Implementacija prilagodljivega ERP sistema z malo ali nič kode

Tomaž Bizjak, Gregor Kovač, Tomaž Kozlar

Bilumina d.o.o., Maribor, Slovenija
tomaz.bizjak@bilumina.com, gregor.kovac@bilumina.com, tomaz.kozlar@bilumina.com

Ob ustanovitvi podjetja smo se srečali s problematiko lažjega in hitrejšega razvoja aplikacije, kot smo ga poznali v preteklosti. V preteklosti smo razvijali aplikacije tako, da smo poslovno logiko programirali na nivoju posamezne vnosne maske. S tem načinom smo hoteli prekiniti, saj smo vedeli, da nas čaka veliko dela, poleg tega smo hoteli lansirati izdelek v nekem doglednem času. Po raziskavi smo naleteli na low-code princip razvoja programske opreme. Odločili smo se za svojo low-code rešitev, ki temelji na programskem jeziku SQL z Javo v ozadju. V članku predstavljamo kakšna orodja smo razvili za podporo low-code principu razvoja programske opreme.

Ključne besede:
low-code,
prenova rešitve,
ERP,
SQL,
Java.

Univerzitetna založba Univerze v Mariboru

ISBN 978-961-299-037-4
DOI https://doi.org/10.18690/um.feri.7.2025.5

1 Uvod

1.1. Zgodovina

Podjetje Bilumina je nastalo leta 2011. Na začetku smo bili štirje, od tega eden, ki je zelo dobro poznal vsebino (poslovne procese podjetij, ...) in trije programerji (Java, SQL). Že od prej smo najbolj poznali trgovinski segment trga (retail), zato smo se tudi v novem podjetju odločili, da hočemo v nekem doglednem času lansirati lastno ERP (Enterprise Resource Planning) rešitev za ta segment trga, skupaj s podporo za skladišča, zaloge, različne vrste dokumentov, blagajna (POS), … Najprej smo si okvirno izračunali kakšen obseg dela nas čaka, če želimo to zahtevo (v doglednem času lansirati ERP) tudi izpolniti.

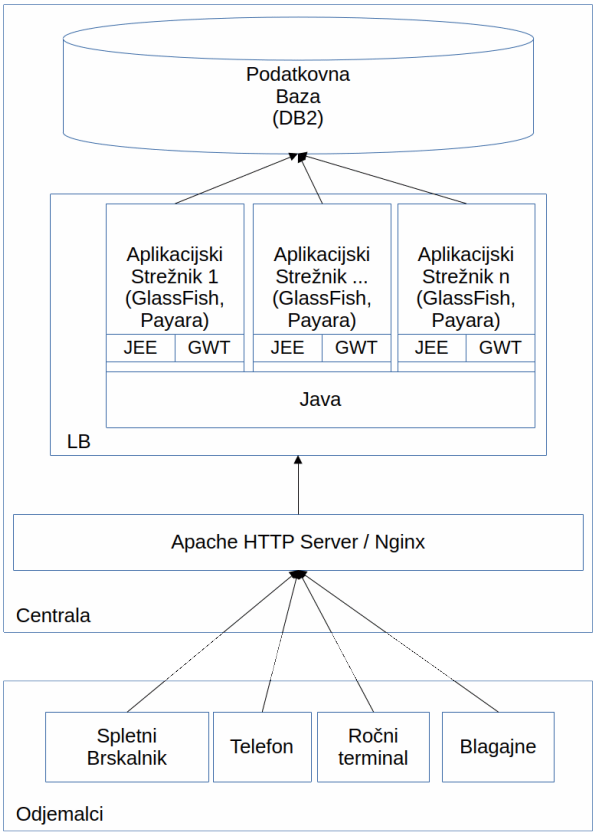
Prišli smo do naslednjih številk:

- nekaj sto vnosnih mask. Ta številka je trenutno narasla na preko tisoč.
- nekaj sto baznih objektov. Ta številka je trenutno narasla na nekaj tisoč baznih objektov (tabel, view-jev, trigger-jev, procedur)

Glede na te številke smo bili že na začetku prepričani, da s tradicionalnim razvojem ne bomo mogli zadostiti zahtevi po doglednem času lansiranja ERP-a. To je bil poglavitni razlog, da smo se odločili, da potrebujemo neko drugo rešitev, nek drug način razvoja. Po pregledu stanja na trgu, kaj sploh obstaja in kakšne te rešitve so (funkcionalnosti, strošek, ...), smo se na koncu odločili, da izdelamo lastno low-code rešitev.

1.2. Arhitektura ERP-a

Odločili smo se za naslednjo arhitekturo ERP rešitve:



Slika 1: Arhitektura ERP rešitve.

Vsi podatki so shranjeni v relacijski podatkovni bazi IBM Db2, do katere, preko aplikacijskih strežnikov, dostopajo klienti preko spletnega brskalnika (Google Chrome, ...), telefoni (Android), ročni terminali (Android) in blagajne (Windows, Java); vsak preko svoje namenske aplikacije. Po navadi odjemalci dostopajo do aplikacijskih strežnikov preko LB (load balancer) komponente, preko kater imajo eno vstopno točko, mi pa lahko za load balancer-jem dodajamo aplikacijske strežnike glede na potrebe. Če je potrebno, pred load balancer postavimo tudi spletni strežnik (Apache HTTP Server ali Nginx). To storimo v primeru, da je potrebno sistem odpreti navzven, recimo v Internet.

1.3. Osnove low-code rešitve

Glede na to, da smo vsi najbolje poznali dva programska jezika, Java in SQL, in da je jezik SQL dosti lažji za učenje kot programski jezik Java, smo se odločili, da za osnovo low-code rešitve uporabimo SQL, ki ga razširimo s funkcijami, ki jih bomo potrebovali.

Primer SQL poizvedbe na vnosni maski: SELECT IME, PRIIMEK FROM UPORABNIKI WHERE ID = :ID

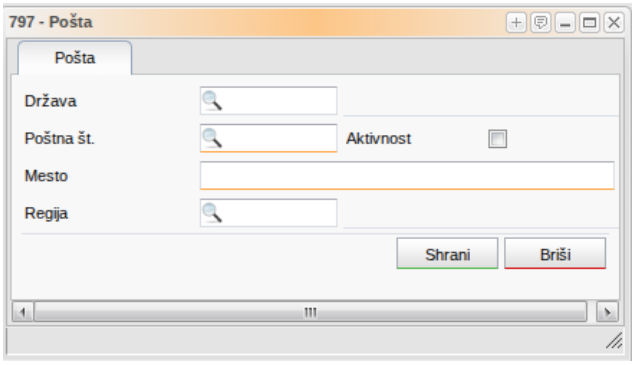
Vrednost parametra ID se uporabi iz maske, ki je to SQL poizvedbo izvedla. Kako in kaj bo predstavljeno v nadaljevanju članka.

2 Orodja

Da smo lahko začeli kreirati strukturo podatkovne baze in vnosnih mask smo morali razviti več orodij.

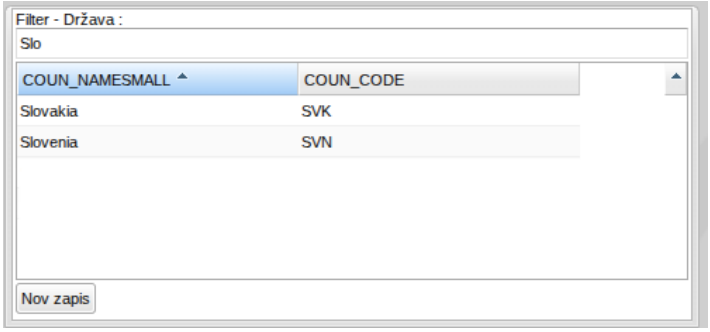
2.1. Design vnosnih mask (t. i. FormTool)

Poglejmo na primeru preprostega vnosa pošt kako kreiramo vnosne maske:



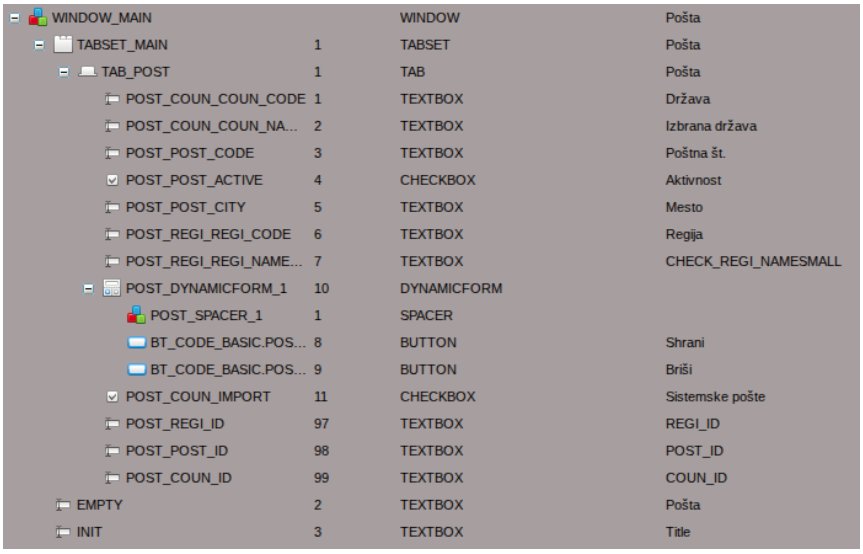
Slika 2: Vnosna maska za vnos pošte.

Prikazana maska ima nekaj vnosnih polj (Država, Poštna št., Mesto, Aktivnost, Regija). Polja z ikono lupe so iskalna polja:



Slika 3: Primer iskanja.

Orodje za urejanje vnosnih mask prikazuje vnosno masko v drevesni strukturi:



WINDOW_MAIN		WINDOW	Pošta
TABSET_MAIN	1	TABSET	Pošta
TAB_POST	1	TAB	Pošta
POST_COUN_COUN_CODE	1	TEXTBOX	Država
POST_COUN_COUN_NA...	2	TEXTBOX	Izbrana država
POST_POST_CODE	3	TEXTBOX	Poštna št.
POST_POST_ACTIVE	4	CHECKBOX	Aktivnost
POST_POST_CITY	5	TEXTBOX	Mesto
POST_REGI_REGI_CODE	6	TEXTBOX	Regija
POST_REGI_REGI_NAME...	7	TEXTBOX	CHECK_REGI_NAMESMALL
POST_DYNAMICFORM_1	10	DYNAMICFORM	
POST_SPACER_1	1	SPACER	
BT_CODE_BASIC.POS...	8	BUTTON	Shrani
BT_CODE_BASIC.POS...	9	BUTTON	Briši
POST_COUN_IMPORT	11	CHECKBOX	Sistemske pošte
POST_REGI_ID	97	TEXTBOX	REGI_ID
POST_POST_ID	98	TEXTBOX	POST_ID
POST_COUN_ID	99	TEXTBOX	COUN_ID
EMPTY	2	TEXTBOX	Pošta
INIT	3	TEXTBOX	Title

Slika 4: Drevesna struktura.

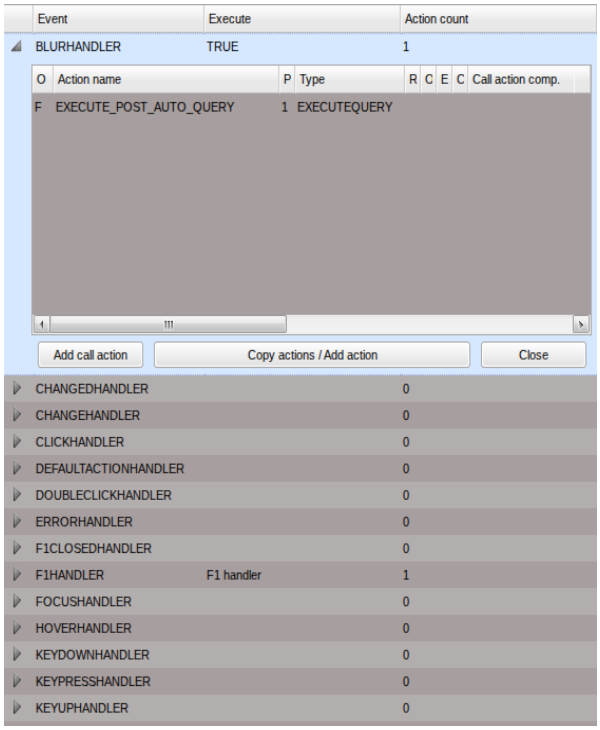
Na sliki 4 vidimo strukturo maske. Vsako polje ima svoje unikatno ime, svoj tip (TEXTBOX, CHECKBOX, BUTTON, …) in labelo (kaj uporabnik vidi ko odpre vnosno masko). Labelo se lahko prevede v primeren jezik uporabnika.
Vsako polje na vnosni maski ima mnogo lastnosti, ki jih lahko nastavimo, da dobimo ustrezen izgled in funkcionalnost vnosne maske.

O	Property name	Value	Type
C	ALIGN	LEFT	STYLE
C	ALLOWUSERVISIBLE	FALSE	STYLE
C	AUTOFIT	TRUE	STYLE
C	CELLSTYLE	DEFAULT	STYLE
C	COLSPAN	1	STYLE
C	HEIGHT	22	STYLE
C	HINTSTYLE	DEFAULT	STYLE
C	HOVERWIDTH	300	STYLE
F	POSITION	8	STYLE
C	PRINTADDWATERMARK	FALSE	STYLE
C	SHOWTITLE	FALSE	STYLE
C	TEXTALIGN	CENTER	STYLE
C	TITLEALIGN	CENTER	STYLE
C	TITLEORIENTATION	LEFT	STYLE
C	TITLESTYLE	DEFAULT	STYLE
C	TITLEVALIGN	TOP	STYLE
C	WIDTH	*	STYLE
C	WRAPTITLE	FALSE	STYLE
C	HINT		TEXT

Slika 5: Nekatere lastnosti polja.

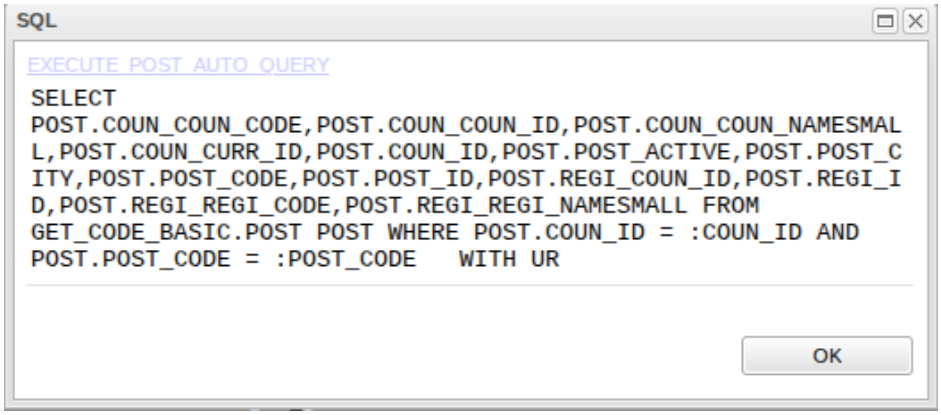
Polju lahko določimo različne skupine lastnosti:

- stilske: kje na maski se polje pojavi, kako je tekst znotraj polja poravnan, ...
- vedenjske: ali se vrednost polja prenese na aplikacijski strežnik, katero je naslednje polje na katerega se prenese vnos, ...
- vsebinske: katere akcije se zaženejo kdaj (slika 6)



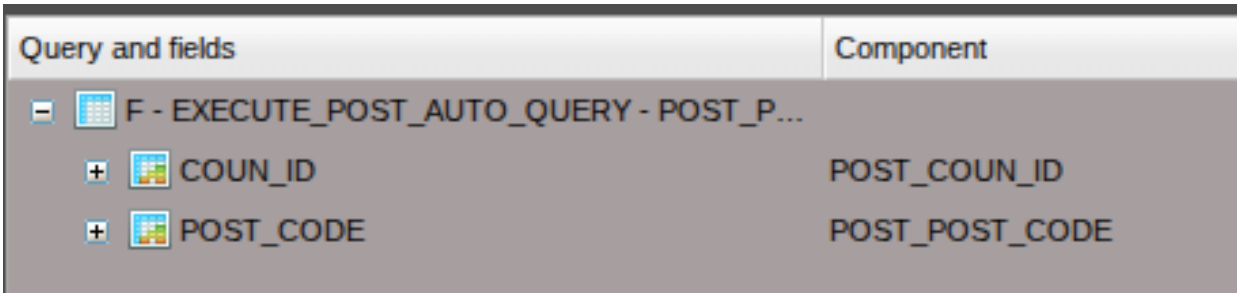
Slika 6: Različni dogodki (event-i) in povezane akcije.

Ob dogodku vnosa podatka v polje »Poštna št.« se sproži akcija EXECUTE_POST_AUTO_QUERY tipa EXECUTEQUERY. Tip EXECUTEQUERY pomeni, da se bo sprožil določen SQL (slika 7) na podatkovni bazi.



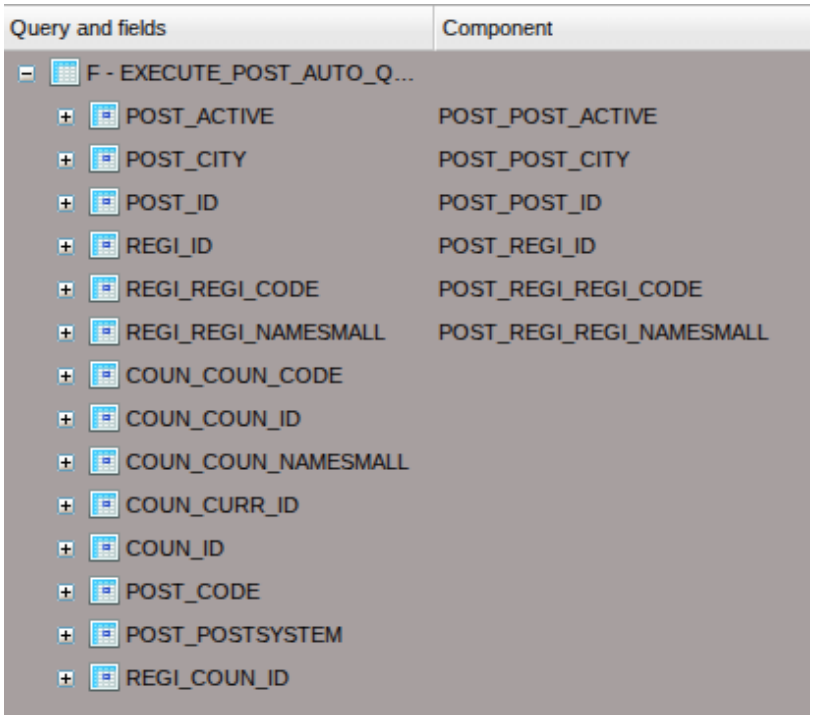
Slika 7: Primer SQL-a.

V SQL-u sta dva parametra (COUN_ID, POST_CODE). Prepoznamo ju, ker imata pred imenom znak :. Vrednost teh dveh parametrov moramo napolniti z ustrezno vrednostjo iz vnosne maske (slika 8).



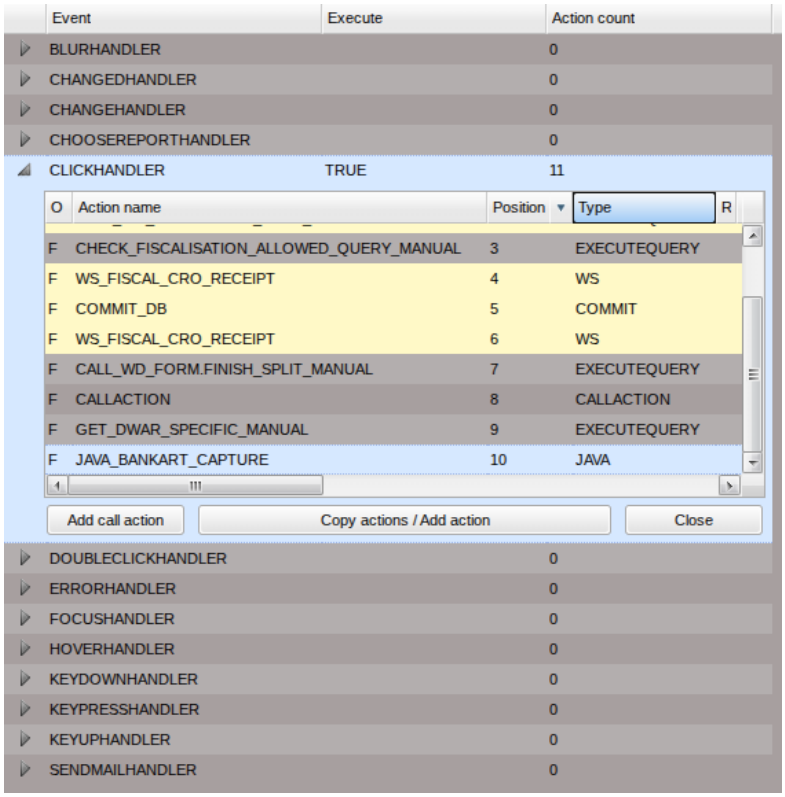
Slika 8: Povezava SQL parametrov s polji na vnosni maski.

Ko se SQL požene in pridobi rezultate iz podatkovne baze, jih mora prikazati na vnosni maski (slika 9).



Slika 9: Povezava polj iz SQL-a in polj na maski.

Poleg SQL-ov lahko akcije na poljih poženejo tudi vnaprej pripravljene akcije napisane v programskem jeziku Java (slika 10).



Slika 10: Primer gumba z večjim številom akcij.

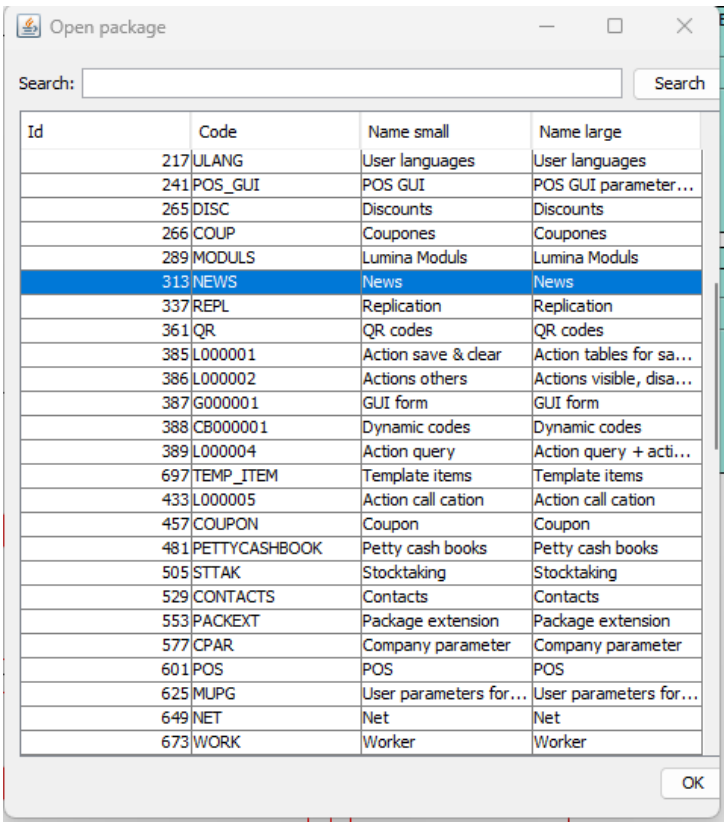
Na sliki 10 vidimo, da lahko ima posamezno polje več akcij, istega ali drugačnega tipa. Na sliki vidim več tipov akcij:

- EXECUTEQUERY: pridobi podatke iz podatkovne baze,
- WS: pridobi podatke iz spletne storitve,
- COMMIT: zaključi transakcijo,
- CALLACTION: pokliči akcijo na drugem polju,
- JAVA: pokliči vnaprej pripravljeno akcijo v programskem jeziku Java.

Pri vseh teh različnih tipih akcij povežemo parametre akcije s polji na maski in rezultat akcije s polji na maski na isti način kot je bilo prikazano pri EXECUTEQUERY akciji (slika 8).

2.2. Design baze (t. i. Linea)

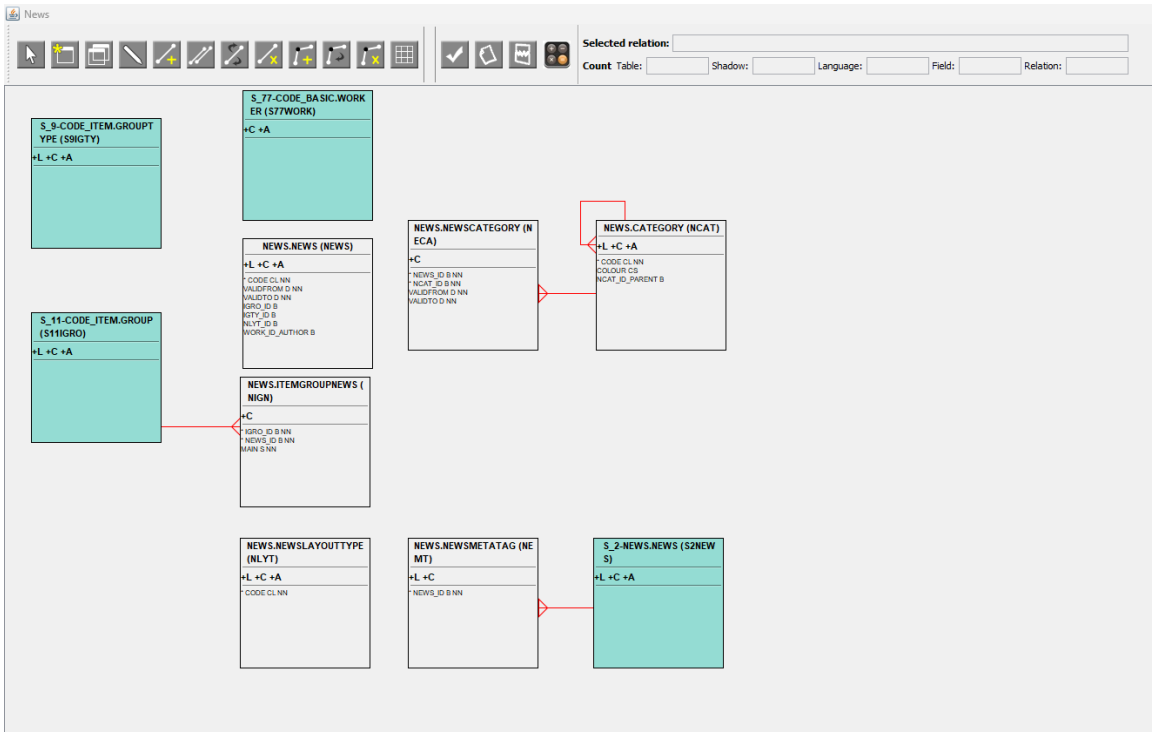
Za hitrejši razvoj strukture baze smo razvili lastno orodje. Tabele so, zaradi lažje vizualizacije, razporejene v posamezne pakete, kar prikazuje slika 11.



Id	Code	Name small	Name large
217	ULANG	User languages	User languages
241	POS_GUI	POS GUI	POS GUI parameter...
265	DISC	Discounts	Discounts
266	COUP	Coupones	Coupones
289	MODULS	Lumina Moduls	Lumina Moduls
313	NEWS	News	News
337	REPL	Replication	Replication
361	QR	QR codes	QR codes
385	L000001	Action save & clear	Action tables for sa...
386	L000002	Actions others	Actions visible, disa...
387	G000001	GUI form	GUI form
388	CB000001	Dynamic codes	Dynamic codes
389	L000004	Action query	Action query + acti...
697	TEMP_ITEM	Template items	Template items
433	L000005	Action call cation	Action call cation
457	COUPON	Coupon	Coupon
481	PETTYCASHBOOK	Petty cash books	Petty cash books
505	STTAK	Stocktaking	Stocktaking
529	CONTACTS	Contacts	Contacts
553	PACKEXT	Package extension	Package extension
577	CPAR	Company parameter	Company parameter
601	POS	POS	POS
625	MUPG	User parameters for...	User parameters for...
649	NET	Net	Net
673	WORK	Worker	Worker

Slika 11: Tabele paketa News.

Slika 12 prikazuje tabele znotraj paketa News.

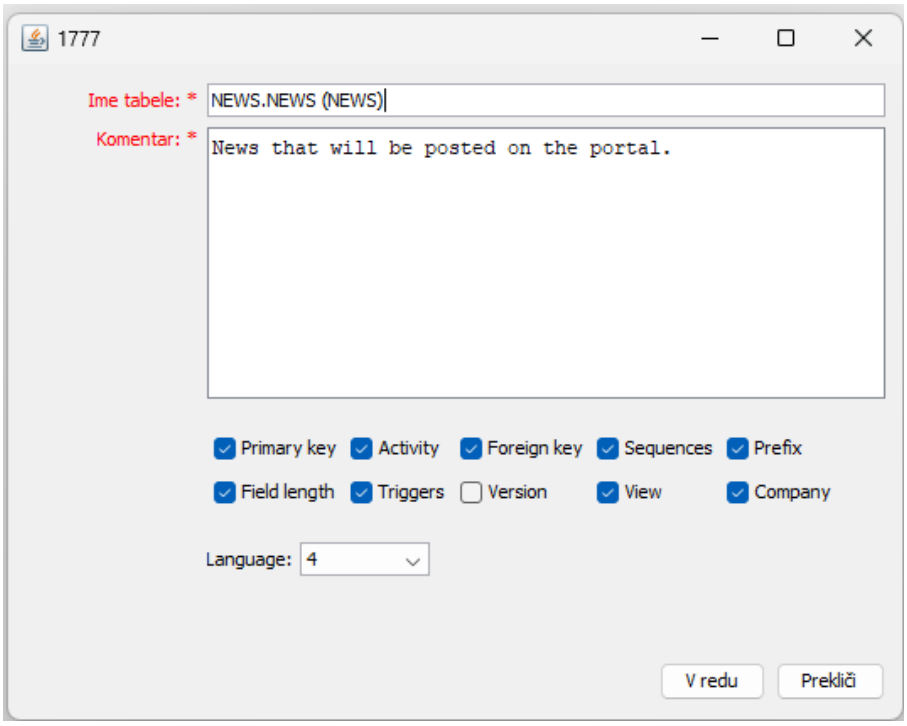


Slika 12: Tabele paketa News.

Tabele v temnejši barvi so iz drugega paketa. Orodje omogoča tudi povezovanje tabel:

- One – to – One,
- One – to – Many,
- Many – to – Many povezave rešujemo z uporabo vmesnih tabel.

Slika 13 prikazuje nastavitve posamezne tabele.

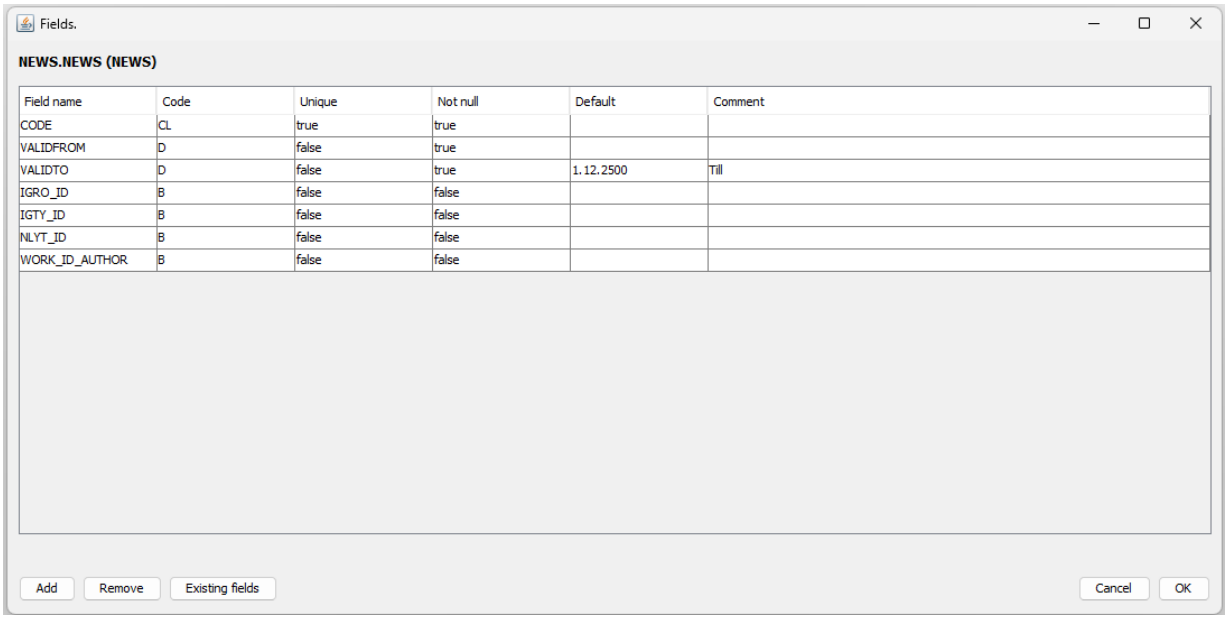


Slika 13: Nastavitve tabele.

Tabeli lahko nastavimo:

- ime tabele,
- opis tabele,
- ali ima tabela primarni ključ,
- ali se za njo kreirajo pogledi,
- ali se za njo kreira ustrezna sekvenca za vrednosti primarnega ključa,
- ali ima ta tabela posebno jezikovno tabelo,
- ali je posamezna vrstica vezana na posamezno podjetje ali je splošna za vsa podjetja.

Slika 14 prikazuje nastavitve kolon določene tabele.



Slika 14: Kolone tabele NEWS.NEWS.

Posamezna kolona ima:

- ime,
- opis,
- podatkovni tip,
- ali dovoljuje vrednost NULL,
- ali je del unikatnega indeksa,
- kakšna je privzeta vrednost.

To orodje pripravi ustrezne tabele, poglede in triggerje na podatkovni bazi, da lahko vnosna maska pravilno zapiše podatke v podatkovno bazo ali pa jih iz nje prebere. Podatek je lahko vezan na posamezno podjetje, lahko pa ima tudi jezikovni del, kot je recimo opis artikla v slovenščini, angleščini, …

3 Mnenja uporabnikov

3.1. Tomaž Bizjak

S prihodom v podjetje sem tudi sam začel ustvarjati enostavne vnosne maske. Že na začetku me je navdušilo dejstvo, da ima orodje že vnaprej implementiranih veliko splošnih gradnikov, ki jih je treba le povezati z ustreznimi tabelami in stolpci v podatkovni bazi. Ta pristop močno pospeši razvoj, saj zmanjša količino ročnega programiranja in s tem tudi možnost napak pri implementaciji. Zaradi tega je tudi uvajanje novih sodelavcev hitrejše in manj zahtevno, saj se lahko že v kratkem času vključijo v razvojne naloge.

Druga pomembna prednost orodja je enostavno prevajanje vnosnih mask, kar je še posebej priročno za stranke, ki poslujejo na več različnih trgih. S tem se bistveno poenostavi prilagajanje aplikacij lokalnim uporabnikom, kar izboljša uporabniško izkušnjo in hkrati zmanjša potrebo po dodatni tehnični podpori. Možnost hitrega preklapljanja med jeziki je pogosto tudi ena izmed prvih funkcionalnosti, ki jih opazijo naše mednarodne stranke, kar pogosto vpliva na njihovo pozitivno začetno izkušnjo.

Na splošno lahko rečem, da je orodje preprosto za uporabo, a hkrati dovolj zmogljivo, da omogoča hitro razširjanje funkcionalnosti glede na potrebe posameznega projekta. Zaradi svoje prilagodljivosti in preglednosti pa IT-kadrom omogoča, da več časa posvetijo zahtevnejšim tehničnim izzivom znotraj podjetja, namesto da bi se ukvarjali z rutinskimi opravili.

3.2. Tomaž Kozlar

Z orodjem FormTool sem začel delati, ko je omogočalo izgradnjo enostavnih šifrantov. Orodje se je z leti postopoma razvijalo in še vedno se. Sedaj že obsega možnost gradnje vedno kompleksnejših vnosnih mask, klicanja zunanjih akcij, še vedno pa se dodajajo novi gradniki, ki jih uporabljamo glede na potrebe stranke in nas. Z orodjem FormTool se lahko hitro naredi nova maska ali pa se doda novo funkcionalnost na obstoječo masko.

Prednost takega načina dela vidim v tem, da je manj možnosti za napake pri razvoju, saj gre za omejen nabor gradnikov, a hkrati omogoča veliko fleksibilnost, ko se lahko različne akcije, ki jih prožijo gumbi/polja napišejo tudi v programskem jeziku Java. Ker je orodje FormTool spletno orodje tudi ni potrebna inštalacija, prav tako pa lahko na različnih formah dela več ljudi hkrati.

Orodje Linea sem pomagal razviti tudi sam. Orodje omogoča grafični prikaz tabel v podatkovni bazi, kolon v posamezni tabeli in relacij med njimi (ER diagram). Omogoča tudi enostaven pregled in upravljanje velikih količin tabel, saj se lahko združujejo tabele v pakete. V Linea-i se nato lahko upravlja z vsakim paketom posebej, kar omogoča večjo preglednost. Prav tako je z njim enostavno pokazati in razložiti relacije ter tabele drugim zaposlenim, ki delajo na istem projektu, saj je lažje stvari predstaviti na grafičnem pogledu. Ker omogoča tudi tiskanje diagrama ter opisov tabel in posameznih polj v tabeli, je tako možno narisano tudi poslati stranki in to uporabiti kot dokumentacijo.

3.3. Dolores Sonički

FormTool je naš interni pripomoček za gradnjo mask v spletnem ERP sistemu. Maske predstavljajo uporabniške obrazce za vnos, pregled in obdelavo podatkov, ki jih zaposleni uporabljamo pri vsakodnevnih procesih. Orodje omogoča, da obrazce sestavljamo samostojno – brez zahtevnega programiranja, če imamo osnovno razumevanje strukture podatkov in delovanja ERP okolja.

Čeprav nisem iz tehničnega okolja, sem se v uporabo FormTool-a hitro vživela. Osnovno razumevanje delovanja ERP sistema mi je pri tem zelo pomagalo. Vmesnik je pregleden, gradniki pa so zasnovani tako, da se jih z nekaj uvoda da hitro osvojiti. Uporaba je logična in strukturirana, kar omogoča samostojno delo tudi tistim, ki niso programerji, a imajo nekaj vsebinskega predznanja.

Vsak gradnik ima svoje lastnosti, ki jih prilagajamo prek enostavnih nastavitev. Mnoga polja že vključujejo prednastavljene akcije, kar pohitri delo in omogoča večjo osredotočenost na funkcionalnost obrazca.

FormTool podpira kompleksne strukture – drevesne prikaze, tabele znotraj tabel, dinamične pogoje, privzete vrednosti in še mnogo več. S tem močno povečamo fleksibilnost, skrajšamo čas prilagajanja procesov ter omogočimo, da pri razvoju ERP okolja sodeluje širši krog uporabnikov z različnih področij.

4 Zaključek

Low-code pristop se je izkazal kot zanesljiva in učinkovita alternativa klasičnemu razvoju ERP sistemov. Omogoča hitro prilagajanje spremembam, večjo vključenost uporabnikov in boljšo izrabo IT virov. Takšen pristop bo v prihodnosti igral čedalje pomembnejšo vlogo pri digitalizaciji poslovanja.