bo prvotno namenjena uporabi na osebnih računalnikih, seveda pa bo omogočena uporaba aplikacije tudi na tablicah in pametnih telefonih, saj Fiori platforma to omogoča. Izgled aplikacije in razmerje zaslona se bosta prilagodila zaslonu naprave uporabnika, na osebem računalniku bo to največkrat 16:9, na novejših pametnih napravah pa bodo tudi druge. Na osebnih računalnikih se bo aplikacija izvajala v brskalniku (Google Chrome, Edge Chromium,...), na mobilnih napravah pa se bo izvajala v mobilni aplikaciji, lahko pa tudi v brskalniku.

4.1.2 Omejitve uporabniškega vmesnika

Pri rešitvi s portalom Fiori se soočam s številnimi omejitvami. Zaradi poenostavljenega uporabniškega vmesnika portala Fiori je kompleksnost rešitve manjša, kar pomeni, da bo imel uporabnik na voljo manjše število funkcionalnosti, kot če bi imel pregled nad plačilnimi listami direktno v SAP sistemu. Pri izdelavi Fiori aplikacije je tudi oteženo razhroščevanje kode. V primeru napak pri izvajanju logike na zalednem sistemu je težko ugotoviti točno, za katero napako gre, kar pripomore k daljšemu času odpravljanja napake.

4.1.3 Dovoljenja in dostop do strojne opreme

Rešitev Fiori bo potrebovala dostop do interneta, saj potrebuje povezavo z zalednim sistemom. Prav tako aplikacija potrebuje dostop do datotek na uporabnikovi napravi za prikazovanje in shranjevanje dokumentov. Za vse to, skupaj z drugimi dostopi, poskrbi že brskalnik ali Fiori mobilna aplikacija na uporabnikovi mobilni napravi, v katerem se aplikacija izvaja.

4.2 ORGANIZACIJSKI VIDIK

4.2.1 Profesionalni deležniki moje fiori aplikacije

Edini profesionalni deležnik in lastnik Fiori aplikacije za pregled plačilnih list je vodstvo podjetja Comcom d.o.o., uporabniki pa so zaposleni omenjenega podjetja.

4.2.2 Pogodba EULA

Pogodba EULA(Licenčna pogodba za končnega uporabnika)[\[3\]](#) bi vsebovala podatke o lastniku / avtorju aplikacije in podatke o končnih uporabnikih aplikacije. Vsebovala bi tudi vse podatke o pravilni uporabi aplikacije. Fiori aplikacija bi se lahko uporabljala samo v sklopu podjetja za interno uporabo. Določenemu končnemu uporabniku se lahko prepove nadaljnja uporaba aplikacija, če je najdena kršitev splošnih pogojev pogodbe EULA. Aplikacija je v lasti podjetja Comcom d.o.o., zato si pridrži vse pravice.

Pogodba EULA bi vsebovala tudi pogoje o pridobitvi, obdelovanju, shranjevanju in prikazovanju osebnih podatkov uporabnika. Prav tako bi veljala tudi splošna uredba o varstvu podatkov GDPR[\[6\]](#), ki ga je določila Evropska Unija.

Lastnik / avtor aplikacije ni odgovoren za nastalo škodo ali morebitno nepravilno delovanje programa, če uporabnik poganja aplikacijo na napravah, ki niso podprte, oziroma na napravah, na katerih uporaba ni predvidena in / ali če končni uporabnik aplikacijo uporablja na način, ki ni bil predviden v splošnih pogojih uporabe.

4.3 EKONOMSKI VIDIK

4.3.1 Prednosti za podjetje

Podjetje bo imelo veliko število prednosti z uporabo Fiori aplikacije. Prva velika prednost je preprost in pregleden uporabniški vmesnik za pregled uporabnikovih plačilnih list. Prav tako je lažje pošiljanje dokumenta preko elektronske pošte in shranjevanje datotek na napravo uporabnika.

Druga prednost razvoja take Fiori aplikacije je možnost uporabe njenega ogrodja za razvoj podobnih aplikacij za pregled in izpis številnih drugih kadrovskih dokumentov zaposlenega, npr. izpis vzdrževanih družinskih članov, izpis odmere letnega dopusta, dohodnine itd. To je možno narediti z relativno malo dela. Ceno razvoja takih aplikacij bi potemtakem lahko drastično znižali.

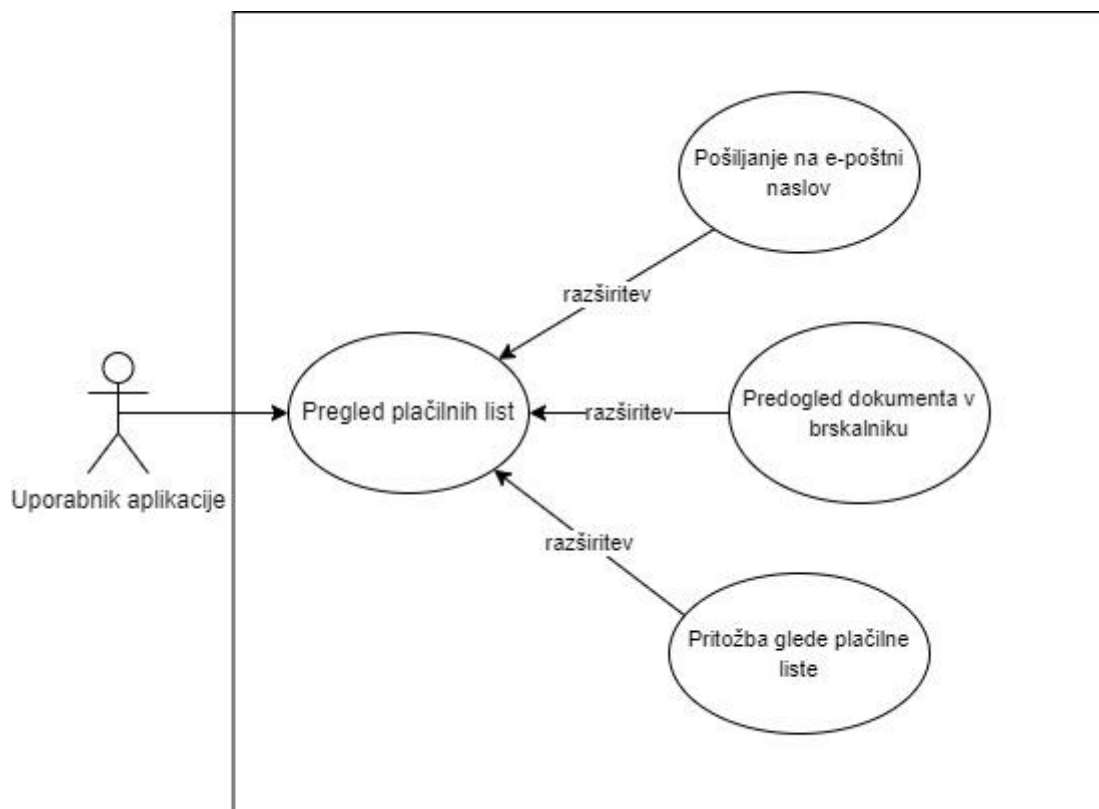
Prednost za podjetje je tudi ta, da bi lahko strukturo in logiko aplikacije kasneje uporabili še za druga podjetja, s katerimi podjetje Comcom d.o.o. posluje, v kolikor bi hoteli imeti aplikacijo za prikazovanje uporabnikovih plačilnih list.

5 ANALIZA, DEFINIRANJE IN MODELIRANJE ZAHTEV

V tej fazi zaključne naloge bom definiral, analiziral in modeliral visoko-nivojske zahteve končnega uporabnika. Za prikaz primerov uporabe bom uporabil UML diagram primerov uporabe [7]. Diagrame bom za to zaključno nalogo modeliral s programsko opremo Draw.io. Vsak primer uporabe bom tudi posebej opisal.

5.1 Namen aplikacije

Glavni namen moje Fiori aplikacije je poenostavljen in pregleden uporabniški vmesnik za pregled nad plačilnimi listami uporabnika. Preko nje si bo lahko uporabnik filtriral plačilne liste po časovnem obdobju, jih ogledal v brskalniku, prenesel na svojo napravo, poslal na svoj e-poštni naslov, omogočena pa mu bo tudi pritožba na posamezno plačilno listo. Oddano pritožbo bo administrator sistema / kadrovik prejel po elektronski pošti na svoj službeni e-poštni naslov. Poenostavljen uporabniški vmesnik Fiori aplikacije pohitri funkcije, za katere bi v SAP sistemu porabili dosti več časa.



Slika 2: Diagram primerov uporabe aplikacije

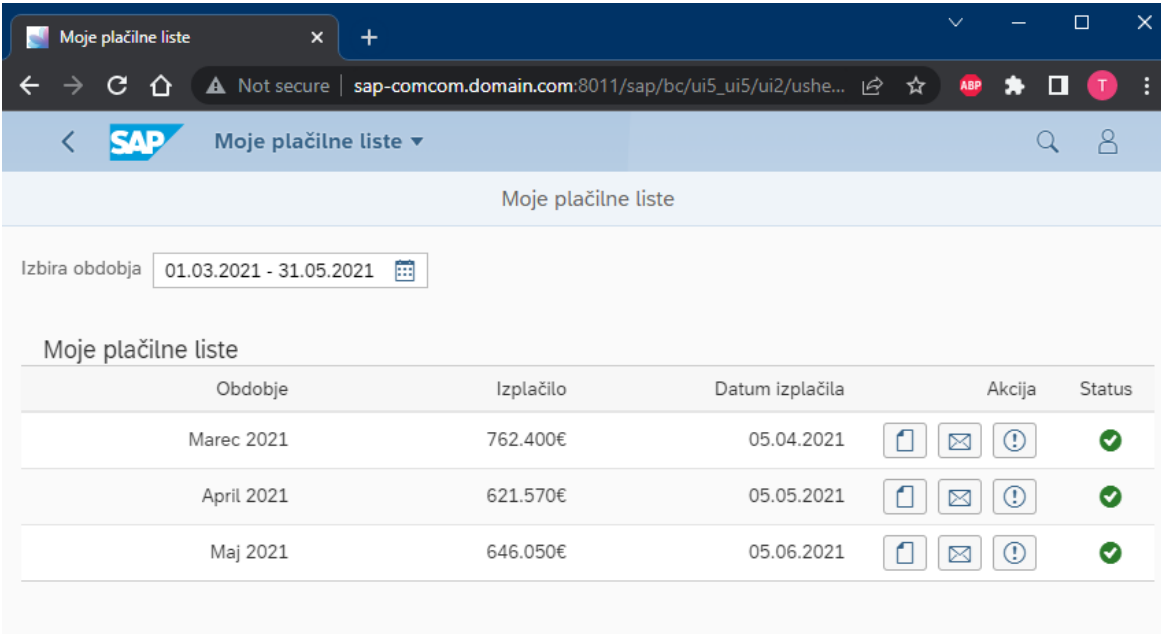
5.2 Pregled plačilnih list

Akter: Uporabnik aplikacije

Sprožilec: Izvajanje tega dela aplikacije se sproži ob samem zagonu aplikacije.

Uporabniku se na zaslону prikaže seznam njegovih plačilnih list za obdobje zadnjih šestih mesecev. Če je posamezna plačilna lista že arhivirana, to pomeni, da so jo v kadrovske službi že odobrili in je zato označena z ikono kljukice, uporabnik pa ima nato omogočene opcije predogleda dokumenta v brskalniku, prenosa dokumenta na lastno napravo, pošiljanja dokumenta na svoj e-poštni naslov in vlaganja pritožbe. Če plačilna lista v prikazanem seznamu še ni arhivirana, to pomeni, da je v SAP sistemu že obračunana, ni pa še odobrena s strani kadrovske službe, zato jo uporabnik še ne more videti in je označena z ikono klicaja, ki predstavlja napako. Če uporabnik v obdobju zadnjih šestih mesecev nima obračunanih plačilnih list, aplikacija uporabnika o tem obvesti.

Alternativni poteki: Uporabnik lahko kadarkoli spremeni časovno obdobje, za katerega bi rad dobil seznam plačilnih list. To lahko stori tako, da na koledarju, namenjenem vnosu časovnega obdobja, izbere novo časovno obdobje. Takoj, ko uporabnik to stori, se mu seznam na ekranu osveži.



The screenshot shows the SAP Fiori application interface for 'Moje plačilne liste'. At the top, there is a search bar and a user profile icon. Below that, the title 'Moje plačilne liste' is displayed. A date range selector shows 'Izbira obdobja' with the selected range '01.03.2021 - 31.05.2021'. The main content area contains a table with the following data:

Obdobje	Izplačilo	Datum izplačila	Akcija	Status
Marec 2021	762.400€	05.04.2021	[Document View] [Email] [Warning]	✓
April 2021	621.570€	05.05.2021	[Document View] [Email] [Warning]	✓
Maj 2021	646.050€	05.06.2021	[Document View] [Email] [Warning]	✓

Slika 3: Skica glavne strani aplikacije

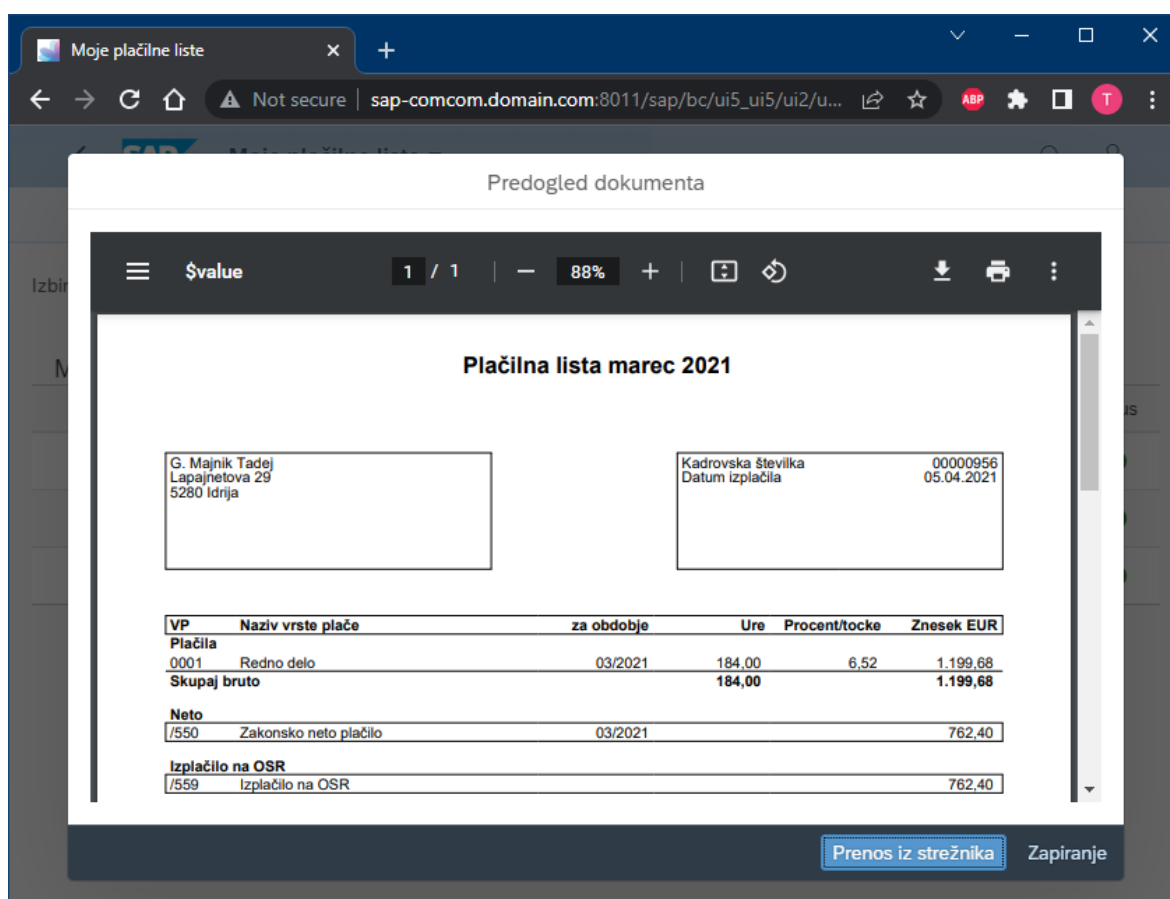
5.3 Predogled dokumenta v brskalniku

Akter: Uporabnik aplikacije

Sprožilec: Ta del aplikacije se sproži, ko uporabnik pritisne na gumb z ikono dokumenta.

Uporabnik ima za vsako odobreno plačilno listo, ki je prikazana na ekranu, možnost predogleda le-te v brskalniku. Ob kliku na gumb z ikono dokumenta se uporabniku odpre novo pogovorno okno, v njem pa je prikazana vsebina izbranega dokumenta. Uporabnik lahko prek tega okna taisti dokument tudi prenese na svojo napravo in natisne na svojem tiskalniku.

Alternativni poteki: Uporabnik lahko pogovorno okno predogleda zapre s klikom na gumb "Zapiranje". V primeru, da plačilna lista še ni arhivirana, so funkcionalnosti zanjo onemogočene, zato si je uporabnik ne more ogledati.



Slika 4: Skica predogleda plačilne liste ter izgled same plačilne liste

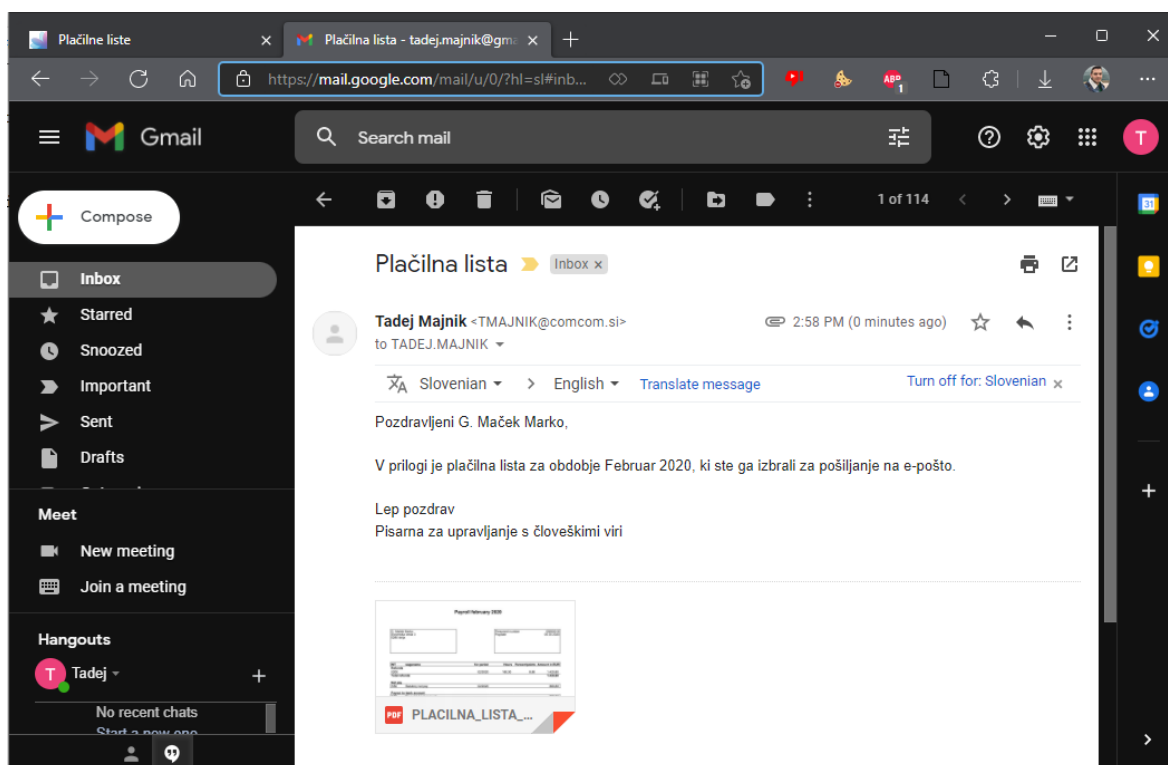
5.4 Pošiljanje na e-poštni naslov

Akter: Uporabnik aplikacije

Sprožilec: Ta del aplikacije se proži, ko uporabnik pritisne na gumb z ikono pisma.

Uporabnik ima za vsako odobreno plačilno listo možnost pošiljanja le-te na svoj e-poštni naslov. Ob kliku na gumb z ikono pisma se na uporabnikov e-poštni naslov iz zalednega sistema pošlje e-poštno sporočilo, ki vsebuje izbrano plačilno listo. Besedilo v e-poštnem sporočilu mora biti dinamično prilagojeno vsakemu uporabniku, vsebovati mora njegov naziv ter podatek o časovnem obdobju, za katerega je izbral opcijo pošiljanja. Pogoji za pošiljanje prek e-pošte je, da ima uporabnik v SAP sistemu pravilno vpisan e-poštni naslov in da je izbrana plačilna lista arhivirana. Če zaposlenemu manjka zapis za e-poštni naslov, ali pa je napačno zapisan, se dokumenta ne pošilja naprej.

Uporabnika nato aplikacija obvesti o uspešnosti pošiljanja na e-poštni naslov. Obvestilo se ob uspešnem pošiljanju obarva v zeleno, ob neuspešnem pa v rdečo.



Slika 5: Primer prejetega e-poštnega sporočila

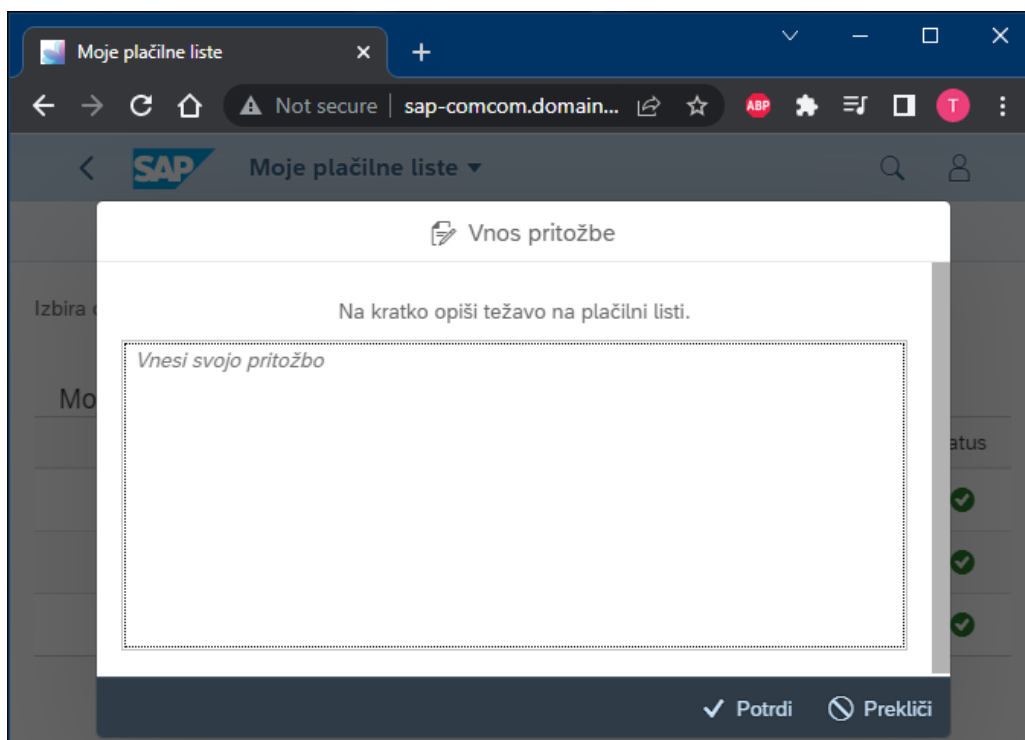
5.5 Oddaja pritožbe na plačilno listo

Akter: Uporabnik aplikacije

Sprožilec: Ta del aplikacije se sproži, ko uporabnik pritisne na gumb z ikono klicaja.

V temu primeru uporabe sta akterja uporabnik aplikacije in kadrovik. Če uporabnik ugotovi, da je na določeni plačilni listi kakršnakoli napaka, lahko odda pritožbo. S klikom na gumb z ikono klicaja se uporabniku odpre novo pogovorno okno za vnos pritožbe. Uporabnik lahko nato vneseno pritožbo potrdi in odda s klikom na gumb "Potrdi". Po oddani pritožbi nato zaledni sistem pošlje e-poštno sporočilo na e-poštni naslov kadrovika ali drugega zaposlenega v kadrovske službi, aplikacija pa uporabnika obvesti, da je bila pritožba uspešno vložena.

Alternativni poteki aplikacije: Uporabnik lahko vlaganje pritožbe prekliče s klikom na gumb "Prekliči". Podatke, ki jih je uporabnik vnesel, nato aplikacija pobriše, uporabnika pa vrne nazaj na prikaz seznama njegovih plačilnih list. Če je uporabnik na določeno plačilno listo že oddal pritožbo, aplikacija o tem obvesti uporabnika, vlaganje prošnje pa se prekine. Če plačilna lista še ni arhivirana, so gumbi in pripadajoče funkcionalnosti zanjo onemogočeni, zato nanjo še ne more vložiti pritožbe. V kolikor uporabnik odda prazno pritožbo, se vlaganje pritožbe prekliče, uporabnika pa o tem obvesti.



Slika 6: Skica pogovornega okna za oddajo pritožbe

5.6 Nefunkcijske zahteve

5.6.1 Varnost uporabnikovih podatkov

Zelo pomembna nefunkcijska zahteva aplikacije je varnost uporabnikovih osebnih in drugih podatkov. Aplikacija mora biti varna za končnega uporabnika. Za to je poskrbljeno že s strani SAP Fiori portala. Ob vstopu v portal se mora uporabnik poistovetiti s tem, da se prijavi s svojim uporabniškim imenom in geslom, dodeljenim v kadrovske službi.

5.6.2 Razpoložljivost

Aplikacija mora uporabniku zagotavljati nemoteno delovanje 24 ur na dan. Nemoteno delovanje aplikacije je neposredno odvisno od dosegljivosti obličnega in zalednega sistema, ki ju aplikacija uporablja, zato je zelo pomembno, da sta sistema vedno na voljo.

5.6.3 Tehnične zahteve kode in implementacije

Aplikacija mora zagotavljati relativno enostavne kasnejše popravke, razširitve in dodajanja novih funkcionalnosti. Koda mora biti zato napisana čimbolj berljivo, razumljivo in enostavno. Koristno je tudi pisanje komentarjev v kodi. Komentarji v kodi so vrstice, ki jih sistem ignorira in so namenjene boljšemu razumevanju zapisane kode. Čitljivost kode je še posebej pomembna v primerih, ko na projektu dela več razvijalcev. Zaželena je tudi dosledna uporaba funkcij in konceptov skozi celotno kodo.

5.6.4 Uporabnost in preglednost vmesnika

Aplikacija mora uporabniku ponuditi kar se da pregleden, čist in enostaven uporabniški vmesnik, ki ne sme biti prenatrpan z ikonami, vse funkcionalnosti pa morajo biti jasno označene. Aplikacija mora uporabniku posredovati razumljivo povratno informacijo po vsaki interakciji z njo.

5.6.5 Učinkovitost delovanja

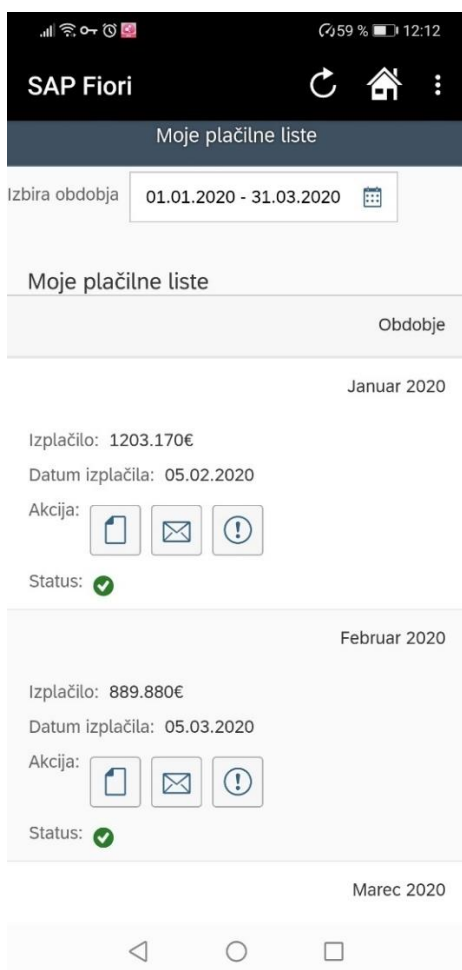
Aplikacija mora uporabniku zagotavljati hiter in odziven uporabniški vmesnik, ki omogoča učinkovito uporabo. Uporabniku moramo zagotoviti čim krajši čas nalaganja aplikacije in čas procesiranja njegovih zahtev. Odziven uporabniški vmesnik in kratki časi nalaganja podatkov so zelo pomembni za dobro uporabniško izkušnjo.

5.6.6 Opcija izbire jezika

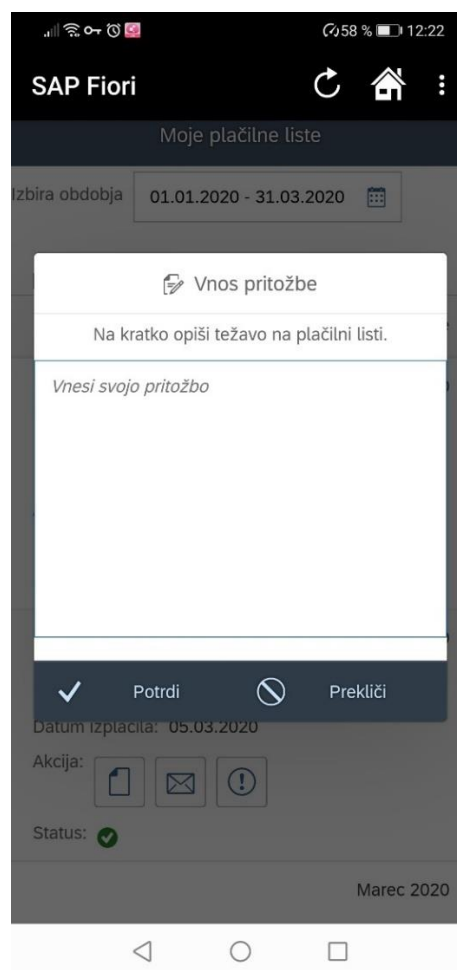
Uporabnik bo imel možnost izbire jezika uporabniškega vmesnika aplikacije in jezika prikazanih plačilnih list. Jezik izbere že ob prijavi v SAP Fiori platformo, uporabniški vmesnik aplikacije in dokument pa se prilagodita izbranemu prijavnemu jeziku. Uporabnik bo lahko izbiral med Slovenščino in Angleščino.

5.6.7 Podpora mobilnih naprav

Aplikacija bo omogočala brezhibno delovanje in imela odziven vmesnik tudi na mobilnih napravah in napravah z upravljanjem na dotik. Razmerje in velikost različnih komponent aplikacije se bosta prilagodila glede na zaslon uporabnikove naprave.



Slika 7: Skica aplikacije na mobilni napravi



Slika 8: Skica okna za oddajo pritožbe na mobilni napravi

6 NAČRTOVANJE SISTEMA

V tej točki bom opisal ustrezne načrtovalske cilje, ki so pomembni za mojo Fiori aplikacijo, aplikacijo razdelil na podsisteme, izbral nekaj že izdelanih komponent, programov in orodij, opisal upravljanje s perzistentnimi podatki in določil politiko dostopa do aplikacije.

6.1 Definiranje načrtovalskih ciljev

6.1.1 Učinkovitost delovanja

Pri načrtovanju aplikacije je pomembno, da jo zasnujemo tako, da je vmesnik hiter in odziven, čas branja podatkov in procesiranja uporabnikovih zahtev pa čim krajši. Slednji je odvisen od hitrosti samega SAP obličnega in zalednega sistema, zato moramo naš SAP sistem pravilno nastaviti za optimalno delovanje in procesiranje zahtev. Aplikacija ne bo zahtevna glede sistemskih resursov, ker bo zelo preprosta. Poraba pomnilnika na uporabnikovi napravi bo zato odvisna izključno od hitrosti brskalnika, ki ga uporablja uporabnik. Uporabniki, ki lahko na svoji napravi poganjajo brskalnik in odpirajo PDF dokumente[1], bodo lahko brez problema poganjali tudi aplikacijo za prikazovanje plačilnih list.

6.1.2 Zanesljivost

Aplikacija mora omogočati varno komunikacijo med obličnim in zalednim, poskrbeti pa mora tudi za varnost uporabnikovih osebnih ter drugih podatkov. Aplikacija prevzame lastnosti zanesljivosti platforme Fiori, saj mora biti uporabnik v platformo prijavljan z uporabniškim imenom in geslom.

Sama aplikacija pa mora biti zasnovana tako, da uporabnik ne bo imel možnosti vnosa napačnih podatkov ali neveljavnega uporabniškega zahtevka, ki bi lahko onemogočile ali ogrozile nemoteno delovanje Fiori aplikacije za prikaz plačilnih list.

Aplikacija mora uporabniku zagotavljati nemoteno delovanje 24 ur na dan. Nemoteno delovanje aplikacije je neposredno odvisno od dosegljivosti obličnega in zalednega sistema, ki ju aplikacija uporablja, zato je zelo pomembno, da sta sistema vedno dosegljiva in na voljo za uporabo.

6.1.3 Vzdrževanje

Aplikacijo mora biti zasnovana tako, da je mogoče relativno enostavno dodajanje novih funkcionalnosti. Omogočati mora tudi enostavno spreminjanje uporabniškega vmesnika ali njenih funkcionalnosti. Izvorna koda aplikacije mora biti zato čimbolj čitljiva, da je dodajanje novih funkcionalnosti lahko hitro in učinkovito. Čitljivost izvorne kode je pomembna tudi, če bi v prihodnosti na projektu delali še drugi razvijalci. Pomembno je tudi pisanje komentarjev v kodi in pisanje dokumentacije. Za pisanje dokumentacije bom uporabil spletno platformo Confluence, dostopno na povezavi <https://www.atlassian.com/>.

6.1.4 Uporabniške zahteve

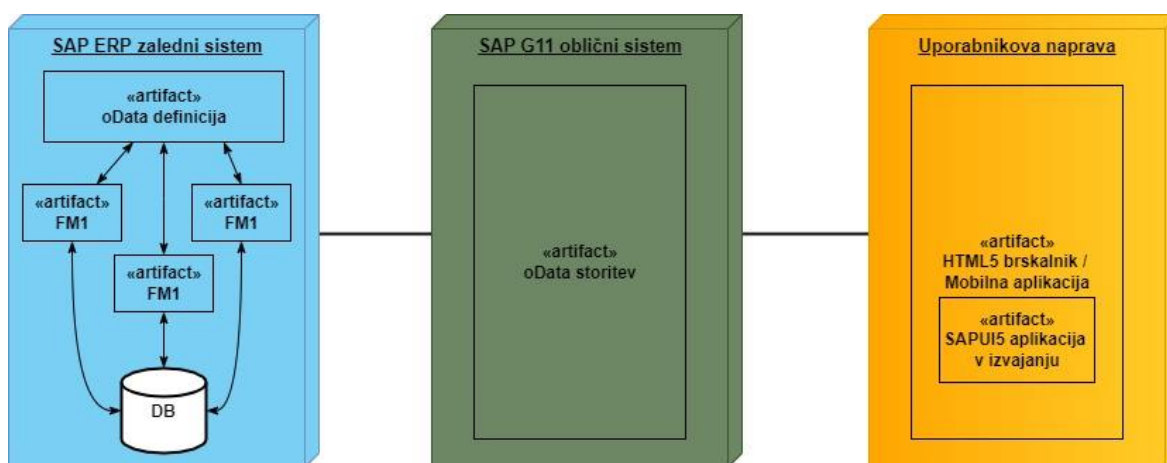
Aplikacija mora biti koristna za uporabnika. Zelo dobro mora podpirati delo uporabnika. Omogočala bo njeno uporabo na številnih napravah od osebnega računalnika pa do tablic in pametnih telefonov. Platforma Fiori ponuja tudi zelo dobro podporo za naprave z upravljanjem na dotik.

Da bo uporaba aplikacije za končnega uporabnika enostavna in preprosta, moramo implementirati poenostavljen uporabniški vmesnik, ki je uporabniku prijazen. Aplikacija mora imeti vse funkcionalnosti zbrane na nekaj ekranih, gumbi, vnosna polja in druga polja pa morajo biti jasno označeni. Jasna morajo biti tudi navodila za uporabnika, ki jo uporablja. Uporabniku se mora prikazati tudi nasvet za uporabo določenega gumba ali vnosnega polja, če se z računalniško miško premakne čez omenjeni gumb ali polje.

6.2 Dekompozicija sistema na podsisteme/komponente

Celotno aplikacijo bom razdelil na dva večja podsistema ali komponente. Prvi podsistem je poslovni del logike, ki ga definiramo kot oData spletno storitev [14]. Ta je nameščena na zalednem sistemu SAP ERP E11. Ta podsistem prek ABAP razredov obdeluje, hrani, piše in bere podatke uporabnikov ter njihovih plačilnih list, ki so shranjeni v podatkovnih bazah. Ko oData spletno storitev definiramo na zalednem SAP sistemu, izvedemo tudi registracijo oData storitve na obličnem sistemu. Omenjena storitev deluje kot nekakšen posrednik in se uporablja za komunikacijo med zalednim in obličnim sistemom.

Drug podsistem ali komponenta je SAPUI5 aplikacija. Aplikacija predstavlja spletni del logike, uporabniški vmesnik. Razvijemo jo lahko v SAP Web IDE razvojnem okolju na SAP spletni storitvi v oblaku SAP HANA Cloud. Aplikacijo bom nato namestil na SAP oblični sistem, konkretno na SAP Netweaver Gateway - sistem G11. Razvita aplikacija se bo izvajala v brskalniku na uporabnikovem računalniku ali pa v aplikaciji za mobilne naprave. Uporabnikov vnos podatkov bo nato sprožil klic ustreznih razredov in funkcij v zalednem sistemu.



Slika 9: Diagram postavitve Fiori aplikacije

6.3 Izbira že izdelanih komponent

Za implementacijo aplikacije bom izbral nekaj že izdelanih komponent oziroma sistemov. Uporabil bom SAP zaledni sistem (E11), na katerem bom implementiral logiko svoje Fiori aplikacije. Zaledni sistem bom nato povezal z obličnim sistemom G11 preko posrednika oData, na katerem bom nato razvil SAP Fiori aplikacijo z uporabo SAP Web IDE razvojnega okolja. Za arhiviranje dokumentov bom uporabil SAP funkcijski modul ARCHIVOBJECT_CREATE_TABLE, ki omogoča enostavno arhiviranje kakršnihkoli dokumentov v SAP-jev arhivski sistem. Za pošiljanje e-poštnih sporočil, tako za plačilne liste, kot za pritožbe, bom uporabil razred CL_BCS, ki je v SAP sistemu že implementiran in je namenjen prav temu. Za izdelavo osnovne strukture in izgleda SAPUI5 aplikacije bom uporabil spletno orodje Build.me, ki omogoča enostavno grafično oblikovanje in urejanje modernih spletnih aplikacij in je dostopna na povezavi <https://www.build.me/>.

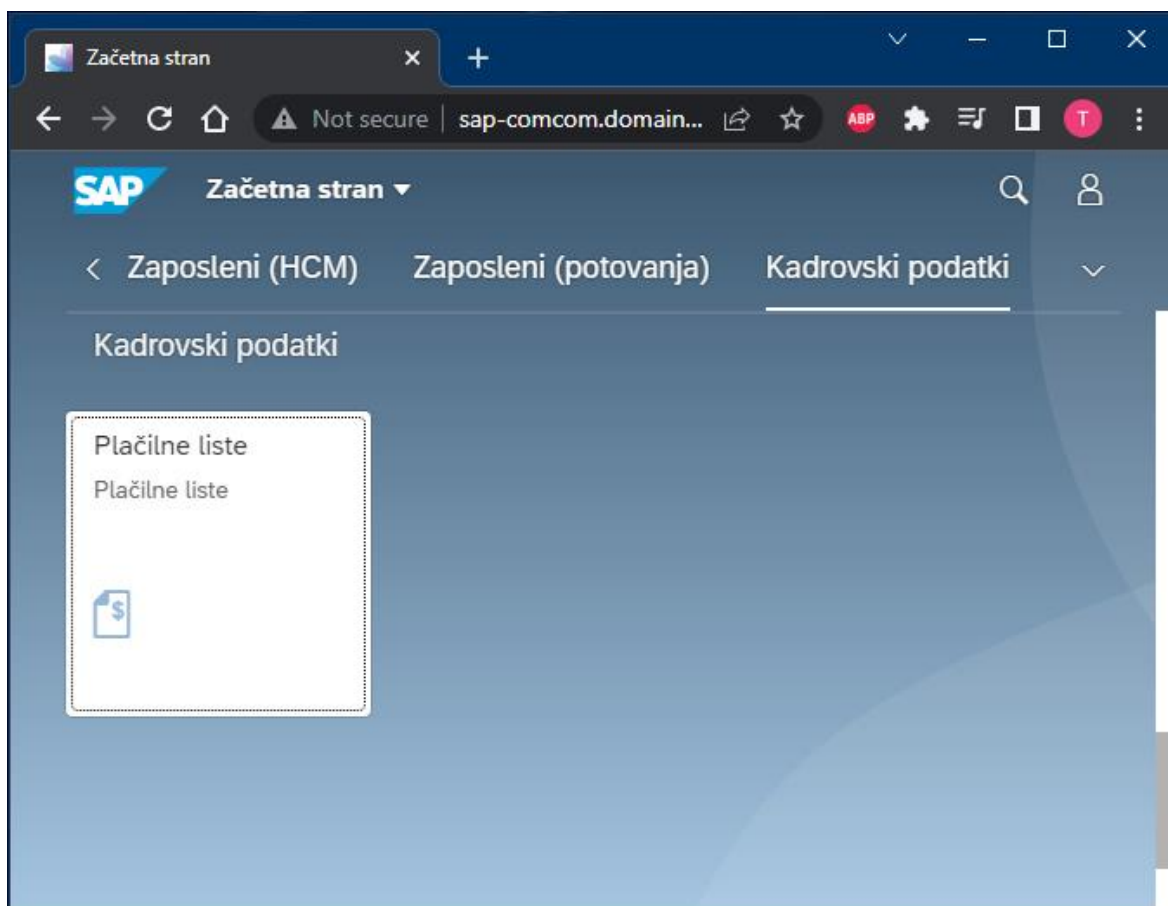
6.4 Upravljanje s perzistentnimi podatki

Pri implementaciji SAP Fiori aplikacije je zelo pomembno tudi upravljanje s perzistentnimi podatki. To funkcionalnost razvijalcem programske opreme ponuja že sam SAP zaledni sistem. Za branje, pisanje, popravljanje, brisanje in upravljanje s podatki se v SAP zalednemu sistemu uporablja open SQL[12], skupek določenih ukazov za poizvedovanje podatkov, ki je popolno integriran v ABAP programski jezik. Sintaksa jezika za poizvedovanje open SQL je zelo podobna standardnemu SQL jeziku. Razvijalcem ponuja ukaze "SELECT", "UPDATE", "DELETE", "JOIN" ter mnoge druge.

6.5 Določitev politike nadzora dostopa

Uporabniki morajo za dostop do Comcom SAP Fiori portala uporabiti storitev VPN, navidezno privatno omrežje[2] ali pa se direktno povežati v Comcom internetno omrežje. Nato se morajo prijaviti v SAP Fiori portal s svojim uporabniškim imenom in geslom, ki jim ga ustvari in pošlje administrator Comcom SAP sistema ali kadrovska služba. Poleg tega mora administrator uporabniku dodeliti ustrezno avtorizacijo, brez katere uporabniki v SAP sistemu ne morejo dostopati do določenih modulov ali aplikacij. Dostop do poslanih pritožb imajo kadrovske delavci, prejmejo pa jih neposredno na službeni e-poštni naslov.

Ko ima uporabnik dodeljene vse ustrezne dostope in avtorizacije, se mu na njegovem začetnem ekranu prikaže ploščica z imenom "Plačilne liste" v razdelku "Kadrovski podatki". S klikom na to ploščico lahko uporabnik zažene aplikacijo.



Slika 10: Prikaz Fiori portala in ploščice za zagon aplikacije

7 IZVEDBA

Izvedbo bom razdelil na 3 glavne dele: Prvi del je implementacija programa za izpis in arhiviranje plačilnih list na zalednem Comcom sistemu E11. Tega dela Fiori aplikacija ne bo direktno uporabljala, vendar ga moramo vseeno razviti za kreiranje in arhiviranje plačilnih list. Drugi del je implementacija logike za pripravo podatkov, podrobneje oData spletne storitve, ki bo služil kot posrednik med zalednim in obličnim sistemom. Odata storitev moramo nato še registrirati na G11 obličnem strežniku. Tretji del je razvoj in oblikovanje SAPUI5 Fiori aplikacije. V naslednjih točkah bom podrobneje opisal razvoj in potek vsakega dela posebej.

7.1 Izbira programskih jezikov in razvojnih orodij

Za programiranje logike na zalednem sistemu SAP bom uporabil programski jezik ABAP. ABAP je visoko-nivoski programski jezik, namenjen razvoju naprednih poslovnih aplikacij. Ta je bil razvit izključno za uporabo v SAP sistemih. Za programiranje logike SAPUI5 aplikacije bom uporabil programski jezik JavaScript. Za strukturo aplikacije pa bom uporabil jezik XML.

Za razvoj in oblikovanje osnovne strukture aplikacije bom uporabil spletno razvojno orodje Build.me, ki omogoča enostavno grafično oblikovanje SAPUI5 aplikacije ter njenih komponent.

7.2 Program za pripravo plačilnih list

7.2.1 Nabiranje podatkov plačilne liste v transakciji HRFORMS

V transakciji HRFORMS definiramo in kreiramo program za izpis plačilne liste in definiramo obrazec ter njegovo vhodno strukturo z imenom HRDATA, ki ga obrazec prejme kot vhodni parameter. Ta vhodna struktura vsebuje podatke posamezne plačilne liste kot npr. osebne podatke zaposlenega, podatke podjetja, kjer je zaposlen, ter podatke plač, dohodnine, prispevkov ter drugih podobnih sklopov.

7.2.2 Izdelava logike in oblikovanja obrazca za plačilne liste

Za oblikovanje obrazcev ter implementacijo njihove logike je v SAP sistemu na voljo več transakcij. Med njimi sta SMARTFORMS in ADOBEFORMS. Za izdelovanje plačilnih list in podobnih dokumentov, povezanih s človeškimi viri, se v našem podjetju najpogosteje uporablja transakcija SMARTFORMS, ker je zanje bolj primerna, poleg tega pa je v njej tudi lažji razvoj obrazcev, sploh obrazcev z več stranmi. V tem zaključni nalogi bom zato izbral transakcijo SMARTFORMS.

7.2.3 Prilagoditev razreda CL_HRFORMS_HRF02

Razred CL_HRFORMS_HRF02 se samodejno generira takoj, ko definiramo obrazec in program za izpis le-tega. Ker je generiran s strani SAP sistema in ga nismo kreirali sami, ga ne moremo neposredno urejati ali spreminjati. K sreči nam SAP ponuja možnost kreiranja razširitve(v izvorniku "Enhancement") takih razredov in funkcijskih modulov, ki omogoča, da razvijalec sam doda svojo kodo, ki se bo izvedla poleg standardne kode SAP sistema. Preko te razširitve dodamo našo kodo, ki je potrebna za arhiviranje, kontrolo arhiviranja in izpis naših dokumentov.

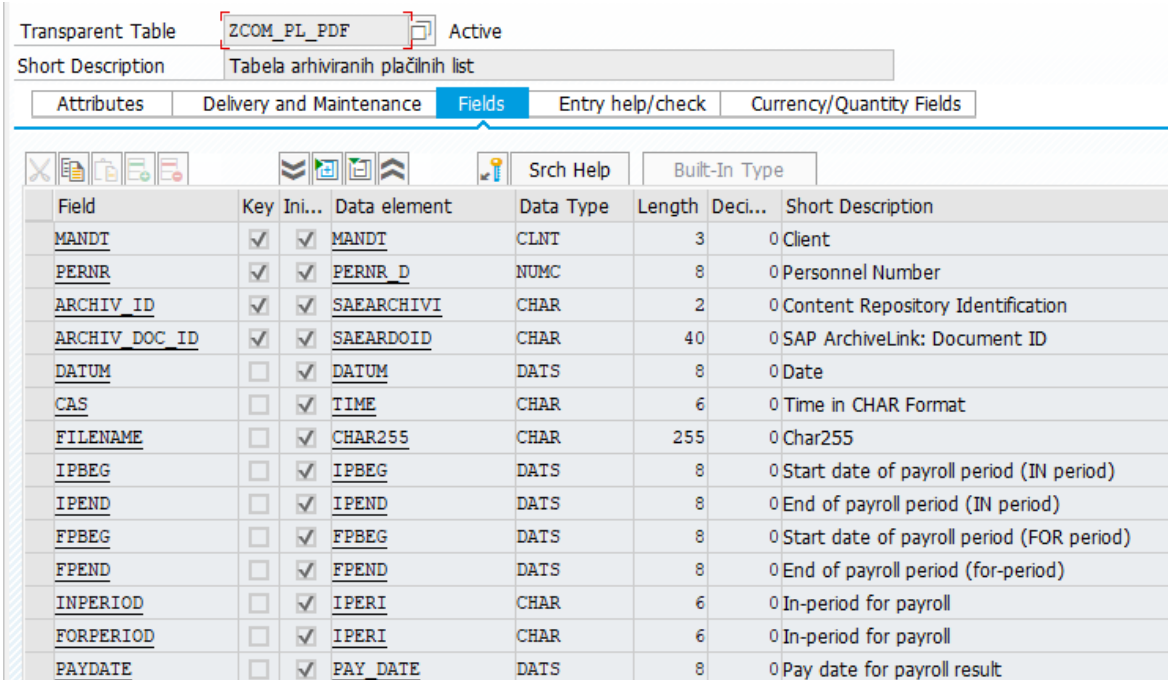
Poskrbeti moramo tudi za kontrolo arhiviranja, bolj natančno, da se isti dokument ne more arhivirati večkrat. Za kontrolo arhiviranja po meri kreiramo novo uporabniško tabelo po imenu ZCOM_PL_PDF. Pred arhiviranjem prečitamo omenjeno tabelo, če za trenutno plačilno listo že obstaja pripadajoč zapis. Če zapis ne obstaja, se izvrši arhiviranje, po uspešnem arhiviranju pa se v tabeli kreira nov zapis, pripadajoč taistemu dokumentu. V nasprotnemu primeru se arhiviranje ne izvede.

7.2.4 Priprava arhiviranja na Comcom zalednem SAP sistemu

Za dejansko arhiviranje in branje iz arhiva kreiramo nov razred z imenom ZCL_PLACILNA_ALINK_HANDLER. Ta prek metode create_pdf pripravi posamezno plačilno listo in jo pošlje v arhiv SAP sistema in prek metode read_pdf bere dokumente iz arhiva. Slednjo bom uporabil kasneje ob razvoju oData storitve.

7.2.5 Kreiranje tabele metapodatkov

V transakciji SE11 definiramo in kreiramo že prej omenjeno podatkovno tabelo, ki nam bo služila za shranjevanje metapodatkov arhiviranih plačilnih list. Pomembno polje v tej tabeli je ARCHIV_DOC_ID, identifikator ali ključ posameznega dokumenta v arhivu.



Field	Key	Ini...	Data element	Data Type	Length	Deci...	Short Description
<u>MANDT</u>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3	0	Client
<u>PERNR</u>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	PERNR_D	NUMC	8	0	Personnel Number
<u>ARCHIV_ID</u>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	SAEARCHIVI	CHAR	2	0	Content Repository Identification
<u>ARCHIV_DOC_ID</u>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	SAEARDOID	CHAR	40	0	SAP ArchiveLink: Document ID
<u>DATUM</u>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	DATUM	DATS	8	0	Date
<u>CAS</u>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	TIME	CHAR	6	0	Time in CHAR Format
<u>FILENAME</u>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	CHAR255	CHAR	255	0	Char255
<u>IPBEG</u>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	IPBEG	DATS	8	0	Start date of payroll period (IN period)
<u>IPEND</u>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	IPEND	DATS	8	0	End of payroll period (IN period)
<u>FPBEG</u>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	FPBEG	DATS	8	0	Start date of payroll period (FOR period)
<u>FPEND</u>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	FPEND	DATS	8	0	End of payroll period (for-period)
<u>INPERIOD</u>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	IPERI	CHAR	6	0	In-period for payroll
<u>FORPERIOD</u>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	IPERI	CHAR	6	0	In-period for payroll
<u>PAYDATE</u>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	PAY_DATE	DATS	8	0	Pay date for payroll result

Slika 11: Struktura tabele metapodatkov

7.2.6 Prevodi plačilnih list

Obrazce za izpis plačilnih list lahko prevedemo iz izvornega jezika v sekundarni jezik s pomočjo transakcije SE63. Tam lahko za vsak tekstovni element na obrazcu določimo njegov prevod v želenem jeziku. Za potrebe te zaključne naloge bom obrazec prevedel v Angleščino. Za prevajanje različnih tekstov si bom pomagal z uporabo spletnih slovarjev Google prevajalnik, ki je dostopen na povezavi <https://translate.google.com/?hl=sl> in Pons dostopen na povezavi <https://sl.pons.com/prevod>.

7.2.7 Arhiviranje plačilnih list

Plačilne liste moramo arhivirati s pomočjo izpisnega programa, katerega kreira sam SAP sistem, ko kreiramo obrazec v transakciji HFORMS. Tega dela uporabnik aplikacije ne bo videl, ker se bo ta del pognal v kadrovske službi na zalednem sistemu. Na začetnem selekcijskem ekranu moramo izbrati želeno obračunsko obdobje, področje obračuna plač, kadrovske številke zaposlenih, za katere poganjamo program, jezik arhiviranega dokumenta in način izpisa. Ko arhiviramo dokumente, moramo v sklopu "Nastavitve specifične za obrazec" v spustnem seznamu izbrati opcijo "X Arhiviranje". Po uspešnem arhiviranju program vstavi nove zapise v tabelo metapodatkov.

Dokazilo o plačilu (neodvisno od držav)

Organizac. struktura Pomoč pri iskanju

Obračunsko obdobje

Obdobje	2 Drugo obračunsko obdobje
Podr.obrač.plač	SA 01.02.2020 - 29.02.2020
Obdobje	2 2020

Pogoji izbire

Kadrovska številka	16	
Področje obr. plač	SA	

Nastavitve specifične za obrazec

<input checked="" type="checkbox"/> Izločitev statistik	
Arhiviranje obrazcev	K Arhiviranje
<input type="checkbox"/> Obrazec v jeziku zaposlenega	

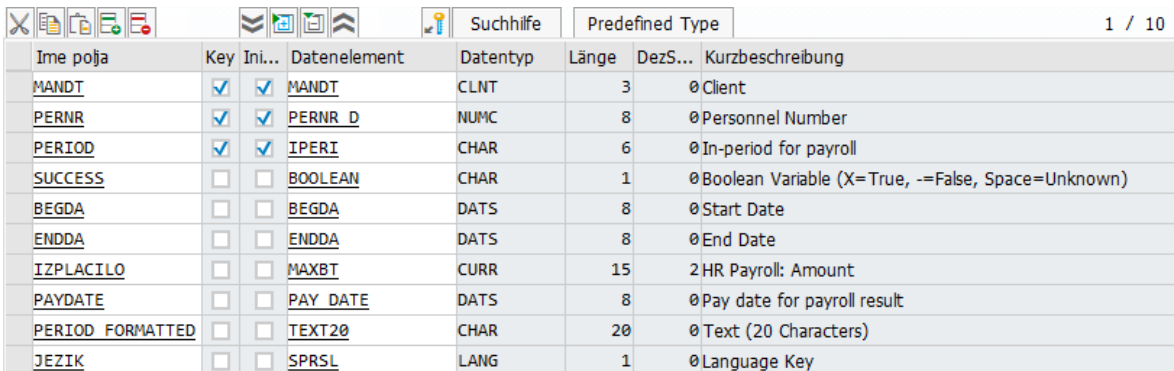
Dodatne omejitve izbire

Jezik obrazca	SL Slovenščina
---------------	----------------

Slika 12: Primer selekcijskega ekrana za pravilno arhiviranje

7.3 Izdelava oData storitve

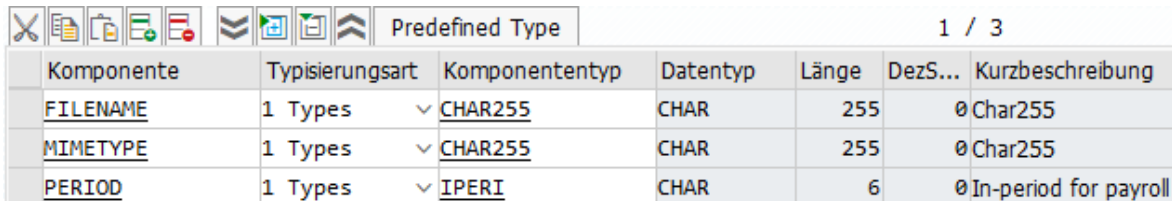
Za izdelavo oData storitve moramo najprej definirati entitetne tipe struktur za vsako posamezno oData storitev. To lahko storimo v ABAP slovarju v transakciji SE11. V strukturah moramo definirati vsa potrebna polja, njihove podatkovne tipe ter potrebne ključe. Za prikaz plačilnih list zaposlenega potrebujemo polja na spodnji sliki. Polje SUCCESS nam v tem primeru pove, ali je bila plačilna lista že arhivirana.



Ime polja	Key	Ini...	Datenelement	Datentyp	Länge	DezS...	Kurzbeschreibung
MANDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3	0	Client
PERNR	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	PERNR_D	NUMC	8	0	Personnel Number
PERIOD	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	IPERI	CHAR	6	0	In-period for payroll
SUCCESS	<input type="checkbox"/>	<input type="checkbox"/>	BOOLEAN	CHAR	1	0	Boolean Variable (X=True, -=False, Space=Unknown)
BEGDA	<input type="checkbox"/>	<input type="checkbox"/>	BEGDA	DATS	8	0	Start Date
ENDDA	<input type="checkbox"/>	<input type="checkbox"/>	ENDDA	DATS	8	0	End Date
IZPLACILO	<input type="checkbox"/>	<input type="checkbox"/>	MAXBT	CURR	15	2	HR Payroll: Amount
PAYDATE	<input type="checkbox"/>	<input type="checkbox"/>	PAY_DATE	DATS	8	0	Pay date for payroll result
PERIOD FORMATTED	<input type="checkbox"/>	<input type="checkbox"/>	TEXT20	CHAR	20	0	Text (20 Characters)
JEZIK	<input type="checkbox"/>	<input type="checkbox"/>	SPRSL	LANG	1	0	Language Key

Slika 13: Struktura entitetnega tipa za seznam plačilnih list

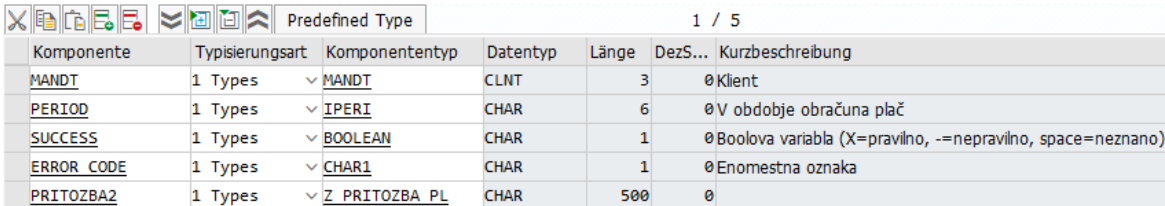
Za prenos dejanskega dokumenta moramo definirati strukturo, kot je na spodnji sliki. Pomembno polje je MIMETYPE, ki ga moramo napolniti kot "application/pdf", če ne, dokumenta drugi sistemi ne bodo prepoznali kot PDF dokument in ga ne bodo znali prikazati.



Komponente	Typisierungsort	Komponententyp	Datentyp	Länge	DezS...	Kurzbeschreibung
FILENAME	1 Types	CHAR255	CHAR	255	0	Char255
MIMETYPE	1 Types	CHAR255	CHAR	255	0	Char255
PERIOD	1 Types	IPERI	CHAR	6	0	In-period for payroll

Slika 14: Struktura entitetnega tipa za prenos datoteke

Za pošiljanje e-poštnih sporočil in pritožb definiramo in uporabimo kar skupno strukturo, saj bosta oData funkciji zelo podobni. Pri obeh se bo dokumente pošiljalo preko e-pošte, razlika je v tem, da pri pošiljanju na e-mail izklopimo polje pritozna2, saj ga ne potrebujemo. Če ima polje SUCCESS vrednost "false", lahko iz polja ERROR_CODE ugotovimo, do kakšne napake je prišlo v zalednem sistemu. Glede na polje ERROR_CODE nato v aplikaciji prikažemo ustrezno obvestilo uporabniku.



Komponente	Typisierungstyp	Komponententyp	Datentyp	Länge	DezS...	Kurzbeschreibung
MANDT	1 Types	MANDT	CLNT	3		0 Klient
PERIOD	1 Types	IPERI	CHAR	6		0 V obdobje obračuna plač
SUCCESS	1 Types	BOOLEAN	CHAR	1		0 Boolova variabla (X=pravilno, -=nepravilno, space=neznano)
ERROR_CODE	1 Types	CHAR1	CHAR	1		0 Enomestna oznaka
PRITZBA2	1 Types	Z PRITZBA_PL	CHAR	500		0

Slika 15: Struktura entitetnega tipa za pošiljanje e-poštnega sporočila

7.3.1 Definiranje oData storitev

Odata storitve lahko v SAP sistemu definiramo preko transakcije SEGW, transakcije za oblikovanje in izdelovanje storitev. V njej definiramo podatkovni model oData storitve, entitetne tipe preko struktur, ki smo jih kreirali v prejšnji točki, ter entitetni nabor (v izvorniku entity set), funkcije, ki jih bomo uporabljali za klice iz obličnega v zaledni sistem. En entitetni nabor posamezne oData storitve lahko vsebuje metode Create, Delete, GetEntity, GetEntitySet in update.

7.3.2 Branje seznama plačilnih list

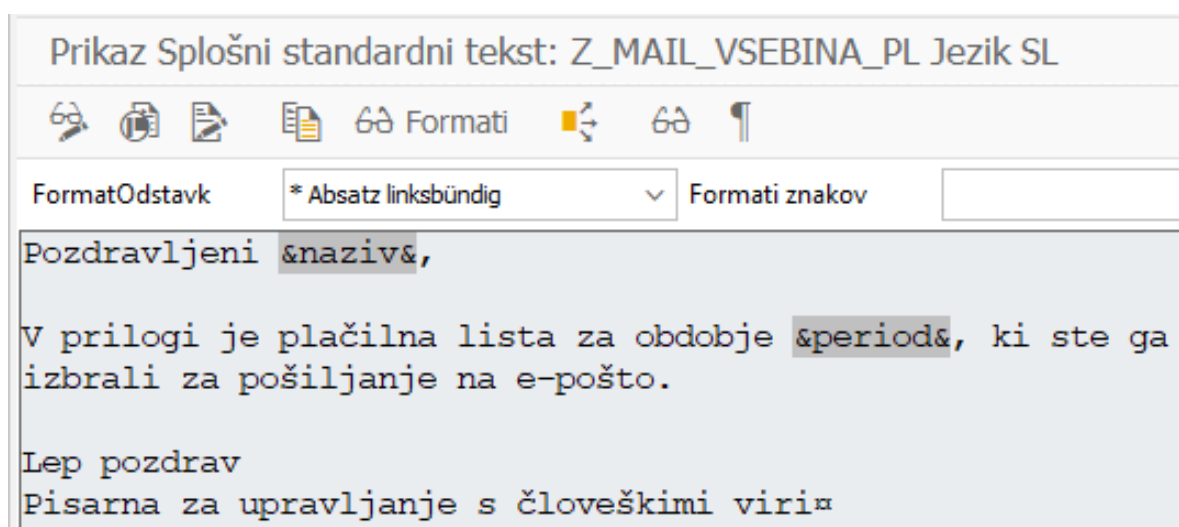
V primeru branja seznama plačilnih list bomo uporabili GetEntitySet, ki po klicu vrne seznam entitet (tabela) v oblični sistem. Za branje obračunanih plačilnih list lahko v SAP-ju uporabimo funkcijski modul CD_READ_RGDIR. Za vse plačilne iz vrnjenega seznama nato preverimo, če so tudi arhivirane. Tiste plačilne liste, ki so tudi arhivirane, bodo kasneje v aplikaciji prikazane z zeleno kljukico, nearhivirane pa bodo prikazane s klicajem.

7.3.3 Prenos dejanskega PDF dokumenta v oblični sistem

Za prenos PDF dokumenta moramo pripadajoči entitetni tip specificirati kot "Media" in v podatkovnem modelu implementirati metodo /IWBEP/IF_MGW_APPL_SRV_RUNTIME~GET_STREAM, ki nam omogoča prenos datoteke na strežnik. V tej metodi moramo SAP interno binarno tabelo, v kateri je shranjen dokument, pretvoriti v obliko XSTRING, ki jo zahteva SAPUI5 aplikacija. Pomembno je tudi, da polje MIMETYPE napolnimo z vrednostjo "application/pdf", da lahko zunanji sistemi pravilno razberejo, da gre za PDF datoteko in jo pravilno odprejo ali uporabijo.

7.3.4 Pošiljanje plačilne liste zaposlenemu prek e-pošte

Za pošiljanje e-poštnih sporočil zaposlenemu implementiramo metodo Create. V tej metodi uporabimo funkcijski modul HR_READ_INFOTYPE za branje infotipa "0105", kjer so shranjeni uporabnikovi zapisi za komunikacijo, in preberemo zapis s podtipom "MAIL", kjer je shranjen uporabnikov e-poštni naslov. Konkretno besedilo, ki jih uporabimo za vsebino e-poštnega sporočila, lahko hranimo in vzdržujemo za vse jezike v transakciji SO10 kot standardne besedila. Te besedila v kodi preberemo z uporabo funkcijskega modula READ_TEXT. V njem lahko tudi specificiramo spremenljivke, ki jih v kodi nadomestimo s konkretnimi vrednostmi. Prednost uporabe standardnih besedil je vzdrževanje besedil, neodvisno od programa. Lažja je tudi implementacija prevodov vsebine besedil. V primeru, da je uporabnikov mail neveljaven ali manjka in v primeru napake pri izvajanju kode se e-poštnega sporočila ne pošlje naprej, uporabnika pa se o tem obvesti v aplikaciji. Za dejansko pošiljanje sporočila uporabimo razred CL_BCS, implementiran v SAP-ju.



Slika 16: Primer standardnega besedila v transakciji SO10

7.3.5 Pošiljanje pritožbe v kadrovske službe

Tako kot za pošiljanje e-poštnih sporočil zaposlenim moramo za pošiljanje pritožb v kadrovske službe implementirati funkcijo Create. Prav tako za vsebino e-poštnega sporočila uporabimo standardne besedila v transakciji SO10. Za dejansko pošiljanje sporočila uporabimo SAP-jev razred CL_BCS.

7.4 Registracija oData storitve in dodajanje avtorizacij

Dokončano oData storitev moram nato registrirati na obličnem strežniku G11. To lahko storimo v transakciji LPD_CUST, kjer vnesemo naziv aplikacije, izberemo tip aplikacije kot "URL" ter nastavimo vse druge potrebne lastnosti in attribute za pravilno delovanje. Kreirati moramo tudi ploščico, preko katere bo lahko kasneje uporabnik dostopal do aplikacije. To lahko storimo preko spletnega orodja Fiori launchpad designer, do katerega dostopamo preko spletnega brskalnika. Nato moramo uporabnikom še dodeliti vse potrebne dostope ter avtorizacije v transakciji PFCG, da bodo lahko uporabljali aplikacijo.

7.5 SAPUI5 Fiori aplikacija

7.5.1 Osnovna struktura aplikacije

Osnovno strukturo in obliko aplikacije sem zasnoval in oblikoval s pomočjo spletnega orodja Build.me, ki omogoča grafično oblikovanje SAPUI5 aplikacije in enostavno ter intuitivno urejanje posameznih komponent in gradnikov v aplikaciji. Grafično oblikovano aplikacijo si nato prenesem na računalnik in jo uvozim v SAP Web IDE razvojno okolje, kjer bom razvil logiko za prikazovanje, procesiranje, pošiljanje ter prejemanje podatkov iz zalednega sistema. Oblikovanje in gradniki aplikacije so shranjeni v XML datoteki, v kateri so navedene vse lastnosti aplikacije. Posamezni gradniki so v XML datoteki nahajajo med znakoma < in > in so označeni z značkami (v izvorniku Tags) npr. <table>, ki predstavlja značko za tabelo. Večina gradnikov ima v znački specificirane tudi dodatne lastnosti, kot npr. širina, višina, obrobe, za gumbe pa je tudi značilna lastnost PRESS, v kateri specificiramo funkcijo, ki se izvrši ob kliku ali pritisku na omenjen gumb. Nekatere gradnike, ki vsebujejo druge gradnike, je potrebno zapreti z enako značko, ki vsebuje takoj za začetnem znakom < tudi znak /, npr. </table>. Nekateri gradniki imajo tudi lastnosti VISIBLE in ENABLED. Prva specificira, če je gradnik viden, druga pa, če je delujoč. Lastnosti gradnikov lahko nastavljamo tudi v kodi na podlagi kakršnihkoli pogojev.

7.5.2 Glavna stran aplikacije

Na glavno stran aplikacije postavimo polje za vnos časovnega obdobja, kjer bo končni uporabnik lahko spreminjal željeno časovno obdobje za prikaz plačilnih list, in tabelo za prikaz podrobnih podatkov posameznih plačilnih list. Zraven polja za vnos časovnega obdobja bom dodal še gumb za brisanje trenutno izbranega časovnega obdobja, in gumb za osvežitev seznama prikazanih plačilnih list. Vsak zapis plačilne liste v prikazani tabeli bo v stolpcu "akcija" vseboval gumb za predogled dokumenta, gumb za pošiljanje dokumenta na uporabnikov e-poštni naslov ter gumb, s katerim odpremo novo pogovorno okno za vnos pritožbe na izbrano plačilno listo. Vsak zapis pa bo tudi vseboval ikono, ki uporabniku sporoča, ali je določen dokument že arhiviran. Gumbi za predogled, pošiljanje na e-pošto in oddajo pritožbe bodo za določeno plačilno listo onemogočeni, če le-ta še ne bo arhivirana.

7.5.3 Branje seznama plačilnih list

Za branje seznama uporabnikovih plačilnih list moramo napolniti vhodne podatke, ki jih je uporabnik izbral v polju za vnos časovnega obdobja in klicati metodo `.read`, ki se sklicuje na funkcijo `getEntitySet` v oData strukturi. Ta nam iz zalednega sistema nato vrne pripravljen seznam plačilnih list. Po klicu uporabim gradnik `BusyIndicator`. Ta na ekranu uporabniku prikaže animacijo, ki mu sporoči, da se podatki še nalagajo, in se izvaja, dokler se podatki povsem ne naložijo.

V primeru, da uporabnik izbere začetek časovnega obdobja na sredini koledarskega meseca, se v aplikaciji začetek obdobja avtomatično popravi na začetek istega meseca. Podobno se zgodi, če uporabnik izbere konec časovnega obdobja na sredini meseca. Razlika je le v tem, da se datum konca avtomatično popravi na konec izbranega meseca.

Primer prilagoditve časovnega obdobja: Uporabnik izbere časovno obdobje od 15.01.2022 do 15.02.2022. Aplikacija bo v tem primeru prilagodila časovno obdobje na obdobje z datumom začetka 01.01.2022 in datumom konca 28.02.2022 (ali 29.02. v primeru prestopnega leta) in novo obdobje uporabila za filtriranje plačilnih list in kreiranje seznama le-teh.

7.5.4 Prenos dokumenta na oblični sistem in njegov predogled

Za prenos datoteke iz zalednega sistema v obličnega moram dostopati preko URL naslova, v katerem moramo navesti ustrezno oData storitev, potrebne parametre, na koncu pa tudi dodati tekst `/$value`, s katerim sistemu povemo, da gre za prenos datoteke in lahko pokliče pravilno metodo v zalednem sistemu. Za predogled dokumenta uporabim instanco razreda PDFViewer, ki nam to omogoča v Fiori aplikaciji. Preko predogleda dokumenta si lahko uporabnik kasneje dokument tudi prenese na napravo ali pa ga natisne.

7.5.5 Prikazovanje obvestil o uspešnem izvajanju aplikacije

Uporabnik mora imeti v aplikaciji za vsako funkcionalnost povratno informacijo, če je bilo izvajanje njegove zahteve uspešno. To lahko enostavno implementiramo z uporabo instance razreda `sap.m.MessageToast`. Ta nam omogoča prikazovanje obvestil na ekranu. Specificirati moramo tudi velikost in koliko časa bo obvestilo prikazano na ekranu. Obvestila o uspešnem izvajanju aplikacije se obarvajo v zeleno barvo, obvestila o napakah pa se obarvajo v rdečo. To lahko implementiramo z uporabo CSS jezika.

7.5.6 Prevodi uporabniškega vmesnika aplikacije

Odločil sem se, da bom aplikacijo prevedel tudi v angleščino. To lahko storimo v ogrodju `i18next`(skrajšano `i18n`)[\[15\]](#), ki je bil razvit v iz za jezik JavaScript in skrbi za prilagajanje računalniške programske opreme različnim jezikom in regijam. Kreiramo datoteki `i18n_sl.properties` in `i18n_en.properties`, notri pa definiramo vse prevode, tekste in besedila za posamezen jezik. Za uporabo le-teh v tekstovnih atributih v XML datoteki lahko do njih dostopamo preko ogrodja `i18n` in sicer na način `text = {i18n>imeTeksta}`. Sistem glede na jezik prijave nato prebere ustrezen tekst, ki je shranjen pod spremenljivko, ki smo jo navedli, in ta tekst uporabi, kjer ga potrebujemo. Če potrebujemo določen tekst v kodi, ga preberemo nekoliko drugače in sicer na način: `this.getView().getModel("i18n").getResourceBundle().getText("imeTeksta")`.

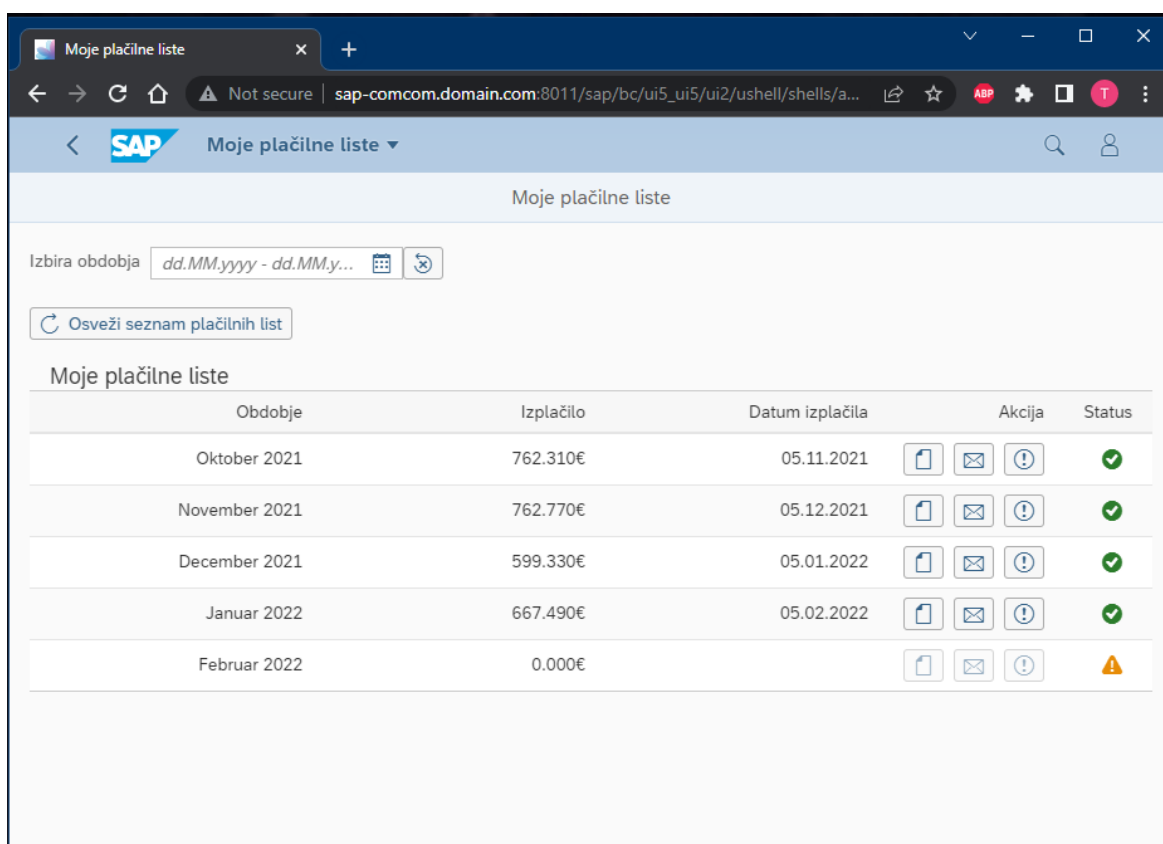
8 TESTIRANJE APLIKACIJE

V tej fazi bom aplikacijo testiral in pri poganjanju le-te odkrival morebitne napake. Po končani fazi testiranja bom odkrite napake odpravil. Testne primere bom smiselno povezal z uporabniškimi zahtevami. Testiranje bi bilo najbolj optimalno, če bi ga izvedel neodvisen preizkuševalec. Ker pa to ni izvedljivo, bom moral testiranje izvesti sam. Pri testiranju bom uporabil koncept vedenjskega testiranja (v izvorniku Black-box testing) [5], ki temelji na zunanjih lastnostih sistema. Za testiranje aplikacije bom uporabil kadrovske številke 16 in 956.

8.1 Pregled plačilnih list

Pri testiranju splošnega pregleda plačilnih list bom najprej uporabil kadrovske številke 956, saj ima zadnjih 6 mesecev obračunane plačilne liste, in kadrovske številke 16, ki zadnjih 6 mesecev nima obračunanih plačilnih list.

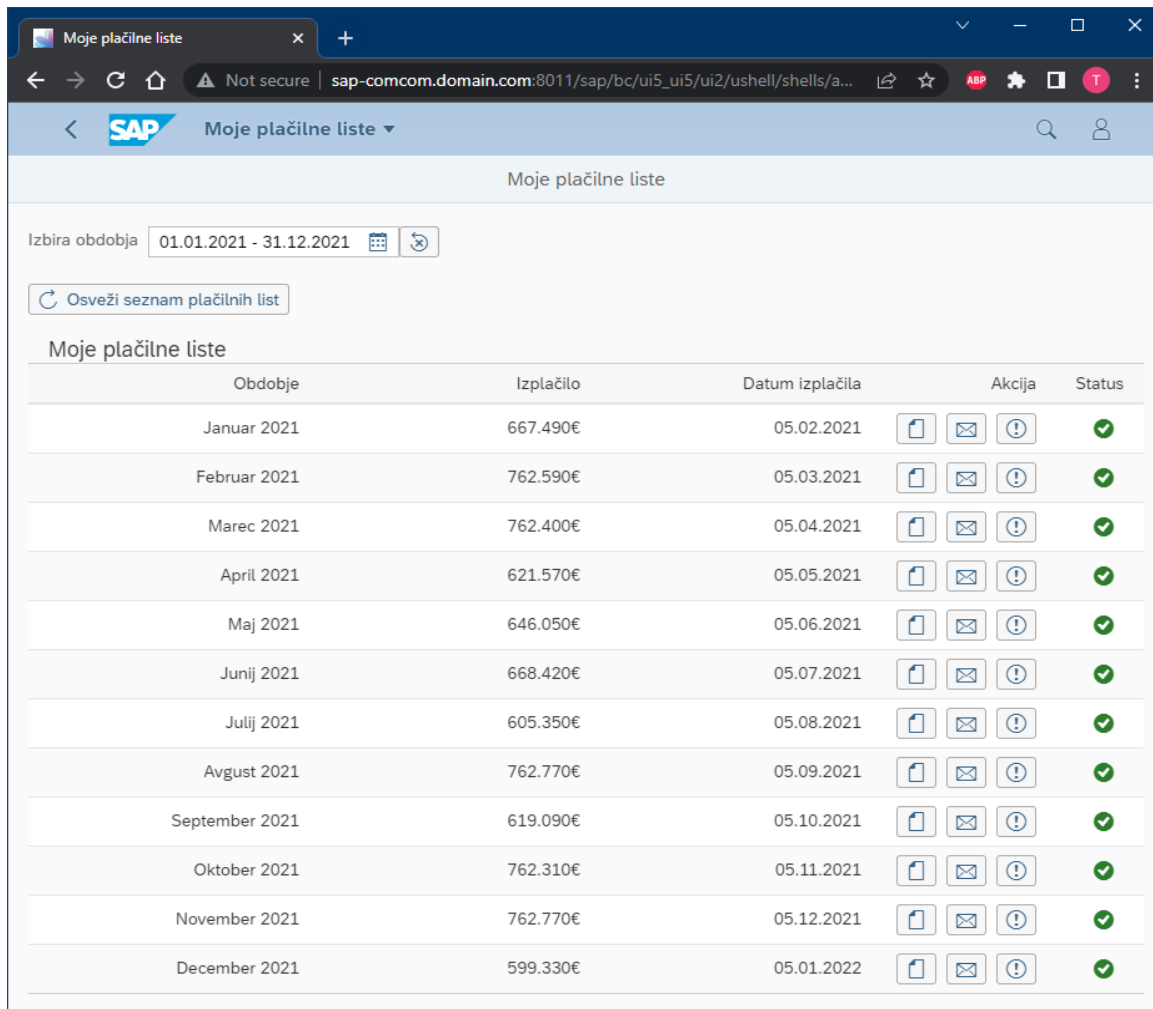
Kadrovska številka 956 ima v zadnjih 6 mesecih obračunanih 5 plačilnih list, zadnja v seznamu pa še ni arhivirana. Prikazani podatki so pravilni in sovpadajo s podatki na zalednem sistemu.



















































Obdobje	Izplačilo	Datum izplačila	Akcija	Status
Oktober 2021	762.310€	05.11.2021	[Print] [Email] [Info]	✓
November 2021	762.770€	05.12.2021	[Print] [Email] [Info]	✓
December 2021	599.330€	05.01.2022	[Print] [Email] [Info]	✓
Januar 2022	667.490€	05.02.2022	[Print] [Email] [Info]	✓
Februar 2022	0.000€		[Print] [Email] [Info]	⚠

Slika 17: Testiranje pregleda plačilnih list za kadrovske 956

Če izberemo drugo obdobje, npr. obdobje prejšnjega leta 01.01.2021 - 31.12.2021, se nam seznam osveži. Tudi tokrat so prikazani podatki ustrezni.

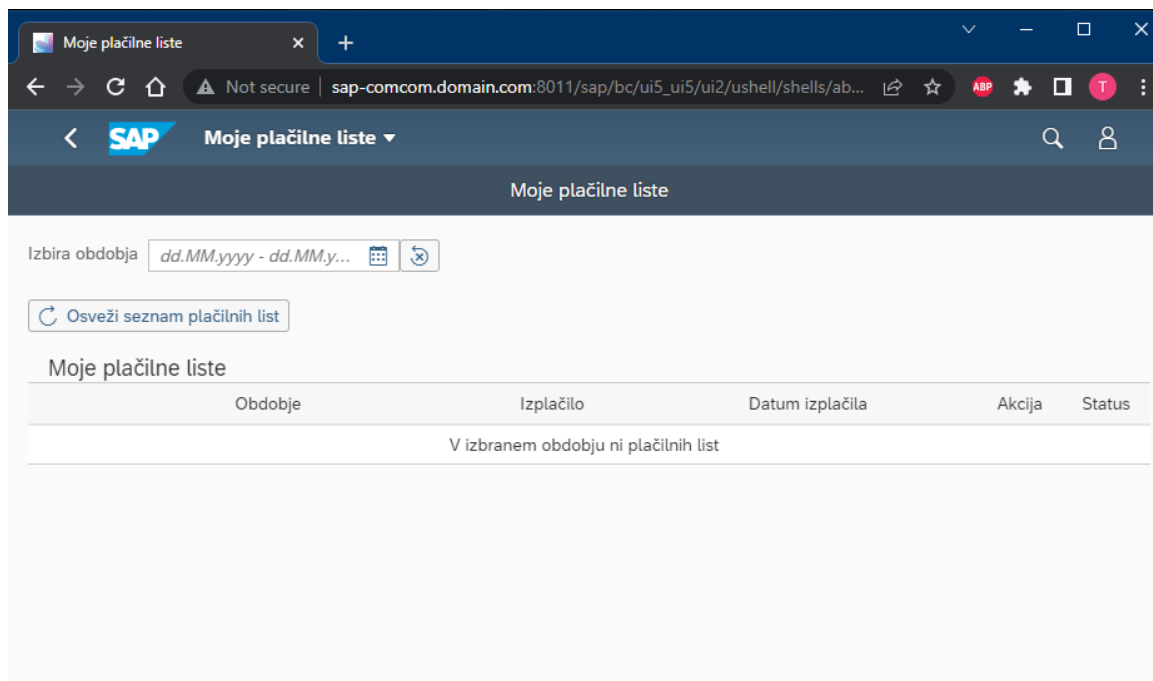


The screenshot shows the SAP Fiori 'Moje plačilne liste' application. At the top, there is a search bar and a user profile icon. Below that, the title 'Moje plačilne liste' is displayed. A filter for the selection period is set to '01.01.2021 - 31.12.2021'. A button labeled 'Osveži seznam plačilnih list' is visible. The main content is a table with the following columns: 'Obdobje', 'Izplačilo', 'Datum izplačila', 'Akcija', and 'Status'. The table lists 12 months of payments for 2021, with the final entry for December 2021 showing a payment of 599.330€ on 05.01.2022. Each row includes icons for document, email, and warning, and a green checkmark in the status column.

Obdobje	Izplačilo	Datum izplačila	Akcija	Status
Januar 2021	667.490€	05.02.2021	  	
Februar 2021	762.590€	05.03.2021	  	
Marec 2021	762.400€	05.04.2021	  	
April 2021	621.570€	05.05.2021	  	
Maj 2021	646.050€	05.06.2021	  	
Junij 2021	668.420€	05.07.2021	  	
Julij 2021	605.350€	05.08.2021	  	
Avgust 2021	762.770€	05.09.2021	  	
September 2021	619.090€	05.10.2021	  	
Oktober 2021	762.310€	05.11.2021	  	
November 2021	762.770€	05.12.2021	  	
December 2021	599.330€	05.01.2022	  	

Slika 18: Testiranje pregleda plačilnih list z izbiro časovnega obdobja

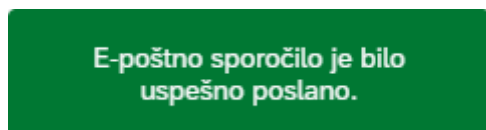
Za tem testiramo še na kadrovske številke 16, ki zadnjih 6 mesecev nima obračunanih plačilnih list. Namesto seznama plačilnih list se nam zato na ekranu izpiše tekst "V izbranem obdobju ni plačilnih list".



Slika 19: Testiranje pregleda plačilnih list pri manjkajočih podatkih

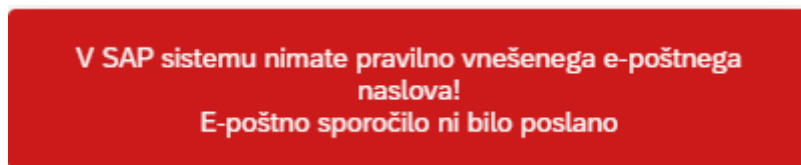
8.2 Pošiljanje dokumenta na e-poštni naslov

Pri testiranju pošiljanja dokumenta moramo biti pozorni na to, ali ima uporabnik vpisan e-poštni naslov v SAP sistemu. Kadrovska številka 956 ima e-poštni naslov vpisan, zato se pošiljanje prek e-pošte izvede uspešno, na ekranu pa se izpiše ustrezno obvestilo.



Slika 20: Obvestilo o uspešno poslanem e-poštnem sporočilu

Kadrovska številka 16 nima vpisanega e-poštnega naslova, zato se na ekranu izpiše obvestilo, da e-poštno sporočilo ni bilo poslano.



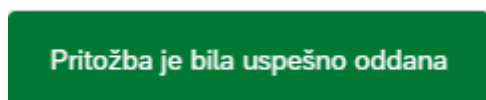
Slika 21: Obvestilo o napaki pri pošiljanju e-poštnega sporočila

8.3 Predogled dokumenta v brskalniku

Opcija predogleda dokumenta mora delovati za vse odobrene plačilne liste na ekranu. Ob testiranju tega pogoja ugotovimo, da se za vsak zapis v seznamu na ekranu prikaže ustrezen dokument.

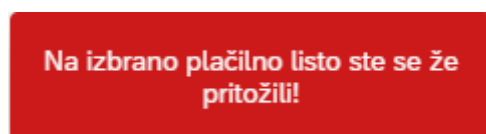
8.4 Oddajanje pritožbe na plačilno listo

Pri testiranju oddajanja pritožb je pomembno, da ni dovoljeno večkratno vlaganje pritožbe na isto plačilno listo. Ko prvič vložimo in potrdimo pritožbo na določeno plačilno listo, se na ekranu prikaže obvestilo, da je bila vloga uspešno oddana.



Slika 22: Obvestilo o uspešno vloženi pritožbi

Če poskusimo ponovno vložiti pritožbo na določeno plačilno listo, se nam na ekranu prikaže opozorilo o napaki.



Slika 23: Obvestilo o napaki pri vlaganju pritožbe

9 MOŽNE KASNEJŠE DOPOLNITVE IN RAZŠIRITVE APLIKACIJE

- Dopolnitev aplikacije z opcijo temnega načina uporabniškega vmesnika

SAPUI5 Fiori aplikacijo bi lahko nadgradil tako, da bi implementiral opcijo za preklon med različnimi temami izgleda uporabniškega vmesnika aplikacije. To bi lahko uresničil, če bi v podjetju nadgradili Fiori sistem na verzijo 3.0 ali kasnejšo, saj trenutna verzija Fiori sistema 2.0, ki jo uporabljamo v podjetju Comcom d.o.o., tega ne omogoča. Na glavno stran aplikacije bi dodal spustni meni (v izvorniku. drop down menu), kjer bi lahko uporabnik izbral temo izgleda, ki mu najbolj ustreza.

10 ZAKLJUČEK

V tej zaključni nalogi sem predstavil vse faze razvoja SAPUI5 Fiori aplikacije za prikaz plačilnih list uporabnika in prikazal implementacijo le-te. Glavni namen omenjene Fiori aplikacije je, da končnemu uporabniku omogoča podroben pregled nad svojimi plačilnimi listami preko enostavnega, modernega in intuitivnega uporabniškega vmesnika. Za razvoj aplikacije sem uporabil kombinacijo programskega jezika ABAP na SAP zalednem sistemu, oData storitve in programskega jezika JavaScript na obličnem sistemu. Aplikacija uporablja tehnologijo HTML5 in se izvaja v internetnem brskalniku na uporabnikovi napravi ali pa v mobilni aplikaciji SAP Fiori aplikaciji.

Ugotovitve:

Pri pisanju te zaključne naloge sem ugotovil, da je za kakovostno aplikacijo potrebno dobro poznavanje uporabniških zahtev, tako funkcijskih kot nefuncijskih. Zato je pomembno sodelovanje med razvijalci in končnim uporabnikom. Prav tako je pomembna faza načrtovanja aplikacije.

Pridobljeno znanje:

Pri izdelavi aplikacije sem poglobil znanje iz programiranja v ABAP in JavaScript programskega jeziku, razvoja SMARTFORMS obrazcev, arhiviranja in razvoja oData storitev. V SAP zalednem sistemu sem bolje spoznal HR modul. Pridobil sem izkušnje pri izdelavi uporabniškega vmesnika za moderne spletne aplikacije in se naučil uporabljati številna moderna razvojna orodja Build.me, SAP Web IDE in razhroščevalnik v brskalniku. Pridobljena znanja mi bodo koristila pri bodoči poklicni karieri.

Celotna izvorna koda projekta je naložena na Github repozitoriju: https://github.com/ThadeusRex/Comcom_placilna_fiori

Video s primerom uporabe aplikacije je naložen na mojem Youtube kanalu in je dosegljiv na internetni povezavi: <https://youtu.be/RgpD2oA60us>. Posnel sem ga s programsko opremo OBS Studio.

Zaznamek: Vsi podatki na prikazanih dokumentih in aplikaciji so izmišljeni in so namenjeni izključno testiranju in ne prikazujejo resničnih podatkov, tudi če so na prvi pogled videti resnični.

11 VIRI IN LITERATURA

- [1] Adobe. (B.D.). What is a PDF? Portable Document Format. Pridobljeno 6. 4. 2022 s <https://www.adobe.com/si/acrobat/about-adobe-pdf.html>
- [2] AO Kaspersky Lab. (5. 7. 2021). What is VPN? How It Works, Types of VPN. Pridobljeno 4. 4. 2022 s <https://www.kaspersky.com/resource-center/definitions/what-is-a-vpn>
- [3] K. J. Dearie, (21. 2. 2022). What Is EULA?. Pridobljeno 6. 4. 2022 s <https://termly.io/resources/articles/what-is-eula/>
- [4] Hack Reactor. (26. 8. 2021). What is JavaScript used for? Pridobljeno 4. 4. 2022 s <https://www.hackreactor.com/blog/what-is-javascript-used-for>
- [5] T. Hamilton, (26. 2. 2022). What is BLACK Box Testing? Techniques, Example & Types. Pridobljeno 4. 4. 2022 s <https://www.guru99.com/black-box-testing.html>
- [6] B. Jamšek, (20. 11. 2018). GDPR: Uredba o varstvu podatkov. Pridobljeno 4. 4. 2022 s <https://mladipodjetnik.si/novice-in-dogodki/novice/gdpr-uredba-o-varstvu-podatkov>
- [7] Lucid Software Inc. (B.D.). UML Use Case Diagram Tutorial. Pridobljeno 4. 4. 2022 s <https://www.lucidchart.com/pages/uml-use-case-diagram>
- [8] C. Marshall, (28. 9. 2017). HTML5: what is it?. Pridobljeno 6. 4. 2022 s <https://www.techradar.com/news/internet/web/html5-what-is-it-1047393>
- [9] Rheinwerk Publishing, Inc. (B.D.), What is ABAP? A Guide to SAP's Coding Language: SAP PRESS. Pridobljeno 4. 4. 2022 s <https://learning.sap-press.com/abap>
- [10] Rheinwerk Publishing, Inc. (B.D.). What is SAP Fiori? A Look at SAP's UI for SAP S/4HANA: SAP PRESS. Pridobljeno 04. 04. 2022 s <https://learning.sap-press.com/sap-fiori>
- [11] SAP (B.D.). What is SAP?: Definition and Meaning. Pridobljeno 4. 4. 2022 s <https://www.sap.com/about/company/what-is-sap.html>
- [12] Tutorials Point Inc. (B.D.). SAP ABAP - Open SQL Overview. Pridobljeno 4. 4. 2022 s https://www.tutorialspoint.com/sap_abap/sap_abap_open_sql_overview.htm
- [13] Wikipedia. (27. 1. 2022). SAP Graphical User Interface. Pridobljeno 4. 4. 2022 s https://en.wikipedia.org/wiki/SAP_Graphical_User_Interface
- [14] Wikipedia. (2. 4. 2022). Open Data Protocol. Pridobljeno 4. 4. 2022 s https://en.wikipedia.org/wiki/Open_Data_Protocol
- [15] Wikipedia. (19. 2. 2022). Internationalization and localization. Pridobljeno 4. 4. 2022 s https://en.wikipedia.org/wiki/Internationalization_and_localization