

GIK299 Projektrapport

Grundläggande information

Jonathan Klinratree & Daniel Stadler, grupp 32
E-post: h25jonkl@du.se & h25dasta@du.se
Kurskod: GIK299
Datum: 2026-01-10
Projekt A (Bokningsapp)

Syfte

Syftet med denna uppgift är att visa våra förmågor inom C#. Visa hur vi har utvecklats under denna kurs och sedan kunna demonstrera vårt projekt. Detta projekt visar att vi kan använda egendefinierande klasser samt objekt.

Att visa att ni har färdigheter och förmågor inom problemlösning, användning av API:er, kontrollstrukturer, datastrukturer, egendefinierade klasser och objekt, samt användning av UML-diagram och flödesscheman.

Projektval och målgrupp

Vi valde projekt A vilket var bokningsapp. Denna app är för alla med tillgång av ett fordon med hjul. Ålderklass spelar ingen roll utan bara ägande av ett objekt med hjul

User stories

1. Sortera bokningar i tidsordning

Som företagsägare vill jag ha en sorterad **lista** på vår data.

Acceptanskriterie

Bokningar ska visas i kronologisk ordning baserat på datum och tid.

Sorteringen ska fungera oavsett i vilken ordning bokstavingarna registreras.

Listan ska uppdateras korrekt efter bokning och avbokning.

2. Sparning i databas

Som företagsägare vill jag kunna registrera kundens namn, registreringsnummer och telefonnummer så att kunduppgifter sparas korrekt i databasen.

Acceptanskriterie

Möjlig att registrera kundens namn, registreringsnummer och telefonnummer.

Alla fält måste vara uppfyllda innan uppgifterna sparas.

Systemet ska kolla om inmatad data har korrekt format.

3. Säkert lösenord

Som företagsägare vill jag ha mitt lösenord sparat säkert.

Acceptanskriterie

Systemet ska kräva ett lösenord vid start.

Lösenordet ska vara sparat säkert.

Försvåra obehörig åtkomst.

4. Avbokningar

Som företagare vill jag kunna ta bort avbokningar i databasen.

Acceptanskriterie

Avbokning genom registreringsnummer.

Avbokning ska sparas efter den är klargjord.

Programmet ska uppdatera både listan och dagens bokningar.

Egna tolkningar, överväganden och val

User stories har varit ett sätt att kunna tänka från ett annat perspektiv och utvecklat vår syn på hur användaren vill ha ett program. Det har varit ett sätt att sätta mål att lösa för att kunna förbättra användarupplevelsen. Sedan har det varit vi som har fattat det slutliga beslutet. Det vi har bestämt är att följa våra 4 user stories och sedan ta hjälp av de 5 user stories vi fick.

Användning av AI-verktyg (t.ex. ChatGPT, Copilot)

AI-verktyg användes men på olika sätt. Riders egna AI-verktyg användes för att generera och autokorrigera kod, som minskade tiden som krävdes för att skriva kod. Detta gjorde det smidigare att arbeta och gav ett bättre flöde under arbetsprocessen.

Chat-GPT användes för ett problem som uppstod. Problemet var att visa alla datum mellan två intervaller, då användes Chat-GPT som ett verktyg att kunna bolla idéer. Frågor som "Vad är optimalt mellan TryParse eller TryParseExact" och "Ge exempel på bokningar som skulle passa inom dessa datum?".

Fel uppstod med inga som inte var enkla att lösa. Genom IDE:ernas egna inbygga kodefunktion kunde problem enkelt sökas igenom och lösas, där vanliga problem var stavfel, felaktig användning av stora och små bokstäver och syntaxfel.

AI-verktyg underlättar arbete men om man använder utan att utveckla sig själv leder det till en negativ effekt på sin egna utveckling. Men om man ser det som en kamrat kan den både utveckla din egna förmåga och ge dig bredare syn på problem.

Metod

En dator behövdes med en nedladdad IDE som stödjer C# i detta fall hade vi på Windows Visual Studio 2026 version 18.1.1, tillsammans med ramverket .NET 10.0. Och på macOS användes Rider 2025.2.4, tillsammans med ramverket .NET 10.0. Samarbete skedde genom programmet GIT..

Design

I vår bokningsapp är gränssnittet uppbyggt kring en tydlig meny som gör det enkelt för användaren att navigera. Adminmenyn innehåller: Boka, Checka alla bokningar, söka intervaller bokningar, kolla dagens bokningar, ta bort bokningar via namn och slutligen avsluta programmet. Programmet vägleder användaren under hela programmets start.

Samarbete

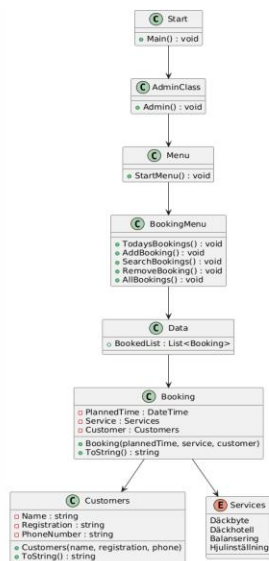
Samarbete skedde med hjälp av GIT där båda arbetade under distans men under samtal. Uppgifter delades och sedan skedde en igenomsökning.

Testning

Testning genomfördes med både normal användare, djävulsk användare och en mormor-test. Inget problem skedde under en normal användning där instruktioner följdes. Även under en djävulsk användare skedde inga stora problem utan programmet beteedde sig som förväntat och försökte vägleda användaren till att ange korrekt format. Och till sist användes en vän utanför IT -världen som ett sista test. Där inga komplikationer uppstod utan programmet visade sig vara väldigt användarvänligt. .

Arkitektur och UML/Flödesscheman

Programmet är uppbyggt enligt en tydlig struktur där klasser är uppdelade som gör koden mer lättläst och enkel att vidareutveckla.



Resultat

```

---Adminpanel---
1. Visa dagens bokningar
2. Lägg till en ny bokning
3. Sök lediga bokningar
4. Ta bort bokning via namn
5. Visa alla bokningar
0. Avsluta Programmet
Skriv numret: 4
Alla bokningar:
Tid: 2025/11/26 10:00 Tjänst: Däckbyte Kund: Anna Svensson, Reg.Nr GIK299, Tel: 070-1234567
Tid: 2025/11/26 14:00 Tjänst: Balansering Kund: Erik Johansson, Reg.Nr XYZ123, Tel: 073-9876543
Tid: 2025/11/28 09:00 Tjänst: Däckhotell Kund: Lisa Karlsson, Reg.Nr ABC456, Tel: 072-5556667
Tid: 2025/11/29 12:00 Tjänst: Hjulinställning Kund: Jonas Lind, Reg.Nr DEF789, Tel: 076-8889990
Tid: 2026/01/14 10:00 Tjänst: Däckbyte Kund: Mikael Andersson, Reg.Nr MKA001, Tel: 070-1112233
Tid: 2026/01/14 14:00 Tjänst: Balansering Kund: Sofia Berg, Reg.Nr SOB002, Tel: 073-4445566
Tid: 2026/01/15 09:00 Tjänst: Däckhotell Kund: Peter Nilsson, Reg.Nr PEN003, Tel: 072-7778899
Vilken kunds bokning ska tas bort? (För och efternamn måste matcha)
|
  
```

Reflektion och slutsats

Arbetet gick smidigt där inga stora problem skedde. Bra kommunikation mellan kamraterna och effektivt arbete. Vi lärde oss att det fanns ett bra hjälpmedel som kan få igång arbete, Scrum. Om det var något vi skulle göra annorlunda är det att använda oss mer av Scrum för att få en tydligare bild på vårt mål och projekt..

Appendix

Parprogrammeringslogg

[illegible]