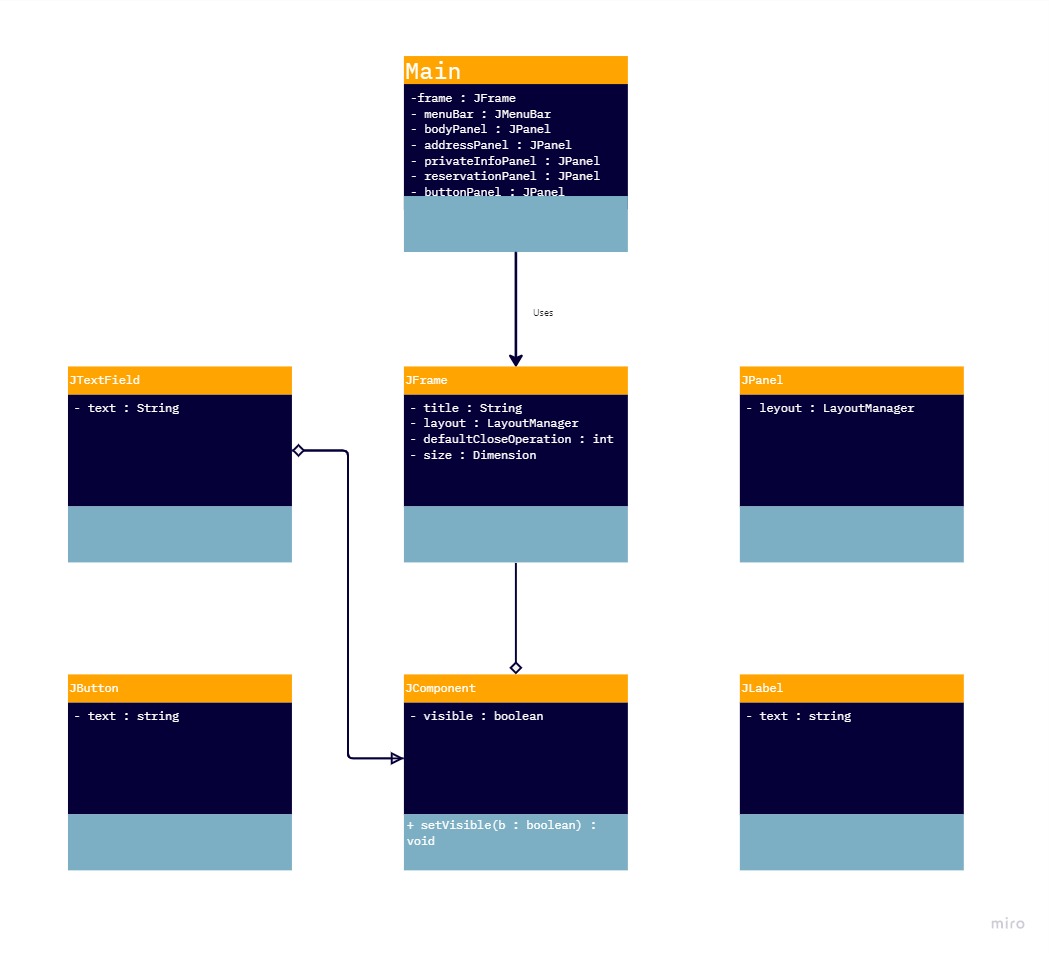
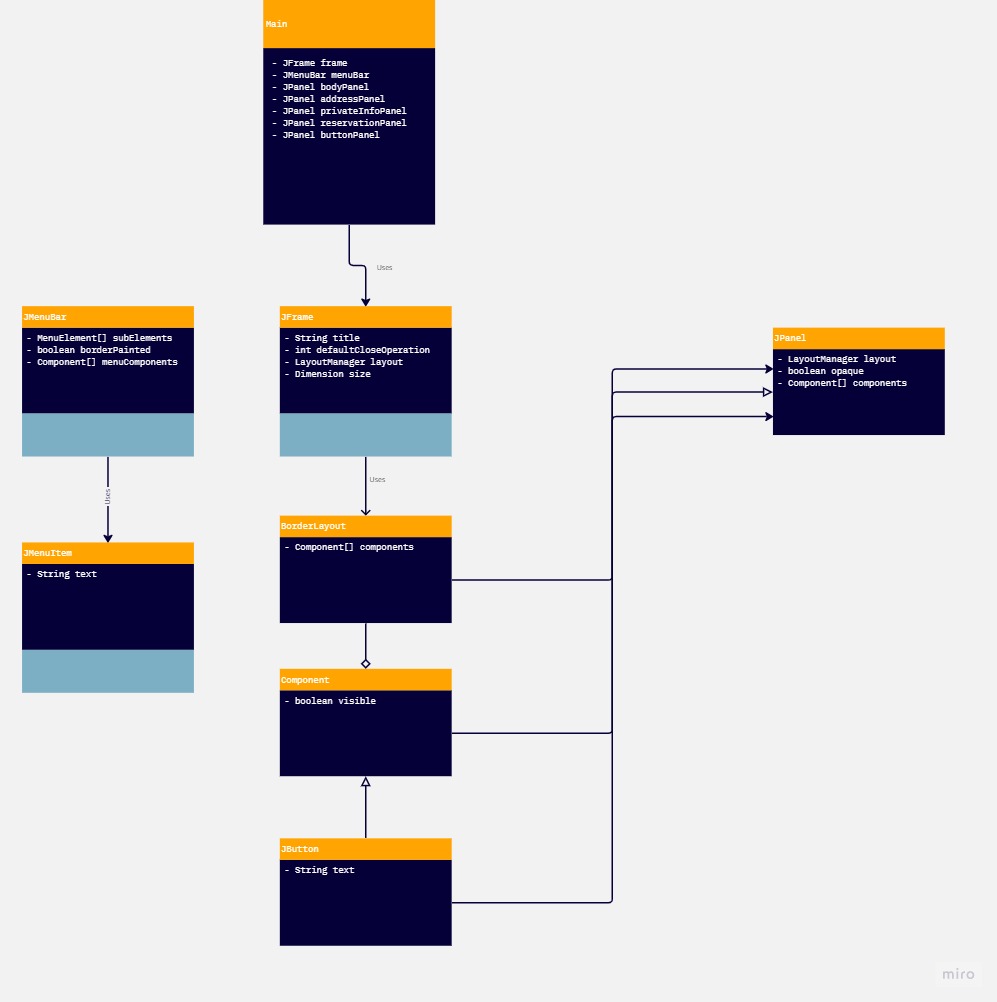
**Documento de diseño:**





1. **Uso de Swing para la interfaz gráfica de usuario (GUI):** Swing es una biblioteca GUI de Java bastante antigua pero aún muy utilizada debido a su madurez y amplia documentación. Proporciona una amplia gama de componentes de interfaz de usuario y es relativamente fácil de usar para construir interfaces de usuario complejas.
2. **Uso de una arquitectura centrada en vistas:** Cada clase representa una "vista" o pantalla en la aplicación. Esto es un enfoque común en la programación de GUI, ya que permite separar la lógica de diferentes partes de la aplicación y hacer que el código sea más fácil de mantener.
3. **Uso de JPanel para agrupar componentes:** Cada método **createXPanel** en **EmployeeView** crea un **JPanel** que agrupa varios componentes relacionados. Esto facilita la disposición de los componentes y hace que el código sea más fácil de entender y mantener.
4. **Uso de BorderLayout y GridLayout para la disposición de los componentes:** Estos son gestores de diseño comunes en Swing que proporcionan un control suficiente para la mayoría de las necesidades de disposición de los componentes.
5. **Estilo global con UIManager:** En **applyGlobalStyles**, el código usa **UIManager** para establecer estilos globales para varios tipos de componentes. Esto asegura que todos los componentes de un tipo tengan el mismo aspecto, proporcionando una apariencia coherente en toda la aplicación.
6. **Uso de la autenticación basada en texto plano:** En el método **authenticate** de **LoginScreen**, se realiza una autenticación muy simple comparando directamente los nombres de usuario y las contraseñas con cadenas de texto plano. Esto es muy probablemente solo para fines de demostración, ya que es una práctica muy insegura en una aplicación real. En una aplicación real, se debería autenticar a los usuarios contra una base de datos y las contraseñas deberían almacenarse de forma segura, por ejemplo, como hashes.
7. **Uso de SwingUtilities.invokeLater:** Este método se utiliza para asegurar que la GUI se crea y se actualiza en el hilo de despacho de eventos (EDT) de Swing, que es el hilo en el que se deben realizar todas las operaciones de la GUI en Swing. Esto puede prevenir varios errores y problemas de rendimiento.

Diagrama

Descripción generada automáticamente