**Содержание**

[Введение 4](#_Toc154006982)

[1 Аналитическая часть 5](#_Toc154006983)

[1.1Описание приложения 5](#_Toc154006984)

[1.2Анализ требований 6](#_Toc154006985)

[1.3 Предварительная оценка трудоемкости разработки 8](#_Toc154006986)

[2 Технологическая часть 10](#_Toc154006987)

[2.1 Описание проектирования логики и данных 10](#_Toc154006988)

[2.2 Определение функциональных типов по данным 12](#_Toc154006989)

[2.3 Описание проекта интерфейса приложения 13](#_Toc154006990)

[2.4 Определение транзакционных функциональных типов 16](#_Toc154006991)

[2.5 Расчет количества функциональных точек 17](#_Toc154006992)

[2.6 Определение основных технико-экономических показателей 19](#_Toc154006993)

[Заключение 22](#_Toc154006994)

[Использованные источники информации 23](#_Toc154006995)

## Введение

Проект “Мир Гитар” – мобильное приложение, созданное с целью помочь людям найти и приобрести гитару мечты, а также аксессуары к ней. Данное приложение предоставляет пользователям возможность искать и покупать гитары, а также изучать информацию о различных брендах и моделях.

Кроме того, приложение позволяет пользователям оставлять отзывы о товарах и общаться с другими пользователями.

Основной целью проекта является создание удобного и функционального приложения, которое поможет людям находить подходящие гитары и аксессуары, а также предоставит информацию о новинках и акциях в магазинах.

Для реализации поставленной цели необходимо решить следующие задачи:

* Проанализировать предметную область продаж музыкальных инструментов;
* Разработать интерфейс мобильного приложения;
* Создать БД необходимую для хранения информации о музыкальных инструментах;
* Разработать API функцию позволяющие работать мобильному приложению;
* Протестировать функционал созданного приложения.

Проект “Мир Гитар” призван помочь пользователям углубить свои знания о мире гитар и музыки, научиться играть на этом инструменте и открыть для себя новые, обширные музыкальные горизонты.

Основанием для разработки проекта является учебный план специальности 09.02.07 «Информационные системы и программирование»

## 1 Аналитическая часть

## 1.1Описание приложения

Мир гитар – это приложение для гитаристов и любителей музыки, которое предлагает широкий спектр функций и возможностей, для выбора инструмента и обучения владения инструментом. Вот описание некоторых основных возможностей этого приложения:

* Каталог инструментов: Приложение содержит огромную базу гитар, что позволяет пользователям выбрать тот инструмент, который будет идеальным именно для них.
* Настройка гитары: с помощью встроенного тюнера можно легко настроить свою гитару и поддерживать ее в идеальном состоянии.
* Метроном: Функция метронома поможет развить чувство ритма и синхронизироваться с музыкой.
* Библиотека аккордов: Приложение содержит список всех базовых аккордов, с помощью которых пользователь уже сможет выучить большое количество разнообразной музыки.
* Социальные функции: Пользователи могут общаться с другими гитаристами, делиться своими достижениями, советоваться по выбору инструмента.

## 1.2Анализ требований

Функциональные требования и возможности представлены на Рис.1 Use Case диаграмма

Use case диаграмма представляет собой визуальное описание функциональных требований системы через взаимодействие её компонентов – актеров и использования случаев. В данном случае, мобильное приложение “Мир гитар” может иметь следующие основные актеры: “Пользователь” и “Администратор”

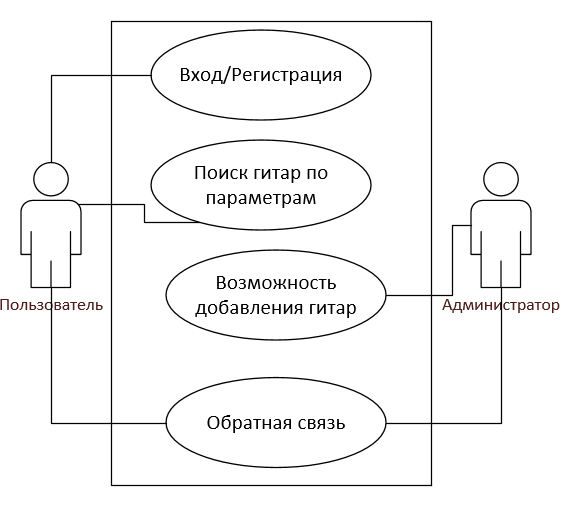


Рисунок 1. Диаграмма Use-Case

Основным прецедентом является «Поиск гитар по параметрам».

Таблица 1 — Прецедент “Поиск по параметрам”

|  |  |
| --- | --- |
| **Название прецедента** | ***Поиск гитар по параметрам*** |
| **Цель** | Выбор интересующую модель гитары. |
| **Актеры** | Пользователь |
| **Ссылки (предусловия)** | Реализованы прецеденты ***Возможность добавления гитар.*** |
| **Результаты (постусловия)** | Пользователь выбирает интересующие его модели гитар. |

Таблица 2 — Основной успешный сценарий

|  |  |
| --- | --- |
| **Действия актера** | **Отклик системы** |
| 1. Пользователь заходит в приложение. 2. Пользователь выбирает инструмент 3. Пользователь просматривает характеристики. | 1. Отображает список инструментов. 2. Отображает характеристику, выбранного инструмента. |

**Альтернативные потоки событий**

**\*.** Произошла ошибка в системе. Система выдает сообщение об ошибке.

**3.** Данные отсутствуют в системе. Система выдает сообщение об ошибке.

**5.** Характеристика гитар недоступна. Система выдает сообщение об ошибке.

Таким образом, была создана Use-Case диаграмма и рассмотрен один из прецедентов.

## 1.3 Предварительная оценка трудоемкости разработки

На основании перечня функций, выявленных на этапе анализа требований была выполнена экспертная оценка трудоемкости разработки приложения, с учетом его функций, представленная в таблице 3.

Таблица 3. Экспертная оценка трудоемкости разработки приложения

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция** | **Лучший случай** | **Наиболее вероятный** | **Худший случай** | **Ожидаемый случай** |
| Возможность добавления гитар | 4 | 6 | 26 | *8* |
| *Возможность поиска гитар по фильтрам* | 4 | 6 | 26 | *8* |
| **Итого** | **14** | **24** | **112** | **16** |

В предварительной оценке трудоёмкости разработки представлено количество дней, за которое возможно реализовать функции из Use-Case диаграммы используя формулу PERT (табл.3).

Оценку средней трудоемкости по каждому элементарному пакету можно определить по формуле:

(1)

1. Выбор игрока.

Oi = 4

Mi = 8

Pi = 32

Средняя трудоемкость рассчитана по формуле (1):

2) Выбор инструмента.

Oi = 4

Mi = 8

Pi = 32

Средняя трудоемкость рассчитана по формуле (1):

На основе экспертная оценки трудоемкости разработки мобильного приложения приложения было выявлено, что ожидаемый случай выполнения разработки программного продукта займет 16 дней

# 2 Технологическая часть

# 2.1 Описание проектирования логики и данных

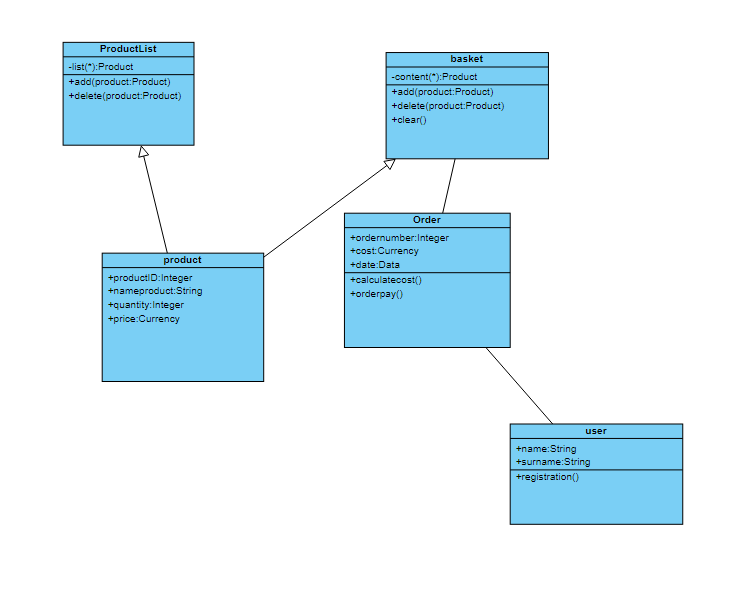


Рисунок 2. Диаграмма классов.

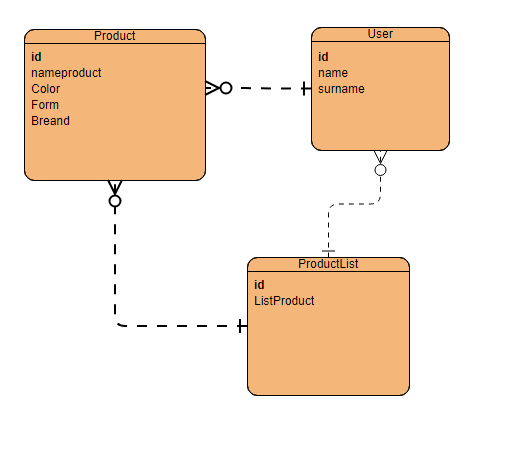


Рисунок 3. ER - Диаграмма.

# 2.2 Определение функциональных типов по данным

На основании ER - диаграммы выполнили оценку функциональных типов по данным. Сначала определили тип данных системы: EIF или ILF. Затем произвели подсчет функциональных точек, связанных с данными, определили сложность данных.

Таблица 4 – Экспертная оценка трудоемкости разработки приложения

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Информационный объект** | **Тип** | **RET** | **DET** | **Сложность** | **Коэффициент** |
| User | ELF | 1 | 4 | Простая | 7 |
| Product | ELF | 1 | 4 | Простая | 7 |
| ProductList | ELF | 1 | 4 | Простая | 7 |

Коэффициент определяется в зависимости от принадлежности объекта к ILF или EIF.

# 2.3 Описание проекта интерфейса приложения

При анализе предметной области был спроектирован проект интерфейса приложения, представленный на рисунках 4-6. Каждая форма выполняет определенную функцию.

|  |  |
| --- | --- |
| Рисунок 4. Экран загрузки | Рисунок 5. Экран регистрации |

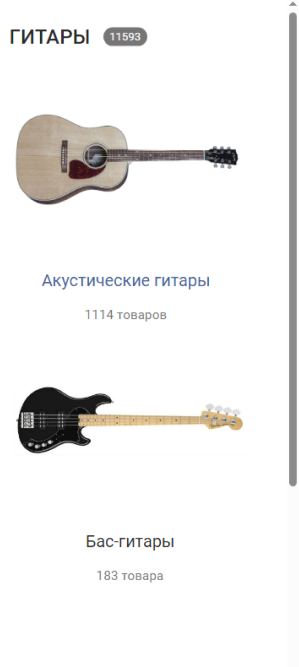


Рисунок 6. Экран каталога товаров

После создания проекта интерфейса была создана схема перемещений по приложению, представленная на рисунке 7:

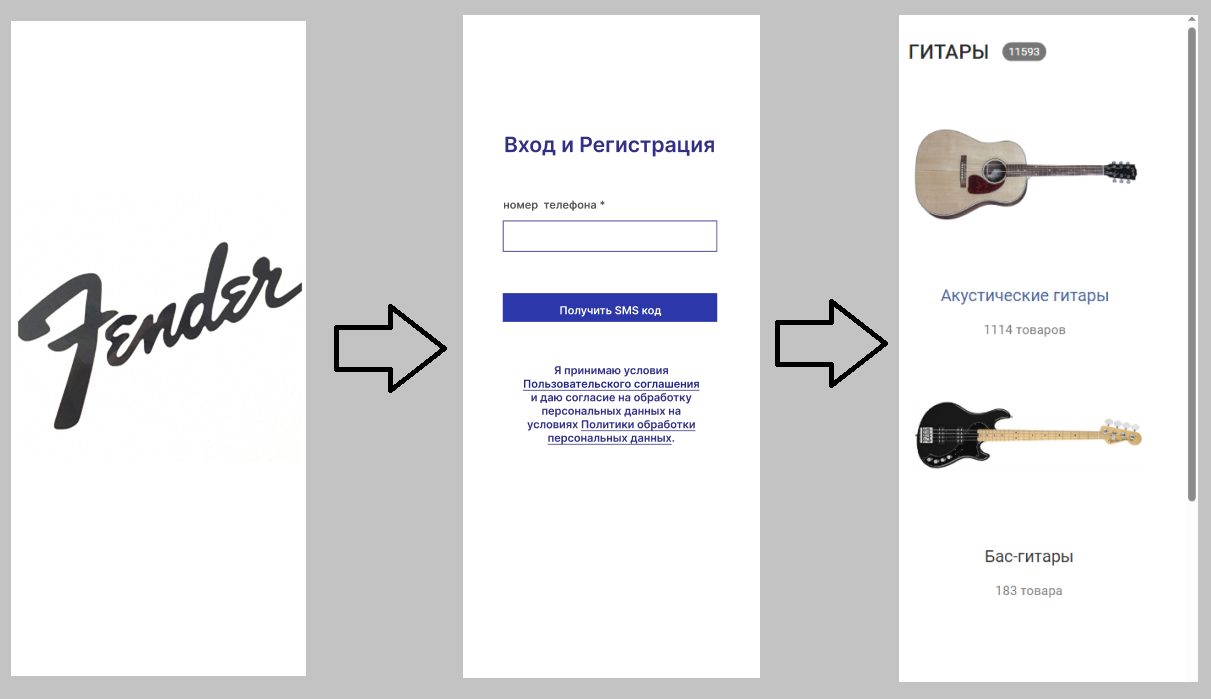


Рисунок 7. Окно “ Схема перемещений по приложению”

На рисунке ясно видно, как происходит работа в приложении.

# 2.4 Определение транзакционных функциональных типов

Следующий шаг анализа по методу функциональных точек производился для каждой формы приложения. Были определены типы транзакций, выполнена оценка сложности транзакций, а также определение сложности и расчет коэффициента. Все расчеты представлены в таблицах 5-7.

Таблица 5 – Форма Загрузочный экран

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Транзакция** | **Тип** | **FTR** | **DET** | **Сложность** | **Коэффициент** |
| Загрузка | EI | 1 | 1 | Легкая | 3 |

Таблица 6 – Форма выбора каталога

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Транзакция** | **Тип** | **FTR** | **DET** | **Сложность** | **Коэффициент** |
| Выбвод каталог гитар | EI | 1 | 1 | Легкая | 3 |

Таблица 7 – Форма Выбор инструмента

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Транзакция** | **Тип** | **FTR** | **DET** | **Сложность** | **Коэффициент** |
| Выбвод списка гитар | EI | 1 | 1 | Легкая | 3 |
| Выбор гитаы | EI | 1 | 1 | Легкая | 3 |

# 2.5 Расчет количества функциональных точек

Таблица 8 — Основной успешный сценарий

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Категория функциональных типов** | **Простые** | **Средние** | **Сложные** | **Кол-во точек** |
| Количество внутренних логических файлов | *3* | *2* | *0* | *5* |
| Количество внешних интерфейсных файлов | *6* | *6* | *0* | *12* |
| Количество входных элементов | *2* | *0* | *0* | *2* |
| Количество выходных элементов | *2* | *0* | *0* | *2* |
| Количество внешних запросов | *7* | *3* | *0* | *10* |
| **Количество функциональных точек (UFP)** | | | | 31 |

Чтобы определить суммарное количество не выровненных функциональных точек, нужно рассчитать по формуле:

Общее суммарное количество не выровненных функциональных точек UFP будет составлять по формуле (2):

(2)

Таблица 12 — Основной успешный сценарий

|  |  |  |
| --- | --- | --- |
| **№** | **Наименование фактора** | **Значение** |
| 1 | Обмен данными | 5 |
| 2 | Распределенная обработка данных | 0 |
| 3 | Производительность | 5 |
| 4 | Эксплуатационные ограничения | 0 |
| 5 | Частота транзакций | 0 |
| 6 | Ввод данных в режиме «онлайн» | 5 |
| 7 | Эффективность работы конечных пользователей | 5 |
| 8 | Онлайновое обновление | 0 |
| 9 | Сложная обработка | 0 |
| 10 | Повторное использование | 5 |
| 11 | Простота установки | 5 |
| 12 | Простота эксплуатации | 5 |
| 13 | Количество возможных установок на различных платформах | 0 |
| **Суммарное значение коэффициентов (М)** | | 35 |

Суммарное значение коэффициентов осуществляется простым суммированием по формуле:

Расчёт суммарного значения коэффициентов будет составлять по формуле (3):

(3)

Расчёт значения фактора выравнивания производится по формуле:

Расчёт значения факторов выравнивания будет составлять по формуле (4):

(4)

Начальная оценка количества выровненных функциональных точек для программного приложения определяется по формуле:

Начальная оценка количества выровненных функциональных точек для программного приложения будет составлять по формуле (5):

(5)

# 2.6 Определение основных технико-экономических показателей

Расчёт количества функциональных точек:

Показатели LOC для языка Kotlin:

Для определения технико-экономических показателей нужно сначала определить тип системы. Данная система относится ко второму типу, так как будет реализована на языке Kotlin.

После расчёта количества функциональных точек необходимо определить размерность мобильного приложения. Размерность определяется по формуле:

Размерность мобильного приложения рассчитана по формуле (6):

(6)

Значения параметров A и E определяются из таблицы коэффициентов математической модели оценки трудозатрат в зависимости от типа программной системы.

Оценка трудозатрат проводится с помощью степенной функции базовой модели COCOMO. Значения параметров A и E определяются из таблицы коэффициентов математической модели оценки трудозатрат в зависимости от типа программной системы.

Единица измерения R в данной формуле - тысяча строк:

Расчет количества человек в месяц произведен по формуле (7):

(7)

Средняя численность сотрудников, занятых в проекте, срок реализации которого 4 месяца, составляет:

Средняя численность сотрудников рассчитана по формуле (8):

(8)

Таким образом, метод функциональных точек определил следующие основные технико-экономические показатели:

* трудозатраты на разработку системы за 4 месяца составят 4 человеко-месяцев;
* необходимые людские ресурсы при реализации системы за 4 месяца составят 1 чел.

# Заключение

До начала реальной части проекта был произведён расчёт трудоёмкости работы. Было выявлено, какое количество человеко-часов необходимо для разработки проекта.

В результате работы была создана логическая модель базы данных в документе Microsoft Visio. Разработаны формы экранов, а также схема перемещения между ними. Информационная система состоит из:

* логической модели базы данных, разработанной в Microsoft Visio;
* формы экранов будущего приложения, разработанные в Microsoft Visio.

В программе предусмотрены возможности:

* просмотра данных гитар;
* регистрации и авторизации.

В ходе работы закреплена технология проектировки баз данных и диаграмм классов в Microsoft Visio, а также планирования создания форм и перемещений между ними будущих экранов мобильного приложения. Были изучены и проведены расчёты необходимого времени и количества человек на разработку приложения.

# Использованные источники информации

1. ГОСТ Р ИСО/МЭК 12207-99 «Процессы жизненного цикла программных средств»;

2. Нестеров С. А. Интеллектуальный анализ данных с использованием SQL Server : учебник для вузов / С. А. Нестеров. – Санкт-Петербург : Лань, 2023. – 160 с.;

3. ГОСТ Р ИСО 9000-2008 «Системы менеджмента качества»;

4. Макшанов А. В. Большие данные. Big Data : учебник для СПО / А. В. Макшанов, А. Е. Журавлев, Л. Н. Тындыкарь. – 3-е изд., стер – Санкт-Петербург : Лань, 2023. – 188 с.;

5. Васильева М. А. Система контроля версий. Основы командной разработки : учебное пособие для вузов / М. А. Васильева, К. М. Филипченко. – Санкт-Петербург : Лань, 2022. – 144 с.;

6. Заяц А. М. Введение в гибридные технологии разработки мобильных приложений : учебное пособие для СПО / А. М. Заяц, Н. П. Васильева. – 2-е изд., стер. – Санкт-Петербург : Лань, 2022. – 160 с.;

7. Старолетов С. М. Основы тестирования программного обеспечения : учебное пособие для СПО / С. М. Старолетов. – 3-е изд., стер. – Санкт-Петербург : Лань, 2023. – 192 с.;

8. Васильев Н. П. Введение в гибридные технологии разработки мобильных приложений : учебное пособие для вузов / Н. П. Васильев, А. М. Заяц. – 3-е изд., стер. – Санкт-Петербург : Лань, 2022. – 160 с.;

9. ГОСТ Р 58412-2019 «Защита информации»;

10. Волк В. К. Базы данных. Проектирование, программирование, управление и администрирование : учебник для СПО / В. К. Волк. – 2-е изд., стер. – Санкт-Петербург : Лань, 2022. – 340 с.;