

SISTEM ZA ZAKAZIVANJE ČASOVA I PRAĆENJE RUTA U AUTO-ŠKOLI

SYSTEM FOR APPOINTMENT BOOKING AND ROUTE TRACKING FOR DRIVING SCHOOLS

Tamara Jančić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Tema ovog rada jeste implementacija veb i mobilne aplikacije za auto-škole, čije su glavne funkcionalnosti zakazivanje termina časova i ispita, kao i praćenje ruta na časovima vožnje.

Ključne reči: veb, mobilna aplikacija, C#, .NET, React, React Native

Abstract – *The thesis deals with development of web and mobile application for driving schools, with the focus on scheduling of driving lesson and exam appointments as well as tracking routes during driving lessons.*

Keywords: web, mobile application, C#, .NET, React, React Native

1. UVOD

Veb i mobilne aplikacije predstavljaju odgovor modernog doba na svakodnevne probleme i pomoć pri izvršavanju brojnih procesa. Cilj opisanih aplikacija jeste olakšanje komunikacije i podele informacija između studenta, instruktora i ispitivača prilikom obavljanja njihovih dužnosti u auto-školi. Veb aplikacija omogućava zakazivanje časova vožnje i ispita, kad su ostvareni uslovi za to, dok zahvaljujući mobilnoj aplikaciji student u svakom momentu ima pristup istoriji ruta kojima je vozio tokom časova. U okviru rada će biti opisane specifikacije ovog sistema, tehnologije koje su korišćene, kao i moguće ideje za dalji razvoj.

2. PREGLED SLIČNIH SISTEMA

Pre detaljnijeg navođenja funkcionalnosti i implementacije opisanog sistema, biće izvršena analiza sličnih sistema. Glavni kriterijumi na osnovu kojih se izdvajaju navedeni sistemi su mogućnost zakazivanja časova, pregled napretka studenta, praćenje ruta tokom časova vožnje, obaveštenja, kao i tehnički detalji aplikacije kao što je dostupnost na različitim platformama. Imajući u vidu da tržište aplikacija za auto-škole nije veliko, kompanije retko prave aplikacije namenjene auto-školama, već im je isplativija opcija da naprave sistem šire namene, koji se može prilagođavati različitim oblastima. Jedan od takvih sistema je Teachworks [1], dizajniran prvenstveno za obrazovne ustanove. Iako postoji verzija prilagođena auto-školama,

nije integrisana mapa ni praćenje ruta studenata. Još jedan od sistema pogodan za zakazivanje termina časova i ispita, ali takođe bez praćenja ruta je Drive Scout [2]. Za razliku od dva opisana softvera, Samsara [3] je softver čija je primarna uloga praćenje ruta, međutim više je fokusiran na brigu o vozilima i sigurnost učesnika u saobraćaju. Zakazivanje termina je deo koji nedostaje ovom sistemu da bi bio odgovarajući izbor za potrebe auto-škole.

3. OPIS KORIŠĆENIH TEHNOLOGIJA

Za implementaciju serverskog dela aplikacije korišćen je programski jezik C# [4] uz pomoć .NET [5] okruženja. Skladištenje podataka je odrađeno uz pomoć MSSQL (Microsoft SQL Server) objektno-relacionog sistema za upravljanje bazama podataka. Klijentski deo veb aplikacije razvijen je koristeći React [6] biblioteku JavaScript jezika za izradu interaktivnih veb aplikacija. Mobilna aplikacija implementirana je uz pomoć React Native [7] biblioteke, takođe bazirane na JavaScript jeziku, koja omogućava razvoj mobilne aplikacije koristeći sličan kod kao kod veb aplikacija.

3.1 C#

C# je objektno-orijentisani jezik, razvijen od strane kompanije Microsoft, koji se izvršava u okviru .NET okruženja. Podržan je na različitim platformama uz pomoć .NET Core i .NET Framework radnih okvira. C# spada u familiju jezika i programeri mogu primetiti njegove sličnosti sa drugim jezicima kao što su C, C++ i Java.

3.2 .NET

Dok je C# programski jezik, .NET predstavlja okruženje koje izvršava kod napisan u C# jeziku. Osim koda napisanog u C#-u, .NET može izvršavati kod napisan i u drugim jezicima, dok je C# napravljen da bude izvršavan u .NET okruženju. S obzirom da je .NET razvijen od strane Microsoft kompanije, prvo je razvijen .NET Framework kao platforma namenjena za Windows operativni sistem. Kasnije, 2016. godine, razvijena je alternativna platforma .NET Core koja je kompatibilna i sa drugim operativnim sistemima. Trenutna verzija .NET 6 ujedinjuje sve mogućnosti ovih platformi i omogućava razvijanje više tipova aplikacija za različite platforme.

3.3. React

React je JavaScript biblioteka za razvoj klijentskih aplikacija koja obezbeđuje pregled podataka zapisanih u

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Milan Vidaković, red. prof.

HTML-u. Razvoj aplikacija se zasniva na upotrebi komponenti i smanjenju zavisnosti u kodu.

3.4 React Native

React Native je radni okvir za razvoj mobilnih aplikacija baziran na JavaScript kodu. Bitna karakteristika ovog radnog okvira je mogućnost pisanja samo jedne verzije koda koja može biti upotrebljena i na Android i na iOS operativnim sistemima. Zbog svoje sličnosti React-u, na kome je i baziran, stekao je veliku popularnost među programerima.

4. SPECIFIKACIJA

4.1 Veb aplikacija

Prvi deo zadatka obuhvata izradu veb stranice sa brojnim korisnim funkcionalnostima namenjenim članovima auto-škole. Kao najvažniji proces, izdvaja se mogućnost zakazivanja časova vožnje i termina polaganja.

Svaki korisnik može videti sve auto-škole iz sistema, pretražiti ih po imenu i prikazati njihove detaljnije informacije koje uključuju i listu instruktora te škole. Neregistrovan korisnik ima mogućnost prijave na sistem ili registracije kao student, ukoliko već nema svoj profil. Prilikom registracije, korisnik bira auto-školu i kategoriju za koju će biti obučavan. Nakon traženja zahteva za registraciju, potrebno je da verifikuje profil uz pomoć imejla koji mu je stigao na njegovu adresu i nakon toga može da se prijavi na sistem. Instruktor i ispitivač na sistem dodaje admin. Dodavanje i brisanje auto-škole je posao admina. Takođe, on može i da ažurira podatke svake auto-škole.

Svi prijavljeni korisnici mogu da pristupe stranici koja prikazuje njihov profil i na tom mestu mogu da ažuriraju svoje podatke. Admin je jedini korisnik koji ima pristup informacijama svih korisnika. On može i da obrisu svakog korisnika iz sistema. Student na svom profilu, pored standardnih informacija, može videti i informacije o rezultatu ispita teorije, broj preostalih časova vožnje i broj izlazaka na polaganje ispita vožnje. Student na svom profilu bira instruktora, iz liste instruktora koji rade u auto-školi čiji je on član i koji obučavaju za kategoriju za koju on vrši obuku. On može i promeniti svog instruktora, biranjem novog iz iste liste.

Zakazivanje časova je moguće za svakog studenta koji je položio teoriju i izabrao svog instruktora. Svaki korisnik ima pristup kalendaru koji prikazuje relevantne informacije za njega. Student vidi raspored njegovog instruktora sa zauzetim i slobodnim terminima. Zakazivanje časa u terminu kad već postoji neki drugi čas je onemogućeno, kao i zakazivanje časa ili ispita u prošlosti. Izborom slobodnog termina i tipa časa, student kreira zahtev koji instruktor može da odobri ili odbije. Tip časa vožnje može biti poligon, grad, autoput, motoput ili slobodna vožnja.

Instruktor na kalendaru vidi sve svoje časove, potencijalne i odobrene. On može da odobri ili odbije čas. Ukoliko je čas odobren, termin tog časa na kalendaru menja boju i postaje uočljiviji, radi lakšeg pregleda. Ukoliko je čas odbijen, tada je taj termin oslobođen, odbijeni čas nestaje sa prikaza. Student i instruktor mogu

u svakom momentu do početka termina, da ga ponište. Tada je taj termin oslobođen, isto kao i kod odbijanja časa. Uvedena su obaveštenja putem imejla, student je obavešten prilikom odobravanja časa o vremenu održavanja, a prilikom odbijanja, obavešteni su i student i instruktor. Nakon odrađenih obaveznih 40 časova vožnje, student ima opciju zakazivanja dodatnih časova, ako smatra da je to potrebno.

Ispitivač može da zakaže termin za polaganje ispita. Prilikom zakazivanja termina, bira auto-školu, a zatim studenta iz liste studenata te škole koji su ostvarili uslove za izlazak na ispit, a to je odrađivanje svih preostalih časova vožnje. Nakon zakazivanja ispita, student može da odobri ili odbije termin. Student može da zakazuje časove vožnje sve do termina ispita. U slučaju da student nije položio ispit, on ponovo dobija mogućnost zakazivanja časova i ispitivač može ponovo da mu zakaže ispit. I pri potvrđivanju termina ispita, student je obavešten putem imejla sa detaljnim informacijama. Odobren termin može biti uklonjen od strane studenta i ispitivača i u tom slučaju novi termin može biti zakazan.

Još jedna važna uloga ispitivača jeste unošenje rezultata ispita. Nakon što termin ispita prođe, ispitivač je dužan da unese rezultat. Student je obavešten putem imejla i prikazom detaljnih informacija o ispitu na kalendaru. Nakon što student položi ispit vožnje, on završava obuku i samim tim više nema mogućnost zakazivanja časova.

4.2 Mobilna aplikacija

Drugi deo zadatka predstavlja izradu mobilne aplikacije baziran na praćenju ruta kojima student vozi na časovima. Mobilna aplikacija je namenjena instruktorima koji beleže rute i studentima koji te rute mogu da pregledaju.

Određene funkcionalnosti veb aplikacije dostupne su i na mobilnoj, kao što su prijava na sistem, registracija, pregled profila korisnika i pregled auto-škola i njihovih detaljnijih informacija.

Ono šta izdvaja mobilnu aplikaciju je mogućnost beleženja ruta. Instruktor je zadužen da označi početak časa i tako započne beleženje putanje. On to radi tako što izabere studenta koji trenutno ima čas, iz liste svojih studenata, otvara mapu i klikće na dugme koje označava početak časa. Tokom pomeranja, na mapi se iscrtava putanja u realnom vremenu. Nakon što je čas završen, student beleži kraj časa i tada putanja ostaje zabeležena, zajedno sa datumom i vremenom tokom kog je čas trajao. Pregled odrađenih ruta je moguć od strane studenta i instruktora. Instruktor ima mogućnost pregleda ruta svih svojih učenika, dok svaki student vidi samo svoje rute.

4.3 Nefunkcionalni zahtevi

Imajući u vidu da je ovo sistem koji može sadržati osetljive podatke, kao što su lični podaci svih korisnika, potrebno je obezbediti određen nivo sigurnosti. Autentifikacija i autorizacija su neophodan deo svake aplikacije. Autentifikacija je proces validacije korisnika, a autorizacija je proces provere korisnikovih permisija u aplikaciji. .NET ima biblioteke koje omogućavaju njihovu implementaciju i korišćene su u ovom radu.

Obaveštenja korisnicima, koja se šalju na imejl adresu su implementirana uz pomoć spoljašnjeg API-ja SendGrid.

Neophodan deo mobilne aplikacije, prikazivanje i beleženje puta, omogućeno je integracijom sa Google mapama. Integracija je moguća uz pomoć biblioteke React radnog okvira, namenjenih za rad sa mapama.

Dostupnost kalendara na kom je moguće dodati termine i prikaz istih je neophodan deo opisane veb aplikacije. Ovaj zahtev je ostvaren upotrebom „*react-big-calendar*“ komponente.

Rad mobilne aplikacije na različitim operativnim sistemima (Android i iOS), omogućen je zahvaljujući React Native-u, koji podržava izvršenje istog koda na njima. Isto tako, bilo je neophodno obezbediti rad veb aplikacije na različitim operativnim sistemima i pretraživačima, što se podrazumeva za aplikacije rađene uz pomoć React-a.

5. IMPLEMENTACIJA

Kao što je već navedeno u tekstu, sistem se sastoji iz tri celine, serverske aplikacije i dve klijentske aplikacije: veb i mobilne. Serverski deo sistema je implementiran u C# programskom jeziku uz pomoć .NET radnog okvira. Za klijentske aplikacije korišćene su biblioteke JavaScript jezika, React i React Native. Radi lakše, brže i stabilnije implementacije, korišćene su i brojne biblioteke .NET-a, React-a i React Native-a, kao i neki spoljašnji API-ji i biblioteke.

5.1 Serverski deo

Serverski deo aplikacije podeljen je u više slojeva, po uzoru na MVC (Model-View-Controller) stil arhitekture sistema. Ovaj način organizacije odlikuje boljom preglednošću i smanjenjem zavisnosti u kodu. Aplikacija se sastoji iz sledećih glavnih slojeva:

- model – klase koje opisuju model podataka,
- repozitorijum – klase koje sadrže metode za pristup i komunikaciju sa bazom podataka,
- servis – vrši se manipulacija podacima, biznis logika aplikacije,
- DTO (Data Transfer Object) – objekti koji se prenose preko mreže i
- kontroler – sloj klasa za komunikaciju sa klijentom.

Obe klijentske aplikacije koriste jednu serversku aplikaciju, slanjem HTTP zahteva API-jima koji se nalaze u kontrolerima. Kontroleri prosleđuju dalje podatke servisima, koji vrše biznis logiku i pozivaju repozitorijume radi ažuriranja baze podataka i vraćaju istim redosledom nazad HTTP odgovor klijentu.

Repozitorijumi i servisi su implementirani u vidu interfejsa i klasa koje ih nasleduju i implementiraju. Na taj način obezbeđena je manja zavisnost u kodu i lakše izmene.

Pored navedenih, osnovnih slojeva, većina aplikacija po potrebi ima i neke dodatne klase i slojeve, koji, u slučaju opisanog sistema, služe za implementaciju autentifikacije i autorizacije, obradu grešaka, podešavanja baze podataka i čuvanja razne konfiguracije.

Entiteti sistema, predstavljeni klasama na serverskom delu koje preslikavaju tabele baze podataka, su User, Admin, Student, Instructor, Examiner, School, Route i Appointment. Najbitniji podaci za veb aplikaciju, termini časova i ispita, sačuvani su u bazi u Appointments tabeli sa svim neophodnim informacijama, kao što su datum i vreme početka i kraja termina, učesnicima, tipom časa, rezultatom ispita i još nekim korisnim informacijama. Svaka ruta kojom je student vozio čuvana je kao lista koordinata koje formiraju putanju, uz vreme održavanja časa i podacima o učesnicima.

5.2 Klijentski deo – veb aplikacija

Veb aplikacija je odgovorna za prikaz podataka. Ona dobija podatke sa servera i prezentuje ih korisniku, upotrebom komponenti i njihovih stanja.

Najzanimljiviji deo opisane veb aplikacije čini implementacija kalendara i njegovih funkcionalnosti. Biblioteka „*react-big-calendar*“ na jednostavan način omogućava vizuelni prikaz kalendara i svih događaja u prikazanom vremenskom periodu. Potrebno je samo da server prosledi podatke o događajima, koji moraju uključivati datum i vreme izvršenja i spomenuta biblioteka će izgenerisati ceo prikaz kalendara i obeleženih događaja. Izgled kalendara se može menjati i prilagođavati svim vrstama aplikacija.

Još jedna bitna biblioteka koja je doprinela uniformnom i lepšem izgledu veb aplikacije je MaterialUI. Kao Google-ova biblioteka korišćena za bolji izgled veb aplikacija, sadrži veliki broj komponenti predefinisano izgleda koje su spremne za upotrebu bez dodatnog menjanja.

5.3 Klijentski deo – mobilna aplikacija

Mobilne aplikacije imaju identičnu ulogu kao veb aplikacije, a to je prikaz podataka čuvanih u bazi, dobijenih putem komunikacije sa serverom. Glavna razlika pri implementaciji mobilne aplikacije u odnosu na veb jeste korišćenje komponenti za mobilne aplikacije umesto HTML koda.

Glavni deo ove aplikacije, prikaz mape, implementiran je uz pomoć „*react-native-maps*“ biblioteke. Potrebno je samo instalirati biblioteku i iskoristiti njenu komponentu za prikaz mapa. Na Android uređajima automatski se prikazuje Google mapa, dok je na Apple uređajima prikazana Apple mapa, ali moguće je podesiti mapu po izboru, kao što je odrađeno u ovoj aplikaciji izborom Google mape.

Operativni sistemi raznih uređaja, u cilju zaštite svojih korisnika, su sve strožiji povodom sigurnosti ličnih podataka. Jedan od najosetljivijih podataka jeste lokacija korisnika i zbog toga je potrebno dobiti dozvolu za podelu lokacije sa aplikacijom. Implementiran je zahtev kojim se traži korisnikova saglasnost o podeli lokacije pri prvom pristupu mapi. Nakon što korisnik pristane, mapa učitava trenutnu lokaciju i prikazuje je na mapi.

Prilikom beleženja rute, putanja se iscrtava na mapi crvenom bojom na osnovu koordinata koje klijentska aplikacija prikuplja u toku pokreta. Nakon što je putanja sačuvana, svaki sledeći prikaz putanje zahteva dobavljanje sačuvanih koordinata iz baze podataka.

Spomenuta biblioteka za rad sa mapama omogućava sve manipulacije nad mapom, kao što su iscrtavanje putanje, označavanje trenutne lokacije, približavanje mape, rotacija i još mnoge druge koje možda nisu ni korišćene u ovom sistemu, a mogu biti značajne za razvoj nekih drugih aplikacija.

6. ZAKLJUČAK

Osnovni zadatak rada je bio razvoj sistema za auto-škole i ideja je bila da sistem bude podeljen na dve aplikacije, veb i mobilnu.

Omogućavanjem zakazivanja časova i ispita u auto-školama putem veb aplikacije, studentima bi put do vozačke dozvole bio brži, a ispitivačima bi način poslovanja bio prijatniji. Beleženje ruta bi značajno pomoglo i studentima i instruktorima u praćenju napretka učenika i smišljanju novih ruta i izazova.

Aplikacija je razvijana u trenutno savremenim tehnologijama koje se proširuju i napreduju i ima dobru osnovu za dalji razvoj. Ove funkcionalnosti mogu postati osnova za razvoj većeg sistema koji pomaže u svakodnevnim procesima auto-škola. Dodavanje opcije za unos godišnjih odmora i neradnih dana instruktora, implementacija servisa za plaćanje, dodavanje mogućnosti izbora i rezervacije vozila za čas su samo neke od ideja koje bi ovu aplikaciju unapredile i napravile ozbiljan sistem namenjen auto-školama.

7. LITERATURA

- [1] <https://teachworks.com/> (pristupljeno u septembru 2023.)
- [2] <https://drivescout.com/> (pristupljeno u septembru 2023.)
- [3] <https://www.samsara.com/> (pristupljeno u septembru 2023.)
- [4] <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/> (pristupljeno u septembru 2023.)
- [5] <https://www.sam-solutions.com/blog/dot-net-vs-c/> (pristupljeno u septembru 2023.)
- [6] [https://en.wikipedia.org/wiki/React_\(software\)](https://en.wikipedia.org/wiki/React_(software)) (pristupljeno u septembru 2023.)
- [7] <https://www.netguru.com/glossary/react-native> (pristupljeno u septembru 2023.)

Kratka biografija:



Tamara Jančić rođena je u Novom Sadu 1997. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Računarstvo i automatika, odbranila je 2023.god. kontakt: tamaraa.jancic@gmail.com