Prolećni semestar, 2016/17

Projektni zadatak

Sistem za rezervaciju restorana od strane agencije

Ime: Ivan Blagojević

Broj Indeksa: 2560

Predmet: Skripting jezici

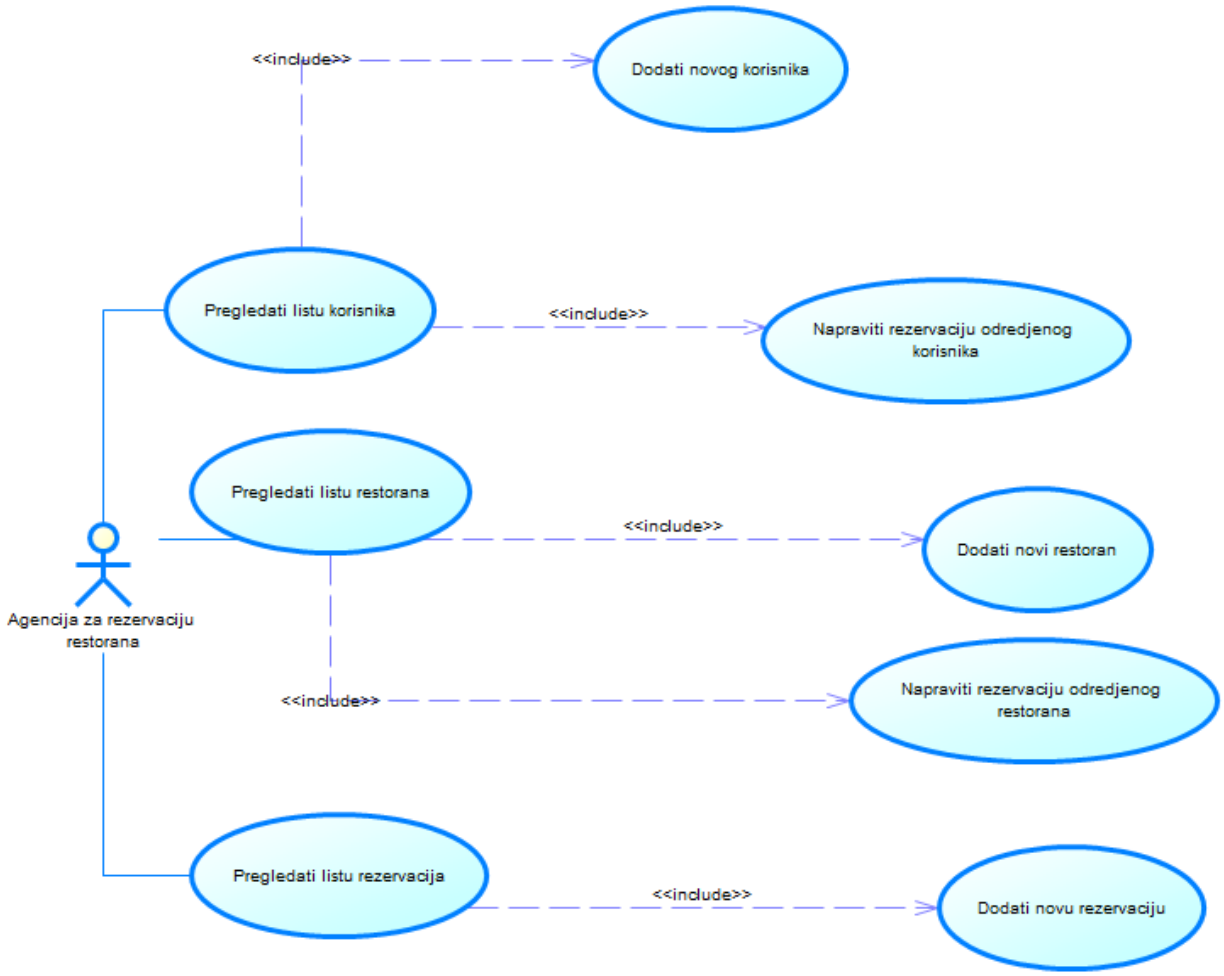
Šifra predmeta: CS324

Sadržaj:

Uvod

Ovaj dokumentacije daje samo bolje razumevanja programa za rezervaciju restorana. Program je rađen u Python programskom jeziku. Pored toga sam dokument prikazaće same slučajeve korišćenja, zajedno sa bazom podataka i prikazom pokretanja projekta kao i sam izgled istog. Program će biti rađen sa običnom sql bazom, sadržaće i jednostavan grafički prikaz koji ćemo detaljno objasniti u narednim poglavljima

Slučajevi korišćenja



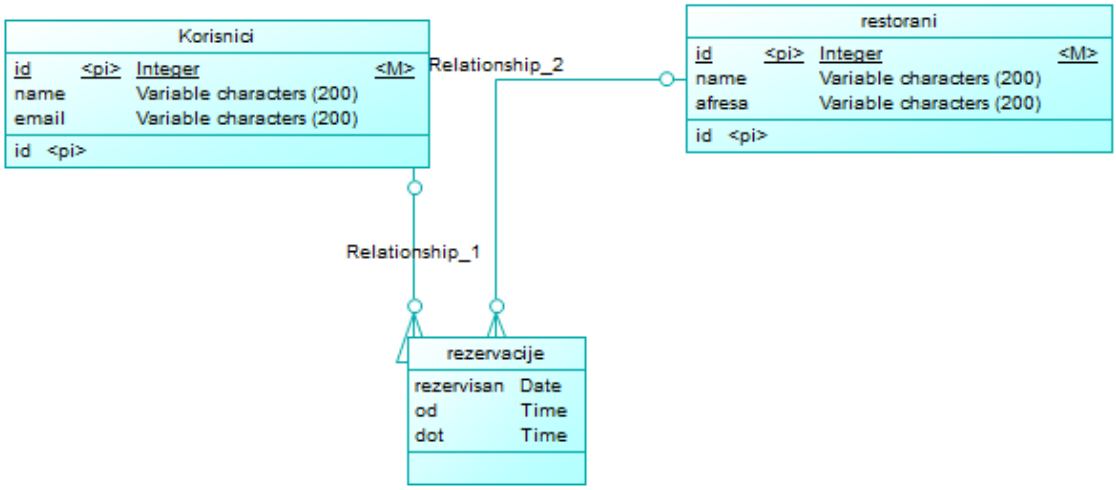
Slika 1. Use Case dijagram agencije za rezervaciju restorana

Kao što možemo videti na slici, sistem je napravljen prvenstveno za agenciju za rezervaciju restorana. Sama agencija kao prve stavke koje može da uradi jeste da pregleda sve korisnike, restorane i rezervacije. Pored samih pregleda ovih stavki, agencija može da doda novog korisnika unoseći sve

potrebne detalje i u isto vreme da odabirom korisnika napravi rezervaciju za restoran. Pored ove funkcije, korisnik može da doda novi restoran, isto kao i sa korisnikom može direktno da napravi rezervaciju birajući korisnika za taj restoran. Kao poslednja funkcija koju korisnik ima jeste da doda novu rezervaciju birajući korisnika, restoran i postavljajući sam datum rezervacije.

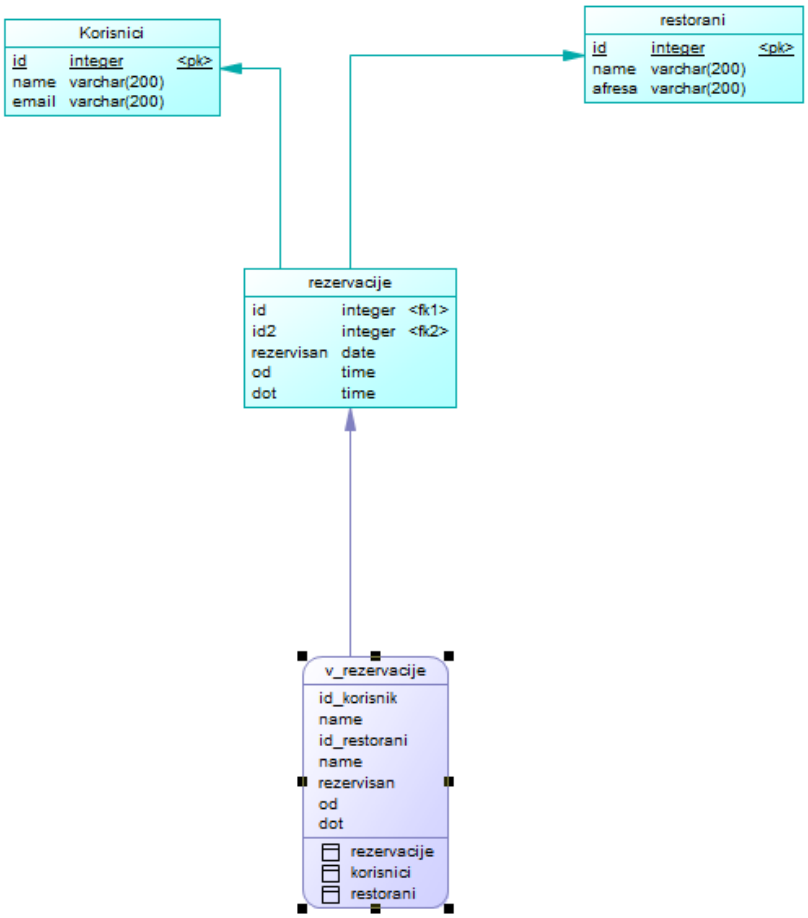
Baza podataka

Kao baza podataka koja je korišćenja u ovoj aplikaciji jeste sqlite. Ovo znači da će sama baza, što uključuje definisanje tabeli, indekse i same podatke, biti deo mašine domaćina. Ovako prost diazjn je zaključen kopletnim fajlom baze na samom početku transakcije.



Slika 2. Konceptualni dijagram baze podataka

Na samoj slici broj 2 možemo videti jednostavni prikaz baze podataka. Sama baza je jednostavno predstavljena i sadrži svega tri entiteta. Sam entitet korisnici se sastoji od imena, email adrese i samog id-ija. Pored ovog entiteta imamo i entitet restorani, koji sadrži id, ime i adresu samog restorana. Kao glavni entitet kao i sama funkcija sistema jeste rezervacija. Ona sadrži ključeve tačnije informacije od korisnika i restorana, pored toga imamo i sam datum kao i vreme rezervacije.



Slika 3. – Fizički prikaz baze podataka

Na slici broj tri imamo fizički prikaz baze podataka. Jedina razlika od samog konceptualnog dijagrama jeste što ovde imamo prikaz i glavnog pogleda. Sam taj pogled nam prikazuje sve informacije kada je određeni korisnik ili restorana rezervisan.

Opis korišćene tehnologije

Sam projek će biti rađen u 3.6.1 Python programskom jeziku. Sam Python je Slobodan softver otverenog koda koji svoj razvoj predstavlja intezivno I može se naći I skunuti besplatno sa interneta. Sam Python je razvijen za više operativnig sistema I samo rešavanja problema se rešava bez ikakvih finesa. On spade u jedan od jednostavnijih programskih jezika za korišćenja. Takođe Python spade u jedan od najpopularnijih I najkorišćenijih jezika. Pored toga sama tehnologija koja je korišćena nije ni sama tehnologija već vrsta baze. To je baza koja je implementirana u samu mašinu korisnika korišćenjem sqlite3, koja je kasnije malo detaljnije objašnjena.

- .

Struktura programa

Sam program se sastoji od sledećih funkcija:

- Napravi bazu
- Popuni bazu
- Select
- Execute
- Dodavanje u bazu
- Stranice

Napravi Bazu

```
DATABASE_FILEPATH = "rezervacije.db"

def napraviBazu():
    """ Povezivanje sa bazom
    """

    if os.path.exists(DATABASE_FILEPATH):
        os.remove(DATABASE_FILEPATH)


    db = sqlite3.connect(DATABASE_FILEPATH)
    q = db.cursor()


    sql = open("baza.sql").read()


    statements = sql.split(";")


    for statement in statements:
        q.execute(statement)

    q.close()
    db.commit()
    db.close()
```

Slika 4. Prikaz funkcije napravi bazu

Na slici broj četiri, vidimo samo način postavljanja putanje i otvaranja baze. Baza je prethodno napravljenja sa svim potrebnim informacijama za funkcionisanje samog sistema.

Popuni bazu

```
def popuniBazu():
    """Popunjavanje baze podataka
    """
    db = sqlite3.connect(DATABASE_FILEPATH)
    q = db.cursor()

    sql = "INSERT INTO korisnici(id, name, email) VALUES(?, ?, ?)"
    q.execute(sql, [1, "Marko Nikolic", "mn@gmail.com"])
    q.execute(sql, [2, "Darko Savic", "donald.duck@gmail.com"])
    q.execute(sql, [3, "Ilica Cabic", None])

    sql = "INSERT INTO restorani(id, name, adresa) VALUES(?, ?, ?)"
    q.execute(sql, [1, "Restoran Albatros", "Kneza Milosa 4"])
    q.execute(sql, [2, "Restoran Mijic", "Diljska 6"])
    q.execute(sql, [3, "Restoran Babic", "Bastenska 7"])


    sql = """
    INSERT INTO
        rezervacije
    (
        id_restorani, id_korisnik, rezervisan, od, dot
    )
    VALUES(
        ?, ?, ?, ?, ?
    )"""
    q.execute(sql, [1, 1, '2014-09-25', '09:00', '10:00'])
    q.execute(sql, [3, 1, '2015-09-25', None, None])
    q.execute(sql, [2, 3, '2014-09-22', '12:00', None])
    q.execute(sql, [1, 2, '2015-02-14', '09:30', '10:00'])

    q.close()
    db.commit()
    db.close()
```

Dell Upd
1 update

Slika 5. Prikaz funkcije za popunjavanje baze podataka

Slika broj pet, nam daje prikaz ubacivanja nekih osnovih podataka u bazu radi lakšeg testiranja našeg programa a i samom prikaza na koji način isti funkcioniše. Sam unos u bazu se vrši putem SQL jezika, koje se postavlja i kasnije samo izvršava putem komande execute.

Select

```
def select(sql_statement, params=None):
    """Citanje iz baze podataka
    """
    if params is None:
        params = []
    db = sqlite3.connect(DATABASE_FILEPATH)
    db.row_factory = sqlite3.Row
    q = db.cursor()
    try:
        q.execute(sql_statement, params)
        return q.fetchall()
    finally:
        q.close()
        db.close()
```

Slika 6. Prikaz funkcije select

Sama ova funkcija nam daje mogućnost da citamo i preuzmemo podatke iz baze podataka. Prvo se vrši konekcija sa bazom a zatim pomoću fetchall se preuzimaju samu bazu radi čitanja podataka iz baze.

Execute

```
def execute(sql_statement, params=None):
    """Unos podataka u bazi
    """
    if params is None:
        params = []
    db = sqlite3.connect(DATABASE_FILEPATH)
    q = db.cursor()
    try:
        q.execute(sql_statement, params)
        db.commit()
    finally:
        q.close()
        db.close()
```

Slika 7. Prikaz funkcije execute

Na slici broj sedam, imamo prikaz funkcije execute pomoću koje se unose podaci u samu bazu. Prvo se vrši konekcija sa bazom a zatim sam pokušaj upisa pomoću commit.

Preuzimanje podataka iz baza

```
def get_korisnik(id_korisnik):
    """Vracanje korisnika sa odredjenim id-om
    """
    for korisnik in select("SELECT * FROM korisnici WHERE id = ?", [id_korisnik]):
        return korisnik

def get_restoran(id_restorani):
    """Vracanje resotrana sa odredjenim id-om
    """
    for restoran in select("SELECT * FROM restorani WHERE id = ?", [id_restorani]):
        return restoran

def get_sveKorisnike():
    """Vracanje svih korisnika koji se nalaze u bazi
    """
    return select("SELECT * FROM korisnici")

def get_sveRestorane():
    """Vracanje svih restorana koji se nalaze u bazi
    """
    return select("SELECT * FROM restorani")

def get_rezervacije():
    """Vracanje svih rezervacija koji su u bazi
    """
    return select("SELECT * FROM v_rezervacije")

def get_rezervacijeZaOdredjenogKorisnika(id_korisnik):
    """Vracanje svih rezervacija za odredjenog korisnika
    """
    return select("SELECT * FROM v_rezervacije WHERE id_korisnik = ?", [id_korisnik])
```

Slika 8. Prikaz funkcija za preuzimanje podataka iz baze

Same get funkcije prikazuju vraćanje tačnije vraćanje podataka iz baze podataka. To se radi putem prethodno definisanje funkcije select i same naredbe select koja uzima podatke iz same baze podataka. To se radi za sve podatke koji se nalaze u samoj bazi podataka.

Dodavanje u bazu

```
def dodajOsobu(name, email):
    """Dodavanje korisnika u bazu
    """
    print("%r, %r" % (name, email))
    execute(
        "INSERT INTO korisnici(name, email) VALUES (?, ?)",
        [name, email]
    )

def dodajRestoran(name, adresa):
    """Dodavanje restorana u bazu podataka
    """
    execute(
        "INSERT INTO restorani(name, adresa) VALUES (?, ?)",
        [name, adresa]
    )

def napraviRezervaciju(id_korisnik, id_restorani, rezervisan, od=None, dot=None):
    """Dodavanje rezervacije u bazu
    """
    execute(
        """
        INSERT INTO rezervacije(id_korisnik, id_restorani, rezervisan, od, dot)
        VALUES(?, ?, ?, ?, ?)
        """,
        [id_korisnik, id_restorani, rezervisan, od, dot]
    )
```

Slika 9. Prikaz funkcija za dodavanje u bazu

Na samoj slici broj devet, možemo videti funkcije za dodavanje osobe, restorana i na kraju pravljenje same rezervacije. Samo dodavanje osobe, uključuje u sebi same podatke koje su potrebne u bazi i na kraju samo upisivanje se vrši putem execute komance povezane sa samom insert into naredbe vezane za sql. Isti način se vrši i za dodavanje restorana i pravljenje rezervacije. Sami podaci se čitaju preko samog tekst boksa koji prihvata te informacije.

Stranice

```
def page(title, content):
    """ Vracanje cele HTML strane
    """
    return """
    <html>
    <head>
    <title>Sistem za rezervaciju restorana {title}</title>
    <style>
    body {{
        background-colour : #cff;
        margin : 1em;
        padding : 1em;
        border : thin solid black;
        font-family : sans-serif;
    }}
    td {{
        padding : 0.5em;
        margin : 0.5em;
        border : thin solid blue;
    }}

    </style>
    </head>
    <body>
    <h1>{title}</h1>
    {content}
    </body>
    </html>
    """.format(title=title, content=content)

def pocetnaStrana(environ):
    """Pocetna strana koja sadrzi sve moguce strane sistema
    """
    html = """
    <ul>
        <li><a href="/korisnici">Korisnici</a></li>
        <li><a href="/restorani">Restorani</a></li>
        <li><a href="/rezervacije">Rezervacije</a></li>
    </ul>
    """
    return page("Rezervacije", html)
```

Slika 10. Prikaz same funckcije page

U ovoj funkciji imamo definisan sam HTML kod našeg projekta, koji se kasnije vraća i prikatuje ostale same komponente našeg sistema. Pored toga možemo videti i samu funkciju početne stranice, što znači da se pozivaju svi likovi ka stranicama koje sistem može ponuditi.

```
def korisnici_page(environ):
    """Stranica sa listom korisnika povezana sa svim rezervacijama tog korisnika
    """
    html = "<ul>"
    for korisnik in get_sveKorisnike():
        html += '<li><a href="/rezervacije/korisnik/{id}">{name}</a> ({email})</li>\n'.format(
            id=korisnik['id'],
            name=korisnik['name'],
            email=korisnik['email'] or "No email"
        )
    html += "</ul>"
    html += "<hr/>"
    html += """<form method="POST" action="/add-korisnik">
    <label for="name">Ime:</label> <input type="text" name="name"/>
    <label for="email">Email:</label> <input type="text" name="email"/>
    <input type="submit" name="submit" value="Dodaj korisnika"/>
    </form>"""
    return page("Korisnici", html)

def restoran_page(environ):
    """Stranica sa listom restorana, povezana sa njenim rezervacijama
    """
    html = "<ul>"
    for restoran in get_sveRestorane():
        html += '<li><a href="/rezervacije/restoran/{id}">{name}</a> ({adresa})</li>\n'.format(
            id=restoran['id'],
            name=restoran['name'],
            adresa=restoran['adresa'] or "Location unknown"
        )
    html += "</ul>"
    html += "<hr/>"
    html += """<form method="POST" action="/add-restoran">
    <label for="name">Ime:</label> <input type="text" name="name"/>
    <label for="adresa">Lokacija:</label> <input type="text" name="adresa"/>
    <input type="submit" name="submit" value="Dodaj restoran"/>
    </form>"""
    return page("Restoran", html)
```

Slika 11. Prikaz funkcija korisnici page i resotoran page

Na slici broj 11 možemo videti prvo prikaz stranice korisnici, ova stranica prikazuje same HTML kod same same stranice za korisnika. Pored toga sama stranica sadrži i prikaz funkcije dodaj. Pored ovoga slična stranica je prikazana, a to je i stranica restoran. Sama ta strenica ima slične podatke kao i korisnik, samo za razliku od korisnika ona sadrži ime, lokaciju i samu opciju dodaj restoran.

```
def all_rezervacije_page(environ):
    """Stranica sa svim rezervacijama
    """
    html = "<table>"
    html += "<tr><td>Ime korisnikan</td><td>Restoran</td><td>Datum</td><td>Vreme</td></tr>"
    for rezervacija in get_rezervacije():
        html += "<tr><td>{korisnik_ime}</td><td>{restoran_ime}</td><td>{rezervisan}</td><td>{od} - {dot}</td>

           restoran_ime=rezervacija['restoran_ime'],
           korisnik_ime=rezervacija['korisnik_ime'],
            rezervisan=rezervacija['rezervisan'],
            od=rezervacija['od'] or "",
            dot=rezervacija['dot'] or ""
        )
    html += "</table>"

    html += "<hr/>"
    html += '<form method="POST" action="/add-rezervacija">'

    html += '<label for="id_korisnik">Korisnik:</label> <select name="id_korisnik">'
    for korisnik in get_sveKorisnike():
        html += '<option value="{id}">{name}</option>'.format(**korisnik)
    html += '</select>'

    html += ' | '

    html += '<label for="id_restorani">Restoran:</label> <select name="id_restorani">'
    for restoran in get_sveRestorane():
        html += '<option value="{id}">{name}</option>'.format(**restoran)
    html += '</select>'

    html += ' | '
    html += '<label for="rezervisan">Datum</label> <input type="text" name="rezervisan" value="{today}"/
    html += ' <label for="od">Izmedju</label> <input type="text" name="od" />'
    html += ' <label for="dot">i</label> <input type="text" name="dot" />'
    html += '<input type="submit" name="submit" value="Dodaj rezervaciju"/></form>'

    return page("Sve rezervacije", html)
```

Slika 12. Prikaz stranice sa svim rezervacijama

Na slici broj 12 možemo videti sam kod za prikaz stranice sa svim rezervacijama i definisanje lejbela i imena koja će nam kasnije dosta pomoći.

```
def rezervacije_korisnik_page(environ):

    id_korisnik = int(shift_path_info(environ))
    korisnik = get_korisnik(id_korisnik)
    html = "<table>"
    html += "<tr><td>Restoran</td><td>Datum</td><td>Vreme</td></tr>"
    for rezervacija in get_rezervacijeZaOdredjenogKorisnika(id_korisnik):
        html += "<tr><td>{korisnik_ime}</td><td>{rezervisan}</td><td>{od} - {dot}</td></tr>".format(
            korisnik_ime=rezervacija['korisnik_ime'],
            rezervisan=rezervacija['rezervisan'],
            od=rezervacija['od'] or "",
            dot=rezervacija['dot'] or ""
        )
    html += "</table>"
    html += "<hr/>"
    html += '<form method="POST" action="/add-rezervacija">'
    html += '<input type="hidden" name="id_korisnik" value="{id_korisnik}"/>'.format(id_korisnik=id_korisnik)
    html += '<label for="id_restorani">Restorani:</label> <select name="id_restorani">'
    for restoran in get_sveRestorane():
        html += '<option value="{id}">{name}</option>'.format(**restoran)
    html += '</select>'
    html += ' | '
    html += '<label for="rezervisan">Datum</label> <input type="text" name="rezervisan" value="{today}"/
    html += ' <label for="od">izmedju</label> <input type="text" name="od" />'
    html += ' <label for="dot">i</label> <input type="text" name="dot" />'
    html += '<input type="submit" name="submit" value="Dodaj rezervaciju"/></form>'
    return page("Rezervacija za %s" % korisnik['name'], html)
```

Slika 13.Prikaz funkcije rezervacije korisnika

Na samoj slici broj 13 možemo videti prikaz HTML koda i samog prikaza svih rezervacija od stane korisnika. Ova stranica je jako bitna, pošto prilikom odabira određenog korisnika doći će do prikaza svih

rezervacija tog korisnika. Takođe pored toga moći će se i ujedno i dodati sama rezervacija nakon odabira korisnika.

```
def rezervacije_restorani_page(environ):

    id_restorani = int(shift_path_info(environ))
    restoran = get_restoran(id_restorani)
    html = "<table>"
    html += "<tr><td>Korisnik</td><td>Datum</td><td>Vreme</td></tr>"
    for rezervacija in get_rezervacijeZaOdredjenuSobu(id_restorani):
        html += "<tr><td>{restoran_ime}</td><td>{rezervisan}</td><td>{od} - {dot}</td></tr>".format(
            restoran_ime=rezervacija['restoran_ime'],
            rezervisan=rezervacija['rezervisan'],
            od=rezervacija['od'] or "",
            dot=rezervacija['dot'] or ""
        )
    html += "</table>"
    html += "<hr/>"
    html += '<form method="POST" action="/add-rezervacija">'
    html += '<input type="hidden" name="id_restorani" value="{id_restorani}"/>'.format(id_restorani=id_restor
    html += '<label for="id_korisnik">Korisnik:</label> <select name="id_korisnik">'
    for korisnik in get_sveKorisnike():
        html += '<option value="{id}">{name}</option>'.format(**korisnik)
    html += '</select>'
    html += ' | '
    html += '<label for="rezervisan">Rezervisan</label> <input type="text" name="rezervisan" value="{tod
    html += ' <label for="od">izmedju</label> <input type="text" name="od" />'
    html += ' <label for="dot">i</label> <input type="text" name="dot" />'
    html += '<input type="submit" name="submit" value="Dodaj rezervaciju"/></form>'
    return page("Rezervacija za  %s" % restoran['name'], html)
```

Slika 14. Prikaz stranice rezervacije restorana

Sama slika prikazuje HTML kod implementiran u Python-u i služi za prikaz svih rezervacija za odabrani restoran. Prikazuje sve informacije o restoranu i rezervacijama za sam taj restoran.

```
restoran_page(environ):
"""Stranica sa listom restorana, povezana sa njenim rezervacijama
"""
html = "<ul>"
for restoran in get_sveRestorane():
    html += '<li><a href="/rezervacije/restoran/{id}">{name}</a> ({adresa})</li>\n'.format(
        id=restoran['id'],
        name=restoran['name'],
        adresa=restoran['adresa'] or "Location unknown"
    )
html += "</ul>"
html += "<hr/>"
html += """<form method="POST" action="/add-restoran">
<label for="name">Ime:</label> <input type="text" name="name"/>
<label for="adresa">Lokacija:</label> <input type="text" name="adresa"/>
<input type="submit" name="submit" value="Dodaj restoran"/>
</form>"""
return page("Restoran", html)
```

Slika 15. Prikaz stranice sa svim restoranima

Sama slika 15 prikazuje sve informacije o postojećim restoranima isto kao i sam prikaz opcije za dodavanje restorana.

Dodatne funkcije

Pored samih ovih funkcija imamo i dodatne funkcije. Prva i jako bitna funckija jeste server za samu običnu stanicu, koja je bazirana na URL zahtevima korinika, restorana i rezervacija.

```
def webapp(environ, start_response):

    setup_testing_defaults(environ)

    status = '200 OK'
    headers = [('Content-type', 'text/html; charset=utf-8')]

    param1 = shift_path_info(environ)
    if param1 == "":
        data = pocetnaStrana(environ)
    elif param1 == "korisnici":
        data = korisnici_page(environ)
    elif param1 == "restorani":
        data = restoran_page(environ)
    elif param1 == "rezervacije":
        data = rezervacije_page(environ)
    elif param1 == "add-korisnik":
        add_korisnik(environ)
        status = "301 Redirect"
        headers.append(("Location", "/korisnici"))
        data = ""
    elif param1 == "add-restoran":
        add_restoran(environ)
        status = "301 Redirect"
        headers.append(("Location", "/restorani"))
        data = ""
    elif param1 == "add-rezervacija":
        add_rezervacija(environ)
        status = "301 Redirect"
        headers.append(("Location", environ.get("HTTP_REFERER", "/rezervacije")))
        data = ""
    else:
        status = '404 Not Found'
        data = "Not Found: %s" % param1

    start_response(status, headers)
    return [data.encode("utf-8")]
```

Slika 16. Prikaz webapp funckcije

Na početku se vrši sama predpostavka da se ide na server sa validnom HTML stranicom. Nakon toga se preuzima prvi segment na putanji i prosleđuju se ostale. Na primer ukoliko se traži korisnici/l/rezervacije, param1 će biti korisnici a ostatak putanje će biti l/rezervacije.

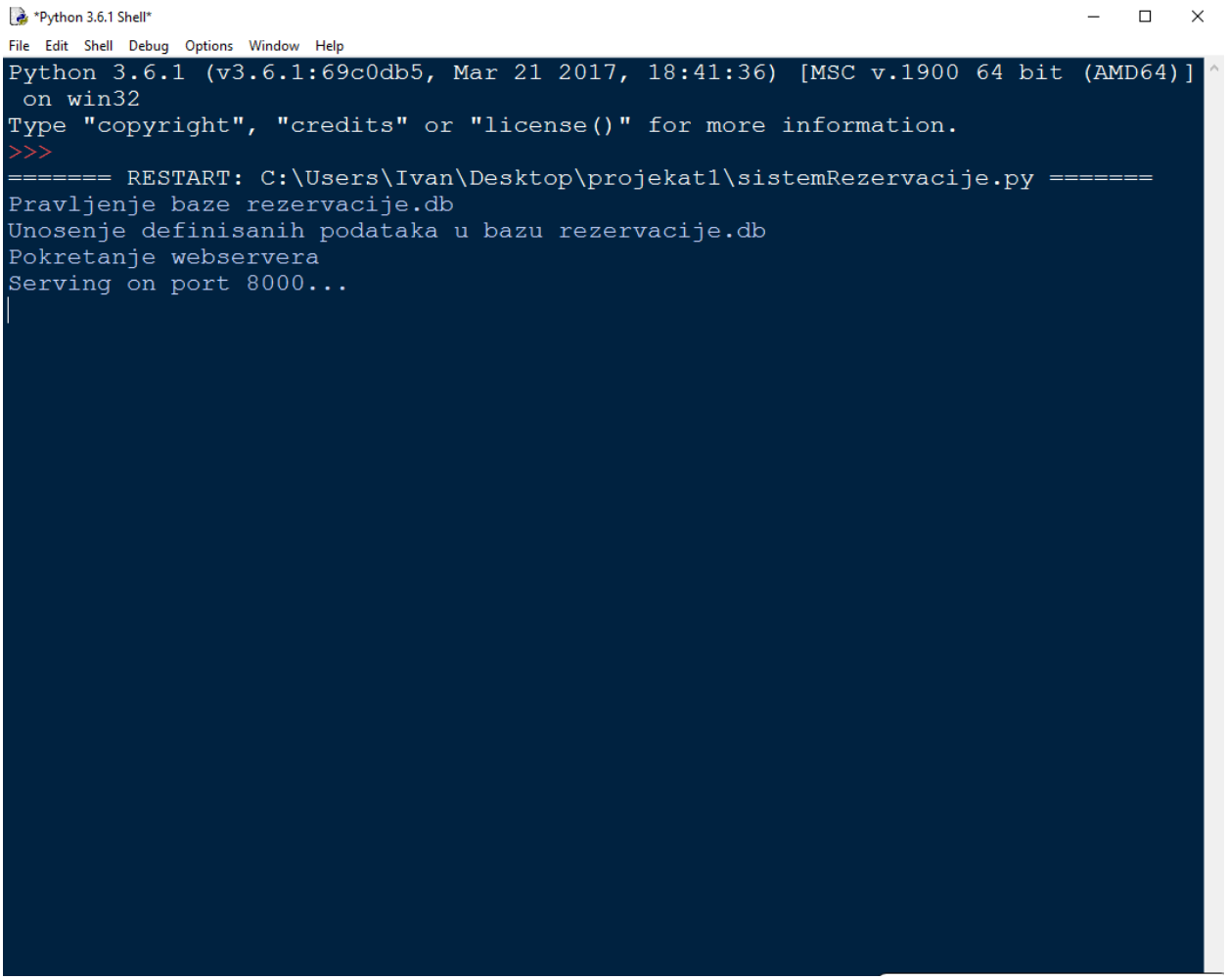
```
def run_website():
    httpd = make_server('', 8000, webapp)
    print("Serving on port 8000...")
    httpd.serve_forever()

if __name__ == '__main__':
    print("Pravljenje baze %s" % DATABASE_FILEPATH)
    napraviBazu()
    print("Unosenje definisanih podataka u bazu %s" % DATABASE_FILEPATH)
    popuniBazu()
    print("Pokretanje webservera")
    run_website()
    print("Finished")
```

Slika 17. Pokretanje website-a

Na samoj slici 17 vidimo samo povezivanje sa serverom i imamo prikazanu samu main funkciju koja poziva funkcije pravljenja i popunjavanja baze i podizanje tačnije pokretanje website-a.

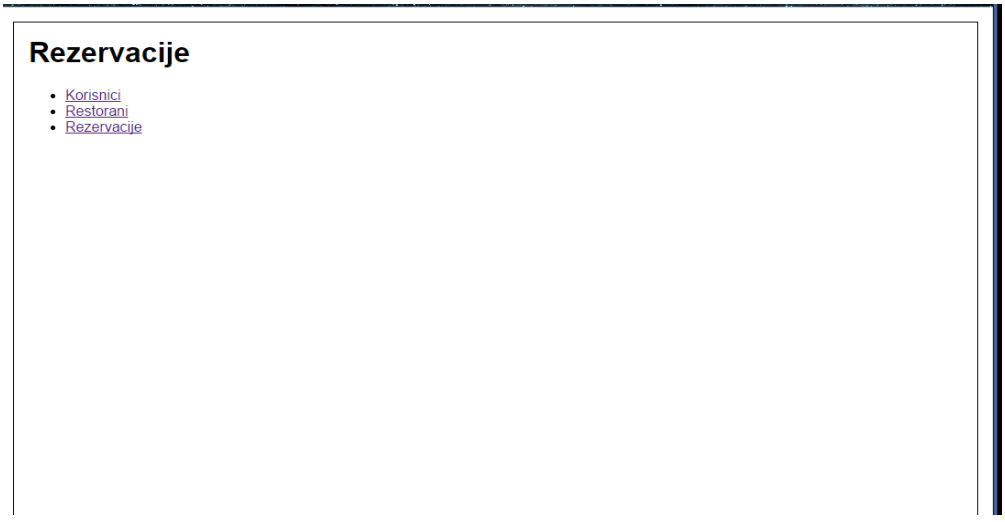
Prikaz pokretanja i rada sistema

Samo pokretanje se radi putem Pzthon 3.6.1 Shell-a koji pokreće samu aplikaciju na localhost:8000 serveru.



Slika 18. Prikaz podizanja servera i samog sistema

Početna stranica

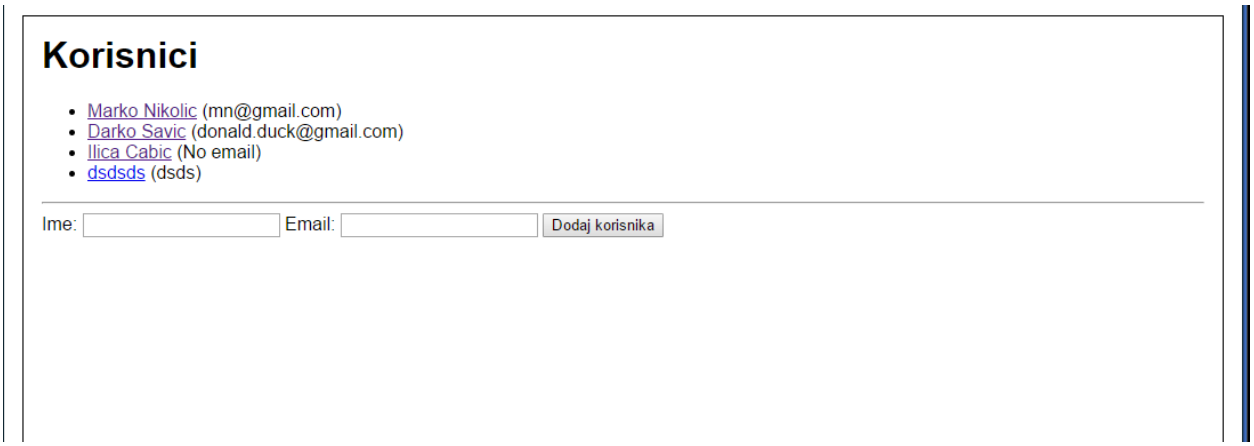


Slika 19. Prikaz početne stranice sistema

Stranica korisnici

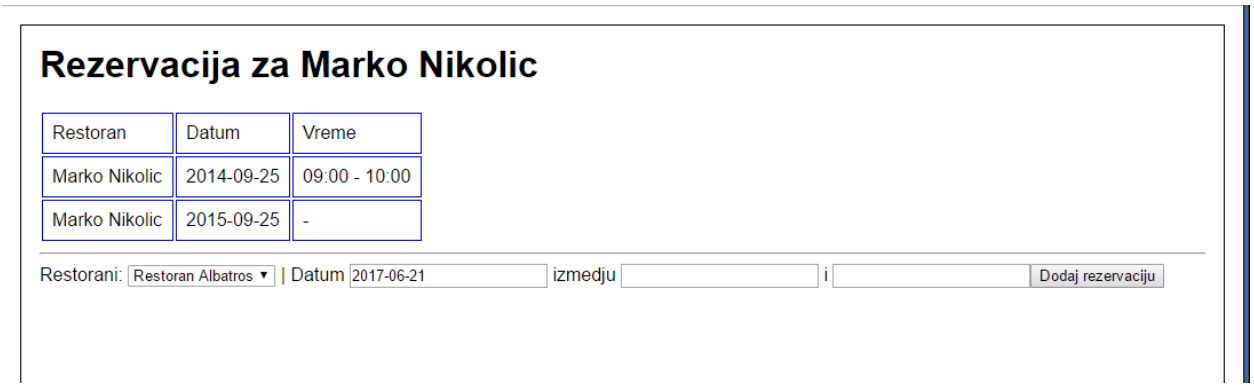
Slika 20. Prikaz stanice sa listom korisnika

Dodavanje novog korisnika



Slika 21. Prikaz stanice sa novim korisnikom

Prikaz rezervacije odabranog korisnika

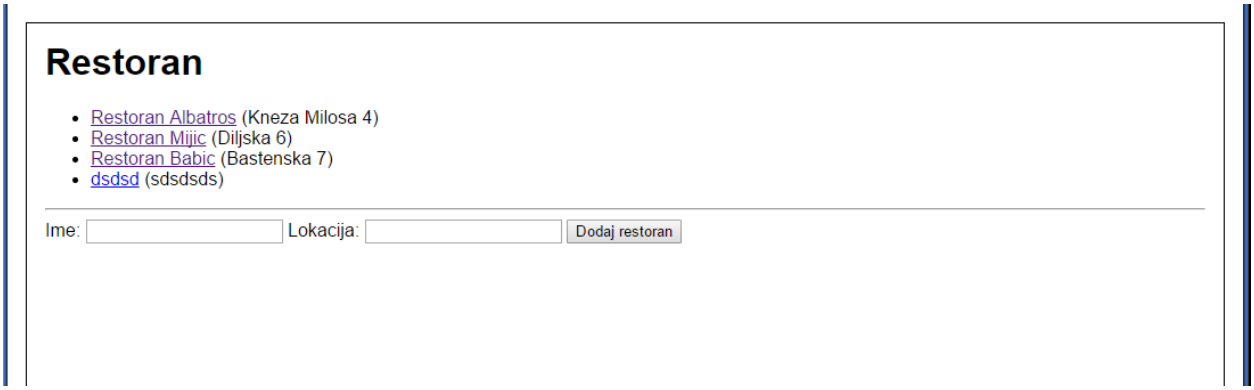


Slika 22. Prikaz rezervacije odabranog korisnika i mogućnost dodavanja nove rezervacije za odabranog korsinika

Stranica sa spiskom restorana

Slika 23. Prikaz liste svih raspoloživih restorana

Dodavanje novog restorana

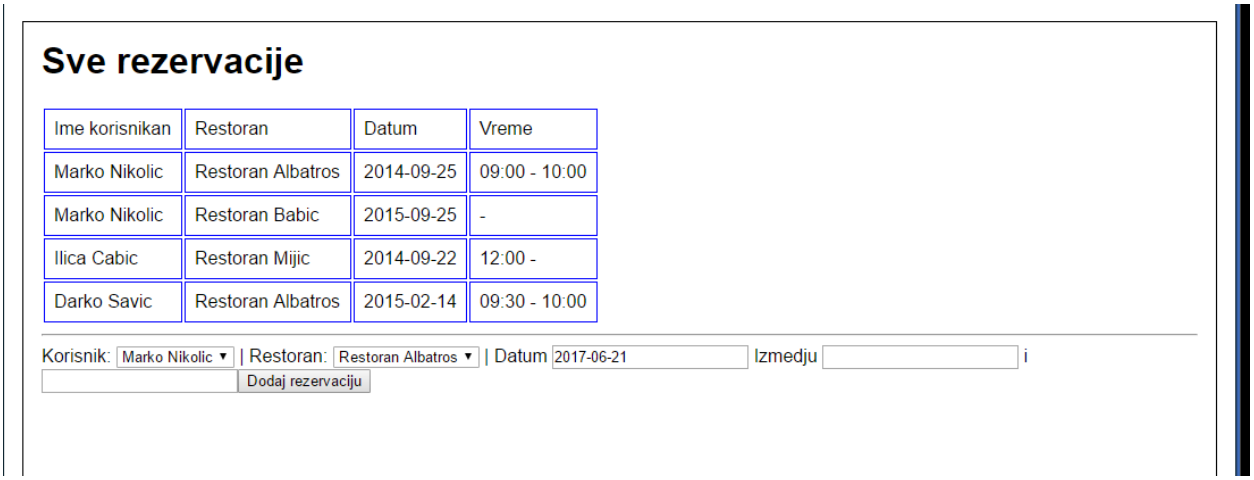


Slika 24. Prikaz stranice sa novim dodatim restoranom

Stanica rezervacije određenog restorana

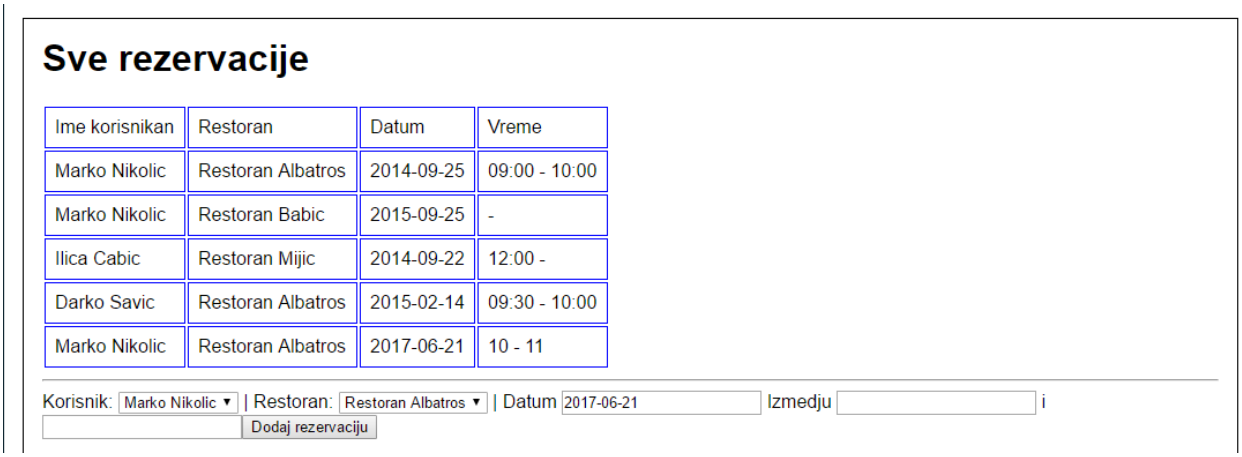
Slika 25. Prikaz rezervacije odabranog restorana i mogućnost dodavanja nove rezervacije za odabranog restorana

Stranica za prikaz svih rezervacija



Slika 26. Prikaz svih rezervacija

Dodavanje nove rezervacije



Slika 27. Prikaz stranice nakon dodate nove rezervacije

Zaključak

Sama izgradnja ove aplikacije je rađena kao sama osnova, pošto je sam nivo trenutnog poznovanja Python jezika na početnom nivou. Sam projekat je zamišljen ovako kao prva i osnovna faza, u sledećim i narednim fazama sam projekat bi dobio više mogućnosti. Moguće poboljšanje jeste em u pogledu same grafike em u samom pogledu mogućih korisnika. Sistem može biti podeljen na više uloga i na taj način bi sistem dobio veći nivo, dok je trenutno na početnom nivou.

Reference

1. https://www.python.org
2. https://www.tutorialspoint.com/python/
3. http://packtpub.ooops.me/2017/04/04/Expert_Python_Programming.pdf