

# Bokningssystem för Skolans Lokaler

Skapad av: Angelica Bergström, Adam Axelsson-Hedman, Alex Bullerjahn & David Berglin

## Översikt

Vårt bokningssystem är designat för att hantera bokningar av skolans olika lokaler, inklusive både klassrum, salar och grupprum. Systemet tillåter användaren att skapa, boka, lista, uppdatera och ta bort bokningar. Programmet använder objektorienterade principer och moderna funktioner i C#, och säkerställer att data sparas mellan körningar genom att använda JSON-filer för persistens.

## Funktioner

- **Skapa rum:** Möjlighet att skapa nya rum med specifik information, såsom namn, typ, antal sittplatser och utrustning (t.ex. projektor eller whiteboard).
- **Boka rum:** Användare kan boka rum genom att ange start- och sluttid, samt specifik rumstyp och e-postadress. Systemet kontrollerar att bokningen inte krockar med andra bokningar.
- **Lista alla rum:** Visa alla rum och deras egenskaper, med möjlighet att sortera och filtrera efter olika kriterier.
- **Lista alla bokningar:** Visa alla bokningar, med möjlighet att filtrera efter år eller specifikt rum.
- **Uppdatera bokning:** Möjlighet att uppdatera en befintlig bokning genom att skriva över tidigare bokning.
- **Ta bort bokning:** Möjlighet att ta bort en bokning baserat på rum och e-postadress.
- **Avsluta programmet:** Programmet avslutas och sparar eventuella ändringar.

## Installation

1. Klon eller ladda ner projektet till din lokala maskin.
2. Öppna projektet i Visual Studio.
3. Starta programmet genom att köra projektet i Visual Studio.

## Användning

När programmet startas visas en huvudmeny med följande alternativ:

### Huvudmeny:

1. **Skapa ett rum**
  - Ange namn för rummet (om namnet redan finns, måste ett nytt namn anges).
  - Välj rumstyp (t.ex. grupprum, klassrum, sal).

- Ange antal sittplatser (t.ex. har grupprum max 15 platser, klassrum max 60 platser och sal max 120 platser).
  - För klassrum eller sal kan du även ange om rummet har projektor och whiteboard.
- 2. Boka rum**
- Välj rumstyp (grupprum, klassrum eller sal) att boka.
  - Ange e-postadress, startdatum och tid samt slutdatum och tid för bokningen.
  - Bokningen kan inte göras om start- eller sluttid redan har passerat, eller om bokningen överstiger 24 timmar.
  - Om bokningen krockar med en befintlig bokning, kommer systemet att visa information om krockade bokningar och ge användaren möjlighet att göra om bokningen.
- 3. Lista alla rum**
- Lista alla rum och deras egenskaper, med följande sorteringsalternativ:
    - Visa alla rum efter typ i alfabetisk ordning
    - Visa salar
    - Visa klassrum
    - Visa grupprum
    - Visa rum efter antal platser (fallande ordning)
    - Visa rum efter antal platser (stigande ordning)
    - Visa rum med projektor
    - Visa rum med whiteboard
    - Gå tillbaka
- 4. Lista alla bokningar**
- Visa alla bokningar, inklusive e-postadress, bokningstid och rumstyp.
  - Undermeny för att filtrera:
    - Visa bokningar för ett specifikt år.
    - Visa bokningar för ett specifikt rum.
- 5. Uppdatera bokning**
- Listar alla aktuella bokningar.
  - Välj vilken bokning som ska uppdateras.
  - Välj ett nytt rum till bokningen.
  - Skriv in din epost.
  - Ange de nya tiderna för bokningen.
- 6. Ta bort bokning**
- Välj rum.
  - Välj vilken bokning som ska tas bort.
- 7. Avsluta**
- Avsluta programmet.

## Kända Begränsningar

- När en bokning uppdateras genom menyalternativet "Uppdatera bokning", tas den gamla bokningen bort innan den nya bokningen sparas. Om du ångrar dig under uppdateringen försvinner den ursprungliga bokningen.
- RoomsConverter skulle kunna ändras så att IRoom-interface används istället för att använda Rooms-klassen

## Val och motiveringar för implementation

Programmet använder en **inputvalideringsmetod** för att säkerställa att användaren inte anger felaktiga eller ogiltiga data, vilket förhindrar att programmet kraschar eller får felaktiga resultat.

## Filformat och Struktur

Alla rum och bokningar lagras i JSON-filer för att säkerställa att data persisterar mellan körningar.

- 
- **BokningssystemGrupp6/**: Huvudmap med alla filer och dokument i projektet.
    - **BokningssystemGrupp6/**: Mapp med projektets kodfiler som används för att köra projektet.
      - **Classes/**: Mapp med klasser som utför olika funktioner i bokningssystemet
        - **LokalClasses/**: Mapp med klasser som utför saker som har med rum och lokaler att göra.
          - **Classroom.cs**: Innehåller kod för klassrum, ärver från Rooms-klassen.
          - **GroupRoom.cs**: Innehåller kod för grupprum, ärver från Rooms-klassen.
          - **Hall.cs**: Innehåller kod för salar, ärver från Rooms-klassen.
          - **Rooms.cs**: Bas-klass med egenskaper och metoder för rum (Rooms), så som att skapa rum, visa rum, etcetera. Rummens data sparas i en JSON-fil
          - **RoomsListAndSort.cs**: Kod för att sortera och visa rummen i olika ordningar, som antal platser, rummens egenskaper.
        - **Bookings.cs**: Innehåller metoder för att boka, ändra, ta bort och visa bokningar. Sparar bokningar i JSON-fil
        - **InputValidation.cs**: Metoder för inputvalidering, ex. kontroll av siffervärden, input format, etc.
        - **Menu.cs**: Huvudmenyn och den som först visas när man kör programmet och visar de funktioner som man kan utföra.
        - **RoomsConverter.cs**: Utför hanteringen av polymorf serialisering och deserialisering för att kunna skapa en lista där olika typer av underklasser där de ärver från en gemensam klass med där de kan innehålla olika mycket information. Den används för att både läsa och skriva till JSON-filerna.

- **Save.cs**: Hantering av att läsa och spara objektlistor (rum och bokningar) i JSON-filerna.
- **Interfaces**: Mapp för interface-filer.
  - **IListable.cs**: Interface som definierar metoder som används i flera klasser.
  - **IRoom.cs**: Interface som definierar egenskaper för rum.
- **BokningssystemGrupp6.csproj**: Projektfil för byggprocessen.
- **Programs.cs**: Startfil, skapar objekt och listor som används i de andra filerna. Deserialiserar JSON-filerna och startar sedan MainMenu metoden.
- **.gitignore**: Gitignore-fil med grundinställningar för Visual Studio-projekt.
- **BokningssystemGrupp6.sln**: Lösingsfil för hela C#-projektet.
- **README.md**: Projektbeskrivning och instruktioner för att köra programmet.

## JSON-filer

- **RoomList.json**: Rum och dess data.
  - **BookingList.json**: Bokningar och dess data.
- 

## Vilken student som haft huvudansvaret för vilka delar

Angelica Bergström, SCRUM-Master

Adam Axelsson-Hedman, Validater

Alex Bullerjahn, GitHub-Controller

David Berglin, DeBugger

Med mycket bra teamwork så har vi tacklat det mesta tillsammans.

## Tack och hej!

Tack för att du använder vårt bokningssystem! Vi hoppas att det kommer att vara användbart för att hantera skolans lokaler på ett effektivt och enkelt sätt.