РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ НА ЯЗЫКЕ C# С ИСПОЛЬЗОВАНИЕМ КРОССПЛАТФОРМЕННОГО ФРЕЙМВОРКА XAMARIN

Таисия ЗЕНЧЕНКО

CZU: 004.41:658.8 zencencotaisia@gmail.com

Mobile devices, especially smartphones, are widely used and beneficial due to their accessibility and features like geolocation, cameras, and more. Leveraging Xamarin, this work creates a cross-platform mobile app for bakery orders, enhancing user convenience and operational efficiency. The use of Xamarin to develop this mobile app marks a major advancement. It meets user expectations for convenience and accessibility while equipping businesses with tools to engage audiences and boost competitiveness.

Технологии являются неотъемлемой частью нашей жизни, охватывая работу, обучение, научные исследования, управление данными, космические технологии и государственные организации. Информационные технологии помогают выполнять функции и хранить данные с помощью мобильных и веб-приложений, а также административных панелей. Наиболее используемым устройством является мобильный телефон, благодаря его доступности и дополнительным ресурсам, таким как геолокация, камера, сканер QR-кодов, микрофон и навигация. Это ускоряет использование мобильных приложений. С ростом популярности мобильных устройств, разработка и дизайн мобильных приложений также прогрессируют, привлекая специалистов по UI/UX-дизайну и разработке логики приложений.

В зависимости от операционной системы (ОС), на которой работает устройство, приложения могут дорабатываться для соответствия её особенностям. ОС, в свою очередь, делятся на пакетные, мультипрограммные, многозадачные, с разделением времени, распределенные и сетевые.

Целью данной работы является создание мобильного приложения с использованием кроссплатформенного фреймворка Xamarin для заказа выпечки. Приложение имеет две точки входа: для администратора и для обычного пользователя. Основные задачи включают анализ требований клиента, создание шаблона и функционала приложения, разработку и тестирование приложения для всех типов мобильных устройств.

Актуальность приложения заключается в его разработке с использованием фреймворка Xamarin, что позволяет адаптировать его под любую операционную систему.

Одними из ключевых архитектурных подходов для создания и реализации практической части данной дипломной работы, являются:

1. **MVVM (Model-View-ViewModel)**
   - **Model** - «Модель» представляет собой данные приложения, которые могут включать массивы, базы данных или данные из веб-сервисов.
   - **View** «Представление» - визуальный интерфейс, с которым взаимодействует пользователь, отображающий данные и синхронизирующийся с моделью представления.
   - **ViewModel** «Модель Представления» - связующее звено между моделью и представлением, обрабатывающее взаимодействия, такие как нажатия кнопок и обновление данных в пользовательском интерфейсе. [1]
2. **MVP (Model-View-Presenter)**
   - **Model** «Модель» - управляет данными и бизнес-логикой приложения.
   - **View** «Представление» - отвечает за отображение данных и пользовательский интерфейс, взаимодействуя с пользователем.
   - **Presenter -** посредник между моделью и представлением, обрабатывает пользовательские действия, обновляет интерфейс и управляет логикой приложения. [2]
3. **Dependency Injection** или ***«Внедрение зависимостей»*** позволяет отделить конкретные типы от кода, который зависит от этих типов. Обычно используется контейнер, включающий список регистраций и соответствий между интерфейсами и абстрактными типами, а также конкретными типами, абстрактными типами, а также конкретные типы, которые реализуют или расширяют эти типы. [3]
4. ***Data Binding*** или ***«Привязка Данных».*** Xamarin.Forms также предлагает мощные возможности привязки данных, позволяя привязывать элементы пользовательского интерфейса непосредственно к источникам данных и автоматически обновлять пользовательский интерфейс при изменении данных. [4]

Все вышеперечисленные архитектурные компоненты Xamarin полезны тем, что помогают разработчику делегировать код по различным слоям и модулям, что способствует улучшению читаемости и навигации в проекте. Эти подходы повышают модульность и тестируемость приложения, облегчая внесение изменений и добавление новых функций, а также улучшают управляемость зависимостей.

Практическая часть данной дипломной работы включала в себя создание мобильного приложения, которое было разработано для небольшой частной компании из РМ по производству и продаже кондитерских изделий.

На рис.1 представлены пошаговые действия пользователя: от входа в систему и создания учетной записи и до выполнения заказа выбранного продукта. Более того, все поля регистрации содержат валидацию, без которой пользователь не сможет иметь доступ ко входу в приложение и создание учетной записи.

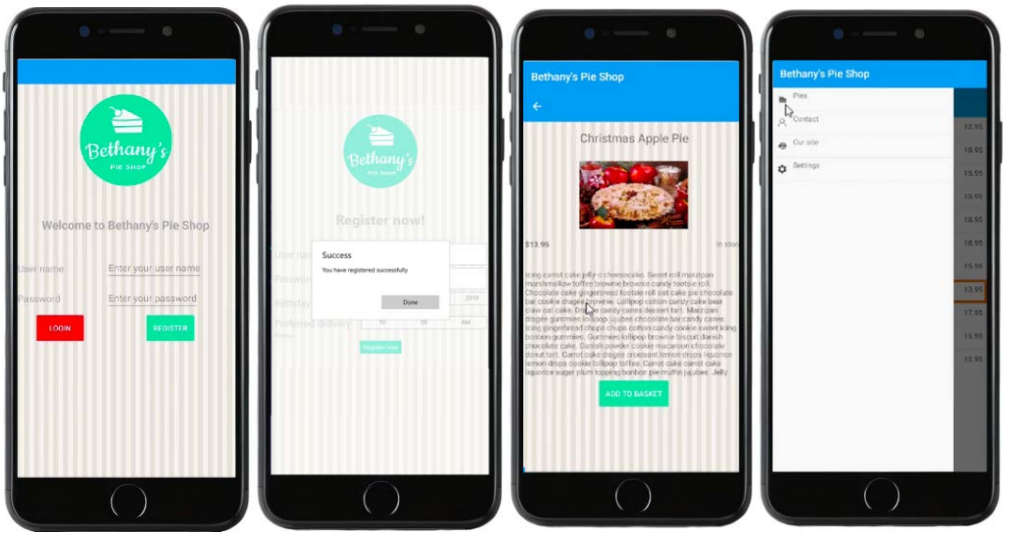


Рис. 1. Последовательность использования мобильного приложения

В ходе дипломной работы создано мобильное приложение с использованием Xamarin, адаптированное под различные операционные системы. Приложение позволяет удобно заказывать выпечку, предоставляет функционал для пользователей и администраторов, и было протестировано на стабильность. Основные задачи включали анализ требований, разработку дизайна и тестирование. Использование Xamarin обеспечивает кроссплатформенную доступность и эффективность. Приложение повышает удобство для пользователей и конкурентоспособность компании, предоставляя надежные инструменты для управления заказами.

Библиография:

1. Документация по MVVM архитектуре- https://metanit.com/sharp/xamarin/4.2.php
2. What is MVP Design Pattern? - https://medium.com/huawei-developers/what-is-mvp-design-pattern-c587feea27d7
3. Microsoft Docs: Dependency Injection - https://learn.microsoft.com/en-us/dotnet/core/extensions/dependency-injection
4. Introduction into Data Binding - https://devblogs.microsoft.com/xamarin/introduction-to-data-binding/

Recomandat
***Greta STURZA,** lector univ.*