Description of Project

The program consists of Login Page ,Registration Page , Transfer Page and a page to view your Account .

The project entails creating a mobile banking app for Android customers for Sisonke Bank using Xamarin.Forms. It has pages for checking account balances, moving money, registering, logging in, and logging out. For user authentication and account management, the app interfaces with a SQLite database, guaranteeing a safe and efficient banking experience. User Opens the App and is introduced to the Login Page and the user can click on the Register Link and will be sent to the Registration Page . The user will then put in the necessary details such as First name ,last name ,email ,phone number and a specific gender .

Error Message

If an Error message occurs the user can click okay to exit the error message

The banking app for Xamarin.Forms handles error signals by employing the DisplayAlert function to show pertinent alerts. Validation checks make sure users submit the necessary data accurately. Error messages alert users to wrong inputs and provide instructions on how to fix them before continuing, making the user experience seamless and intuitive.

Visual Studio 2022 Resource was used

How to set up the App

1. **Install Visual Studio**: Download and install Visual Studio, ensuring you have the necessary components for Xamarin.Forms development.
2. **Clone or Download the Project**: Obtain the project files either by cloning the repository or downloading the project ZIP file from the source provided.
3. **Open the Project in Visual Studio**: Launch Visual Studio and open the downloaded project by selecting "Open a project or solution" from the start page or the File menu.
4. **Set up Emulator or Device**: Set up an Android emulator (if using Windows PC) or iOS emulator (if using a Mac) for testing. Alternatively, you can connect a physical Android or iPhone device for testing.
5. **Restore NuGet Packages**: Right-click on the solution in Visual Studio's Solution Explorer and select "Restore NuGet Packages" to ensure all required packages are installed.
6. **Configure Database Connection**: If using a SQLite ensure the database connection is correctly configured in the app. Update connection strings or authentication details as needed.
7. **Build and Run the App**: Build the solution by selecting "Build" from the Build menu. Once built successfully, select your desired emulator or device from the debugging dropdown menu and click "Start Debugging" (or press F5) to run the app.

User Docmentation

1. **LoginUI**: The is a logo that is displayed and it a logo of a bank. This page allows users to log in with their email and password. It validates the input and navigates to the mainapge upon successful authentication. If the user doesn't have an account, they can navigate to the registration page using the link “Register “
2. **RegistrationPage**: Here, users can register by providing their first name, last name, email, password, mobile number, and gender. It performs basic validation on the input fields and displays appropriate error messages and when the user sees the error, she can click OK and input the correct information . Upon successful registration and a message will be displayed, users are navigated back to the login page.
3. **MainPage**: The user is greeted by a welcome message After logging in, users are directed to the main page, where they can view their account balances, transfer funds between accounts, and log out. The user's first name is dynamically displayed on this page.
4. **TransferBetweenAccountsPage**: Users can initiate transfers between their current and savings accounts from this page. It verifies the input amount and transfer direction and displays a success message upon completion.
5. **ViewAccountBalancePage**: This page displays the current and savings account balances. Users can navigate back to the main page from here.