МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Южно-Уральский государственный университет»**

**(национальный исследовательский университет)**

**Высшая школа электроники и компьютерных наук**

**Кафедра системного программирования**

|  |  |
| --- | --- |
| РАБОТА ПРОВЕРЕНА  Рецензент  <ученая степень, ученое звание>  \_\_\_\_\_\_\_\_\_\_ <И.О. Фамилия рецензента>  “\_\_\_”\_\_\_\_\_\_\_\_\_\_\_ 2018 г. | ДОПУСТИТЬ К ЗАЩИТЕ  Заведующий кафедрой, д.ф.-м.н., профессор  \_\_\_\_\_\_\_\_\_\_ Л.Б. Соколинский  “\_\_\_”\_\_\_\_\_\_\_\_\_\_\_ 2018 г. |

**Разработка мобильной кулинарной книги для ОС iOS с функцией подбора сочетаемых ингредиентов**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

ЮУрГУ – 02.03.02.2018.115-007.ВКР

|  |  |
| --- | --- |
|  | Научный руководитель  к.ф.-м.н., доцент  \_\_\_\_\_\_\_\_\_\_Г.И. Радченко    Автор работы, студент группы КЭ-401  \_\_\_\_\_\_\_\_\_\_Г.В. Волкова  Ученый секретарь  (нормоконтролер)  \_\_\_\_\_\_\_\_\_\_\_\_\_ О.Н. Иванова  “\_\_\_”\_\_\_\_\_\_\_\_\_\_\_ 2018 г. |

Челябинск-2018

**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 3](#_Toc511863019)

[1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И СУЩЕСТВУЮЩИХ РАБОТ ПО ТЕМАТИКЕ ДИПЛОМА 5](#_Toc511863020)

[1.1. Анализ предметной области 5](#_Toc511863021)

[1.2. Анализ схожих проектов 6](#_Toc511863022)

[1.3. Анализ существующих решений для реализации 8](#_Toc511863023)

[ВЫВОД 9](#_Toc511863024)

[2. ТРЕБОВАНИЯ К СИСТЕМЕ 9](#_Toc511863025)

[2.1. Функциональные требования к системе 10](#_Toc511863026)

[2.2. Нефункциональные требования к системе 10](#_Toc511863027)

[2.3. Варианты использования системы 10](#_Toc511863028)

[ЛИТЕРАТУРА 12](#_Toc511863029)

# 

# ВВЕДЕНИЕ

**Актуальность темы**

Мобильные приложения входят в жизнь людей с огромной скоростью. По данным [1] за 2016 год, в магазине приложений App Store располагалось более 2 миллионов приложений. Все они относятся к различным категориям: игры, здоровье, музыка, кулинария, образование и пр. Мобильные приложения облегчают повседневные задачи, помогают в их выполнении, разнообразят рутинные дела, вовлекают в процесс выполнения, казалось бы совершенно обычных и даже скучных дел, и даже могут помочь улучшить некоторые навыки и способности.

В наше время, практически у каждого человека есть смартфон, который всегда под рукой. Люди стали быстрее получать различную информацию, и сами данные стали доступнее с появлением смартфонов. По статистике, человек, пользующийся смартфоном, в среднем проверяет свой телефон раз в 6,5 минут [2]. Поэтому, актуальность темы мобильных приложений достаточно высока.

Данная работа направлена на создание мобильного приложения, которое облегчит такую повседневную обязанность, как приготовление еды. Отличительной особенностью системы будет подбор сочетающихся ингредиентов к выбранному пользователем продукту, для улучшения вкусовых качеств повседневных блюд.

**Цели и задачи работы**

Целью данной работы является разработка мобильной кулинарной книги для ОС iOS с функцией подбора сочетаемых ингредиентов. Для достижения цели работы, необходимо решить следующие задачи:

* Анализ литературы и смежных проектов, связанных с мобильной кулинарией;
* Определение требований к мобильному приложению для мобильной кулинарной книги;
* Разработка архитектуры мобильного приложения для мобильной кулинарной книги;
* Разработка схемы взаимодействия пользователя с интерфейсом приложения;
* Создание базы данных с сочетаемыми ингредиентами для мобильной кулинарной книги;
* Разработка архитектуры базы данных для хранения данных, вводимых пользователем.

**Объем и структура работы**

**Краткое содержание работы**

## АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И СУЩЕСТВУЮЩИХ РАБОТ ПО ТЕМАТИКЕ ДИПЛОМА

## Проектирование UI мобильных приложений

Разработка мобильных приложений на сегодняшний день – долгий и трудоемкий процесс. Чтобы приложение было успешным на рынке, необходимо учесть и проработать многие аспекты, такие как дизайн, функционал, хранение и обработка данных, база данных.

Важной частью в создании мобильных приложений является создание удобного к применению интерфейса. В статье [3] представлены основные проблемы проектирования мобильных приложений, такие как:

* разные размеры экранов мобильных телефонов и других похожих устройств;
* трудности ввода данных в мобильное приложение;
* легкость в использовании мобильного приложения;
* безопасность хранения данных.

Так, опираясь на статью [3], для создания мобильного приложения необходимо разработать простой в использовании и приятный для пользователя интерфейс.

Надо больше примеров, источников

До сюда.

## Анализ схожих проектов

Для получения и хранения рецептов различных блюд существует множество мобильных приложений и вeб-сервисов. Но не все они выполняют функцию подбора сочетаемых ингредиентов. Рассмотрим функционал некоторых из них, доступных на территории РФ (рис. 1). Приложение «Eat this much» (рис. 1-а) обеспечивает пользователю следующие возможности:

* Подбор персонального рациона на неделю;
* Рецепты подобранных блюд;
* Подсчет калорий и БЖУ блюд;
* Добавление продуктов в корзину.

В приложении отсутствует функция подбора сочетаемых ингредиентов, также оно доступно только на английском языке. Приложение «Твои рецепты» (рис. 1-б) реализует следующий функционал:

* Предоставление информации о рецептах по категориям, коллекциям или авторам;
* Поиск рецептов по категориям;
* Добавление понравившихся рецептов во вкладку «Избранное»;
* Добавление ингредиентов в корзину.
* Приложение не реализует функцию подбора сочетаемых ингредиентов. Также, многие рецепты доступны только в платной версии приложения.

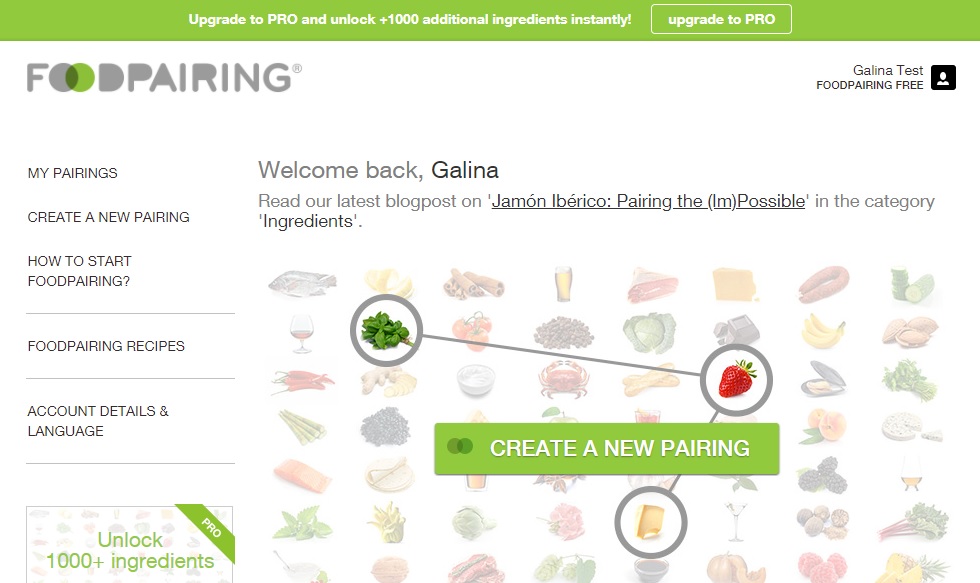
Приложение «Что с чем» (рис. 1-в) реализует функцию подбора сочетаемых ингредиентов. Однако, продукты объединены в категории (например, нельзя отдельно найти ингредиенты для свинины, только для мяса). Соответственно, подбор ингредиентов происходит так же по категориям, а не по отдельным продуктам.

|  |  |  |
| --- | --- | --- |
| C:\Users\Иван\Desktop\Диплом\diplom\скрины\eat this much.jpg  а) | б) | в) |

**Рис. 1.** Скриншоты мобильных приложений по кулинарии:

а) Eat this much; б) Твои рецепты; в) Что с чем

Отдельного внимания заслуживает вeб-приложение «FoodPairing» (рис. 2). При помощи данного сервиса, пользователь может самостоятельно подобрать продукты разной степени сочетаемости. Количество выбранных продуктов не ограничивается, и можно подобрать сочетания как с одним, так и с несколькими продуктами. Сервис предоставляется по премиум модели: основной функционал и некоторые продукты доступны бесплатно, дополнительные функции и продукты – по подписке. Также, в данном приложении нет поддержки русского языка.



**Рис. 2.** Вeб-приложение «FoodPairing»

## Анализ подходов к реализации мобильных приложений

Для реализации приложения можно выделить два типа решений:

1. Кроссплатформенные – средства, которые позволяют вести разработку приложения под несколько различных платформ;
2. Нативные – средства, предоставляемые компанией-разработчиком платформы.

Так как в рамках данной работы разрабатывается мобильное приложения для операционной системы iOS, рассмотрим инструментарий для разработки, предоставленный компанией Apple – iOS SDK. iOS SDK [4] – это пакет средств для разработки, которые могут понадобиться для создания мобильного приложения на смартфоны или планшеты компании Apple. Основными средствами для создания проекта будут следующие компоненты:

1. Xcode – интегрированная среда разработки со встроенным эмулятором iPhone;
2. Interface Builder [5] – компонент Xcode, необходимый для создания интерфейса программ и связей между событиями интерфейса и кодом программы.

Для создания мобильных приложений существует множество шаблонных решений – паттернов. Однако, наиболее распространенным является такой паттерн проектирования, как MVC (Model View Controller)[6]. MVC классифицирует объекты в приложении по ролям. Существует три вида ролей:

1. Модель (model) содержит данные приложения и определяет механизмы манипуляций над ними;
2. Представление (view) отвечает за визуальное представление модели, а также элементов управления, с которыми пользователь может взаимодействовать;
3. Контроллер (controller) управляет всей работой. Он имеет до-ступ к данным модели, отображает эти данные в представлениях, подписывается на события и манипулирует данными

# ВЫВОД

После изучения существующих аналогов, было принято решение реализовать мобильное приложение для поиска сочетаемых продуктов на операционной системе iOS [7].

## ТРЕБОВАНИЯ К СИСТЕМЕ

Проект Appetizer будет представлять собой мобильное приложение, помогающее пользователю в подборе сочетаемых ингредиентов (продуктов) для выбранной категории блюд, создании и хранении собственных рецептов, формировании списка покупок. С помощью данного мобильного приложения, пользователь сможет быстро подбирать ингредиенты (продукты) для выбранного блюда, записывать и хранить понравившиеся рецепты в одном месте и иметь к ним быстрый доступ. Исходя из выбранной пользователем категории пищи, приложение будет выводить на экран ингредиенты, сочетаемые с данным блюдом, которые будут храниться в базе данных продуктов. Система должна облегчить пользователю выбор между различными специями и значительно улучшить вкусовые качества готового продукта. Также, в системе есть возможность создать список покупок, в который можно добавлять как собственные пункты, так и предложенные системой ингредиенты.

Для использования мобильного приложения пользователю понадобится смартфон с операционной системой iOS. Доступ к интернету для использования приложения не требуется.

## Функциональные требования к системе

Можно определить следующий набор функциональных требований к системе:

* 1. Система Appetizer должна отображать все имеющиеся ингредиенты (продукты), сочетающиеся с выбранным пользователем продуктом.
  2. Система Appetizer должна позволять пользователю вводить рецепты по готовому шаблону и сохранять их.
  3. Система Appetizer должна позволять пользователю создавать списки покупок с возможностью добавлять туда как свои пункты, так и элементы, предлагаемые системой при подборе сочетаемых продуктов.

## Нефункциональные требования к системе

Выделяются следующие нефункциональные требования к системе:

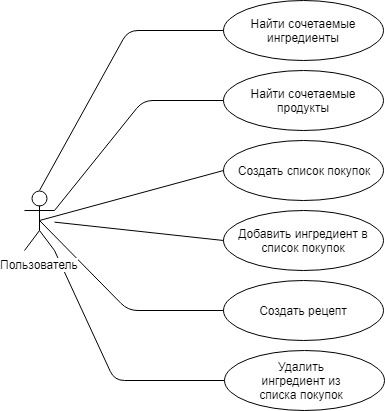
1. Система Appetizer должна быть написана на языке Swift.
2. Система Appetizer должна быть доступна на смартфонах с операционной системой iOS v XX и выше.
3. Система Appetizer должна брать данные о сочетаемых продуктах из готовой базы данных.
4. Хранение данных системы Appetizer должно быть реализовано при помощи Core Data.

## Варианты использования системы

Мы выделяем одного актера, взаимодействующего с системой (см. рис. 3).

Пользователь – это человек, взаимодействующий с приложением Appetizer, который может управлять вводом новых рецептов, просматривать все имеющиеся данные в приложении, управлять списком покупок.

Не понятно, чем отличаются продукты от ингредиентов, добавить и удалить что-то в/из списка покупок необходимо сделать как связанный вариант использование с блоком «создать список покупок» - автоматизированная генерация списка покупок на основе рецепта.



**Рис.3.** Диаграмма вариантов использования

Пользователь может реализовать следующие варианты использования системы:

Пользователь может выбрать продукт, предложенный из готового списка, и найти к нему все подходящие специи либо другие продукты.

Пользователь может создавать и управлять списками покупок. Пользователь может добавить собственные элементы в список, а может добавить продукты из подобранного программой списка сочетаний.

Пользователь может создавать свои рецепты по готовому шаблону и сохранять их в памяти устройства.

1. **АРХИТЕКТУРА СИСТЕМЫ**

Архитектура мобильного приложения Appetizer построена на основе паттерна проектирования MVC. Используя данный паттерн, мы разбиваем мобильное приложение на блоки: модель, отображение и контроллер., которые имеют слабую связь друг с другом. Тем самым, внося изменения в один блок, мы оказываем минимальное воздействие на остальные.

UP – UML и унифицированный процесс : <http://cmcstuff.esyr.org/vmkbotva-r15/4%20%D0%BA%D1%83%D1%80%D1%81/7%20%D0%A1%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80/%D0%9E%D0%9E%D0%90%D0%9F/%D0%9A%D0%BD%D0%B8%D0%B3%D0%B8/Arlou_i_Neyshtadt_UML_i_UP.pdf>

Глава 8 – выявление классов анализа

Глава 9 – отношения между классами

Результат: диаграмма классов анализа системы.

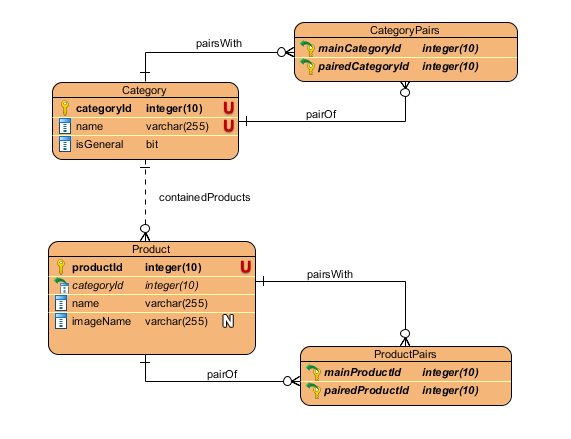
## Компоненты системы

Мобильное приложение Appetizer состоит из следующих компонентов:

1. локальная база данных, хранящая в себе информацию о продуктах и их сочетаемости(о базе данных подробнее в пункте 3.2);
2. контроллеры представлений:
   1. экран выбора главного продукта;
   2. экран подбора сочетаемых продуктов;
   3. экран вывода подобранных сочетаний;
   4. экран списка покупок
3. модели, реализующие логику связи контроллеров с базой данных:
   1. модель, позволяющая…
   2. модель, позволяющая…

## Структура базы данных

Для работы системы была разработана схема базы данных (см. рис 4), содержащая информацию о продуктах и об их сочетаемости между собой.



**Рис. 4.** Диаграмма отношений сущностей базы данных

Рассмотрим подробнее таблицы, представленные в базе данных:

**Category** – категория, к которой принадлежит продукт. Имеет следующие поля:

*CategoryID* – уникальный идентификатор категории. Является первичным ключом;

*Name* – имя категории;

*IsGeneral* – поле, в котором отмечается, является ли выбранная категория продукта главной (то есть, к ней ли мы ищем сочетания).

**Product –** таблица, хранящая в себе информацию о продукте. Здесь присутствуют следующие поля:

*ProductID* – уникальный идентификатор продукта. Является первичным ключом;

*CategoryID* – уникальный идентификатор категории, к которой принадлежит продукт. Значение берется из таблицы Category;

*Name* – имя продукта;

*ImageName* – название иконки, привязанной к продукту.

**CategoryPairs** – категория, сочетаемая с главной категорией, к кото-рой привязан выбранный продукт. У данной таблицы есть такие поля, как:

*MainCategoryID* – уникальный идентификатор категории выбранного продукта. Значение берется из таблицы Category;

*PairedCategoryId* – уникальный идентификатор категории, которая сочетается с категорией выбранного продукта. Значение берется из табли-цы Category.

1. **РЕАЛИЗАЦИЯ СИСТЕМЫ**

## Наполнение базы данных

Отдельного внимания в разработке мобильного приложения-кулинарной книги заслуживает создание базы данных сочетаемых ингредиентов. Для решения данной задачи существует несколько путей:

1. Интеграция существующих баз данных о сочетаниях;
2. Автоматическое наполнение базы данных с помощью обхода и парсинга кулинарных сайтов;
3. Создание базы данных вручную с помощью статей, сайтов и книг.

Лучшим решением была бы интеграция существующей базы данных, однако, такой базы в открытом доступе не существует. Поэтому, этот способ не подходит для решения поставленной задачи.

Автоматическое наполнение базы данных на основе кулинарных статей и книг достаточно быстрый способ создания базы данных, однако он не является надежным. В базе могут появляться ненужные объекты, они могут дублироваться и создавать проблемы при использовании приложения.

Наполнение базы данных вручную – самый долгий и трудозатратный процесс, однако при этом варианте наполнение будет наиболее эффективным и достоверным, так как будет контролироваться непосредственно разработчиком. Поэтому, в рамках данной работы, было принято решение о создании базы данных вручную, используя различные кулинарные сайты, книги, статьи.

Что было сделано в итоге – количество записей, формат заполнения, установка связей, теггирование и др.

# ЛИТЕРАТУРА

1. Статистика мобильных приложений в AppStore. [Электронный ресурс] URL: https://www.theverge.com/2016/6/13/11922926/apple-apps-2-million-wwdc-2016 (дата обращения 19.04.2018).
2. Suhailah Mohd Yusof, Sharifah Fateen Syuhada Syed Lokman. Personal Financial Planner: A Mobile Application that Implementing Forward Chaining Technique for Notification Mechanism // IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), April, 2014, pp. 65-69.
3. Peter Tarasewich. Designing Mobile Commerce Applications // Communications of the acm. Vol. 46, Issue 12, December 2003, pp. 57-60.
4. Документация Apple iOS. [Электронный ресурс] URL: https://developer.apple.com/library/ios/documentation (дата обращения: 19.04.2018).
5. Interface Builder. [Электронный ресурс] URL: https://developer.apple.com/xcode/interface-builder/ (дата обращения: 05.05.2017).
6. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Д. Приемы объектно-

ориентированного проектирования. Паттерны проектирования; [пер. с англ.: А. Слинкин науч. ред.: Н. Шалаев]. – Санкт-Петербург [и др.]: Питер, 2014. - 366 с.

1. Марк Д., Наттинг Д., Ламарш Д., Олссон Ф. iOS 6 SDK. Разработка приложений для iPhone, iPad и iPod touch на Objective-C в Xcode. – СПб.:Вильямс, 2013. – 672 с