

2.1 Generic UI Development

A good User Interface (UI) focuses on making user's interactions simple and efficient. User would appreciate a website with intuitive user interface that leads them towards their task in most engaging way. User Interface (UI) design focuses on thinking of a user, what they might need to do when they visit website and ensure that the interface has elements that are easy to access and understand. Being a UI designer, one need to understand the goals, skills, preferences and tendencies of the user to make a better interface.

The Generic User Interface (Generic UI, GUI) framework allows you to create UI screens using Java and XML. XML is optional but it provides a declarative approach to the screen layout and reduces the amount of code which is required for building the user interface.

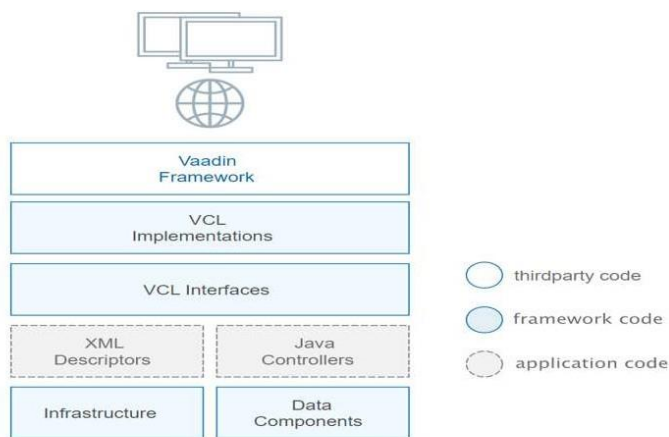


Figure 18: The Structure of Generic User Interface

The application screens consist of the following parts:

- **Descriptors** – XML files for declarative definition of the screen layout and data components.
- **Controllers** – Java classes for handling events generated by the screen and its UI controls and for programmatic manipulation with the screen components.

The code of application screens interacts with visual component interfaces (VCL Interfaces). These interfaces are implemented using the Vaadin framework components.

- **Visual Components Library (VCL)** contains a large set of ready-to-use components.

- Data components provide a unified interface for binding visual components to entities and for working with entities in screen controllers.
- Infrastructure includes the main application window and other common client mechanisms.

A screen is a main unit of the generic UI. It contains visual components, data containers and non-visual components. A screen can be displayed inside the main application window either in the tab or as a modal dialog.

The main part of the screen is a Java or Groovy class called controller. Layout of the screen is usually defined in an XML file called descriptor.

In order to show a screen, the framework creates a new instance of the `Window` visual component, connects the window with the screen controller and loads the screen layout components as child components of the window. After that, the screen's window is added to the main application window.

A fragment is another UI building block which can be used as part of screens and other fragments. It is very similar to screen internally, but has a specific lifecycle and the `Fragment` visual component instead of `Window` at the root of the components tree. Fragments also have controllers and XML descriptors.

A screen controller is a Java or Groovy class that contains the screen initialization and event handling logic. Normally, the controller is linked to an XML descriptor which defines the screen layout and data containers, but it can also create all visual and non-visual components programmatically.

All screen controllers implement the `FrameOwner` marker interface. The name of this interface means that it has a reference to a frame, which is a visual component representing the screen when it is shown in the main application window. There are two types of frames:

- `Window` – a standalone window that can be displayed inside the main application window in a tab or as a modal dialog.
- `Fragment` – a lightweight component that can be added to windows or other fragments.

Controllers are also divided into two distinct categories according to the frames they use:

- `Screen` – a base class of window controllers.
- `ScreenFragment` – a base class of fragment controllers.

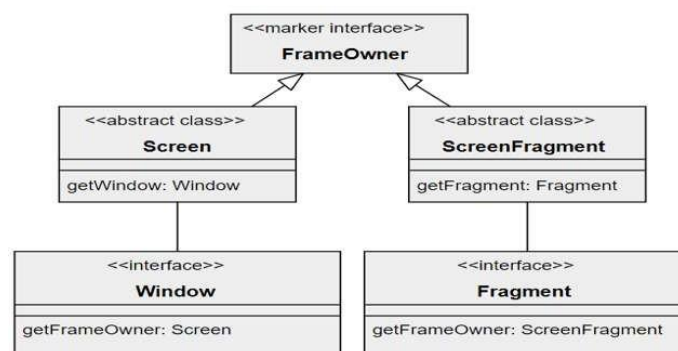


Figure 19: Controllers and Frames

Most of these user interactions are touch-based and happen on colorful touch screen displays that are bursting with high-level interactions. Naturally, basic mobile UI design principles differ from those of a traditional desktop UI.

After all, users are, by definition, on the move; control is limited, giving new meaning to the phrase >all thumbs. Actions and information need to be big, bold, clear, and simple.

As mobile adoption continues to rise year-by-year, it's time to develop a mobile-first strategy, embraced by the likes of Facebook or other social networks, who make sure their iOS and Android apps offer a polished user experience on hand-held devices.

After all, when users have more choice and freedom to find mobile applications that work for them, a poor user experience can easily devalue your brand, hurt your revenue, and disengage your users.

Aside from investing in mobile applications, many ecommerce stores see the increase of purchases coming from mobile. If an online store doesn't optimize checkout experience, usability or their mobile app design, they may lose market share or even render themselves obsolete.