

Theme

Overview

This file defines the app's theme using Jetpack Compose's Material3 theming system. It provides two color schemes - a light theme and a dark theme - and applies them based on the user's system settings or preferences.

Code Explanation

```
package com.example.vociapp.ui.theme

// ... IMPORTS

private val DarkColorScheme = darkColorScheme(
    primary = ColorPalette.PrimaryDark,
    onPrimary = ColorPalette.OnPrimaryDark,
    primaryContainer = ColorPalette.PrimaryContainerDark,
    onPrimaryContainer = ColorPalette.OnPrimaryContainerDark,
    inversePrimary = ColorPalette.InversePrimaryDark,
    secondary = ColorPalette.SecondaryDark,
    // ... (Other color properties)
)

private val LightColorScheme = lightColorScheme(
    primary = ColorPalette.PrimaryLight,
    onPrimary = ColorPalette.OnPrimaryLight,
    primaryContainer = ColorPalette.PrimaryContainerLight,
    onPrimaryContainer = ColorPalette.OnPrimaryContainerLight,
    inversePrimary = ColorPalette.InversePrimaryLight,
    secondary = ColorPalette.SecondaryLight,
    // ... (Other color properties)
)

@Composable
fun VociAppTheme(
    darkTheme: Boolean = isSystemInDarkTheme(),
    // Dynamic color is available on Android 12+
    dynamicColor: Boolean = true,
    content: @Composable () -> Unit
) {
    val colorScheme = when {
        dynamicColor && Build.VERSION.SDK_INT >= Build.VERSION_CODES.S -> {
            val context = LocalContext.current
            if (darkTheme) dynamicDarkColorScheme(context) else
            dynamicLightColorScheme(context)
        }
        darkTheme -> DarkColorScheme
        else -> LightColorScheme
    }
```

```

    }

    val view = LocalView.current
    if (!view.isInEditMode) {
        SideEffect {
            val window = (view.context as Activity).window
            window.statusBarColor = colorScheme.primary.toArgb()
            WindowCompat.getInsetsController(window,
view).isAppearanceLightStatusBars = darkTheme
        }
    }

    MaterialTheme(
        colorScheme = colorScheme,
        typography = Typography,
        content = content
    )
}

```

Breakdown:

1. Color Schemes:

- `DarkColorScheme` and `LightColorScheme` are defined using `darkColorScheme()` and `lightColorScheme()` from Material3.
- They utilize color values from the `ColorPalette` object, which is defined in a separate file (`Colors.kt`). See that file for detailed color definitions.

2. VociAppTheme Composable:

- This is the main composable function that applies the theme to your app's content.
- It takes `darkTheme` and `dynamicColor` as parameters to control theme selection.
- It uses `isSystemInDarkTheme()` to determine the user's system preference for dark mode.
- It uses `dynamicDarkColorScheme()` and `dynamicLightColorScheme()` (if available on Android 12+) to create dynamic color schemes based on the user's wallpaper.
- It applies the selected color scheme using `MaterialTheme`.

3. Window/Status Bar:

- This sets the window's status bar color to the primary color and controls the status bar's light/dark appearance.

Related Files

- **Colors.kt:** Defines the color palette used in the theme.
- **Typography.kt:** Defines the typography styles used in the theme (if applicable).

Additional Notes

- This file defines the overall structure and logic for applying the app's theme.
- It relies on the `ColorPalette` object in `Colors.kt` for specific color values.
- It uses Material3 theming components for consistency with Material Design guidelines.

- The `dynamicColor` option allows for dynamic color adaptation based on the user's wallpaper (Android 12+).