

Blagajna za maloprodajnu trgovinu

Šušnjara, Antonio

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:929218>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom](#).

Download date / Datum preuzimanja: **2024-07-13**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA**

Stručni studij

BLAGAJNA ZA MALOPRODAJNU TRGOVINU

Završni rad

Antonio Šušnjara

Osijek, 2022.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****Obrazac Z1S: Obrazac za imenovanje Povjerenstva za završni ispit na preddiplomskom stručnom studiju**

Osijek, 02.12.2022.

Odboru za završne i diplomske ispite**Imenovanje Povjerenstva za završni ispit
na preddiplomskom stručnom studiju**

Ime i prezime Pristupnika:	Antonio Šušnjara
Studij, smjer:	Prediplomski stručni studij Računarstvo
Mat. br. Pristupnika, godina upisa:	AI 4639, 27.07.2017.
OIB Pristupnika:	18346305487
Mentor:	Robert Šojo, mag. ing. comp.
Sumentor:	Doc. dr. sc. Tomislav Galba
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Marina Peko, dipl. ing.
Član Povjerenstva 1:	Robert Šojo, mag. ing. comp.
Član Povjerenstva 2:	mr.sc. Željko Štanfel
Naslov završnog rada:	Blagajna za maloprodajnu trgovinu
Znanstvena grana završnog rada:	Programsko inženjerstvo (zn. polje računarstvo)
Zadatak završnog rada	Kreirati aplikaciju blagajne za maloprodajnu trgovinu. Aplikacija ima mogućnost korištenja od strane administratora i regularnog blagajnika. Administrator može odobravati vraćanje proizvoda, unos novih proizvoda, korisnika (blagajnika), izmijene cijena, radne sate pojedinog blagajnika i ostala analitika. Regularni blagajnik ima mogućnost prijave, kreiranje računa, pridruživanje popusta i tome slično. Opisati tehnologije u procesu izrade aplikacije, opisati pojedine dijelove aplikacije, te kreirati potpuno funkcionalnu aplikaciju. Tema rezervirana za: Antonio Šušnjara Sumentor s FERIT-a: Tomislav Galba
Prijedlog ocjene pismenog dijela ispita (završnog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene od strane mentora:	02.12.2022.
<i>Potvrda mentora o predaji konačne verzije rada:</i>	<i>Mentor elektronički potpisao predaju konačne verzije.</i>
	Datum:

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**IZJAVA O ORIGINALNOSTI RADA**

Osijek, 14.12.2022.

Ime i prezime studenta:

Antonio Šušnjara

Studij:

Preddiplomski stručni studij Računarstvo

Mat. br. studenta, godina upisa:

AI 4639, 27.07.2017.

Turnitin podudaranje [%]:

9

Ovom izjavom izjavljujem da je rad pod nazivom: **Blagajna za maloprodajnu trgovinu**

izrađen pod vodstvom mentora Robert Šojo, mag. ing. comp.

i sumentora Doc. dr. sc. Tomislav Galba

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija. Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

Sadržaj

1. UVOD	1
1.1 Zadatak završnog rada	1
2. PREGLED PODRUČJA	2
2.1 Fiskalna.hr	2
2.2 Loyverse POS	2
2.3 eHopper POS	3
3. KORIŠTENE TEHNOLOGIJE	4
3.1 Java	4
3.2 JavaFX	5
3.3 IntelliJ IDEA	5
3.4 Scene Builder	6
3.5 SQL	7
4. FUNKCIONALNOST APLIKACIJE	8
4.1 Baza podataka	8
4.2 MVC arhitektura	10
4.3 Funkcionalnost prijave	12
4.4 Funkcionalnost administratora	13
4.5 Funkcionalnost zaposlenika	21
5. KORIŠTENJE APLIKACIJE	28
5.1 Administrator	28
5.1.1 Upravljanje zaposlenicima	28
5.1.2 Upravljanje računima	30
5.1.3 Upravljanje artiklima	30
5.2 Zaposlenik	31
6. ZAKLJUČAK	33
LITERATURA	34

SAŽETAK.....	35
ABSTRACT.....	36

1. UVOD

Ovaj rad prolazi kroz kreiranje i korištenje GUI (engl. *Graphic User Interface*) aplikacije za korištenje u sustavu blagajne maloprodajne trgovine zajedno s bazom podataka u kojoj se spremaju podaci o artiklima u trgovini i njihovo stanje. Aplikacija omogućava dvije vrste korisnika: administrator i zaposlenik. Obje vrste korisnika imaju drugačije obaveze i s time drugačije mogućnosti pri korištenju iste aplikacije.

U drugom poglavlju su istražene aplikacije i sustavi u istom području. U trećem poglavlju su opisane tehnologije korištene u aplikaciji. U četvrtom poglavlju je opisana funkcionalnost aplikacije i u petom poglavlju je objašnjeno korištenje aplikacije.

1.1 Zadatak završnog rada

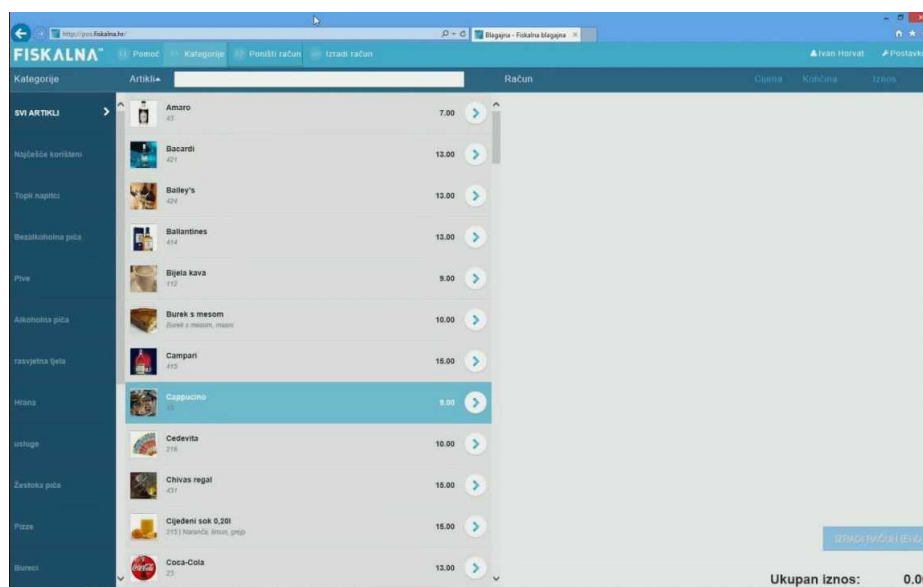
Kreirati aplikaciju blagajne za maloprodajnu trgovinu miješane robe. Aplikacija ima mogućnost korištenja od strane administratora i regularnog blagajnika. Administrator može odobravati vraćanje proizvoda, unos novih proizvoda, kreiranje blagajnika, izmjene cijena, radne sate pojedinog blagajnika i ostala analitika. Regularni blagajnik ima mogućnost prijave, izdavanje računa, pridruživanje popusta i tome slično. Opisati tehnologije u procesu izrade aplikacije, opisati pojedine dijelove aplikacije, te kreirati potpuno funkcionalnu aplikaciju.

2. PREGLED PODRUČJA

U ovom poglavlju su ukratko istražene funkcionalnosti i dizajn aplikacija u istom području. Iz prikazanih sučelja aplikacija se može vidjeti da je jednostavan dizajn zajednička odlika svih aplikacija za blagajnu.

2.1 Fiskalna.hr

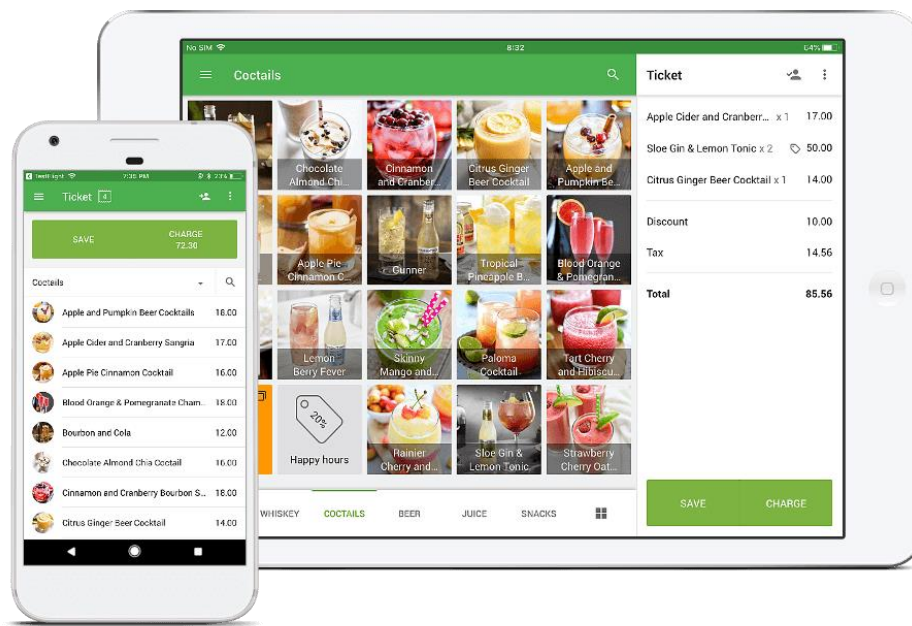
Fiskalna.hr je tvrtka koja prodaje opremu za blagajnu i uz to prodaje aplikacije za mobitel, tablet, računalo i internet preglednik s kojima se koristi ta oprema. Sučelje aplikacije Fiskalna.hr u internet pregledniku je prikazano u slici 2.1 [1]. Aplikacija Fiskalna.hr ima jednostavan dizajn namijenjen za lagano korištenje ali i dalje omogućava sve potrebne funkcionalnosti.



Slika 2.1. Sučelje aplikacije Fiskalna.hr u internet pregledniku.

2.2 Loyverse POS

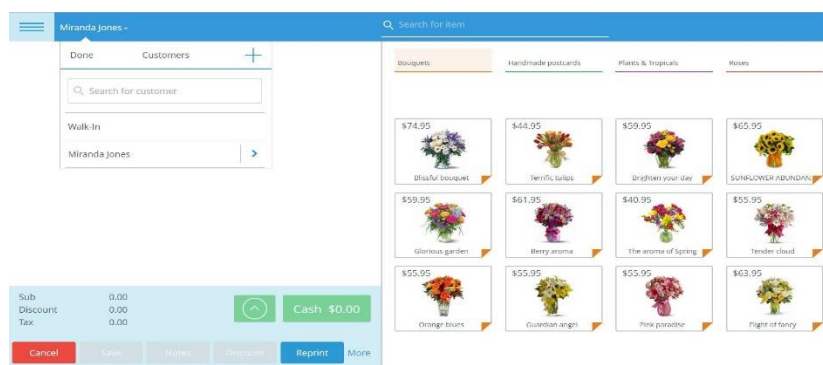
Loyverse POS (engl. *Point-Of-Sale*) je aplikacija za blagajnu koja sve osnovne funkcionalnosti nudi besplatno, primarno za mobilne uređaje ali naplaćuju dodatne funkcionalnosti. Uz to omogućuje i poseban način rada za restorane gdje više računala može biti povezano što omogućuje da se u kuhinji na računalu prikazuju narudžbe čim su zaprimljene. Sučelje Loyverse POS aplikacije na mobitelu i tabletu je prikazano u slici 2.2 [2].



Slika 2.2. Sučelje Loyverse POS aplikacije.

2.3 eHopper POS

eHopper POS je aplikacija za blagajnu koja ima besplatnu verziju ali s jako malo funkcionalnosti i kompatibilna je s više uređaja. Sučelje eHopper aplikacije je prikazano slikom 2.3 [3].



Slika 2.3. Sučelje eHopper POS aplikacije.

3. KORIŠTENE TEHNOLOGIJE

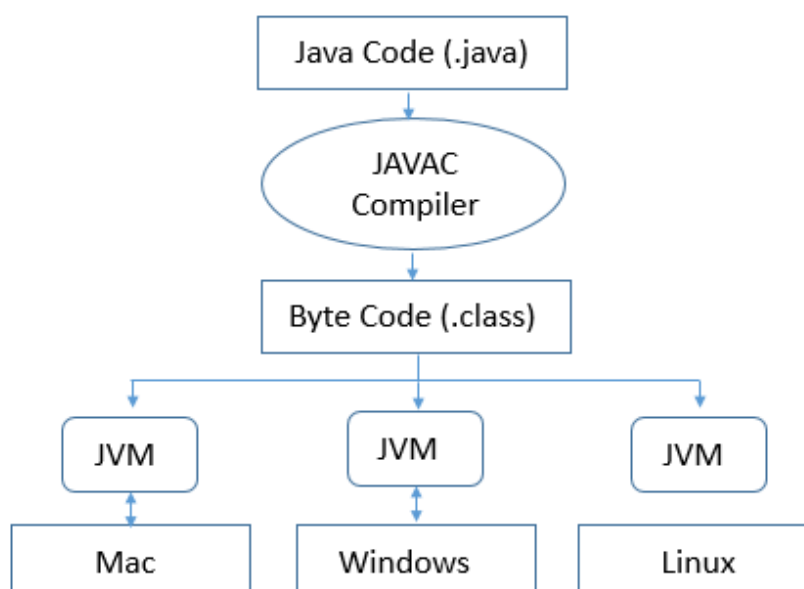
U ovom poglavlju su opisane korištene tehnologije što podrazumijeva korištene programske jezike i programe za pisanje i pravljenje aplikacije.

3.1 Java

Java je besplatni programski jezik koji je 1995. godine napravila tvrtka Sun Microsystems, a od 2009. godine je u vlasništvu Oracle Corporation tvrtke. Java je ažuran jezik koji je jedan od najkorišteniji i najpopularnijih jezika u svijetu. Razlog popularnosti je sličnost sintakse C/C++ jeziku ali ne zahtjeva prevođenja koda za svaku platformu zasebno zato što koristi JVM (engl. *Java Virtual Machine*) koji omogućava izvršavanje istog Java koda na više platformi [4].

Java je objektno-orijentirani programski jezik što je popularniji način programiranja jer omogućava definiranje klasa i njihovih metoda i s time omogućava kreiranja objekata i ostvarivanje apstrakcije, polimorfizma, nasljeđivanja i enkapsulacije [5].

U slici 3.1 je prikazano kako se Java kod prevodi u Byte Code i kako se isti Byte Code izvršava na različitim platformama tako što svaka platforma ima svoj JVM [6].



Slika 3.1. Prikaz izvršavanja istog koda na više platformi.

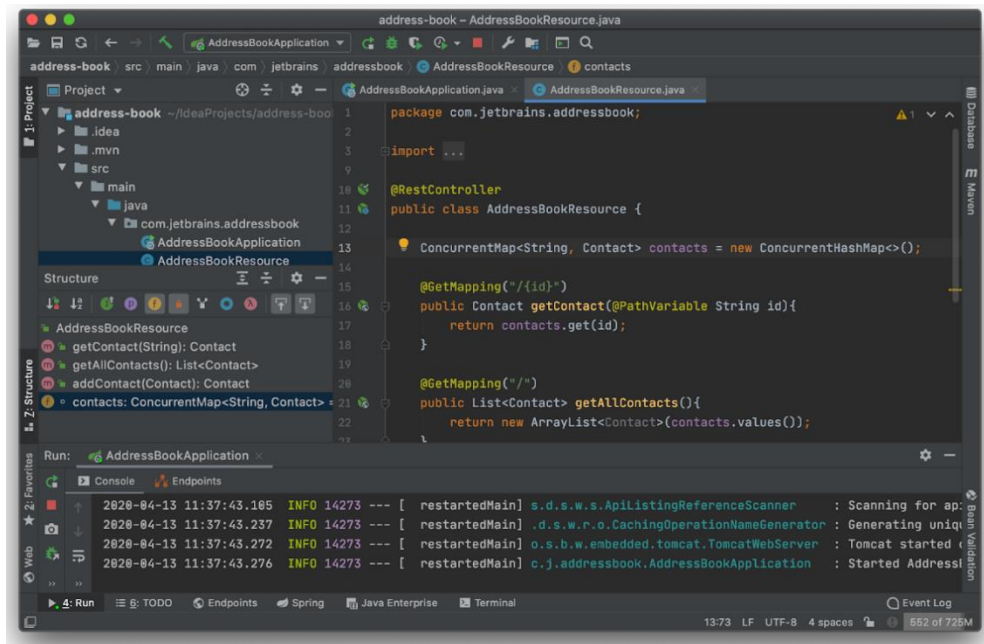
3.2 JavaFX

JavaFX je dodatak otvorenog koda (engl. *open-source code*) Java programskom jeziku koji omogućava kreiranje grafičkog korisničkog sučelja i definiranje komunikacije između Java programa i objekata sučelja. Kao dodatak Java jeziku zadržava mogućnost izvršavanja i pokretanja na više platformi koristeći JVM. JavaFX je napravljen kao zamjena za Swing koji je stariji način kreiranja sučelja [7].

JavaFX sučelje se može kreirati vizualno načinom premještanjem elemenata (engl. *drag-and-drop*) koristeći program Scene Builder ili pisanjem koda u FXML jeziku, koji je baziran na XML-u (engl. *Extensible Markup Language*). Izgled FXML koda je prikazan u slici 3.2.



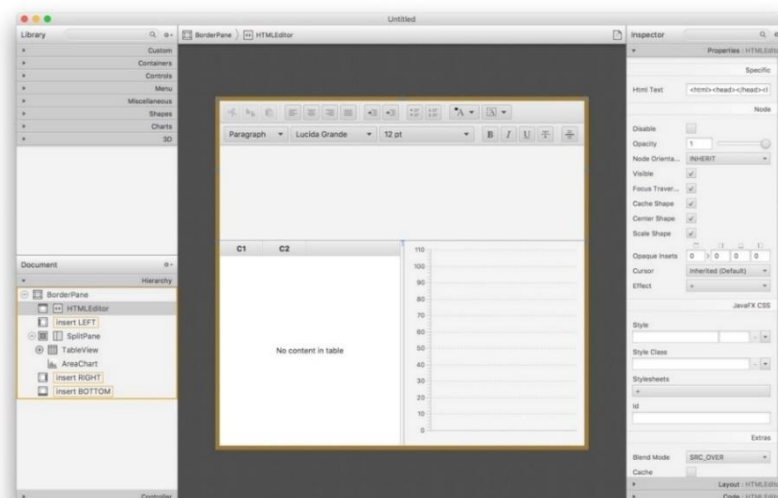
```
mainScene.fxml x
1  <?xml version="1.0" encoding="UTF-8"?>
2  <?language javascript?>
3
4  <?import javafx.scene.control.Label?>
5  <?import javafx.scene.layout.VBox?>
6  <?import javafx.scene.control.Button?>
7
8  <VBox xmlns="http://javafx.com/javafx"
9      xmlns:fx="http://javafx.com/fxml">
10 <Label fx:id="mainTitle" text="Hello world!"/>
11 <Label fx:id="subTitle" text="This is a simple demo application."/>
12 <Button fx:id="mainButton" text="Click me!" onAction="buttonClicked()"/>
13 <fx:script>
14     function buttonClicked() {
15         mainButton.setText("Click me again!")
16     }
17 </fx:script>
18 </VBox>
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
```



Slika 3.3. Prikaz IntelliJ IDEA sučelja.

3.4 Scene Builder

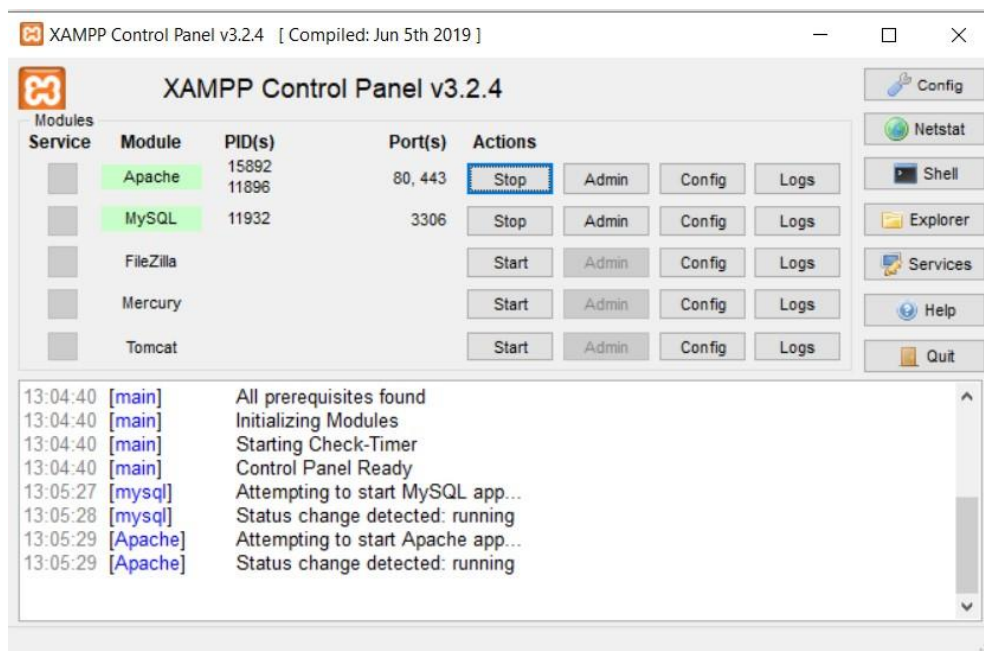
Scene Builder je besplatan i program otvorenog koda za vizualno kreiranje grafičkog sučelja. Radi načinom premještanja elemenata koji automatski generira FXML kod za korištenje u programu [9]. Prikaz sučelja programa Scene Builder za premještanje elemenata je prikazano slikom 3.4.



Slika 3.4. Prikaz Scene Builder sučelja.

3.5 SQL

SQL (engl. *Structured Query Language*) je programski jezik za relacijske baze podataka koji se zasniva na postavljanju „upita“ poslužitelju. Omogućava kreiranje i korištenje baze podataka koja sadržava tablice i definira njihove odnose [10]. Komunikacija između baze podataka i Java programa se radi postavljanjem „upita“ od strane Java programa SQL poslužitelju. Za rad baze podataka je korišten program XAMPP koji sadrži phpMyAdmin. phpMyAdmin je program u kojem se koristi MySQL i omogućava vizualno sučelje u Internet pregledniku. Sučelje programa XAMPP s pokrenutim SQL poslužiteljem je prikazano slikom 3.5.



Slika 3.5. Prikaz XAMPP sučelja.

4. FUNKCIONALNOST APLIKACIJE

U ovom poglavlju su opisane funkcionalnosti i dizajn programa, opis programskog koda aplikacije i dizajn baze podataka s kojom aplikacija radi.

4.1 Baza podataka

Baza podataka sastoji se od 5 tablica: „*article*“, „*articles_in_invoice*“, „*invoice*“, „*employee*“, „*work_hours*“. Tablica „*article*“ sadrži artikle u ponudi, primarni ključ, naziv, kategoriju, cijenu i trenutno stanje tog artikla u trgovini. Tablica „*article*“ prikazana je slikom 4.1.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	category	enum('food', 'cleaning', 'tools', 'drinks', 'school...)		No	undefined			Change Drop More
<input type="checkbox"/>	3	name			No	None			Change Drop More
<input type="checkbox"/>	4	price			No	None			Change Drop More
<input type="checkbox"/>	5	storage			No	None			Change Drop More

Slika 4.1. Tablica „*article*“.

Tablica „*invoice*“ sadrži račune koji se naprave tokom rada u aplikaciji. Tablica „*invoice*“ sadrži primarni ključ, strani ključ zaposlenika koji je primarni ključ tablice „*employee*“, vrijeme izdavanja računa i popust primijenjen na taj račun u postotcima. Tablica „*invoice*“ prikazana je slikom 4.2.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	idEmployee			No	None			Change Drop More
<input type="checkbox"/>	3	time			No	current_timestamp()			Change Drop More
<input type="checkbox"/>	4	discount			No	0			Change Drop More

Slika 4.2. Tablica „*invoice*“.

Tablica „*articles_in_invoice*“ je tablica koja služi kao relacija između tablica „*article*“ i „*invoice*“ tako što nema primarni ključ, nego sadrži dva strana ključa koji su primarni ključevi tablica između kojih pravi relaciju. Uz to sadrži i broj za količinu artikla u računu. Tablica „*articles_in_invoice*“ prikazana je slikom 4.3.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 idArticle 🔑	int(8)			No	None			Change Drop More
<input type="checkbox"/>	2 idInvoice 🔑	int(8)			No	None			Change Drop More
<input type="checkbox"/>	3 count	int(5)			No	None			Change Drop More

Slika 4.3. Tablica „*articles_in_invoice*“.

Tablica „*employee*“ sadrži podatke o zaposlenicima, primarni ključ, ime, prezime, korisničko ime, lozinku i oznaku za administratora sustava. Tablica „*employee*“ prikazana je slikom 4.4.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id 🔑	int(8)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 name	varchar(30)	utf8mb4_croatian_ci		No	None			Change Drop More
<input type="checkbox"/>	3 surname	varchar(30)	utf8mb4_croatian_ci		No	None			Change Drop More
<input type="checkbox"/>	4 user 🔑	varchar(30)	utf8mb4_croatian_ci		No	None			Change Drop More
<input type="checkbox"/>	5 pass	varchar(30)	utf8mb4_croatian_ci		No	None			Change Drop More
<input type="checkbox"/>	6 admin	tinyint(1)			No	0			Change Drop More

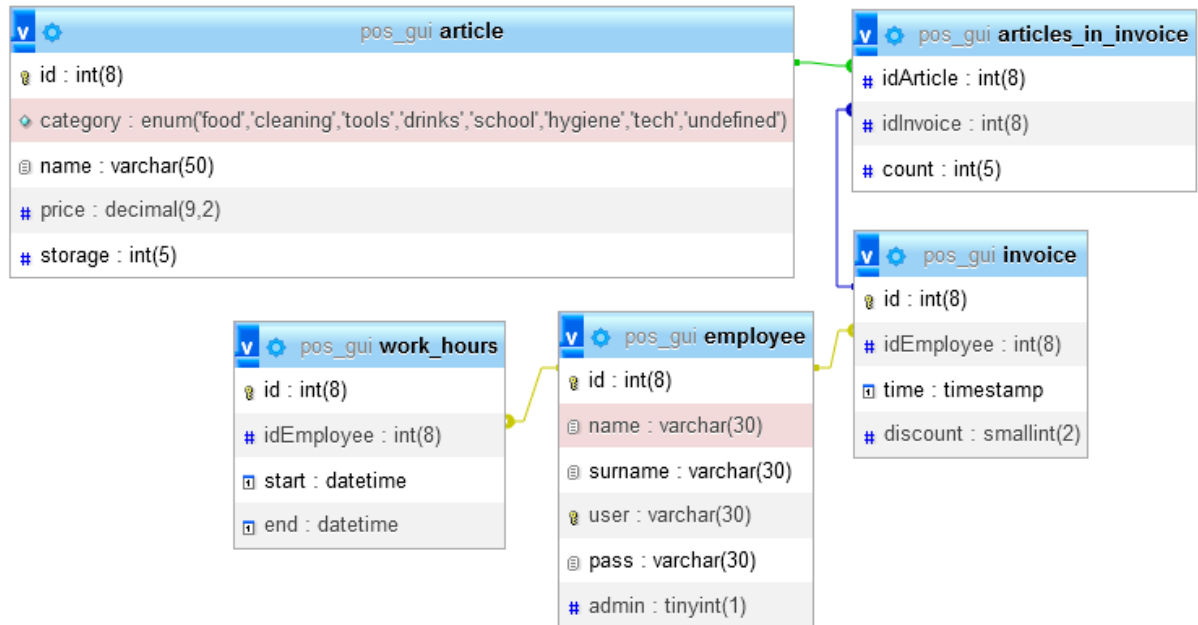
Slika 4.4. Tablica „*employee*“.

Tablica „*work_hours*“ sadrži podatke o početku i kraju rada zaposlenika koji se prate prijavom i odjavom u aplikaciju. Tablica „*work_hours*“ sadrži primarni ključ, strani ključ zaposlenika, početak i kraj rada u vremenskom zapisu. Tablica „*work_hours*“ prikazana je slikom 4.5.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id 🔑	int(8)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 idEmployee 🔑	int(8)			No	None			Change Drop More
<input type="checkbox"/>	3 start	datetime			No	None			Change Drop More
<input type="checkbox"/>	4 end	datetime			No	current_timestamp()			Change Drop More

Slika 4.5. Tablica „*work_hours*“.

PhpMyAdmin omogućuje vizualni prikaz relacija u bazi podataka. Dijagram relacija je prikazan slikom 4.6.



Slika 4.6. Dijagram relacija baze podataka.

Kako bi koristili bazu podataka u programu potrebno je spojiti aplikaciju s bazom podataka. Spajanje s bazom podataka koje se koristi na potrebnim mjestima je prikazano slikom 4.7.

```

public Connection getConnection() {
    Connection conn;
    try {
        conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/pos_gui", "user: \"root\", password: \"");
        return conn;
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return null;
}

```

Slika 4.7. Programski kod za dohvaćanje konekcije s bazom podatak.

4.2 MVC arhitektura

MVC (engl. *Model - View - Controller*) je arhitektura s kojom se lagano mogu odrediti komponente programskog koda i njihove odgovornosti. „*Model*“ komponente sadrže opis složenih podataka, konstruktor i metode postavljanja i dohvaćanja vrijednosti. Najčešće se koriste kao prikaz tablice iz baze podataka u programskom kodu. „*View*“ je komponenta koja definira vizualni prikaz podataka u prozoru aplikacije. „*View*“ komponente u ovoj aplikaciji su napravljene koristeći JavaFX i FXML „*markup*“ jezik. „*Controller*“ komponenta upravlja zadanom „*View*“ komponentom i koristi podatke iz „*Model*“ komponente za rad.

Korištene „Model“ komponente u ovoj aplikaciji su: „Article.java“ za prikaz tablice „article“, „Invoice.java“ za prikaz tablice „invoice“, „ArticleInInvoice.java“ koja je drugačija od tablice „articles_in_invoice“ po tome što sadrži ime artikla, količinu i izračunatu cijenu jer se u programu koristi samo kao element unutar konteksta računa, „Employee.java“ za prikaz tablice „employee“ i „WorkHours.java“ za prikaz tablice „work_hours“. Primjer „Model-a“ „ArticleInInvoice.java“ prikazan je slikom 4.8.

```
public class ArticleInInvoice {
    private String article;
    private Integer amount;
    private Float price;

    public ArticleInInvoice(String article, Integer amount, Float price) {
        this.article = article;
        this.amount = amount;
        this.price = price;
    }

    public String getArticle() { return article; }
    public void setArticle(String article) { this.article = article; }
    public Integer getAmount() { return amount; }
    public void setAmount(Integer amount) { this.amount = amount; }
    public Float getPrice() { return price; }
    public void setPrice(Float price) { this.price = price; }
}
```

Slika 4.8. Model „ArticleInInvoice.java“.

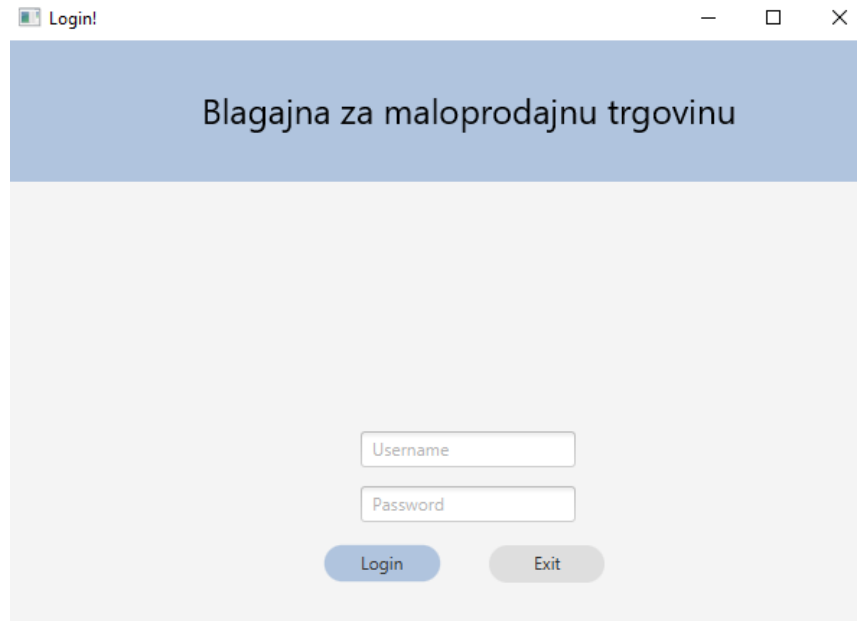
Korištene „View“ komponente i pripadajuće „Controller“ komponente su:

- loginView.fxml – LoginController.java
- adminView.fxml – AdminController.java
 - o adminArticleView.fxml – AdminArticleController.java
 - o adminEmployeeView.fxml – AdminEmployeeController.java
 - o adminInvoiceView.fxml – AdminInvoiceController.java
- employeeView.fxml – EmployeeController.java
 - o employeeMainView.fxml – EmployeeMainController.java
 - o employeeInventoryView.fxml – EmployeeInventoryController.java
 - o employeePersonalView.fxml – EmployeePersonalController.java

„View“ i „Controller“ komponente su podijeljene jer se unutar jednog „View-a“ može instancirati neki drugi „View“ koji ima svoj pripadajući „Controller“.

4.3 Funkcionalnost prijave

Pri pokretanju programa koristi se JavaFX metoda „*start()*“ koja instancira „*loginView.fxml*“ prikazan na slici 4.9.



Slika 4.9. Sučelje za prijavu.

Pri instanciranju „*loginView-a*“ se kreira i „*LoginController*“ objekt za kontrolu „*View-a*“ za prijavu. „*LoginController*“ sadrži programski kod koji nakon pritiska na tipku „*Login*“ pokreće metodu „*login()*“. Metoda „*login()*“ nakon provjere jesu li potrebna polja popunjena, pravi upit bazi podataka i dohvaća rezultat zaposlenika ako su uneseni točni podaci. Na slici 4.10 je prikazano dohvaćanje zaposlenika iz baze podataka.

```
private void login() {
    if (tfUsername.getText() == null || tfUsername.getText().trim().isEmpty()
        || tfPassword.getText() == null || tfPassword.getText().trim().isEmpty()) {
        lbMsg.setText("Fill both fields!");
    } else {
        Connection conn = getConnection();
        String query = "SELECT * FROM employee " +
            "WHERE user ='" + tfUsername.getText() + "' " +
            "AND pass='" + tfPassword.getText() + "'";
        PreparedStatement st;
        ResultSet rs;
        try {
            Stage stage = (Stage) btnLogin.getScene().getWindow();
            st = conn.prepareStatement(query);
            rs = st.executeQuery();
        }
    }
}
```

Slika 4.10. Programski kod za dohvaćanje zaposlenika.

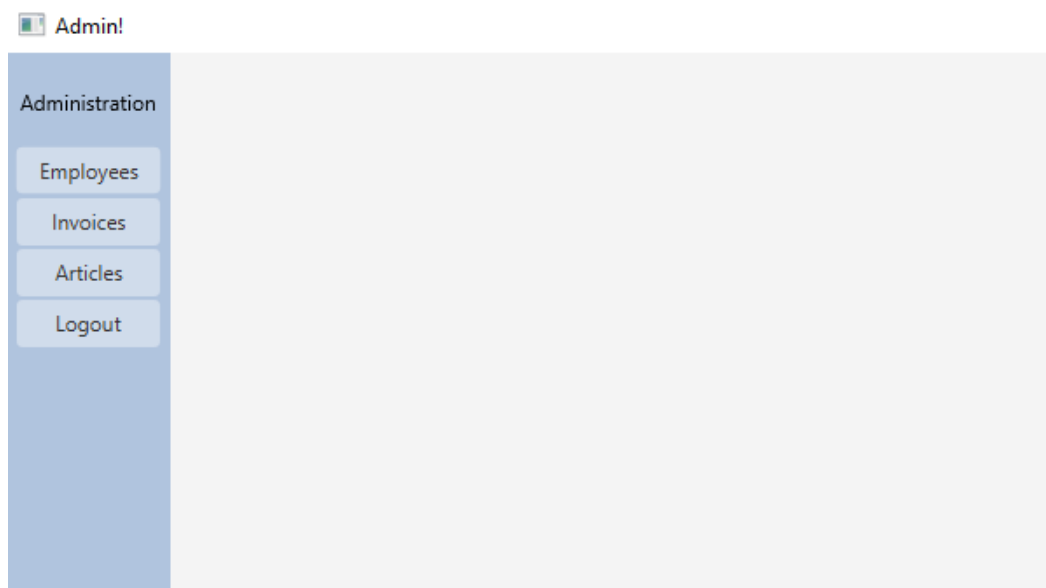
Ako dohvaćeni zaposlenik nije administrator otvara se „*employeeVew*“ i „*EmployeeController-u*“ se prosljeđuje „*Employee*“ zbog korištenja podataka zaposlenika u daljnjim „*Controller-ima*“. Ako je dohvaćeni zaposlenik administrator, programski kod se grana i otvara „*adminView*“. Grananje i otvaranje pripadajućih „*View-ova*“ je prikazano na slici 4.11.

```
if (rs.getInt("admin") == 1) {
    FXMLLoader fxmLoader = new FXMLLoader(MainApp.class.getResource("adminView.fxml"));
    Scene scene = new Scene(fxmLoader.load(), 1200, 700);
    stage.setTitle("Admin!");
    stage.setScene(scene);
    stage.show();
} else {
    FXMLLoader fxmLoader = new FXMLLoader(MainApp.class.getResource("employeeView.fxml"));
    Scene scene = new Scene(fxmLoader.load(), 1200, 700);
    EmployeeController employeeController = fxmLoader.getController();
    employeeController.setUser(
        new Employee(rs.getInt("id"),
            rs.getString("name"),
            rs.getString("surname"),
            rs.getString("user"),
            rs.getString("pass")));
    stage.setTitle("POS");
    stage.setScene(scene);
    stage.show();
}
```

Slika 4.11. Programski kod za otvaranje pripadajućih „*View-ova*“ nakon uspješne prijave.

4.4 Funkcionalnost administratora

Prvo sučelje koje se prikazuje administratoru nakon prijave je prikazano na slici 4.12.



Slika 4.12. Sučelje aplikacije za administratora.

Sučelje za administratora nakon prijave je prazno dok se ne odabere jedna od opcija za rad. Pritiskom tipke s lijeve strane prazni prostor se popuni jednim od „View-ova“ pod „adminView“. Tipka „Logout“ vraća aplikaciju na sučelje za prijavu. Programski kod za otvaranje pripadajućih „View-ova“ je prikazan slikom 4.13.

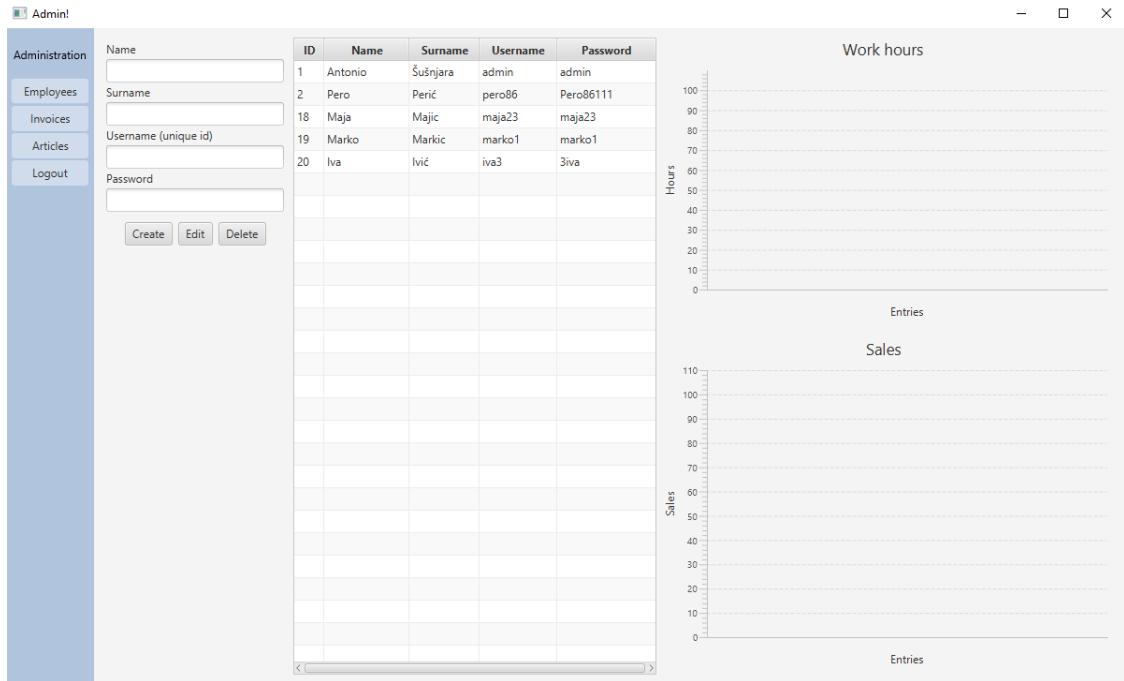
```
@FXML
protected void showEmployees() throws IOException {
    AdminController.contentPane = FXMLLoader.load(getClass().getResource("adminEmployeeView.fxml"));
    try {
        this.pane.getChildren().clear();
        this.pane.getChildren().add(AdminController.contentPane);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

@FXML
protected void showArticles() throws IOException {
    AdminController.contentPane = FXMLLoader.load(getClass().getResource("adminArticleView.fxml"));
    try {
        this.pane.getChildren().clear();
        this.pane.getChildren().add(AdminController.contentPane);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

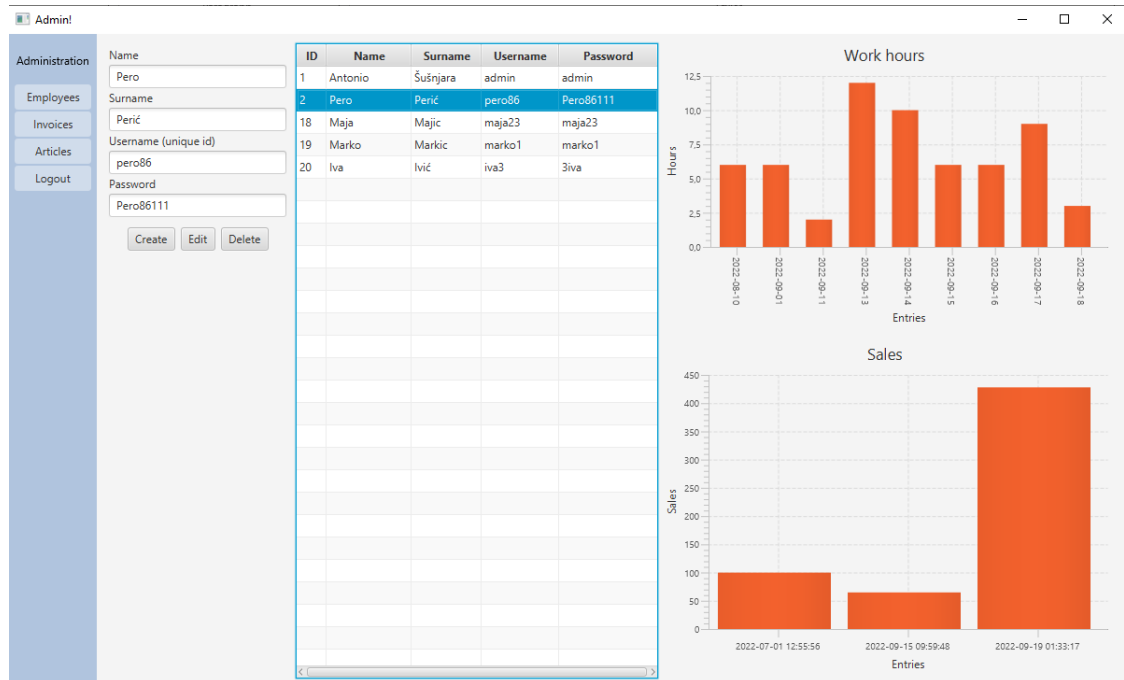
@FXML
protected void showInvoices() throws IOException {
    AdminController.contentPane = FXMLLoader.load(getClass().getResource("adminInvoiceView.fxml"));
    try {
        this.pane.getChildren().clear();
        this.pane.getChildren().add(AdminController.contentPane);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Slika 4.13. Programski kod odabira sučelja za rad administratora.

Aplikacija nakon pritiska tipke „Employees“ otvara „adminEmployeeView“. Sučelje aplikacije za administraciju zaposlenika je prikazano na slici 4.14. Sučelje za administraciju zaposlenika sadrži formu za unos novih zaposlenika, uređivanje podataka i brisanje postojećih zaposlenika. Uz to sadrži tablicu svih zaposlenika i grafove koji prikazuju zadnjih deset unosa za radne sate i prodaju. Forma i grafovi se popune dvostrukim klikom na redak željenog zaposlenika. Na slici 4.15 je prikazano sučelje s popunjenim podacima.



Slika 4.14. Sučelje za administraciju zaposlenika.



Slika 4.15. Popunjeno sučelje za administraciju zaposlenika.

Na slici 4.16 je prikazana metoda za popunjavanje tablice pri inicijalizaciji „View-a“ podacima zaposlenika koje dohvati metodom „*getEmployees()*“. Na slici 4.17 je prikazana metoda „*getEmployees()*“ koja dohvaća sve zaposlenike iz baze podataka.

```
private void updateTable() {
    ObservableList<Employee> employees = getEmployees();

    colId.setCellValueFactory(new PropertyValueFactory<>("id"));
    colName.setCellValueFactory(new PropertyValueFactory<>("name"));
    colSurname.setCellValueFactory(new PropertyValueFactory<>("surname"));
    colUser.setCellValueFactory(new PropertyValueFactory<>("username"));
    colPass.setCellValueFactory(new PropertyValueFactory<>("password"));

    tvEmployees.setItems(employees);
}
```

Slika 4.16. Programski kod popunjavanja tablice.

```
public ObservableList<Employee> getEmployees() {
    ObservableList<Employee> employeeList = FXCollections.observableArrayList();
    Connection conn = getConnection();
    String query = "SELECT * FROM employee";
    Statement st;
    ResultSet rs;
    try {
        st = conn.createStatement();
        rs = st.executeQuery(query);
        Employee employee;
        while (rs.next()) {
            employee = new Employee(
                rs.getInt("id"),
                rs.getString("name"),
                rs.getString("surname"),
                rs.getString("user"),
                rs.getString("pass"));
            employeeList.add(employee);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return employeeList;
}
```

Slika 4.17. Programski kod dohvaćanja zaposlenika.

Na slici 4.18 je prikazana metoda za unos novih zaposlenika. Metode za uređivanje i brisanje rade na označenom zaposleniku. Prikaz upita za bazu podataka pri uređivanju i brisanju su na slikama 4.19 i 4.20.

```

@FXML
private void onCreateEmployee() {
    if (isEmpty()) {
        Alert a = new Alert(Alert.AlertType.WARNING);
        a.setContentText("Fill all of text fields");
        a.show();
    } else {
        String query = "INSERT INTO employee VALUES(NULL," +
            "'" + tfName.getText() + "'," +
            "'" + tfSurname.getText() + "'," +
            "'" + tfUser.getText() + "'," +
            "'" + tfPass.getText() + "', '0')";

        Connection conn = getConnection();
        Statement st;
        try {
            st = conn.createStatement();
            st.executeUpdate(query);
            updateTable();
        } catch (Exception e) {
            Alert a = new Alert(Alert.AlertType.WARNING);
            a.setContentText("Could not create a new Employee");
            a.show();
            e.printStackTrace();
        }
    }
}
}

```

Slika 4.18. Programski kod unosa novog zaposlenika.

```

Employee employee = tvEmployees.getSelectionModel().getSelectedItem();

String query = "UPDATE employee SET " +
    "name='" + tfName.getText() + "'," +
    "surname='" + tfSurname.getText() + "'," +
    "user='" + tfUser.getText() + "'," +
    "pass='" + tfPass.getText() + "'" +
    "WHERE id='" + employee.getId() + "'";

```

Slika 4.19. Upit za uređivanje zaposlenika.

```

Employee employee = tvEmployees.getSelectionModel().getSelectedItem();

String query = "DELETE FROM employee " +
    "WHERE user='" + employee.getUsername() + "' " +
    "AND id='" + employee.getId() + "'";

```

Slika 4.20. Upit za brisanje zaposlenika.

Grafovi se popunjavaju dohvaćanim podacima iz baze podataka za odabranog zaposlenika. Upiti za bazu podataka su prilagođeni tako da dohvaćaju podatke u traženom obliku. Upit za

dohvaćanje radnih sati je prikazan na slici 4.21, a upit za dohvaćanje podataka o prodaji je prikazan na slici 4.22.

```
String query = "SELECT * FROM " +  
    "(SELECT TIMESTAMPDIFF(MINUTE, start, end) AS min, CONVERT(start, DATE) AS date " +  
    "FROM work_hours " +  
    "WHERE idEmployee=" + clickedRow.getId() + " " +  
    "ORDER BY start DESC limit 0,10) AS selection " +  
    "ORDER BY date ASC";
```

Slika 4.21. Upit za dohvaćanje radnih sati zaposlenika.

```
String query = "SELECT * FROM " +  
    "(SELECT SUM(a.price*ai.count) AS price, CONVERT(i.time, DATETIME) AS date, i.discount AS discount " +  
    "FROM article a " +  
    "INNER JOIN articles_in_invoice ai ON a.id = ai.idArticle " +  
    "INNER JOIN invoice i ON ai.idInvoice = i.id " +  
    "WHERE i.idEmployee = '" + clickedRow.getId() + "' " +  
    "GROUP BY i.id " +  
    "ORDER BY date DESC limit 0,10) AS selection " +  
    "ORDER BY date ASC";
```

Slika 4.22. Upit za dohvaćanje podataka o prodaji zaposlenika.

Pritiskom tipke „Invoices“ u lijevom stupcu aplikacija otvara „*adminInvoicesView*“ koji sadrži tablicu svih računa iz baze podataka, detaljan ispis odabranog računa i tipku za brisanje odabranog računa. Na slici 4.23 je prikazano sučelje nakon što je popunjeno dvostrukim klikom na željeni račun u tablici.

The screenshot shows a web application interface with a sidebar on the left containing navigation links: Administration, Employees, Invoices, Articles, and Logout. The main area is divided into two panels. The left panel displays a table of invoices with columns for id, Employee, Time, and Discount / %. The right panel, titled 'Invoice details', shows a table of items with columns for Article, Amount, and Price. Below this table, it displays the employee name (iva3), the time of issue (2022-09-19 00:10:39), the price (89.5), the price after a 10% discount (80.55), and the unique ID (46). A 'Delete this invoice' button is located at the bottom of the details panel.

id	Employee	Time	Discount / %
46	iva3	2022-09-19 00:10:39	10
15	maja23	2022-09-18 21:32:37	0
16	maja23	2022-09-18 21:35:01	0
17	maja23	2022-09-18 21:35:52	0
18	maja23	2022-09-18 21:35:58	0
19	maja23	2022-09-18 21:36:01	0
20	maja23	2022-09-18 21:36:08	0
21	maja23	2022-09-18 21:36:16	0
22	maja23	2022-09-18 21:36:19	0
23	maja23	2022-09-18 21:36:22	0
24	maja23	2022-09-18 21:36:43	20
25	maja23	2022-09-18 21:37:09	0
26	maja23	2022-09-18 21:37:11	0
27	maja23	2022-09-18 21:37:14	0
43	maja23	2022-09-19 00:02:12	0
44	maja23	2022-09-19 00:02:19	20
47	maja23	2022-09-19 01:30:47	0
28	marko1	2022-09-18 21:50:36	0
29	marko1	2022-09-18 21:50:44	0
30	marko1	2022-09-18 21:50:48	0
31	marko1	2022-09-18 21:50:52	0
32	marko1	2022-09-18 21:50:54	0
33	marko1	2022-09-18 21:50:56	0
34	marko1	2022-09-18 21:50:59	0
35	marko1	2022-09-18 21:51:01	0
36	marko1	2022-09-18 21:55:12	0
37	marko1	2022-09-18 23:46:52	0

Article	Amount	Price
Bleach	1	13.0
Detergent	3	46.5
White bread	5	30.0

Employee: iva3
Time of issue: 2022-09-19 00:10:39
Price: 89.5
Price after discount(10%): 80.55
Unique ID: 46

Delete this invoice

Slika 4.23. Sučelje za administraciju računa.

Na slici 4.24 je prikaz metode „*getInvoices()*“ s upitom za bazu podataka koji dohvaća i korisničko ime zaposlenika koji je napravio račun. Na slici 4.25 je prikaz metode „*getArticles()*“ kojom se popunjava lista „*Invoice details*“ s artiklima iz odabranog računa.

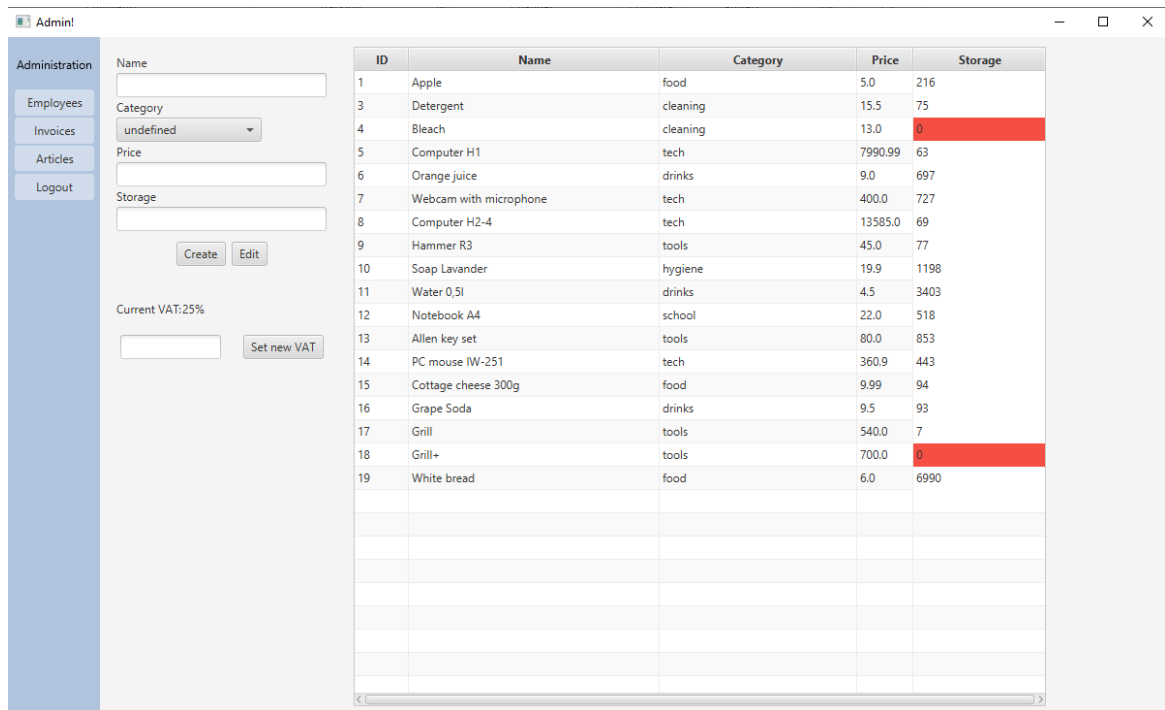
```
public ObservableList<Invoice> getInvoices() {
    ObservableList<Invoice> invoiceList = FXCollections.observableArrayList();
    Connection conn = getConnection();
    String query = "SELECT i.id, e.user, i.time, i.discount FROM invoice i " +
        "INNER JOIN employee e ON e.id = i.idEmployee";
    Statement st;
    ResultSet rs;
    try {
        st = conn.createStatement();
        rs = st.executeQuery(query);
        Invoice invoice;
        while (rs.next()) {
            invoice = new Invoice(
                rs.getInt("id"),
                rs.getString("user"),
                rs.getString("time"),
                rs.getInt("discount"));
            invoiceList.add(invoice);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return invoiceList;
}
```

Slika 4.24. Programski kod metode „*getInvoices()*“.

```
private ObservableList<ArticleInInvoice> getArticles(Invoice clickedRow) {
    ObservableList<ArticleInInvoice> articleList = FXCollections.observableArrayList();
    Connection conn = getConnection();
    String query = "SELECT a.name AS article, a.price AS price, ai.count AS amount FROM articles_in_invoice ai " +
        "INNER JOIN article a ON a.id = ai.idArticle " +
        "WHERE ai.idInvoice = '" + clickedRow.getId() + "'";
    Statement st;
    ResultSet rs;
    try {
        st = conn.createStatement();
        rs = st.executeQuery(query);
        ArticleInInvoice article;
        while (rs.next()) {
            article = new ArticleInInvoice(
                rs.getString("article"),
                rs.getInt("amount"),
                rs.getFloat("price") * rs.getInt("amount"));
            articleList.add(article);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return articleList;
}
```

Slika 4.25. Programski kod metode „*getArticles()*“.

Pritiskom tipke „Articles“ u lijevom stupcu aplikacija otvara „*adminArticlesView*“ koji sadrži tablicu svih artikala iz baze podataka i formu za unos novih artikala ili uređivanje postojećih artikala. Uz to sadrži i tipku „Set new VAT“ za promjenu poreza na dodanu vrijednost (engl. *value added tax*). U tablici se crvenom bojom označavaju ćelije stupca „Storage“ kad je količina tog artikla „0“. Forma se na isti način popunjava dvostrukim klikom na redak u tablici. Na slici 4.26 je prikazano sučelje za administraciju artikala.



Slika 4.26. Sučelje za administraciju artikala.

Na slici 4.27 je prikazana metoda „*onCreateArticle()*“ koja unosi novi artikl u bazu podataka. Na slici 4.28 je prikazana metoda „*onEditArticle()*“ koja uređuje označeni artikl u tablici.

```

@FXML
private void onCreateArticle() {
    if (isEmpty()) {
        Alert a = new Alert(Alert.AlertType.WARNING);
        a.setContentText("Fill all of text fields");
        a.show();
    } else {
        String query = "INSERT INTO article VALUES(NULL," +
            "'" + choiceCategory.getSelectionModel().getSelectedItem().toString() + "'," +
            "'" + tfName.getText() + "'," +
            "'" + tfPrice.getText() + "'," +
            "'" + tfStorage.getText() + "')";

        Connection conn = getConnection();
        Statement st;
        try {
            st = conn.createStatement();
            st.executeUpdate(query);
            updateTable();
        } catch (Exception e) {
            Alert a = new Alert(Alert.AlertType.WARNING);
            a.setContentText("Could not create a new Article");
            a.show();
            e.printStackTrace();
        }
    }
}
}

```

Slika 4.27. Programski kod metode „onCreateArticle()“.

```

@FXML
private void onEditArticle() {
    if (isEmpty()) {
        Alert a = new Alert(Alert.AlertType.WARNING);
        a.setContentText("Fill all of text fields");
        a.show();
        return;
    }
    Article article = tvArticle.getSelectionModel().getSelectedItem();

    String query = "UPDATE article SET " +
        "name='" + tfName.getText() + "'," +
        "category='" + choiceCategory.getSelectionModel().getSelectedItem().toString() + "'," +
        "price='" + tfPrice.getText() + "'," +
        "storage='" + tfStorage.getText() + "'" +
        "WHERE id='" + article.getId() + "'";

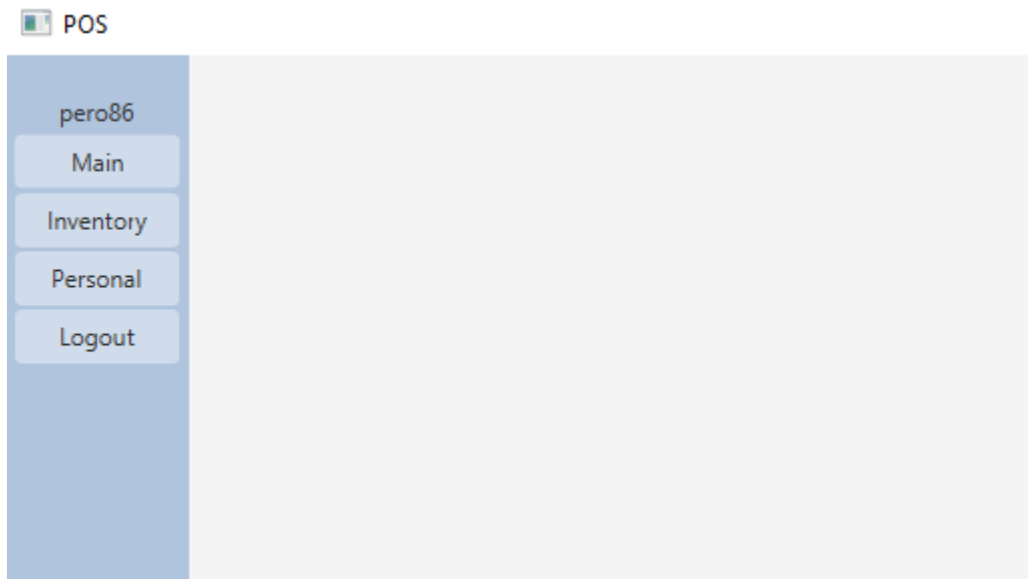
    Connection conn = getConnection();
    Statement st;
    try {
        st = conn.createStatement();
        st.executeUpdate(query);
        updateTable();
    } catch (Exception e) {
        Alert a = new Alert(Alert.AlertType.WARNING);
        a.setContentText("Could not edit Article");
        a.show();
        e.printStackTrace();
    }
}
}

```

Slika 4.28. Programski kod metode „onEditArticle()“.

4.5 Funkcionalnost zaposlenika

Prvo sučelje koje se prikaže zaposleniku nakon prijave je prikazano na slici 4.29.



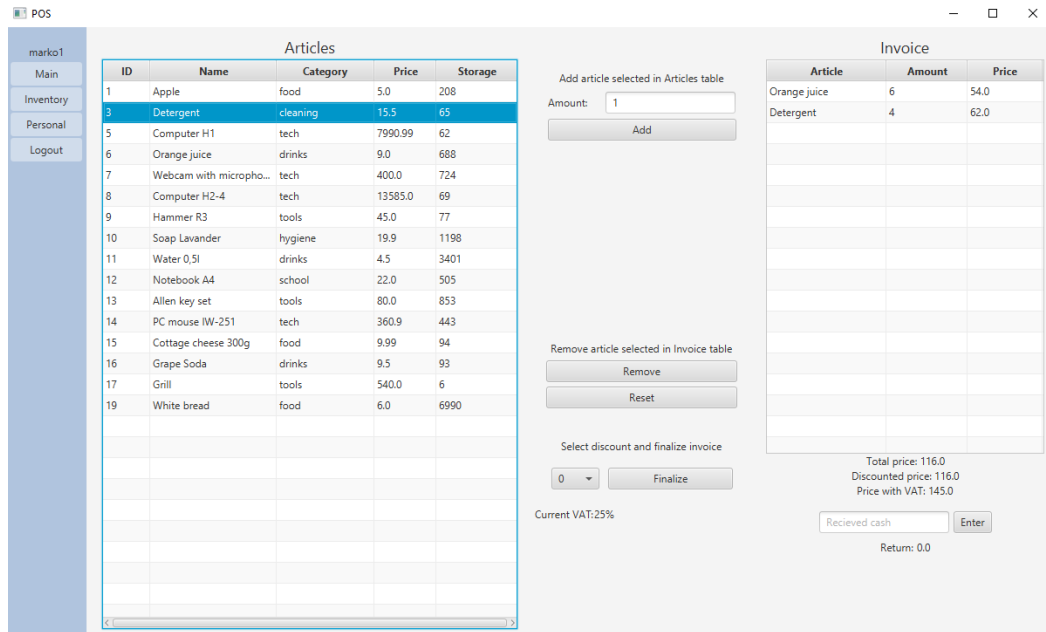
Slika 4.29. Sučelje zaposlenika nakon prijave.

Tipka „Logout“ pokreće metodu „logout()“ koja uzima vrijeme koje je spremljeno pri prijavi zaposlenika i unosi novi element u tablicu „work_hours“ za praćenje radnog vremena u bazi podataka, nakon toga vraća aplikaciju na sučelje za prijavu. Upit za unos novog elementa radnog vremena je prikazan na slici 4.30.

```
protected void logout() throws IOException {  
    String query = "INSERT INTO work_hours " +  
        "VALUES(NULL, '" + user.getId() + "', '" + timeFormat.format(startTime) + "', current_timestamp());";
```

Slika 4.30. Upit za unos radnog vremena.

Tipka „Main“ pokreće metodu „showMain()“ koja otvara novo sučelje za rad zaposlenika i „employeeMainController-u“ prosljeđuje podatke o prijavljenom zaposleniku. Na slici 4.31 je prikaz sučelja za rad zaposlenika. Zaposleniku se u tablici „Articles“ ne prikazuju artikli kojih trenutno nema u trgovini.



Slika 4.31. Sučelje za rad zaposlenika.

Tablica „Articles“ se popunjava na isti način kao i tablica u sučelju za administraciju artikala. Pritiskom tipke „Add“ se poziva metoda „addItem()“ koja dodaje iznad navedenu količinu označenog artikla u račun, koji je s desne strane ispisan u trenutnom stanju. Metoda „addItem()“ je prikazana na slici 4.32.

```

@FXML
private void addItem() {
    try {
        String article = tvArticle.getSelectionModel().getSelectedItem().getName();
        Integer amount = Integer.parseInt(tfAmount.getText());
        Float price = tvArticle.getSelectionModel().getSelectedItem().getPrice() * amount;
        Integer storage = tvArticle.getSelectionModel().getSelectedItem().getStorage();
        if (amount > storage) {
            Alert a = new Alert(Alert.AlertType.WARNING);
            a.setContentText("Can't add amount more than in storage");
            a.show();
        } else {
            articleInInvoice.add(new ArticleInInvoice(article, amount, price));
            articles.forEach((item) -> {
                if (item.getName() == article) {
                    item.setStorage(storage - amount);
                }
            });
            updateArticles();
        }
    } catch (Exception e) {
        Alert a = new Alert(Alert.AlertType.WARNING);
        a.setContentText("Error with article selection");
        a.show();
        e.printStackTrace();
    }
    tvInvoice.setItems(articleInInvoice);
    updatePrice();
    refreshInputs();
}

```

Slika 4.32. Programski kod metode „addItem()“.

Metoda „addItem()“ poziva dodatne metode za osvježavanje podataka na trenutno stanje, slične metode se koriste u skoro svim tablicama. Pri dodavanju artikala u račun se smanjuje broj

artikala na stanju i aplikacija ne dopušta stanje ispod. Pritiskom tipke „*Remove*“ poziva se metoda „*removeItem()*“ kojom se iz računa briše označeni artikl i ta količina se vrati u tablicu artikala. Tipka „*Reset*“ poziva metodu „*removeItems()*“ koja obriše sve iz računa i vrati sučelje u početno stanje. Na slici 4.33 su prikazane metode „*removeItem()*“ i „*removeItems()*“.

```
@FXML
private void removeItem() {
    articles.forEach((item) -> {
        if (item.getName() == tvInvoice.getSelectionModel().getSelectedItem().getArticle()) {
            item.setStorage(item.getStorage() + tvInvoice.getSelectionModel().getSelectedItem().getAmount());
        }
    });
    updateArticles();
    articleInInvoice.remove(tvInvoice.getSelectionModel().getSelectedItem());
    tvInvoice.setItems(articleInInvoice);
    updatePrice();
    refreshInputs();
}

@FXML
private void removeItems() {
    articleInInvoice.clear();
    tvInvoice.setItems(articleInInvoice);
    refresh();
}
```

Slika 4.33. Programski kod metoda „*removeItem()*“ i „*removeItems()*“.

Pokraj tipke „*Finalize*“ se može odabrati željeni popust na cijeli račun. Pritiskom tipke „*Finalize*“ poziva se metoda „*finalizeInvoice()*“ koja čita podatke iz tablice računa i iz tih podataka pravi upite za kreiranje elemenata u „*invoice*“ i „*articles_in_invoice*“ tablicama u bazi podataka. Nakon toga se generira tekstualna datoteka koja u imenu sadrži „*id*“ računa i unutar datoteke su ispisani svi artikli i detalji računa. Na slikama 4.34 i 4.35 je prikazan programski kod metode „*finalizeInvoice()*“. Na slici 4.36 je prikazan izgled ispisa računa u tekstualnoj datoteci.

```

for (ArticleInInvoice item : articleInInvoice) {
    for (Article itemArticle : articles) {
        if (item.getArticle() == itemArticle.getName()) {
            String query = "INSERT INTO articles_in_invoice " +
                "VALUES(' + itemArticle.getId().toString() + ',' +
                "' + invoice.getId().toString() + ',' +
                "' + item.getAmount().toString() + '");

            String query2 = "UPDATE article " +
                "SET storage=storage-" + item.getAmount().toString() + " " +
                "WHERE id=" + itemArticle.getId().toString();

            Statement st;
            try {
                st = conn.createStatement();
                st.executeUpdate(query);
                st.executeUpdate(query2);
            } catch (Exception e) {
                Alert a = new Alert(Alert.AlertType.WARNING);
                a.setContentText("Error making an Invoice");
                a.show();
                e.printStackTrace();
            }
        }
    }
}
}
}

```

Slika 4.34. Programski kod metode „finalizeInvoice()“ za unos u bazu podataka.

```

try {
    float invoicePrice = getInvoicePrice();
    float invoiceDiscount = invoice.getDiscount();
    File file = new File("d:/Projects/GuiPos/output", "invoice-" + invoice.getId().toString() + "-output.txt");
    if(!file.exists()){
        file.createNewFile();
    }
    PrintWriter printer = new PrintWriter(file);
    printer.println("Store GuiPos");
    printer.println("Adresa: 5th Street 42, Old Town");
    printer.println();
    printer.println("Invoice ID: " + invoice.getId().toString());
    printer.println("Time of issue: " + invoice.getTime());
    printer.println();
    printer.println("Price: " + getVatApplied(invoicePrice));
    if(invoiceDiscount != 0) {
        float invoicePriceAfterDiscount = invoicePrice - ((invoiceDiscount/100)*invoicePrice);
        printer.printf("Discount: %.0f\n", invoiceDiscount);
        printer.printf("Price after discount: %.2f", getVatApplied(invoicePriceAfterDiscount));
        printer.println();
    }
    printer.println("Employee: " + user.getName() + " " + user.getSurname());
    printer.println();
    printer.println("List of items in invoice:");
    printer.println();
    printer.printf("%-30s %-10s %-10s %-10s %-10s\n", "Article", "Amount", "Price/Unit", "Price/Total", "Price with VAT");
    for (ArticleInInvoice itemInInvoice : articleInInvoice) {
        printer.printf("%-30s %-10d %-10.2f %-10.2f %-10.2f \n",
            itemInInvoice.getArticle(),
            itemInInvoice.getAmount(),
            itemInInvoice.getPrice()/itemInInvoice.getAmount(),
            itemInInvoice.getPrice(),
            getVatApplied(itemInInvoice.getPrice()));
    }
    printer.close();
} catch (Exception e) {
}

```

Slika 4.35. Programski kod metode „finalizeInvoice()“ za ispis tekstualnog računa.

Store GuiPos
Adresa: 5th Street 42, Old Town

Invoice ID: 58
Time of issue: 2022-11-27 23:24:45

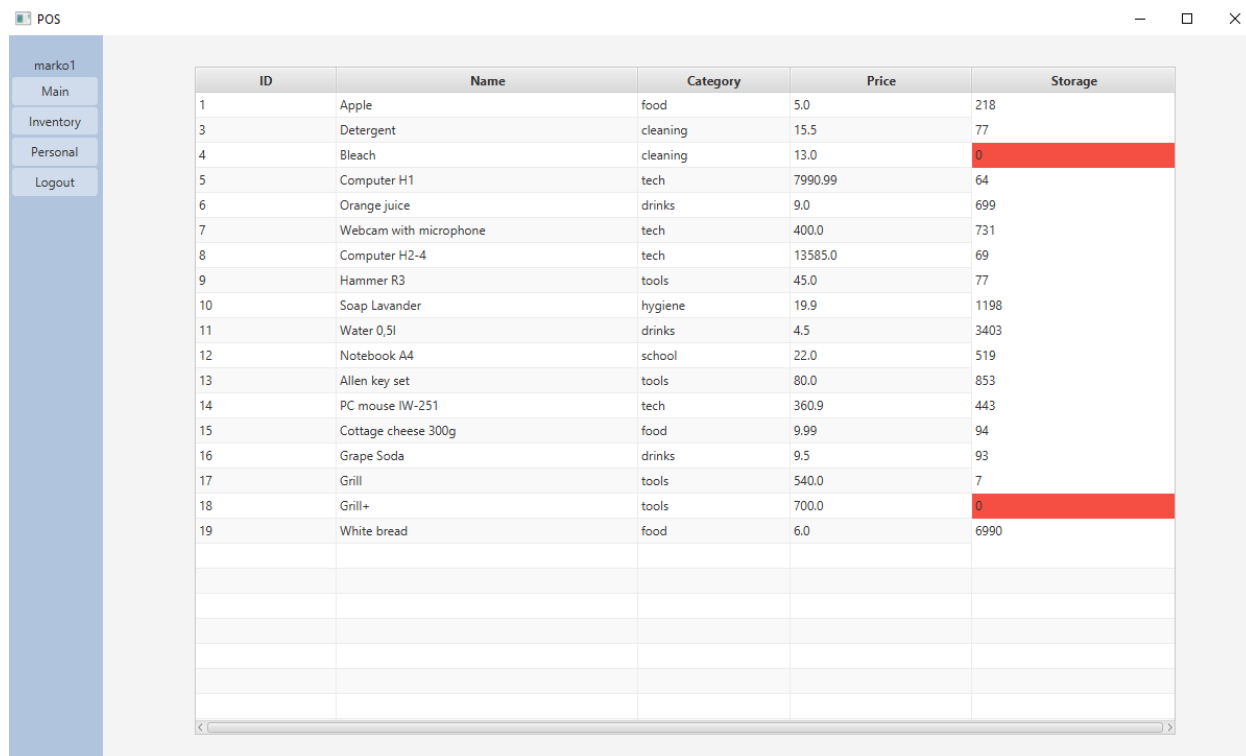
Price: 1071.88
Discount: 10
Price after discount: 964.69
Employee: Marko Markic

List of items in invoice:

Article	Amount	Price/Unit	Price/Total	Price with VAT
Detergent	1	15.50	15.50	19.38
Webcam with microphone	1	400.00	400.00	500.00
Water 0,5l	1	4.50	4.50	5.63
Notebook A4	1	22.00	22.00	27.50
Webcam with microphone	1	400.00	400.00	500.00
Detergent	1	15.50	15.50	19.38

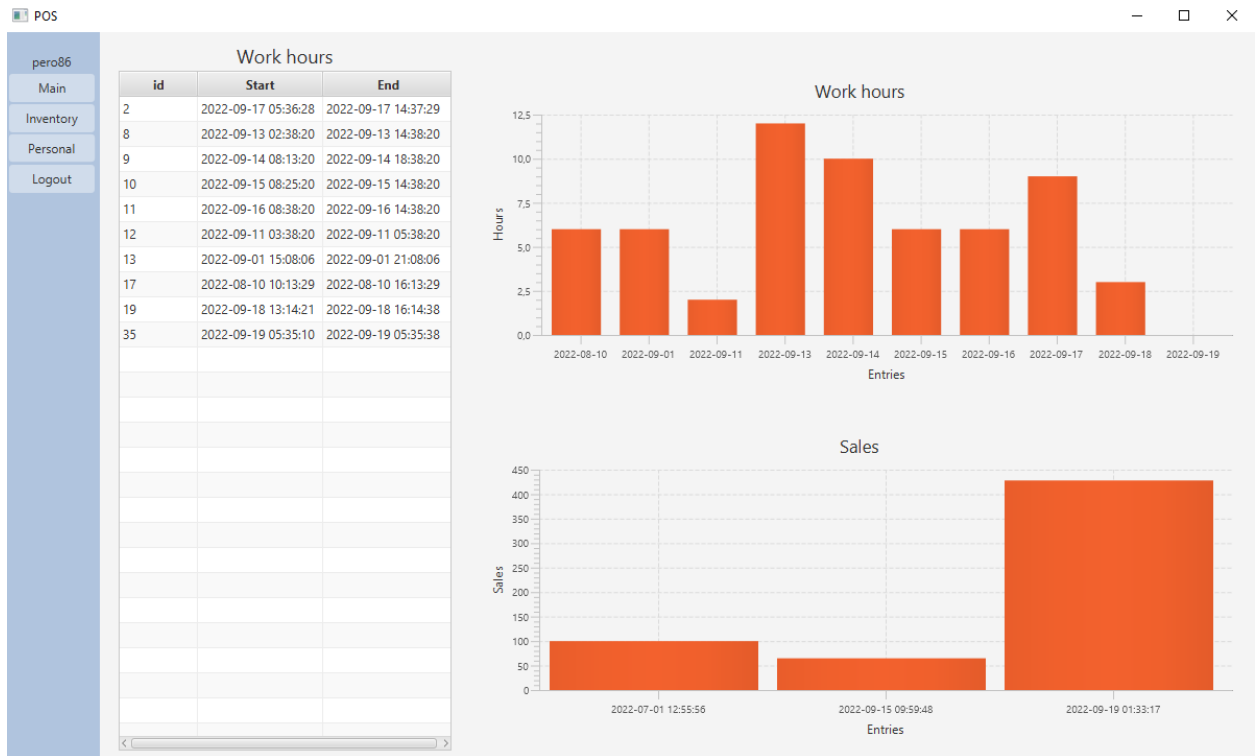
Slika 4.36. Izgled ispisanog računa.

Pritiskom tipke „Inventory“ u lijevom stupcu se otvara sučelje koje prikazuje stanje artikala u bazi podataka. Pritiskom „Personal“ se otvara sučelje koje prikazuje unose radnih sati, graf radnih sati i graf prodaja prijavljenog zaposlenika. Na slici 4.37 je prikazano sučelje „Inventory“ i na slici 4.38 je prikazano sučelje „Personal“. Oba sučelja dohvaćaju podatke istim metodama kao i u prijašnjim sučeljima za te podatke.



ID	Name	Category	Price	Storage
1	Apple	food	5.0	218
3	Detergent	cleaning	15.5	77
4	Bleach	cleaning	13.0	0
5	Computer H1	tech	7990.99	64
6	Orange juice	drinks	9.0	699
7	Webcam with microphone	tech	400.0	731
8	Computer H2-4	tech	13585.0	69
9	Hammer R3	tools	45.0	77
10	Soap Lavander	hygiene	19.9	1198
11	Water 0,5l	drinks	4.5	3403
12	Notebook A4	school	22.0	519
13	Allen key set	tools	80.0	853
14	PC mouse IW-251	tech	360.9	443
15	Cottage cheese 300g	food	9.99	94
16	Grape Soda	drinks	9.5	93
17	Grill	tools	540.0	7
18	Grill+	tools	700.0	0
19	White bread	food	6.0	6990

Slika 4.37. Prikaz sučelja „Inventory“.



Slika 4.38. Prikaz sučelja „Personal“.

5. KORIŠTENJE APLIKACIJE

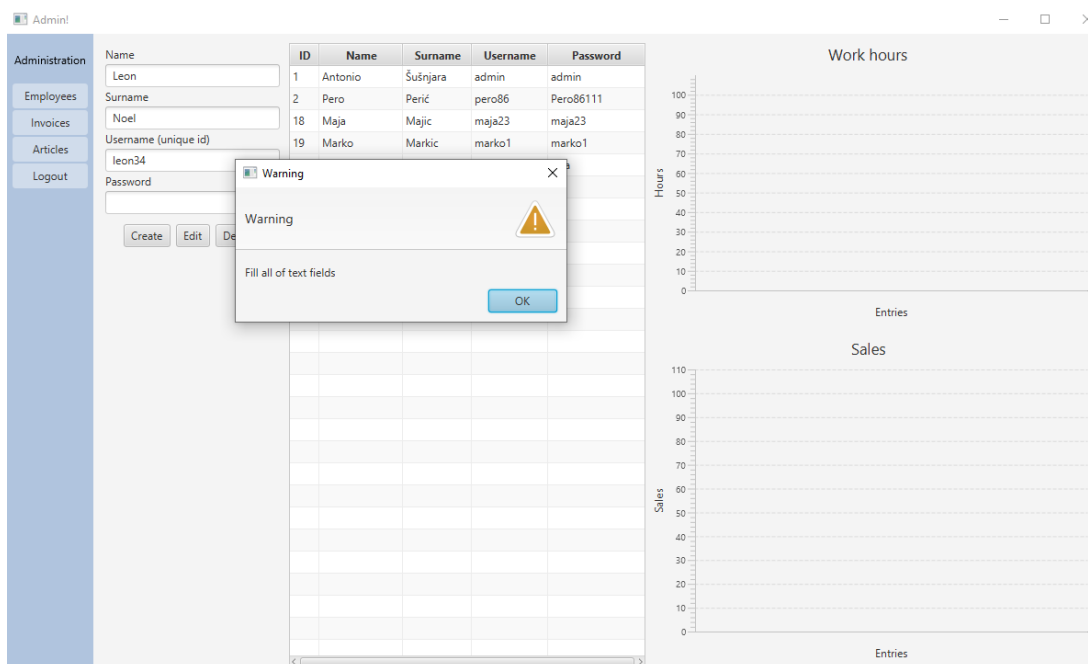
U ovom poglavlju bit će objašnjeno kako koristiti aplikaciju, koje su mogućnosti rada i koja su ograničenja.

5.1 Administrator

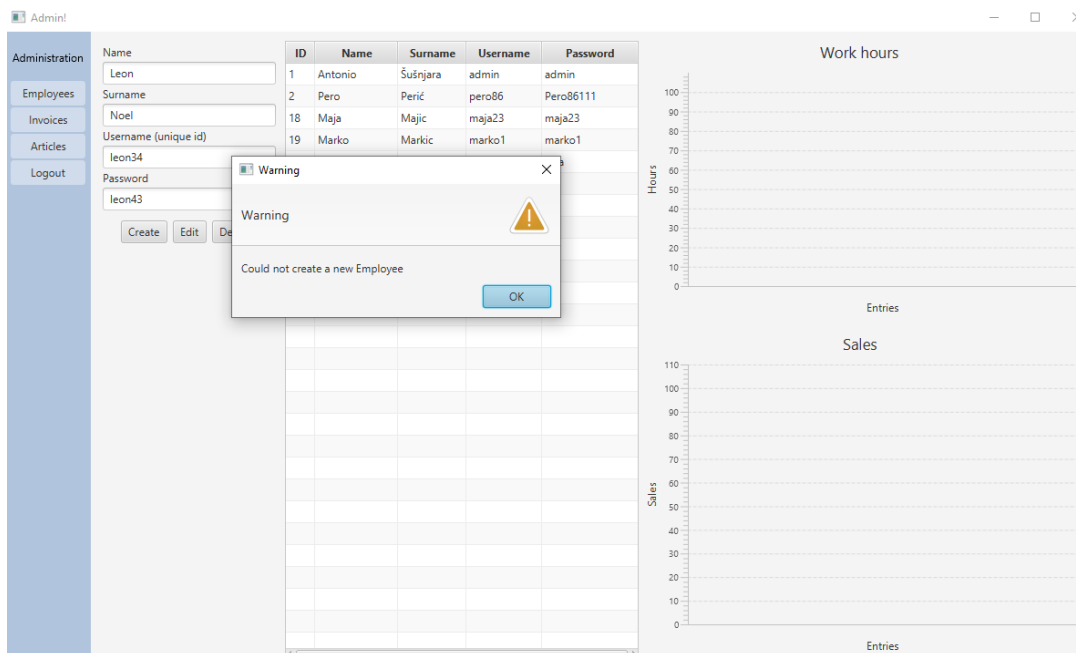
Administrator može upravljati zaposlenicima, računima i artiklima. Svaka kategorija ima svoje zasebno sučelje. Pritiskom tipke „Logout“ administrator se odjavljuje i aplikacije otvara sučelje za prijavljivanje.

5.1.1 Upravljanje zaposlenicima

Administrator može unositi nove, uređivati i brisati postojeće zaposlenike. Unošenje novog zaposlenika se izvodi tako da se u formu upišu svi podaci i zatim pritisne tipka „Create“. Ako pri kreiranju zaposlenika neki podatak nije unesen aplikacija će prikazati upozorenje na slici 5.1. U slučaju greške na strani baze podataka će se prikazati upozorenje na slici 5.2.

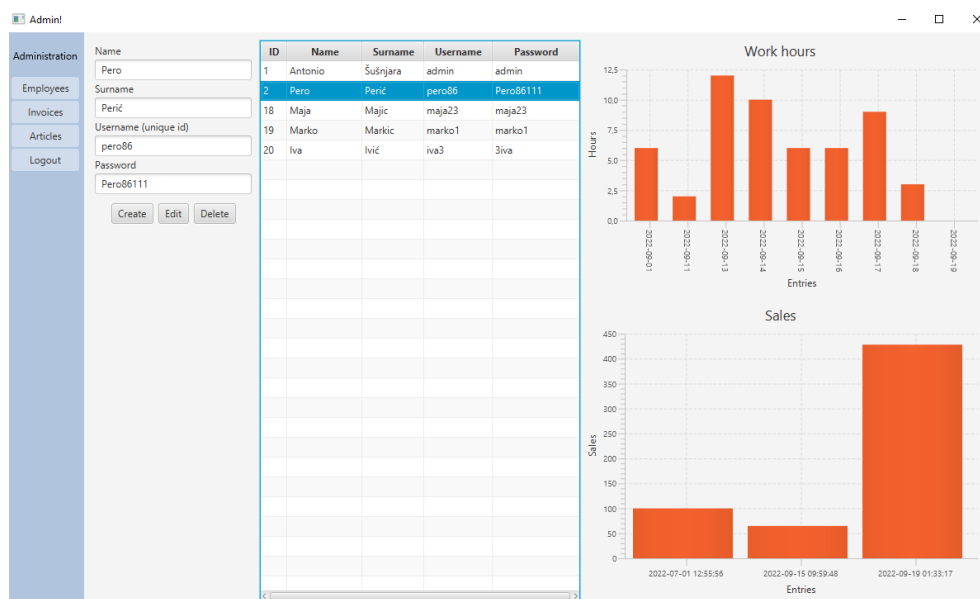


Slika 5.1. Sučelje aplikacije pri unosu zaposlenika bez svih podataka.



Slika 5.2. Sučelje aplikacije nakon greške u bazi podataka.

Dvostrukim klikom na redak željenog zaposlenika se ispuni forma i grafovi s podacima tog zaposlenika. Administrator može dvostrukim klikom ispuniti formu i zatim promijeniti željenu vrijednost pritiskom tipke „Edit“. Uređivanje zaposlenika ima istu provjeru kao i kreiranje za unesene podatke. Označavanjem zaposlenika u tablici i pritiskom tipke „Delete“ se obriše zaposlenik. Zaposlenik se ne može obrisati ako u bazi podataka još postoje računi u kojima je taj zaposlenik naveden, jer bi to dovelo do ne planiranog gubitka računa. Na slici 5.3 je prikazano sučelje s označenim zaposlenikom i popunjenim podacima.



Slika 5.3. Sučelje aplikacije popunjeno podacima odabranog zaposlenika.

5.1.2 Upravljanje računima

Administrator može vidjeti sve račune i njihov sadržaj, uz to može obrisati račun. Računi su prikazani u tablici gdje administrator može dvostrukim klikom na željeni račun popuniti tablicu sadržaja i detalje tog računa. Tipkom „Delete this invoice“ može obrisati račun koji je označen u tablici. Na slici 5.4 je prikaz sučelja za upravljanje računima popunjeno podacima računa.

The screenshot shows a web application interface for managing invoices. On the left, there is a navigation menu with options: Administration, Employees, Invoices, Articles, and Logout. The main area displays a table of invoices with columns: id, Employee, Time, and Discount / %. The row with id 45 is selected. To the right, the 'Invoice details' panel shows a table of items with columns: Article, Amount, and Price. Below this table, there is a 'Delete this invoice' button.

id	Employee	Time	Discount / %
25	maja23	2022-09-18 21:37:09	0
26	maja23	2022-09-18 21:37:11	0
27	maja23	2022-09-18 21:37:14	0
43	maja23	2022-09-19 00:02:12	0
44	maja23	2022-09-19 00:02:19	20
47	maja23	2022-09-19 01:30:47	0
28	marko1	2022-09-18 21:50:36	0
29	marko1	2022-09-18 21:50:44	0
30	marko1	2022-09-18 21:50:48	0
31	marko1	2022-09-18 21:50:52	0
32	marko1	2022-09-18 21:50:54	0
33	marko1	2022-09-18 21:50:56	0
34	marko1	2022-09-18 21:50:59	0
35	marko1	2022-09-18 21:51:01	0
36	marko1	2022-09-18 21:55:12	0
37	marko1	2022-09-18 23:46:52	0
38	marko1	2022-09-18 23:49:00	0
39	marko1	2022-09-18 23:51:04	0
40	marko1	2022-09-18 23:52:40	0
41	marko1	2022-09-18 23:56:30	0
42	marko1	2022-09-19 00:00:30	0
45	marko1	2022-09-19 00:04:05	10
49	marko1	2022-09-19 01:35:03	0
50	marko1	2022-09-19 01:36:02	0
1	pero86	2022-07-01 12:55:56	50
4	pero86	2022-09-15 09:59:48	10
48	pero86	2022-09-19 01:33:17	0

Article	Amount	Price
Detergent	1	15.5
Bleach	1	13.0
Apple	1	5.0

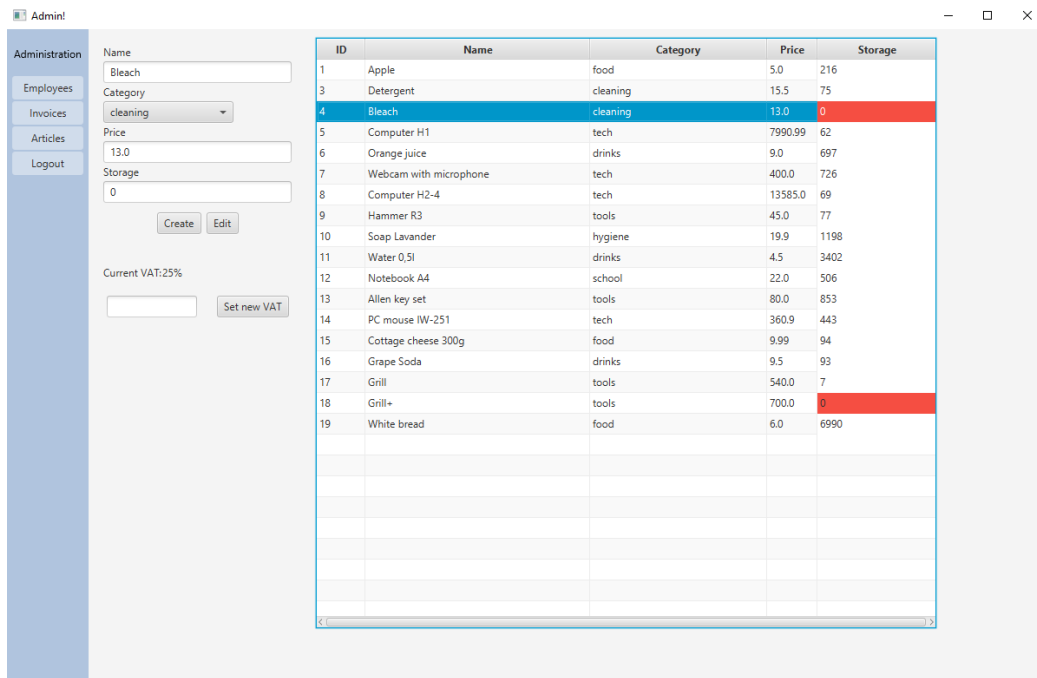
Employee: marko1
Time of issue: 2022-09-19 00:04:05
Price: 33.5
Price after discount(10%): 30.15
Unique ID: 45

Delete this invoice

Slika 5.4. Sučelje aplikacije popunjeno podacima odabranog računa.

5.1.3 Upravljanje artiklima

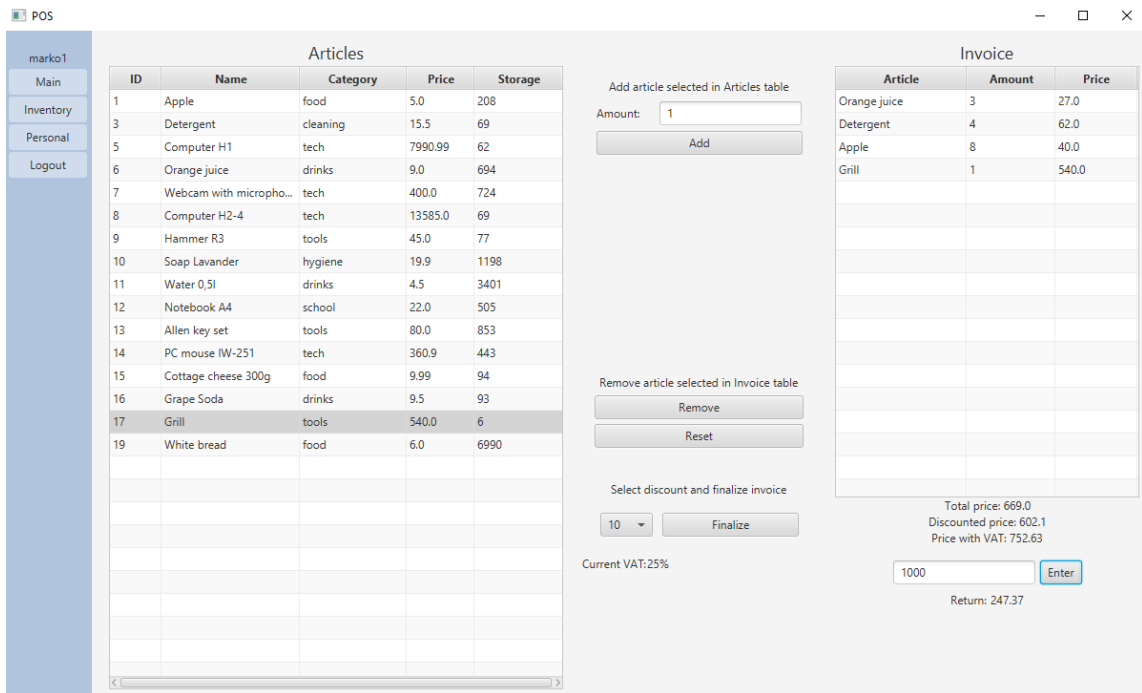
Administrator može vidjeti sve artikle i njihovo stanje u bazi podataka. Može unositi nove i uređivati postojeće artikle. Dvostrukim klikom na artikl u tablici se popuni forma gdje se onda može promijeniti podatak i pritiskom tipke „Edit“ izvršiti promjena. Unošenjem svih traženih podataka i pritiskom tipke „Create“ se mogu unijeti novi proizvodi. Artikli se ne mogu brisati iz aplikacije zbog postojećih računa u kojima su sadržani što bi dovelo do promjene izdanih računa. Pritiskom tipke „Set new VAT“ se ažurira vrijednost koja je spremljena u bazi podataka. Na slici 5.5 je prikaz sučelja za upravljanje artiklima popunjeno podacima odabranog artikla.



Slika 5.5. Sučelje aplikacije popunjeno podacima odabranog artikla.

5.2 Zaposlenik

Zaposlenik može izdavati račune, vidjeti stanje artikala i vidjeti osobne podatke rada i prodaje. Pritiskom tipke „Logout“ zaposlenik se odjavljuje i s time unosi novi element radnih sati u bazu podatak u rasponu vremena prijavljivanja i odjavljivanja. Pritiskom tipke „Inventory“ zaposlenik može vidjeti opis i stanje artikala. Pritiskom tipke „Personal“ zaposlenik može vidjeti osobne unose radnih sati i grafove radnih sati i prodaje. Pritiskom tipke „Main“ se otvara glavno sučelje za rad zaposlenika gdje može izdavati račune. Zaposlenik klikom na artikl odabere koji artikl želi dodati u račun, zatim unosom broja u „Amount“ podatak odrediti količinu za dodati u račun. Zaposlenik može klikom odabrati artikl u tablici računa i pritiskom tipke „Remove“ maknuti taj artikl iz računa. Pritiskom tipke „Reset“ se tablica račun isprazni i sučelje se vrati na prvotno stanje. Pritiskom tipke „Enter“ nakon unosa primljenog novca ispisati će se količina koju je potrebno vratiti. Pored tipke „Finalize“ zaposlenik može odabrati popust za primijeniti na račun, odabiri mogu biti 0%, 5%, 10% ili 20%. Na slici 5.6 je prikaz sučelja tokom unošenja artikala u račun. Pritiskom tipke „Finalize“ se kreira novi račun ako ima barem jedan artikl naveden u računu, nakon toga se kreira tekstualna datoteka koja u imenu sadrži unikatni „id“ računa i sve potrebne detalje. Ispisani račun sadrži cijene s porezom na dodanu vrijednost. Na slici 5.7 je prikaz računa u tekstualnoj datoteci.



Slika 5.6. Sučelje aplikacije pri kreiranju računa.

Store GuiPos

Adresa: 5th Street 42, Old Town

Invoice ID: 59

Time of issue: 2022-11-28 00:10:10

Price: 836.25

Discount: 10

Price after discount: 752.63

Employee: Marko Markic

List of items in invoice:

Article	Amount	Price/Unit	Price/Total	Price with VAT
Orange juice	3	9.00	27.00	33.75
Detergent	4	15.50	62.00	77.50
Apple	8	5.00	40.00	50.00
Grill	1	540.00	540.00	675.00

Slika 5.7. Prikaz ispisanog računa.

6. ZAKLJUČAK

Cilj ovog završnog rada bio je izrada aplikacije kao programsko rješenje za blagajnu u maloprodajnoj trgovini. Aplikacija izrađena u sklopu ovog završnog rada omogućava jednostavan rad administratoru sustava i zaposleniku trgovine, no uz manjak visoke razine sigurnosti i velikog broja značajki koje su jako bitne za sustav upravljanja trgovinom i novcem. Aplikacija je napravljena u JavaFX okviru što je omogućilo korištenje Java programskog jezika i mnoge dodatke koji su korisno za kreiranje aplikacije s grafičkim sučeljem i povezivanje s MySQL bazom podataka. MVC arhitektura je jako korisna za lakše shvaćanje povezanosti komponenti i organizaciju koda što je bila jako velika prednost za pisanje aplikacije. Pogledi (engl. *Views*) za aplikaciju su pravljani u SceneBuilderu koji je nakon teškog vremena učenja postao jako lagan i intuitivan alat za koristiti. SQL upiti i dizajniranje relacija za bazu podataka je zahtijevalo puno planiranja unaprijed što je prouzročilo otežano snalaženje po povezanosti tablica u bazi podataka.

Aplikacija omogućava jednostavan način kreiranja računa s odabranim artiklima i količinama uz spremljeni račun u tekstualnoj datoteci i u bazi podataka, uz to omogućava kreiranje profila zaposlenika i praćenje njihovog rada i prodaje.

Moguća poboljšanja aplikacije je bolji prikaz i korisnost uvida u radno vrijeme zaposlenika trgovine. Nova funkcionalnost za administratora koja prikazuje radno vrijeme u određenom vremenskom rasponu. Uz to moguće poboljšanje je korištenje dvostrukog cijelog broja (engl. *integer*) za prikaz cijene gdje prvi broj označava cjelobrojnu vrijednost cijene i drugi označava decimalni dio, zato što postoji nepreciznost pri zaokruživanju decimalnog broja s pomičnom točkom (engl. *float*).

LITERATURA

- [1] »Upute-online-aplikacija,« <https://fiskalna.hr/dokumenti/Upute-online-aplikacija.pdf> [7.7.2021.]
- [2] »Bar, Brewery Software - Loyverse POS System,« <https://loyverse.com/bar-pos> [7.7.2021.]
- [3] »eHopper POS Features,« <https://ehopper.com/product-features> [7.7.2021.]
- [4] »What is Java? Definition, Meaning & Features of Java Platforms,« <https://www.guru99.com/java-platform.html> [7.7.2021.]
- [5] »Java OOP,« https://www.w3schools.com/java/java_oop.asp [7.7.2021.]
- [6] »Introduction to Java Virtual Machine (JVM) – SUNIXI,« <https://sunixi.com/introduction-to-java-virtual-machine-jvm> [7.7.2021.]
- [7] »JavaFX vs Swing: The Key Differences | Career Karma,« <https://careerkarma.com/blog/javafx-vs-java-swing> [7.7.2021.]
- [8] »IntelliJ IDEA: The Capable & Ergonomic Java IDE by JetBrains,« <https://www.jetbrains.com/idea> [7.7.2021.]
- [9] »Scene Builder - Gluon,« <https://gluonhq.com/products/scene-builder> [7.7.2021.]
- [10] »What Is SQL And How Does it Work?,« <https://in.springboard.com/blog/what-is-sql-and-how-does-it-workn> [7.7.2021.]

SAŽETAK

Tema: Blagajna za maloprodajnu trgovinu

GUI aplikacija za korištenje računala na blagajni maloprodajne trgovine. Aplikacija omogućava rad zaposlenika za prodaju artikala. Aplikacija omogućava administratoru sustava upravljanje artiklima i praćenje rada zaposlenika kroz jednostavni prikaz podataka koji su spremljeni u bazi podataka. Aplikacija je napravljena u Java programskom jeziku kako bi se omogućilo korištenje na različitim računalima neovisno o arhitekturi.

Ključne riječi: GUI, JavaFX, MySQL, trgovina, prodaja

ABSTRACT

Title: Retail store cash register

GUI application for use on a cash register computer in a retail store. Application enables employees to sell items. The application enables the system administrator to manage items and monitor the work of employees through a simple display of data from a database. The application is made in the Java programming language to enable use on different computers regardless of architecture.

Ključne riječi: GUI, JavaFX, MySQL, sale, store