**Univerzitet „Džemal Bijedić“ Datum:** 07.9.2018.

**Fakultet informacijskih tehnologija**

**Mostar**

**Integralni ispit**

iz predmeta Analiza i dizajn softvera

1. Potrebno je projektovati softversko rješenje za podršku poslovanja kompanije koja se bavi održavanjem i naplatom parking mjesta. Aplikaciju će koristiti tri tipa korisnika: službenici centrale, kontrolori i krajnji korisnici usluga kompanije.

Službenici centrale određuju tarife pojedinih zona kao i pripadnost ulica pojedinim zonama. Oni također imaju mogućnost evidencije podataka o kontrolorima, zatim mogućnost administracije vozila krajnjih korisnika, administracije zona, pregled kazni, pregled dostupnih automata i detalja upotrebe parking usluga za sve korisnike.

Sistem je potrebno povezati sa sistemom MUP-a za evidenciju kazni, te svaku kaznu koja je evidentirana u sistemu proslijediti sistemu MUP-a. MUP prosljeđuje podatak o svakoj kazni za parking koja je plaćena.

Kontrolori predstavljaju zaposlenike zadužene za bilježenje parkiranih vozila i manuelnu provjeru naplate parking usluga za pretplaćene korisnike. Provjera naplate parking usluga vrši se tako što kontrolor unese registracijsku oznaku vozila nakon čega od sistema dobiva sve detalje parkinga (datum, period u kojem je vozilo zauzimalo određeno parking mjesto, iznos parking usluge kao i podatak o tome da li je parking plaćen). U slučaju da parking nije plaćen kontrolor ima mogućnost ispisati kaznu pod odgovarajućim registracijskim brojem vozila.

Krajnji korisnici imaju mogućnost plaćanja parkinga putem mobilnih uređaja ako su registrovani korisnici usluga ili putem parking automata. Kada je u pitanju naplata putem mobilne aplikacije, primarno je potrebno odabrati zonu u kojoj je vozilo parkirano, a zatim unijeti registracijsku oznaku vozila. Nakon toga korisnici dobivaju upute za online plaćanje jednokratne usluge ili potvrdu o uspjehu ako se radi o mjesečnom pretplatniku. Kada je u pitanju naplata putem parking automata, korisnici definišu planirani period zauzimanja parking mjesta te preuzimaju potvrdu o plaćanju.

1. **(20 bodova)** Na osnovu prethodne specifikacije zahtjeva kreirati dijagram slučajeva korištenja.
2. **(10 bodova)** Napraviti use case document za slučaj upotrebe plaćanja parking putem mobilnog uređaja.
3. **(15 bodova)** Za predloženo softversko rješenje kreirati dijagram rasporeda.
4. **(30 bodova)** Prevesti dati C++ kod u jedan od modela podataka po izboru (konceptualni, logički ili fizički model podataka).

class Users{

private:

int UserId;

string FirstName;

string ParentName;

string LastName;

Date BirthDate;

char Gender;

byte ProfilePhoto[];

string Username;

string Password;

Date CreatedDate;

bool Active;

vector<ContactDetails> contactDetails;

};

class ContactTypes {

private:

float ContactTypeId;

string Name;

vector<ContactDetails> contactDetails;

};

class ContactDetails {

private:

string Value;

bool Primary;

ContactTypes contactType;

Users user;

};

class Members : public Users {

private:

Date MembershipDate;

string Title;

string Position;

MembershipTypes membershipType;

vector<Announcements> announcements;

};

class MembershipTypes {

private:

int MembershipTypeId;

string Name;

};

class Announcements {

private:

int AnnouncementId;

string Title;

string ShortDescription;

string Text;

string IconPath;

Date CreatedDate;

bool Active;

bool AllMembers;

AnnouncementDocuments announcementDocument;

std::vector<Members> members;

};

class AnnouncementDocuments {

private:

int AnnouncementDocumentId;

string DocumentName;

string DocumentPath;

string DocumentType;

Announcements announcement;

};

1. **(10 bodova)** U kojim fazama projekta razvoja softverskog proizvoda se koriste dijagram aktivnosti i dijagram stanja i na koji način su povezana (ako uopšte jesu) ova dva dijagrama?
2. **(15 bodova)** Kako poziv rekurzivne poruke na sekvencijalnom dijagramu može utjecati na stanje objekta jedne klase?