

Разработка программного обеспечения для автоматизации работы предприятия общественного питания

По мере развития современного общества усложняются многие сферы человеческой деятельности. Сейчас практически всем видам бизнеса необходимо контролировать большие объемы информации. В результате предприятия сталкиваются с большими потоками документов, которые отнимают время на их ведение и непрерывное упорядочивание. Новые информационные технологии дают много новых возможностей для управления бизнесом и его частичной или полной автоматизации.

Была поставлена задача автоматизации предприятия общественного питания, с использованием рассматриваемой в данном тезисе информационной системы.

Цель - повышение производительности обслуживания, увеличение скорости обработки заказов, уменьшение количества бумажной документации, привлечение посетителей.

Решением поставленной задачи является внедрение на производство системы, состоящей из двух частей. Первая подсистема - это программный продукт, распространяющийся преимущественно на мобильных устройствах. Данное приложение представляет собой электронное меню той или иной сети общественного питания. Вторая подсистема является серверным программным обеспечением, которое обеспечивает взаимодействие с клиентами и сбор сопутствующей аналитики.

Данный продукт является системы, которая включает в себя 2 части: серверной-администраторская часть и клиентская часть. Клиент и сервер могут взаимодействовать между собой. Клиентом может быть, также независимо разработанный программный продукт, для любой аппаратной платформы, совокупность которых способна реализовать «общение» с сервером, учитывая определенное API серверной части (Рис. 1).

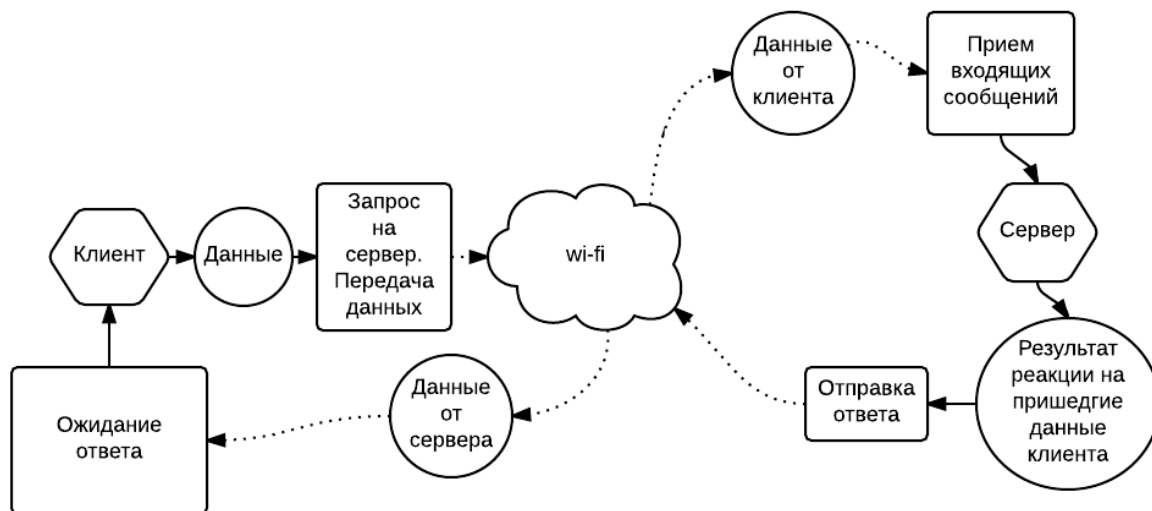


Рис.1 Взаимодействие разрабатываемого программного продукта и возможного клиента

Одна из частей системы реализована и внедрена – это клиентская часть. Сейчас можно загрузить приложение из магазина Play Market. На данный момент, клиентская часть приложения позволяет пользователям просматривать блюда на их Android устройствах, содержащиеся в меню данного заведения. Также, приложение позволяет узнать о блюдах такую информацию, как: цена, вес, описание. Каждое блюдо пользователь может добавить в корзину, затем перейти в нее и узнать общую стоимость заказа. Если же посетителя не устроит содержимое его корзины, он может убрать из нее выбранные блюда. На данный момент, клиентское приложение совершенствуется. Ведется работа по обеспечению

Секция 26. Современные технологии проектирования программного обеспечения

посетителей более широкими возможностями, такими как: возможность отправки заказа и комментариев на сервер. Также ведется проектирование клиентских приложений под iOS и Windows Phone.

Серверная подсистема находится на стадии разработки, и внедрение предстоит в ближайшем будущем. На стороне сервера происходит прием заявок от клиентов - это могут быть: сформированный заказ, комментарии к определенному блюду или оповещение о подключении клиента. В базе данных приложения хранятся информация о категориях и их блюдах, эти данные могут быть отправлены клиенту для их отображения. Так же пользователь этой подсистемы может самостоятельно сформировать заказ для определенного посетителя. В конечном итоге будет распечатан чек. Серверная часть обрабатывает все входящие данные, собирает сопутствующую аналитику и записывает их в базу данных. Собранная информация может быть отображена пользователю приложения в виде распределения блюд по числу заказов, общей выручки.

Разработка серверной части ведется на языке программирования C#. Язык C# выбран, так как он очень бурно развивается и поддерживается корпорацией Microsoft. Язык C# позволяет вести разработку под различные операционные системы, такие как Android, Windows 7, 8, Windows Phone, IOS, Linux, Mac OS. На этом языке можно разрабатывать очень производительные продукты, так как язык является очень мощным и гибким. [2]

Проектирование графического интерфейса ведётся с использованием технологии WPF (Windows Presentation Foundation) и шаблона программирования MVVM(Model-View-ViewModel). WPF выбран по многим причинам. В основе WPF лежит векторная система визуализации, не зависящая от разрешения устройства вывода. WPF предоставляет инструменты для создания визуального интерфейса, включая язык XAML, привязку данных, двухмерную и трёхмерную графику, анимацию, стили, и т.д. Графической технологией, лежащей в основе WPF, является DirectX, в отличие от Windows Forms, где используется GDI/GDI+. Производительность WPF выше, чем у GDI+ за счёт использования аппаратного ускорения графики через DirectX.[3]

MVVM это не отдельная технология, а часть WPF. MVVM является руководством к разработке бизнес логики и графического интерфейса. Самый важный момент WPF и MVVM – это инфраструктура привязки данных. За счет привязки свойств представления к модели представления получается слабое связывание этих компонентов, что полностью освобождает разработчика от необходимости писать в модели представления код, непосредственно обновляющий представление. Такой подход повышает читаемость кода, производительность, увеличивает возможности поддержки и переноса бизнес логики, а также упрощает написание тестов для приложения. Приведенная ниже разметка XAML демонстрирует привязку свойства, хранящей список категорий к элементу в окне, отображающей этот список:

```
<ListBox ItemsSource="{Binding mdl.Categories, Mode=TwoWay}"
```

Таким образом, нет необходимости писать логику, которая отслеживает изменения данных и применяет их к компоненту. Любые изменения, происходящие с переменной, моментально отображаются визуально.[3]

Современный мир требует качественные и удобные программные продукты, которые позволяют автоматизировать бизнес процессы. Это является важным этапом развития предприятия. Особенно важно использовать современные инструменты для создания приложений, требующих высокую степень поддержки разработчиками и производительности. Корпорация Microsoft обеспечивает удобные производительные средства разработки высококачественных программных систем и средств тестирования, что позволит разработчикам производить и развивать программные продукты в кратчайшие сроки и с надлежащим качеством.

Литература

1. Pro WPF 4.5 in C# 5.0: Windows Presentation Foundation in .NET 4.5, 4th edition Matthew MacDonald
2. C# 5.0 in a Nutshell: The Definitive Reference Joseph Albahari, Ben Albahari
3. Windows Presentation Foundation Unleashed (WPF). Adam Nathan