

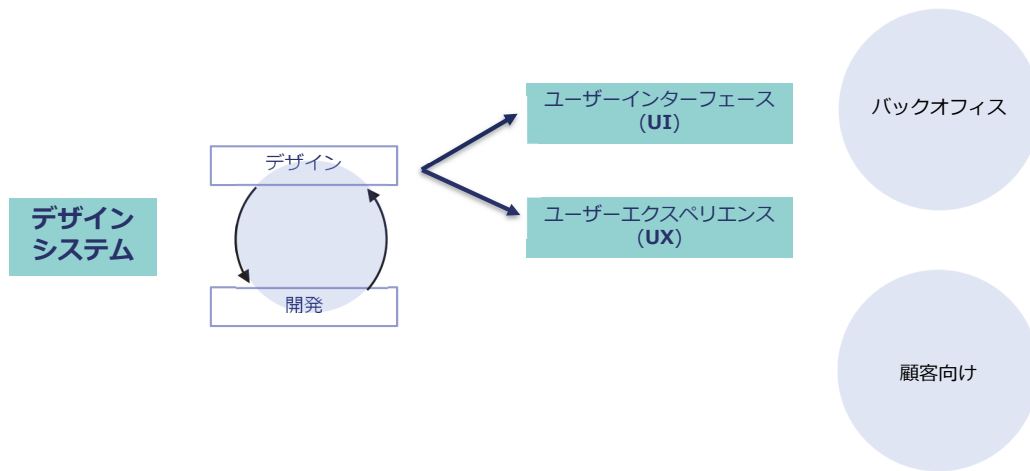
顧客向けに重点を置いた Web 画面

デザインシステム

*GeneXus*TM

これまでの、パネルの機能面に注目し、デザイン面については検討していませんでした。

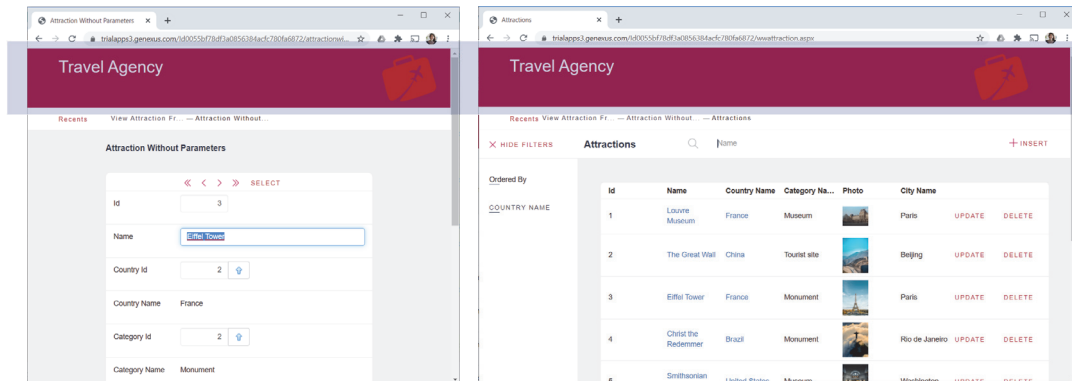
ここでは、アプリケーションにデザインを組み込む方法を確認します。



以前の章で、バックオフィスアプリケーションおよび顧客向けアプリケーションの両方にデザインとユーザービリティの要件があり、それによってデザインシステムの概念が導入されたことを確認しました。

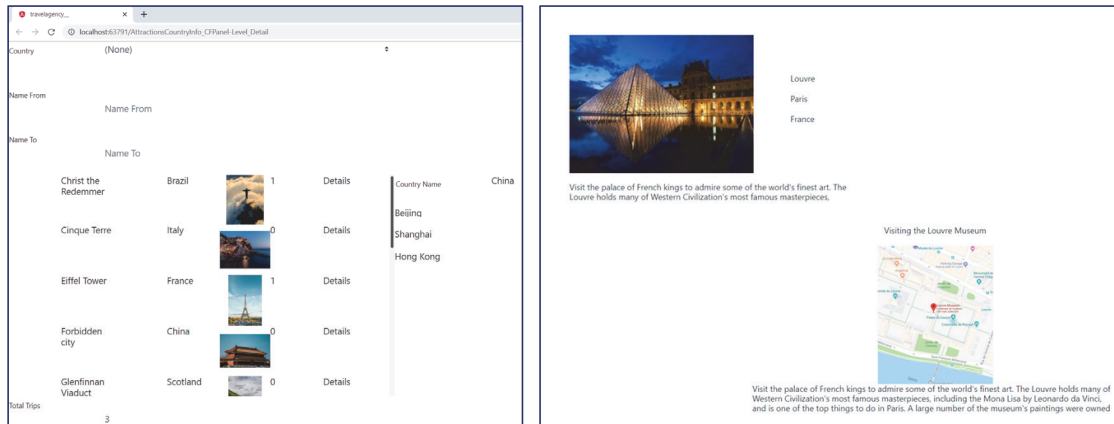
デザインシステムには、アプリケーションの一貫性、統一性、堅牢性を表す一連の原則、パターン、慣習が含まれます。これによって、開発者およびデザイナーの両者を含む開発サイクルが定義されます。

Web アプリケーションの事前定義済みデザインシステム



また、バックオフィスアプリケーションの一部であるトランザクションおよび Web パネルには、事前定義済みデザインシステムがあることも確認しました。

顧客向けアプリケーションの事前定義済みデザインシステム

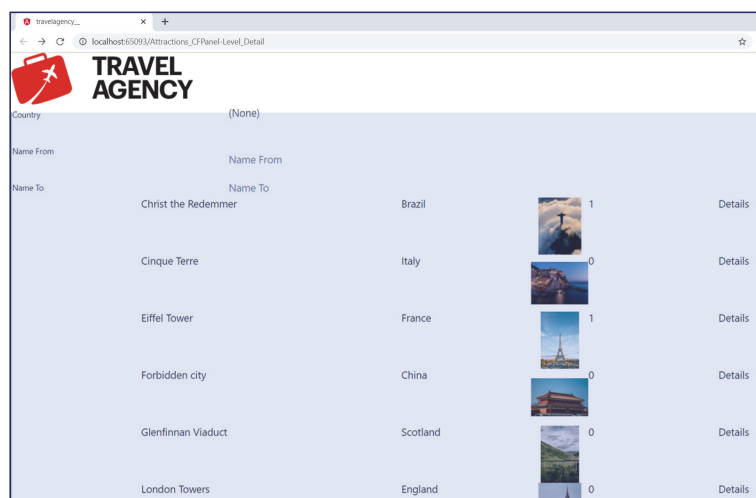


バックオフィスアプリケーションの場合と同様に、顧客向けアプリケーションで使用するオブジェクトにも事前定義済みデザインシステムがあります。

既定でマスターページが割り当てられる Web Panel および Transaction オブジェクトとは対照的に、フロントエンドの顧客向けアプリケーションのオブジェクトは、既定では Master Panel オブジェクトは割り当てられません。Master Panel オブジェクトは、アプリケーションを実行するためのコンテナとなり、実行のための基本的なデザインコンポーネントを含むものです。

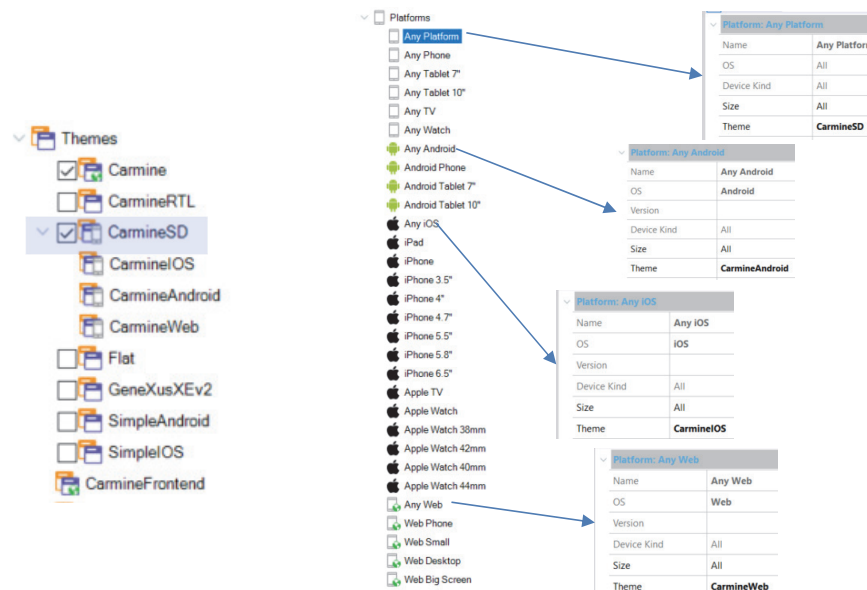
それでも、コントロールはデザインに従い、見た目は一貫性があるものになります。これらすべてが事前定義された形で行われます。また、Master Panel オブジェクトをアプリケーションのパネルに割り当てることができるということも既に確認しました。

顧客向けアプリケーションの事前定義済みデザインシステム



コントロールはデザインに従い、見た目は一貫性があるものになります。これらすべてが事前定義された形で行われます。また、Master Panel オブジェクトをアプリケーションのパネルに割り当てることができるということも既に確認しました。

顧客向けアプリケーションの Theme オブジェクト



画面上のコントロールの見た目は、CarmineSD という名前の Theme オブジェクト向けの定義に基づきます。

そして、選択したプラットフォームに応じたサブテーマの 1 つを使用します。たとえば、Android で生成する場合は CarmineAndroid、Apple の場合は CarmineIOS、Angular の場合は CarmineWeb テーマの定義を使用します。

これを KB エクスプローラーで [Platforms] ノードを開いて確認することもできます。プラットフォームごとに [Theme] プロパティが表示されます。たとえば、[Any Platform] には CarmineSD テーマが割り当てられています。[Any Android] には CarmineAndroid テーマ、[Any iOS] には CarmineIOS テーマ、[Any Web] には CarmineWeb テーマが割り当てられています。



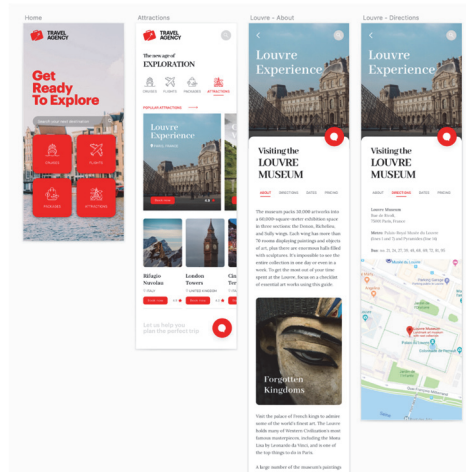
デザインシステムの一部として挙げたコンポーネントに関して言えば、顧客向けアプリケーションには、Master Panel、Theme オブジェクト、Theme クラスもあります。レスポンシブテーブル専用のコントロールはありません。これは、Web アプリケーションを Angular で生成する場合、テーブルコントロールでコンテンツを自動的に調整できるからです。

Web Panel オブジェクトを生成する Work With for Web パターンとは異なり、顧客向けオブジェクト用の Work With パターンは、Panel オブジェクトではなく、WorkWithDevices と呼ばれる専用の 1 つのオブジェクトを生成します。

デザインシステムのほかのすべてのコンポーネントが表示されていますが、各コンポーネントの実装はプラットフォーム (Web またはネイティブ) および使用するジェネレーター (Angular、Android、Apple) によって異なります。

バックオフィスアプリケーション向けのデザインのカスタマイズの場合と同様に、顧客向けアプリケーションでも、デザインシステムのさまざまな要素 (クラス、テーマ、コントロールなど) を使用して、できる範囲で最高のユーザーエクスペリエンスを提供するデザインを実現できます。

Sketch からのデザインのインポート

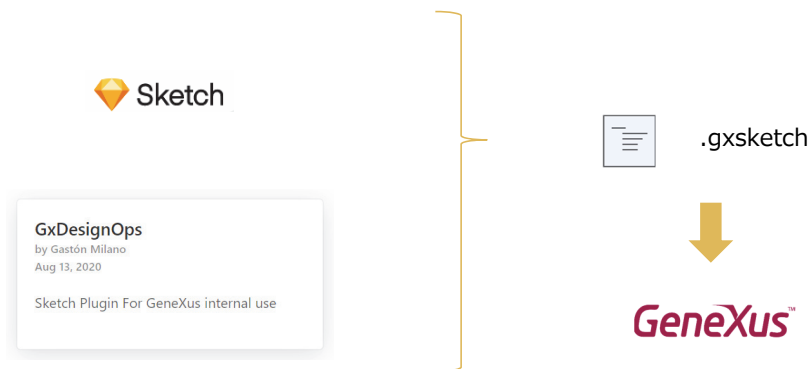


ただし、デザイナーがデザインを行うためのシンプルかつ最高の方法は、デザイン全体を定義してからナレッジベースにインポートすることです。

アプリケーションのさまざまなパーツに対応するボタンがある初期ページがほしくなり、観光名所に関連するデータだけを表示する画面のデザインも必要になるはずです。訪問可能な観光名所の一覧から、ボタンでクリックされると、その観光名所の詳細と地図のタブを持つ画面に遷移されます。

まずモバイルデバイス向けのデザインを送信するようデザイナーに要求した場合、インポートで作成される同じオブジェクトを使用して、後からそのアプリケーションを Angular で生成できます。

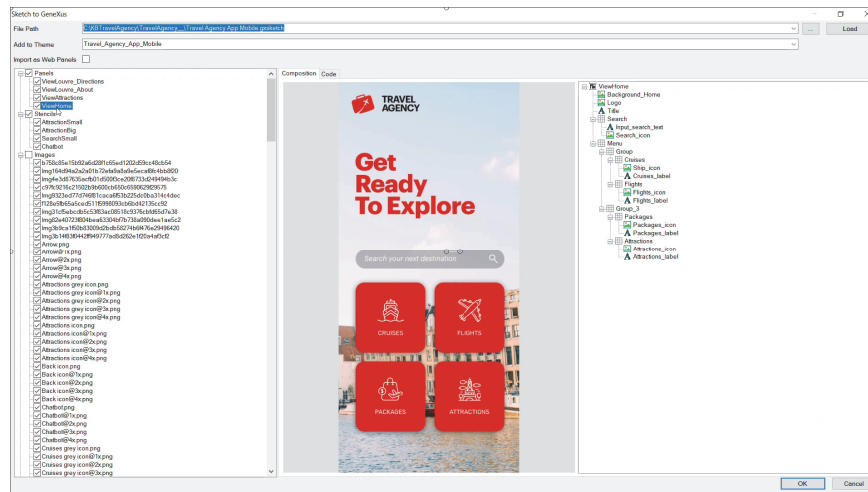
Sketch からのインポート



デザイナーは要件に応じて Sketch でデザインを作成し、Sketch にインストールした GeneXus プラグインを使って拡張子 gxsketch のファイルを作成し、送信してきます。

そのファイルをナレッジベースにインポートして、すべてのデザインコンポーネントと統合する必要があります。また、定義を格納するために、対応する GeneXus オブジェクトを作成する必要があります。

.sketch ファイルのインポート



デザイナーから受け取った .sketch ファイルをインポートするには、[ツール] > [アプリケーションの統合] から [Sketch インポート] を選択します。

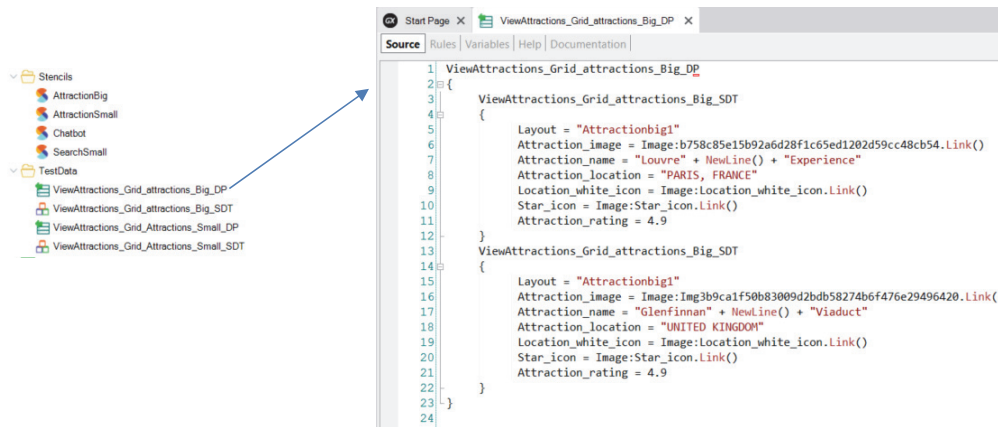
Web パネルではなく Panel オブジェクトを作成する必要があるので、[Web パネルとしてインポート] のチェックボックスの選択を解除します。

インポートによって作成されたパネルをクリックすると、プレビューが表示されます。ここで、検証したデザインに対応したものになっていることを確認します。右側には、パネルに含まれるコントロールの情報が表示されます。

また、固定データがあるパネルを実行して、ロードされた観光名所を表示するために、画像をインポートすることも確認できます。このデータは、アプリケーションのデータベース内に保存されたもので後から置き換える必要があります。

下の方では、デザインで使用するフォントを確認できます。問題がなければ [OK] をクリックします。

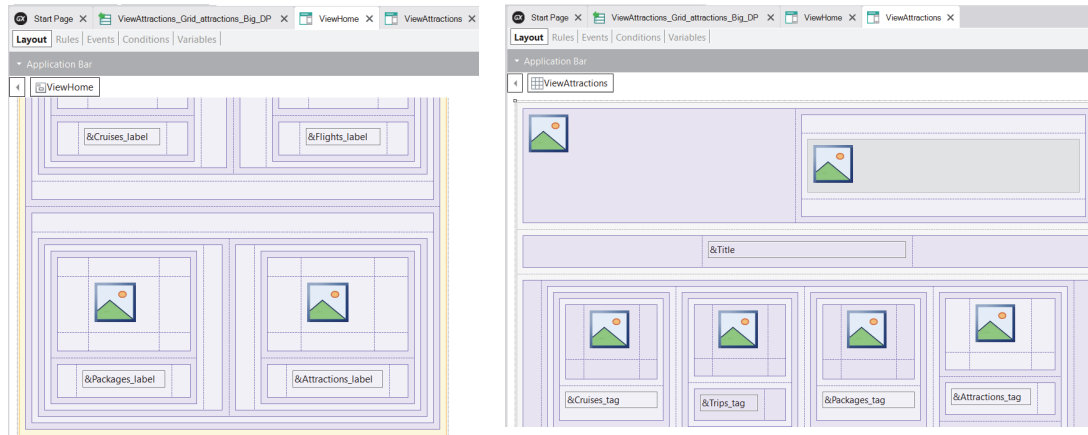
インポートで作成されるオブジェクト



[出力] ウィンドウにインポートの進捗と結果が表示されます。エラーはありません。2つのフォルダが作成されたことを KB エクスプローラーで確認できます。1つは Stencils です。このフォルダには、アプリケーションでデザインをカプセル化するために使用される Stencil が格納されています。もう1つは TestData です。このフォルダには、固定データをロードするためのデータプロバイダーおよび SDT が含まれています。

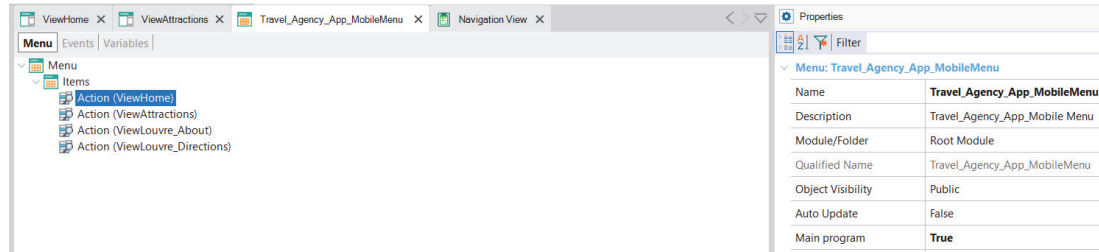
この時点で1つ目のデータプロバイダーを開くと、画像をロードするコードが表示されます。

インポートで自動的に作成される Panel オブジェクト



インポートウィザードの画面で確認した Panel オブジェクトは自動的に作成されました。また、最初のページの Panel オブジェクト (ViewHome) と、ViewAttractions および ViewAttractions から呼び出される Panel オブジェクト (ViewLouvre_About および ViewLouvre_Directions) も作成されました。ViewHome のフォームを確認すると、視覚的なコンテンツに使用するすべてのコンポーネントが作成されていることが分かります。ViewAttractions についても同様です。

アプリケーションのメインオブジェクト

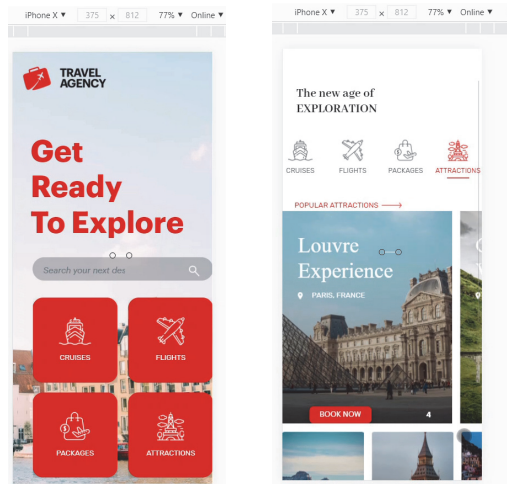


ここでは、Menu オブジェクトも作成されたことが分かります。これが実行するメインオブジェクトになります。

このオブジェクトは、ネイティブプラットフォーム向けに生成した場合にのみ表示されます。Angular の場合、最初に表示されるオブジェクトは、メニューの最初の項目である ViewHome です。

実行するために、このオブジェクトを開始オブジェクトとして設定し、F5 キーを押します。

固定データがあるアプリケーションの実行環境

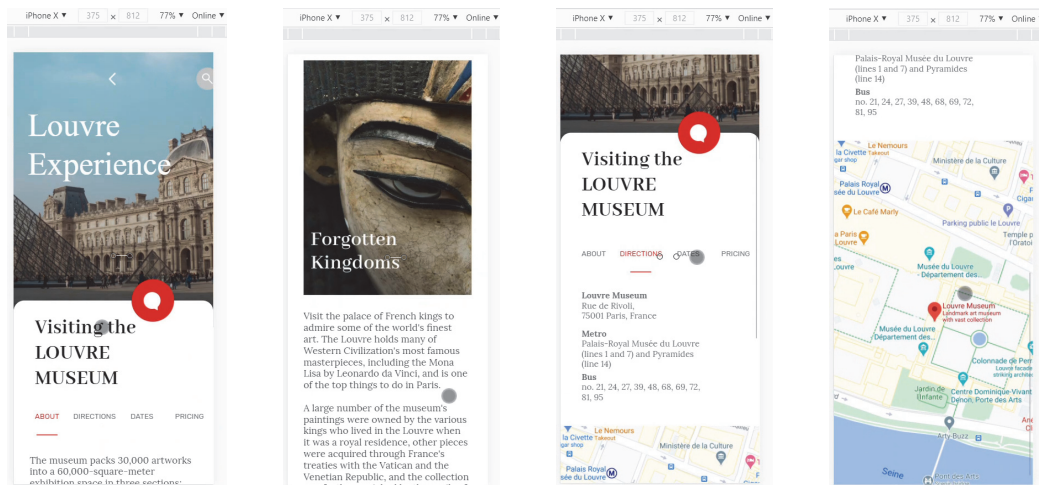


デザインは元々モバイルプラットフォーム向けのものなので、iPhone X の画面サイズを選択します。デスクトップ画面で表示するために必要な変更を後で加える必要があります。

最初の画面が実行され、アプリケーションのさまざまなパーツにつながるボタンが表示されます。[ATTRACTIONS] をクリックすると、人気の観光名所のリストが表示され、その下にはほかの観光名所が表示されます。

前にも述べたとおり、インポートのデータプロバイダーにより、固定データがロードされています。

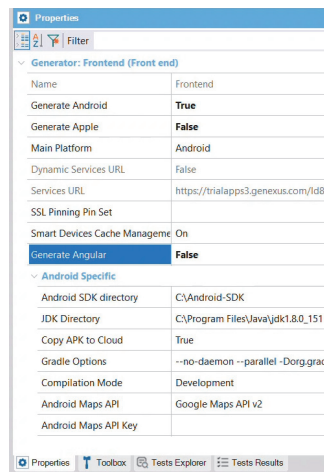
固定データがあるアプリケーションの実行環境 (続き)



ルーブル美術館をクリックすると、詳細情報のページが表示されます。美しく、整理されたデザインになっています。

[DIRECTIONS] をクリックすると、パネルが開いて、ルーブル美術館の住所と地図が詳しく表示されます。

Android での顧客向けアプリケーションの作成と実行



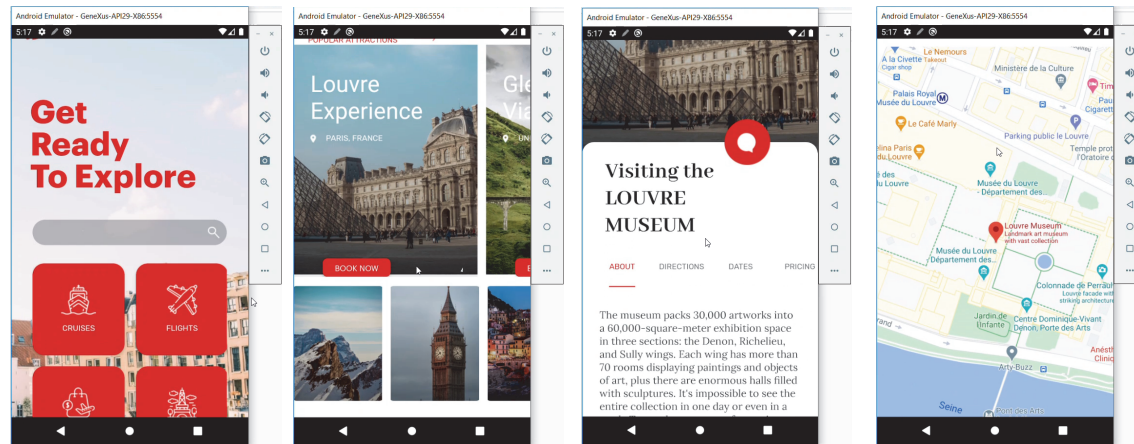
次に、アプリケーションを Android で生成して、モバイルプラットフォームで新しいデザインがどのように表示されるかを確認します。

そのために、フロントエンドのプロパティに移動し、[Generate Android] プロパティを True に設定します。

ここでは Android モバイルアプリケーションのみを確認するので、[Generate Apple] プロパティおよび [Generate Angular] プロパティを False に設定します。

Menu オブジェクトは既に開始オブジェクトになっているので、このまま F5 キーを押します。

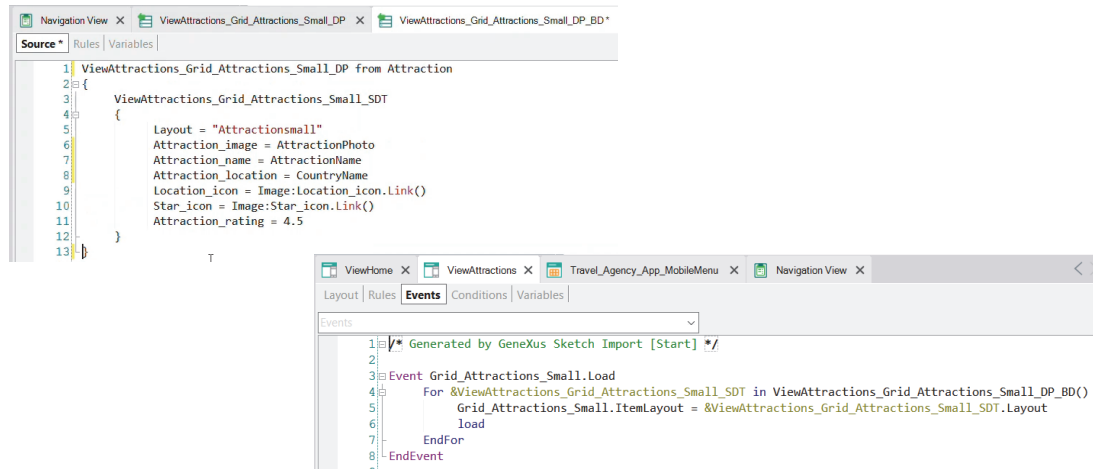
Android での顧客向けアプリケーションの作成と実行



エミュレーターが開き、アプリケーションが Android で実行されます。見た目はデザイナーが定義したデザインシステムに従っています。

この例では、顧客向けアプリケーションの開発プロセスを確認しました。また、さまざまな人々が最高のソリューションのために貢献するチームワークの重要性を説明しました。

データベースのデータにアクセスするための新規オブジェクトの変更



ここまでは、テスト用の固定データを見てきました。これは、新しいデザインでのアプリケーションの動作を示すために、デザイナーが追加したものでした。次に、変更を加えて、データベースの実際のデータを見てみます。

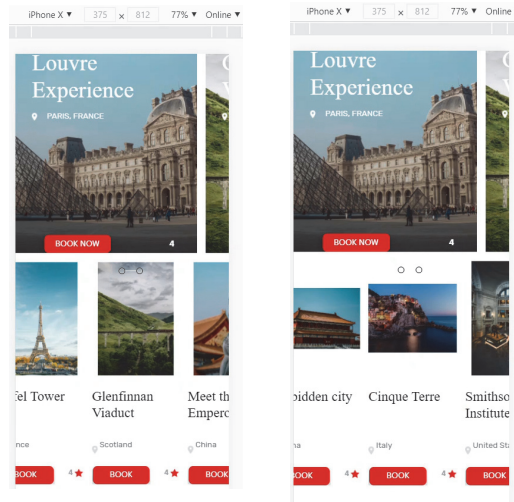
ViewAttractions パネルでは、SDT タイプの変数、ViewAttractions_Grid_Attractions_Small_SDT にロードされた観光名所がグリッドに表示されます。パネルのイベントでは、データプロバイダーによって観光名所がロードされていることが分かります。そのデータプロバイダーを開いて、名前を付けて保存し、ViewAttractions_Grid_Attractions_Small_DP_BD という名前にします。

from Attractions 節を追加し、Attraction_image フィールドに AttractionPhoto 項目属性の値を割り当てます。Attraction_name フィールドには AttractionName 項目属性、Attraction_location フィールドには CountryName 項目属性を割り当てます。それから、不要なものをすべて削除します。

ViewAttractions パネルに戻り、観光名所をロードするデータプロバイダーを変更して、作成したばかりの新しいものにします。

保存して、Angular での生成を再度有効にします。メインオブジェクトを右クリックして、[実行] を選択します。

実際のデータでのアプリケーションの実行

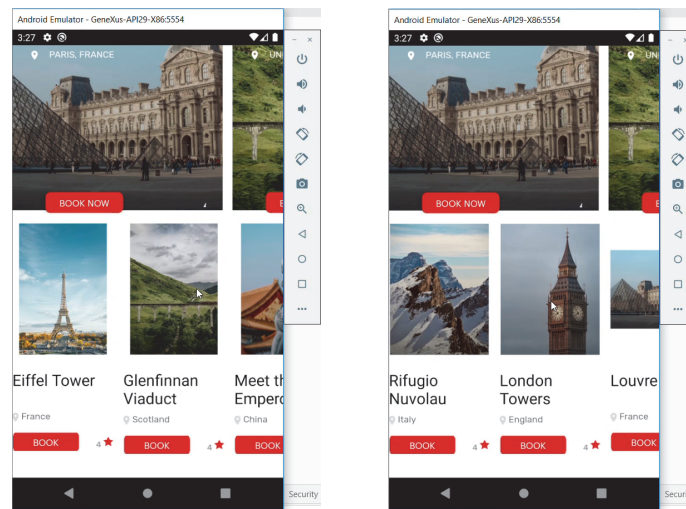


観光名所に移動すると、ロードされた観光名所が表示されます。アプリケーションのデザインは、より適切なものになっています。

チームにデザイナーがいる場合は、クラスやプロパティを手動で変更することはありません。それらの定義を Sketch からインポートするほうがプロフェッショナルらしいデザインのやり方だからです。こうすれば、アプリケーションの機能面に集中することができます。

最後に Android で生成し、データベースのデータをモバイルデバイスで表示する、最終的な成果を確認しましょう。

実際のデータでのアプリケーションの実行



Android でも、データベースにあるすべての観光名所が新しいデザインで表示されます。

このように、プロフェッショナルによるデザインを GeneXus プロジェクトに統合することには、いくつかのメリットがあります。作業が最適化され、チームの各メンバーが、各自の最も得意なことで貢献できるからです。

デザインが開発のアクティビティの一部となる作業の進め方は、DevOps の方法論で適用されます。この点については、このコースの以降の章で説明します。