# Rapport Laboration 3

Detta projekt är en bokningsapplikation för träningspass. Applikationen låter användare boka och avboka olika pass baserat på tid, kategori och namn. Målet med projektet är att skapa en applikation som är lätt att använda och utbyggbar för framtida ändringar.

**Layout**

Jag har designat användargränssnittet med hjälp av en Grid layout för att placera komponenter strukturerat. I layouten definieras antal rader och kolumner, och varje komponent placeras exakt där den behövs genom att specificera rad och kolumnindex. Detta ger en flexibel layout som är enkel att justera.

Layouten består av en ListView som visar alla tillgängliga träningspass, och information som datum, tid, antal platser och bokade platser. Till vänster finns en ComboBox där användaren kan välja att sortera passen efter olika kategorier, såsom namn, kategori och tid. Till höger finns en sökruta som uppdaterar listan dynamiskt baserat på användarens input.

Applikationen innehåller även två knappar, en boka-knapp som låter användaren boka ett pass och uppdaterar antalet bokade platser i listan, samt en avboka-knapp som tar bort bokningen.

**Kodstruktur**

Min kod är strukturerad i flera klasser för att hantera olika ansvarsområden:

* **Bokningshantering**: Ansvarar för att hantera alla pass och bokningar. Den innehåller en lista över alla träningspass samt metoder för att kontrollera och hantera användares bokningar.
* **Pass**: Håller reda på ett träningspass och hanterar information om passets namn, kategori, tid och antal platser. Den innehåller även metoder för att boka och avboka platser.
* **Användare**: Håller reda på en användare och hanterar användarens namn och bokade pass. Detta gör att man kan koppla bokningar till specifika användare.
* **Main.xaml**: Innehåller den grafiska biten av applikationen.
* **Main.xaml.cs**: Innehåller applikationens logik och hanterar interaktioner mellan gränssnitt och funktionalitet.

**Händelsestyrd Programmering**

Applikationen är byggd på händelsestyrd programmering, där olika händelser agerar dynamiskt beroende på användarens interaktioner. Till exempel:

* **TextChanged-händelsen** på sökrutan uppdaterar listan över pass dynamiskt när användaren skriver in text. Detta gör att listan filtreras direkt utan att användaren behöver trycka på en sökknapp.
* **Button\_Click-händelser** för Boka- och Avboka-knapparna kontrollerar om passet är fullbokat eller om användaren redan har bokat det valda passet. Om användaren har gjort en bokning eller avbokning uppdateras listan direkt.

**Skalbarhet**

Genom att separera logik och användargränssnitt, samt separera med klasser som ansvarar för specifika funktioner, är applikationen enkel att utöka. Om man till exempel vill lägga till fler typer av pass eller införa nya regler för bokningar, kan detta göras genom att utöka klasserna utan att påverka användargränssnittet.