

ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ MVVM В СРЕДЕ РАЗРАБОТКИ XAMARIN STUDIO ДЛЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ПОД УПРАВЛЕНИЕМ IOS

Ларионов С. С.

Цапко И. В.

Томский политехнический университет

ssl1@tpu.ru

Введение

Возможность кроссплатформенного программирования активно развивается в сфере разработки приложений для мобильных устройств. Уже сейчас существует достаточное количество средств разработки, которые позволяют создавать приложения на одном языке программирования для различных мобильных устройств. В пример таких средств разработки можно привести программные продукты Appcelerator Titanium, IBM WorkLight, Verivo Akula, Xamarin [1].

При создании программного продукта для управления проектами предприятия было решено использовать среду разработки Xamarin Studio. Также в ходе реализации проекта было принято решение использовать шаблон MVVM для разделения представления (пользовательского интерфейса) и модели данных (базы данных проектов на устройстве) [2].

Подробнее о MVVM

Шаблон MVVM удобен для использования в среде разработки Xamarin Studio. При работе с ним, различные мобильные устройства (устройства под управлением Android, iOS, WP) взаимодействуют с одной общей моделью данных, созданной с использованием MVVM, при помощи одной, общей для всех типов устройств, модели представления. Данный подход позволяет не создавать отдельно для каждого устройства свой контроллер для взаимодействия с моделью данных [3].

Тем не менее, само представление данных (View) создается для каждого устройства (Android / iOS / WP) отдельно. Это необходимо потому, что построение пользовательского интерфейса на устройствах под управлением Android, iOS, WP достаточно сильно отличается друг от друга. По требуемым шаблонам дизайна приложения в процессе выполнения работы был реализован удовлетворяющий требованиям пользовательский интерфейс для устройств под управления iOS.

Команды и связи MVVM в приложении

Для приложения управления проектами предприятия были разработаны шаблоны экранов для Android и iOS устройств, а также элементы ViewModel для каждого экрана. В результате был создан мобильный клиент приложения для устройств под управлением iOS, а также элементы ViewModel для экранов приложения на различных устройствах. Каждый экран приложения был связан со своим элементом ViewModel

посредством команд и связей. Команды и связи были разработаны и реализованы для работы пользовательского интерфейса с моделью данных. Каждый элемент ViewModel имеет доступ к модели данных, и подключен к экрану, который работает с моделью посредством этого элемента ViewModel.

Разработка данного проекта основывалась на создании следующих элементов: серверного API – интерфейса, элементов ViewModel, клиента Android и клиента iOS.

Связь (Binding) – подписка на изменение свойств объекта. В приложении управления проектами предприятия связи устанавливались для заполняемых текстовых полей логина и пароля, для имен экранов приложения, для таблицы проектов и комментариев к проектам, для текстовых полей на экране создания новой задачи, что будет рассмотрено на примере экрана «Создание новой задачи».

Команда (Command) – некое действие, описанное в модели представления (View Model). Данное действие вызывает пользовательский интерфейс. Команды назначены на все кнопки в приложении, а также на некие действия, например, выход из текущего экрана, переход в экран профиля пользователя, отображение предупреждения об ошибке.

На примере экрана «Создание новой задачи» (представленном на рисунке) можно рассмотреть подробнее команды и связи в приложении.

«IssueTextView» - элемент TextView (текстовое поле) пользовательского интерфейса, который имеет связь (Binding) с неким полем IssueDescription в модели представления данного экрана. При изменении значения поля данных данного элемента, связанное с ним поле в модели представления также изменяется.

«Apply» - элемент Button (кнопка) пользовательского интерфейса, запускающий команду (Command) создания новой задачи «ApplyIssueClickedCommand» в модели представления данного экрана. При выполнении, данная команда создает новую задачу, исходя из значений полей модели представления.

Примеры связи элемента представления с элементом модели представления, а также пример команды в представлении, запускающей команду в модели представления, представлены в листинге 1 и 2 соответственно.

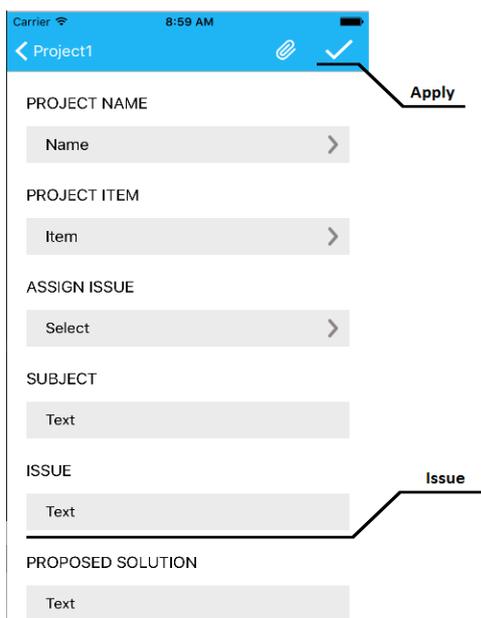


Рис. Экран «Создание задачи»

Список использованных источников

Шаблоны [Электронный ресурс] / Сеть разработчиков Microsoft. – URL: <http://msdn.microsoft.com/ru-ru/magazine/dd419663.aspx> (дата обращения 13.09.2016).

MVVM Explained [Электронный ресурс] / Code Project. – URL: <http://codeproject.com/Articles/100175/Model-View-ViewModel-MVVM-Explained> (дата обращения 13.09.2016).

iOS – Xamarin [Электронный ресурс] / Developer Center Xamarin. – URL: <http://developer.xamarin.com/guides/ios/> (дата обращения 14.09.2016). Пит Браун. Описание MVVM // Silverlight. Практическое руководство. – 2012. – Гл.

Листинг 1

```
_bindings.Add (this.SetBinding (()  
=> _viewModel.IssueDescription, ()  
=> IssueTextView.Text,  
BindingMode.TwoWay));
```

Листинг 2:

```
ApplyNavBarButton.Clicked +=  
(sender, ea) => {  
  
_viewModel.ApplyIssueClickedCommand.  
Execute (null);  
};
```

Заключение

В итоге можно выделить такие сильные стороны использования шаблона MVVM, как:

- возможность связи представления и модели представления;
- модель представления является некой абстракцией самого представления, поэтому нет необходимости изменять представление при изменениях в логике программы;
- при связи представления и модели представления, изменение связанных свойств происходит автоматически, в том числе есть возможность двухсторонней связи;
- в силу отделения программного кода от пользовательского интерфейса, разработка может вестись параллельно (программистами с одной стороны, и дизайнерами с другой).

Шаблон MVVM, предложенный корпорацией Microsoft, широко применяется в настоящее время в различных сферах разработки программного обеспечения. В том числе, данный шаблон нашел себе хорошее применение в разработке для мобильных устройств, что и было рассмотрено в данной работе.