Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

К ЗАЩИТЕ ДОПУСТИТЬ

Зав. каф. ЭВМ

\_\_\_\_\_\_\_\_\_\_ Б.Н. Никульшин

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к дипломному проекту

на тему

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ ПРОСМОТРА И

ОБРАБОТКИ ИНФОРМАЦИИ О РАСПИСАНИИ ЗАНЯТИЙ

БГУИР ДП 1–40 02 01 01 019 ПЗ

Студент А.В. Гуринович

Руководитель Д.В. Басак

Консультанты:

от кафедры ЭВМ Д.В. Басак

по экономической части В.Г. Горовой

Нормоконтролер Е.Е. Клинцевич

Рецензент

МИНСК 2023

РЕФЕРАТ

Дипломный проект предоставлен следующим образом. Электронные носители: 1 компакт-диск. Чертёжный материал: 5 листов формата А1. Пояснительная записка: N страниц, N рисунков, N таблиц, N литературных источников, X приложений.

Ключевые слова: Swift, SwiftUI, CoreData, App Store, мобильная разработка, масштабируемые графические пользовательские интерфейсы, iOS, iPadOS, macOS, REST API.

Предметной областью данного проекта является расписание занятий Белорусского государственного университета информатики и радиоэлектроники. Объектом разработки является кроссплатформенное нативное приложение для просмотра и обработки информации о расписании занятий Белорусского государственного университета информатики и радиоэлектроники.

Целью разработки данного дипломного проекта является создание приложения с высокой степенью удобства и быстродействия, которое будет использовать всю доступную через интерфейс программирования приложений интегрированной информационной системы Белорусского государственного университета информатики и радиоэлектроники информацию для отображения расписания, и иной информации.

Для разработки данного проект использовалась интегрированная среда разработки Xcode, язык программирования Swift и платформа SwiftUI для создания пользовательского интерфейса.

В результате разработки создано программное средство для просмотра и обработки информации о расписании занятий для операционных систем iOS, iPadOS и macOS. Программное средство доступно для загрузки на платформе Apple App Store.

Практическим применением разработки является удобное и быстрое получение структурированной информации для студентов и сотрудников университета. Пользователи смогут легко находить информацию о своих занятиях, преподавателях, группах, кабинетах, а также добавлять задания со сроком выполнения и управлять ими. Проект улучшает опыт обучения в университете для пользователей на платформах Apple, снижает время, затрачиваемое на получение и обработку информации.

Проект является эффективным с экономической стороны, при этом эффективно выражается не только в прямой прибыли от продаж, но и в повышении эффективности пользователей в процессе обучения.

Данные проект полностью завершён и доступен пользователям на рынке, в частности в Apple App Store. Дальнейшему развитию приложения может поспособствовать открытый исходный код, который доступен на web-сервисе GitHub, где другие пользователи могут сообщать об ошибках, предлагать и реализовывать улучшения программного средства.

Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет: ФКСиС. Кафедра: ЭВМ.

Специальность: 40 02 01 «Вычислительные машины, системы и сети».

Специализация: 40 02 01-01 «Проектирование и применение локальных компьютерных сетей».

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

\_\_\_\_\_\_\_\_\_ Б.В.Никульшин

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2023 г.

ЗАДАНИЕ

по дипломному проекту студента

Гуриновича Андрея Викторовича

1 Тема проекта: «Программное средство для просмотра и обработки информации о расписании занятий» – утверждена приказом по университету от 24 марта 2023 г. № 743-с.

**2** Срок сдачи студентом законченного проекта: 1 июня 2023 г.

**3** Исходные данные к проекту:

1. Получение данных из интегрированной информационной системы Белорусского государственного университета информатики и радиоэлектроники.
2. Операционные системы: iOS, iPadOS и macOS.
3. Среда разработки: Xcode.
4. Язык программирования: Swift.

**4** Содержание пояснительной записки (перечень подлежащих разработкевопросов):

Введение 1. Обзор литературы. 2. Системное проектирование. 3. Функциональное проектирование. 4. Разработка программных модулей. 5. Программа и методика испытаний. 6. Руководство пользователя. 7. Технико-экономическое обоснование разработки и реализации на рынке программного средства для просмотра и обработки информации о расписании занятий. Заключение. Список использованных источников. Приложения.

**5** Перечень графического материала (с точным указанием обязательных чертежей):

1. Вводный плакат. Плакат.
2. Заключительный плакат. Плакат.
3. Программное средство для просмотра и обработки информации о расписании занятий. Схема структурная.
4. Программное средство для просмотра и обработки информации о расписании занятий. Модель данных.
5. Программное средство для просмотра и обработки информации о расписании занятий. Диаграмма последовательностей.
6. Программное средство для просмотра и обработки информации о расписании занятий. Диаграмма последовательностей.
7. Программное средство для просмотра и обработки информации о расписании занятий. Схема программы.

**6** Содержание задания по экономической части: «Технико-экономическое обоснование разработки и реализации на рынке средства для просмотра и обработки информации о расписании занятий».

ЗАДАНИЕ ВЫДАЛ \_\_\_\_\_\_\_\_ В.Г. Горовой

КАЛЕНДАРНЫЙ ПЛАН

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование этапов дипломного проекта | Объем этапа, % | Срок выполнения этапа | Примечания |
| Подбор и изучение литературы | 10 | 23.03 – 29.03 |  |
| Структурное проектирование | 10 | 29.04 – 05.04 |  |
| Функциональное проектирование | 20 | 05.04 – 18.04 |  |
| Разработка программных модулей | 30 | 18.04 – 08.05 |  |
| Программа и методика испытаний | 10 | 08.05 – 11.05 |  |
| Расчёт экономической эффективности | 10 | 11.05 – 15.05 |  |
| Оформление пояснительной записки | 10 | 15.05 – 21.05 |  |

Дата выдачи задания: 23 марта 2023 г.

Руководитель \_\_\_\_\_\_\_\_ Д.В. Басак

ЗАДАНИЕ ПРИНЯЛ К ИСПОЛНЕНИЮ \_\_\_\_\_\_\_\_ А.В. Гуринович

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 6](#_Toc132676050)

[1 Обзор литературы 7](#_Toc132676051)

[1.1 Выбор платформы для разработки 7](#_Toc132676052)

[2 Системное проектирование 10](#_Toc132676053)

[3 Функциональное проектирование 11](#_Toc132676054)

[4 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ̆ 12](#_Toc132676055)

[5 Программа и методика испытаний 13](#_Toc132676056)

[6 Руководство пользователя 14](#_Toc132676057)

[7 Технико-экономическое обоснование разработки и реализации на рынке ПРОГРАММНОГО средства для просмотра и обработки информации о расписании занятий 15](#_Toc132676058)

[7.1 Характеристика программного средства, разрабатываемого для реализации на рынке 15](#_Toc132676059)

[7.2 Расчёт инвестиций в разработку программного средства 15](#_Toc132676060)

[7.3 Расчёт экономического эффекта от реализации программного средства на рынке 18](#_Toc132676061)

[7.4 Расчёт показателей экономической эффективности разработки и реализации программного средства на рынке 20](#_Toc132676062)

[7.5 Вывод об экономической целесообразности реализации проектного решения 20](#_Toc132676063)

[Заключение 22](#_Toc132676064)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 23](#_Toc132676065)

[ПРИЛОЖЕНИЕ А 25](#_Toc132676066)

[ПРИЛОЖЕНИЕ Б 26](#_Toc132676067)

ВВЕДЕНИЕ

# Обзор литературы

## Выбор платформы для разработки

Создание приложений на языке программирования Swift в первую очередь подразумевает использование трёх платформ: UIKit для iOS и iPadOS или AppKit для macOS и SwiftUI для всех операционных систем (далее – ОС). Они используют соответственно парадигмы Model-View-Controller и View-ViewModel-Model.

### Model-View-Controller

Model-View-Controller (далее – MVC) является паттерном, который разделяет приложение Модель (Model), Представление (View) и Контроллер (Controller) [[[1]](#endnote-2)].

Модель отвечает за хранение данных и логику обработки данных: в ней описывается бизнес-логика, правила валидации и другие операции, связанные с данными. Модель не имеет информации о Представлением и Контроллером, то есть не имеет доступа к их полям, функциям и иному.

Представление отображает данные пользователю и предоставляет возможность взаимодействия с приложением через графический интерфейс. В ней определяется внешний вид и макет интерфейса, это происходит в графическом редакторе. Представление не имеет представления о существовании Модели и Контроллера.

Контроллер обрабатывает пользовательский ввод и управляет взаимодействием Модели и Представления. В ней содержится логика приложения, отвечающая за обработку событий, получаемых от Представления, и изменение Модели соответствующим образом. Контроллер имеет исходящую связь с Представлением или Моделью.

### UIKit и AppKit

UIKit является основной платформой для создания пользовательских интерфейсов на iOS и iPadOS [[[2]](#endnote-3)]. В платформе применяется в первую очередь императивное программирование. UiKit представляет собой набор готовых компонентов интерфейса, таких как кнопки, текстовые поля, таблицы, контейнеры, иные элементы интерфейса и контейнеры для них. SwiftUI паттерн проектирования архитектуры MVC.

В целом, UIKit и SwiftUI представляют собой две разные платформы для создания пользовательских интерфейсов в операционных системах Apple, каждая со своими плюсами и минусами. Разработчики должны выбирать платформу, которая лучше всего подходит для их задач и опыта в программировании.

AppKit же является платформой для создания приложений под macOS [[[3]](#endnote-4)]. Платформа является набором инструментов и ресурсов, который разработчики могут использовать для создания графических пользовательских интерфейсов.

UIKit и AppKit имеют следующую парадигму для создания пользовательских интерфейсов: используются файлы xib или Storyboard, в которых определяются все элементы интерфейса и их свойства, а уже в файлах Swift программируется поведение и особенности интерфейса, программный код в данном случае пишется в императивной парадигме. Обе платформы используют паттерн Model-View-Controller.

AppKit и UIKit очень похожи и имеют одинаковую структуру и логику. Элементы имеют одинаковые названия и одинаковые параметры, хоть и существуют особенные для платформ элементы. Несмотря на это, эти платформы не являются взаимозаменяемыми, хоть технически и существует возможность создания приложения сразу на двух платформах, по сути это два разных приложения, которые объединены только Моделью, а все элементы графического интерфейса необходимо имплементировать отдельно для каждой из платформ.

### Model-View-ViewModel

Model-View-ViewModel (далее – MVVM) является схожим с MVC паттерном, используемым для разделения компонентов приложения на три основных уровня: Модель, Представление и Модель Представления (ViewModel).

Модель, аналогична Модели MVC, не имеет информации о Представлением и Модели Представления.

Представление, частично похоже на MVC, отображает данные пользователю и обрабатывает пользовательский ввод. При это в отличии от MVC, представление напрямую обращается к Модели Представления, последняя чаще всего является полем Представления.

Модель Представления является посредником между моделью и представлением и чем-то схожа с Контроллером MVC. Модель обновляет Представления при изменении данных Модели. В MVC же Контроллер сам запрашивает обновления модели.

Одним из недостатков MVVM является некоторая его избыточность для простых приложений, которые имеют небольшое количеством логики и данных. MVVM требует больше кода, чем, например, простая парадигма Model-View-Controller. При этом эти недостатки нивелируются при разработке относительно больших приложений, так как MVVM предоставляет огромные возможности для повторного использования различных элементов Представления. На начальных этапах разработка приложение, использующего платформу MVVM, может быть затратным с точки зрения количества кода и времени разработки, однако повторное использование и универсальность стандартизированных представлений.

### SwiftUI

SwiftUI является более современной платформа для создания пользовательских интерфейсов, которая была запущена в 2019 году [[[4]](#endnote-5)]. В платформе применяется в первую очередь декларативное программирование. SwiftUI основан на декларативном подходе, где для создания интерфейса описывается его конечное представление в виде дерева вложенных представлений. SwiftUI паттерн проектирования архитектуры MVVM.

Одним из основных преимуществ SwiftUI является более простой и интуитивный синтаксис для описания пользовательского интерфейса. Кроме того, SwiftUI обладает рядом функций, которых нет в UIKit, таких как автоматические анимации и адаптивный дизайн, что упрощает создание интерфейсов, которые могут адаптироваться к разным размерам экранов.

Также преимуществом SwiftUI является возможность непосредственной работы с данными в представлениях, благодаря связыванию данных (Data Binding), который позволяет отслеживать изменения данных в Модели или Модели Представления и автоматически обновлять Представление, что значительно упрощает разработку приложений.

На данный момент SwiftUI не поддерживает все функции, которые доступны в UIKit, для SwiftUI 4.0 чаще всего это узкоспециализированные возможности. Для добавления элементов из UIKit в структуру SwiftUI существует UIViewControllerRepresentable [[[5]](#endnote-6)], который позволяет взаимодействовать с элементами UIKit и преобразовывать их в элементы SwiftUI.

### Сравнение UIKit и AppKit с SwiftUI в контексте разрабатываемого программного средства

Среди исходных данных к дипломному проекту приведена кроссплатформенность, а именно реализация приложения для iOS, iPadOS и macOS. Разработка кроссплатформенного приложения UIKit и AppKit потребует большего количества времени, так как Представление и Контроллер необходимо разрабатывать отдельно для каждой платформы. Хоть это и были бы крайне схожие элементы, необходимость параллельной разработки значительно затруднила бы какие-либо изменения проекта. Однако, среди преимуществ UIKit и AppKit можно выделить возможность создания более продвинуты и эстетически нестандартных пользовательских графических интерфейсов

# Системное проектирование

Системное проектирование – аналог разработки структурной̆ схемы для аппаратных дипломных проектов. В этом разделе на основе системного подхода определяется обобщённая структура пакета программ, программы или модуля, описывается назначение выделенных блоков, а также связи между ними. Если программные средства разработки не заданы, то производится их выбор. Данный раздел должен сопровождать схему структурную либо схему работы системы и содержать ссылки на неё.

# Функциональное проектирование

Функциональное проектирование – аналог соответствующего раздела аппаратного дипломного проекта. Это основной раздел ПЗ, дающий ключ к пониманию функционирования разрабатываемой программы и исчерпывающую информацию о её структуре с точки зрения описания данных и обрабатывающих их подпрограмм (функций и процедур). Обработка данных, как известно, является основной целью работы любой программы. Поэтому здесь описываются заданные константы, пользовательские переменные, внутренние и внешние массивы и так далее. При определении необходимости в какой-либо обработке данных вводится соответствующая подпрограмма. При использовании объектно-ориентированного подхода могут описываться структура и взаимоотношения между классами со ссылками на чертеж диаграммы классов. Если при программировании (особенно в системах визуального программирования) используются стандартные библиотеки, то упор должен делаться на описание самостоятельно разработанных фрагментов кода. Стандартные функции лишь упоминаются. При использовании базы данных описывается структура таблиц и связи между ними со ссылками на чертеж модели данных. Здесь же раскрываются и потоки данных между подпрограммами, т.е. входные и выходные аргументы процедур и функций. Рекомендуется структурировать раздел в соответствии с блоками, выделенными на этапе системного проектирования. Данный раздел должен сопровождать чертеж диаграммы последовательности и содержать ссылки на него. Могут быть ссылки на чертеж схемы данных.

# РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ̆

Разработка программных модулей̆ – аналог разработки принципиальной схемы аппаратного дипломного проекта. В этом разделе подробно описываются уже внутренние алгоритмы ключевых процедур и функций с разбиением на отдельные подразделы. Здесь же описывается реализация наиболее интересных алгоритмов, например, алгоритмов шифрования. Данный раздел может сопровождать чертежи схем программ и содержать ссылки на них.

# Программа и методика испытаний

В разделе, посвящённом программе и методике испытаний, описываются внутренние (если самотестирование заложено в программу) и внешние средства тестирования. Могут использоваться как оригинальные, так и стандартные тесты. Рассматриваются способы проверки надёжности (устойчивости, стабильности и т. д.) разработанной программы в различных режимах, включая многопользовательский и многозадачный режимы, а также корректность обработки входных, промежуточных и выходных данных, в том числе: в области граничных значений допустимых диапазонов, заведомого неправильных данных, файлов большого размера. Для каждого из тестов приводятся исходные данные, параметры и результаты.

# Руководство пользователя

В руководстве пользователя даётся описание работы с программой. Указываются требования к аппаратному и программному обеспечению. Описывается процесс инсталляции с указанием каталогов, ключей реестра конфигурационных файлов и так далее. Также описывается пользовательский интерфейс с указанием элементов управления (пунктов меню, кнопок, закладок и т. д.), режимов работы и последовательности действий. Здесь могут приводиться скриншоты работы программы.

# Технико-экономическое обоснование разработки и реализации на рынке ПРОГРАММНОГО средства для просмотра и обработки информации о расписании занятий

## Характеристика программного средства, разрабатываемого для реализации на рынке

Созданный дипломный проект представляет собой кроссплатформенное приложение для ОС iOS, iPadOS и macOS, которое позволяет студентам и сотрудникам БГУИР просматривать расписание занятий и экзаменов для групп, преподавателей и кабинетов, а также общее расписание для нескольких сущностей; просматривать подробную информацию о группах, преподавателях, кабинетах, специальностях; добавлять собственные занятия; добавлять и отслеживать задания.

Целью разработки данного проекта является упрощение получения и обработки данных о занятиях и другой информации. Приложение предназначено для использования в учебных целях и помогает студентам и сотрудникам эффективнее планировать своё время, управлять заданиями и получать своевременную информацию об изменениях в учебном расписании.

Целевой аудиторией данного приложения являются студенты, и сотрудники БГУИР, которым необходим удобный инструмент для получения информации из API ИИС. Также потенциальными покупателями могут быть иные учебные заведения, которые могут быть заинтересованы в использовании данного приложения.

На iOS в момент разработки существует два приложения со схожим функционалом: BSUIR Schedule, BSUIR Timetable. Существующие решения используют данные ИИС БГУИР не в полном объёме. Функционал этих решений ограничивается отображением расписания преподавателей и групп, однако API предоставляет больше информации: кабинеты и их типы, название специальностей, факультетов и кафедры, отношение преподавателей к кафедрам и другое. На iPadOS существует только BSUIR Schedule, а на macOS отсутствуют какие-либо приложения с таким функционалом.

Планируется распространение приложения через Apple App Store с использованием модели монетизации с бесплатной базовой версией и платной расширенной версией, которая имеет больший функционал.

## Расчёт инвестиций в разработку программного средства

### Расчёт зарплат на основную заработную плату разработчиков

Расчёт затрат на основную заработную плату разработчиков производится исходя из количества людей, которые занимаются разработкой программного продукта, месячной зарплаты каждого участника процесса разработки и сложности выполняемой ими работы. Затраты на основную заработную плату рассчитаны по формуле:

|  |  |  |
| --- | --- | --- |
|  |  | (7.1) |

где коэффициент премий и иных стимулирующих выплат;

категории исполнителей, занятых разработкой программного средства;

часовая заработная плата исполнителя *i*-й категории, р;

трудоемкость работ, выполняемых исполнителем *i*-йкатегории, ч.

Разработкой всего приложения занимается инженер-программист, Обязанности тестирования приложения лежат на инженере-тестировщике. Задачами инженера-программиста, который занимается являются создание модели данных, графического интерфейса, связи между моделью данных и графическим интерфейсом. Инженер-тестировщик занимается выявлением неработоспособных частей приложения, а также оценивает пользовательский опыт, получаемый от использования приложения.

Месячная заработная плата основана на медианных показателях для Junior инженера-программиста за 2023 год по Республике Беларусь, которая составляет 910 Долларов США в месяц, а для Junior инженера-тестировщика – 600 Долларов США [[[6]](#endnote-7)].

По состоянию на 15 апреля 2023 года, 1 Доллар США по курсу Национального Банка Республики Беларусь составляет 2,9441 Белорусских рублей [[[7]](#endnote-8)]. В перерасчёте на Белорусские рубли месячные оклады для инженера-программиста и инженера-тестировщика соответственно составляют составляет 2 679,13 и 1 760,46 Белорусских рублей соответственно.

Часовой оклад исполнителей высчитывается путём деления их месячного оклада на количество рабочих часов в месяце, то есть 160 часов.

За количество рабочих часов в месяце для инженера-программиста и инженера-тестировщика принято соответственно 196 и 32 часа.

Коэффициент премии приравнивается к единице, так как она входит сумму заработной платы. Затраты на основную заработную плату приведены в таблице ниже.

Таблица 7.1 – Затраты на основную заработную плату

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Категория исполнителя | Месячный оклад, р | Часовой оклад, р | Трудоёмкость работ, ч | Итого, р |
| Инженер-программист | 2 679,13 | 16,74 | 196 | 3 281,04 |
| Инженер-тестировщик | 1 760,46 | 11 | 32 | 352 |
| Итого | | | | 3 633,04 |
| Премия и иные стимулирующие выплаты (0%) | | | | 0 |
| Всего затраты на основную заработную плату разработчиков | | | | 3 633,04 |

### Расчёт затрат на дополнительную заработную плату разработчиков

Расчёт затрат на дополнительную заработную плату команды разработчиков рассчитывается по формуле:

|  |  |  |
| --- | --- | --- |
|  |  | (7.2) |

где норматив дополнительной заработной платы.

Значение норматива дополнительной заработной платы принимает за 10 %.

### Расчёт отчислений на социальные нужды

Размер отчислений на социальные нужды определяется согласно ставке отчислений, которая на апрель 2023 г. равняется 35%: 29% отчисляется на пенсионное страхование, 6% − на социальное страхование. Расчёт отчислений на социальные нужды вычисляется по формуле:

|  |  |  |
| --- | --- | --- |
|  |  | (7.3) |

где норматив отчислений в ФСЗН.

### Расчёт прочих расходов

Расчёт затрат на прочие расходы определяется при помощи норматива прочих расчётов. Эта величина имеет значение 30 %. Расчёт прочих расходов вычисляется по формуле:

|  |  |  |
| --- | --- | --- |
|  |  | (7.4) |

где норматив прочих расходов.

### Расчёт расходов на реализацию

Для того, чтобы рассчитать расходы на реализацию, необходимо знать норматив расходов на неё. Принимаем значение норматива равным 3%. Формула, которая использована для расчёта расходов на реализацию:

|  |  |  |
| --- | --- | --- |
|  |  | (7.5) |

где Нр – норматив расходов на реализацию.

### Расчёт общей суммы затрат на разработку и реализацию

Определяем общую сумму затрат на разработку и реализацию как сумму ранее вычисленных расходов: на основную заработную плату разработчиков, дополнительную заработную плату разработчиков, отчислений на социальные нужды, расходы на реализацию и прочие расходы. Значение определяется по формуле:

|  |  |  |
| --- | --- | --- |
|  |  | (7.6) |

Таким образом, величина затрат на разработку программного средства высчитывается по указанной выше формуле и указана в таблице ниже.

Таблица 7.2 – Затраты на разработку

|  |  |  |
| --- | --- | --- |
| Название статьи затрат | Формула/таблица для расчёта | Значение, р. |
| 1. Основная заработная плата разработчиков | См. таблицу 7.1 | 3 633,04 |
| 2. Дополнительная заработная плата разработчиков |  | 363,3 |
| 3. Отчисление на социальные нужды |  | 1 398,71 |
| 4. Прочие расходы |  | 1 089,91 |
| 5. Расходы на реализацию |  | 108,99 |
| 6. Общая сумма затрат на разработку и реализацию |  | 6 593,95 |

## Расчёт экономического эффекта от реализации программного средства на рынке

Для расчёта экономического эффекта организации-разработчика программного средства, а именно чистой прибыли, необходимо знать такие параметры как объем продаж, цену реализации и затраты на разработку.

Соответственно необходимо создать обоснование возможного объёма продаж, количества проданных лицензий расширенной версии программного средства, купленного пользователями. В БГУИР обучается примерно 16 000 студентов, а работают в университете более 2 200 человек, что в сумме составляет примерно 18 000 человек [[[8]](#endnote-9)]. Процент пользователей в Республике Беларусь, которые используют iOS среди других мобильных ОС [[[9]](#endnote-10)], iPadOS среди других планшетных ОС [[[10]](#endnote-11)] и macOS среди других настольных ОС [[[11]](#endnote-12)] на март 2023 года соответственно составляют 36,97%, 40,08% и 14,13%.

Учитывая высокую для студентов стоимость устройств компании Apple, отсутствие информации о количестве пользователей, распределённом по возрастным и социальным группам, примем за процент перспективных пользователей приложения 30% от общего числа связанных с БГУИР людей, то есть 5 400 человек. Допустим, что из них 4 000 человек установят приложение, а из них 2 000 человек приобретут расширенную версию.

Стоимость в App Store задаётся по конкретным заранее определённым компанией уровням, то есть установить любую стоимость нельзя. Расчёты в белорусском регионе App Store ведутся в Долларах США. Так как функции расширенной версии не представлены ни в одном другом приложении и не имеют аналогов, наиболее оптимальной ценой данной версии предполагается 7,99 Долларов США, что с вычетом комиссии Apple в 30% составляет 5,6 Доллара США. Таким образом, отпускная цена копии программного средства составляет 16,48 Белорусских рубля.

Для расчёта прироста чистой прибыли необходимо учесть налог на добавленную стоимость, который высчитывается по следующей формуле:

|  |  |  |
| --- | --- | --- |
|  |  | (7.7) |

где N – количество копий(лицензий) программного продукта, реализуемое за год, шт.;

– отпускная цена копии программного средства, р. ;

– количество приобретённых лицензий;

– ставка налога на добавленную стоимость, %.

Ставка налога на добавленную стоимость по состоянию на 15 апреля 2023 года, в соответствии с действующим законодательством Республики Беларусь, составляет 20%. Используя данное значение, посчитаем НДС:

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

Посчитав налог на добавленную стоимость, можно рассчитать прирост чистой прибыли, которую получит разработчик от продажи программного продукта. Для этого используется формула:

|  |  |  |
| --- | --- | --- |
|  |  | (7.8) |

где *N* – количество копий(лицензий) программного продукта, реализуемое за год, шт.;

Цотп – отпускная цена копии программного средства, р.;

– сумма налога на добавленную стоимость, р.; Нп – ставка налога на прибыль, %;

– рентабельность продаж копий;

– рентабельность продаж копий.

Ставка налога на прибыль, согласно действующему законодательству, по состоянию на 14.04.2023 равна 20%. Рентабельность продаж копий взята в размере 40%. Зная ставку налога и рентабельность продаж копий (лицензий), рассчитывается прирост чистой прибыли для разработчика:

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

## Расчёт показателей экономической эффективности разработки и реализации программного средства на рынке

Для того, чтобы оценить экономическую эффективность разработки и реализации программного средства на рынке, необходимо рассмотреть результат сравнения затрат на разработку данного программного продукта, а также полученный прирост чистой прибыли за год.

Сумма затрат на разработку меньше суммы годового экономического эффекта, поэтому можно сделать вывод, что инвестиции окупятся менее, чем за один год. Таким образом, оценка экономической эффективности инвестиций производится при помощи расчёта рентабельности инвестиций (Return on Investment, ROI). Формула для расчёта ROI:

|  |  |  |
| --- | --- | --- |
|  |  | (7.9) |

где – прирост чистой прибыли, полученной от реализации программного средства на рынке информационных технологий, р.;

Зр – затраты на разработку и реализацию программного средства, р.

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

## Вывод об экономической целесообразности реализации проектного решения

Проведённые расчёты технико-экономического обоснования позволяют сделать предварительный вывод о целесообразности разработки данного программного продукта. Общая сумма затрат на разработку и реализацию составила 6 593,95 Белорусских рублей, а отпускная цена была установлена на уровне 16,48 Белорусских рублей.

Прирост чистой прибыли за год, исходя из предполагаемого объёма продаж в размере 2000 расширенных версий в год, составляет 8 789,33 Белорусских рублей. Рентабельность инвестиций за год составляет 33,29%.

Это означает, что разработка данного программного продукта является целесообразной и реализация программного средства по установленной цене имеет смысл.

Однако, следует учитывать возможные риски, связанные с конкуренцией со стороны аналогов, что может привести к незамеченности продукта на рынке. Кроме того, высокая рентабельность связана с рисками, и расчётные результаты были получены при предполагаемом объёме продаж в 2000 копий в год.

Тем не менее, при поддержке проект может получить долгосрочное и успешное развитие, и количество проданных копий может превысить предполагаемое количество.

Масштабирование проекта на другие учебные заведения может способствовать его успешному развитию. В целом, инвестирование в предложенный проект также оправдано.

Заключение

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[Введите текст] [Электронный ресурс]. – [Введите текст] – Режим доступа: [Введите текст]– Дата доступа: 17.04.2023

[Введите текст] [Электронный ресурс]. – [Введите текст] – Режим доступа: [Введите текст]– Дата доступа: 17.04.2023

[Введите текст] [Электронный ресурс]. – [Введите текст] – Режим доступа: [Введите текст]– Дата доступа: 17.04.2023

ПРИЛОЖЕНИЕ А

(обязательное)

**Спецификация**

ПРИЛОЖЕНИЕ Б

(обязательное)

**Ведомость документов**

1. [] Apple Developer [Электронный ресурс]. – Model-View-Controller – Режим доступа: https://developer.apple.com/library/archive/documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html – Дата доступа: 17.04.2023 [↑](#endnote-ref-2)
2. [] Apple Developer [Электронный ресурс]. – UIKit – Режим доступа: <https://developer.apple.com/documentation/uikit> – Дата доступа: 17.04.2023 [↑](#endnote-ref-3)
3. [] Apple Developer [Электронный ресурс]. – AppKit – Режим доступа: <https://developer.apple.com/documentation/appkit> – Дата доступа: 17.04.2023 [↑](#endnote-ref-4)
4. [] Apple Developer [Электронный ресурс]. – SwiftUI – Режим доступа: <https://developer.apple.com/xcode/swiftui/> – Дата доступа: 17.04.2023 [↑](#endnote-ref-5)
5. [] Apple Developer [Электронный ресурс]. – Interfacing with UIKit – Режим доступа: <https://developer.apple.com/tutorials/swiftui/interfacing-with-uikit> – Дата доступа: 17.04.2023 [↑](#endnote-ref-6)
6. [] Интернет-издание «Dev.by» [Электронный ресурс]. – Зарплата в ИТ – Режим доступа: <https://salaries.devby.io> – Дата доступа: 15.04.2023 [↑](#endnote-ref-7)
7. [] Национальный банк Республики Беларусь [Электронный ресурс]. – Официальные курсы белорусского рубля по отношению к иностранным валютам, устанавливаемые Национальным банком Республики Беларусь ежедневно, на 15.04.2023 – Режим доступа: <https://www.nbrb.by/statistics/rates/ratesdaily.asp> – Дата доступа: 15.04.2023 [↑](#endnote-ref-8)
8. [] Белорусский государственный университет информатики и радиоэлектроники [Электронный ресурс]. – БГУИР сегодня – Режим доступа: <https://www.bsuir.by/ru/bguir-segodnya> – Дата доступа: 15.04.2023 [↑](#endnote-ref-9)
9. [] Statcounter GlobalStats [Электронный ресурс]. – Mobile Operating System Market Share Belarus – Режим доступа: <https://gs.statcounter.com/os-market-share/mobile/belarus> – Дата доступа: 15.04.2023 [↑](#endnote-ref-10)
10. [] Statcounter GlobalStats [Электронный ресурс] Tablet Operating System Market Share Belarus – Режим доступа: <https://gs.statcounter.com/os-market-share/tablet/belarus> – Дата доступа: 15.04.2023 [↑](#endnote-ref-11)
11. [] Statcounter GlobalStats [Электронный ресурс]. – Desktop Operating System Market Share Belarus – Режим доступа: <https://gs.statcounter.com/os-market-share/desktop/belarus> – Дата доступа: 15.04.2023 [↑](#endnote-ref-12)