

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc168284325)

[АНАЛИТИЧЕСКАЯ ЧАСТЬ 6](#_Toc168284326)

[1 АНАЛИЗ БИЗНЕС-ПРОЦЕССА Мобильного приложения «личный кабинет студента» 6](#_Toc168284327)

[1.1 Описание бизнес-процессов мобильного приложения «Личный кабинет студента» 6](#_Toc168284328)

[1.1.1 Функциональные возможности приложения 7](#_Toc168284329)

[1.2 Анализ существующего сайта Московского Университета имени С.Ю. Витте с доступом в личный кабинет студента 7](#_Toc168284330)

[1.3 Анализ существующих аналогов мобильных приложений 9](#_Toc168284331)

[1.4 Моделирование деятельности разработки мобильного приложения «Личный кабинет студента» 11](#_Toc168284332)

[1.4.1 Текущая деятельность программы 11](#_Toc168284333)

[1.4.2 Диаграмма процессов взаимодействия с системой 12](#_Toc168284334)

[1.5 Анализ требований пользователей к разрабатываемому мобильному приложению 13](#_Toc168284335)

[1.6 Выводы по разделу 14](#_Toc168284336)

[ПРОЕКТНАЯ ЧАСТЬ 15](#_Toc168284337)

[2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПРОЕКТА 15](#_Toc168284338)

[2.1 Архитектура мобильного приложения «Личный кабинет студента» 15](#_Toc168284339)

[2.1.1 Технологии и инструменты 15](#_Toc168284340)

[2.1.2 Описание модулей 15](#_Toc168284341)

[2.2 Моделирование и разработка информационной системы 16](#_Toc168284342)

[2.2.1 Дерево функций. 16](#_Toc168284343)

[2.2.2 Разработка дизайна 17](#_Toc168284344)

[2.2.3 Реализация проекта 18](#_Toc168284345)

[2.3 Техническое задание на разработку мобильного приложение «Личный кабинет студента» 27](#_Toc168284346)

[2.3.1 Общие сведения о проекте 27](#_Toc168284347)

[2.3.2 Цели и назначение создания автоматизированной системы 27](#_Toc168284348)

[2.3.3 Характеристика объектов автоматизации 28](#_Toc168284349)

[2.3.4 Требования к программе или программному изделию. 28](#_Toc168284350)

[2.3.5 Состав и содержание работ по созданию автоматизированной системы 29](#_Toc168284351)

[2.3.6 Порядок разработки автоматизированной системы 29](#_Toc168284352)

[2.3.7 Тестирование автоматизированной системы 29](#_Toc168284353)

[2.4 Руководство пользователя мобильного приложения «Личный кабинет студента» 30](#_Toc168284354)

[2.5 Выводы по главе 2 32](#_Toc168284355)

[ЭКОНОМИЧЕСКАЯ ЧАСТЬ 33](#_Toc168284356)

[3 ОБОСНОВАНИЕ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ОТ РАЗРАБОТКИ ИНФОРМАЦИОННОЙ СИСТЕМЫ 33](#_Toc168284357)

[3.1 Расчет затрат на разработку информационной системы 33](#_Toc168284358)

[3.2 Выводы по главе 3 34](#_Toc168284359)

[ЗАКЛЮЧЕНИЕ 36](#_Toc168284360)

[СПИСОК ЛИТЕРАТУРЫ 37](#_Toc168284361)

# ВВЕДЕНИЕ

В современном мире мобильные технологии играют значительную роль в повседневной жизни каждого человека, в том числе и студента университета. Мобильное устройство помогает упростить доступ к информации, улучшить коммуникацию и повысить эффективность обучения. В связи с этим актуальным становится создания мобильного приложения личного пользования студента, которое обеспечит быстрый и удобный доступ к различным сервисам и услугам для студентов.

В данном отчёте мы рассмотрим процесс разработки такого мобильного приложения, его архитектуру и функциональность. Также узнаем преимущества использования данного приложения для студента.

Разработка информационной системы, которая будет инуитивно понятной и доступной для использования широким кругом пользователей, ялвяется актуальной темой исследования.

**Объектом исследования** моей выпускной квалификационной работы является сайт с личным кабинетом Московского Университета имени С. Ю. Витте.

**Предметом исследования** выпускной квалификационной работы является процесс изучения разработки мобильного приложения «Личный кабинет Студента»

**Цель** моей выпускной квалификационной работы является разработка мобильного приложения для студентов ЧОУ ВО «Московский университет им. С. Ю. Витте, которое предоставляет расписание занятий и дополнительные функции с актуальной информацией для пользователя, для улучшения учебного процесса.

Основные задачи необходимые для достижения цели:

1. Провести анализ предметной области для выявления бизнес-процессов взаимодействия пользователя с приложением
2. Сформировать необходимые функциональные требования к будущей ИС
3. Проанализировать базы данных - БД) и средства разработки для будущей ИС
4. Создать модель данных на физическом и логическом уровне.
5. Разработать ИС и создать диаграмму использования готового решения.
6. Оценить целесообразность разработки ИС и ее себестоимость.

Моя бакалаврская работа состоит из:

* Введения
* Первой главы - теоретическая часть)
* Второй главы - практическая часть)
* Третьей главы - экономическая часть)
* Заключения - выводы)
* Списка литературы

Первая глава, посвящена описанию структуры организации и ее деятельности, а также ранее разработанному бизнес-процессу по приему заявок. Выявлены недостатки данного программного обеспечения и сформированы новые требования к разрабатываемой информационной системе.

Вторая глава, посвящена процессу разработки базы данных и самой информационной системы с диаграммой использования.

В третьей главе, описана экономическая часть проекта. Дана оценка себестоимости и целесообразности разработки данного продукта.

# АНАЛИТИЧЕСКАЯ ЧАСТЬ

# АНАЛИЗ БИЗНЕС-ПРОЦЕССА Мобильного приложения «личный кабинет студента»

## Описание бизнес-процессов мобильного приложения «Личный кабинет студента»

Мобильное приложение «Личный кабинет студента» разработано с целью предоставить студентам удобный и быстрый доступ к информации, связанной с учебным процессом.

Основные функции приложения:

* **Вход по личным данным пользователя:** студенты смогут входить по логину и паролю от сайта МУИВ.
* **Просмотр расписания занятий:** пользователи могут оперативно узнавать о времени и месте проведения занятий.
* **Просмотр успеваемости:** студенты могут просматривать свои задолжности и зачетную книжку.
* **Переход на центр поддержки и почту студента:** создан более короткий путь через приложение на сервисы.

Необходимо определиться с методом разработки программы, языком программирования, проанализировать функционал и аналоги программы.

### Функциональные возможности приложения

Рисунок 1.1 – UML диаграмма.

Рассмотрим диаграмму, построенную мной на рисунке 1.1. Здесь описано взаимодействие студента с программой. Необходимо ввести данные для дальнейшего функционала с приложением, далее студент уже сам выбирает какая информация ему необходима для учебного процесса.

## Анализ существующего сайта Московского Университета имени С.Ю. Витте с доступом в личный кабинет студента

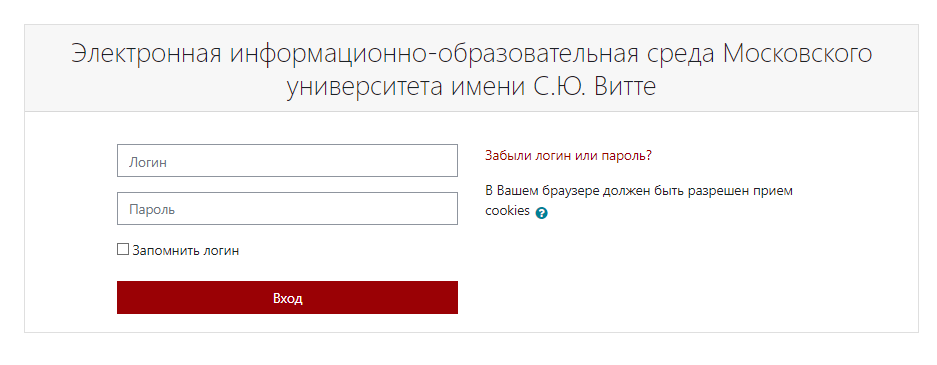
В настоящий момент у ЧОУ ВО «Московский Университет им. С.Ю. Витте» имеет собственный информационный сервис в виде сайта. В нем имеется личный кабинет для студентов, в котором находится вся информация для учащихся.

Рисунок 1.2 – Вход в личный кабинет пользователя МУИВ

На рисунке 1.2 сайт просит ввести данные (логин и пароль). По данному аналогу будет разработан уже в моем приложении вход в систему.

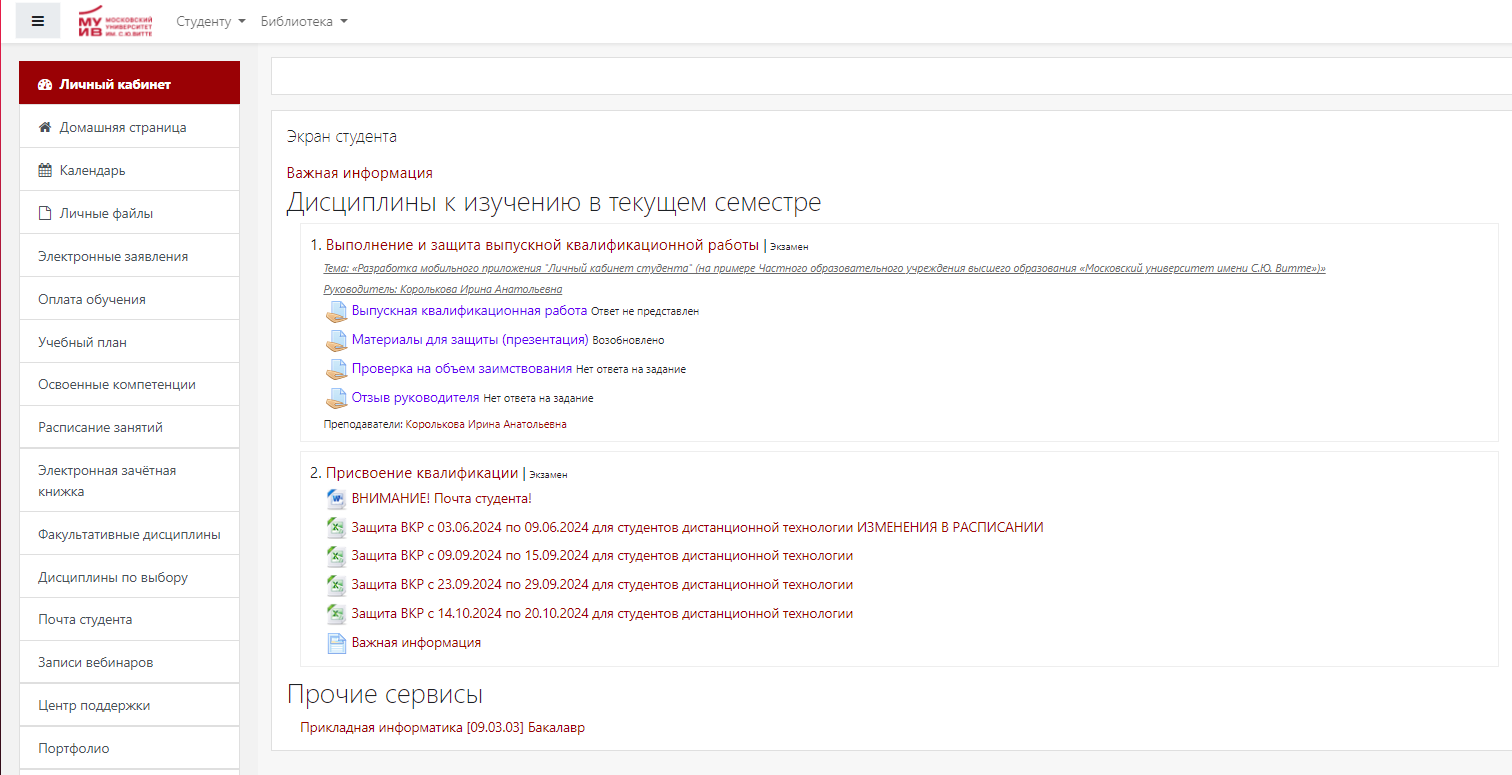
Далее после входа мы попадаем в личный кабинет пользователя.

Рисунок 1.2 – Страница личного кабинета на сайте

На странице личного кабинета находится информация с дисциплин. В левой части странице находится список функций, необходимых для учебного процесса студента.

Таким образом, могу сказать, что мы обладаем достаточной информационной базой для разработки моей программы с возможностью импортировать её в индивидуальное мобильное устройство каждого студента университета.

## Анализ существующих аналогов мобильных приложений

Во многих образовательных учереждений уже имеется свои мобильные приложения для общего пользования студентов, одну из которых предимонстрирую.

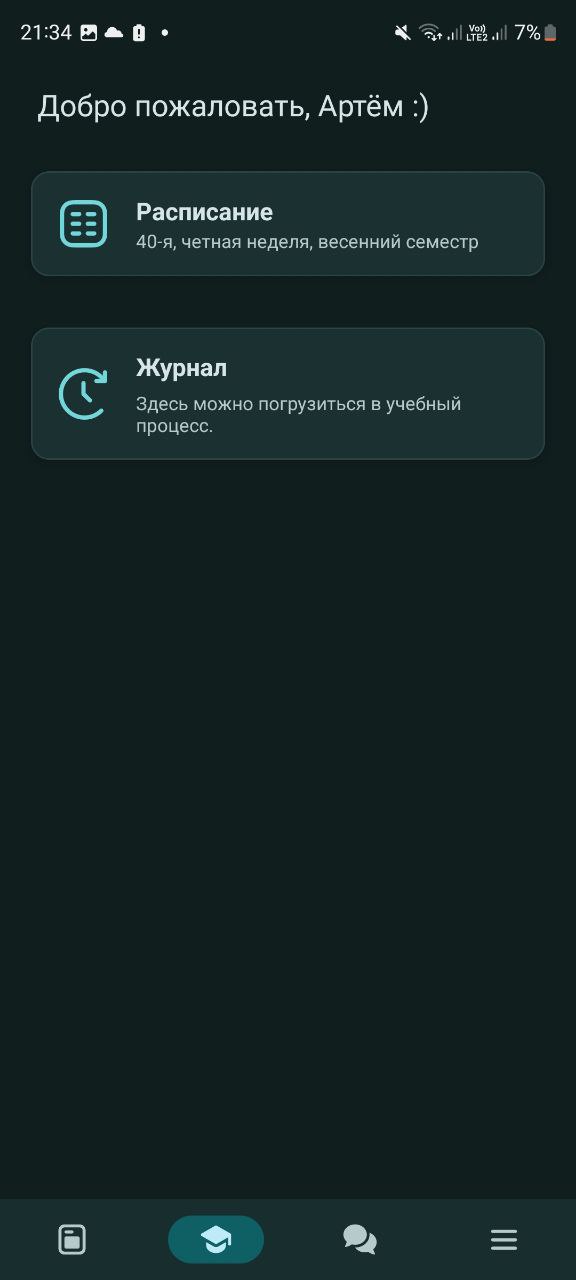


Рисунок 1.3 – мобильное приложение «Мой МГОТУ»

В данном приложении имеется несколько функций, такие как: расписание занятий, оценки (журнал), новости и чат, как с студентами, так и с преподавателями.

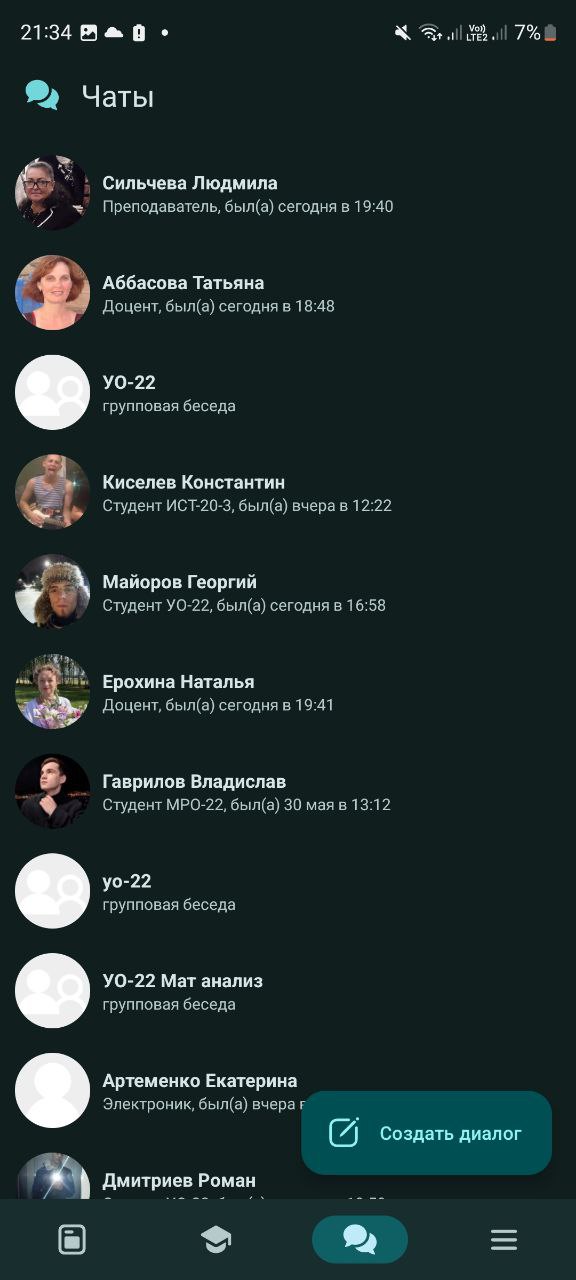
 В данном приложении имеется раздел с чатом между студентов и преподователей, что добовляет хороший функционал связи.

Рисунок 1.4 – Экран чатов

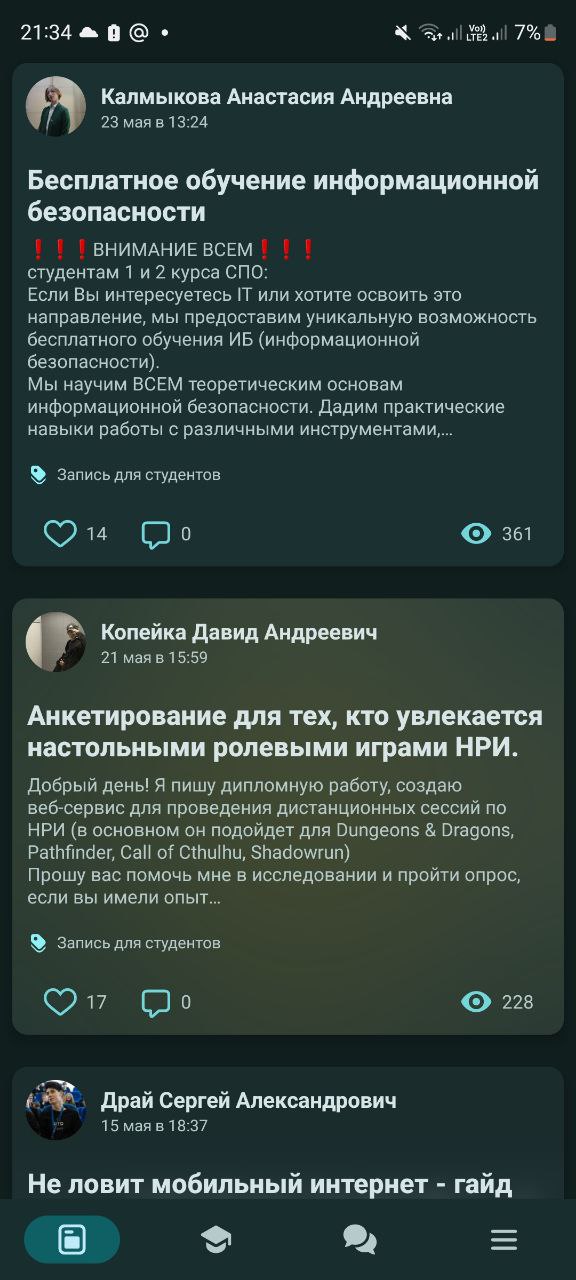


Рисунок 1.5 – Раздел новостей

Из интересного функционала можно выделить новостную ленту в самом приложении, в ней отображается автор, наименование, сама статья, а также возможность оставлять «лайки» и комментировать посты.

В программе можно подчеркнуть несколько минусов данного приложения:

* Мало функционала: в данном приложении не имеется больше информации для самого студента.
* Темный интерфейс: приложение отображает только темную тему, что для некоторых пользователей может показатся совершенно не удобным решением для вида интерфейса.

Таким образом мы видим, что во многих образовательных организаций имеется индивидуальные мобильные программы для удобства пользования студентам. По аналогам других приложений, будет структурировано моя программа для Московского Университета им. С.Ю. Витте, а также будут учтены все недочеты других программ.

## Моделирование деятельности разработки мобильного приложения «Личный кабинет студента»

В данном разделе будет описано текущее состояние бизнес-процессов в системе, а именно:

1. Описание текущей деятельности программы
2. Диаграммы процессов
3. Анализ текущих процессов

### Текущая деятельность программы

В настоящее время, приложение «Личный кабинет студента» используется для следующих задач:

* Просмотр расписания занятий
* Получение информации по успеваемости
* Переход в центр поддержки и на почту студента
* Формирования отчетов

Данные функции помогут эффективнее получать актуальную информацию об учебных процессах.

### Диаграмма процессов взаимодействия с системой

Для создания программы, была простроенна диаграмма IDEF0. Данная диаграмма – проектирует методологию для построения и интерпритации моделей.

Рисунок 1.3 – Диаграмма IDEF0

А-0: Использование мобильного приложения «Личный кабинет студента»

* **Входы:** Входные данные пользователя, расписание занятий, дисциплины, оценки, отчеты;
* **Механизмы:** Мобильное устройство, ПО, интернет;
* **Управление:** Стандарты качества, разрешения пользования, права доступа;
* **Выходы:** Информация для пользователя, мобильный доступ.

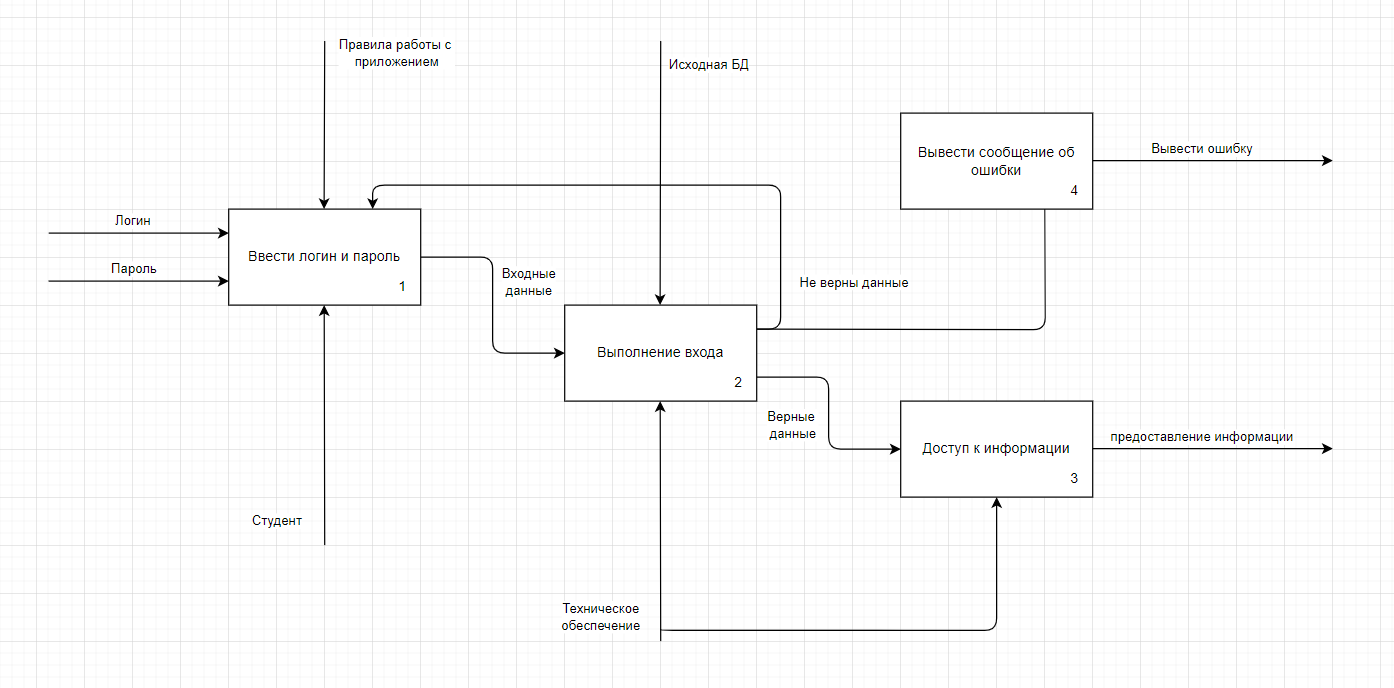


Рисунок 1.4. – Диаграмма А1

На рисунке 1.4. мы видим диаграмму А1, представляет собой выполнение логики доступа к информации.

1. Мы вводим логин и пароль для входа
2. Программа выполняет вход через портал сайта «МУИВ»
3. При верных данных, программа предоставляет доступ к информации
4. При не верных данных, приложение должно выводить ошибку и просить повторить попытку.

## Анализ требований пользователей к разрабатываемому мобильному приложению

Основное требование к моей задачи относится более быстрая и упрощенная возможность получить информацию студенту о занятиях и успеваемости.

Также из требований по взаимодействию пользователя с программой можно подчеркнуть следующие пункты:

1. Приложение должно корректно работать на различных форматах экрана.
2. Приложение должно иметь интуитивно понятный и удобный интерфейс.
3. Приложение долэно быстро загружатся и обеспечивать высокую скорость работы.
4. Приложение должно занимать минимальное количество памяти и заряда батареи.
5. Приложение должно поддерживать авторизацию через учетную запись МУИВ.
6. Пользователи должны иметь возможность просматривать расписание своих занятий в удобной и наглядной форме.
7. Пользователи должны иметь возможность скачать формулирующие отчеты.
8. Пользователи должны иметь возможность сохранять данные для входа.

Данные перечисленные требования обязаны быть выполнены при разработки мобильного приложения.

## Выводы по разделу

В данной главе, мы провели аналитическую часть по разработке мобильного приложения, проанализировали существующий сайт Московского Университета им. С. Ю. Витте, а также аналоги программ, решили о необходимости разработки и внедрения проекта, провели моделирование автоматизированных бизнес-процессов, и разобрали требования пользователей к программе. После проделанной работы можно сделать выводы, что мы имеем мысль о том, что создание мобильного приложения для студентов очень необходимо.

# ПРОЕКТНАЯ ЧАСТЬ

# ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПРОЕКТА

## Архитектура мобильного приложения «Личный кабинет студента»

Архитектура приожения построен по принципу клиент-сервер. Мобильное приложение взаимодействует с сайтом МУИВ, для получения данных через HTTP-запросы. Основные компоненты архитектуры включают пользовательский интерфейс (UI), обработку данных, сеть и кэширование.

### Технологии и инструменты

1. Язык программирования: Kotlin
2. Среда разработки: Android Studio
3. Библиотека для парсинга: Jsoup для парсинга HTML-страниц

### Описание модулей

1. Пользовательский интерфейс: взаимодействие пользователя с экранами профиля, расисания, дисциплин, оценок и отчетов.
2. Обработка данных: Логика приложения для обработки и преобразования данных, полученных от сервера.
3. Сеть: Модуль для выполнения HTTP-запросов с сайтом.
4. Кэширование: Хранения данных на устройстве.

## Моделирование и разработка информационной системы

### C:\Users\Sainrs\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Снимок экрана 2024-06-02 053009.pngДерево функций.

Рисунок 2.1 – Древо функций

Для разработки программы, необходимо определить функционал приложения, для этого было построено древо функций. В нем описано основные функции, такие как: просмотр расписания, просмотр дисциплин, просмотр оценок, формирование отчетов. И также служебные: сохранение логина и пароля, скачивание отчетов.

### Разработка дизайна

Для разработки приложения, важным является создание внешнего вида, как пользователи будут видеть интерфейс.

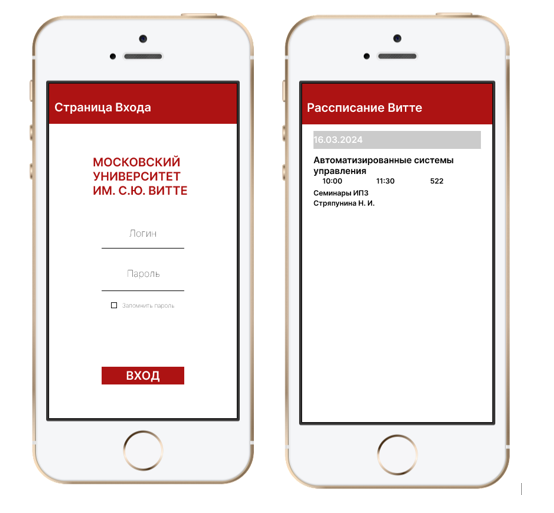
Для своей работы я использовал программу Figma, чтобы выявить примерный внешний вид моей разработки.

Рисунок 2.2 - Макет предварительного дизайна приложения.

Главное в интерфейсе – это простота испльзования. Пользователь на начальном экране вводит логин и пароль, позже может нажать на кнопку «запомнить пароль», чтобы при следующем запуске, не нужно тратить время на ввод данных. Данный дизайн был взят по аналогам представленные в первой части.

Далее пользователь попадает на экран с собственным расписанием, где представлена дата, время, наименование предмета, вид занятия и фамилия преподавателя.

Благодаря конструктора, встроенного в программу android studio, удалость построить удобный интерфейс на основе макета. Во время разработки приложения было решено добавить недавно обновленный логотип МУИВ.

### Реализация проекта

#### **Вход**

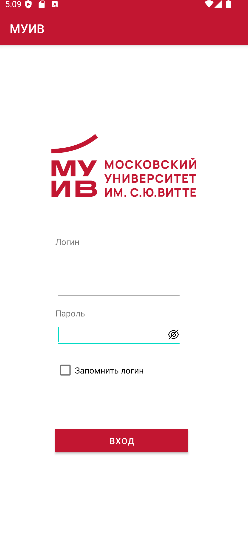
Первым пунктом я опишу разработку начального экрана с входом на страницу приложения

Рисунок 2.3 – Интерфейс входа

Приложение написано на Kotlin и представляет собой типичное Android-приложение с функционалом авторизации. Главная активность (MainActivity) отвечает за управление интерфейсом пользователя и обработку логики авторизации. Ниже представлено подробное описание основных компонентов и функциональности MainActivity.

Для работы данного сервиса, необходимо прописать код входа в систему через портал МУИВ. В моем коде он происходит методе «perforLogin»:

private fun performLogin(username: String, password: String, remember: Boolean) {  
 *CoroutineScope*(Dispatchers.IO).*launch* **{** try {  
 val sessionManager = SessionManager(this@MainActivity, username, password)  
 sessionManager.createSession()  
  
 saveLoginDetails(username, password, remember)  
  
 navigateToSecondActivity()  
 } catch (e: Exception) {  
 withContext(Dispatchers.Main) **{** Toast.makeText(this@MainActivity, "Ошибка входа: ${e.message}", Toast.*LENGTH\_LONG*).show()  
 **}** }  
 **}**}

Данный метод выпоняет следующие фукнции:

* Получает имя пользователя и пароль.
* Создает объект “SessionManger” и вызывает метод “createSession” для аутентификации пользователя.
* В случае успешной аутентификации сохраняет детали входа и переходит к следующей активности “SecondActivity”.
* В случае ошибки входа отображает сообщение об ошибке пользователю.

Далее формируется класс “SessionManager”, он содержит логику для аутентификации пользователя и создания сессии.

class SessionManager(  
 private val context: Context,  
 private val username: String,  
 private val password: String  
) {  
 private val loginURL = *Url*("https://e.muiv.ru/login/index.php")  
 private var client: HttpClient  
  
 init {  
 client = createHttpClient()  
 }  
  
 suspend fun createSession(): String {  
 val initialResponse: HttpResponse = client.request(loginURL) **{** method = HttpMethod.Get  
 **}**

Данный класс отвечаез за выполнение HTTP-запросов для аутентификации пользователя на сайте e.muiv.ru и получения сессионных данных.

После написания программного кода, приложение проводит вход через портал сайта, тем самым позволяет попасть в личный кабинет через мобильное устройство.

#### **Расписание занятий**

Вторичная активность приложения (SecondActivity) отвечает за отображение и управление различными аспектами учебного процесса пользователя. Она включает в себя навигацию через боковое меню и отображение данных, таких как расписание, текущие дисциплины, задолженности и оценки.

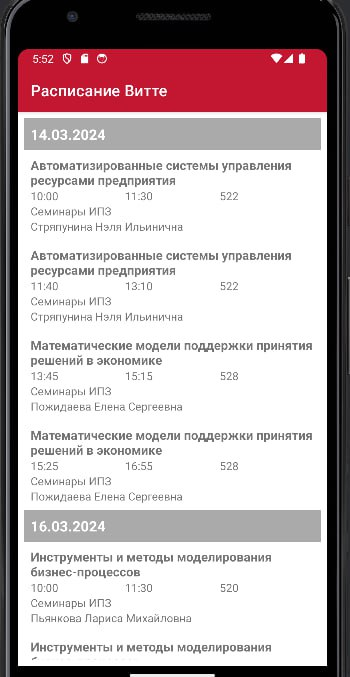


Рисунок 2.4 – Расписание занятий

Данное расписание как было описано ранее парсируется с сайта университета под структуру, которая написанно в данном коде:

private fun getJsonFromLesson(scheduleDiscipline: ScheduleDiscipline): JSONObject {  
 return JSONObject().*apply* **{** put("date", scheduleDiscipline.date.toString())  
 put("startTime", scheduleDiscipline.startTime)  
 put("endTime", scheduleDiscipline.endTime)  
 put("name", scheduleDiscipline.name)  
 put("room", scheduleDiscipline.room)  
 put("type", scheduleDiscipline.type)  
 put("teacher", scheduleDiscipline.teacher)  
 **}** }  
 }  
}

Структура расписания: Дата, наименование предмета, время начала пары, время окончания пары, формат её (семинар или лекция), и ФИО преподавателя.

Получение и отображение данных

Методы для получения и отображения расписания, текущих дисциплин, задолженностей и оценок выполняются асинхронно с использованием корутин. Примеры таких методов:

Получение и отображение расписания

private fun fetchAndDisplaySchedule() {  
  
 binding.webView.*visibility* =View.*GONE* binding.textMessage.*visibility* = View.*GONE* binding.scheduleRecyclerView.*visibility* = View.*GONE* binding.currentDisciplinesRecyclerView.*visibility* = View.*GONE* binding.debtDisciplinesRecyclerView.*visibility* = View.*GONE* binding.progressCircular.*visibility* = View.*VISIBLE  
 CoroutineScope*(Dispatchers.IO).*launch* **{** try {  
 val scheduleHtml =  
 sessionManager.getPageContent(*Url*("https://e.muiv.ru/local/student\_timetable/view.php"))  
 val scheduleDisciplines = scheduleDisciplineWorker.parseSchedule(scheduleHtml)  
 scheduleDisciplineWorker.saveScheduleToFile(  
 scheduleDisciplines,  
 "schedule\_disciplines.txt"  
 )  
  
 runOnUiThread **{** binding.scheduleRecyclerView.*visibility* = View.*VISIBLE* binding.currentDisciplinesRecyclerView.*visibility* = View.*GONE* binding.debtDisciplinesRecyclerView.*visibility* = View.*GONE* binding.progressCircular.*visibility* = View.*GONE* displaySchedule(scheduleDisciplines)  
 **}** } catch (e: Exception) {  
 runOnUiThread **{** Toast.makeText(  
 this@SecondActivity,  
 "Ошибка загрузки расписания: ${e.message}",  
 Toast.*LENGTH\_LONG* ).show()  
 **}** }  
 **}**}

#### **Дисциплины**

Дисциплины имеют схожий формат структуры с расписанием занятий.

#### **Задолженности**

При наличии академической задолженности в отдельной вкладке у студента будет отображатся наименование предмета и у какого преподователя имеется задолженность. Если у студента нет долгов, то будет отображено на экране «у вас нет задолженности».

#### **Оценки**

Рисунок 2.5 – Оценки студента

Оценки выводятся из отдельного отчета с сайта университета. В нем предоставляется список всех пройденых предметов и полученых баллов.

#### **Отчеты**

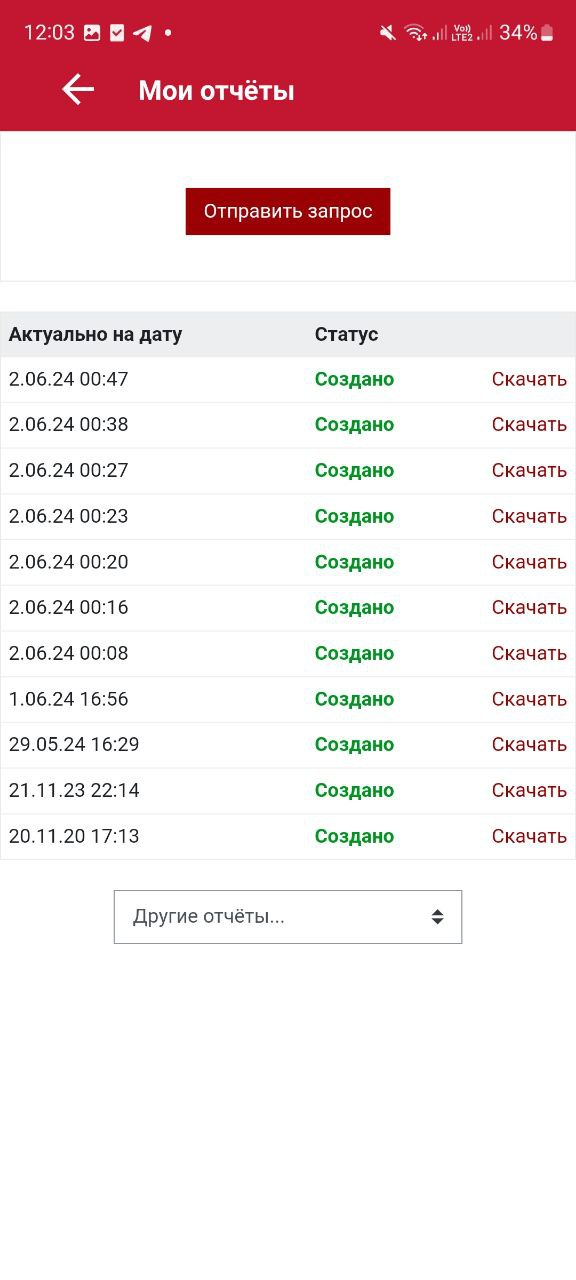
Формирование отчетов является очень важной функцией для студента. Здесь формируются такие отчеты, как: электронная отчетная книжка, запрос квитанции, финансовый отчет и освоенные компетенции.

Рисунок 2.6 – Формирование отчетов

Третья активность приложения (ThirdActivity) предназначена для отображения и загрузки отчетов студентов. Она включает в себя функционал отображения отчетов в WebView и загрузки файлов с использованием кастомного загрузчика:

val webView: WebView = binding.webView  
webView.*settings*.*javaScriptEnabled* = true  
webView.*webViewClient* = object : WebViewClient() {  
  
 override fun shouldOverrideUrlLoading(view: WebView?, request: WebResourceRequest?): Boolean {  
 Log.d("check\_url", "shouldOverrideUrlLoading: ${request?.*url*.*toString*()}")  
 val url = request?.*url*.*toString*()  
 if(url.*contains*("https://e.muiv.ru/local/student\_reports/getreport.php?id=")){  
 webView.loadUrl(url)  
 getStudentReport(url)  
 }  
  
 binding.progressBar.*visibility* = View.*VISIBLE* webView.*visibility* = View.*GONE* return super.shouldOverrideUrlLoading(view, request)  
 }  
  
 override fun onPageFinished(view: WebView?, url: String?) {  
 super.onPageFinished(view, url)  
 view?.loadUrl(  
 "javascript:(function() { " + "var targetDiv = document.getElementById('student\_report\_container');" + "if (targetDiv) {" + "document.body.innerHTML = '';" + "document.body.appendChild(targetDiv);" + "}" + "})()"  
 )  
  
 view?.postDelayed(**{** binding.progressBar.*visibility* = View.*GONE* webView.*visibility* = View.*VISIBLE* **}**, 1000)  
 }  
}

Здесь происходит настройка “WebView” для загрузки страниц и обработки переходов по URL. Если URL содержит определенный паттен, вызывается кнопка «запрос отчета».

Метод для скачивания отчета:

private fun getStudentReport(url: String){  
 val downloader = AndroidDownloader(context = this)  
 downloader.downloadFile(url, sessionManager.cookies)  
 Toast.makeText(*baseContext*, "Загрузка началась", Toast.*LENGTH\_SHORT*).show()  
 this@ThirdActivity.finish()  
  
}

Метод getStudentReport создает экземпляр для скачиванияи вызывает его метод downloadFile для скачивания файла по указанному URL с использованием кук сессии.

Основные компоненты и функциональность

ActivityThirdBinding - биндинг для доступа к элементам интерфейса, определенным в layout-файле.

SessionManager - менеджер сессий для управления состоянием авторизации и получения данных с защищенных ресурсов.

WebView - для отображения веб-страниц с отчетами пользователей и поддержкой JavaScript.

AndroidDownloader - для загрузки отчетов.

CookieManager - для управления куками.

Инициализация и настройка

При создании ThirdActivity происходит инициализация элементов интерфейса, загрузка сессии пользователя и настройка WebView.

#### **Мой профиль**

Рисунок 2.7 – Мой профиль

Данная функция отображает данные о пользователе, его факультет, направление, форма обучения, срок и год поступления.

Отображение профиля пользователя

Метод myProfile настраивает WebView для отображения страницы профиля пользователя.

private fun myProfile(){  
 binding.webView.*visibility* =View.*GONE* binding.textMessage.*visibility* = View.*GONE* binding.scheduleRecyclerView.*visibility* = View.*GONE* binding.currentDisciplinesRecyclerView.*visibility* = View.*GONE* binding.debtDisciplinesRecyclerView.*visibility* = View.*GONE* binding.progressCircular.*visibility* = View.*VISIBLE* val webView: WebView = binding.webView  
 webView.*settings*.*javaScriptEnabled* = true  
 webView.*settings*.*loadsImagesAutomatically* = true  
 webView.*settings*.*domStorageEnabled* = true  
 webView.*settings*.*useWideViewPort* = true  
 webView.*settings*.*loadWithOverviewMode* = true  
 webView.*settings*.*mixedContentMode* = WebSettings.*MIXED\_CONTENT\_ALWAYS\_ALLOW* webView.*webViewClient* = object : WebViewClient() {  
  
 override fun onReceivedSslError(view: WebView?, handler: android.webkit.SslErrorHandler?, error: android.net.http.SslError?) {  
 handler?.proceed() // Ignore SSL certificate errors for now (not recommended for production)  
 }  
 override fun onPageFinished(view: WebView?, url: String?) {  
 super.onPageFinished(view, url)  
 view?.loadUrl("javascript:(function() { " +  
 "var targetDiv1 = document.getElementsByClassName('pffio')[0];" +  
 "var targetDiv2 = document.getElementsByClassName('pffoto')[0];" +  
 "var targetDiv3 = document.getElementsByClassName('pfstudinfo')[0];" +  
 "document.body.innerHTML = '';" +  
 "if (targetDiv1) {" +  
 "document.body.appendChild(targetDiv1);" +  
 "}" +  
 "if (targetDiv2) {" +  
 "document.body.appendChild(targetDiv2);" +  
 "}" +  
 "if (targetDiv3) {" +  
 "document.body.appendChild(targetDiv3);" +  
 "}" +  
 "})()")  
  
 view?.postDelayed(**{** binding.progressCircular.*visibility* = View.*GONE* webView.*visibility* = View.*VISIBLE* **}**, 1000)  
 }  
 }  
 // Enable cookies  
 val cookieManager = CookieManager.getInstance()  
 cookieManager.setAcceptCookie(true)  
  
 // Load URL  
 val url = "https://e.muiv.ru/local/portfolio/view.php"  
 cookieManager.setAcceptCookie(true)  
 cookieManager.setCookie(  
 url,  
 "${sessionManager.cookies.get(0).name}=${sessionManager.cookies.get(0).value}"  
 )  
  
 webView.loadUrl(url)  
}

#### **Почта студента и центр поддержки**

У университета для данных функций имеются отдельные страницы, было решено перейти настроить переадресацию на данные страницы из приложения.

#### **Выход из аккаунта**

Эта функция дает возможноть выйти в начальное меню экрана.

## Техническое задание на разработку мобильного приложение «Личный кабинет студента»

Техническое задание на автоматизированной системы является основным документом, определяющим требования и порядок создания автоматизированной системы, в соответствии с которым проводится разработка АС и ее приемка. Техническое задание оформлено строго по ГОСТ 34.602-2020.

### Общие сведения о проекте

- полное наименование АС и ее условное обозначение: «Мобильное приложение личного кабинета студента»

- наименование организации - заказчика АС, наименование организации-разработчика: ЧОУ ВО Московский Университет имени С. Ю. Витте.

- перечень документов, на основании которых создается АС, кем и когда утверждены эти документы: Утвержденная тема выпускной клафикационной работы

- плановые сроки начала и окончания работ по созданию АС: 14.04.2024 – 31.05.2024

### Цели и назначение создания автоматизированной системы

цели создания АС: Обеспечение удобного доступа к личным данным студента, интеграция с существующей информацией университета.

- назначение АС: Создание мобильного приложения для студентов МУИВ, что способствует удобству и многофункциональности для учащихся.

### Характеристика объектов автоматизации

Мобильное приложение представляет собой в конечном виде необходимый продукт для удобства во время учебного периода жизни студента. В настоящий момент информацию для студента можно смотреть только через браузер, что для многих в наше время является не совсем упрощённым действием.

### Требования к программе или программному изделию.

4.1 Требования к функциональным характеристикам.

Основными требованиями АС являются: структура, функциональность, и общие технические требования.

К требованиям к структуре относится: адаптивный интерфейс, поддержка различных разрешений экранов.

К требованиям к функциям: Синхронизация с сайтом, возможность сохранять данные о входе, простота в использовании.

К общим технических требованиям к мобильному приложению является наличии подключение к интернету, наличие данных для входа в систему, и телефон на базе android.

Требования для запуска приложения:

* Операционная система Android 4.1 или выше.
* Минимум 512 МБ оперативной памяти.
* Не менее 10 МБ свободного места на устройстве.
* Разрешение экрана не менее 320 x 480 пикселей.
* Доступ к интернету для загрузки ресурсов и обновлений.
* Наличие логина и пароля от личного кабиента на портале МУИВ.

### Состав и содержание работ по созданию автоматизированной системы

Для разработки АС был построен план последовательности по созданию мобильного приложения:

* Изучение аналогов мобильных программ с схожими функциями 30.11.2023
* Выбор языка программирования для написания кода 11.12.2023
* Составление диаграмм для решения функцинирования приложения 11.12.2023
* Разработка дизайна интерфейса 12.12.2023
* Написание программного кода на языке Kotlin 13.12.2023-03.01.2024
* Провести тестировку работоспособности программы 05.01.2024
* Провести доработки по программному коду 05.01.2024

### Порядок разработки автоматизированной системы

Этапы по разработке АС:

* Установка и настройка среды разработки Android Studio
* Установка необходимых библиотек
* Создание экранов приложения
* Создание тестового пользователя для проверки работоспособности приложения
* Настройка авторизации из сайта в приложение на страницу расписания

### Тестирование автоматизированной системы

Для тестирования мобильного приложение необходимо установить APK файл на мобильное устройство системы android (либо эмулятор). После чего необходимо ввести данные пользователя и нажать на кнопку «войти», далее мы оказываемся на экране приложения с несколькими вкладками.

## Руководство пользователя мобильного приложения «Личный кабинет студента»

Для дальнейшей эксплуатации информационной системы, необходимо составить сопроводительный документ – “руководство пользователя”.

Сама структура и все необходимые требования к содержанию руководства пользователя по ГОСТ 34, описаны в РД 50-34.698-90.

**Введение**

Приложение «Личный кабинет студента» предназначено для студентов Московского Университета имени С.Ю. Витте. Оно позволяет студентам получить доступ к их учебной информации

**Назначение и условия применения**

Приложение разработано для предоставления студентам удобного способа доступа к учебной информации и коммуникации с университетом. Оно предназначено для использования на устройствах с операционной системой Android и требует подключится к интернету для загрузки и обновления данных.

**Подготовка к работе**

Для работы приложения нужно следующее:

1. Загрузите и установите приложение на устройство Android.
2. Проверьте соединение вашего устройства с сетью интернет.
3. Запустите приложение и войдите в систему, используя свои учетные данные.

**Проверка корректности работы**

1. Запустите приложение и войдите в программу.
2. Проверьте все функции приложения, проверьте даты дисциплин.
3. Сформируйте отчет и проверьте на работоспособность скачивания.
4. Проверьте переадрисации на портал поддержки и почту.

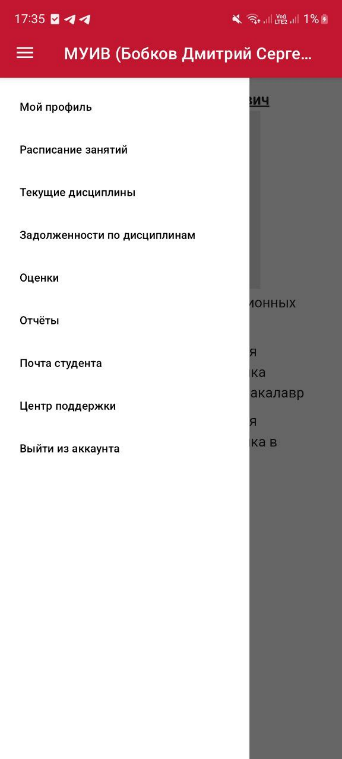
**Выполняемы функции и задачи**

Рисунок 4.1 – Список функций

* Перейдите в раздел «Профиль», чтобы посмотреть ваши личные данные, такие как ФИО, год поступления и факультет.
* Перейдите в раздел «Расписание», чтобы посмотреть актуальное расписание занятий.
* Перейдите в раздел «Дисциплины», чтобы увидеть список всех дисциплин текущего семестра.
* Перейдите в раздел «Задолженности», чтобы просмотреть список задолженностей при их наличии.
* Перейдите в раздел «Оценки», чтобы загрузить и посмотреть ваши оценки за все дисциплины.
* Перейдите в раздел «Отчеты» и выберите нужный отчет (отчет об оплате, зачетная книжка и т.д.)
* Перейдите в раздел «Центр поддержки» или «Почта студента» для доступа к данным разделам.

## Выводы по главе 2

После постройки архитектуры программы, создания его дизайна и проведения тестировки, можно сделать вывод, что приложение, написанное мною, выполняет полный свой функционал, исполняет все требования, и имеет хороший потенциал в дальнейшей модернизации и расширении функционала.

# ЭКОНОМИЧЕСКАЯ ЧАСТЬ

# ОБОСНОВАНИЕ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ОТ РАЗРАБОТКИ ИНФОРМАЦИОННОЙ СИСТЕМЫ

## Расчет затрат на разработку информационной системы

Для того, чтобы расчитать затраты на разработку проекта, мы выделим для начала расчет затрат на разработку программного продукта.

Для разработки мобильного приложения нам понадобится один разработчик android приложений знающий язык kotlin. О имеет среднюю зарплату от 900 рублей в час.

Учет фактических временных затрат представляем в виде таблицы 1.

|  |  |
| --- | --- |
| **Этапы разработки** | **Фактические часы** |
| 1. Подготовка технического задания с заказчиком | 8 |
| 1. Подготовка макета дизайна приложения | 14 |
| 1. Написание программного кода | 32 |
| 1. Тестировка приложения | 5 |
| 1. Внесение вправок | 7 |
| 1. Составление отчета по выполненной работы | 9 |
| **ИТОГ** | **75** |

Итого при расчете заработной платы программиста и проработанных часов можно вычислить: = 900 руб \* 75 ч. = 67 500 руб. будет выплачено за работу специалиста в данной сфере.

## Затраты на оборудование

Для разработки программного кода были задействованы личный персональный компьютер. Для разработки приложения рекомендуется использовать мощный ПК, с такими характеристиками:

* Процессор: Intel Core i7 или i9, частота 3.0 GHz.
* Опертивная память: от 16GB
* Графическая карта: NVIDIA GeForce RTX 3060 или лучше
* Объем HDD: 1TB

Примерная рыночная цена компьютера с данными характеристиками на данный момент составляет от 90 000 руб.

|  |  |
| --- | --- |
| **Параметр** | **Значение** |
| Стоимость ПК | 90 000 |
| Количество дней эксплуатации ПК, дней | 30 |
| Срок службы ПК, лет | 5 |
| Норма амортизации в год, % | 20 |
| Амортизация в период разработки | 1256 |

(затраты на средства вычислительной техники) = 1256 рублей.

## Оценка эффективности внедрения проекта

Данная разработка подразумивается для мобильного удобства взаимодействия студента с периодом учебного процесса, поэтому для начала мы можем рассчитать её эффективность по экономии времени студентов:

В данный момент чтобы студенту попасть в личный кабинет, ему необходимо найти станицу в браузере, далее перейти на вход и ввести данные, что примерно составляет 3-4 минуты. Приложение позволяет сохранять данные входа, также оно находится на самом устройстве, что экономит время студента как минимум до 1-2 минуты.

Экономия на печатных материалах: Переход к электронным отчетам и расписанию снижает потребность в печатных материалах, что приводит к экономии средств на бумаге, печати и хранении. Допустип что университет тратит примерно в год 500 000 рублей в год, переход всех отчетов может сократить данные расходы на 80%, что составит экономию на 400 000 рублей в год.

Улучшение пользовательского опыта: Удобный и интуитивно понятный интерфейс приложения улучшает пользовательский опыт, что может привести к повышению удовлетворенности студентов.

Привлечение новых студентов: Современные и удобные IT-решения могут повысить привлекательность университета для потенциальных студентов, что, в свою очередь, может способствовать к привлечению для потенциальных студентов, что позволяет повысить число поступающих и доход университета.

Модернизация проекта: Само наличие мобильного приложения может повлиять на привлечения новшеств, добавления больше функционала и создания собственного месседжера, где студенты и сотрудники могут публиковать новости, создавать чат и формировать группы.

## Выводы по главе 3

Выполнив главу по экономической части, можно сделать выводы, что разработанное приложение иммеет экономическую выгоду по своей разработки для её реализации. Оно не имеет высоких затрат для организации, и дает возможность экономить пользователям время для получения информации.

Разработка является экономически целесообразной.

# ЗАКЛЮЧЕНИЕ

В данной выпускной квалификационной работе, были выполнены поставленные задачи по проведению анализа предметной области для выявления процессов автоматизации, были сформированы необходимые функциональные требования к разрабатываемой информационной системе, проведен подробный анализ баз данных и средств разработки, построена модель данных на физическом и логическом уровне, разработано мобильное приложение «Личный кабинет студента», дана экономическая оценка целесообразности разработки и ее себестоимости.

В результате выполнения задач, были проанализированы существующие приложения, и сайт по основе которой было сформулировано приложение. Также для работы были созданы диаграммы IDEF0, USE CASE, и древо функций. Был разработан дизайнерский макет, разработан программный код на Android Studio, сформировано ТЗ.

Также была сформулирована экономическая часть с выгодой эффективности проекта.

Данное приложение будет доступо каждому студента Московского университета имени С. Ю. Витте, позволит ему лучше адаптироваться в учебном процессе.

По результатам был сделан расчет затрат на разработку информационной системы и сделан анализ экономической эффективности и целесообразности разработки и внедрения данного продукта.

# СПИСОК ЛИТЕРАТУРЫ

1. Руководство по библиотеке JSOP: Официальный сайт – URL: <https://pypi.org/project/jsop/> (дата обращения 17.05.2024) – Текст электронный.
2. Руководство по Android Studio: Перевод: Яндекс переводчкик – URL: <https://translated.turbopages.org/proxy_u/en-ru.ru.fc41e1f1-665de8c1-a8683096-74722d776562/https/www.geeksforgeeks.org/a-complete-guide-to-learn-android-studio-for-app-development/> - Текст электронный.
3. ГОСТ 34.602-2020. Информационные технологии. Техническое задание на создание автоматизированной системы. Национальный стандарт Российской Федерации: дата введения 2022-01-01 – Текст: непосредственный.
4. ЧОУ ВО «Московский университет им. С.Ю. Витте», прикладная информатика 09.03.03, образовательная программа: официальный сайт. - Москва. – URL: <https://www.muiv.ru/sveden/education/oop/> - Текст: электронный.
5. Руководство по языку Kotlin: Перевод по руководству – «Metanit», - URL: <https://metanit.com/kotlin/tutorial/> - Текст электронный.
6. Документация по JavaScript: Справочная документация. – URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide> (дата обращения 15.05.2024). – Текст электронный.
7. Информация по образовательным программам – бакалавриат – Режим доступа: – URL: <https://www.muiv.ru/sveden/education/oop/> (дата обращения: 26.04.2024г). – Текст электронный.
8. Пирская Л. В., Разработка мобильных приложений в среде Android Studio: учебное пособие. 2019 г.– URL: https://biblioclub.ru/index.php?page=book\_view\_red&book\_id=598634
9. Колисниченко Д. Программирование для Android. Самоучитель. — СПб.: Санкт-Петербург, 2017. — 736 с. URL:

<https://www.litres.ru/get_pdf_trial/5582097.pdf>

1. Майер Р. Android 2. Программирование приложений для планшетных компьютеров и смартфонов. — СПб.: Санкт-Петербург, 2017. — 672 с.
2. Статьи о программировании для Android. — URL: http://flashbot.ru/android-dev
3. Официальная справка по среде программирования. — URL: http://www.jetbrains.com
4. Форум о программировании для Android. — URL: http://www.cyberforum.ru/android-dev
5. Форум о программировании для мобильных устройств. — URL: <http://www.4pda.ru>.
6. Официальная справка по среде программирования [Электронный ресурс]// URL: <http://www.jetbrains.com>
7. Официальная справка для Android разработчиков[Электронный ресурс]// URL <http://developer.android.com/index.html>
8. Парфенова М.Я., Маликов С.Н [Методологические исследования эффективности информационных технологий: монография](https://online.muiv.ru/lib/pdf/117050.pdf). Издательство: МУ им. С.Ю. Витте, 2017. <https://online.muiv.ru/lib/pdf/117050.pdf>