**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**«ПОЛОЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Факультет | *информационных технологий* | | | | | | |  | | | |
|  | Кафедра | *технологий программирования* | | | | | | |  | | | |
|  |  |  |  | | | | | |  | | | |
|  |  |  | **ДОПУЩЕН К ЗАЩИТЕ** | | | | | | | | | |
|  |  |  | Заведующий кафедрой | | | | | | | | | |
|  |  |  |  | | | |  | *О.Н. Петрович* | | | | |
|  |  |  | подпись | | | |  | инициалы и фамилия | | | | |
|  |  |  | « |  | » |  | | | | 20 | *20* | г. |
|  | | | | | | | | | | | | |

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**ДИПЛОМНОГО ПРОЕКТА**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| « | *Мобильное приложение «GoodBooze» по поиску напитков* | | | | | | | |  | |
|  | | | *для платформы Android (клиентская часть)* | | | | | | » | |
| наименование темы | | | | | | | | | |
| Специальность | | | | *1-40 01 01 «Программное обеспечение информационных технологий»* | | | | | | |
|  | | | | код, наименование специальности | | | | | | |
|  | | | |  | | | | | | |
| Специализация | | | | *1-40 01 01 01 «Веб-технологии и программное обеспечение мобильных систем»* | | | | | | |
|  | | | | код, наименование специализации | | | | | | |
|  | | | |  | | | | | | |
| Студент-дипломник | | | | |  |  |  |  | | |
| группы | | *16-ИТ-3* | | |  |  |  | *А.С. Яблонский* | | |
|  | | номер | | |  | подпись, дата |  | инициалы и фамилия | | |
| Руководитель | | | | | |  |  | *О.Н. Забелендик* | | |
|  | | | | | | подпись, дата |  | инициалы и фамилия, уч. степень, звание | | |

Консультанты:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| по разделу | *Экономическая часть* |  |  |  | *Г. Н. Ярыгина* |
|  | наименование раздела |  | подпись, дата |  | инициалы и фамилия, уч. степень, звание |
| по разделу | *Нормоконтроль* |  |  |  | *А. В. Войтехович* |
|  | наименование раздела |  | подпись, дата |  | инициалы и фамилия, уч. степень, звание |
| по разделу |  |  |  |  |  |
|  | наименование раздела |  | подпись, дата |  | инициалы и фамилия, уч. степень, звание |
| по разделу |  |  |  |  |  |
|  | наименование раздела |  | подпись, дата |  | инициалы и фамилия, уч. степень, звание |
| по разделу |  |  |  |  |  |
|  | наименование раздела |  | подпись, дата |  | инициалы и фамилия, уч. степень, звание |
| по разделу |  |  |  |  |  |
|  | наименование раздела |  | подпись, дата |  | инициалы и фамилия, уч. степень, звание |
| по разделу |  |  |  |  |  |
|  | наименование раздела |  | подпись, дата |  | инициалы и фамилия, уч. степень, звание |

Объем проекта:

пояснительная записка - \_76\_ страниц;

графическая часть - \_4\_ листов;

магнитные (цифровые) носители - \_ единиц;

**Новополоцк 2020**

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**«ПОЛОЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | | | | | | |  | | | |
| Кафедра | *технологий программирования* | | | | | | | |  | | | |
|  |  |  |  | | | | | |  | | | |
|  |  |  | **УТВЕРЖДАЮ** | | | | | | | | | |
|  |  |  | Заведующий кафедрой | | | | | | | | | |
|  |  |  |  | | | |  | *Петрович О.Н.* | | | | |
|  |  |  | подпись | | | |  | (фамилия, инициалы) | | | | |
|  |  |  | « |  | » |  | | | | 20 | *20* | г. |
|  | | | | | | | | | | | | |

**ЗАДАНИЕ НА ДИПЛОМНЫЙ ПРОЕКТ**

Студенту \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Яблонский А.С\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, инициалы)

1. Тема дипломного проекта \_\_ *Мобильное приложение «GoodBooze» по поиску напитков для платформы Android (клиентская часть)* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(наименование темы)

Утверждена приказом руководителя учреждения высшего образования

от «\_\_*05*\_\_» \_\_\_\_\_\_\_*05*\_\_\_\_\_\_\_\_\_\_\_\_\_ 20 \_*20*\_ г. № \_\_\_\_\_*256ув* \_\_\_\_\_\_\_\_

1. Исходные данные к дипломному проекту \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*техническое задание от руководителя дипломного проекта* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

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

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

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

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

1. Перечень подлежащих разработке вопросов или краткое содержание расчетно-пояснительной записки \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Введение*

1. *Анализ исходных данных и постановка задачи*
2. *Проектирование программного обеспечения*
3. *Реализация и тестирование*

*Заключение*

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

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

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

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

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

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

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

1. Перечень графического материала (с точным указанием обязательных чертежей и графиков) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. *Диаграмма вариантов использования*
3. *Диаграмма себестоимости программного обеспечения*
4. *Экранные формы приложения*
5. *Диаграмма распределения времени работы при разработке продукта*

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

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

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

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

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

1. Консультанты по дипломному проекту с указанием относящихся к ним разделов

*Экономическая часть* *‒ Г. Н. Ярыгина*

*Нормоконтроль* ‒ *А. В. Войтехович*

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

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

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

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

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

1. Примерный календарный график выполнения дипломного проекта

*22 апреля 2020 50% дипломного проекта и пояснительной записки к нему*

*13 мая 2020 80% дипломного проекта и пояснительной записки к нему*

*3 июня 2020 100% дипломного проекта и пояснительной записки к нему*

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

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

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

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

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

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

1. Дата выдачи задания «\_\_23\_\_» \_\_\_\_\_03\_\_\_\_\_\_\_\_ 20 \_*20*\_ г.
2. Срок сдачи законченного дипломного проекта «\_3\_» \_\_\_\_06\_\_\_\_\_ 20 \_*20*\_ г.

**Руководитель** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_О.Н. Забелендик \_\_\_\_\_

(фамилия, инициалы)

**Подпись обучающегося** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_А.С.Яблонский\_\_\_\_\_\_\_\_

(фамилия, инициалы)

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20 \_*20*\_ г.

# РЕФЕРАТ

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ «GOODBOOZE» ПО ПОИСКУ НАПИТКОВ ДЛЯ ПЛАТФОРМЫ ANDROID (КЛИЕНТСКАЯ ЧАСТЬ): дипломный проект / А.С. Яблонский. – Полоцк: ПГУ, 2020, - п.з. - 76 с., 45 рис., 13 табл., 13 источников, графический материал – 4 л. формата А4.

ANDROID-ПРИЛОЖЕНИЕ, СЕРВИС, ПОИСК НАПИТКОВ, АВТОРИЗАЦИЯ, GOODBOOZE, КЛИЕНТСКАЯ ЧАСТЬ, ПЛАТФОРМА-ANDROID

Объектом разработки является создание мобильного приложения Android «GoodBooze» для быстрого просмотра и выбора напитков в незнакомом для пользователя месте, основываясь на рейтинге и комментариях других пользователей.

Цель проекта: разработка Android приложения для поиска напитков: «GoodBooze».

В процессе проектирования выполнены следующие разработки:

* система навигации;
* заглушка реализации сервера;
* подсистема отправки запросов;
* подсистема авторизации;
* подсистема пользователя;
* подсистема модератора.

Областью возможного практического применения программного продукта является использование в незнакомом для человека месте, для изучения местного культурного разнообразия продуктов питания и напитков. Сервис может оказаться полезным и для заведений, помогая им продвигать свой продукт и привлекать новых клиентов. Результат работы имеет перспективные варианты коммерческого развития и привлечения инвестиций; может найти широкое применение как в Республике Беларусь, так и за её пределами.

Студент-дипломник подтверждает, что приведенный в дипломном проекте расчетно-аналитический материал объективно отражает состояние разрабатываемого объекта, все заимствованные из литературных и других источников теоретические и методологические положения и концепции сопровождаются ссылками на их авторов.

# СОДЕРЖАНИЕ

[Введение 7](#_Toc43587104)

[1 АНАЛИЗ ИСХОДНЫХ ДАННЫХ И ПОСТАНОВКА ЗАДАЧИ 9](#_Toc43587105)

[1.1 Описание предметной области 9](#_Toc43587106)

[1.2 Анализ аналогов и прототипов 10](#_Toc43587107)

[1.3 Постановка задач 14](#_Toc43587108)

[1.4 Выбор среды и средств разработки программных средств 15](#_Toc43587109)

[1.4.1 Выбор языка программирование 15](#_Toc43587110)

[1.4.2 Выбор библиотек 16](#_Toc43587111)

[1.5 Разработка технического задания 17](#_Toc43587112)

[1.6 Выводы по первой главе 18](#_Toc43587113)

[2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИя 19](#_Toc43587114)

[2.1 Функциональная структура 19](#_Toc43587115)

[2.2 Варианты использования 20](#_Toc43587116)

[2.3 Проектирование навигации 21](#_Toc43587117)

[2.4 Проектирование заглушки сервера 23](#_Toc43587118)

[2.5 Выбор системы авторизации 24](#_Toc43587119)

[2.6 Обработка сетевых исключений 26](#_Toc43587120)

[2.7 Проектирование моделей данных 28](#_Toc43587121)

[2.8 Выводы по второй главе 30](#_Toc43587122)

[3 РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ 31](#_Toc43587123)

[3.1 Реализация функциональных частей программного обеспечения 31](#_Toc43587124)

[3.2 Стиль пользовательского интерфейса 31](#_Toc43587125)

[3.3 Подсистема авторизации пользователей 32](#_Toc43587126)

[3.3.1 Подсистема навигации 33](#_Toc43587127)

[3.3.2 Обработка авторизованных запросов 34](#_Toc43587128)

[3.3.3 Динамическая загрузка элементов 35](#_Toc43587129)

[3.4 Тестирование и отладка 37](#_Toc43587130)

[3.5 Выводы по третьей главе 40](#_Toc43587131)

[4 ЭКОНОМИЧЕСКАЯ ЧАСТЬ 41](#_Toc43587132)

[4.1 Обоснование необходимости разработки программного обеспечения 41](#_Toc43587133)

[4.2 Этапы разработки программного обеспечения 42](#_Toc43587134)

[4.3 Составление сметы затрат на разработку программного обеспечения 43](#_Toc43587135)

[4.4 Экономический эффект у разработчика ПО 49](#_Toc43587136)

[4.5 Экономический эффект от использования программного обеспечения у пользователя (заказчика) 51](#_Toc43587137)

[4.6 Вывод по четвёртой главе 51](#_Toc43587138)

[ЗАКЛЮЧЕНИЕ 52](#_Toc43587139)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 53](#_Toc43587140)

[ПРИЛОЖЕНИЕ А (обязательное) ТЕХНИЧЕСКОЕ ЗАДАНИЕ 54](#_Toc43587141)

[ПРИЛОЖЕНИЕ Б (обязательное) ОПИСАНИЕ ПРОГРАММЫ 58](#_Toc43587156)

[ПРИЛОЖЕНИЕ В (обязательное) МЕТОДИКА ИСПЫТАНИЙ 61](#_Toc43587164)

[ПРИЛОЖЕНИЕ Г (обязательное) РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 66](#_Toc43587172)

[ПРИЛОЖЕНИЕ Д (обязательное) РУКОВОДСТВО МОДЕРАТОРА 71](#_Toc43587176)

[ПРИЛОЖЕНИЕ Е (необязательное) СОЦИОЛОГИЧЕСКИЙ ОПРОС 73](#_Toc43587180)

# Введение

Путешествия – неотъемлемая часть современного мира. Ежедневно десятки тысяч людей перемещаются между городами, странами и континентами. По данным ООН число совершенных во всем мире в 2019 году международных туристических поездок составило 1,5 млрд, прирост по сравнению с 2018 годом на 4%. Данная статистика говорит о том, туризм стал одной из самых быстрорастущих и стабильных отраслей мировой экономики. Несомненно, одной из причин такого значительного увеличения пассажиропотока стала глобализацией интернета, распространением смартфонов и популяризации социальных сетей.

Интернет неисчерпаемый источник информации и всемирным способом коммуникации. Несмотря на текущие масштабы интернета, тысячи терабайт данных продолжают создаваться ежесекундно. Крупнейшие университеты публикуют научные работы, репортеры готовят новую статью, а начинающий кинорежиссёр загружает дебютную работу на Vimeo. Именно благодаря интернету появилась возможность, не отрываясь от ноутбука, освоить новую специальность или погулять по виртуальному Лувру. А планирование поездок никогда не было таким простым как сегодня. Два клика, ты нашел живописное место, еще пару движений мыши - ты покупаешь билеты по лучшей цене. Выбрать лучший отель помогают многочисленный комментарии и отзывы пользователей со всего мира.

Приводить примеры использования интернета нет смысла, их бесчисленное множество, а основное – они появляются быстрее, чем о них успеваешь писать. По данным глобального отчета Digital за 2020 год количество интернет-пользователей в мире выросло до 4,54 миллиарда, что на 7% больше прошлогоднего значения. И более 5,19 миллиарда человек пользуются смартфонами – прирост на 124 миллиона (2,4%) за 2019 год.

Действительно, смартфон для каждого является персональным проводником во всемирную паутину. Его главное преимущество – он всегда в твоем кармане. Это не только удобно для пользователей, но и разработчиком дает возможность предоставлять пользователям сервисы, которые нужны прямо здесь и сейчас. По данным аналитической компании Gartner в 2019 году операционные системы Android и iOS, увеличили общую долю на рынке до 99,9%. За год было продано 1,32 млрд смартфонов с Android или 85,9% от общего количества и около 215 млн смартфонов на базе операционной системы iOS или 14%.

Изученная статистика все больше доказывает востребованность мобильных приложений, а только растущий пассажира поток   
свидетельствует о возможном спросе на туристические сервисы и приложения. Подтверждениям этой теории является такие крупные приложения, как Aviasales – поиск дешевый авиабилетов, TripAdvisor – крупнейшая в мире платформа для путешественников, или сервисы для поиска жилья в путешествиях – Booking и Airbnb. Основной аудиторией которых являются пользователи, панирующие или находящиеся в путешествии.

Кроме комфортного транспорта в место назначения и теплого ночлега, путешественников несомненно интересует еда и напитки. По всему миру существуют тысячи рецептов, названий блюд и напитков. И этого возникает проблема: как не потеряется во всем разнообразии и выбрать то, и что придется тебе по вкусу?

В рамках данного дипломного проекта будет найден ответ на этот вопрос, путем создания мобильного приложения по поиску напитков для платформы Android. Разрабатываемы программный продукт должен решить проблему поиска напитков и, в перспективе, блюд, во время путешествия или незнакомой для пользователя среде.

# АНАЛИЗ ИСХОДНЫХ ДАННЫХ И ПОСТАНОВКА ЗАДАЧИ

## Описание предметной области

Туризм – одна из самых быстрорастущих отраслей экономики как в Беларуси, так и во всем мире. По данным UNWTO [11] число совершенных во всем мире в 2019 году международных туристических поездок составило 1,5 млрд, что является приростом на 4% по сравнению с 2018 годом. В некоторых случая прирост составил от 10% до 40,2%. Что свидетельствует о непрекращающемся росте туристического потока.

На рынке мобильных платформ на конец 2019 года операционных систем Android является ведущей с рыночной долей 39%, следом идет Windows, которая имеет долю 35%. Третье место занимает iOS - 13,87%. Опирая на данную статистику, можно сделать вывод, что наиболее рациональной для запуска нового продукта является платформа Android. Она позволит добиться наибольшего охвата аудитории при равных затратах в сравнении с другими операционными системами.

Рынок мобильных приложение Android продемонстрировал рост за последний год. По данным App Annie [12], отрасль мобильных приложений создала $41,1 млрд валового годового дохода. Согласно прогнозам Statista [13], в 2020 году валовой годовой доход превысит $189 млрд. А в 2021 году рынок мобильных приложений должен вырасти на 385%.

Данные прогнозы свидетельствует о том, что рынок мобильных приложений в ближайшее будущее будет только расти, а значит будет расти и востребованность в мобильных приложения среди аудитории.

Наиболее популярные приложения можно разделить на следующие категории: социальные приложения, персонализация, оптимизация устройства, про жизнь, сервисы для путешествий и спорта. Сервисы для путешествий и спорта – составляют около 30% процентов от всего множества приложений, согласно отчету App Annie на 2019 год [12].

Для получение достоверных и актуальных данных о востребованности разрабатываемого продукта проведено статистическое исследование. Исследование должно ответить на такие вопросы, как:

* есть ли трудности при выборе блюд питания в незнакомых местах;
* есть ли трудности при выборе напитков в незнакомых местах;
* нужен ли сервис для решения подобных проблема.

Для большего культурного и материального разнообразия опрашиваемых, исследование проводилось в двух группах: среди жителей Франции и среди жителей Беларуси.

В результате исследования было опрошено 90 человек: 66 белорусов в возрасте от 18 лет, 24 француза в возрасте от 18 до 45 лет. Вопросы и статистику ответов можно наблюдать в таблице 1.1.

Таблица 1.1 – Результаты опроса

|  |  |  |  |
| --- | --- | --- | --- |
| **Вопрос** | **Вариант ответа** | **Статистика ответов** | |
| **Беларусь** | **Франция** |
| Испытывали ли вы сложность при выборе выбора блюд питания в незнакомом месте? | Да | 22,7% | 41,7% |
| Нет | 66,2% | 58,3% |
| Другой вариант | 11.1% | 0% |
| Испытывали ли вы сложность при выборе выбора напитков (алкогольных или безалкогольных) в незнакомом месте? | Да | 16,7% | 12,5% |
| Нет | 83,3% | 87,5% |

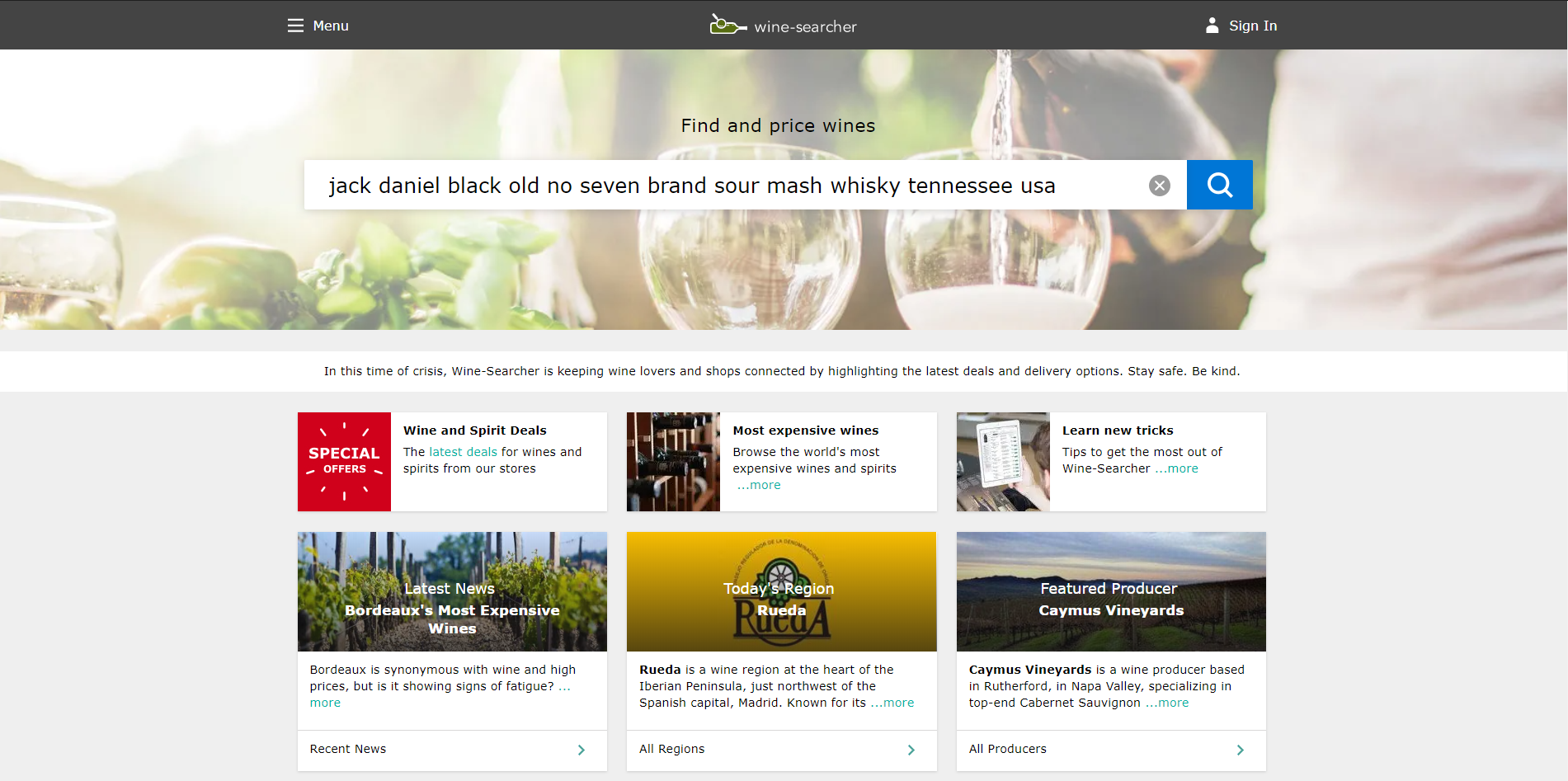
Проведенное исследование и ранее изученная статистика говорит от востребованности и популярности сервисов направленных на решение проблем во время пребывания человека в незнакомой для него среде. Питание является одной из базовых потребностей, что гарантирует непрекращающийся интерес к данной теме. А платформа Android занимает лидирующие позиции на рынки мобильных операционных систем, что может гарантировать максимальный охват аудитории при наименьших затратах на разработку.

## Анализ аналогов и прототипов

Для выявления частичных или полных аналогов разрабатываемого программного продукта был произведен поиск сервисов и приложений. Сравнительный анализ выявил несколько подобных решений, однако ни одно из них не являлась полным аналогом разрабатываемого продукта.

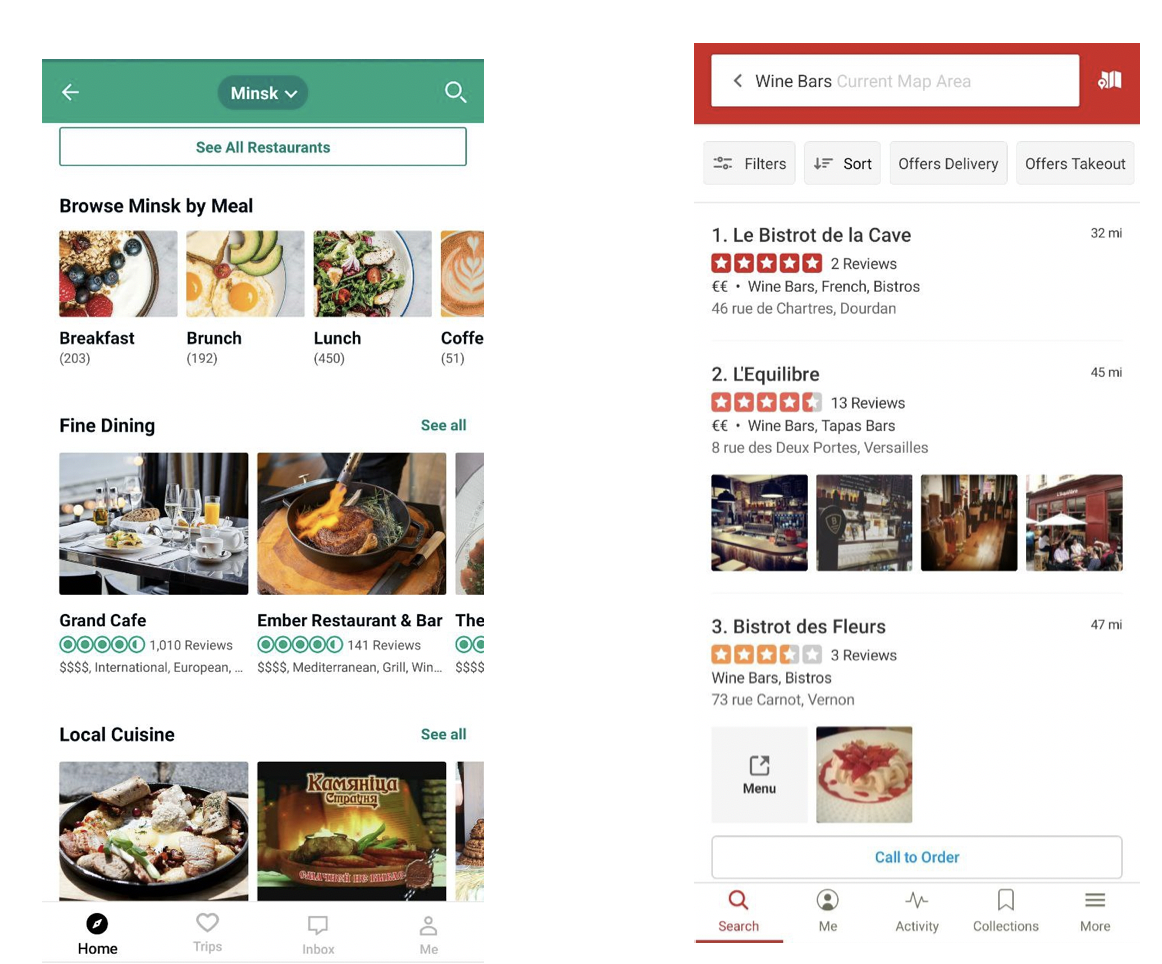
Некоторые из найденных приложений или сервисов ориентированы только на одну продукцию, например вино. Другие же предоставляют список информации о дорогих напитках. Также было отмечено, что многие из существующих приложение созданы для продвижения и рекламы какой-либо марки продукции продукцию.

Большинство найденных сервисов ориентировано на рынок Западной Европы и Америки. Одним из таких является сайт по поиску цен на вина - wine-searcher.com, представлен на рисунке 1.1.

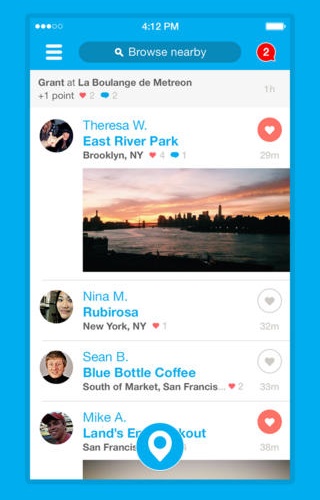


**Рисунок 1.1** – Сайт для изучения цен на вина

Одними из непрямых конкурентов являются сервисы-агрегаторы. К этой категории относятся такие приложения как TripAdvisor, Yelp (рисунок 1.2), Foursquare (рисунок 1.3) и др. На данных и подобных площадках люди добавляют, комментируют и оценивают заведениями, в которых они побывали. Это могут кафе, рестораны, бары, магазины или сувенирные лавки. TripAdvisor также может рекомендовать популярные места и достопримечательности поблизости. Но всё же главной функцией данных приложений является поиск заведений, а не продукции.

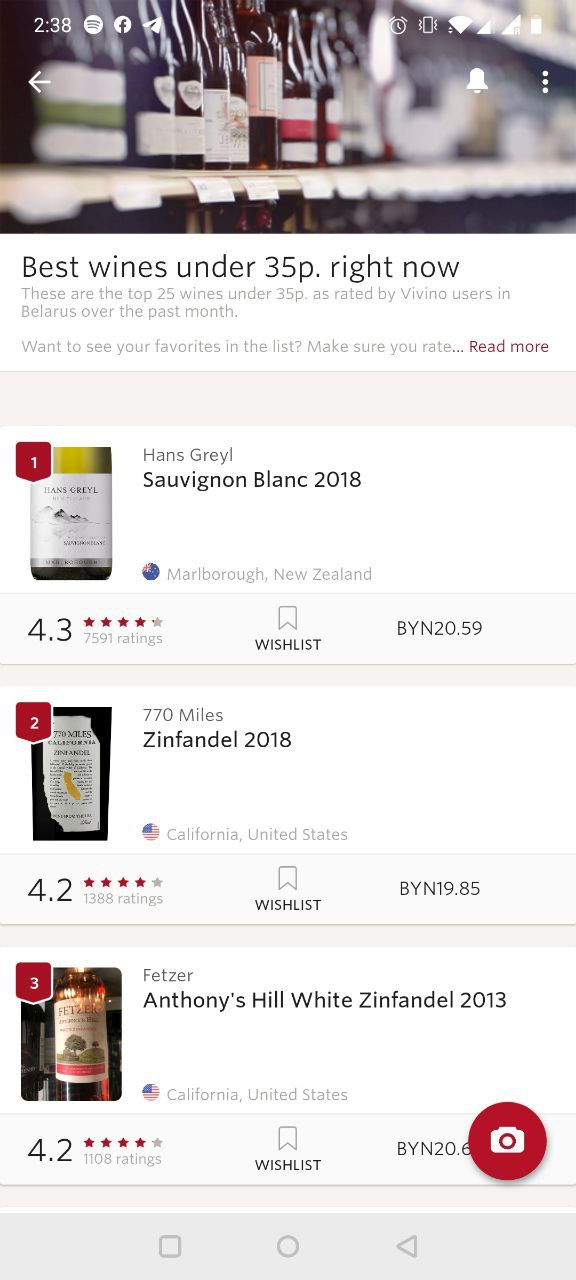
****

**Рисунок 1.2** – Главный экран TripAdvisor и Yelp



**Рисунок 1.3** – Скриншот приложения Foursquare

Наиболее схожим с разрабатываемым продуктом сервисом является Vivino. Vivino предоставляет поиск вин по названию или по заданным параметрам. Сервис существует для платформ Android, iOS, а также в веб-версии. Из описания приложения заметно, что разработчики делают упор на винную продукцию, позволяя при добавлении продукта заполнить такие поля, как: сорт винограда, год сбора урожая, дата разлива и краткое описание конкретного винного производства. Но основной функцией приложения является поиск вина по фотографии или сохраненному изображению этикетки. На рисунке 1.4 находится скриншот главного экрана Vivino.



**Рисунок 1.4** – Скриншот приложения Vivino

Анализ существующего ранка дал представление о конкурентах. Условно, все существующие подобные сервисы и приложения можно разделить на следующие несколько групп.

1. Крупные сервисы-агрегаторы – многофункциональные продукты, предоставляющие общую информацию полезную вовремя путешествие или пребывания в незнакомой среде. Обладают общиной базой данных в различных областях знаний. К минусам подобных приложение можно отнести поверхностное раскрытие предоставляемых тем, из-за их множества.
2. Ориентированные на продукт или узкую категорию продуктов сервисы. Подобные приложения отлично справляются с информированием в своей предметной области, но, как правило, являются достаточно узконаправленными, что значительно уменьшает целевую группу подобных приложений.
3. Рекламные приложения – направлены на продвижение конкретного продукта: напитка, магазина-дистрибьютора, музея и т. д.

Для создания успешного и востребованного продукта пользователя нужно предоставить новый сервис, который ответит на еще нерешенные вопросы пользователей, а также исправит недостатки уже существующих на рынке приложений.

## Постановка задач

Целью выполняемой работы является проектирование и реализация мобильного приложения для платформы Android для добавления и просмотра рейтинга продуктов.

Приложение должно предоставлять возможность добавления нового и просмотра списка из уже добавленных продуктов; просмотра детальной информации, а также комментирования и оценки каждого продукта; просмотра базовой информации о пользователях; редактировании собственного профиля.

Сбор отзывов реальных пользователей приложение является одним из важнейших этапов при запуске продукта. Мнение пользователей позволит увидеть проблемы на начальных этапах развития продукта и скорректировать курс дальнейшей разработки. Для этого стоит сосредоточиться на разработке минимально жизнеспособного продукта. Поэтому в качестве первоначальной категории продуктов, необходимо выбрать достаточно узкое направление, например, напитки. Однако архитектура приложения должна иметь возможность для быстрого масштабирования и на другие виды продуктов.

При разработке клиентского программного обеспечения, чем и является Android приложение, необходимо учитывать зависимость от разработки сервера. Разные сроки выполнения поставленных задач на клиентской и серверной стороне могут привести к общему росту сроков реализации проекта. Для минимизации влияния временных затрат разработки серверной части стоит разработать механизм, способный эмулировать поведение сервера. При этом он должен быть реализован на самом низком уровне взаимодействия клиента и сервера для дальнейшего выполнения качественного тестирования и исключения возможны дефектов.

Одной из основных частей приложения будет являться система авторизации. Она должна быть способной обрабатывать всевозможные исключительные ситуации и предлагать пользователю их решение. К некоторым из ситуаций можно отнести следующие случаи:

* ввод неверных данных авторизации;
* ошибка доступа при отправке авторизованного (приватного) запрос;
* блокирование аккаунта уже авторизованного в приложении пользователя.

Частым способом решения ранее приведенных проблем будет является повторная авторизация в приложении.

Для получения комфортного для пользователя опыта от использования необходимо наличие автоматической навигации на целевой экран. Из этого следует, что система навигации должна быть гибко-настраиваемой и обрабатывать два состояния пользователя: авторизованное и неавторизованное. Иметь возможность быстро и корректно перенаправлять пользователя из авторизованной в неавторизованную и обратно частей приложения. Обратная навигации также должна правильно обрабатывать системное событие – назад, особенно в случае повторной авторизации.

В приложение будут присутствовать текстовые поля ввода данных. В зависимости от вводимых данных значения должны проверяться. В случае ошибки - пользователь должен быть осведомлён о проблеме.

## Выбор среды и средств разработки программных средств

Операционная система Android является целевой платформой разработки программного продукта. Изначально ОС разрабатывалась одноименной компанией – Android Inc., которая в 2004 году была выкуплена компанией Google. Однако первая публичная версия операционной системы официально вышла только в 2008 году.

## Выбор языка программирование

С 2008 года в качестве основного языка программирования мобильных приложений под платформу Android использовался – Java. Спустя десять лет, в 2017 году, Google заявила о переходе на Kotlin, как на основной язык разработки приложений. Несмотря на это заявление, согласно официальной статистике Google, Java и в настоящее время остается самым популярным языком программирования для операционной системы. Однако эта статистика обусловлена огромным количеством приложений, опубликованных до 2017 года.

Язык Kotlin – молодой в сравнении с Java язык. Был разработан в российской компании JetBrains в 2011 году. Как и Java, C и C++, Kotlin – это статически типизированный язык. Он поддерживает как объектно-ориентированное, так и процедурное программирование.

Сегодня у Android разработчиков все ещё остаётся возможность выбора основного языка разработки приложения. Более того, Java и Kotlin могу быть использованы в рамках одного проекта и даже внутри одного пакет. А написанные ранее Java библиотеки полностью совместимы с проектом на Kotlin. Такая глубокая обратная совместимость достигается за счет компиляции Kotlin кода в Java Virtual Machine байткод. Соответственно и исполнение может производиться на той же JVM, что и Java кода.

В качестве основного языка разработки продукта будет использован Kotlin. Выбор обусловлен официальными рекомендациями Google для Android разработчиков. Помимо этого, в проекте будут использованы некоторые сторонние библиотеки и утилитные классы, написанные ранее на языке Java.

Согласно официальным рекомендациям в качестве системы конфигурации и сборки проекта будет использована Gradle.

Для разработки Android приложений может быть применена любая интегрированная среда разработки, поддерживающая работу с Java или Kotlin. Но наиболее удобной является Android Studio. Ядро данной среда было основано на разработке компании JetBrains, что может гарантировать высокую производительности при использовании с языком Kotlin. Кроме этого, в среде разработки реализован удобный интерфейс и интегрирован редактор XML разметки; специализированный редактор кода, поддерживающий синтаксис языков Java, Kotlin, Groove, XML и др.; менеджер SQL Lite баз данных; отладчик; менеджер системы контроля версий и другие полезные для программиста инструменты.

## Выбор библиотек

Операционная систем Android предоставляет разработчика достаточно низкоуровневый контроль над устройство. Начиная с получения информации от встроенных датчиков и камер, заканчивая изменением интерфейса при изменении угла закрытия гибкого экрана. Это дает практически неограниченную возможность реализации задуманных идей. Но востребована такая детализация внутренних программных интерфейсов не всегда. Например, нет необходимости для каждого HTTP запроса создавать сокет для отправки запроса или выставлять вручную настройки выдержки и ISO камеры опираясь на данный об экспозиции.

Еще одним хорошим пример является построение интерфейса. Система позволяет прорисовывать интерфейс с точностью до пикселя, однако такие возможности будут только препятствовать реализации полей ввода и кнопок согласно общим требования к дизайну платформы.

Для упрощения работы с данными программными интерфейсами лучшим решением будет прибегнуть к использованию фасада, которые скроет детали взаимодействия и предоставит нам только необходимый функционал. В большинстве случаем данные фасады уже разработаны и представлены в виде библиотек.

В списке приведены необходимые для быстрой и комфортной разработки приложения библиотеки.

1. Volley by Google – позволяет управлять обработкой и кэшированием сетевых запросов. Предоставляет удобный API для отмены и динамического изменения запроса; инструменты отладки и трассировки.
2. Kotlin Coroutines – предоставляет гибкую возможность реализации асинхронно выполняемого кода. В отличии от встроенных в Java средств, позволяет писать асинхронный код в стиле классического синхронного за счёт использования async-await синтаксиса.
3. AndroidX AppCompact – библиотека стилей и компонентов стандартных элементов пользовательского интерфейса Android. Предоставляет методы рисования интерфейса совместимые со старыми версиями операционной системы.
4. Glide – утилитная библиотека предоставляющая интерфейс для загрузки, кэширования и трансформации медиа ресурсов. Имеет плагин для интеграции с Volley.

## Разработка технического задания

После выполнения анализа и изучения предметной области сформулированы следующие требования к функциональности разрабатываемого мобильного приложения для устройств Android.

Приложение должно обрабатывать два состояния пользователя: авторизованное и неавторизованное. Позволять в любой момент выполнения перемещаться из авторизованной части в неавторизованную, для повторного ввода данных, а после возвращать в исходное место авторизованной части.

Приложение должно предоставлять пользовательский функционал, перечисленный далее.

1. Авторизацию пользователей по логину и паролю; оба поля являются обязательными.
2. Отображение общей ленты продуктов.
3. Добавление нового продукта в общую ленту с указанием следующих данных:
4. название продукта;
5. рейтинг (количество оценок);
6. стоимость;
7. описание;
8. место приобретения;
9. url адрес на изображение.
10. Отображение деталей для каждого продукта.
11. Отображение комментариев для каждого продукта.
12. Возможность добавить новый комментарий продукту.
13. Возможность оценить продукт.
14. Экран профиля пользователя со следующей информацией:
15. имя;
16. email;
17. изображение профиля;
18. дата рождения;
19. количество добавленных продуктов;
20. дата рождения;
21. краткая биография.
22. Экран просмотра текущего профиля с возможностью редактировать следующие поля:
23. имя (обязательное поле);
24. url адрес на изображение профиля (необязательное поле);
25. email (обязательное поле);
26. краткая биография (необязательное поле).

Кроме основного функционала необходимо реализовать общие компоненты, такие как:

* индикатор загрузки;
* экран отображения ошибки.

В случае исключительной ситуации во время отправки любого из запросов к серверу, приложение должно корректно обрабатывать её и отображать пользователю сообщение об ошибки. Предоставлять пользователю возможность повтора, если это возможно.

Интерфейс приложения должен быть локализован на два языка – русский и английский. Производить автоматический выбор локализации в зависимости от языка системы.

Расширенное техническое задание на разрабатываемое программное обеспечение приведено в приложении А.

## Выводы по первой главе

В результате анализа выделены основные цели дипломного проектирования, поставлены задачи для реализации программного продукта. Проанализировано состояние рынка предметной области и выявлена востребованность продукта. Изучены сильные и слабые места схожих сервисов. Осуществлен выбор программных средств и инструментов для реализации мобильного приложения. Разработано расширенное техническое задание.

# ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИя

## Функциональная структура

Основываясь на описанных характеристиках и представленных требований технического задания. Необходимо спроектирована структура базовых компонентов разрабатываемого мобильного приложения.

В любом Android приложение базовый класс должен наследовать реализацию класса Application. Единовременно может существовать только один экземпляр-наследник. Расширенная реализация Application – основной контейнер всех компонентов приложения.

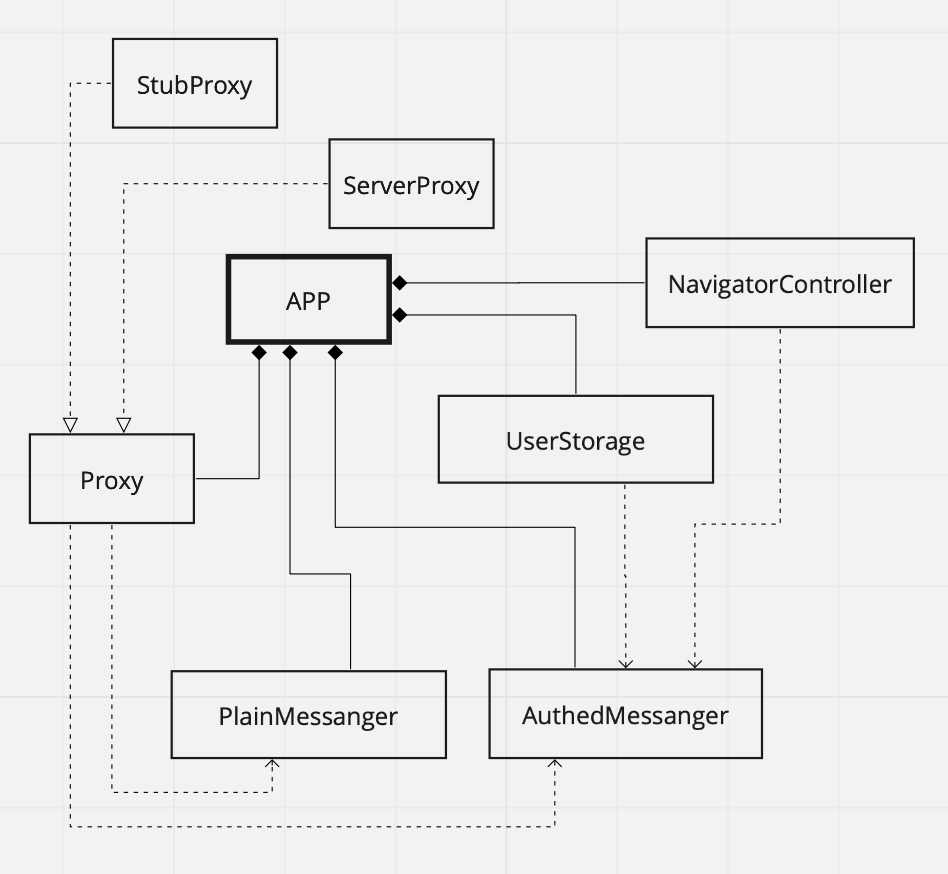
Для реализация поставленных задач требуется разработать следующие компоненты.

1. В конечном продукте планируется я использовать множество экранов. Для их смены, передачи параметров и управлением историей переходов необходим модель навигации – NavigationController. Кроме навигации между экранами, контроллер должен предоставлять способ переход между авторизованной и не авторизованной частью приложения.
2. С определенной точки зрения приложение является посредником для взаимодействия с серверной логикой. Следовательно, необходим интерфейс для выполнения клиент-серверного взаимодействия путем отправки запросов – Proxy. Для каждого запроса должна быть возможность установки следующих параметров:
   * 1. endpoint-адрес;
     2. типа запроса (POST, GET, PUT, DELETE, и др.);
     3. заголовок;
     4. параметры;
     5. метода обработки ответа.

Для уменьшения затрат на разработку мобильного приложения также необходима лже-реализация данного интерфейса, использующая локальный подменный сервер – StubProxy.

1. При использовании одного сервера в различных запросах множество могут повторяться. Например, настройки локализации, данные авторизации, логика обработки ошибок. Для предотвращения дублирования кода необходимо инкапсулировать общие параметры запросов. Эту задачу предстоит выполнить классу PlainMessanger.
2. Для реализации возможности авторизации и исключения необходимости ввода пользовательских данных при каждом перезапуске требуется реализация защищенного хранилища сессии – UserStorage.
3. Кроме хранения данных авторизации, для безопасной системы авторизации необходима система отправки защищенных запросов c подстановкой и, при необходимости, обновлением токенов авторизации авторизации. В случае отсутствия или истечения сроков годности токенов, компонент должен востребовать повторную авторизацию пользователя. Реализация данного компонента будет скрыта в классе AuthedMessanger.

Из разработанных ранее компонентов была смоделирована функциональная структура, приведенная на рисунке 2.1.



**Рисунок 2.1** – Функциональная структура

## Варианты использования

Для описания возможных вариантов использования во время создания программного продукта применяется диаграмма использования. Она является исходным концептуальным представлением в процессе её проектирования и разработки.

Целью данной диаграммы является представление проектируемой системы в виде множества сущностей или актёров, использующих варианты использования для взаимодействия с системой. Каждый вариант использования определяет набор некоторых действий, совершаемый системой при взаимодействии с актёром.

Между компонентами диаграммы вариантов использования могут существовать различные отношения, которые описывают взаимодействие экземпляров одних актёров и вариантов использования с экземплярами других вариантов или актёров.

Для выделения актёров и дальнейшего построения схем взаимодействия между ними рассмотрим типы пользователей.

Рядовой пользователь – как правило, человек находящийся в незнакомой или новой для себя среде, желающий ознакомиться с продуктами доступными поблизости, или поделиться отзывом о опробованном продукте. Доступные действия:

1. просмотра списка добавленных продуктов;
2. добавления нового продукта;
3. просмотра информации о продукте;
4. оценки продукта;
5. просмотра комментариев других пользователей о каждом продукте;
6. добавление комментария;
7. редактирование профиля;
8. удалением собственных продуктов.

Второй тип пользователей – модератор. Для него доступны все те же действия, что и для рядового пользователя, но в добавок, он обладает правами на просмотр списка продуктов отправленных на модерацию. Для продуктов из данного списка доступный действия одобрения и отклонения. При просмотре комментариев к продукту, модератор также может удалять комментарии пользователей. «Модератор» является потомком «рядового пользователя».

## Проектирование навигации

Навигация валяется ключевым звеном при разработке многоэкранного мобильного приложения. Также она оказывает большое влияние на комфорт от использования. Корректно спроектированная и реализованная модель навигации должна быть интуитивно понятной для пользователя, и на подсознательно уровне отвечать вопросы о нахождении необходимого функционала.

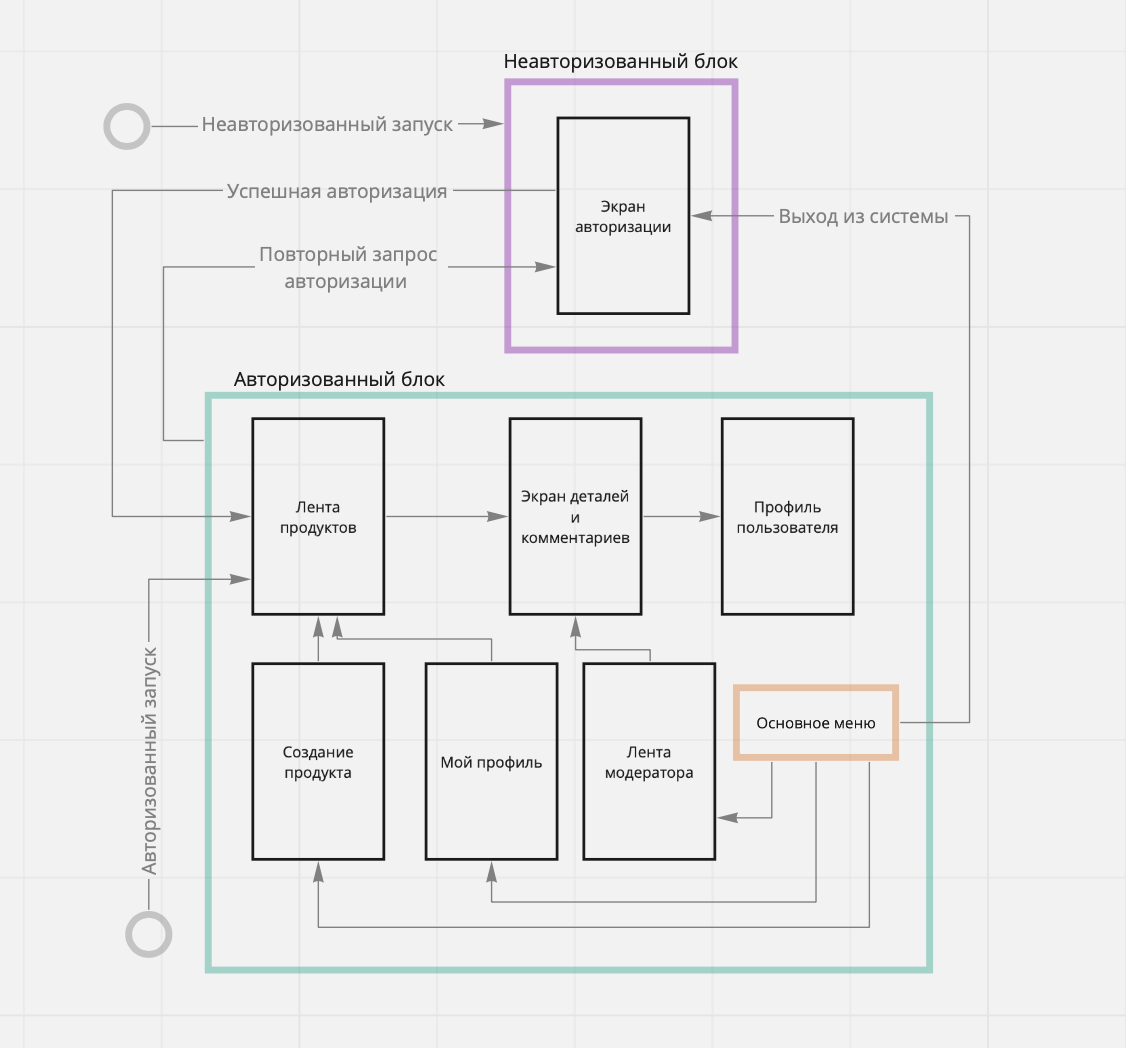
При реализации навигации между экранами на платформе Android необходима корректная обработка основных системных событий:

1. запуск приложения из меню приложений;
2. запуск приложения из окна «поделиться»;
3. открытие уже запущенного приложения;
4. обработка действия «назад»;
5. пересоздании приложение при нехватки системных ресурсов;
6. сбросе настроек приложения.

Из описания технического задания (см. приложение А) следует, что пользователь может находиться в двух состояниях: авторизованном и неавторизованном. Система навигации, кроме последовательного перемещения между экранами, должна иметь возможность переключения между блоками: при авторизации пользователя, в случае необходимости повторного входа, или при выходе пользователя из системы. При этом история навигации в каждом блоке должна храниться опосредованно для предсказуемого поведения.

Для объединения доступных для пользователя экранов и переходов необходима реализация общего меню, доступного с разных экранов приложения.

Согласно выдвинутым требования к механизму межэкранной навигации разработана схема навигационной модели мобильного приложения – рисунок 2.2.



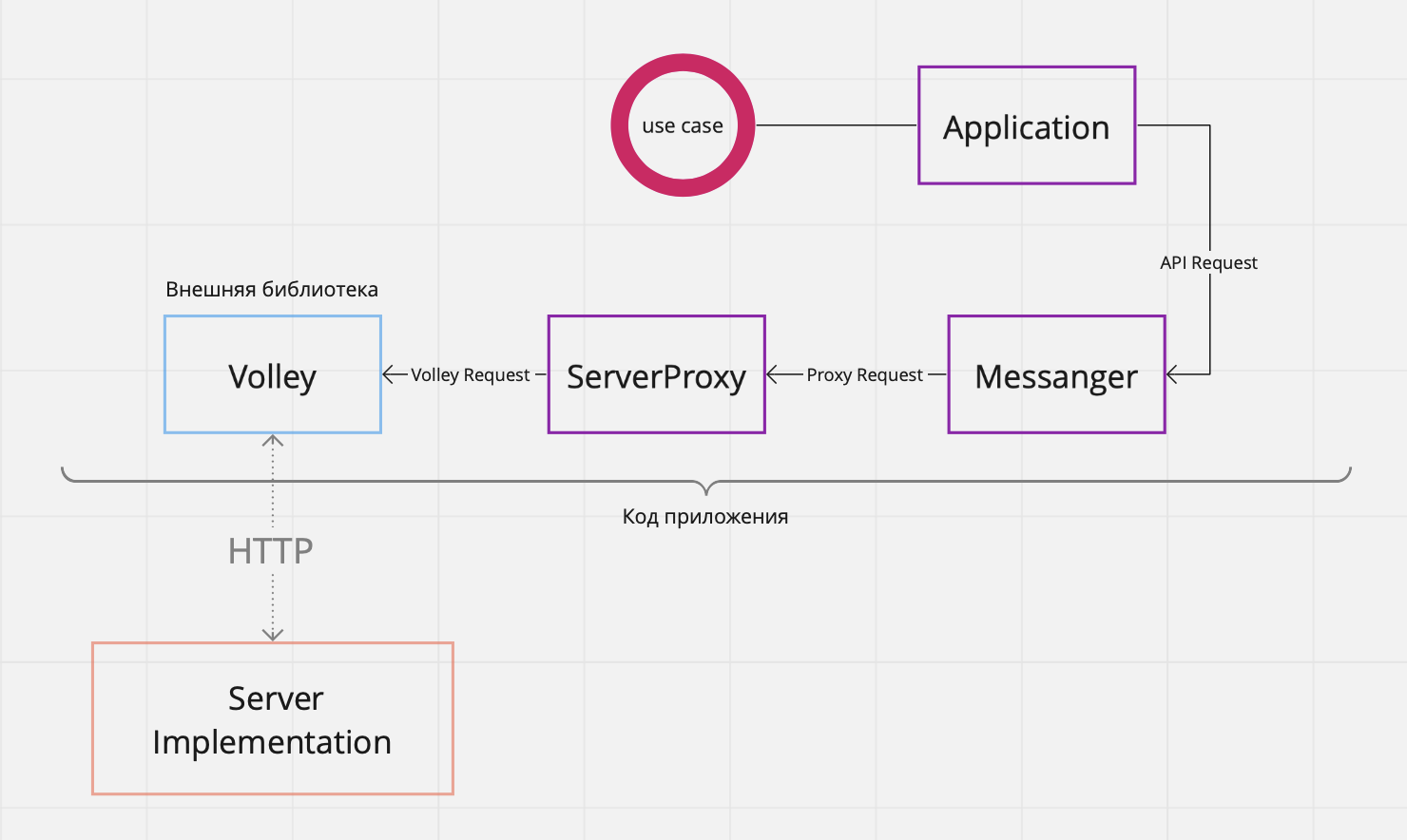
**Рисунок 2.2** – Навигационная модель приложения

## Проектирование заглушки сервера

Во время разработки и отладки приложения зачастую возникает необходимость отладить систему при определённом ответе сервера. Но быстро изменить реализацию сервера и подменить возвращаемые данные не всегда представляется возможность. В другом случае разработка отдельного функционала серверной части может задержаться и заблокировать доработку приложения. Для избежание трудностей в подобных ситуация следует спроектировать заглушку сервера на стороне приложения.

Основным требование к проектированию данного программного функционала является минимизация отличий от реализации истинного механизма коммуникации с сервером. Следование данного правила позволит исключить ошибки в реальной реализации при тестировании функционала через заглушки. Выполнение требований по реализации заглушки сервера может быть достигнуто путем использования идентичных интерфейсов классов для заглушки и истинного механизма.

На рисунке 2.3 приведена схема последовательности вызовов классов, при отправке запроса. Из схемы видно, что крайней доступной для модификации точкой приложения является класс ServerProxy. Его реализацию и следует заменять заглушкой сервера.

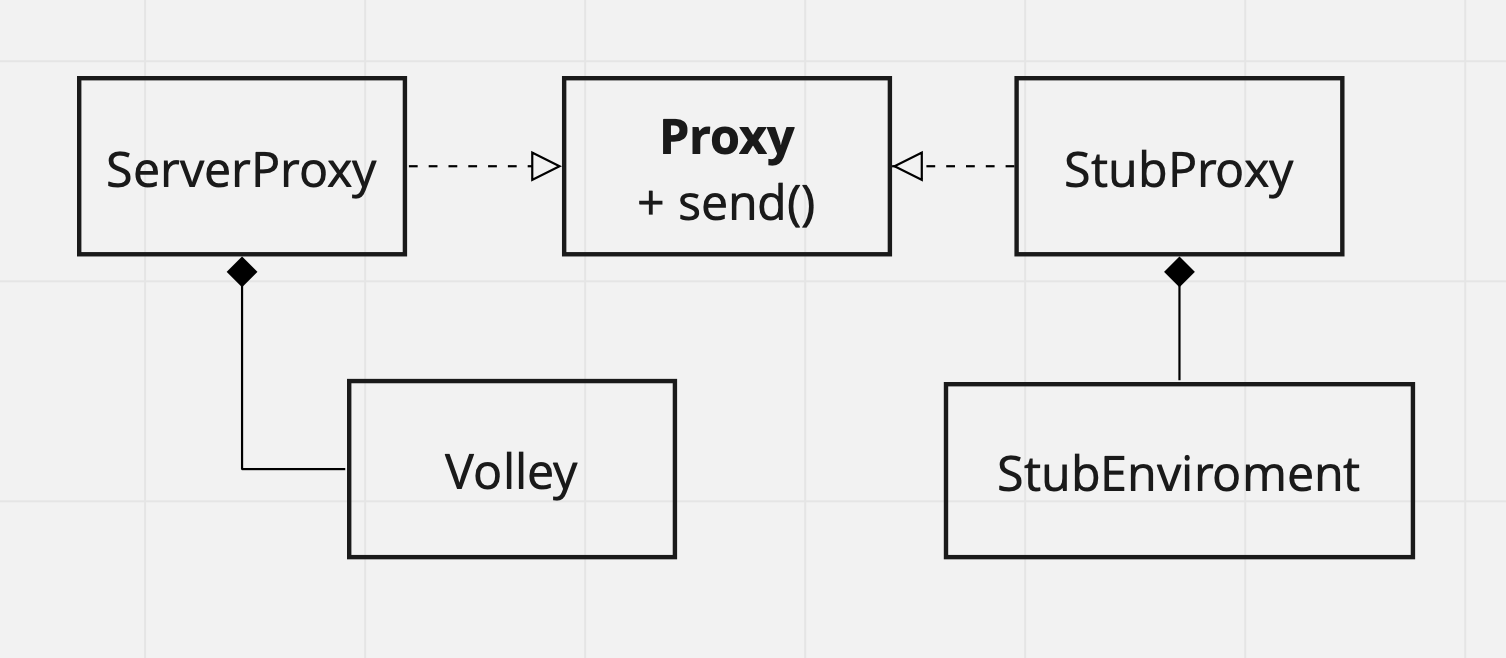


**Рисунок 2.3** – Последовательность вызова при отправке запроса

В ходе проектирования был разработан программный интерфейс Proxy описывающий метод взаимодействия с сервером. ServerProxy является истинной реализацией этого интерфейса через библиотеку Volley для коммуникации с сервером.

Второй реализацией интерфейса Proxy является заглушка – StubProxy. Данный класс перенаправляет все запросы на локальную среду обработки – StubEnviroment, которая полностью независима от сети.

Схема интерфейса и его реализаций приведена на рисунке 2.4.



**Рисунок 2.4** – Реализации интерфейса Proxy

## Выбор системы авторизации

Для клиент-серверных приложений существует большое множество способов реализации системы авторизации. Наиболее популярными механизмами считаются OAuth, JWT, Server Side Session и OpenId.

1. OAuth – протокол авторизации, позволяющий выдать одному сервису (приложению) права на доступ к ресурсам пользователя на другом сервисе. Протокол избавляет от необходимости доверять приложению логин и пароль, а также позволяет выдавать ограниченный набор прав, а не все сразу. Протокол основан на использовании базовых веб-технологий: HTTP-запросах, перенаправления и т. п. За счет это использование OAuth возможно на любой платформе с доступом к интернету или браузеру: на сайтах, в мобильных и desktop-приложениях, плагинах для браузеров.
2. JWT (веб-токен JSON) – представляет собой стандартизированный, в некоторых случаях подписанный и/или зашифрованный, формат упаковки данных, который используется для безопасной передачи информации между двумя сторонами. Токен JWT состоит из трех частей: заголовка, полезной нагрузки и подписи или данных шифрования. Первые два элемента — это JSON объекты определенной структуры. Третий элемент вычисляется на основании первых и зависит от выбранного алгоритма (в случае использования не подписанного JWT может быть опущен.
3. Server-Side Session – подход, при котором сервер отвечает за создание сеанса для конкретного пользователя, когда пользователь входит в систему, после чего идентификатор сеанса сохраняется в файле cookie в клиенте пользователя. Для каждого запроса, отправленного пользователем, также будет отправляться файл cookie, где сервер может сравнивать идентификатор сеанса из файла cookie с информацией о сеансе, хранящейся на сервере, так что идентификация пользователя проверяется.
4. OpenId - открытый стандарт децентрализованной системы аутентификации, предоставляющей пользователю возможность создать единую учётную запись для аутентификации на множестве не связанных друг с другом  интернет-ресурсов, используя услуги третьих лиц. Базовой функцией OpenID является предоставление портативного, клиент-ориентированного, цифрового идентификатора для свободного и децентрализованного использования.

При разработке программного продукта по указанным требованиям и выдвинутому техническому заданию, наиболее предпочтительным, как для серверного разработчика, так и для мобильного, является использование технологии JWT.

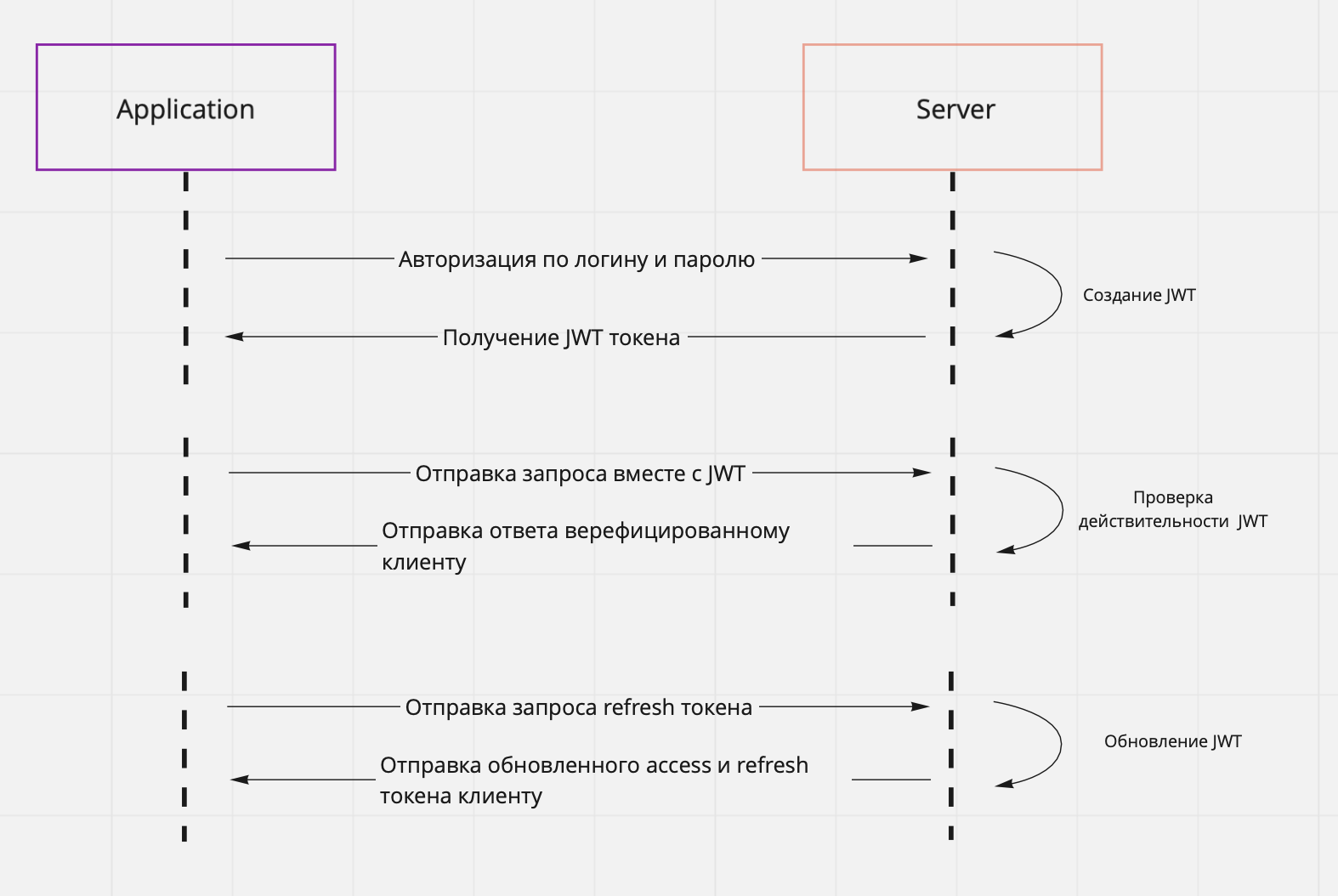
JSON Web Token является современным подходом, используемым в новых веб-приложениях и для мобильных устройств. При таком подходе пользовательское состояние не сохраняется на сервере, а сохраняется в токене.

JSON токена имеет ряд преимуществ в сравнении с другими механизмами.

1. Система легко масштабируется в зависимости от размера приложения, поскольку токен не имеет состояния и не хранит пользовательскую информацию.
2. Безопасность за счёт использования локального хранилища и короткого времени жизни токена; использовании HTTPS и SSL для передачи данных между клиентом и сервером.
3. Снижение нагрузки на сервер путём переноса создания клиентских токенов на внешние сервисы.
4. JWT делает возможным предоставление одновременного доступа к различным доменам и сервисам.

Как правило, при использовании JSON-токенов в клиент-серверных приложениях реализуется следующая схема (рисунок 2.5):

1. клиент проходит [аутентификацию](https://ru.wikipedia.org/wiki/%D0%90%D1%83%D1%82%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F) в приложении (к примеру, с использованием логина и пароля);
2. в случае успешной аутентификации сервер отправляет клиенту access- и refresh-токены;
3. при дальнейшем обращении к серверу клиент использует access-токен. Сервер проверяет токен на корректность и предоставляет клиенту доступ к ресурсам;
4. в случае, если access-токен становится не валидным, клиент отправляет refresh-токен, в ответ на который сервер предоставляет два обновленных токена;
5. в случае, если refresh-токен становится не валидным, клиент опять должен пройти процесс аутентификации.



**Рисунок 2.5** – Реализация JWT

## Обработка сетевых исключений

При использовании любого приложения могут возникать исключительные ситуации: ошибка чтений данных из хранилища, некорректные пользовательские данные, непредвиденное завершение Activity и др. В клиент-серверном приложении основной и наиболее частой причиной возникновения ошибок будет является проблема связи с сервером. Для комфортного использования приложения необходима реализация механизма обработки подобных ситуаций.

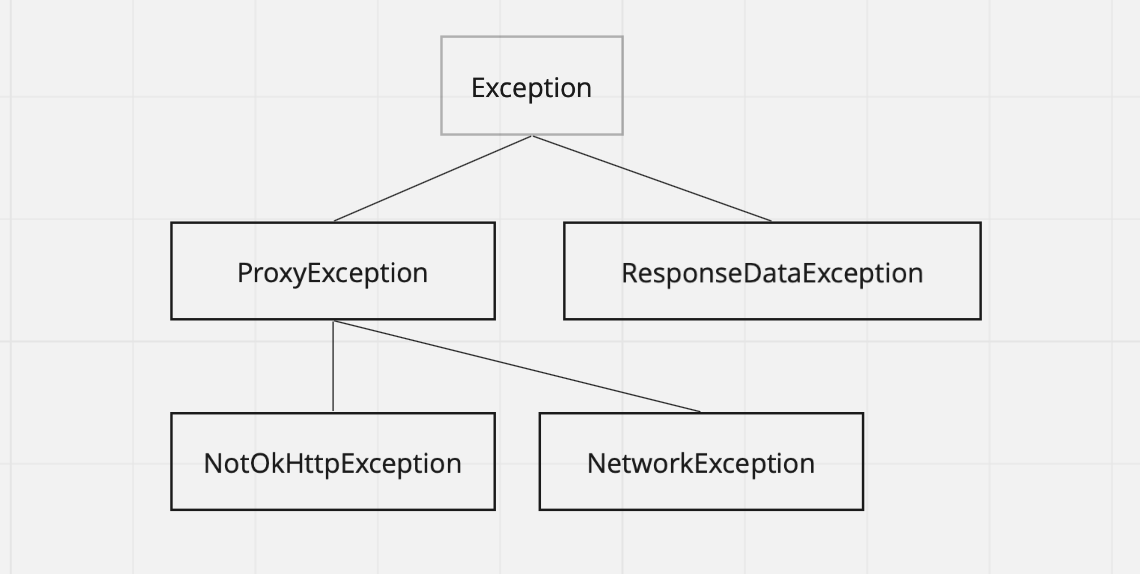
Для проектирования системы обработки ошибок необходимо рассмотреть возможные места и причины возникновения исключений при коммуникации с сервером. Схема последовательности вызовов классов при отправке запроса изображена на рисунке 2.3.

1. Ошибка сети – ситуация вызвана полным отсутствием или нестабильным соединяем с сетью. Тип данных исключений будет зависеть от способа реализации сетевой коммуникации. В разрабатываемом приложении за это отвечает сторонняя библиотека Volley. Для унифицирования и последующей обработки подобных ошибок необходимо конвертировать их в типизированный вид исключения – ProxyNetworkException на уровне Proxy.
2. Ошибка HTTP запроса. Статус завершения HTTP запроса описывается трёхзначным числом в диапазоне 100 – 500. В случае успешного выполнения, статус ответа равен 200.   
   Далее приведет список наиболее часто встречающихся статусов HTTP запроса.
   1. 400 – некорректный запроса.
   2. 404 – не найдено.
   3. 407 – необходима аутентификация.
   4. 429 – слишком много запросов.
   5. 499 – соединение закрыто.
   6. 500 – внутренняя ошибка сервера.
   7. 503 – сервис недоступен.
   8. 504 – шлюз не отвечает.
   9. 520 – неизвестная ошибка.

Данный тип ошибок должен быть конвертирован в исключение NotOkHttpException на уровне реализации Proxy.

1. Ошибка обработки ответа – последний тип ошибки, способный возникнуть при невозможности клиентом обработать успешный ответ сервера – ResponseDataException. В спроектированной схеме сетевого взаимодействия, обработка ответа производится на уровне реализации Messanger.

Основываясь на выявленных сетевых исключениях, разработана иерархия сетевых ошибок, изображённая на рисунке 2.6.



**Рисунок 2.6** – Иерархия сетевых ошибок

Как видно на рисунке 2.6 все сетевые ошибки делятся на две группы: ошибка данных и ошибка соединения. Для обработки ошибок был разработан интерфейс ExceptionHandler. Реализация данного интерфейса обязана конвертировать исключение в понятное для пользователя описание ошибки. Предполагается использование данного интерфейса на уровне реализации Messanger.

## Проектирование моделей данных

Взаимодействие с моделями данных является ключевой частью при реализации экранов приложения. Данный могут отображаться в виде простого текста, индикатора, списка или же определять поведение навигации и приложения в целом.

В проектируемом клиент-серверном приложение большинство информации необходимо получать от сервера, следовательно, нужна конвертация серверной модели данных в модель данных приложения. Реализация данного преобразования будет скрыта в конкретном классе запроса. А при необходимости создания структуры данных из результатов нескольких запросов может быть использована нестандартная реализация Messanger или паттерн «репозиторий».

Первой модель с которой придется столкнуться при разработке программного продукта является модель пользователя. Она должна существовать и быть доступной разработчику на всём протяжении нахождения пользователя в авторизованной части приложения. Описание модели пользователя представлено в таблице 2.1.

Таблица 2.1 – Модель авторизованного пользователя

|  |  |
| --- | --- |
| **Название поля** | **Тип данных** |
| Id | String |
| Name | String |
| Second Name | String |
| Login | String |
| Email | String |
| AccessToken | Token |
| RefreshToken | Token |
| IsModer | Boolean |

Модель объекта Token представлена в таблице 2.2.

Таблица 2.2 – Модель Token

|  |  |
| --- | --- |
| **Название поля** | **Тип данных** |
| Token | String |
| ExpiredDate | Long |

Объектами главного экрана в авторизованной части являются продукты. Каждый продукт содержит собственную информацию и список комментариев. Описание модели продукта представлено в таблице 2.3.

Таблица 2.3 – Модель продукта

|  |  |
| --- | --- |
| **Название поля** | **Тип данных** |
| Id | String |
| Title | String |
| CreateDate | Long |
| ImageId | String |
| Address | Location |
| Price | Double |
| UserId | String |
| Likes | Int |
| IsLiked | Boolean |
| Description | String |
| Comments | List<Comment> |

Кроме стандартного перечня типов данных, платформа Android множество сторонних, например, тип Location. Объект класса Location позволяет хранить данные о местоположении: координаты и текстовое представление адреса. Кроме того, объекты данного класса могут быть использованы в сторонних библиотеках Google, таких как MapsAPI.

Модель комментария необходима для отображения отзывов пользователей для каждого продукта. Поля модели описаны в таблице 2.4.

Таблица 2.4 – Модель комментария

|  |  |
| --- | --- |
| **Название поля** | **Тип данных** |
| Id | String |
| UserId | String |
| UserName | String |
| Date | Long |
| ImageId | String |
| Text | Text |

Для реализации экранов просмотра профилей пользователей и экрана редактирования собственного пользователя необходима модель данного объекта. Описание подели пользователя находится в таблице 2.5.

Таблица 2.5 – Модель пользователя

|  |  |
| --- | --- |
| **Название поля** | **Тип данных** |
| Id | String |
| Name | String |
| Email | String |
| ImageId | String |
| Birthday | Long |
| Bio | String |
| UserId | String |
| ProductsCount | Int |
| OriginLocation | Location |

Модели некоторых экранов, например, продукты в ленте обычного пользователя и модератора, идентичны. Что позволяет экономить ресурсы, требуемые приложению.

## Выводы по второй главе

В результате проектирования программного обеспечения осуществлён анализ предметной области; проектирование основных компонентов; выделение требуемого набора сущностей. На основании поставленной задачи по реализации функционала сформирована схема взаимодействия основных компонентов и сущностей. Спроектированы системные механизмы для разработки и отладки. Рассмотрены основные моменты создания мобильного Android приложения.

# РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ

## Реализация функциональных частей программного обеспечения

В данном пункте представлены результаты разработки основных функциональных частей, реализованных в составе Android приложения «GoodBooze». Описание результатов приводится вместе с описанием интерфейса программного обеспечения.

## Стиль пользовательского интерфейса

Интерфейс элементов и приложения был реализован согласно Material Design рекомендациям от Google.

Стиль Material Design гласит о использовании в приложении ограниченного набора цвета и оттенков. В качестве цветовой схемы были выбраны цвета, приведенные в таблице 3.1

Таблица 3.1 – Цветовая схема приложения

|  |  |  |
| --- | --- | --- |
| **Назначение цвета** | **HEX код** | **Цвет** |
| Primary | #3b8ea5 |  |
| Secondary | #2d728f |  |
| Accent | #f49e4c |  |
| Accent Light | #f5ee9e |  |
| Accent Dark | #ab3428 |  |
| Text Primary | #212121 |  |
| Text Primary Light | #ffffff |  |
| Text Secondary | #9d9d9d |  |
| Text Secondary Light | #a6ffffff |  |

Для правильного воплощения концепции Material Design необходимо описание общих стилей компонентов. В реализованном мобильном приложении созданы следующие стили:

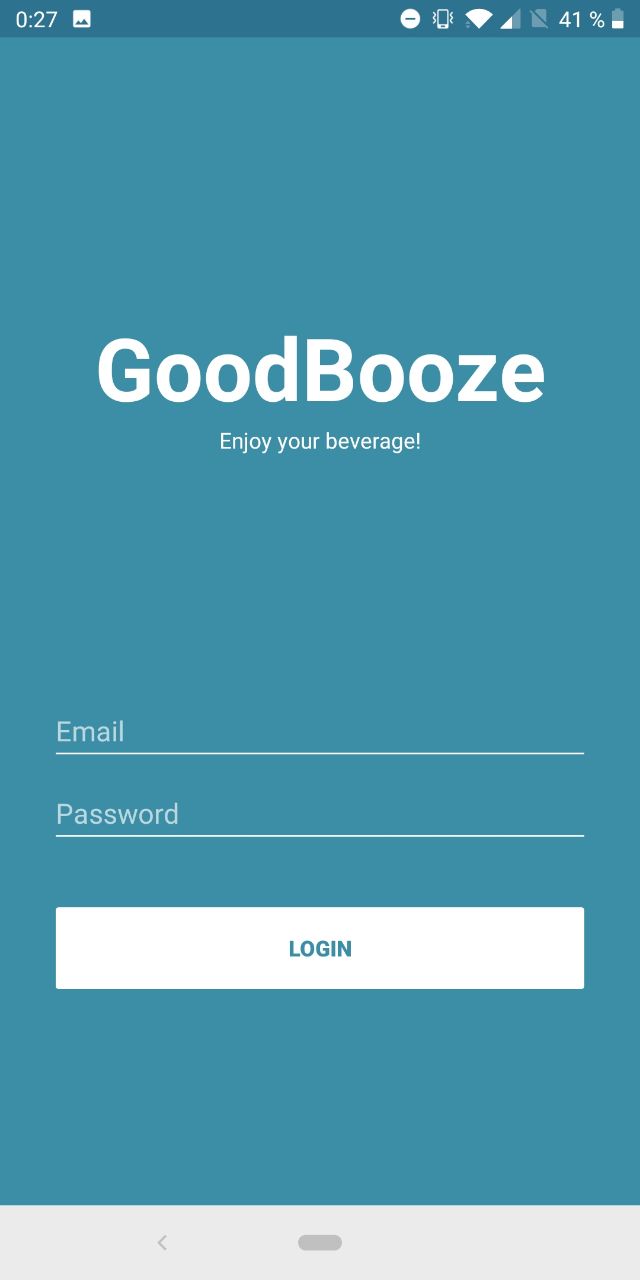
* главная кнопка;
* вторичная кнопка;
* главный текст;
* вторичный текст;
* поле ввода.

Большинство стилей были описаны в двух вариантах: для использования в светлой и темной части интерфейса. На рисунке 3.1 и рисунке 3.2 изображены два варианта отображения главной кнопки.

|  |  |
| --- | --- |
|  |  |
| **Рисунок 3.1** – Стиль главной кнопки для тёмного дизайна | **Рисунок 3.2** – Стиль главной кнопки для светлого дизайна |

## Подсистема авторизации пользователей

При первом запуске приложения или в других случаях отсутствия данных авторизации пользователя, открывается экран входа, представленный на рисунке 3.3.



**Рисунок 3.3** – Экран авторизации

Кроме ранее перечисленных случаев, пользователь может быть перенаправлен на этот экране, в случае неактуальность JWT токена или выхода из системы для повторной аутентификации.

Пользователю необходимо ввести данные входа: логин и пароль; после нажать кнопку входа. Во время обработки данных отображается индикатор загрузки – рисунок 3.4. Если данные пользователя не найдены или некорректны, будет отображена ошибка, представленная на рисунке 3.5.

|  |  |
| --- | --- |
|  |  |
| **Рисунок 3.4** – Экран авторизации: индикатор загрузки | **Рисунок 3.5** – Экран авторизации: некорректные данные входа |

В случае успешного входа пользователь перенаправляется на экран ленты продуктов.

## Подсистема навигации

Для быстрого и удобного перехода между экранами была реализована система навигации. Видимыми для пользователя компонентами системы навигации является: боковое (главное) меню и заголовок окна с кнопкой навигации.

Меню доступно только на главном экране авторизованной части приложения. Содержит в себе краткую информацию о авторизованном пользователе и основные действия перехода между экранами. На рисунке 3.6 и 3.7 изображено меню в открытом виде для рядового пользователя и модератора соответственно.

Кнопка навигация присутствует на каждому экране с заголовком в авторизованной части.

|  |  |
| --- | --- |
|  |  |
| **Рисунок 3.6** – Основное меню | **Рисунок 3.7** – Основное меню: модератор |

## Обработка авторизованных запросов

В выбранном механизме авторизации безопасность достигается за счёт использовании двух видов токена с разными времени жизни: access и refresh. Access токен имеет короткое время жизни и, как правило, используется для одного или нескольких запросов. Для обновления Access токена в классе AuthedMessanger был реализован интерфейс Messanger, который перед отправкой основного запроса, обновит токен. Отрывок реализация с основной логикой приведен в листинге 3.1.

Листинг 3.1 – Отрывок реализации AuthedMessanger

1. @Suppress("MoveVariableDeclarationIntoWhen", "LiftReturnOrAssignment")
2. override suspend fun send(): MessageResult<D> {
3. progressIndicator?.showProgress()
4. errorIndicator?.hide()
5. val tokens = app.*dataStorageProvider*.*userStorage*.readTokens()
6. if (tokens != null) {
7. if (tokens.*accessToken*.isValid()) {
8. return sendAuthedRequest(tokens)
9. } else {

Продолжение листинга 3.1

1. LOGGER.debug("Access token is not valid. Trying to refresh tokens...")
2. return tryUpdateRefreshToken(tokens.refreshToken)
3. }
4. } else {
5. return tokensAuthFailed()
6. }
7. }
8. private suspend fun tryUpdateRefreshToken(refreshToken: Token): MessageResult<D> {
9. if (refreshToken.isValid()) {
10. return updateAccessToken(refreshToken)
11. } else {
12. LOGGER.debug("Refresh token is not valid.")
13. return tokensAuthFailed()
14. }
15. }
16. private suspend fun onTokenUpdated(tokenPair: TokenPair): MessageResult<D> {
17. LOGGER.debug("The tokens have been updated!")
18. app.dataStorageProvider.userStorage.updateTokens(tokenPair)
19. return sendAuthedRequest(tokenPair)
20. }

## Динамическая загрузка элементов

Главный экран авторизованной части приложения является бесконечным списком элементов. Для реализации загрузки списка продуктов был использован паттерн «Pagination», позволяющий разбить бесконечный поток данных на группы ограниченной длинны.

Для работы паттерна «Pagination» был разработан котроллер списка, определяющий следующие события:

* отсутствие элементов в списке;
* достижение конца доступных элементов в списке;
* отсутствие новых данных для списка.

Код реализации абстрактного котроллера приведен в листинге 3.2.

Листинг 3.2 – Листинг PaginationController

1. @Suppress("MoveVariableDeclarationIntoWhen", "LiftReturnOrAssignment")
2. abstract class PaginationController(
3. private val layoutManager: LinearLayoutManager,

Продолжение листинга 3.2

1. private val pageSize: Int
2. ) : RecyclerView.OnScrollListener() {
3. abstract val isLastPage: Boolean
4. abstract val isLoading: Boolean
5. override fun onScrolled(recyclerView: RecyclerView, dx:Int, dy: Int) {
6. super.onScrolled(recyclerView, dx, dy)
7. val firstVisibleItemPosition = layoutManager.findFirstVisibleItemPosition()
8. val visibleItemCount = layoutManager.childCount
9. val totalItemCount = layoutManager.itemCount
10. if (!isLoading && !isLastPage) {
11. if (firstVisibleItemPosition + visibleItemCount >= totalItemCount
12. && firstVisibleItemPosition >= 0
13. && totalItemCount >= pageSize) {
14. loadMoreItems(totalItemCount, pageSize)
15. }
16. }
17. }
18. protected abstract fun loadMoreItems(offset: Int, count: Int)
19. }

Реализация pagination-котроллера позволила отображать пользователю продукты с иллюзией бесконечного списка. На рисунках 3.8 и 3.9 изображен пример отображения списка в состоянии покоя и в состоянии загрузки соответственно.

|  |  |
| --- | --- |
|  |  |
| **Рисунок 3.8** – Список продуктов | **Рисунок 3.9** – Список продуктов: загрузка новых данных |

## Тестирование и отладка

Тестирование программного обеспечения является неотъемлемой частью создания программного продукта. От того, насколько досконально проведены тесты, зависит то, как скоро проект будет сдан окончательно, и будет ли необходимость впоследствии устранять ошибки. Тестирование программного продукта на разных стадиях создания – залог качественного выполнения работы.

Опираясь на данные принципы тестирование приложения проводилось как в ходе разработки всего проекта, после завершения реализации ее составных частей, так и по завершении реализации всей системы в целом. Это способствовало не только выявлению ошибок, опечаток и неточностей, но и позволяло улучшить пользовательский опыт, дорабатывая и исправляя некоторые функции.

За счет возможности замены сервера заглушкой, тестирование производилось сначала с использование локальных данных, и только потом с реальными данными сервера. Это позволило проверить большое количество исключительных ситуаций в разных частях приложения. А также облегчило поиск и исправление возникающих ошибок.

При выполнение большей части тестов, работоспособность основных компонентов проверялось с помощью тестирования методом «черного» ящика.

Для проверки таким способом использовались два метода:

1 причина/следствие – метод, когда процесс тестирования сопровождается вводом комбинаций условий для получения ответа от тестируемой системы;

2 предугадывание ошибки – метод, когда знание тестируемой системы используется для выявления ошибок, которые могут быть скрыты в ней.

В таблице 3.2 приведен перечень выполненных проверок с результатами тестирования.

Таблица 3.2 – Тестирование разработанного приложения

|  |  |  |
| --- | --- | --- |
| **Действие** | **Ситуация** | **Реакция программы** |
| Авторизация пользователя | Логин и пароль корректны | Запускается основной экрана; отображается приветственное сообщение |
| Логин и пароль некорректны | Отображается сообщение об ошибки входа: некорректные данные пользователя |

Продолжение таблицы 3.2

|  |  |  |
| --- | --- | --- |
| **Действие** | **Ситуация** | **Реакция программы** |
| Авторизация пользователя | Отсутствует соединение с интернетом | Отображается сообщение об ошибки входа: отсутствует соединение с интернетом |
| Запуск экрана списка продуктов | Имеется соединение с интернетом | Отображается индикатор загрузки, через несколько секунд отображаются загруженные продукты |
| Отсутствует соединение с интернетом | Отображается сообщение о ошибку подключения; отображается кнопка повтора |
| Клик по продукт в списке | Имеется соединение с интернетом | Запускается экран деталей продукта, загружаются дополнительные детали с отображением индикатора загрузки |
| Отсутствует соединение с интернетом | Отображается сообщение о ошибке подключения; отображается кнопка повтора |
| Нажатие системной кнопки «назад» в авторизованной части | Запущен список продуктов | Приложение сворачивается; отображается предыдущее открытое приложение или главный экран устройства |
| Нажатие системной кнопки «назад» в авторизованной части | Запущен экран деталей продукта | Экран деталей закрылся, отображается список продуктов |
| Запущен экран деталей профиля | Экран с деталями профиля закрылся, отобразился предыдущий экран приложения |

Продолжение таблицы 3.2

|  |  |  |
| --- | --- | --- |
| **Действие** | **Ситуация** | **Реакция программы** |
| Нажатие системной кнопки «назад» в авторизованной части | Все поля заполнены корректно, поле название оставлено пустым. | Отображается ошибка добавления продукта с текстом подсказки: введите название продукта |
| Добавления нового продукта | Все поля заполнены корректно, поле цена оставлено пустым. | Отображается ошибка добавления продукта с текстом подсказки: укажите стоимость продукта |
| Все поля заполнены корректно | Отображается индикатор загрузки; через несколько секунд экран добавления продукта закрывается; отображается сообщение о отправки продукта на проверку; запускается экран со списком продуктов. |
| Все поля заполнены корректно; отсутствует соединение с интернетом | Добавление прервано; отображается сообщение о ошибке подключения к интернету; отображается кнопка для выполнения операции повторно |
| Главный экран, открытие главного меню | Авторизован обычный пользователь | Главное меню открывается,  Отображается имя авторизованного пользователя, отображаются пункты: мой профиль, лента, добавить продукт, выйти. |

Окончание таблицы 3.2

|  |  |  |
| --- | --- | --- |
| **Действие** | **Ситуация** | **Реакция программы** |
| Главный экран, открытие главного меню | Авторизован пользователь-модератор | Главное меню открывается,  Отображается имя авторизованного пользователя, отображаются пункты: мой профиль, лента, добавить продукт, добавленные продукты, выйти. |
| Запущен экран деталей продукта, пользователь производит длинный клик по комментарию | Авторизованный пользователь – обычный пользователь | Никаких действий не происходит |
| Авторизованный пользователь – модератор | Открывается диалог подтверждения удаления комментария. При подтверждении, комментарий удаляется |

Тестирование проводилось в соответствии с документом «Программа и методика испытаний», представленном в приложении В.

Результаты тестирования показали, что разработанное приложение удовлетворяет требованиям технического задания.

## Выводы по третьей главе

В результате выполненных работы в соответствии с функциональными требованиями, описанными в документе «Техническое задание», реализовано Android приложение. Программный продукт протестирован в соответствии с документом «Программа и методика испытаний».

# ЭКОНОМИЧЕСКАЯ ЧАСТЬ

## Обоснование необходимости разработки программного обеспечения

Цель экономического раздела дипломного проекта – рассчитать затраты на разработку программного обеспечения и определить экономическую эффективность от его внедрения.

В рамках дипломного проекта разрабатывается сервис по поиску алкогольных напитков.

Сервис будут использовать люди, которые любят путешествовать и не любят рисковать в плане выбора напитков. Продукт будет широко применим среди различных категорий людей из-за того, что там будут предоставлены разные напитки с разным ценовым диапазоном.

В отличии от других приложений данное приложение нацелено только на напитки, что делает его более узкоспециальным, а значит им будут пользоваться только те, люди, кто в этом действительно заинтересован.

Создание программного продукта, как и любого другого продукта, должно быть обосновано также и с экономической точки зрения. Ввод в эксплуатацию программного продукта требует достаточно крупных денежных и временных затрат, поэтому при его выборе необходимо оценить, через какое время вернутся вложенные деньги и прочие ресурсы и что возможно будет получить в результате разработки. Проектирование заведомо не пользующимся спросом программного продукта не может являться целесообразным капиталовложением.

Программные средства вычислительной техники являются сложным продуктом, имеющим определённые особенности. Задачи, возложенные на программные средства, как правило, можно решить традиционными методами, не прибегая к использованию вычислительной техники. Но затраты ресурсов в этом случае многократно возрастают и сводят на нет выгоду от решения задачи.

Разработка программных средств требует определенных интеллектуальных и трудовых затрат, а также обязательного использования компьютерной техники, что определяет особенности расчета себестоимости программного продукта.

Проведённый в разделе 1 сравнительный анализ аналогов и прототипов показал, что они имеют ряд недостатков. Это позволит разрабатываемому программному продукту найти своих потребителей.

## Этапы разработки программного обеспечения

Для разработки программного обеспечения необходимо время и структура этапов разработки, которое выражается в количестве недель, затраченных на его разработку. Стадия разработки программного средства может быть разделена на следующие этапы:

* проектирование;
* реализация;
* тестирование;
* доработка.

Каждый этап включает набор действий, процессов, задач, по завершению которых происходит переход к следующему пункту цикла.

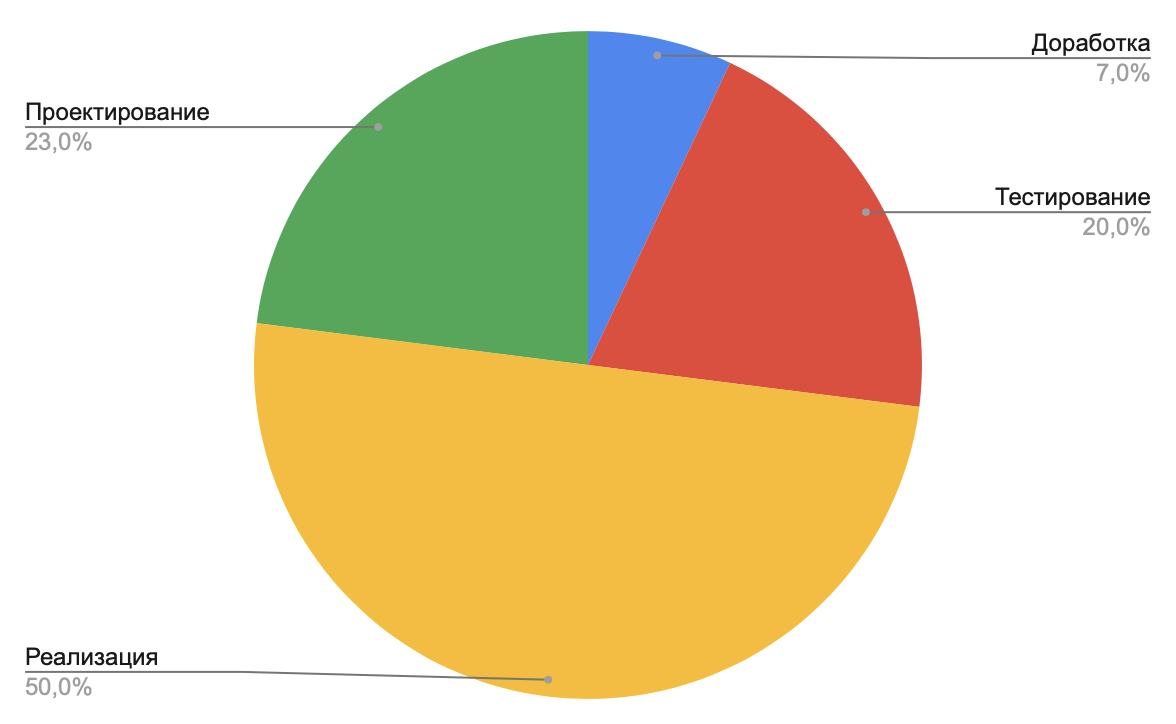
К этапу «Проектирование» относится определение, спецификация, анализ и модификация аппаратно-программной архитектуры проекта программы и базы данных. На этом этапе создается основной план поведения программы, главный набор ее действий. Ошибки в проектировании структуры программы очень трудно исправить в дальнейшем, т.к. их исправление может привести к полной переделке ранее написанных модулей. На данный этап отводится довольно длительное время, так как качественно выполненное проектирование позволяет сократить временные затраты на последующие этапы разработки.

На этапе «реализация» происходит непосредственно реализация спроектированного продукта, т.е. воплощение в программном коде модулей, разработанных на предыдущем этапе. Данный этап является самым продолжительным и составляет от 40 до 60% от всего времени разработки. На этапе реализации поведение программы выстраивается на основе выбранных при проектировании технологий.

На этапе «Тестирование и испытание системы» проводится проверка соответствия результата этапа реализации с результатами, полученными на этапе проектирования. По продолжительности данный этап занимает около 25% времени разработки.

На этапе «Доработка» проводится заключительный приёмо-сдаточные испытания, при необходимости – внесение незначительных исправлений в программный продут и оформление заключительных глав пояснительной записки.

Диаграмма, отображающая стадии разработки программного обеспечения, приведена на рисунке 4.1.



**Рисунок 4.1** – Этапы разработки программного обеспечения

В таблице 4.1 представлена структура этапов и время в неделях и процентном соотношении для каждого этапа, необходимого для разработки продукта.

**Таблица 4.1** – Время этапов разработки ПО

|  |  |  |
| --- | --- | --- |
| **Этап разработки** | **Затраты времени, недели** | **Процент от общего времени, %** |
| Проектирование | 2,76 | 23 |
| Реализация | 6 | 50 |
| Тестирование | 2,4 | 20 |
| Доработки | 0,84 | 7 |
| Итого | 12 | 100 |

## Составление сметы затрат на разработку программного обеспечения

Стоимостная оценка ПО предполагает составление сметы затрат, которая в денежном выражении включает следующие статьи расходов:

* материалы и комплектующие *(М)*;
* электроэнергия (Э);
* основная заработная плата разработчиков (Зо);
* дополнительная заработная плата разработчиков (Зд);
* отчисления на социальные нужды (Зсз);
* амортизация основных средств и нематериальных активов (А);
* расходы на спецоборудование (Рс);
* накладные расходы (Рн);
* прочие прямые расходы (Пз).

*Материалы и комплектующие*. Отражают расходы на магнитные носители, бумагу, красящие ленты и другие материалы, необходимые для разработки ПО. (Заполняйте своими данными – итоговая сумма в таблицу 4.2.)

**Таблица 4.2** – Стоимость материалов

|  |  |  |  |
| --- | --- | --- | --- |
| **Наименование материала** | **Количество** | **Цена за единицу, *руб*** | **Сумма, *руб*** |
| Бумага формата а4 | 100 листов | 1,25 | 12,50 |
| Компакт-диск CD-RW | 1 шт. | 1,0 | 1,0 |
| Всего | | | 13,50 |

Общая стоимость основных и вспомогательных материалов М = 13,5 рублей.

*Электроэнергия*. Стоимость электроэнергии необходимой для разработки программного обеспечения представлена в таблице 4.3. Для расчета данного показателя необходим иметь в наличии следующие данные:

* продолжительность разработки;
* количество часов потребления электроэнергии за весь период разработки;
* стоимость одного кВт/ч электроэнергии;
* значение курса белорусского рубля по отношению к доллару США, установленного Национальным банком РБ, на дату установления тарифа на электрическую энергию.

Период разработки программы определяется в соответствии с длительностью дипломного проектирования и составляет 3 месяца. При 8-часовом рабочем дне и 5-дневной неделе получаем в неделю 40 часов, а за 12 недель разработки – 480 часов.

, (4.1)

где *Hw* – количество рабочего времени, затраченного на разработку,

*H* – продолжительность рабочего дня (часы),

*Wd* – количество рабочих дней,

*Ws* – сумма недель разработки.

Для прочих потребителей базовый тариф составляет 0,33048 BYN за 1 кВт/ч (согласно Приказа Министерства антимонопольного регулирования и торговли Республики Беларусь от 31.01.2020г). Такая стоимость установлена исходя из соотношения курса белорусского рубля к доллару США 2,1085:1.

При изменении курса доллара США тариф индексируется по формуле:

, (4.2)

где *Тн* – тариф на электрическую энергию, подлежащий применению на день оформления платежных документов и день оплаты потребителем за потребленную электрическую энергию,

*Тб* – тариф на электрическую энергию, установленный декларацией,

*Кн* – значение курса белорусского рубля по отношению к доллару США, установленного Национальным банком РБ, на день оформления платежных документов и день оплаты за потребленную электрическую энергию (на 05.05.2020 курс белорусского рубля к доллару США 2,4404:1),

*Кб* – значение курса белорусского рубля по отношению к доллару США, установленного Национальным банком РБ, на дату установления тарифа на электрическую энергию.

На 05.05.2020 курс белорусского рубля к доллару США 2,4404:1 установлен Национальным банком РБ. На основе данного курса рассчитаем тариф на электрическую энергию.

Более экономично, при потреблении электроэнергии только ПЭВМ, рассчитать затраты электроэнергии можно по следующей формуле.

, (4.3)

где *Ew* – затраты на электроэнергию за период разработки (руб.);

*Ppc* – потребление электроэнергии ПЭВМ согласно технической документации (0,25 КВт/ч); h – продолжительность рабочего дня (часы);

*Wd* – количество рабочих дней;

*Ws* – сумма недель разработки;

*T* – тариф кВт/ч.

*Основная заработная плата разработчиков.* Основной статьей расходов на создание ПО является заработная плата разработчиков (исполнителей) проекта, в число которых входит разработчик.

Расчет основной заработной платы исполнителей. Общая трудоемкость, плановая численность работников и плановые сроки разработки ПО являются базой для расчета основной заработной платы разработчиков проекта. Оплата труда осуществляется на основе Единой тарифной сетки Республики Беларусь (ETC), в которой даны тарифные разряды и тарифные коэффициенты.

Расчет по статье «Основная заработная плата» (ЗО) осуществляется по формуле:

, (4.4)

где – средняя заработная плата работников РБ в области информационных технологий и деятельности в области информационного обслуживания за месяц по данным Национального статистического комитета РБ,

*n* – количество исполнителей;

– количество месяцев, затраченных на разработку ПО.

*Дополнительная заработная плата.* Включает оплату отпусков, льготных часов, времени выполнения государственных обязанностей и других выплат, не связанных с основной деятельностью исполнителей, и определяется по нормативу в процентах к основной заработной плате. Норматив дополнительной заработной платы принимается в размере 7-15%.

, (4.5)

где З*дi* – дополнительная заработная плата;

З*Оi* – основная заработная плата;

*Нд* – норматив дополнительной заработной платы на предприятии, выраженный в процентах (принимается равным 15%).

*Отчисления на социальные нужды.* Включает в себя отчисления в фонд социальной защиты населения (Нсз – 34%) и отчисления на обязательное страхование от несчастных случаев (Нстр – 0,1 - 0,6%).

Отчисления на социальные нужды определяются в соответствии с действующими законодательными актами по нормативу в процентном отношении к фонду основной и дополнительной зарплаты исполнителей, определенной по нормативу, установленному в целом по организации:

, (4.6)

где *Нсз* – норматив отчислений в фонд социальной защиты населения (%);

*Нстр* – норматив отчислений на обязательное страхование от несчастных случаев (%).

(руб.)

*Амортизация основных средств и нематериальных активов.* Рассчитываются амортизационные отчисления, исходя из стоимости основных средств и нематериальных активов, используемых в процессе разработки программного обеспечения, сроков эксплуатации оборудования и годовой нормы амортизации. Балансовая стоимость персонального компьютера определяется исходя из его комплектации по рыночным ценам, действующим на время создания программного продукта, включая расходы на его доставку и сборку. Расчет стоимости персонального компьютера представлен в таблице 4.3.

**Таблица 4.3** – Балансовая стоимость персонального компьютера

|  |  |  |  |
| --- | --- | --- | --- |
| **Наименование** | **Количество, шт.** | **Цена, руб./шт.** | **Сумма, руб.** |
| Ноутбук Dell G3 15 | 1 | 2285,21 | 2285,21 |
| Мышь Logitech M280 | 1 | 49.90 | 49.90 |
| Итого | | | 2335,11 |

Стоимость прикладного программного обеспечения определяется исходя из его состава и уровня рыночных цен, действующих на время создания программного продукта в таблице 4.4.

**Таблица 4.4** – Стоимость прикладного программного обеспечения

|  |  |  |  |
| --- | --- | --- | --- |
| **Наименование** | **Количество, шт.** | **Цена, руб./шт.** | **Сумма, руб.** |
| Windows 10 | 1 | 238,03 | 238,03 |
| Microsoft office 2019 | 1 | 235.65 | 235.65 |
| IntelliJ IDEA Ultimate | 1 | 354.92 | 354.92 |
| Java 14 | 1 | 0 | 0 |
| Итого | | | 828.60 |

Определим установленный размер амортизационных отчислений на полное восстановление, выраженное в процентах по формуле (4.7):

, (4.7)

где *На* – годовая норма амортизации,

*Тн* – нормативный срок службы.

Месячная норма амортизации рассчитывается по формуле.

, (4.8)

где *Нам* – месячная норма амортизации, руб,

*На* – годовая норма амортизации, руб.

Тогда норма амортизационных отчислений за месяц составит:

.

Сумма амортизации по ноутбуку (Aос) рассчитывается по формуле:

, (4.9)

где *Аос* – амортизация прикладного программного обеспечения,

*БСос*– балансовая стоимость программного обеспечения,

*НАм* – норма амортизации за месяц, М – количество месяцев.

Сумма амортизаций по прикладному программному обеспечению рассчитывается по формуле.

, (4.10)

где *БСппо* – балансовая стоимость программного обеспечения,

*Нам* – норма амортизации за месяц,

*М* – количество месяцев,

*Аппо* – амортизация прикладного программного обеспечения.

Общая сумма амортизационных отчислений равна сумме амортизационных отчислений персонального компьютера и амортизационных отчислений прикладного программного обеспечения и составляет:

*Расходы на спецоборудование.* Включает затраты средств на приобретение вспомогательных специального назначения технических и программных средств, необходимых для разработки конкретного ПО, включая расходы на их проектирование, изготовление, отладку, установку и эксплуатацию. При разработке рассматриваемого продукта технические и программные средства специального назначения не использовались, поэтому затраты по статье «Расходы на спецоборудование» отсутствуют.

*Прочие прямые расходы.* К прочим прямым расходам относятся затраты на доступ в сеть Интернет. Тарифный план высокоскоростного нелимитируемого доступа в сеть Интернет («Рекорд 200» (скорость прием/передача до 200/100 Мбит/с)), в месяц 43 руб.

43 руб. \* 3 месяца = 129 руб*.*

На основании полученных данных по отдельным статьям затрат рассчитана плановая себестоимость программного обеспечения (таблица 4.5).

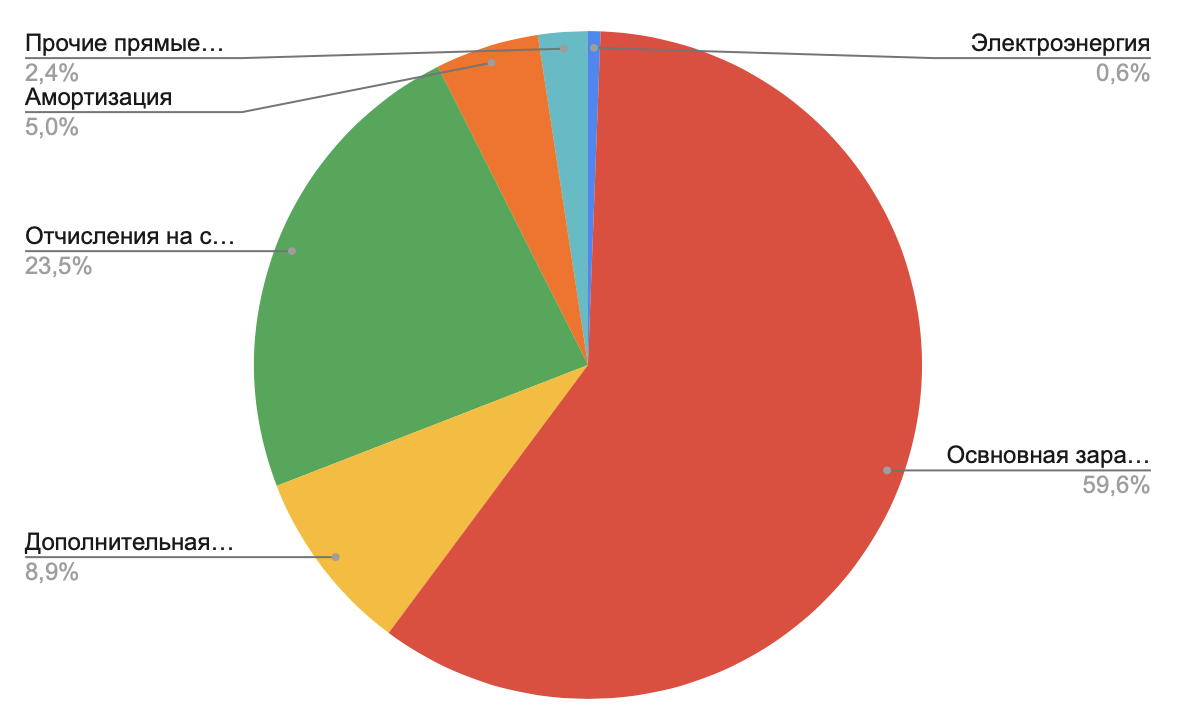
**Таблица 4.5** – Расчет плановой себестоимости программного обеспечения

|  |  |
| --- | --- |
| **Статья затрат** | **Сумма затрат, руб.** |
| Материалы и комплектующие | 13,50 |
| Электроэнергия | 33,3 |
| Основная заработная плата | 3150 |

Продолжение таблицы 4.5

|  |  |
| --- | --- |
| Дополнительная заработная плата | 472,50 |
| Отчисления на социальные нужды | 1238,90 |
| Амортизация | 263,38 |
| Прочие прямые расходы | 129 |
| Общая сумма расходов по смете (плановая себестоимость): | 5300,58 |

Таблицу 4.5 удобно представить в виде диаграммы, отображающей влияние статей затрат на разработку программного обеспечения на окончательную себестоимость программы. Данная диаграмма представлена на рисунке 4.2.



**Рисунок 4.2** – Структура затрат на разработку программы

После вычислений, проведенных в данном пункте, можно сделать вывод, что плановая себестоимость разрабатываемого программного средства равна 5300,58 белорусских рублей.

## Экономический эффект у разработчика ПО

Заказчик оплачивает разработчику всю сумму расходов по проекту, включая прибыль. После уплаты налогов из прибыли в распоряжении разработчика остается чистая прибыль от проекта. Ввиду того, что программное обеспечение разрабатывается для одного объекта, чистую прибыль можно считать в качестве экономического эффекта разработчика от реализованного программного обеспечения.

В дипломном проекте отпускная цена программного обеспечения, представляет собой не цену за единицу продукции, а цену проекта вместе с его исходными кодами и документацией, за которую его можно продать и получить определенную выгоду.

Отпускная цена продукции формируется из плановой себестоимости, всех видов установленных налогов и прибыли, качества, потребительских свойств и конъюнктуры рынка. Отпускная цена продукции рассчитывается по формуле:

, (4.11)

где *ОЦ* – отпускная цена, руб;

*С* – плановая себестоимость, руб;

*П* – прибыль, руб.

Прибыль закладывается в цену по нормативу рентабельности и расчет

производится по следующей формуле:

, (4.12)

где *R* – норматив рентабельности (например, если рентабельность 20 %, то при расчете переводим в коэффициент и получаем 0,2).

*С* – плановая себестоимость, руб;

Отпускная цена с учетом налога на добавленную стоимость представляет собой сумму отпускной цены и НДС. НДС рассчитывается по формуле:

, (4.13)

где *НДС* – величина налога на добавленную стоимость;

*Ставка НДС%* – ставка налога на добавленную стоимость, выраженная в процентах (установлена в размере 20%).

Таким образом, цена программного обеспечения с учетом НДС

рассчитывается как сумма отпускной цены и величины налога на добавленную стоимость. Для разработанной информационной системы цена с учетом НДС составляет:

Величина ОЦ с НДС, равная 7632,84 белорусских рубля, покрывает затраты и обеспечивает прибыль за разработку проекта.

## Экономический эффект от использования программного обеспечения у пользователя (заказчика)

Для проведения сравнительного анализа программного продукта с существующими аналогами необходимо выделить критерии сравнения. Разработанный программный продукт представляет собой сервис, который предоставляет API для разработчиков клиентской части приложения и которое обладает следующими возможностями:

* добавление продукта вне зависимости от типа напитка;
* просмотр продукта;
* просмотр пользователя;
* авторизация пользователя в системе;
* удаление продукта;
* редактирование продукта;
* добавления в избранное;
* комментирования продукта;
* поиск с применением фильтров для более удобного поиска;
* редактирование пользователя.

Ближайший аналог для данного сервиса является приложение Vivino. Приложение Vivino предоставляет те же функции за исключением только того, что оно базируется только на винной продукции, а также не дает информации о том, где в ближайшем месте можно купить тот или иной продукт.

Как видно из сравнительного анализа, разрабатываемый программный продукт имеет не только идентичные функции, но уникальный функционал, который будет удобен более широкому пользователю. Из вышеизложенного можно сделать вывод, что разрабатываемое приложение будет реализовано по всем современным тенденциям веб-разработки.

## Вывод по четвёртой главе

При разработке мобильного приложения «GoodBooze» для платформы Android была рассчитана себестоимость программного продукта и его отпускная цена с учетом НДС. Для расчета себестоимости реализованного приложения были подсчитаны следующие значения:

* материалы и комплектующие;
* электроэнергия;
* основная заработная плата;
* дополнительная заработная плата;
* отчисления на социальные нужды;
* амортизация;
* прочие прямы расходы.

# ЗАКЛЮЧЕНИЕ

Результатом дипломного проектирования является разработанное мобильное приложение для платформ Android «GoodBooze» для поиска, оценки и продвижения напитков. Несмотря на полную готовность выпуска продукта на публичный рынок, в нём всё еще присутствуем множество перспективных направлений для дальнейшее доработки и усовершенствования.

В процессе реализации дипломного проекта были достигнуты следующие результаты:

1. проведен анализ предметной области, выявлены основные функции, которые должен решать программный продукт;
2. проведено изучение рынка, в ходе которого было определено направление и цели проектирования; проведен анализ аналогов разработанного продукта;
3. выбраны инструментальные средства проектирования и разработки. Для реализации приложения использовались языки программирования Java и Kotlin; изучены возможности сторонних библиотек; улучшен навык владения интегрированной средой разработки Android Studio;
4. на основании поставленных задач технического задания спроектирована и реализована функциональная часть приложения;
5. создан пользовательский интерфейс приложения, отвечающий современным требования и стандартам платформы Android;
6. разработана проектная документация в составе:
7. техническое задания (см. приложение А),
8. описание программы (см. приложение Б),
9. программа и методика испытаний (см. приложение В);
10. руководство пользователя (см. приложение Г);
11. руководство модератора (см. приложение Д).
12. рассчитана сметная стоимость на разработку и содержание продукта.

Разработанный программный продукт позволяет решить проблему питания в незнакомом для человека месте, узнать больше о местной культуре или разнообразить рацион пользователя. Сервис может оказаться полезным и для заведений, помогая им продвигать свой продукт и привлекать новых клиентов. Результат работы имеет перспективные варианты коммерческого развития и привлечения инвестиций; может найти широкое применение как в Республике Беларусь, так и за её пределами.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Википедия – свободная энциклопедия [Электронный ресурс]. – Режим доступа: http://ru.wikipedia.org/. – Дата доступа: 19.05.2020.
2. Официальный сайт для разработчиков приложений для Android [Электронный ресурс]. – Режим доступа: https://developer.android.com/. – Дата доступа: 26.05.2020.
3. Документация языка программирования Kotlin [Электронный ресурс]. – Режим доступа: https://kotlinlang.org. – Дата доступа: 26.05.2020.
4. Документация по языку Java от компании Sun [Электронный ресурс]. – Электрон. текстовые дан. (270 Мб). – Sun Microsystems, Inc., 2006
5. Арлоу Д., Саливон И.И. Java для профессионалов: Учеб. пособие. – Мн: Технология, 2007.
6. Стив МакКоннелл, 2019. Совершенный код: учебное пособие / Русская редакция, 2019. – 342с.
7. Дженифер Тидвелл. Разработка пользовательских интерфейсов, 2011. – 480 с.
8. Диаграмма вариантов использования [Электронный ресурс]. – Режим доступа: http://www.znannya.org/?view=uml\_use\_case\_diagram\_4. – Дата доступа: 16.06.2020.
9. Арлоу Д., Нейшталд А. UML и Унифицированный процесс: практи- ческий объектно-ориентированный анализ и проектирование, 2-е издание. – Пер. с англ. – СПб: Символ-Плюс, 2007. – 624 с.
10. Вендеров, А.М. CASE-технологии. Современные методы и средства проектирования информационных систем / А.М. Вендеров. – М.: Финансы и статистика, 1998 – 123с.
11. Всемирная туристская организация [Электронный ресурс]. – Режим доступа: https://www.unwto.org/statistics. – Дата доступа: 28.03.2020.
12. Отчёт App Annie 2018-2019 [Электронный ресурс]. – Режим доступа: https://qmobi.agency/blog/app-annie-mobile-2018-2019/. – Дата доступа: 26.03.2020.
13. Прогноз рынка мобильных приложений App Annie [Электронный ресурс]. – Режим доступа: https://qmobi.agency/blog/app-annie-predictions/ – Дата доступа: 26.03.2020.

# ПРИЛОЖЕНИЕ А (обязательное) ТЕХНИЧЕСКОЕ ЗАДАНИЕ

## Введение

Наименование продукта: мобильное приложение «GoodBooze» по поиску напитков для платформы Android (клиентская часть).

Продукт будет использоваться пользователями для публикации, поиска, оценки и комментирования напитков.

## Основание для разработки

Данный программный продукт разрабатывается в рамках дипломного проекта студентом Полоцкого государственного университета факультета информационных технологий специальности 1-40 01 01 «Программное обеспечение информационных технологий» группы 16-ИТ-3 Яблонский Андрей Сергеевич.

Основанием для разработки проекта является дипломный проект на тему «мобильное приложение «GoodBooze» по поиску напитков для платформы Android (клиентская часть) ».

## Назначение разработки

Основной целью разработки приложения является Android приложение, для быстрого просмотра и выбора напитков в незнакомом для пользователя месте, основываясь на рейтинге и комментариях других пользователей

## Требование к программе или программному изделию

## Требования к функциональным характеристикам

В реализованном приложении функциональными характеристиками для авторизованного пользователя выдвигаются следующие требования:

1. просмотр списка добавленных продуктов;
2. добавление нового продукта;
3. просмотр информации о продукте;
4. оценка продукта;
5. просмотр комментариев других пользователей о каждом продукте;
6. добавление комментария;
7. редактирование профиля;
8. удалением собственных продуктов.

Продукты добавляются обычными пользователями и отправляются на проверку модератором, который может разрешить или запретить публикацию.

## Требования к надежности

Приложение должно стабильно работать на устройствах Android. В случае возникновения внутренних ошибок связи с сервером; некорректном вводе пользовательских данных, выдавать соответствующие сообщение с описанием ошибки.

## Требования к составу и параметрам технических средств

Минимальными достаточными требованиями по конфигурации устройства пользователя для функционирования приложения является:

* процессор Snapdragon 630 или сопоставимые по производительности аналог;
* 2 Гб оперативной памяти;
* диагональ экрана от 4-8 дюйма;
* стабильное подключение к сети Интернет (со скоростью соединения не менее 1 Мбит/с);
* 100 Мб постоянного хранилища.

## Требования к информационной и программной совместимости

Для совместимости на программном уровне устройство пользователя должно работаться на операционной системе Android версии 5.0 и выше с установленным пакетом Google Mobile Services.

## Требования к маркировке и упаковке

Требования к маркировке и упаковке отсутствуют.

## Требования к транспортированию и хранению

Требования к Требования к транспортированию и хранению отсутствуют.

## Требования к программной документации

Требования к программным документам устанавливаются государственными стандартами ЕСПД.

Программная документация проекта должна быть представлена следующим комплектом:

* техническое задание;
* описание программы
* программа и методика испытаний.

Программная документация должна быть оформлена в соответствии со следующими нормативными документами:

* ГОСТ 19.201-78 ЕСПД. Техническое задание – Издательство стандартов, 1982 год;
* ГОСТ 19.402-2000 ЕСПД. Описание программы – Издательство стандартов, 2001 год;
* ГОСТ 19.301 – 2000 ЕСПД. Программа и методика стандартов. – Издательство стандартов, 2001 год.

## Технико-экономические показатели

Быстродействие и эффективность данного программного продукта обуславливается оптимизацией под платформу и отсутствием избыточного функционала.

## Этапы разработки

Этапы разработки программного продукта:

* анализ аналогов;
* постановка задач;
* выбор инструментальных средств;
* анализ исходных данных;
* проектирование
* разработка;
* тестирование;
* разработка технической документации

## Порядок контроля и приемки

Для контроля выполнения требования технического задания необходимо провести испытания. Порядок и состав испытания определяются программой и методикой испытаний.

Приемка готового программного средства осуществляется комиссией по защите дипломного проекта в соответствии с программой и методикой испытаний.

# ПРИЛОЖЕНИЕ Б (обязательное) ОПИСАНИЕ ПРОГРАММЫ

## Общие сведенья

Наименование программного средства – мобильное приложение «GoodBooze» по поиску напитков для платформы Android (клиентская часть).

Продукт предоставляет пользователям, находящимся в незнакомом для себя месте найти подходящий для них продукт или напиток основываясь на отзывах и комментариях других пользователей.

## Функциональное назначение

Мобильное приложение предназначено для решения следующих задач:

* помочь с выбором хорошего продукта среди локального ассортимента;
* рассказать пользователям о плюсах и минусах каждого продукта;
* предоставить информации о ближайшей точке продажи искомого продукта;
* популяризировать продукты местной кухни.

## Описание логической структуры

При первом запуске приложения пользователь попадает на экран авторизации.

После успешной авторизации запускается главный экран – лента продуктов. Каждый элемент ленты представлен в виде изображения продукта, названия, стоимости и ближайшего адреса продажи.

При клике по продукте отображается экран с деталями продукта. Также появляется возможность оценить продукт, оставить отзыв или прочитать уже оставленные комментарии.

При клике по комментарию отображается детали профиля автора.

Из главного меню приложения авторизованному пользователю доступна настройка собственного профиля: имя, email, биография и др.

Нажатие на кнопку добывания продукта в боковом меню запустит экран нового продукта. После корректного заполнения всех полей, продукт отправляется на проверку модераторам.

Кнопка выходы в главном меню вернёт пользователя на экран авторизации.

При входе как модератор, в боковом меню отображается дополнительный функционал – «продукты на проверку». Данный пункт запускает экран со списком отправленных на проверку продуктов. Модератор может разрешить или запретить публикацию длинным удержанием по элементу в списке продуктов.

При просмотре комментариев продукта, длинным нажатием модератор может удалять комментарии нарушающие правила публикации.

## Используемые технические средства

Для корректной работы приложение устройство должно удовлетворять следующим минимальным программно-техническими требованиям:

* процессор Snapdragon 630 или сопоставимые по производительности аналог;
* 2 Гб оперативной памяти;
* диагональ экрана от 4-8 дюйма;
* стабильное подключение к сети Интернет (со скоростью соединения не менее 1 Мбит/с);
* 100 Мб постоянного хранилища;
* операционная система Android версии 5.0 или выше.

## Вызов и загрузка

Запуск программного продукта возможен из лаунчера системы Android или любыми другими способами, предоставляемыми операционной системой.

## Входные данные

Входными данными для приложения «GoodBooze» является:

* логин и пароль при авторизации пользователя;
* комментарий о продукте;
* данный авторизованного пользователя;
* информация при добавлении нового продукта.

## Выходные данные

Выходные данные в приложении представлены в виде текстовой и графической информации. Такими данными является:

* продуктов с деталями: картинкой, названием, стоимостью, ближайшим местом продажи, описанием;
* комментарий к продукту: имя автора, текст комментария;
* детали пользователя: имя, email, дата рождения, количество публикаций, описание профиля.

# ПРИЛОЖЕНИЕ В (обязательное) МЕТОДИКА ИСПЫТАНИЙ

## Объект испытаний

Объектом испытания является мобильное приложение «GoodBooze» по поиску напитков для платформы Android (клиентская часть).

## Цель испытания

Тестирование программного продукта «Мобильное приложение «GoodBooze» по поиску напитков для платформы Android (клиентская часть)», проверка его работоспособности, соответствия техническому заданию, выявление, дефектов, несоответствий в работе программы и в программной документации.

## Требование к программе

Разработанный программный продукт должен предоставлять отличный функционал для пользователя и модератора.

Для пользователя должен быть доступен следующий функционал:

* просмотр списка добавленных продуктов;
* добавление нового продукта;
* просмотр информации о продукте;
* оценка продукта;
* просмотр комментариев других пользователей о каждом продукте;
* добавление комментария;
* редактирование профиля;
* удалением собственных продуктов.

В добавок к возможностям пользователей, модератор имеет дополнительный функционал:

* просмотр списка заявок на добывание продукта;
* одобрение и отклонение заявки добавления продукта;
* удаление комментариев пользователей.

Список продуктов, комментариев и профилей пользователей должен храниться независимо от устройства, на удаленном сервере.

## Требования к программной документации

Проект должен содержать следующие программные документы:

* описание программы;
* руководство пользователя;
* руководство модератора.

## Средства и порядок испытаний

## Требования к функциональным характеристикам

Для проведения испытаний над мобильным приложением наличие Android устройства с возможностью выхода в сеть Интернет. Для проверки корректности функционирования разработанного программного обеспечения применялось Android устройство со следующими заявленными характеристиками:

* центральный процессор Snapdragon 835;
* диагональ экрана 6 дюймов;
* операционная система Android версии 9.0;
* оболочка операционной системы OxygenOS 9.0.11;
* оперативной памяти 8 Гб;
* свободного месте в постоянном хранилище 34 Гб;
* WiFi подключение к Интернету со скоростью 25 Мбит/с.

## Порядок испытаний

В таблице В.1 приведен перечень выполненных проверок с результатами тестирования.

Таблица В.1 – Результаты тестирования приложения

|  |  |  |
| --- | --- | --- |
| **Действие** | **Ситуация** | **Реакция программы** |
| Авторизация пользователя | Логин и пароль корректны | Запускается основной экрана; отображается приветственное сообщение; досткупна кнопка запуска главного меню. |

Продолжение таблицы В.1

|  |  |  |
| --- | --- | --- |
| **Действие** | **Ситуация** | **Реакция программы** |
| Авторизация пользователя | Логин и пароль некорректны | Отображается сообщение об ошибки входа: некорректные данные пользователя |
| Отсутствует соединение с интернетом | Отображается сообщение об ошибки входа: отсутствует соединение с интернетом |
| Запуск экрана списка продуктов | Имеется соединение с интернетом | Отображается индикатор загрузки, через несколько секунд отображаются загруженные продукты |
| Отсутствует соединение с интернетом | Отображается сообщение о ошибку подключения; отображается кнопка повтора |
| Клик по продукт в списке | Имеется соединение с интернетом | Запускается экран деталей продукта, загружаются дополнительные детали с отображением индикатора загрузки |
| Отсутствует соединение с интернетом | Отображается сообщение о ошибке подключения; отображается кнопка повтора |
| Нажатие системной кнопки «назад» в авторизованной части | Запущен список продуктов | Приложение сворачивается; отображается предыдущее открытое приложение или главный экран устройства |
| Запущен экран деталей продукта | Экран деталей закрылся, отображается список продуктов |
| Запущен экран деталей профиля | Экран с деталями профиля закрылся, отобразился предыдущий экран приложения |
| Добавления нового продукта | Все поля заполнены корректно, поле цена оставлено пустым. | Отображается ошибка с подсказкой указать стоимость продукта |
| Все поля заполнены корректно; отсутствует соедниение с интернетом | Отображается сообщение о ошибке подключения к интернету; отображается кнопка повтора |

Продолжение таблицы В.1

|  |  |  |
| --- | --- | --- |
| **Действие** | **Ситуация** | **Реакция программы** |
| Добавления нового продукта | Все поля заполнены корректно | Отображается индикатор загрузки; через несколько секунд экран добавления продукта закрывается; отображается сообщение о отправки продукта на проверку; запускается экран со списком продуктов. |
| Главный экран, открытие главного меню | Авторизован обычный пользователь | Главное меню открывается,  Отображается имя авторизованного пользователя, отображаются пункты: мой профиль, лента, добавить продукт, выйти. |
|  | Авторизован пользователь-модератор | Главное меню открывается,  Отображается имя авторизованного пользователя, отображаются пункты: мой профиль, лента, добавить продукт, добавленные продукты, выйти. |
| Запущен экран деталей продукта, пользователь производит длинный клик по комментарию | Авторизованный пользователь – модератор | Открывается диалог подтверждения удаления комментария. При подтверждении, комментарий удаляется |
| Авторизованный пользователь – обычный пользователь | Никаких действий не происходит |

Окончание таблицы В.1

|  |  |  |
| --- | --- | --- |
| **Действие** | **Ситуация** | **Реакция программы** |
| Запущен экран деталей продукта, пользователь производит длинный клик по комментарию | Авторизованный пользователь – модератор | Открывается диалог подтверждения удаления комментария. При подтверждении, комментарий удаляется |
| Запущен экране настройки профиля | В системе установлен русский язык | Подсказки в полях отображаются на русском языке |
| Запущен экране настройки профиля | В системе установлен английский язык | Подсказки в полях отображаются на английском языке |
| Запущен экран авторизации | В системе установлен русский язык | Текст подзаголовка, текст кнопки, подсказки в полях ввода отображаются на русском языке |
|  | В системе установлен английский язык | Текст подзаголовка, текст кнопки, подсказки в полях ввода отображаются на английском языке |

# ПРИЛОЖЕНИЕ Г (обязательное) РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

## Назначение программы

Наименование программного средства – Мобильное приложение «GoodBooze» по поиску напитков для платформы Android (клиентская часть).

Продукт предоставляет пользователям, находящимся в незнакомом для себя месте найти подходящий для них продукт или напиток основываясь на отзывах и комментариях других пользователей.

## Условия выполнения программы

Для корректной работы приложение устройство должно удовлетворять следующим минимальным программно-техническими требованиям:

* процессор Snapdragon 630 или сопоставимые по производительности аналог;
* 2 Гб оперативной памяти;
* диагональ экрана от 4-8 дюйма;
* стабильное подключение к сети Интернет (со скоростью соединения не менее 1 Мбит/с);
* 100 Мб постоянного хранилища;
* операционная система Android версии 5.0 или выше.

## Выполнение программы

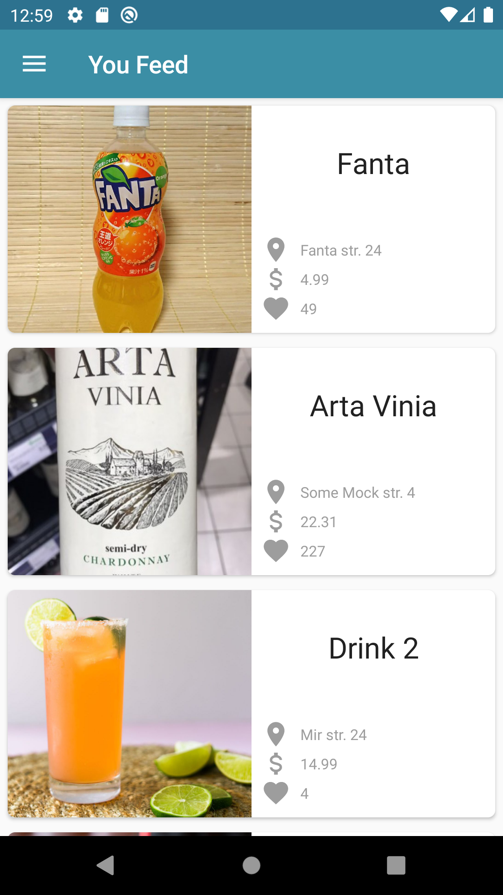
При первом запуске приложения пользователь попадает на экран авторизации – рисунок Г.1. Для успешной авторизации необходимо ввести корректный логин и пароль. В случае ввода некорректных данных будет выдана ошибка – рисунок Г.2.

При последующих запуска приложение перенаправляет пользователя сразу в авторизованную часть.

Авторизация невозможна при отсутствии или нестабильном подключении к интернету.

|  |  |
| --- | --- |
|  |  |
| **Рисунок Г.1** – Экран авторизации | **Рисунок Г.2** – Экран авторизации: некорректные данные входа |

После успешной авторизации запускается главный экран – лента продуктов. Каждый элемент ленты представлен в виде изображения продукта, названия, стоимости и ближайшего адреса продажи – рисунок Г.3.



**Рисунок Г.3** – Лента продуктов

При клике по продукте отображается экран с деталями продукта, рисунок Г.4. Также появляется возможность оценить продукт, оставить отзыв или прочитать уже оставленные комментарии – рисунок Г.5.

|  |  |
| --- | --- |
|  |  |
| **Рисунок Г.4** – Экран деталей: описание продукта | **Рисунок Г.5** – Экран деталей: комментарии |
|  |  |
| **Рисунок Г.6** – Профиль пользователя | **Рисунок Г.7** – Настройка профиля |

При клике по комментарию отображается детали профиля автора – рисунок Г.6.

Из главного меню приложения авторизованному пользователю доступна настройка собственного профиля: имя, email, биография и др. – рисунок Г.7.

Нажатие на кнопку добывания продукта в боковом меню запустит экран нового продукта – рисунок Г.8. После корректного заполнения всех полей, продукт отправляется на проверку модераторам – рисунок Г.9.

|  |  |
| --- | --- |
|  |  |
| **Рисунок Г.8** – Добавление продукта | **Рисунок Г.9** – Добавление продукта: отправлен на проверку |

Кнопка выходы в главном меню вернёт пользователя на экран авторизации. После чего будет необходим повторный вход в систему.

Для смены языка приложения необходимо изменить язык системы. Поддерживается два языка: русский, английский. Экран авторизации на русском и английском языке отображен на рисунке Г.12 и Г13 соответственно.

|  |  |
| --- | --- |
|  |  |
| **Рисунок Г.10** – Экран авторизации: английский язык | **Рисунок Г.11** – Экран авторизации: русский язык |

# ПРИЛОЖЕНИЕ Д (обязательное) РУКОВОДСТВО МОДЕРАТОРА

## Назначение программы

Наименование программного средства – Мобильное приложение «GoodBooze» по поиску напитков для платформы Android (клиентская часть).

Продукт предоставляет пользователям, находящимся в незнакомом для себя месте найти подходящий для них продукт или напиток основываясь на отзывах и комментариях других пользователей.

## Условия выполнения программы

Для корректной работы приложение устройство должно удовлетворять следующим минимальным программно-техническими требованиям:

* процессор Snapdragon 630 или сопоставимые по производительности аналог;
* 2 Гб оперативной памяти;
* диагональ экрана от 4-8 дюйма;
* стабильное подключение к сети Интернет (со скоростью соединения не менее 1 Мбит/с);
* 100 Мб постоянного хранилища;
* операционная система Android версии 5.0 или выше.

## Выполнение программы

Для получения доступа к функционалу модератора необходимо авторизоваться данными модератора.

При входе как модератор, в боковом меню отображается дополнительный функционал – «продукты на проверку», изображен на рисунке Д.1. Данный пункт запускает экран со списком отправленных на проверку продуктов – рисунок Г.2. Модератор может разрешить или запретить публикацию длинным удержанием по элементу в списке продуктов – рисунок Г.3.



**Рисунок Д.1** – Боковое меню модератора

|  |  |
| --- | --- |
|  |  |
| **Рисунок Д.2** – Лента продуктов для проверки | **Рисунок Д.3** – Диалог проверки продукта |

При просмотре комментариев продукта, длинным нажатием модератор может удалять комментарии нарушающие правила публикации.

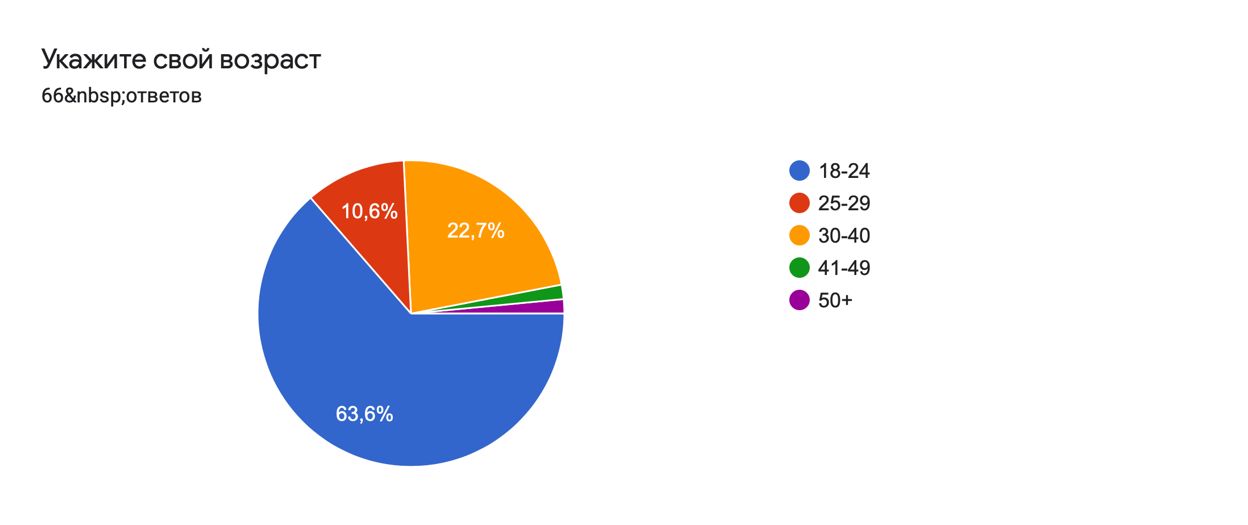
# ПРИЛОЖЕНИЕ Е (необязательное) СОЦИОЛОГИЧЕСКИЙ ОПРОС

Перед реализацией проекта было необходимо провести анализ рынка и создать представление о спросе к разрабатываемому продукт. Для выполнения анализа рынка мобильных приложений был разработан опрос.

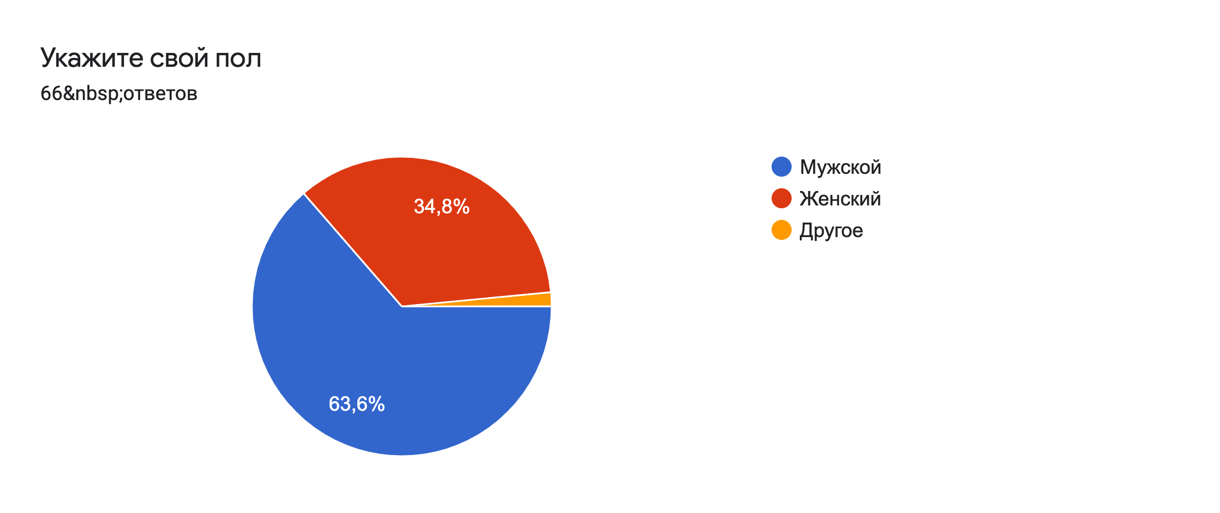
С целью охвата большего культурного и материального разнообразия опрашиваемых, исследование проводилось в двух группах: среди жителей Франции и среди жителей Беларуси.

В результате исследования было опрошено 90 человек. Среди респондентов оказалось 66 белорусов в возрасте от 18 лет; 24 француза в возрасте от 18 до 45 лет.

Графики с результатами опросов среди населения Беларуси приведены на рисунках Е.1 – E.5.



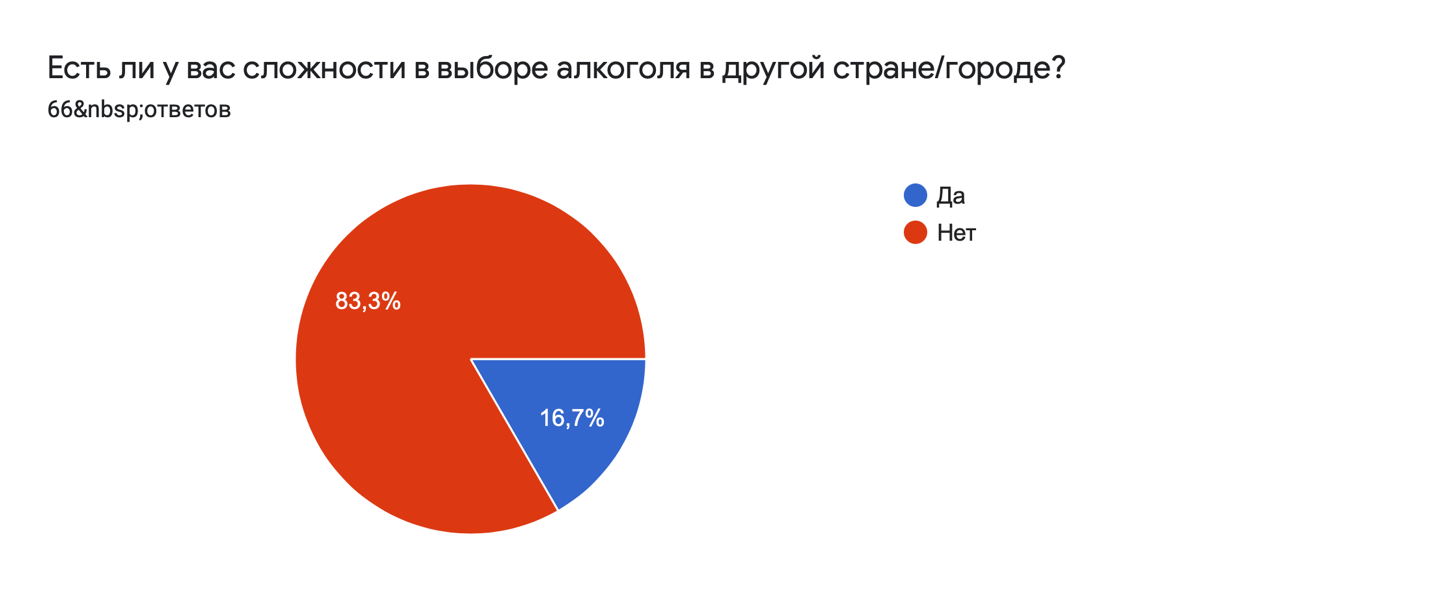
**Рисунок Е.1** – Опрос белорусов: возраст респондента



**Рисунок Е.2** – Опрос белорусов: пол респондента



**Рисунок Е.3** – Опрос белорусов: регулярность путешествий

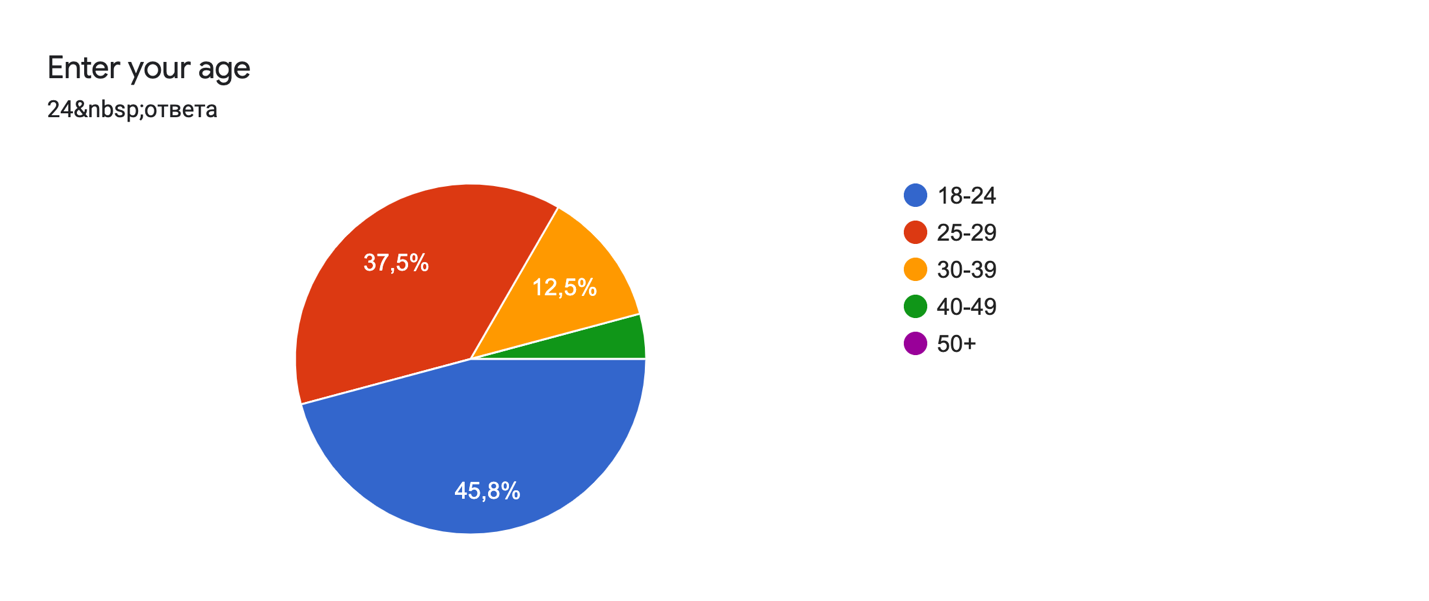


**Рисунок Е.4** – Опрос белорусов: сложность при выборе напитков

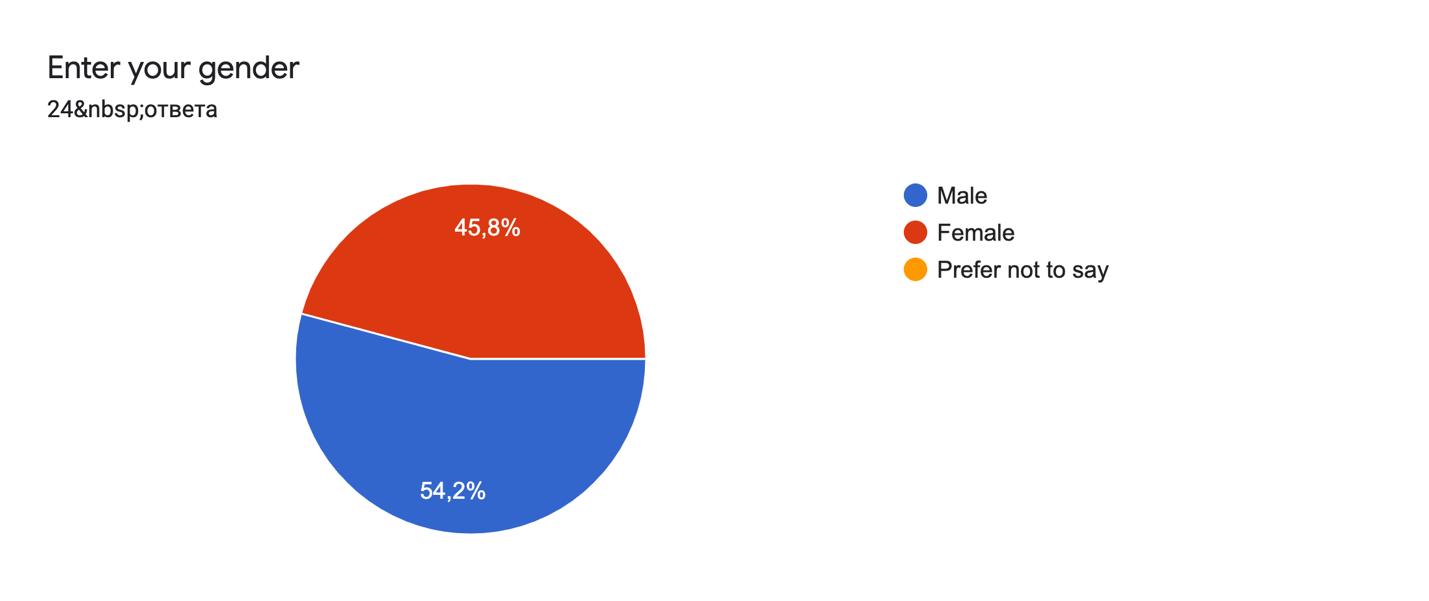


**Рисунок Е.5** – Опрос белорусов: сложность при выборе питания

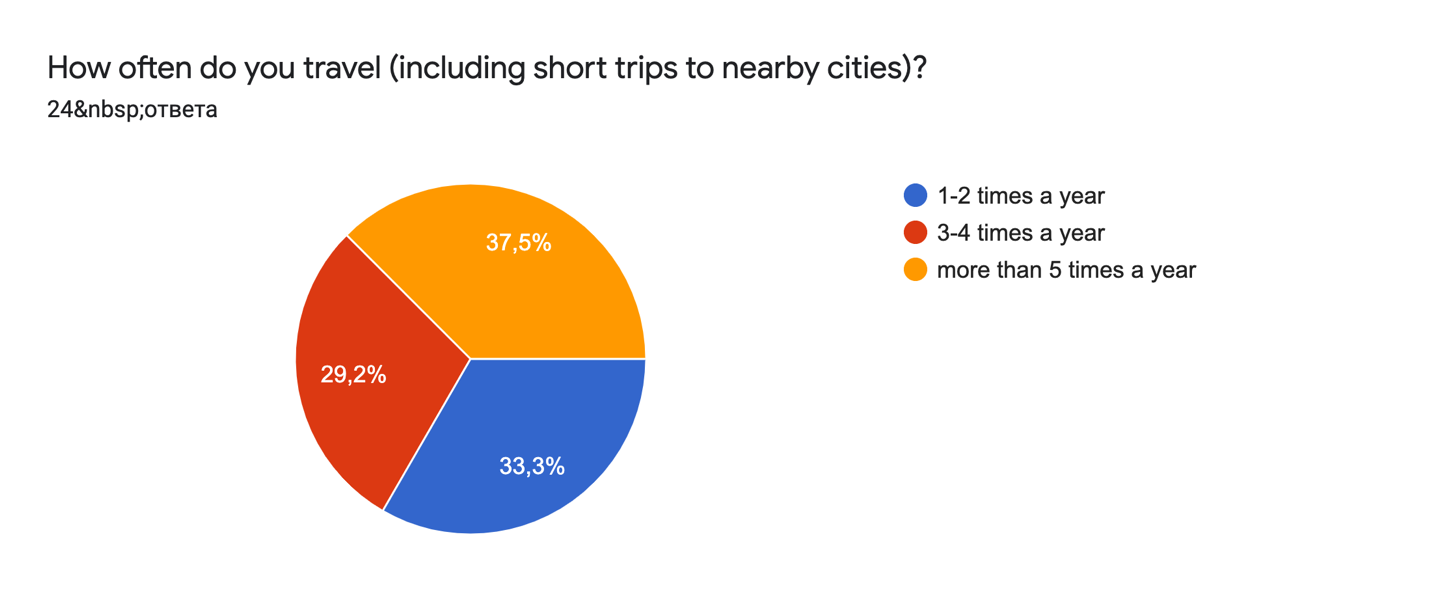
Графики с результатами опросов среди населения Франции с вопросами на английском языке приведены на рисунках Е.6 – E.10.



**Рисунок Е.6** – Опрос французов: возраст респондента



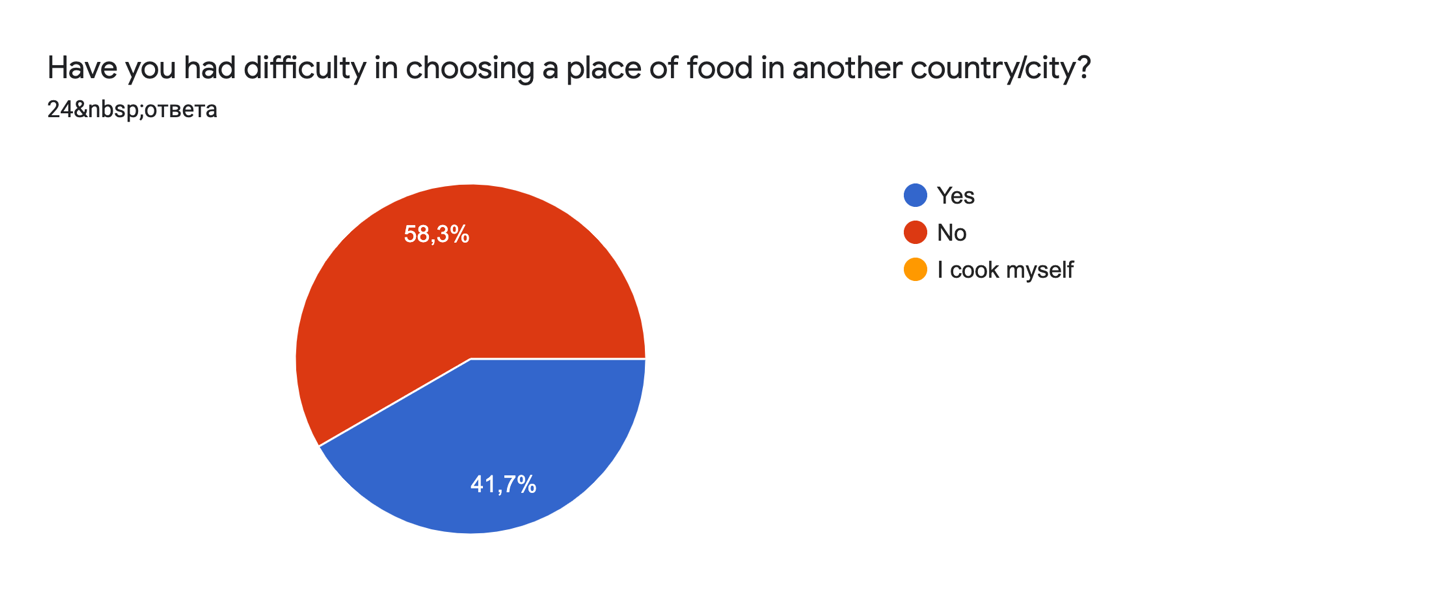
**Рисунок Е.7** – Опрос французов: пол респондента



**Рисунок Е.8** – Опрос французов: регулярность путешествий

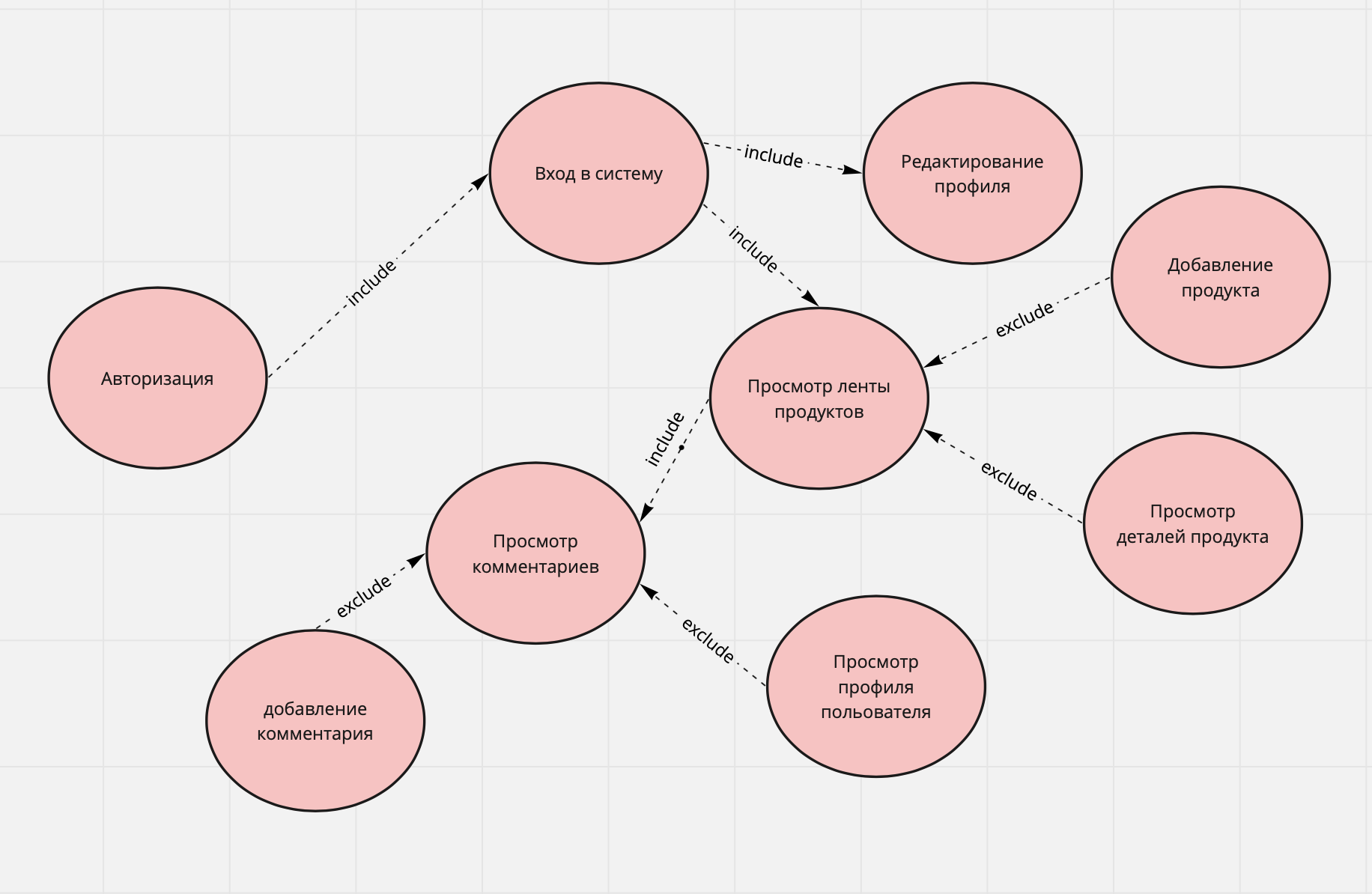


**Рисунок Е.9** – Опрос французов: сложность при выборе напитков

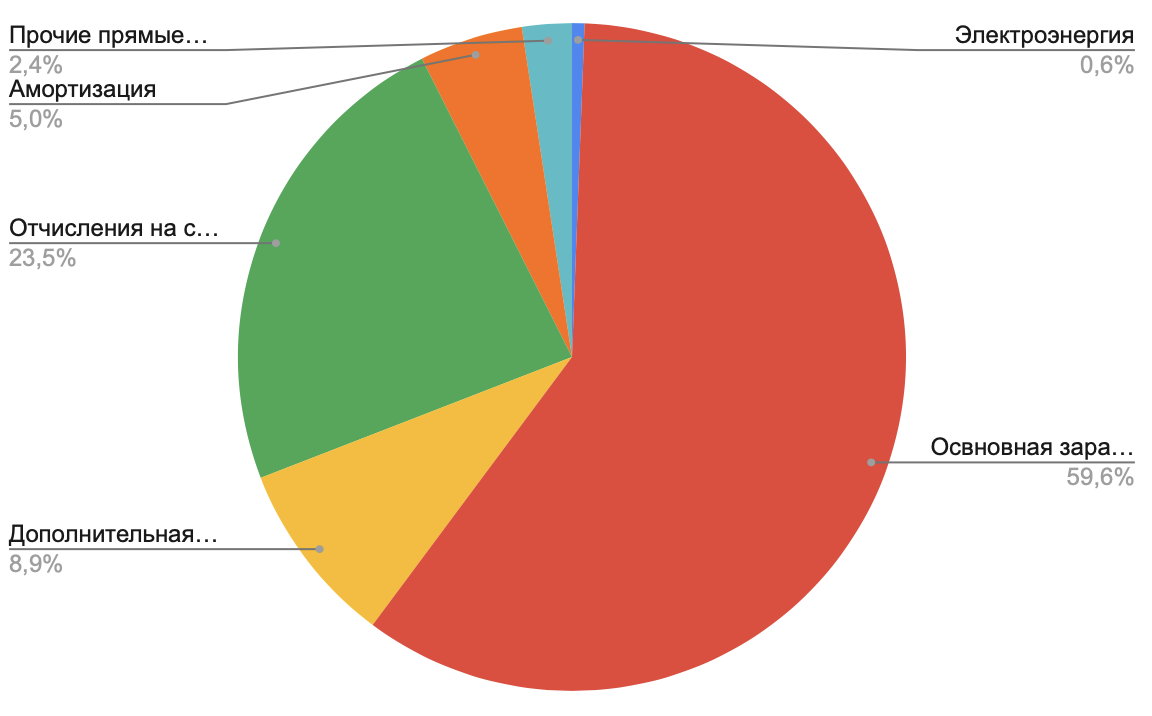


**Рисунок Е.10** – Опрос французов: сложность при выборе питания

**ДИАГРАММА ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ**



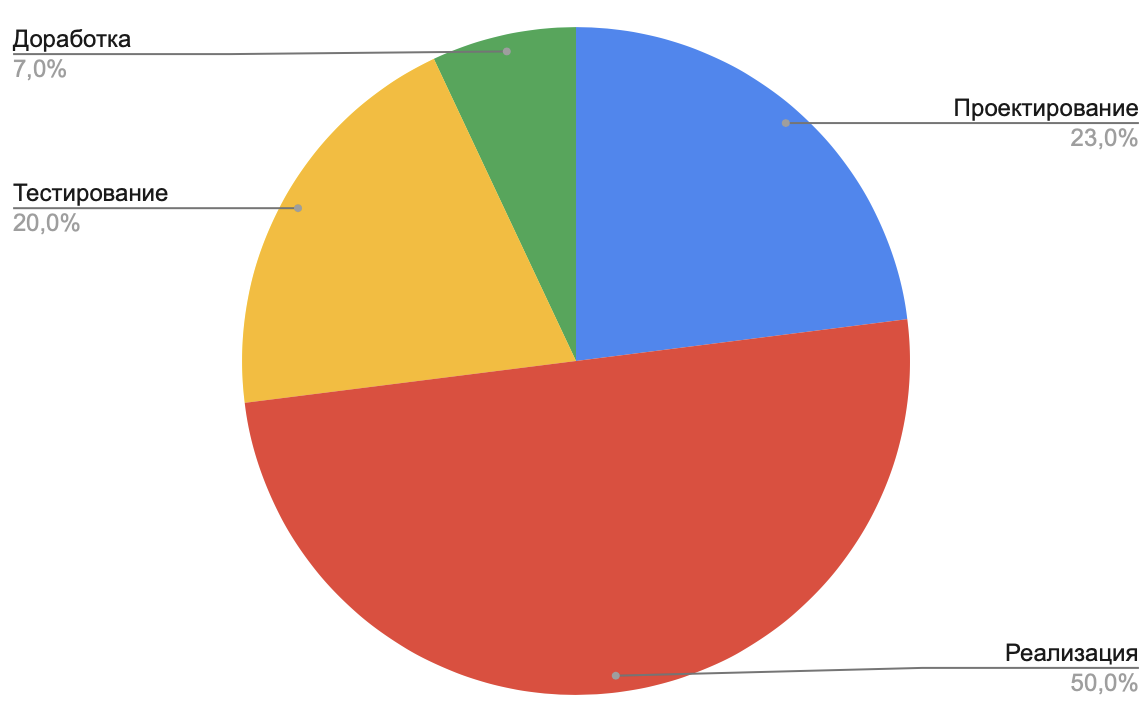
**ДИАГРАММА СЕБЕСТОИМОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

****

**ЭКРАННЫЕ ФОРМЫ ПРИЛОЖЕНИЯ**

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

**ДИАГРАММА РАСПРЕДЕЛЕНИЯ ВРЕМЕНИ РАБОТЫ ПРИ РАЗРАБОТКЕ ПРОДУКТА**

****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Формат | Обозначение | Наименование | Количество  листов | Примечание |
|  |  |  |  |  |
| А4 | - | Задание по дипломному проектированию | 1 |  |
|  |  |  |  |  |
| А4 | - | Реферат | 1 |  |
|  |  |  |  |  |
| А4 | ЯАС.1610574.061.ПЗ | Пояснительная записка | 76 |  |
|  |  |  |  |  |
| А4 | ЯАС.1610574.061.01 | Диаграмма вариантов использования | 1 |  |
|  |  |  |  |  |
| А4 | ЯАС.1610574.061.02 | Диаграмма себестоимости программного обеспечения | 1 |  |
|  |  |  |  |  |
| А4 | ЯАС.1610574.061.03 | Экранные формы приложения | 1 |  |
|  |  |  |  |  |
| А4 | ЯАС.1610574.061.04 | Диаграмма распределения времени работы при разработке продукта | 1 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |