Министерство образования Московской области

Государственное образовательное учреждение высшего образования Московской области

«Государственный гуманитарно-технологический университет»

Ликино-Дулевский политехнический колледж – филиал ГГТУ

КУРСОВОЙ ПРОЕКТ

«Разработка мобильного приложения по организации работы с клиентами кафе»

МДК 02.01 «Технология разработки программного продукта»

Выполнил:

Толстов Тимур Артемьевич

Студент группы ИСП.19А

09.02.07 Информационные системы и программирование)

очной формы обучения

Руководитель:

Селиверстова Ольга Михайловна­­

Оценка

\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

Подпись руководителя

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ликино-Дулево

2023 год

Оглавления

[Введение 3](#_Toc130806606)

[1. Разработка системного проекта 3](#_Toc130806607)

[1.1. Назначение разработки 3](#_Toc130806608)

[1.2. Требования к функциональным характеристикам 4](#_Toc130806609)

[1.3. Требования к надёжности и безопасности 4](#_Toc130806610)

[1.4. Требования к составу и параметрам технических средств 4](#_Toc130806611)

[1.5. Требования к информационной и программной совместимости 5](#_Toc130806612)

[2. Разработка технического проекта 6](#_Toc130806613)

[2.1. Обоснование выбора Case – средств 6](#_Toc130806614)

[2.1.1. Диаграмма прецедентов 8](#_Toc130806615)

[2.1.2. Моделирование бизнес-процессов 9](#_Toc130806616)

[2.1.3. Словесный алгоритм бизнес-процесса 9](#_Toc130806617)

[2.1.4. Диаграмма действий бизнес-процесса 9](#_Toc130806618)

[2.1.5. Таблица операций 10](#_Toc130806619)

[3.1. Обоснование выбора средств разработки 12](#_Toc130806620)

[3.2. Руководство программиста 12](#_Toc130806621)

[4.Тестирование и отладка 18](#_Toc130806622)

[5. Методы и средства защиты бд 20](#_Toc130806623)

[Заключение 22](#_Toc130806624)

[Приложения 24](#_Toc130806625)

[Приложение 1 «Блок-схема программы» 24](#_Toc130806626)

# Введение

Автоматизация - это неотъемлемый спутник нынешнего времени, он представляет собой один из множества подходов к управлению процессами, во главе которого стоит применение информационных технологий. Это позволяет управлять всевозможными элементами: операциями, данными, ресурсами и информацией.

Cоздание собственного мобильного приложения для доставки еды, и это – лучшее решение для средних и крупных заведений. В 2011 году компания Starbucks отказалась от агрегатора и создала собственное мобильное приложение с соблюдением фирменного узнаваемого стиля. Через приложение можно оплатить заказ, участвовать в программе лояльности, смотреть последние новости и новинки от компании. В 2016 году 21% от общих доходов приходился на онлайн покупки. Собственный сервис сохраняет доходность от блюд, расширяя аудиторию. Диджитализация взаимоотношений между заведениями питания и клиентами – тенденция ближайшего будущего.

Независимо от выбранной модели организации доставки еды, предприятие питания получает массу выгод. Главная из них – высокая конкурентоспособность и повышение лояльности аудитории.

По объему онлайн-продаж товаров и услуг среди всех других каналов торговли несколько лет подряд лидируют мобильные приложения. Динамика эта не удивляет. 90% россиян владеют смартфонами, получая через эти карманные устройства всю необходимую информацию. За последние два года россияне провели в мобильных приложениях магазинов, торгующих едой, ресторанов, кафе на 49% больше времени.

При создании приложения используется следующее программное обеспечение:

* Microsoft Visual Studio – интегрированная среда разработки, используемый для разработки компьютерных программ, веб -сайтов. веб – приложений, веб – сервисов и мобильных приложений.
* Firebase.DataBase – сервис облачных баз данных.
* Microsoft Office – комплекс программного обеспечения включающий текстовый редактор, редактор таблиц, редактор презентаций и другие программные продуты.

# 1. Разработка системного проекта

## 1.1. Назначение разработки

Чтобы оформить заказ клиенту ресторана необходимо позвонить в ресторан сообщить позиции заказа оператору ресторана, после ожидать курьера доставки ресторана и оплатить заказа по терминалу курьера или наличкой. Для автоматизации этого процесса разработано мобильное приложение «Pizza30d»

Мобильное приложение предназначено для автоматизации оформления заказов в ресторане «Pizza 30d». Пользователями программы являются клиентами ресторана. Приложение формирует заказы пользователей к ресторану, на основе их выбора в меню.

## 1.2. Требования к функциональным характеристикам

Функциональные требования к приложению:

* Авторизация и регистрация в приложении.
* Фильтрация меню по категориям и по названию продукта.
* Вывод данных из коллекций базы данных (
* Проверка вводимых данных пользователем
* Удаление, Добавление, Редактирование данных в коллекциях базы данных(Адресы, Карты, Корзина, Пользователь)
* Формирование заказа в приложении и сохранение его в базу данных.

## 1.3. Требования к надёжности и безопасности

Разрабатываемое программное обеспечение должно иметь:

* Парольную защиту при запуске программы
* Ограничение несанкционированного доступа к данным
* Исключение несанкционированного копирования
* Возможность резервного копирования базы данных

## 1.4. Требования к составу и параметрам технических средств

Таблица №1 «Минимальные системные требования»

|  |  |
| --- | --- |
| Процессор | MediaTek Helio P22, 2000МГц, 8-ми ядерный |
| Операционный система | Android 5.1 |
| Дисплей | 5.45", IPS |
| Разрешение дисплея | 1440x720 |
| Обьем оперативной памяти | От 2 Гб |
| Свободного места в памяти | 200 мб |

Таблица №2 «Рекомендуемые системные требования»

|  |  |
| --- | --- |
|  |  |
| Процессор | Snapdragon 8 Gen 2 |
| Операционный система | Android 10 |
| Дисплей | 6.5", IPS |
| Разрешение дисплея | 2560x1440 |
| Обьем оперативной памяти | От 4 Гб |
| Свободного места в памяти | 200 мб |

## 1.5. Требования к информационной и программной совместимости

Требования к информационной и программной совместимости:

Для корректной работы программы необходимо: ОС Android 5.1 и выше.

Visual Studio 2017 (и выше) – продукт включающая интегрированную среду разработки программного обеспечения и ряд других инструментальных инструментов. Нужна для реализации основного кода программы на языке C#

Язык программирования C# - компилируемый статически типизированный язык программирования общего был перенесён на множество других платформ

Nu-get библиотека Firebase.Database – служит для доступа к данным из базы данных Firebase к приложению.

# 2. Разработка технического проекта

## 2.1. Обоснование выбора Case – средств

При разработке приложения использовалось следующее case средство: Draw.io - инструмент для создания диаграмм, блок-схем, интеллект-карт, бизнес-макетов, отношений сущностей, программных блоков и другого. Сервис распространяется на бесплатной основе с открытым исходным кодом. Draw.io обладает богатым набором функций для визуализации большинства задач пользователя.

Классификация CASE средств

1. CASE средства верхнего уровня. Эти CASE средства ориентированы на начальные этапы построения информационной системы. Они связаны с анализом и планированием. CASE средства верхнего уровня обеспечивают стратегическое планирование, расстановку целей, задач и приоритетов, а также графическое представление необходимой информации. Все CASE средства верхнего уровня содержат графические инструменты построения диаграмм, таких как диаграммы сущность-связь (ER диаграммы), диаграммы потока данных (DFD), структурные схемы, деревья решений и пр.
2. CASE средства нижнего уровня. Эти CASE средства больше сфокусированы на последних этапах разработки информационной системы – проектирование, разработка программного кода, тестирование и внедрение. CASE средства нижнего уровня зависят от данных, которые предоставляют средства верхнего уровня. Они используются разработчиками приложений и помогают создать информационную систему, однако не являются полноценными инструментами разработки программного обеспечения.
3. Интегрированные CASE средства (I – CASE). Эти CASE средства охватывают полный жизненный цикл разработки информационной системы. Они позволяют обмениваться данными между инструментами верхнего и нижнего уровня и являются своего рода «мостом» между CASE средствами верхнего и нижнего уровней.

Из всего многообразия CASE средств, существующих на сегодняшний день, можно выделить три основные группы. Эти группы связаны с этапами разработки информационных систем и их жизненным циклом. Классификация CASE средств осуществляется в зависимости от того, какие из этапов разработки они поддерживают.

Для моделирования и оптимизации бизнес-процессов применяются CASE средства верхнего уровня и интегрированные CASE средства. Они позволяют повысить качество моделей бизнес процессов за счет автоматического контроля, дают возможность оценить ожидаемый результат, ускоряют процесс проектирования, обеспечивают возможности по изменению и обновлению моделей.

Выбор CASE средств для анализа и моделирования процессов зависит от многих факторов – финансовых возможностей, функциональных характеристик, подготовки персонала, применяемых информационно-технических средств и пр.

Приводить исчерпывающий состав этих факторов не имеет смысла, т.к. в ситуации выбора для каждого конкретного случая этот состав будет изменяться. Тем не менее, можно определить набор «базовых» факторов, на основании которых определяются критерии по выбору CASE средств.

К таким «базовым» факторам можно отнести следующие:

Цели моделирования и анализа процессов. Исходя из целей моделирования, определяются необходимые методы, которые должны поддерживать CASE средства. Также цели моделирования определяют необходимый уровень детализации моделей и формы представления отчетов.

Удобство для пользователей. Этот фактор определяет набор критериев для представления результатов моделирования наиболее понятным и приемлемым способом. Выбор CASE средств необходимо проводить с учетом того, чтобы пользователям приходилось затрачивать как можно меньше усилий на работу в среде CASE средств. CASE средства должны быть визуально и интуитивно понятны пользователям.

Применение стандартных методологий. Этот фактор определяет критерии выбора CASE средств, связанные с применением стандартных методологий анализа и моделирования бизнес-процессов. Как правило, моделирование не заканчивается созданием новых моделей процессов. Модели используются для внедрения информационных систем управления и автоматизации процессов. За счет стандартизации обеспечивается упрощение взаимодействия между CASE средствами и различными информационными системами.

Удобство эксплуатации. При выборе CASE средств необходимо учитывать такие характеристики как эффективность применения, сопровождаемость, переносимость моделей с одной системы на другую. Этот фактор в значительной степени связан с критериями, относящимися к техническим характеристикам аппаратного обеспечения.

Трудоемкость. Этот фактор определяет набор критериев, связанных с освоением и изучением работы CASE средств. При выборе следует учесть, сколько времени потребуется на обучение пользователей.

Субъективность. Данный фактор также не следует исключать из набора критериев по выбору CASE средств. При выборе могут существовать субъективные соображение выбора того или иного CASE средства, не связанные с рациональными критериями выбора.

## 2.1.1. Диаграмма прецедентов

Диаграмма прецедентов или диаграмма вариантов использования (англ. use case diagram) в UML — диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Основное назначение диаграммы — описание функциональности и поведения, позволяющее заказчику, конечному пользователю и разработчику совместно обсуждать проектируемую или существующую систему.

При моделировании системы с помощью диаграммы прецедентов системный аналитик стремится:

чётко отделить систему от её окружения;

определить действующих лиц (актёров), их взаимодействие с системой и ожидаемую функциональность системы;

определить в глоссарии предметной области понятия, относящиеся к детальному описанию функциональности системы (то есть прецедентов).

Работа над диаграммой может начаться с текстового описания, полученного при работе с заказчиком. При этом нефункциональные требования (например, конкретный язык или система программирования) при составлении модели прецедентов опускаются (для них составляется другой документ).

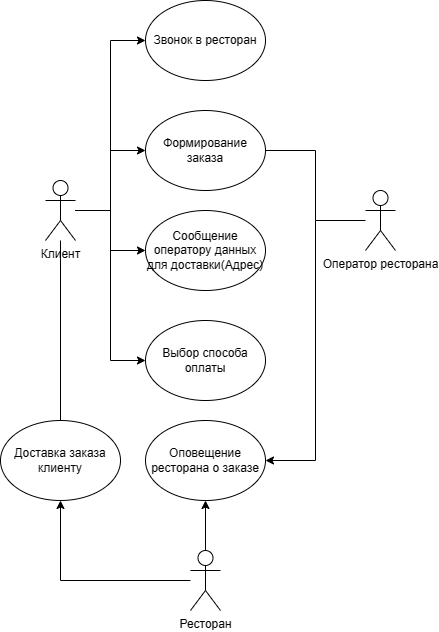


Рис.1 «Диаграмма прецедентов»

### 2.1.2. Моделирование бизнес-процессов

Моделирование бизнес-процессов (BPM) в управлении бизнес-процессами и системной инженерии — это деятельность по представлению процессов предприятия, позволяющая анализировать, улучшать и автоматизировать текущие бизнес-процессы. BPM обычно выполняется бизнес-аналитиками, которые предоставляют экспертные знания в области моделирования; экспертами по предмету, которые обладают специальными знаниями о моделируемых процессах; или, чаще всего, командой, состоящей из обоих. В качестве альтернативы модель процесса может быть получена непосредственно из журналов событий с помощью инструментов интеллектуального анализа процессов.

Таблица №3 «Кодировка бизнес-процессов»

|  |  |
| --- | --- |
| Номер бизнес-процесса | Название бизнес-процесса |
| 1-звон-в-р | Звонок в ресторан |
| 2-фор-за | Формирование заказа |
| З-соо-о-да-о-д | Сообщение данных для доставки оператору |
| 4-выб-спо | Выбор способа оплаты |
| 5-опо-рес-о-з | Оповещение ресторана о заказе |
| 6-до-зак-кли | Доставка заказа клиенту |

### 2.1.3. Словесный алгоритм бизнес-процесса

Для осуществления доставки клиенту необходим совершить звонок оператору ресторана. Для этого он должен: позвонить на номер оператора, сообщить заказ оператору, ожидать звонка курьера для получения доставки. После совершенных действий клиент получит свой заказ. Если клиент правильно сообщил адрес и оплата прошла успешно – доставка будет осуществлена.

### 2.1.4. Диаграмма действий бизнес-процесса

Диаграмма деятельности — UML-диаграмма, на которой показаны действия, состояния которых описаны на диаграмме состояний. Под деятельностью понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчинённых элементов — вложенных видов деятельности и отдельных действий, соединённых между собой потоками, которые идут от выходов одного узла ко входам другого.

Диаграммы деятельности используются при моделировании бизнес-процессов, технологических процессов, последовательных и параллельных вычислений.

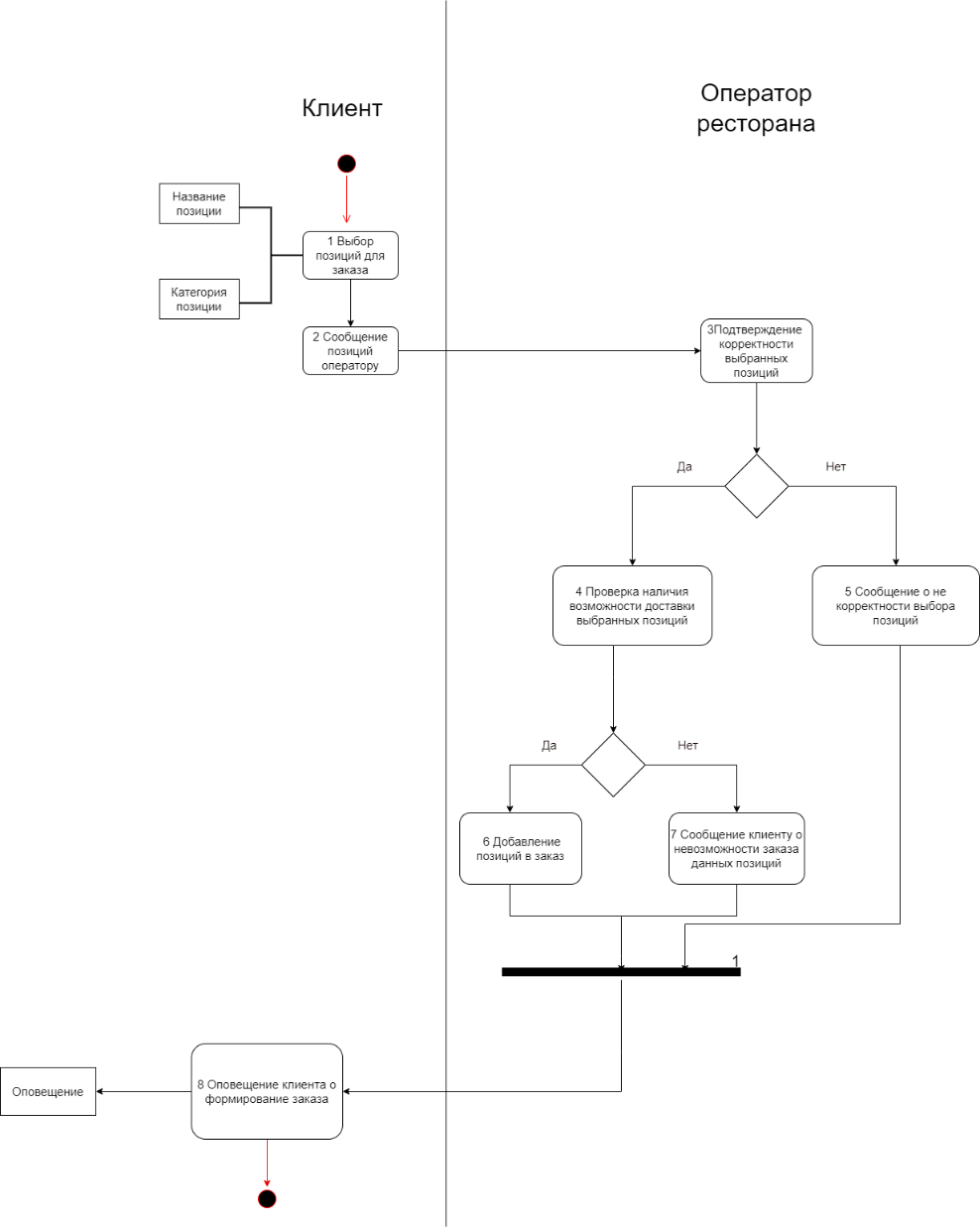


Рис. 2 «Диаграмма действий»

### 2.1.5. Таблица операций

Таблица №4 «Таблица операций»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Диаграмма и номер операции на  диаграмме | Операция | Исполнитель | Как часто | Входящие документы  (документы-  основания) | Исходящий  документ  (составляе­мый  документ) |
| 2-фор-за\_1 | Выбор позиций для заказа | Клиент | Один раз | Название позиции, категория позиции | Нет |
| 2-фор-за\_2 | Сообщение позиций оператору | Клиент | Один раз | Нет | Нет |
| 3 пот-кор-вы-поз | Подтверждение корректности выбранных позиций | Оператор ресторана | По мере необходимости | Нет | Нет |
| 4 про-на-воз-до-вы-поз | Проверка наличия возможности доставки выбранных позиций | Оператор ресторана | По мере необходимости | Нет | Нет |
| 5 сооб-о-не-кор-выб-поз | Сообщение о не корректности заказа | Оператор Ресторана | По мере необходимости | Нет | Нет |
| 6 доб-поз-в-зак | Добавление позиций в заказ | Оператор ресторана | По мере необходимости | Нет | Нет |
| 7 сооб-кл-о-не-зак-да-поз | Сообщение клиенту о невозможности заказа данных позиций | Оператор ресторана | По мере необходимости | Нет | Нет |
| 8оп-кл-о-фор-за | Оповещение клиента о формирование заказа | Клиент | Один раз | Оповещение | Нет |

3. Реализация

## 3.1. Обоснование выбора средств разработки

Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментов. Данные продукты позволяют разрабатывать как консольные приложения, так и игры и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, UWP а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, .NET Core, .NET, MAUI, Xbox, Windows Phone .NET Compact Framework и Silverlight. После покупки компании Xamarin корпорацией Microsoft появилась возможность разработки IOS и Android программ.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как, например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server).

## 3.2. Руководство программиста

Программа разработана на Visual Studio 2019 на платформе Xamarin.forms.

В приложении используется облачая база данных Firebase.DataBase, для работы с ней была установлена nu-get библиотека Firebase.Database.

Код для подключения (Ссылка берется из проекта в Firebase.Console)

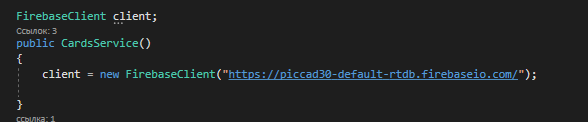
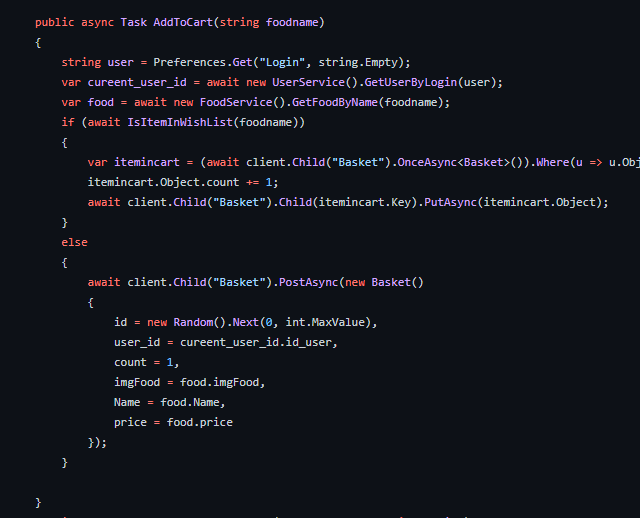


Рис. 14. «Подключение к базе данных»

Запросы к базе данных осуществляются асинхронно. Асинхронность нужна для выполнения нескольких процессов одновременно (Без «зависаний» приложения)

Пример написания асинхронного запроса:



Модель базы данных в обозревателе решений

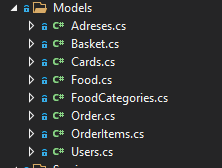


Рис. 15. «Модель данных»

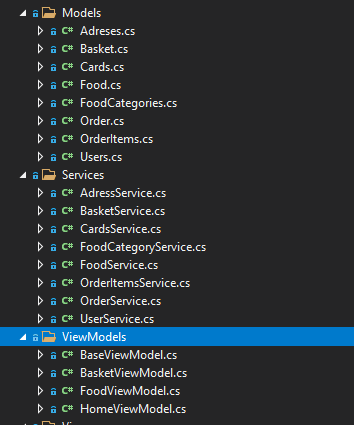


Рис. 17. «Классы программы»

Классы Services – для запросов к базе данных.

Классы models – модели данных нужные для работы с базой данных.

Классы ViewModels – классы имеющие интерфейс InotifyPropertyChanged. Нужны для привязки данных к определенным страницам (Отделения программной логики от интерфейса)

3.3. Руководство пользователя

|  |  |
| --- | --- |
| Действие | Скриншот |
| Запустить приложение на мобильном устройстве. И дождаться загрузки приложения и появления окна авторизации.  Введите данные для авторизации, если их нет пройдите регистрацию и войдите в приложение.  При вводе ошибочных данных, появится всплывающее окно сообщающее о ошибке. |  |
| После авторизации откроется главная страница, тут можно посмотреть меню рестораны. И выбрать понравившееся блюдо. Для этого нужно нажать по нему и откроется страница добавления этого блюда в корзину. |  |
| На данной странице нажимаем добавить в корзину. После этого данная позиция попадает в корзину.  При нажатии кнопки «Добавить в корзину» появится всплывающее окно сообщающее о добавление позиции в корзину. |  |
| На странице подробнее можно перейти в профиль и заполнить адрес и карты для оформления заказа, также выйти из аккаунта.  При выходе из аккаунт появится всплывающее окно, которое потребует второе подтверждение(Выхода из аккаунта) |  |
| На странице «Рестораны» можно увидеть расположения ресторана на карте Google |  |
| На данной странице можно добавить адреса и информацию о своих картах, для дальнейшего оформления заказа  При корректном добавление нового адреса появится всплывающее окно, подтверждающее успешное добавление нового адреса. |  |
| Для того чтобы перейти в корзину необходимо нажать на иконку корзины в правом верхнем углу. На этой странице можно увидеть информацию о заказанных блюдах и также перейти к оформлению заказа. |  |
| Выбираем адрес и карту и нажимаем «заказать». Заказ оформлен  При корректном оформление заказа появится всплывающее окно о успешно пройденной оплате, иначе приложении сообщит о ошибки ввода данных. |  |

# 4.Тестирование и отладка

Тестирование - процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определённым образом

В первую очередь рассмотрим уровни тестирования. Выделяют 4 основных уровня тестирования:

1. Компонентное/модульное тестирование (Component/Unit Testing).

2. Интеграционное тестирование (Integration Testing).

3. Системное тестирование (System Testing).

4. Приемочное тестирование (Acceptance Testing).

Каждый уровень тестирования направлен на определенную часть программы и выполняет свои цели.

Компонентное/модульное тестирование

Этот вид тестирования выполняется на самой ранней стадии разработки программы — во время написания кода. Обычно его выполняет сам программист, который пишет код. Следовательно, ошибки, в большинстве случаев, исправляются сразу же и не попадают к специалистам по тестированию.

Интеграционное тестирование

Интеграционное тестирование необходимо для того ,чтобы тестировать взаимосвязь между чем-либо.

Системное тестирование

При системном тестировании наша задача уже состоит в том, чтобы убедиться в корректности работы в целом всей системы. Программа в этом случае должна быть максимально приближена к конечному результату. А наше внимание должно быть сосредоточено на общем поведении системы с точки зрения конечных пользователей.

Ошибка при выходи из аккаунта, при переходе при помощи абсолютной ссылки на страницу

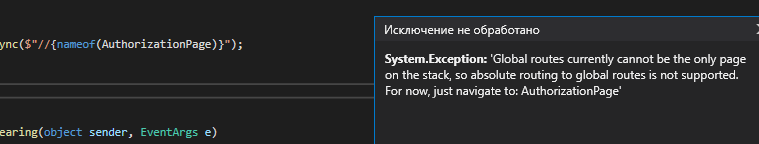


Рис. 33. «Ошибка при переходе на страницу»

Решение: Замена абсолютной ссылки на вызов модального окна.

Await Shell.Current.Navigation.PushModalAsync(new AuthorizationPage());

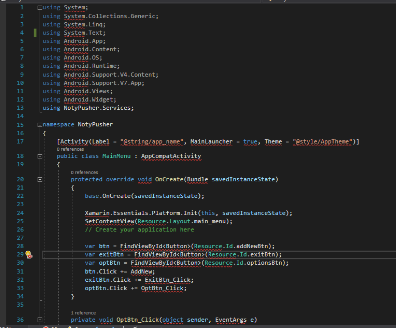


Рис. 34. «Ошибка системных библиотек»

Решение: Перестройка проекта с целью восстановления системных библиотек

# 5. Методы и средства защиты бд

Методы защиты баз данных: защита паролем, шифрование, разграничение прав доступа. Методы защиты баз данных в различных СУБД несколько отличаются друг от друга. Анализ современных СУБД показывает, что они условно делятся на две группы: основные и дополнительные.

* К основным средствам зашиты относится:
* зашита паролем;
* шифрование данных и программ;
* разграничение прав доступа к объектам базы данных;
* защита полей и записей таблиц БД.

Защита паролем представляет собой простой и эффективный способ зашиты БД от несанкционированного доступа. Пароли устанавливаются пользователями или администраторами БД. Учет и хранение паролей выполняется самой СУБД. Обычно, пароли хранятся в определенных системных файлах СУБД в зашифрованном виде. После ввода пароля пользователю СУБД предоставляются все возможности по работе с БД. Парольная зашита является достаточно слабым средством, особенно если пароль не шифруется. Основной ее недостаток состоит в том, что все пользователи, использующие одинаковый пароль, с точки зрения вычислительной системы неразличимы. Более мощным средством зашиты данных от просмотра является их шифрование. Шифрование — это преобразование читаемого текста в нечитаемый текст. При помощи некоторого алгоритма: применяется для зашиты уязвимых данных. Процесс дешифрования восстанавливает данные в исходное состояние. В целях контроля использования основных ресурсов СУБД во многих системах имеются средства установления прав доступа к объектам БД. Права доступа определяют возможные действия над объектами. Владелец объекта (пользователь, создавший объект). а также администратор БД имеют все права. Остальные пользователи к разным объектам могут иметь различные уровни доступа.

Разрешение на доступ к конкретным объектам базы данных сохраняется в файле рабочей группы. Файл рабочей группы содержит данные о пользователях группы и считывается во время запуска. Файл содержит следующую информацию: имена учетных записей пользователей, пароли пользователей, имена групп. в которые входят пользователи. По отношению к таблицам могут предусматриваться следующие права доступа:

* просмотр (чтение) данных;
* изменение (редактирование) данных;
* добавление новых записей;
* добавление и удаление данных;
* изменение структуры таблицы.

К данным, имеющимся в таблице, могут применяться меры защиты по отношению к отдельным полям и отдельным записям. Защита данных в полях таблиц предусматривает следующие уровни прав доступа:

* полный запрет доступ;
* только чтение;
* разрешение всех операций (просмотр. ввод новых значений, удаление и изменение).

К дополнительным средствам защиты БД можно отнести такие, которые нельзя прямо отнести к средствам зашиты, но которые непосредственно влияют на безопасность данных. Их составляют следующие средства:

* встроенные средства контроля значений данных в соответствии с типами;
* повышения достоверности вводимых данных;
* обеспечения целостности связей таблиц;
* организации совместного использования объектов БД в сети

# Заключение

Для разработки приложения использовались следующие средства: Microsoft Visual Studio 2019, Firebase RealtimeDatabase. Для отладки и тестирование приложения использовались встроенные программные средства языка C#. Приложение обладает удобным графическим интерфейсом, имеется главная форма с меню навигации. Приложение имеет средства защиты в виде логина и пароля для входа в систему. Разработана техническая документации в соответствии с ГОСТом.

Для отладки и тестирование приложения использовались встроенные программные средства языка C#.

На основании созданного приложения и в соответствие с ГОСТами ГОСТ «19.505 – 79», ГОСТ «19.503 – 79». Разработана техническая документация: руководство пользователя, руководство программиста, программа и методика испытаний, текст программы.

Список литературы

1. Илюшечкин, В. М. Основы использования и проектирования баз данных: учебник для среднего профессионального образования /В. М. Илюшечкин. — испр. и доп. — Москва: Издательство Юрайт, 2019. — 213 с. — (Профессиональное образование). — ISBN 978-5-534-01283-5.

2. Нестеров, С. А. Базы данных: учебник и практикум для среднего профессионального образования / С. А. Нестеров. — Москва: Издательство Юрайт, 2020. — 230 с. — (Профессиональное образование). — ISBN 978-5-534-11629-8.

3. Федорова Г.Н. Основы проектирования баз данных: учебник для студентов СПО /Г.Н. Федорова. - 2-е изд., стер. – Москва: Академия, 2018. – 219с.

4. Гордеев, С. И. Организация баз данных в 2 ч. Часть 1: учебник для среднего профессионального образования / С. И. Гордеев, В. Н. Волошина. — 2-е изд., испр. и доп. — Москва: Издательство Юрайт, 2020. — 310 с. — (Профессиональное образование). — ISBN 978-5-534-11626-7.

5. Гордеев, С. И. Организация баз данных в 2 ч. Часть 2: учебник для среднего профессионального образования / С. И. Гордеев, В. Н. Волошина. — 2-е изд., испр. и доп. — Москва: Издательство Юрайт, 2020. — 513 с. — (Профессиональное образование). — ISBN 978-5-534-11625-0.

6. Стасышин, В. М. Базы данных: технологии доступа: учебное пособие для среднего профессионального образования / В. М. Стасышин, Т. Л. Стасышина. — 2-е изд., испр. и доп. — Москва: Издательство Юрайт, 2020. — 164 с. — (Профессиональное образование). — ISBN 978-5-534-09888-4.

7. Рудаков А.В. Технология разработки программных продуктов: учеб. Пособие для студ. сред. проф. Образования / А. В. Рудаков – 2-е изд., стер. – М.: Издательский центр «Академия», 2006 – 208 с.

8. Черников В. А. Разработка мобильных приложений на С# для ios и Android / Черников В. А. – 1-е изд., стер. – М.: Издательский центр «Питер», 2006 – 208 с.

9. Дэвид Гриффитс, Дон Гриффитс. Head First. Программирование для Android. 2-е изд. (2018)

10. Ян Ф. Дарвин / Android. Сборник рецептов. Задачи и решения для разработчиков приложений. (2018)

**Интернет-ресурсы**

1. ЭБС «Университетская библиотека online» - http://biblioclub.ru/
2. ЭБС «ЮРАЙТ» - https://urait.ru/

# Приложения

# Приложение 1 «Блок-схема программы»

