

Uvod

Ovaj projekat je realizovan kao deo predmeta *Softver nadzorno-upravljačkih sistema* i ima za cilj implementaciju sistema za **upravljanje robotskom rukom** korišćenjem **WCF tehnologije**.

Sistem se sastoji iz:

- **Serverske strane** – WCF servis implementiran u projektu *SnusProject*, koji omogućava definisane operacije nad robotskom rukom, proveru permisija, bezbednu komunikaciju i evidenciju svih pokušaja operacija u bazi podataka.
- **Klijentskih aplikacija** – projekat *ClientApp*, koji predstavlja korisnički interfejs preko koga tri različita klijenta mogu slati zahteve robotskoj ruci. Svaki klijent ima unapred definisane permisije i prioritet izvršavanja.

Korišćene tehnologije i alati:

- Programski jezik: **C#**
- Framework: **.NET (WCF)**
- Klijentski UI: **WPF**
- Baza podataka: **Entity Framework + PostgreSQL**
- Sigurnost komunikacije: korišćen je kriptografski algoritam (detalji u posebnom poglavlju).

Cilj ovog projekta je da se omogući simulacija višekorisničkog upravljanja robotskom rukom, sa kontrolisanim pristupom i zabeleženim istorijatom svih izvršenih i odbijenih operacija.

Specifikacija sistema

Na osnovu zadatka definisana je sledeća specifikacija sistema:

Dozvoljene operacije

Robotskoj ruci se mogu poslati sledeće operacije:

- Pomeri se levo za 1.
- Pomeri se desno za 1.
- Pomeri se gore za 2.
- Pomeri se dole za 2.
- Zarotiraj ruku za 90°.

Ograničenje prostora

Robotska ruka se nalazi u prostoru dimenzija **5 × 5**.

- Ukoliko bi operacija dovela do izlaska iz ovog prostora, zahtev se odbija.
- Na primer, šest uzastopnih pomeranja ulevo rezultiraće odbijanjem šeste operacije.

Permisije klijenata

Tri klijenta koja pristupaju sistemu imaju različite permisije:

- **Klijent 1:** pun pristup svim operacijama.
- **Klijent 2:** dozvoljena samo pomeranja, ne i rotacija.
- **Klijent 3:** dozvoljena samo rotacija ruke.

Ukoliko klijent pokuša da izvrši operaciju koja nije u okviru njegovih permisija, zahtev se odbija.

Prioriteti klijenata

- **Klijent 1** ima najviši prioritet i njegovi zahtevi se uvek izvršavaju pre svih ostalih.
- **Klijent 2 i Klijent 3** imaju jednak prioritet.

Sigurnost komunikacije

Komunikacija između servera i klijenata je zaštićena kriptografskim algoritmom (opis u posebnom poglavlju o bezbednosti).

Evidencija operacija

Svaka operacija – bilo uspešno izvršena ili odbijena – evidentira se u bazi podataka sa sledećim informacijama:

- identitet klijenta,
- tip operacije,
- vreme pokušaja,
- status (izvršena / odbijena).

Arhitektura sistema

Arhitektura projekta zasnovana je na **klijent–server modelu**. Glavni deo sistema čini **WCF servis** (projekat *SnusProject*), dok klijentske aplikacije (projekat *ClientApp*) komuniciraju sa njim putem definisanih servisa i metoda. Svi zahtevi i rezultati beleže se u bazi podataka (*PostgreSQL*).

Pregled komponenti

- **Server (SnusProject)**
 - Implementira logiku rada robotske ruke.
 - Definiše **servisne kontrakte** (*IRobotArmService*, *IRobotArmCallback*).
 - Implementira proveru permisija i prioriteta klijenata.
 - Osigurava da robotska ruka ostane u okviru 5×5 prostora.
 - Čuva sve pokušaje operacija u bazi preko *DbContext* klase i *Entity Framework* migracija.

- Upravlja bezbednošću komunikacije (pomoću klase SecurityHelper).
- **Klijenti (ClientApp)**
 - Predstavljani kroz WPF aplikaciju.
 - Omogućavaju korisnicima da biraju operacije i šalju ih ka serveru.
 - Svaki klijent ima svoj skup permisija i poštovanje prioriteta.
 - U interfejsu korisnik vidi trenutne dostupne komande i rezultat izvršenja (ili odbijanja).
- **Baza podataka (PostgreSQL)**
 - Sadrži tabelu za logovanje svih pokušaja operacija.
 - Svaki zapis sadrži: ID operacije, klijent, tip komande, vreme i status.

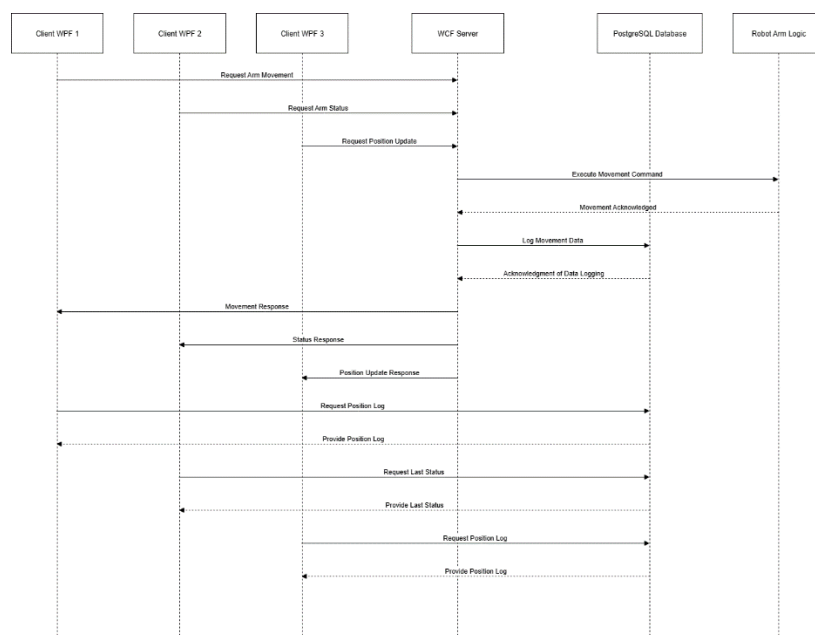
Komunikacija

Komunikacija između servera i klijenata realizovana je putem **WCF dual binding** mehanizma:

- **Klijenti** šalju zahtev serveru koristeći metode definisane u IRobotArmService.
- **Server** može slati povratne informacije ka klijentima koristeći IRobotArmCallback.

Dijagram sistema

Dijagram prikazuje odnose između komponenti:



Bezbednost

S obzirom da sistem omogućava komunikaciju između više klijentskih aplikacija i servera, bilo je neophodno obezbediti da razmena podataka bude zaštićena od manipulacija i neovlašćenog pristupa.

Kriptografski algoritam

Za zaštitu komunikacije korišćen je **HMAC (Hash-based Message Authentication Code)** mehanizam baziran na algoritmu **SHA-256**.

- Implementacija se nalazi u klasi `SecurityHelper`.
- Koristi se tajni ključ (`secretKey`), poznat samo serveru i klijentima, za generisanje i verifikaciju heširanih vrednosti.

Proces generisanja i verifikacije

- Kada klijent šalje poruku ka serveru, generiše se HMAC pomoću metode `ComputeHmac(message)`.
- Server, po prijemu poruke, ponovo računa HMAC na osnovu sadržaja i upoređuje ga sa vrednošću koju je poslao klijent.
- Ako se vrednosti poklapaju, poruka se smatra autentičnom i dozvoljava se dalje procesiranje.
- Ukoliko se vrednosti razlikuju, poruka se odbacuje.

Prednosti korišćenog rešenja

- **Integritet podataka** – garantuje da poruka nije menjana u prenosu.
- **Autentičnost pošiljaoca** – samo klijent i server koji poznaju tajni ključ mogu generisati validan HMAC.
- **Jednostavna implementacija** – ne zahteva dodatnu infrastrukturu kao što su sertifikati, što je pogodno za akademske projekte.

Ilustracija metode `ComputeHmac` iz `SecurityHelper` klase:

```
public static string ComputeHmac(string message)
{
    using (var hmac = new HMACSHA256(Encoding.UTF8.GetBytes(secretKey)))
    {
        byte[] hash = hmac.ComputeHash(Encoding.UTF8.GetBytes(message));
        return Convert.ToBase64String(hash);
    }
}
```

Implementacija

Serverska strana (*SnusProject*)

Serverska aplikacija implementirana je kao **WCF servis**. Njegove glavne komponente su:

- **Servisni kontrakti**
 - `IRobotArmService` definiše operacije koje klijenti mogu pozivati (pomeranje i rotacija).
 - `IRobotArmCallback` omogućava serveru da šalje povratne informacije ka klijentima (dual binding).

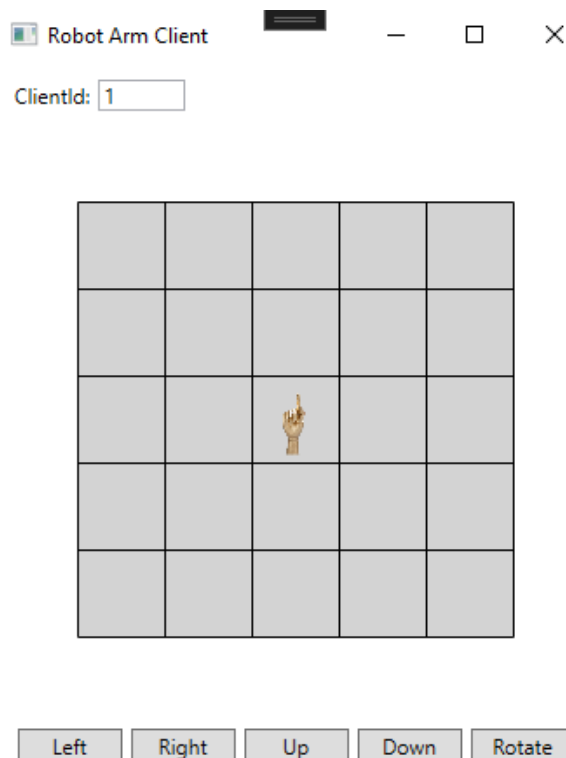
- **Logika robota**
 - Implementacija servisnih metoda proverava da li klijent ima potrebne permisije za traženu operaciju.
 - Proverava se da li bi izvršenje operacije izazvalo izlazak iz 5×5 prostora; u tom slučaju operacija se odbija.
 - Poštovanje prioriteta: zahtevi **Klijenta 1** se obrađuju uvek pre svih ostalih.
- **Persistencija podataka**
 - Za svaki pokušaj operacije kreira se zapis u bazi preko klase DatabaseContext.
 - Zapis sadrži identitet klijenta, tip operacije, vreme i status (izvršena / odbijena).
- **Bezbednost**
 - Klasa SecurityHelper se koristi za verifikaciju autentičnosti poruka putem HMAC-SHA256.

Klijentska strana (*ClientApp*)

Klijentska aplikacija je realizovana kao **WPF aplikacija**, što omogućava grafički interfejs za interakciju korisnika.

- Korisnik bira jednu od dostupnih operacija.
- Zahtev se šalje serveru uz HMAC vrednost za proveru autentičnosti.
- Klijent prikazuje odgovor servera: da li je operacija uspešno izvršena ili odbijena (zbog nedostatka permisija, prioriteta ili izlaska iz 5×5 granica).

Na slici je prikazan korisnički interfejs klijentske aplikacije. Korisnik može odabrati operaciju nad robotskom rukom i odmah dobija informaciju o statusu izvršenja:



Primeri metoda iz IRobotArmService:

```
[OperationContract]
1 reference
Task<OperationResult> EnqueueRotateAsync(int clientId, string hmac);

[OperationContract]
2 references
RobotArmState GetCurrentState();

[OperationContract]
1 reference
void Subscribe();

[OperationContract]
1 reference
void Unsubscribe();
```

Testiranje

Kako bi se potvrdila ispravnost sistema i poštovanje svih definisanih pravila, sprovedeni su testovi različitih scenarija.

6.1 Testiranje permisija

- **Scenario 1:** Klijent 2 pokušava da izvrši rotaciju ruke.
 - **Očekivano:** Operacija se odbija jer Klijent 2 nema permisiju za rotaciju.
 - **Rezultat:** Sistem ispravno odbija zahtev i beleži neuspešan pokušaj u bazi.
- **Scenario 2:** Klijent 3 pokušava da pomeri ruku ulevo.
 - **Očekivano:** Operacija se odbija jer Klijent 3 ima dozvolu samo za rotaciju.
 - **Rezultat:** Sistem ispravno odbija zahtev i beleži ga u bazi.

Testiranje granica prostora

- **Scenario 3:** Klijent 1 šalje šest uzastopnih komandi "Pomeri se ulevo".
 - **Očekivano:** Prvih pet operacija se izvršavaju, šesta se odbija jer bi ruka izašla iz 5×5 prostora.
 - **Rezultat:** Sistem ispravno izvršava prvih pet i odbija šestu, beležeći sve pokušaje u bazi.

Testiranje prioriteta

- **Scenario 4:** Klijent 2 i Klijent 3 šalju istovremeno operacije, dok Klijent 1 takođe šalje zahtev.
 - **Očekivano:** Operacija Klijenta 1 se izvršava prva, zatim se izvršavaju operacije Klijenta 2 i Klijenta 3.
 - **Rezultat:** Sistem ispravno poštuje prioritet, beležeći redosled izvršenja.

Testiranje bezbednosti

- **Scenario 5:** Klijent šalje zahtev sa pogrešno izračunatim HMAC-om.
 - **Očekivano:** Server odbija zahtev jer ne može da verifikuje autentičnost poruke.
 - **Rezultat:** Sistem ispravno odbija zahtev i beleži neuspešan pokušaj.

Testiranje logovanja

- U svim scenarijima, izvršene i odbijene operacije proverene su u bazi podataka.
- Svaki zapis sadrži tačan identitet klijenta, tip komande, vreme pokušaja i status.
- Rezultati potvrđuju da je sistem uvek ispravno evidentirao sve pokušaje.

Zaključak

U ovom projektu implementiran je sistem za **upravljanje robotskom rukom** zasnovan na **WCF tehnologiji**, prema unapred zadatoj specifikaciji.

Sistem omogućava:

- višekorisničko upravljanje robotskom rukom sa jasno definisanim **permisijama i prioritetima**,
- **bezbednu komunikaciju** između servera i klijenata putem HMAC-SHA256 mehanizma,
- proveru granica kretanja robota u okviru definisanog prostora od 5×5 polja,
- **logovanje svih pokušaja operacija** u PostgreSQL bazu podataka, uključujući i neuspešne zahteve.

Testiranjem je potvrđeno da sistem pravilno odbacuje neovlašćene i nevalidne zahteve, poštuje prioritete klijenata i uvek evidentira stanje u bazi.

Ovaj projekat uspešno demonstrira osnovne principe:

- **distribuiranih sistema** (klijent–server komunikacija),
- **upravljanja privilegijama korisnika**,
- **sigurnosti komunikacije** korišćenjem kriptografskih mehanizama,
- kao i **pouzdanog čuvanja podataka** o svim izvršenim operacijama.

Moguća unapređenja

Iako sistem u potpunosti zadovoljava zadatu specifikaciju, postoje mogućnosti za dalji razvoj:

- Uvođenje **detaljnije vizuelizacije robotske ruke** kroz animacije ili 3D prikaz.
- Primena **složenijih algoritama za enkripciju**.
- Dodavanje **administratorskog interfejsa** za pregled logova i statistike o korišćenju sistema.