

Sveučilište u Zagrebu

Prirodoslovno-matematički fakultet

Matematički odsjek

SPECIFIKACIJA SUSTAVA

MUZEOMAT

Filip Radivojević, Viktoria Brozović, Gabriela Novosel, Matija Petrović

Zagreb, siječanj 2026.

SPECIFIKACIJA SUSTAVA MUZEOMAT

1. UVOD

Ovaj dokument predstavlja specifikaciju softverskog rješenja za sustav Muzeomat - interaktivni kiosk za optimizaciju obilaska muzejskih prostorija. Specifikacija je izrađena u skladu s Unified Process (UP) metodologijom i koristi UML notaciju za modeliranje sustava.

1.1 Svrha dokumenta

Dokument služi za:

- utvrđivanje i modeliranje funkcionalnih i nefunkcionalnih zahtjeva,
- identifikaciju aktera i njihovih interakcija sa sustavom,
- modeliranje sustava pomoću UML dijagrama (use case, activity, class dijagram),
- analizu domene i osnovni analitički model klase.

Specifikacija je pisana na razini analize i fokusira se na što sustav radi, a ne kako je implementiran. Model je jednostavan i daje globalnu sliku sustava, razumljiv korisnicima i prikazan u terminima poslovne domene.

1.2 Opseg sustava

Sustav Muzeomat omogućuje posjetiteljima muzeja da odaberu željene prostorije koje žele obići, nakon čega sustav automatski izračunava optimalnu rutu obilaska koristeći TSP (Travelling Salesman Problem) algoritam. Administratorsko sučelje omogućuje upravljanje prostorijama, praćenje statistika te nadzor nad radom sustava.

Ovaj dokument predstavlja prvi dio projektnog zadatka te služi kao podloga za izradu zasebnog dokumenta oblikovanja, u kojem će sustav biti dodatno razrađen pomoću state machine, deployment i sequence dijagrama na razini oblikovanja.

2. ANALIZA PROBLEMA I ZAHTJEVA

2.1 Opis problema

Muzej kao kulturna institucija nudi posjetiteljima brojne prostorije za obilazak, od kojih svaka sadrži različite eksponate i zahtjeva različito vrijeme za razgledavanje. Posjetitelji često ne znaju optimalnim redoslijedom obići prostorije kako bi minimizirali prijeđenu udaljenost i maksimizirali kvalitetu doživljaja.

Tradicionalno, posjetitelji koriste papirne karte ili se oslanjaju na savjete osoblja, što nije uvijek učinkovito. Nedostaje sustav koji bi automatski planirao rutu na temelju odabranih prostorija i fizičkog rasporeda muzeja.

2.2 Potrebe korisnika

Predloženi sustav treba omogućiti:

- jednostavan odabir željenih prostorija putem interaktivnog sučelja na kiosku,
- automatski izračun optimalne rute koja minimizira prijeđenu udaljenost,
- prikaz rute na karti s redoslijedom posjeta i procjenom vremena,
- administratorima upravljanje prostorijama (dodavanje, izmjena, deaktiviranje),
- praćenje statistika korištenja sustava (najpopularnije rute, prosječno vrijeme).

2.3 Granice sustava

Unutar sustava:

- odabir prostorija putem touchscreen sučelja

- izračun optimalne rute (TSP algoritam)
- prikaz rute na grafičkom prikazu
- upravljanje prostorijama (CRUD operacije)
- evidencija ruta i statistike

Izvan sustava:

- fizičko održavanje kiosk hardvera
- izrada sadržaja o eksponatima
- integracija s vanjskim turističkim sustavima
- sustavi naplate ulaznica
- real-time praćenje pozicije posjetitelja (indoor GPS)

3. ZAHTJEVI SUSTAVA

3.1 Funkcionalni zahtjevi

ID	Opis zahtjeva	Prioritet
FR-01	Sustav mora prikazati popis svih aktivnih prostorija muzeja	Must Have
FR-02	Sustav mora omogućiti posjetitelju odabir više prostorija	Must Have
FR-03	Sustav mora provjeriti da je odabrano najmanje 2 prostorije	Must Have
FR-04	Sustav mora izračunati optimalnu rutu (TSP algoritam)	Must Have
FR-05	Sustav mora prikazati rutu s redoslijedom i ukupnom udaljenošću	Must Have
FR-06	Sustav mora pohraniti izračunate rute u bazu s vremenskim žigom	Must Have
FR-07	Sustav mora omogućiti administratoru dodavanje novih prostorija	Must Have
FR-08	Sustav mora omogućiti administratoru izmjenu podataka prostorija	Must Have
FR-09	Sustav mora omogućiti administratoru deaktiviranje prostorija	Must Have
FR-10	Sustav mora omogućiti administratoru pregled statistika	Should Have
FR-11	Sustav mora omogućiti administratoru prijavu u sustav	Must Have
FR-12	Sustav mora validirati unos podataka pri kreiranju prostorije	Must Have

3.2 Nefunkcionalni zahtjevi

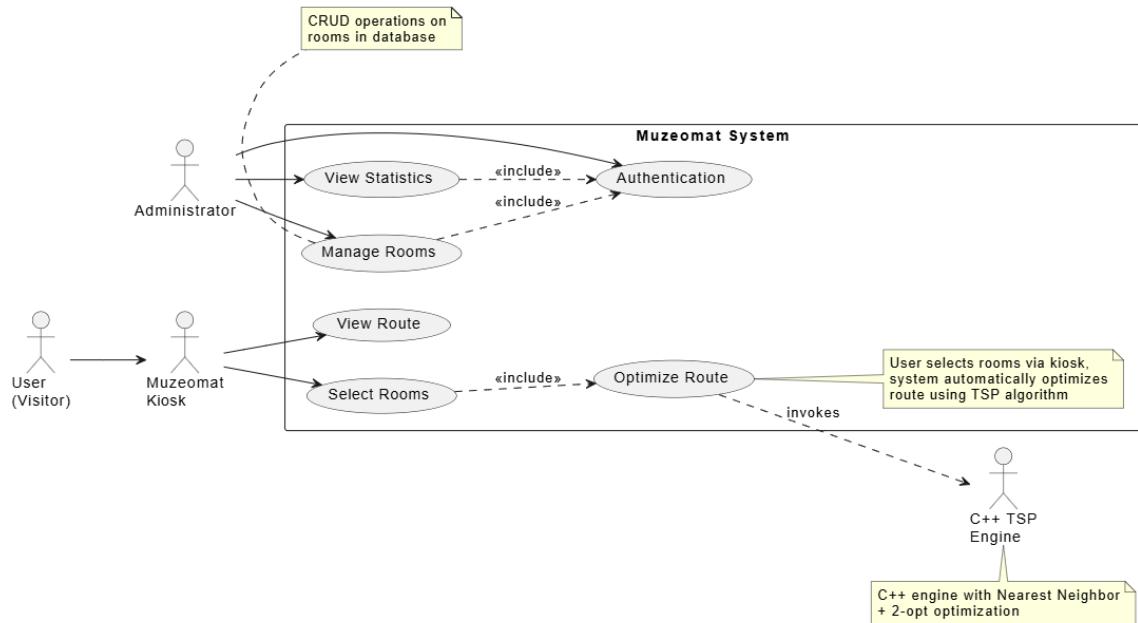
ID	Opis zahtjeva	Prioritet
NFR-01	Vrijeme izračuna rute mora biti kraće od 5 sekundi (do 20 prostorija)	Must Have
NFR-02	Kiosk aplikacija mora biti prilagođena touchscreen interakciji	Must Have
NFR-03	Sustav mora biti dostupan 99% radnog vremena muzeja	Should Have
NFR-04	Sučelje mora biti intuitivno za osobe s osnovnim digitalnim vještinama	Must Have
NFR-05	Administratorski panel mora biti dostupan samo autoriziranim korisnicima	Must Have
NFR-06	Podaci o rutama moraju se čuvati najmanje 12 mjeseci	Should Have

4. USE CASE MODELIRANJE

4.1 Identifikacija aktera

Aktor	Specifikacija
Posjetitelj	Osoba koja dolazi u muzej i koristi kiosk za planiranje rute. Odabire prostorije koje želi vidjeti i prima optimalnu rutu. Ne zahtijeva prijavu.
Administrator	Zaposlenik muzeja odgovoran za upravljanje sadržajem sustava (prostorije, postavke) i pregled statistika. Zahtijeva prijavu s valjanim kredencijalima.
TSP Engine	Vanjski modul koji izvršava algoritam optimizacije rute. Apstraktni aktor koji predstavlja algoritamsku komponentu sustava.

4.2 Use Case dijagram



4.3 Opis use case-ova

Use Case 1: Odabir prostorija za obilazak

Element	Opis
Naziv	Odabir prostorija za obilazak
Akter	Posjetitelj
Preduvjeti	Kiosk aplikacija je pokrenuta i prikazuje popis prostorija

Osnovni tok	<ol style="list-style-type: none"> 1. Posjetitelj pregledava popis dostupnih prostorija 2. Posjetitelj odabire željene prostorije označavanjem 3. Sustav omogućuje odabir više prostorija 4. Posjetitelj potvrđuje odabir klikom na "Optimiziraj rutu"
Alternativni tokovi	<p>A1: Odabrano manje od 2 prostorije</p> <ul style="list-style-type: none"> - Sustav prikazuje poruku "Odaberite najmanje 2 prostorije" - Povratak na korak 2 <p>A2: Posjetitelj odustane</p> <ul style="list-style-type: none"> - Sustav resetira sesiju nakon 5 minuta neaktivnosti
Postuvjeti	Sustav ima validnu listu odabranih prostorija spremnu za optimizaciju

Use Case 2: Optimizacija rute obilaska

Element	Opis
Naziv	Optimizacija rute obilaska
Akter	Posjetitelj (primarni), TSP Engine (sekundarni)
Preduvjeti	Odabrane su najmanje 2 prostorije
Osnovni tok	<ol style="list-style-type: none"> 1. Sustav dohvaća koordinate odabranih prostorija 2. Sustav poziva TSP Engine za izračun optimalne rute 3. TSP Engine vraća optimizirani redoslijed prostorija 4. Sustav izračunava ukupnu udaljenost i vrijeme obilaska 5. Sustav pohranjuje rutu u bazu podataka
Alternativni tokovi	<p>A1: Timeout optimizacije (>5 sekundi)</p> <ul style="list-style-type: none"> - Sustav prikazuje poruku "Optimizacija traje predugo, pokušajte s manje prostorija" <p>A2: Greška u izračunu</p> <ul style="list-style-type: none"> - Sustav prikazuje poruku "Došlo je do greške, pokušajte ponovno"
Postuvjeti	Optimizirana ruta je izračunata i spremna za prikaz

Use Case 3: Prikaz optimalne rute

Element	Opis
Naziv	Prikaz optimalne rute
Akter	Posjetitelj

Preduvjeti	Optimizirana ruta je izračunata
Osnovni tok	1. Sustav prikazuje grafički prikaz rute na tlocrtu muzeja 2. Sustav prikazuje popis prostorija u optimalnom redoslijedu 3. Sustav prikazuje ukupnu udaljenost i procijenjeno vrijeme 4. Posjetitelj pregledava rutu
Alternativni tokovi	A1: Posjetitelj želi novu rutu - Klik na "Nova ruta" vraća na odabir prostorija
Postuvjeti	Posjetitelj ima informacije potrebne za obilazak muzeja

Use Case 4: Upravljanje prostorijama

Element	Opis
Naziv	Upravljanje prostorijama
Akter	Administrator
Preduvjeti	Administrator je prijavljen u sustav
Osnovni tok	1. Administrator odabire opciju "Upravljanje prostorijama" 2. Sustav prikazuje popis svih prostorija 3. Administrator odabire akciju (dodaj, izmjeni, deaktiviraj) 4. Sustav prikazuje odgovarajući obrazac 5. Administrator unosi/mijenja podatke 6. Sustav validira unos 7. Sustav sprema promjene u bazu
Alternativni tokovi	A1: Validacija ne prolazi - Sustav prikazuje grešku i vraća se na korak 5
Postuvjeti	Podaci o prostoriji su ažurirani u bazi podataka

Use Case 5: Pregled statistika

Element	Opis
Naziv	Pregled statistika
Akter	Administrator
Preduvjeti	Administrator je prijavljen u sustav

Osnovni tok	<ol style="list-style-type: none"> 1. Administrator odabire "Statistika" 2. Sustav dohvaca podatke iz baze 3. Sustav izracunava statistike (broj ruta, najpopularnije prostorije) 4. Sustav prikazuje grafičke prikaze i tablice
Alternativni tokovi	-
Postuvjeti	Administrator ima uvid u korištenje sustava

5. ANALIZA DOMENE

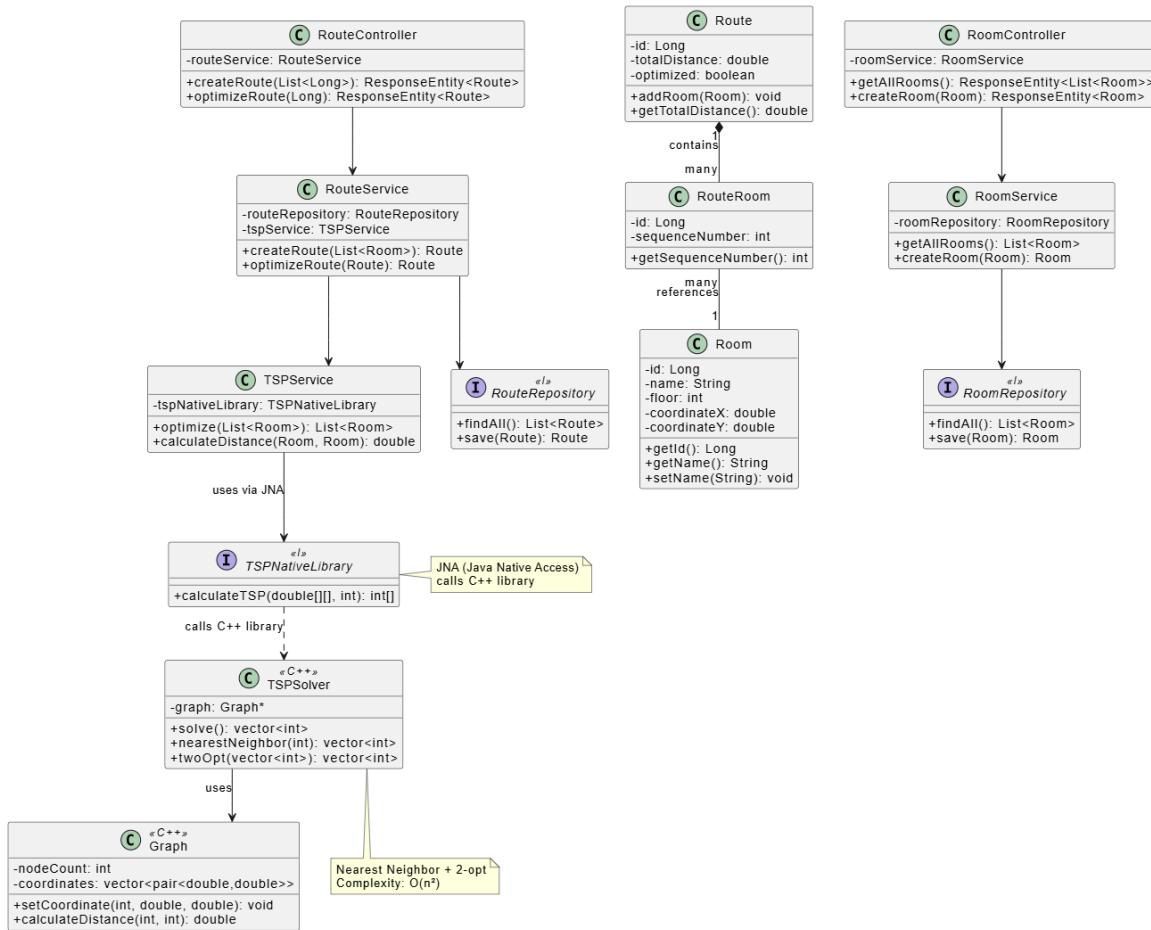
5.1 Konceptualni model

Analiza sustava počinje identifikacijom ključnih koncepta u domeni problema. Ovi koncepti predstavljaju entitete iz stvarnog svijeta koje sustav treba modelirati.^[1]

Glavni koncepti:

- Prostorija (Room) - fizička prostorija u muzeju s eksponatima
- Ruta (Route) - izračunata optimalna ruta obilaska
- Korisnik (User) - administratorski korisnik sustava
- Upravitelj ruta (RouteManager) - koordinira proces optimizacije
- Optimizator (TSPSolver) - algoritam za rješavanje TSP problema

5.2 Class dijagram (razina analize)



5.3 Opis ključnih klasa

Klasa: Room (Prostorija)

Predstavlja jednu prostoriju u muzeju.

Atributi:

- name: String - naziv prostorije
 - description: String - opis sadržaja
 - floorNumber: Integer - kat na kojem se nalazi
 - coordinateX: Double - X pozicija na tlocrtu
 - coordinateY: Double - Y pozicija na tlocrtu

- `avgVisitDuration: Integer` - prosječno vrijeme razgledavanja (minute)
- `isActive: Boolean` - je li prostorija aktivna

Operacije:

- `calculateDistanceTo(other: Room): Double` - računa udaljenost do druge prostorije

Veze:

- Prostorija može biti dio jedne ili više ruta (many-to-many s `Route`)

Klasa: Route (Ruta)

Predstavlja izračunatu optimalnu rutu obilaska.

Atributi:

- `roomSequence: List<Room>` - lista prostorija u optimalnom redoslijedu
- `totalDistance: Double` - ukupna prijeđena udaljenost
- `totalTime: Integer` - procijenjeno ukupno vrijeme
- `createdAt: Timestamp` - vrijeme kreiranja

Operacije:

- `getTotalTime(): Integer` - vraća ukupno vrijeme obilaska
- `getOptimalOrder(): List<Room>` - vraća optimizirani redoslijed

Veze:

- Route sadrži više Room objekata (many-to-many)

Klasa: User (Korisnik)

Predstavlja administratorskog korisnika sustava.

Atributi:

- `username: String` - korisničko ime
- `passwordHash: String` - hash lozinke
- `role: UserRole` - uloga (ADMIN, OPERATOR)

Operacije:

- `authenticate(password: String): Boolean` - autentikacija korisnika
- `hasPermission(action: String): Boolean` - provjera dozvola

Klasa: RouteManager

Koordinira proces optimizacije rute.

Atributi:

- `selectedRooms: List<Room>` - odabrane prostorije
- `currentRoute: Route` - trenutna ruta

Operacije:

- `addRoom(room: Room)` - dodaje prostoriju u odabir
- `optimizeRoute(): Route` - poziva optimizaciju i vraća rutu
- `saveRoute(route: Route)` - sprema rutu u bazu

Veze:

- Koristi `TSPSolver` za optimizaciju (dependency)
- Upravlja `Room` objektima (association)

Klasa: TSPSolver

Algoritam za rješavanje TSP problema.

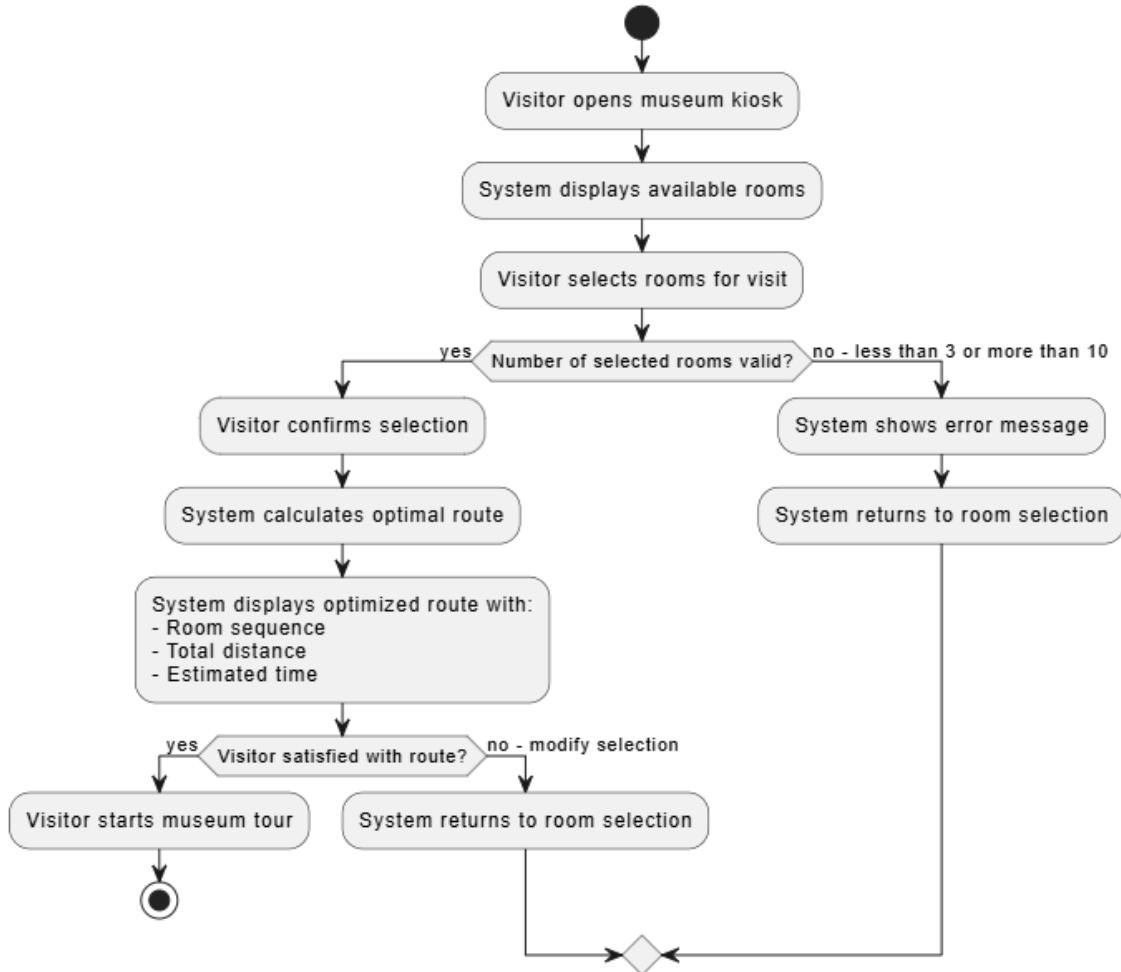
Operacije:

```
- `calculateOptimalPath(rooms: List<Room>): List<Room>` - izračunava optimalnu rutu
```

6. MODELIRANJE PONAŠANJA SUSTAVA

6.1 Activity dijagram - Optimizacija rute

Route Optimization - Activity Diagram (Analysis Level)



6.2 Opis toka aktivnosti

Activity dijagram prikazuje:

1. Pokretanje sesije - posjetitelj započinje interakciju
2. Odabir prostorija - iterativno označavanje željenih prostorija
3. Validacija odabira - provjera minimalnog broja prostorija (decision node)

4. Optimizacija rute - poziv TSP algoritma
5. Prikaz rezultata - grafički prikaz optimalne rute
6. Pohrana u bazu - paralelna aktivnost spremanja podataka

Swimlanes prikazuju odgovornosti:

- Posjetitelj - korisničke radnje (odabir, pregled)
- Kiosk sustav - validacija, prikaz, upravljanje sesijom
- TSP Engine - izračun optimalne rute
- Baza podataka - pohrana i dohvata podataka