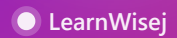


Navigazione riutilizzabile e permessi

Le quattro viste che costruirai, un unico shell che governa la navigazione e un primo assaggio dei permessi.



Obiettivo della lezione

Guarda l'obiettivo che costruirai nel lab: uno shell riutilizzabile la cui area contenuti passa tra quattro viste — Dashboard, Tickets, Customers e Settings — mentre intestazione, navigazione e barra di stato restano uguali.

1. Le quattro viste

Lo shell esterno non cambia mai. Ogni pulsante di navigazione carica un diverso **UserControl** nel pannello dei contenuti.

Vista	Che cosa mostra
DashboardView	Schede di riepilogo (Open Tickets, Customers, System Status) e un'azione rapida come <i>Run Health Check</i> .
TicketsView	Un elenco di ticket di assistenza con le azioni <i>New Ticket</i> e <i>Close Selected</i> .
CustomersView	Un elenco di clienti con le azioni <i>Add Customer</i> e <i>View Selected</i> .
SettingsView	Il ruolo corrente più un pulsante <i>Save Settings</i> che introduce il comportamento dei permessi.



Perché gli UserControl

Mettere ogni pagina nel proprio UserControl mantiene l'interfaccia ripetuta organizzata e riutilizzabile, e permette allo shell di restare piccolo — governa solo la navigazione e il pannello dei contenuti.

2. Un unico shell governa la navigazione

I nomi esatti possono cambiare, ma l'idea resta la stessa: lo shell crea la vista giusta, la mette in dock per riempire il pannello dei contenuti e aggiorna breadcrumb e stato.

```
// Lo shell principale governa i pulsanti di navigazione e il pannello dei contenuti.
public partial class MainPage : Page
{
    private string currentRole = "Support Agent";

    public MainPage()
    {
        InitializeComponent();
        NavigateTo("Dashboard");
    }
}
```

```

        ApplyPermissions();
    }

    private void NavigateTo(string pageName)
    {
        pnlContent.Controls.Clear();

        UserControl view;
        switch (pageName)
        {
            case "Tickets":    view = new TicketsView();    break;
            case "Customers":  view = new CustomersView(); break;
            case "Settings":   view = new SettingsView(currentRole); break;
            default:           view = new DashboardView(); break;
        }

        view.Dock = DockStyle.Fill;
        pnlContent.Controls.Add(view);
        lblBreadcrumb.Text = "Home / " + pageName;
        lblStatus.Text = DateTime.Now.ToString("hh:mm:ss tt") + " - Opened " + pageName + "
page.";
    }

    private void ApplyPermissions()
    {
        // Versione per chi inizia: un Support Agent può vedere Settings ma non salvare.
        // In seguito diventerà vera autenticazione e autorizzazione.
    }
}

```

3. Settings in sola lettura — il primo permesso

La vista Settings è il punto in cui i permessi diventano visibili. Per un **Support Agent**, mostra le impostazioni ma disabilita il salvataggio — il pulsante *Save Settings* è disattivato, con una nota come *"Settings are view-only for this role."* Un Manager potrebbe invece salvare. Così introduci i permessi **prima** che esista un sistema di login.

4. Checklist per il Designer

- Usa **Dock** o **Anchor** su intestazione, navigazione a sinistra, pannello dei contenuti e barra di stato.
- Rimuovi il pulsante di riduzione a icona da una Window/Form1, a meno che tu non costruisca una barra delle applicazioni o un'area di ripristino.
- Preferisci una **Page** quando lo shell deve riempire il browser come una normale app web.
- Tieni la logica di navigazione centralizzata nello shell invece di duplicarla in ogni pagina.
- Usa gli UserControl per i contenuti ripetuti — **DashboardView**, **TicketsView**, **CustomersView**, **SettingsView**.



Sei pronto per il lab

Prossimo passo: guarda la videoguia sullo shell di navigazione, supera la verifica delle conoscenze, poi costruisci lo shell di navigazione riutilizzabile — intestazione, quattro pulsanti di navigazione, un'area contenuti che cambia, un log di stato e Settings in sola lettura per un Support Agent.