**Projektuppgift**

*DT071G*

**Programmering i C#.NET: Projekt**

AI Datorer AB:s Systemadministrativa Kommandotolk

**Max Karlstedt**

**MITTUNIVERSITETET  
Avdelningen för informationssystem och -teknologi**

**Författare:** Max Karlstedt, [maka2207@student.miun.se](mailto:xxxxx0000@student.miun.se)  
**Utbildningsprogram:** Webbutveckling, 120 hp  
**Huvudområde:** Datateknik  
**Termin, år:** HT, 2023

# Sammanfattning

**Syfte:** Webbprojektet avser AI Datorer AB:s systemadministrativa intranät där en systemadministratör via en kommandotolk i C#.NET ska kunna logga in och sedan administrera användare och deras behörigheter via anslutning till ett REST API vilket i sin tur kommunicerar med en icke-relationell databas av slaget MongoDB. **Teori:** Sammanfattningsvis är C# ett starkt typat objektorienterat programmeringsspråk medan .NET är en särskild utvecklingsplattform åt just det starkt typade objektorienterade programmeringsspråket. MongoDB är en icke-relationell databasform medan REST API är ett mellangränssnitt mellan mjukvaror - så att C#.NET kan göra ändringar i MongoDB indirekt via REST API:t. **Metod:** Utvecklingsmiljön har varit Visual Studio 2022 för C#.NET-delen medan MongoDB Compass har använts för att kontrollera att kommandotolkens kommandon faktiskt gjort ändringar i databasen. **Konstruktion:** a. **Resultat:** a. **Diskussion:** a. **Slutsatser:** a. Källförteckning och eventuella bilagor tillkommer.

.

# Innehållsförteckning

[Sammanfattning iii](#_Toc152675786)

[Innehållsförteckning iv](#_Toc152675787)

[1 Introduktion 1](#_Toc152675788)

[1.1 Bakgrund och kunden 1](#_Toc152675789)

[1.2 Kundens målgrupp 1](#_Toc152675790)

[1.3 Kundens budget 1](#_Toc152675791)

[1.4 Juridiska överväganden 1](#_Toc152675792)

[1.5 Överenskommen kravspecifikation 2](#_Toc152675793)

[1.6 Kapitelöversikt 2](#_Toc152675794)

[2 Teori 4](#_Toc152675795)

[2.1 C#.NET 4](#_Toc152675796)

[2.1.1 .NET 4](#_Toc152675797)

[2.1.2 C# 5](#_Toc152675798)

[2.2 Kommandotolk 5](#_Toc152675799)

[2.3 REST API & MongoDB 5](#_Toc152675800)

[3 Metod 6](#_Toc152675801)

[3.1 Integrerad utvecklingsmiljö 6](#_Toc152675802)

[3.2 Publicering 6](#_Toc152675803)

[4 Konstruktion 7](#_Toc152675804)

[4.1 Anslutning till REST API via httpClient & httpClientHandler 7](#_Toc152675805)

[4.2 Tolkning av kommandon och deras parametrar 7](#_Toc152675806)

[4.3 (Fel)hantering av inmatade data 7](#_Toc152675807)

[4.4 JSON-hantering via Newtonsoft.JSON 7](#_Toc152675808)

[5 Resultat 8](#_Toc152675809)

[5.1 a 8](#_Toc152675810)

[6 Slutsatser 9](#_Toc152675811)

[Källförteckning 10](#_Toc152675812)

# Introduktion

Projektrapporten börjar med att klargöra vem kunden är, varför kunden har beställt denna IT-lösning, såväl som deras budget (det vill säga, så kallade avgränsningar i vad som ska (och inte) levereras), eventuella juridiska överväganden, samt den överenskomna kravspecifikationen mellan kunden och uppdragstagaren. Till sist i detta inledande kapitel presenteras de resterande kapitlen i form av en kapitelöversikt.

## Bakgrund och kunden

AI Datorer har bedrivit försäljning av datorkomponenter sedan några år tillbaka och nu i samband med en separat beställning av att ta fram ett MERN-stacksbaserat Intranät (se *HT2023 DT162G - Projekt av Max Karlstedt*) för att underhålla deras inventerande av datorkomponenter de har till salu så önskar de även ett separat mjukvaruprogram som enbart en utnämnd systemadministratör har tillgång till.

Med andra ord önskar kunden en så kallad kommandotolk där systemadministratören kan logga in för att sedan kommunicera direkt med företagets databas där både datorkomponenter och användare (=anställda vid företaget AI Datorer AB) finns lagrade för att på så vis styra över användarnas olika behörigheter till databasen.

## Kundens målgrupp

I denna separata beställning av en systemadministrativ kommandotolk så är det endast den utvalda systemadministratören som är kundens målgrupp. Denna utvalda person kommer att få exklusiv tillgång att kunna utläsa, skapa, ändra och/eller radera befintliga användare via den direkta åtkomsten till databasen av slaget MongoDB - allt tack vare textbaserade kommandon som skrivs i en simpel kommandotolk.

## Kundens budget

I och med att kunden redan har en separat beställning om ett MERN-stacksbaserat Intranät där utvecklingen av REST API:t så kommer denna systemadministrativa kommandotolk vara begränsad till att kunna hantera/administrera användare samt deras behörigheter för det separata Intranätet. REST API:t och MongoDB kommer dessutom att ej redogöras här mer än dess grundläggande teori.

## Juridiska överväganden

Kunden har gett samtycke till uppdragstagaren att få använda REST API:t och dess anslutna MongoDB vilket har tagits fram i en separat beställning. Beträffande känslig information så lagras endast tillfälligt en så kallad *JSON Web Token* (JWT; 2) inuti kommandotolkprogrammet vilket används för att på ett autentiserande och auktoriserande vis kommunicera med REST API:t och dess anslutna MongoDB. När programmet stängs så försvinner även denna JWT och då systemadministratören måste logga in på nytt igen varje gång kommandotolkprogrammet startas så kommer JWT:n att nollställas.

## Överenskommen kravspecifikation

Kunden önskar kunna göra följande i den systemadministrativa kommandotolken:

* Ha ett särskilt lösenord för att ens kunna få ange inloggningsuppgifter som systemadministratör mot REST API:t och dess anslutna databas.
* Efter lyckad inloggning ska systemadministratören kunna:
  + Utläsa alla befintliga användares uppgifter och behörigheter.
  + Ändra en befintlig användares uppgifter och behörigheter.
  + Radera en befintlig användares uppgifter och behörigheter.
  + Skapa en ny användare med uppgifter och behörigheter.
  + Logga ut och/eller avsluta kommandotolken via ett kommando.
* Kommandon ska kunna skrivas i stilen "adduser -username -password -email -access:get\_images,get\_components" där mellanslag och bindestreck används för att ange parametrar efter det huvudsakligt kommando.
* Omfattande felhantering för att inte kunna råka skicka otillåtna REST API-anrop (exempelvis REST API-anrop om att radera/ändra nuvarande systemadministratör eller att exempelvis skapa en ny systemadministratör). En stor del av detta kommer dock redan vara åtgärdat på REST API-sidan, men kunden vill ha samma strikthet implementerat i kommandotolken också bara för säkerhetens skull.
* Omfattande hjälp direkt inuti kommandotolken om hur olika kommandon ska skrivas med dess parametrar så att systemadministratören aldrig behöver lämna kommandotolken vid användning för att läsa på om hur något kommando ska skrivas som.

## Kapitelöversikt

I kommande kapitel kommer följande att avhandlas:

[Kapitel 2](#_Teori) går igenom grundläggande teori för C#.NET, kommandotolkar i allmänhet, samt mycket kortfattat om REST API och MongoDB trots att de två sistnämnda ej är det primära fokuset i denna beställning från kunden.

[Kapitel 3](#_Metod) visar metoderna för att ens kunna genomföra det som beskrivs i de nästkommande kapitlen.

[Kapitel 4](#_Konstruktion) berör konstruktionen av kommandotolkprogrammet i C#. NET samt hur olika saker och ting har lösts för att leva upp till kundens kravspecifikationer.

[Kapitel 5](#_Resultat) presenterar bilder med olika körda kommandon i kommandotolken, hur dessa då speglats sig i REST API:t och MongoDB, såväl som hur dessa då förhållit sig till kundens kravspecifikationer.

[Kapitel 6](#_Slutsatser) är slutsatser med reflektioner om arbetet samt knyter ihop säcken för projektrapporten. Efter detta följer vedertagen källförteckning samt eventuella bilagor.

# Teori

Detta kapitel redogör grundläggande teori som behövs för att förstå de valda implementerade lösningarna åt AI Datorer AB:s framtagna Systemadministrativa Kommandotolk vilket endast systemadministratörer har behörighet till.

## C#.NET

Trots att C#.NET skrivs hopsatt genomgående i denna projektrapport så är det värt att känna till att kunna särskilja på C# och .NET där det förstnämnda beskriver ett statiskt typat objektorienterat programmeringsspråk *(se* [*2.1.2 C#*](#_C#)*)* vars första version först släpptes i februari 2002(1, s.46). .NET *(se* [*2.1.1 .NET*](#_.NET)*)* är istället en av flera olika överlappande utvecklingsplattformar för utvecklingen av applikationer och (webb)tjänster(1, s.10).

### .NET

Det så kallade .NET öppen källkod och så kallad *managed code*-baserat mjukvaruramverk för Windows-, Linux- samt macOS-operativsystem, och det drivs av anställda hos Microsoft via .*NET Foundation*(3).

Då C# är ett .NET-kompatibelt språk så kan det först kompileras till ett intermediärt språk (eng. *Common Intermediate Language; CLI - observera att CLI också kan stå för Command-Line Interface: sv. kommandotolk*) vilket i sin tur sedan kompilerar det intermediära språket till körbar maskinkod(3, *Architecture*).

.NET ska inte förväxlas med .NET Framework vilket är skyddad mjukvara som körs på Windows-baserade operativsystem utvecklade av Microsoft(4) med start redan så tidigt som på sena 1990-talet och var då en del av den så kallade ".NET-strategin"(4, *History*).

När .NET 5.0 släpptes så kunde det egentligen ha hetat .NET Core 4.0 men ordet "Core" släpptes från varumärket och versionsnumret 4.0 undveks för att .NET Framework 4.x-versioner hade funnits redan sen 2010(3, *History*).

Sammanfattningsvis så används moderna .NET för utveckling på flera olika operativsystem inklusive mobila operativsystem såsom iOS och Android, medan .NET Framework är endast för att kunna underhålla befintliga applikationer på Windows-operativsystem enbart(1, s.19).

Till sist så kan Xamarin användas för att utveckla mobila appar bland annat(1,s.19) men värt att nämna är att det kommer att bytas ut och ersättas fullständigt av .NET MAUI någon gång i maj 2024, detta meddelat av Microsoft redan den 23:e maj 2022(5).

### C#

C# anammar statiskt typat, starkt typat, lexikalt skopat, imperativt, deklarativt, funktionellt, generiskt, objekt-/klassorienterad programmering såväl som komponentbaserad programmering. Språket utvecklades av danska Anders Hejlsberg vid Microsoft under februari 2000 - med första versionen C# 1 släppt i februari 2002(1, s.46) - och från och med i november 2023 så gäller C# versionen 12.0 i .NET 8.0(6).

(1, s.62-63)

## Kommandotolk

a

## REST API & MongoDB

a

# Metod

Följande kapitel redogör för vilka verktyg som har använts i projektet.

## Integrerad utvecklingsmiljö

I projektet - inuti Windows 10 - har *Visual Studio Community 2022* använts som integrerad utvecklingsmiljö för att koda C#-kod. Hämtat här: [https://visualstudio.microsoft.com/#vs-section](https://visualstudio.microsoft.com/%23vs-section) (*Visual Studio Community 2022* specifikt).

Versionsaspekterna är:

* Version 17.7.6 (senast uppdatering 2023-10-31) för själva *Visual Studio Community 2022 (Help -> Check for updates -> "Current version:")*.
* Version .NET 7.0 för .NET-ramverket *(Project -> Properties -> Application -> General -> "Target framework")*.
* Version 11.0 för C#-programmeringsspråket (*Project -> Properties -> Build -> Advanced -> "Language version")*.

Det grafiska användargränssnittet *MongoDB Compass* har använts för att kontrollera att data faktiskt har ändrats i databasen via REST API:t. Hämtat här: <https://www.mongodb.com/try/download/compass>

Här har MongoDB hämtats och installerats på Windows 10: <https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-windows/>

## Publicering

Projektet har ingen publicerad webbplats utan projektet finns att klona från följande GitHub repo: <https://github.com/WebbkodsLarlingen/dt071g-maka2207-projekt>

# Konstruktion

Detta kapitel avhandlar det faktiska praktiska arbetet i hur saker och ting har lösts samt varför de har lösts på de sätt de har lösts på.

## Anslutning till REST API via httpClient & httpClientHandler

a.

## Tolkning av kommandon och deras parametrar

a

## (Fel)hantering av inmatade data

a

## JSON-hantering via Newtonsoft.JSON

a

# Resultat

I detta näst sista kapitel innan slutsatser redogörs webbplatsens implementerade utseende och funktionalitet i tre olika populära webbläsare. Sedan reflekteras juridiska frågor rörande webbplatsen erhållna från projektuppgiftens beskrivning.

## a

a.

# Slutsatser

a.

Källförteckning

1. Price, Mark J. *C# 11 and .NET 7 - Modern Cross-Platform Development Fundamentals - Start building websites and services with ASP.NET Core 7, Blazor, and EF Core 7 (Seventh Edition)*. UK: Packt Publishing Ltd. 2022
2. Wikipedia, "JSON Web Token", <https://en.wikipedia.org/wiki/JSON_Web_Token> Ändrad senast: 2023-12-01 Hämtad: 2023-12-05
3. Wikipedia, ".NET", <https://en.wikipedia.org/wiki/.NET> Ändrad senast: 2023-11-21 Hämtad: 2023-12-05
4. Wikipedia, ".NET Framework", <https://en.wikipedia.org/wiki/.NET_Framework> Ändrad senast: 2023-11-22 Hämtad: 2023-12-05
5. Wikipedia, "Xamarin", [https://en.wikipedia.org/wiki/Xamarin#.NET\_MAUI](https://en.wikipedia.org/wiki/Xamarin%23.NET_MAUI) Ändrad senast: 2023-11-25 Hämtad: 2023-12-05
6. Wikipedia, "C Sharp (programming language)", <https://en.wikipedia.org/wiki/C_Sharp_(programming_language)> Ändrad senast: 2023-11-18 Hämtad: 2023-12-05
7. Wikipedia, "", a Ändrad senast: Hämtad: 2023-12-05
8. Wikipedia, "", a Ändrad senast: Hämtad: 2023-12-05
9. Wikipedia, "", a Ändrad senast: Hämtad: 2023-12-05
10. Wikipedia, "", a Ändrad senast: Hämtad: 2023-12-05
11. Wikipedia, "", a Ändrad senast: Hämtad: 2023-12-05
12. Wikipedia, "", a Ändrad senast: Hämtad: 2023-12-05
13. Wikipedia, "", a Ändrad senast: Hämtad: 2023-12-05
14. Wikipedia, "", a Ändrad senast: Hämtad: 2023-12-05
15. Wikipedia, "", a Ändrad senast: Hämtad: 2023-12-05
16. Wikipedia, "", a Ändrad senast: Hämtad: 2023-12-05
17. Wikipedia, "", a Ändrad senast: Hämtad: 2023-12-05
18. Wikipedia, "", a Ändrad senast: Hämtad: 2023-12-05