

# PlusUI Framework - Feature Requests & Issues

Erstellt am: 2025-10-30

Aktualisiert am: 2025-10-30 (nach Theme System Update)

Basierend auf: Login Page Implementation Analysis

## ⚠ KRITISCHE FEATURES (P0 - Blocker)

### #1 Password Entry Control

Priorität: P0 - Critical Security Issue

Status: Missing

**Beschreibung:**

Entry Control hat keinen Password-Modus. Passwörter werden im Klartext angezeigt.

**Anforderung:**



csharp

`new Entry()`

```
.SetIsPassword(true) // Maskiert Text mit '•'  
.BindText(nameof(vm.Password), () => vm.Password, v => vm.Password = v)
```

**Alternative API:**



csharp

`new PasswordEntry() // Dediziertes Control`

```
.BindText(nameof(vm.Password), () => vm.Password, v => vm.Password = v)
```

**Details:**

- Maskierung mit konfigurierbarem Zeichen (default: '•')
- Optional: Toggle-Button für "Passwort anzeigen"
- Plattform-native Keyboard (Passwort-Modus)

### #2 Icon System

Priorität: P0 - Essential for Professional UI

Status: Missing

**Beschreibung:**

Keine Möglichkeit, Icons anzuzeigen. Aktuell nur Emojis als Workaround.

## Anforderung:

### Option A - Icon Control:



csharp

```
new Icon()
```

```
.SetIconSource("settings") // Material Icons  
.SetIconSize(24)  
.SetIconColor(SKColors.White)
```

### Option B - Button mit Icon:



csharp

```
new Button()
```

```
.SetIcon("settings")  
.SetIconPosition(IconPosition.Left) // Left, Right, Top, Bottom  
.SetText("Settings")
```

### Option C - Dediziertes IconButton:



csharp

```
new IconButton()
```

```
.SetIcon("settings")  
.SetCommand(vm.SettingsCommand)  
.SetIconSize(24)
```

## Icon Libraries:

- Material Icons (empfohlen)
- Font Awesome
- Custom SVG Support
- Platform-native Icons (SF Symbols iOS, Material Android)

## Details:

- SVG-Rendering mit SkiaSharp
- Color Tinting
- Size Presets: Small (16), Medium (24), Large (32)

# HIGH PRIORITY (P1 - Must Have)

## #3 Activity Indicator / Loading Spinner

**Priorität:** P1

**Status:** Missing

**Beschreibung:**

Kein Loading-Indikator für asynchrone Operationen.

**Anforderung:**



csharp

```
newActivityIndicator()
```

```
.BindIsRunning(nameof(vm.IsLoading), () => vm.IsLoading)
```

```
.SetColor(SKColors.Blue)
```

```
.SetSize(ActivityIndicatorSize.Medium) // Small, Medium, Large
```

**Alternative - Progress Bar:**



csharp

```
new ProgressBar()
```

```
.BindProgress(nameof(vm.Progress), () => vm.Progress) // 0.0 - 1.0
```

```
.SetColor(SKColors.Blue)
```

---

## #4 Entry - Keyboard & Input Configuration

**Priorität:** P1

**Status:** Missing

**Beschreibung:**

Entry Control hat keine Keyboard-Konfiguration für bessere UX.

**Anforderung:**



csharp

## new Entry()

```
.SetKeyboardType(KeyboardType.Email) // Email, Number, Phone, Url, Default  
.SetReturnKeyType(ReturnKeyType.Next) // Next, Done, Go, Search, Send  
.SetAutoCapitalization(AutoCapitalization.Sentences) // None, Words, Sentences, Characters  
.SetAutoCorrect(false)  
.SetMaxLength(50)  
.SetPlaceholder("Email eingeben")  
.SetPlaceholderColor(SKColors.Gray)
```

## Enums:



csharp

```
public enum KeyboardType { Default, Email, Number, Phone, Url, Decimal }  
public enum ReturnKeyType { Default, Done, Go, Next, Search, Send }  
public enum AutoCapitalization { None, Words, Sentences, Characters }
```

---

## #5 Focus Management

**Priorität:** P1

**Status:** Missing

**Beschreibung:**

Keine Möglichkeit, Focus programmatisch zu setzen.

**Anforderung:**



csharp

```

// Imperative API
public interface IFocusService
{
    void SetFocus(UiElement element);
    UiElement? CurrentFocus { get; }
}

// Usage
_focusService.SetFocus(usernameEntry);

// Oder deklarativ
new Entry()
    .SetAutoFocus(true) // Fokussiert beim Laden der Page
    .SetTabIndex(1) // Für Tastatur-Navigation

```

---

## #6 Text Wrapping & Truncation

**Priorität:** P1

**Status:** Undocumented

**Beschreibung:**

Unklar, ob/wie Text-Wrapping funktioniert. Benötigt für lange Texte.

**Anforderung:**



csharp

**new Label()**

```

    .SetText("Sehr langer Text der unterbrochen werden sollte...")
    .SetTextWrapping(TextWrapping.Wrap) // NoWrap, Wrap, WordWrap
    .SetMaxLines(3) // Optional: Begrenzung
    .SetTextTruncation(TextTruncation.End) // None, Start, Middle, End

```

---

## #7 Toolbar / AppBar Control

**Priorität:** P1

**Status:** Missing

**Beschreibung:**

Keine dedizierte Toolbar. Aktuell manuell mit HStack gebaut.

**Anforderung:**



csharp

```
new Toolbar()
    .SetTitle("Meine App")
    .AddLeftButton(new IconButton()
        .SetIcon("menu")
        .SetCommand(vm.OpenMenuCommand))
    .AddRightButton(new IconButton()
        .SetIcon("settings")
        .SetCommand(vm.SettingsCommand))
    .SetBackgroundColor(SKColors.Blue)
```

### Alternative - Simpler:



csharp

```
new AppBar()
    .SetTitle("Meine App")
    .SetNavigationIcon("menu", vm.MenuCommand)
    .AddAction("settings", vm.SettingsCommand)
    .AddAction("search", vm.SearchCommand)
```

---

## #8 Theme System - Erweiterungen

**Priorität:** P1 - Enhancement

**Status:** Implemented (Nice-to-have Enhancements)

**Beschreibung:**

Theme-System ist bereits implementiert und funktioniert!

**Bereits vorhanden:**

- IThemeService.SetTheme(Theme) / SetTheme(string)
- IThemeService.CurrentTheme
- style.AddStyle<T>(Theme.Light/Dark, ...) für theme-spezifische Styles
- Automatisches Re-Styling bei Theme-Wechsel
- Theme enum: Default, Light, Dark

**Nice-to-have Erweiterungen:**

**A. ToggleTheme() Helper**



csharp

```
public interface IThemeService
{
    string CurrentTheme { get; }
    bool SetTheme(Theme theme);
    bool SetTheme(string theme);

    // NEU (Helper für häufigen Use-Case):
    void ToggleTheme(); // Toggle zwischen Light ↔ Dark
}
```

### Aktueller Workaround (funktioniert):



csharp

[RelayCommand]

```
private void ToggleTheme()
{
    var newTheme = _themeService.CurrentTheme == "Light" ? Theme.Dark : Theme.Light;
    _themeService.SetTheme(newTheme);
}
```

### B. System Theme (Optional)



csharp

```
public enum Theme
{
    Default,
    Light,
    Dark,
    System // NEU: Folgt OS Dark Mode
}
```

### C. Persistence Helper (Optional)



csharp

```
public interface IThemeService
{
    // ... existing methods

    void EnablePersistence(string storageKey = "AppTheme");
    void LoadPersistedTheme();
}
```

**Hinweis:** Theme-System ist vollständig funktional. Diese Erweiterungen sind "Quality of Life"-Verbesserungen.

---

## 🟡 MEDIUM PRIORITY (P2 - Should Have)

### #9 Input Validation auf UI-Ebene

**Priorität:** P2

**Status:** Missing

**Beschreibung:**

Keine UI-seitige Input-Validierung. Alle Validierung im ViewModel.

**Anforderung:**



csharp

```
new Entry()
```

```
    .SetMaxLength(50)
    .SetAllowedCharacters("abcdefghijklmnopqrstuvwxyz0-9") // Whitelist
    .SetInputFilter(@"^@[a-z0-9]+$") // Regex
    .SetValidationRule(value => value.Contains("@"), "Muss @ enthalten")
```

---

### #10 IsEnabled Property für Controls

**Priorität:** P2

**Status:** Documented as Missing

**Beschreibung:**

Dokumentiert: "✖ IsEnabled property (use command CanExecute)"

Unklar wie visueller Disabled-State funktioniert.

**Anforderung:**



csharp

```
new Button()
    .SetText("Login")
    .SetCommand(vm.LoginCommand) // CanExecute sollte visuell reflektiert werden
    .SetIsEnabled(true) // Explizite Kontrolle
    .BindIsEnabled(nameof(vm.CanLogin), () => vm.CanLogin)
```

## Details:

- Disabled State sollte visuell erkennbar sein (z.B. Opacity 0.5, Grau)
- Command.CanExecute sollte automatisch IsEnabled beeinflussen
- Disabled Controls sollten keine Interaktion erlauben

## ● LOW PRIORITY (P3 - Nice to Have)

### #11 Custom Font Support

Priorität: P3

Status: Documented as Missing

Beschreibung:

Dokumentiert: "Font properties (uses default)"

Anforderung:



new Label()

```
.SetText("Heading")
    .SetFontFamily("Roboto")
    .SetFontWeight(FontWeight.Bold) // Thin, Light, Regular, Medium, Bold, Black
    .SetFontStyle(FontStyle.Italic)
```

Font Registration:



```
// In App.cs
public void ConfigureApp(HostApplicationBuilder builder)
{
    builder.RegisterFont("Roboto-Bold.ttf", "Roboto");
}
```

## #12 Shadow / Elevation Support

**Priorität:** P3

**Status:** Missing

**Beschreibung:**

Keine Shadow/Elevation für Material Design Look.

**Anforderung:**



csharp

```
new Border()
```

```
    .SetShadow(new Shadow
```

```
{
```

```
    Color = SKColors.Black.WithAlpha(50),
```

```
    Offset = new Point(0, 4),
```

```
    Blur = 8,
```

```
    Spread = 0
```

```
)
```

## #13 Gradient Backgrounds

**Priorität:** P3

**Status:** Missing

**Beschreibung:**

Nur solid colors, keine Gradienten.

**Anforderung:**



csharp

```
new VStack()
```

```
    .SetBackgroundGradient(new LinearGradient
```

```
{
```

```
    StartColor = SKColors.Blue,
```

```
    EndColor = SKColors.Purple,
```

```
    Angle = 45
```

```
)
```

*// Oder*

```
new RadialGradient(centerColor, edgeColor)
```

## #14 Placeholder für Entry

**Priorität:** P3

**Status:** Missing (Teil von #4)

**Beschreibung:**

Entry hat keinen Placeholder-Text.

**Anforderung:**



```
new Entry()
```

```
    .SetPlaceholder("Email eingeben...")
    .SetPlaceholderColor(SKColors.Gray)
```

**Hinweis:** Dies ist bereits Teil von Issue #4 (Entry Configuration).

---

## #15 Animation System

**Priorität:** P3

**Status:** Partially Documented

**Beschreibung:**

SetVisualOffset() dokumentiert, aber kein vollständiges Animation-System.

**Anforderung:**



```
element.Animate()
```

```
    .To(nameof(UiElement.VisualOffset), new Point(100, 0))
    .Duration(300)
    .Easing(Easing.EaseInOut)
    .Start();
```

```
// Oder deklarativ
```

```
new Label()
    .SetText("Fade In")
    .AnimateOnAppear(Animation.FadeIn(duration: 500))
```

---

# DOCUMENTATION IMPROVEMENTS

## #16 Image Resource Handling Documentation

Priorität: P2

Status: Undocumented

Was fehlt:

- Wie werden Images in Projekte eingebunden?
- Build Actions für verschiedene Plattformen
- Unterstützte Formate (PNG, JPG, SVG?)
- Namenskonventionen
- Pfad-Syntax (relative vs. absolute)

Benötigt:



markdown

### # Image Resources

#### ## Setup

1. Fügen Sie Bilder zum Projekt hinzu
2. Setzen Sie Build Action: 'EmbeddedResource'
3. Referenzieren Sie mit 'SetImageSource("image.png")'

#### ## Unterstützte Formate

- PNG (empfohlen)
- JPG
- SVG (falls unterstützt)

#### ## Plattformspezifische Ressourcen

- iOS: @2x, @3x Retina Support
- Android: hdpi, xhdpi, xxhdpi

---

## SUMMARY

### Kritische Blocker für Production (P0)

- #1 Password Entry Control
- #2 Icon System

### Must-Have für gute UX (P1)

- #3 Activity Indicator
- #4 Entry Keyboard Configuration
- #5 Focus Management

- #6 Text Wrapping
- #7 Toolbar Control
- #8 Theme System Enhancements (Nice-to-have, bereits funktional)

## Should-Have (P2)

- #9 Input Validation
- #10 IsEnabled Property

## Nice-to-Have (P3)

- #11 Custom Fonts
- #12 Shadow Support
- #13 Gradient Backgrounds
- #14 Placeholder (Teil von #4)
- #15 Animation System

## Documentation

- #16 Image Resource Handling
- 

# 🎯 EMPFOHLENE ROADMAP

## Phase 1 - Security & Core UX (Vor Release)

1. Password Entry (#1)
2. Icon System (#2)
3. Activity Indicator (#3)

## Phase 2 - Enhanced UX (Release 1.1)

4. Entry Improvements (#4)
5. Focus Management (#5)
6. Toolbar (#7)
7. Text Wrapping (#6)

## Phase 3 - Professional Features (Release 1.2)

8. Theme Enhancements (#8)
9. IsEnabled State (#10)
10. Documentation (#16)

## Phase 4 - Polish (Release 2.0)

11. Custom Fonts (#11)
  12. Shadows & Gradients (#12-13)
  13. Animation System (#15)
-

# BEREITS IMPLEMENTIERT

## Theme System (Komplett)

-  IThemeService mit SetTheme(Theme) und SetTheme(string)
-  Theme enum: Default, Light, Dark
-  style.AddStyle<T>(Theme, ...) für theme-spezifische Styles
-  Automatisches Re-Styling bei Theme-Wechsel
-  Custom Theme Namen Support

**Hervorragende Implementierung!** Nur kleine "Quality of Life"-Enhancements offen (siehe #8).

---

## ENTFERNT FEATURES

Die folgenden Features aus der ursprünglichen Liste wurden nach Review entfernt:

- **Spacer Control** - Grid + Alignment ist die richtige Lösung (Best Practice)
- **Image Sizing Improvements** - Aktuelle API ist ausreichend
- **Responsive Design Utilities** - Auto-Relayout bereits vorhanden
- **Accessibility** - Zu früh, erst nach Core-Features
- **Badge/Tooltip** - Nice-to-have, nicht essential
- **Binding Performance Docs** - Nicht kritisch
- **Custom Control Guide** - Später in erweiteter Doku
- **Error Handling in Bindings** - Framework-interne Optimierung

**Grund für Entfernung:** Fokus auf essenzielle Features, die real blocking sind für Production-Apps.

---

## Ende der Feature-Liste

*Viel Erfolg bei der Implementierung!* 

**Total Issues:** 16 (reduziert von 25)

**P0:** 2 | **P1:** 6 | **P2:** 2 | **P3:** 5 | **Docs:** 1