BILLS

Kompletna dokumentacija o aplikaciji za pregled, stvaranju novih i brisanje nekih pogrešno upisanih računa

Aplikacija za pregled računa, narudžbenica I slično

Contents

[1.Uvod 1](#_Toc165103042)

[2.Specifikacije, opis, tehnologije izrade i zahtjevi aplikacije 2](#_Toc165103043)

[2.1. Opis aplikacije 2](#_Toc165103044)

[2.2 Tehničke specifikacije 3](#_Toc165103045)

[2.2.1. Hardware 3](#_Toc165103046)

[2.2.2. Software 4](#_Toc165103047)

[2.3. Tehnologije izrade aplikacije 5](#_Toc165103048)

[2.4 Zahtjevi aplikacije 6](#_Toc165103049)

# 1.Uvod

Kako bismo željeli pratiti i kontrolirati našu kupovinu, neovisno dali naručujemo preko interneta ili samo kupujemo u trgovini, potrebna nam je aplikacija koja će sve to spremati i prikazivati u obliku stranice. Danas, u vrijeme razvoja umjetne inteligencije, često smo za računalom pa čak i kad nešto ne trebamo, vidimo nešto što nam se sviđa i to naručimo i kupimo. Naravno, uz to dobijemo i potvrdu da smo to kupili i pritom bismo željeli to sačuvati u digitalnom obliku kao kopiju, dok se danas postavlja i pitanje dali se uopće isplati ispisivati račune u papirnatom obliku.

Izrađivati ćemo aplikaciju za ispisivanje detalja o lokacijama kupovine, detalja o računima i artiklima. Imati ćemo i mogućnost i uređivanja računa, a o brisanju tek treba istražiti dali to uopće trebamo. Možemo jednostavno urediti bilo što što se tiče našeg računa.

Kakvi su zahtjevi? Potrebna nam je baza, koristiti ćemo maria db i xampp besplatni localhost koji će simulirati kao stvarnu bazu. Koristiti ćemo sql, a kao programske jezike html, css, php. Prvi korak je pravljenje baze i dokumentiranje iste, kao i predstavljanje zahtjeva aplikacije i opis tehnologija koje koristimo tijekom naše izrade. Uvodna poglavlja su u biti opisivanje zahtjeva i tehnologija s kojima ćemo razvijati našu aplikaciju. Sljedeća poglavlja će biti dizajn i opis uporabe aplikacije. Naravno, trebati ćemo koristiti i neke uml[[1]](#footnote-1) dijagrame, tipa use case kako bismo opisali što korisnik može s aplikacijom raditi, prikazati ćemo i shemu baze podataka, i naravno napraviti ćemo i neki dijagram klase. Pa, krenimo.

# 2.Specifikacije, opis, tehnologije izrade i zahtjevi aplikacije

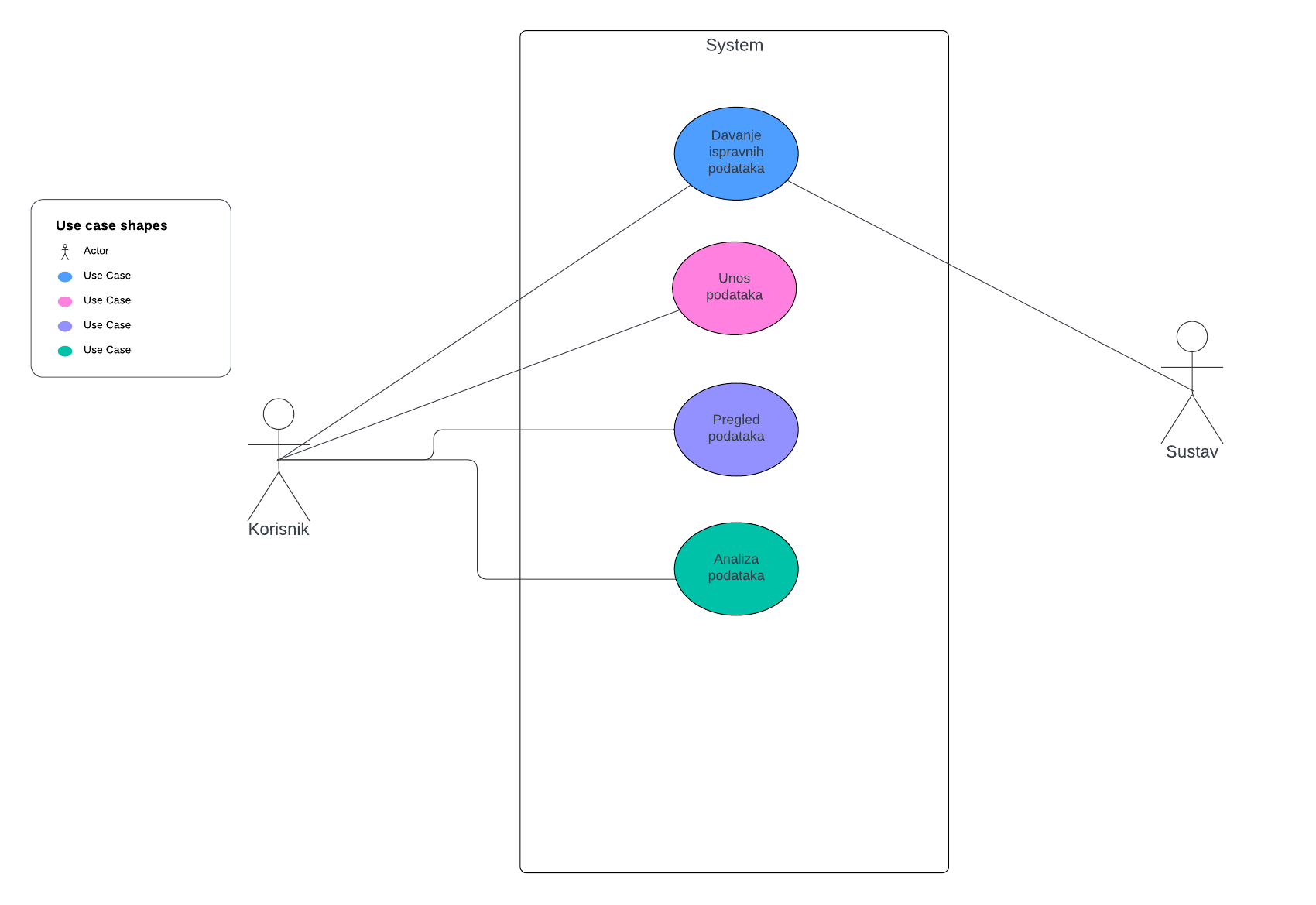
## 2.1. Opis aplikacije



Slika . Primjer računa

Slika 1 je primjer računa kojeg trebamo upisati u bazu i čiji će se podaci izvlačiti putem sql query-a. Ako analiziramo račun odozgo prema dolje možemo doći do nekakve logike kako bismo mogli napraviti kostur[[2]](#footnote-2), naravno neki podaci bi sadržavali više vrijednosti, pa ćemo koristiti i json[[3]](#footnote-3) polja, najčešće zbog toga da ne bi morali nove stupce praviti u tablici radi nekakvih vrijednosti koje su najvjerojatnije različite od jednog do drugog računa. Imamo prvo dvije tablice artikala i shopinga koje će sadržavati šifre i nazive. Imamo poslje dodatnu tablicu za detalje od trgovine, s obzirom da postoji i sjedište, imati ćemo i njihov oib i tako ćemo povezati pojedinu trgovinu sa različitim adresama, ali i račune sa nazivima te trgovine bez da ponavljamo podatke. A to je i cilj korištenja baze podataka. Imati ćemo stavke računa, koje sadrže artikle i koje količine je korisnik uzeo. Na kraju ćemo imati i dodatnu tablicu koja će sadržavati logotipove trgovine. Ovo je kratak opis što bismo zapravo trebali napraviti i na kraju ćemo preko php-a, html-a eventualno i javascripta, uzeti ćemo i tehnologije poput css-a i bootstrapa kako bismo napravili i nekakav izgled svih tih podataka.

Što bi korisnik trebao raditi sa aplikacijom? Sljedeći dijagram to najbolje opisuje. Korisnik bi trebao imati mogućnost unosa nove trgovine, novih detalja i to sve piše na računu, kao i nekakvih ostalih detalja kojih nema na računu i koje tek treba napraviti, tipa logotip. Korisnik bi trebao i imati i mogućnost uređivanja tih podataka. Što je zadaća sustava? Davanje ispravnih podataka korisniku, kako bi ih korisnik mogao izanalizirati i pregledati.



Slika Uml use case dijagram

Definirali smo što bi trebala apliakcija raditi, sad bi trebali napisati ostale detalje što ćemo koristiti kao alate za izradu te aplikacije. Mobilnu aplikaciju nećemo izrađivati.

## 2.2 Tehničke specifikacije

### 2.2.1. Hardware

* Server: XAMPP Control Panel v3.3.0
* Računalo:

Naziv uređaja: DESKTOP-K6JD4G7

Procesor: Intel(R) Core(TM) i3-1005G1 CPU @ 1.20GHz 1.19 GHz

Instalirani ram: 8,00 GB (7,79 GB iskoristivo)

ID uređaja: F17EDDF2-97D4-4685-8097-9E9192E1EE97

ID proizvoda: 00331-10000-00001-AA381

Vrsta sustava: 64-bitni operacijski sustav, procesor x64

Olovka i dodir: Za ovaj zaslon nije dostupan unos olovkom ili dodirom

### 2.2.2. Software

* Specifikacije sustava windows:

Izdanje:Windows 10 Pro

Verzija:22H2

Instalirano: ‎22.‎4.‎2021.

Međuverzija operacijskog sustava:19045.4170

Doživljaj:Windows Feature Experience Pack 1000.19054.1000.0

Apache Version Apache/2.4.56 (Win64) OpenSSL/1.1.1t PHP/8.0.28

* Apache API Version 20120211
* Server Administrator postmaster@localhost
* Hostname:Port localhost:80
* Max Requests Per Child: 0 - Keep Alive: on - Max Per Connection: 100
* Timeouts Connection: 300 - Keep-Alive: 5
* Virtual Server No
* Server Root C:/Users/Korisnik/Desktop/xmp/apache
* Loaded Modules core mod\_win32 mpm\_winnt http\_core mod\_so mod\_access\_compat mod\_actions mod\_alias mod\_allowmethods mod\_asis mod\_auth\_basic mod\_authn\_core mod\_authn\_file mod\_authz\_core mod\_authz\_groupfile mod\_authz\_host mod\_authz\_user mod\_autoindex mod\_cgi mod\_dav\_lock mod\_dir mod\_env mod\_headers mod\_include mod\_info mod\_isapi mod\_log\_config mod\_cache\_disk mod\_mime mod\_negotiation mod\_proxy mod\_proxy\_ajp mod\_rewrite mod\_setenvif mod\_socache\_shmcb mod\_ssl mod\_status mod\_version mod\_php

PHP Version 8.0.28

* System Windows NT DESKTOP-K6JD4G7 10.0 build 19045 (Windows 10) AMD64
* Build Date Feb 14 2023 12:10:00
* Build System Microsoft Windows Server 2019 Datacenter [10.0.17763]
* Compiler Visual C++ 2019
* Architecture x64

Database server

* Server: 127.0.0.1 via TCP/IP
* Server type: MariaDB
* Server connection: SSL is not being used
* Server version: 10.4.28-MariaDB - mariadb.org binary distribution
* Protocol version: 10
* User: root@localhost
* Server charset: UTF-8 Unicode (utf8mb4)

## 2.3. Tehnologije izrade aplikacije

Već smo spomenuli da ćemo koristiti određene tehnologije za izradu aplikacije. To su xampp server, html, php, javascript, bootstrap, sql. Prva verzija će biti jednostavna, koristiti će se obične tehnologije, bez frameworka. [[4]](#footnote-4) Primjeri frameworka:

* Umjetničko crtanje, glazbena kompozicija i mehanički CAD
* Aplikacije financijskog modeliranja
* Primjene modeliranja sustava Zemlje
* Sustavi za podršku odlučivanju
* Reprodukcija medija i autorska izrada
* Web okvir
* Middleware
* Cactus Framework – znanstveno računalstvo visokih performansi.
* Aplikacijski okvir – Opće GUI aplikacije.
* Okvir Enterprise Architecture
* Oracle Application Development Framework
* Laravel (PHP Framework)
* Malware , na primjer Pipedream
* Php4delphi

Što se grafičkih alata tiče, koristiti ćemo uml dijagrame, kao što su use case, već smo bili to koristili, dijagram klasa. SQL bazu i njen opis prikazati ćemo grafički, i uz svako ažuriranje ponovno ćemo napraviti sliku i pratiti proces ažuriranja ukoliko je to potrebno. Nećemo brisati staro stanje da se zna kako smo napravili točno aplikaciju. Svaki postupak ćemo opisati. Naravno, potrebni su nam i neki konceptualni alati za izradu prototipa web stranice. O tome vrijedi dodatno istraživati koji su sve to alati. Za logoipove koristimo sljedeće:

* Paint.net verzija 5.0.13 (za izrdau logotipa)
* Greenshot 1.2.10 Build 6 (64 bit) (za slikanje logotipa)

## 2.4 Zahtjevi aplikacije

Imamo primjer da smo kupili recimo četkicu za zube prije dva mjeseca, ali smo zaboravili datum i ne znamo kad trebamo kupiti novu. Preporuka je da se četkica za zube mijenja svaka dva mjeseca. Jedan od zahtjeva aplikacije je da korisnik ima tražilicu, upiše naziv proizvoda i dobije potrebne rezultate. To vrijedi uglavnom za svaki proizvod. U biti, to je u početnoj stranici, i sada samo trebamo odabrati na koji način ćemo to implementirati. Sljedeći zahtjev je unos novih web shopova i trgovina te unos novih proizvoda, njihovih detalja i transakcijskih detalja. Tu ćemo morati koristiti i kameru, primjerice želimo slikati barkod računa. Za neke podatke bismo trebali koristiti i google lens, kao izvlačenje teksta iz slike, primjerice jako dugih brojeva tipa zki i broj računa. Ažuriranje računa će negdje biti dozvoljeno, ali uglavnom korisniku neće biti moguće. Možda će biti bolje da napravimo kao neki zahtjev za izmjenu računa u obliku forme koja će se poslati administratoru. Za logotipove i recimo detalje računa i web shopova gdje ćemo upisivati podatke koji su jedinstveni za pojedine račune, trebati ćemo napraviti poseban api za to.

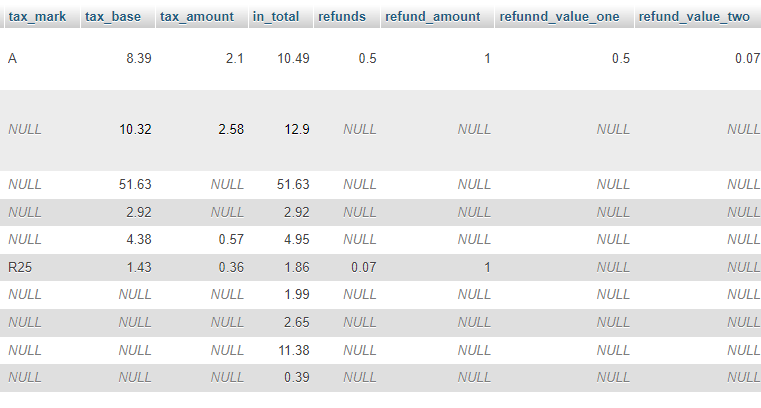
Dakle, trenutni zahtjevi obuhvaćaju ovo:

* Početna stranica sa tražiicom
* Forme za unos novih web shopova, proizvoda, transakcija itd..
* Korištenje kamere za slikanje bar kodova
* Korištenje google lensa za izvlačenje podataka sa slike
* Forma za slanje zahtjeva administratoru za ažuriranje računa
* Izrada api-a za jedinstvene dijelove računa koje ne postoje u bazi, ali polja postoje

Što se tiče logotipova slika mora biti ovakva:

* Pixel size mora biti 400x400
* Rezolucija 72,00
* Print size: width 5,56, height 5,56 8 bit

Trenutno zahtjev za login ne postoji, ali trebamo razmisliti da to implemeniramo. Trebamo razlikovati običnog korisnika od administratora.



Ovi stupci se upisuju samo ako postoji jedan proizvod na računu. Ako postoji više proizvoda, postoji druga tablica:



1. Unificirani jezik za modeliranje je jezik vizualnog modeliranja opće namjene koji je namijenjen pružanju standardnog načina vizualizacije dizajna sustava. [↑](#footnote-ref-1)
2. Naziv za našu sql bazu, koja će biti osnova za cijelu aplikaciju [↑](#footnote-ref-2)
3. Json - JSON je otvoreni standardni format datoteke i format za razmjenu podataka koji koristi tekst čitljiv ljudima za pohranu i prijenos podatkovnih objekata koji se sastoje od parova atributa i vrijednosti i nizova. [↑](#footnote-ref-3)
4. U računalnom programiranju , softverski okvir je apstrakcija u kojoj se softver , pružajući generičku funkcionalnost, može selektivno mijenjati dodatnim kodom koji je napisao korisnik, čime se pruža softver specifičan za aplikaciju. Omogućuje standardni način za izradu i implementaciju aplikacija i univerzalno je softversko okruženje za višekratnu upotrebu koje pruža posebne funkcije kao dio veće softverske platforme za olakšavanje razvoja softverskih aplikacija , proizvoda i rješenja. [↑](#footnote-ref-4)