**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых» (ВлГУ)

Кафедра ВТиСУ

**КУРСОВАЯ РАБОТА**

по дисциплине «Разработка мобильных приложений»

на тему «Разработка мобильных приложений на Android. Часть 7.

Разработка мобильного приложения – переводчика».

Студент: Гребенщикова Ю.В.

Институт: ИИТЭ

Направление: 09.03.03

Группа: ПИ-121

Руководитель: Шутов А.В.

Владимир, 2024 г.

Содержание

[Введение 3](#_Toc184480629)

[1. Теоретическая часть 4](#_Toc184480630)

[1.1 Понятие мобильного приложения 4](#_Toc184480631)

[1.2 Классификация мобильных приложений 5](#_Toc184480632)

[2. Практическая часть 9](#_Toc184480633)

[2.1 Основные функции реализуемого приложения 9](#_Toc184480634)

[2.2 Анализ существующих аналогов реализуемого приложения 9](#_Toc184480635)

[2.3 Обзор ОС для разработки мобильного приложения 11](#_Toc184480636)

[2.3.1 iPhone SDK 12](#_Toc184480637)

[2.3.2 Android SDK 12](#_Toc184480638)

[2.4 Обзор платформ для разработки мобильного приложения 14](#_Toc184480639)

[2.4.1 React Native 15](#_Toc184480640)

[2.4.2 Xamarin 15](#_Toc184480641)

[2.4.3 Flutter 16](#_Toc184480642)

[2.5 Интеграция с API Yandex Cloud 18](#_Toc184480643)

[2.6 Обзор разработанного приложения 21](#_Toc184480644)

[Заключение 27](#_Toc184480645)

[Список использованной литературы: 28](#_Toc184480646)

# **Введение**

В современном мире число людей, говорящих на различных языках, постоянно растёт, и международные связи становятся всё более важными. Путешествия, бизнес и культурный обмен требуют от нас способности понимать и общаться с представителями других языков и культур. Однако языковой барьер часто становится серьёзным препятствием для эффективного взаимодействия. В связи с этим мобильные приложения, выполняющие функцию переводчиков, становятся всё более востребованными и актуальными.

Целью данного курсового проекта является разработка мобильного приложения – переводчика, которое будет сочетать в себе удобство использования, высокое качество перевода, поддержку всех языков и неограниченное количество символов для перевода.

Приложение будет служить помощником в повседневной жизни, будь то общение с иностранцами, изучение нового языка или путешествие за границу.

Задачи курсового проекта, которых необходимо решить для достижения цели:

1. Постановка цели и задач на разработку;
2. Разработка компонентов пользовательского интерфейса;
3. Выбор платформы разработки приложения;
4. Интеграция с API Yandex cloud;
5. Разработка логики приложения.
6. **Теоретическая часть**

**1.1 Понятие мобильного приложения**

Мобильное приложение — это программное обеспечение, специально

разработанное под конкретную мобильную платформу (iOS, Android, Windows Phone и т. д.). Предназначено для использования на смартфонах, фаблетах, планшетах, умных часах и других мобильных устройствах.

Мобильные приложения пишутся на языках программирования высокого уровня, а затем компилируются в машинный код операционной системы для получения максимальной производительности.

Разработка приложений имеет свои особенности: мобильные устройства работают от батареи и комплектуются менее производительными процессорами, чем персональные компьютеры. Кроме того, современные смартфоны и планшеты повсеместно оснащены дополнительными устройствами, такими как гироскопы, акселерометры и фотокамеры, дающими уникальные возможности для расширения функционала приложения.

Обычно мобильные устройства продаются уже с некоторыми предустановленными приложениями. Остальные по желанию пользователя можно скачать (как платно, так и бесплатно) на специализированных сервисах:

Apple AppStore, Google Play, Windows Phone Store и других. Первые магазины приложений, такие как Apple AppStore и Android Market, ставший впоследствии Google Play, появились в 2008 году. Спустя два года Американское диалектическое общество назвало термин «приложение» словом года.

По итогам 2015 года в каталогах двух самых крупных маркетов было

порядка 3 млн приложений. Количество загрузок за один год превысило 300

миллионов [1].

Сегодня у каждой организации есть выбор в форме размещения информации. Очевидно, что ПК-версия сайта, это первое, что организация пытается разработать, однако на этом не стоит останавливаться. Дополнительно разработав мобильное приложение, содержащее весь контент ПК-версии, можно получить множество преимуществ и плюсов.

Одним из них является то, что приложение достаточно лишь один раз

загрузить на мобильное устройство, и можно пользоваться им в любое время

без подключения к интернету и повторного скачивания. Это особенно удобно

в тех случаях, когда деньги на балансе телефона закончились, а приложением

нужно срочно воспользоваться.

Использование приложений экономит время и избавляет от нудных

ожиданий загрузки необходимых элементов (графика, изображение и ряд других), так как программа уже находится на телефоне в своём полном комплекте.

Создатели приложений могут разработать приложения для мобильных,

которые будут подходить отдельно для каждого типа мобильного телефона

или планшета. Так, для разных устройств могут быть предусмотрены различные функции, но они всегда остаются полезными и актуальными для их владельцев.

Пользователи приложения, как правило, устанавливают связь с нужным сайтом, оставаясь при этом в курсе всех происходящих событий и новостей определённой организации. Сегодня это имеет особую важность для тех, кто в своей деятельности большое внимание уделяет использованию интернета.

Совершенствование приложения и укрепление авторитета и репутации

компании взаимосвязаны.

## **1.2 Классификация мобильных приложений**

Классификация мобильных приложений достаточно сложная, так как многие из них являются «гибридами», то есть предлагают пользователю функции различного рода. Но все же некоторые виды мобильных предложений можно классифицировать однозначно.

Контентные приложения очень популярны среди пользователей мобильных приложений. На сегодняшний день такие виды активности, как прослушивание музыки, просмотр различных фильмов, клипов и фотографий, а также чтение цифровых книг являются максимально доступными и удобными для любого владельца мобильного гаджета, что и рождает спрос на данный сегмент мобильных приложений (рис. 1).

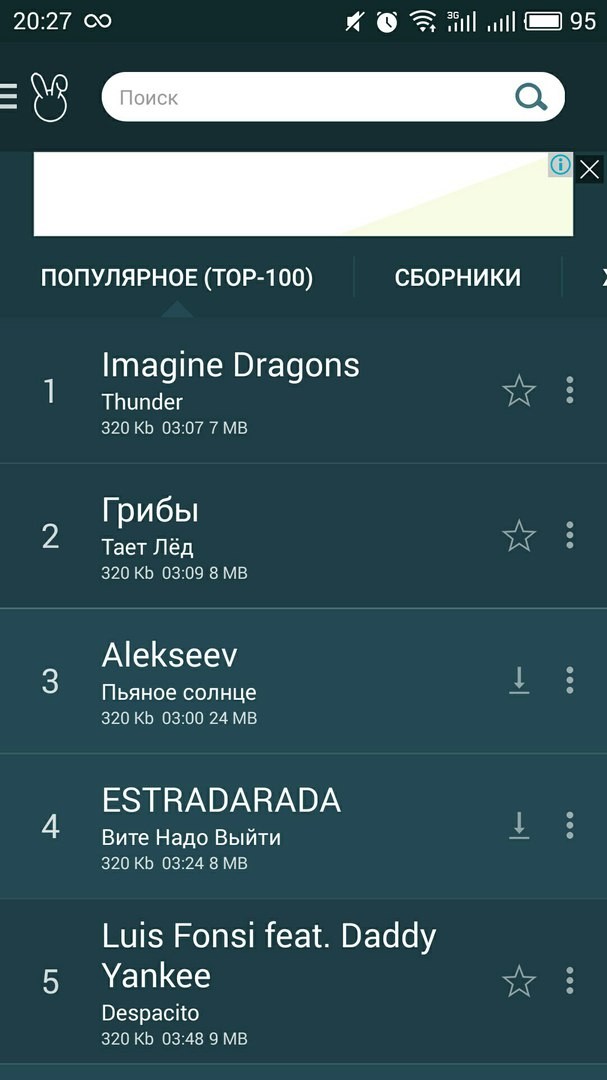


Рисунок 1 — Пример контентного приложения

Бизнес-приложения (рис. 2) стали необходимым средством для многих пользователей, которое поможет им упростить их офисную работу. В настоящий момент сегмент бизнес-приложений является предпочтительным

для инвесторов, но сложность для данного сегмента составляет перевод бизнес-задач на мобильные телефоны.

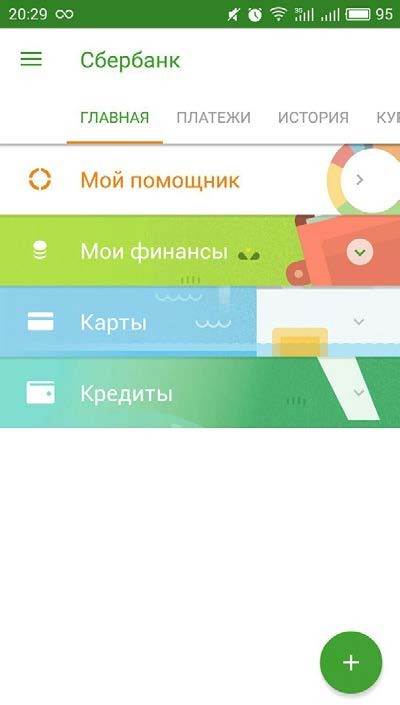


Рисунок 2 — Пример бизнес-приложения

Мобильные игры (рис. 3) наиболее востребованы на рынке мобильных приложений на сегодняшний день. Разработчики придумывают новые игры или совершенствуют уже выпущенные. Игры притягивают внимание все большей аудитории. Они становятся неотъемлемой частью жизни многих пользователей.



Рисунок 3 — Мобильная игра

Мобильные социальные сети с каждым днем набирают все большую популярность, наращивая многочисленную аудиторию по всему миру.

Социальными сетями на сегодняшний день пользуется все большее количество людей, на что оказывает влияние другая уже сложившаяся тенденция: увеличение количества пользователей смартфонов [2]. Можно сказать, что это самый популярный вид мобильных приложений, так как при покупке смартфона на нем уже по умолчанию будет установлена одно или несколько аналогичных мобильных программ.

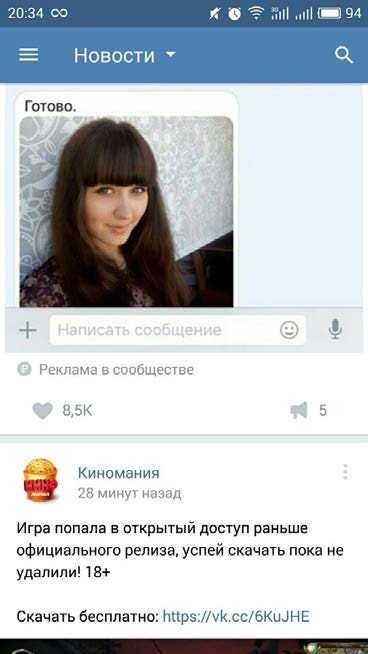


Рисунок 4 — Социальная сеть

# **Практическая часть**

**2.1 Основные функции реализуемого приложения**

В приложении разработаны следующие функции:

1. Перевод текста:

Основная функция приложения — это перевод текста с любого языка, который поддерживает Yandex Переводчик. Пользователь вводит текст и выбирает исходный и целевой языки, после чего приложение предоставляет переведённый текст.

1. Управление шрифтами:

Пользователь может уменьшить или увеличить размер шрифта для обеспечения комфортного чтения переведённого текста.

1. Изменение цвета компонента поле переводчика:

Пользователь может изменять цвет поля, где отображается перевод. Это может быть полезно для персонализации интерфейса и улучшения восприятия информации.

1. История переводов:

Сохранение предыдущих запросов на перевод, чтобы пользователь мог легко к ним вернуться или дополнить.

**2.2 Анализ существующих аналогов реализуемого приложения**

Прежде чем приступить к созданию приложения необходимо провести анализ существующих решений для того, чтобы выявить слабые стороны и не допустить таких ошибок в будущей разработке.

Проанализировав существующие решения на рынке было выбрано 3 сервиса для перевода текстов.

1. Онлайн-переводчик Google

Старая добрая «классика». Уже много лет Google Translate — один из самых популярных и доступных переводчиков с английского на русский.

Сейчас он «знает» около 130 языков. Правда, словосочетание «гугл-перевод» не зря стало саркастическим мемом: качество машинного перевода часто не на высоте. В ряде случаев, особенно с редкими языками, сервис выдает вместо осмысленного текста просто некую последовательность слов, в которой понять что-либо невозможно.

Система работает на Windows, Android, iOS. Поддерживает перевод голоса, изображений с камеры смартфона. Сохраняет историю работы. Лимит по символам — до 5000.

Преимущества:

* доступный и простой интерфейс;
* отлично справляется с переводом с английского на русский (или наоборот) отдельных фраз или словосочетаний;
* работает с разными типами «носителей»: печатным текстом, диктовкой, фото.

Недостатки:

* сложные тексты ему не по зубам;
* иногда не справляется с грамматикой;
* переводы нужно редактировать.

1. Переводчик от Яндекса

Этот сервис похож на предыдущий. На данный момент «знает» около сотни языков. Лучше всего справляется с английским и другими европейскими.

Работает на Windows, Android, iOS. Лимит по знакам — до 10 000. Поддерживает работу с голосом и видеокамерой. Может прочитать переведенный текст. Пожалуй, лучший переводчик для быстрого решения задач «без звездочки».

Преимущества:

* доступный и удобный;
* дает транскрипцию;
* понимает фразеологизмы и крылатые выражения.

Недостатки:

* тормозит при работе с редкими языками;
* не справляется со сложными текстами со специальной терминологией.

1. «DeepL Translator»

Относительно молодой сервис. Сейчас DeepL «говорит» на 32 языках. Доступен на Windows, Android, iOS. Без регистрации есть ограничение по символам — не более 1500 в одном тексте. С регистрацией — максимум 5000.

Работает с голосом и подгруженными файлами. При «обучении» этого переводчика также используют нейросети, так что он умеет «понимать» контекст и выдавать качественный результат.

Преимущества:

* прост в использовании;
* хорошо справляется даже со сложными текстами.

Недостатки:

* мало языков по сравнению с другими сервисами;
* ограничения по количеству символов.

В целом, выбор переводчика зависит от индивидуальных потребностей пользователя. DeepL подходит для качественного литературного перевода, Яндекс — для многоязычной поддержки, а Google Translate — для универсальности и дополнительных функций.

Поскольку каждый из сервисов имеет как преимущества, так и недостатки, разработав собственное приложение «переводчик» можно решить данную проблему, превратив недостатки в достоинства.

## **2.3 Обзор ОС для разработки мобильного приложения**

В настоящее время существует хороший выбор языков программирования для разработки мобильных приложений. Это связанно с тем, что для различных мобильных устройств, приходится использовать различные языки программирования.

Обычно это связано с тем, что мобильные устройства имеют различные операционные системы (ОС), например, iPhone SDK (основной язык – ObjectiveC), Android SDK (основной язык – Java).

### **2.3.1 iPhone SDK**

Разработка под iPhone под операционную систему iOS возможна только под Mac OS X. Но в Интернете можно найти статьи, как можно программировать и на Macintosh и даже на VM. Стоит заметить, что Apple предоставляет инструменты бесплатно, платить придется за подписку разработчика.

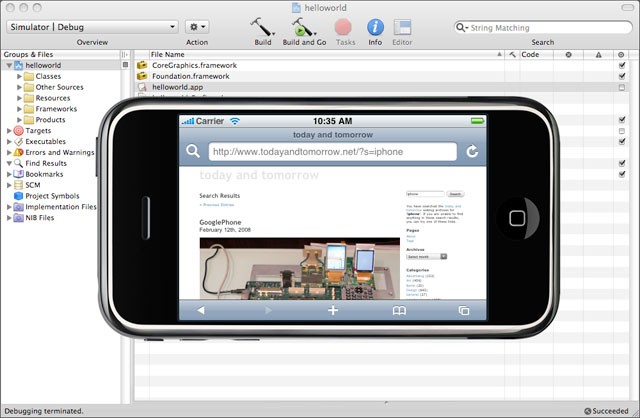


Рисунок 5 — Средство разработки для iPhone

Для написания программ под iPhone предлагается использовать ObjectiveC (рис. 5). При этом есть возможность писать так же и на C и на C++ (для этого необходимо изменять расширения файлов с .m на .mm).

Правда при этом полностью уйти от ObjectiveC не удастся, почти весь API рассчитан именно на него, исключения составляют, например OpenGL (хотя для его инициализации придется использовать несколько строк кода на ObjectiveC), так же полностью доступны стандартные библиотеки C/C++ (так, например, с файловой системой можно работать как средствами SDK на ObjectiveC, так и используя стандартную библиотеку С для ввода/вывода (fopen(), fgetc(), etc()).

Отладка приложения происходит с помощью среды XCode и эмулятора iPhone установленного в ней.

### **2.3.2 Android SDK**

Для разработки под Android можно использовать среду Eclipse которая до недавнего времени был наиболее популярной средой разработки Android, многие опытные разработчики продолжают его использовать или Android Studio (рис. 6) — более молодая среда разработки, тем не менее быстро набирающая популярность как более интеллектуальная и удобная. Кроме того, Android Studio активно развивается и поддерживается компанией Google как официальная среда разработки Android приложений.

Основой среды Android Studio для разработки Android–приложений стал проект российской компании IntelliJ IDEA.

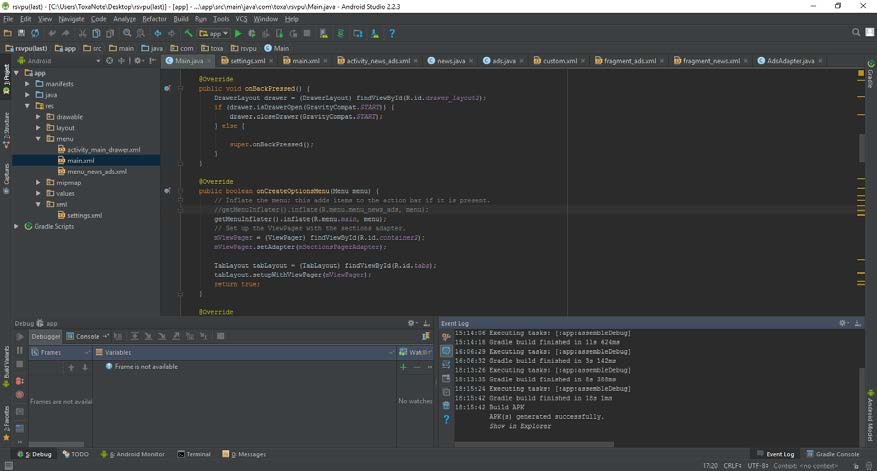


Рисунок 6 — Средство разработки для Android

Компания Google в рамках конференции Google I/O представила собственную интегрированную среду разработки Java–приложений для Android под названием Android Studio. Эта среда базируется на открытой платформе IntelliJ IDEA, которая была разработана питерской компанией JetBrains.

До настоящего времени у Google не было собственной интегрированной среды разработки (Integrated Development Environment – IDE), с помощью которой разработчики бы могли писать Android-приложения. Поэтому они пользовались другими доступными средами.

Наиболее популярными являются: Eclipse, свободная среда, поддерживаемая организацией Eclipse Foundation; свободная среда NetBeans и IntelliJ IDEA.

Существует различные версии SDK, которые используются для написания кода для различных версий Android. Поддерживается почти полная обратная совместимость версий.

Кроме разработки на языке Java поддерживается возможность более низкоуровневая разработка с использованием Android NDK (Native Development Kit) на языке C/C++.

Если совсем еще недавно можно было утверждать, что создание приложений лучше начинать с iOS–устройства, то на сегодня ответ уже не столь очевиден. Android–устройства заполонили рынок и их число уже превысило общее предложение гаджетов, работающих на iOS. Поэтому желая осуществить заказ на игры на Android или другого софта, вы планируете открыть свой бизнес для огромной аудитории [[13](#_bookmark38)].

Android–аудитория не только широкая, но и весьма разноликая. Скачать на Android программы могут люди самого разного достатка, самых разных возрастов. Компании–разработчики Android – устройств изготавливают свои гаджеты не только в сегменте дорогих моделей, но и предлагают бюджетные их варианты. Большинство пользователей могут скачать приложения со специального хранилища Google Play.

Разработка под Android имеет ряд особенностей. Операционная среда Android за время своей жизни пережила множество выпусков.

Android гаджеты характеризуются широким выбором размеров и разрешений дисплеев. Каждый изготовитель старается удивить своих пользователей, побаловать их новыми возможностями. Поэтому в продаже доступны устройства с размером дисплеев от 4 до 10 дюймов. Даже среди смартфонов есть 6–дюймовые девайсы. А разрешительная способность представлена еще шире. Это другая особенность Android гаджетов.

Android имеет большую базу пользователей и, как следствие, у представленного в приложении продукта будет широкая аудитория.

**2.4 Обзор платформ для разработки мобильного приложения**

Для разработки приложения на Android существует огромное множество платформ, рассмотрим некоторые из них:

**2.4.1 React Native**

React Native — это популярная платформа с открытым исходным кодом, разработанная Facebook, которая позволяет разработчикам создавать нативные приложения с использованием JavaScript и React. Благодаря обширному сообществу и постоянным обновлениям React Native стал одной из наиболее широко используемых платформ для разработки кроссплатформенных приложений.

Ключевые особенности React Native включают в себя:

* Единая кодовая база для приложений Android и iOS.
* Использование собственных компонентов для повышения производительности взаимодействия с пользователем.
* Горячая перезагрузка для быстрой разработки и тестирования приложений.
* Интеграция с популярными библиотеками и инструментами React.

React Native доступен для Windows, macOS и Linux, и его можно использовать бесплатно.

Недостатки написания приложения на React Native:

* Может потребоваться написание нативного кода для некоторых сложных задач.
* Производительность немного ниже, чем у нативных приложений или Flutter в некоторых сценариях.

**2.4.2 Xamarin**

Xamarin, продукт Microsoft, представляет собой кроссплатформенную среду разработки приложений, использующую C# для создания нативных приложений для Android, iOS и Windows. Xamarin позволяет разработчикам совместно использовать код на разных платформах, сохраняя при этом возможность доступа к собственным API и функциям для повышения производительности.

Ключевые особенности Xamarin включают в себя:

* Общая база кода C# для приложений Android, iOS и Windows.
* Интеграция Visual Studio для беспрепятственного опыта разработки.
* Доступ к встроенному API для повышения производительности и возможностей приложений.
* Поддержка популярных библиотек и инструментов .NET.

Xamarin доступен для Windows и macOS и бесплатен для проектов с открытым исходным кодом и небольших команд.

Недостатки написания приложения на Xamarin:

* + Большой размер приложения из-за дополнительных библиотек.
  + Меньшее сообщество по сравнению с React Native и Flutter, что может затруднить поиск помощи.

**2.4.3 Flutter**

Flutter — это пакет разработки пользовательского интерфейса с открытым исходным кодом, созданный Google, который позволяет создавать скомпилированные в собственном коде высокопроизводительные приложения для Android, iOS, Интернета и настольных компьютеров с единой кодовой базой. Flutter, созданный на языке программирования Dart, предлагает отличную производительность и множество готовых виджетов для быстрого и бесперебойного процесса разработки.

Особенности Flutter включают в себя:

* Единая кодовая база для Android, iOS, Интернета и настольных компьютеров.
* Быстрая разработка с горячей перезагрузкой и богатым набором готовых виджетов.
* Производительность, подобная нативной, благодаря заранее скомпилированному коду Dart.
* Настраиваемые и расширяемые компоненты пользовательского интерфейса с помощью пакетов.

Flutter доступен для Windows, macOS и Linux, и его можно использовать бесплатно.

Недостатки написания приложения на Flutter:

* + Относительно новый фреймворк, что приводит к меньшему количеству доступных библиотек, особенно для специфичных задач.
  + Меньшая поддержка для некоторых платформ по сравнению с нативной разработкой.

В таблице 1 приведены основные характеристики выбранных средств разработки:

Таблица 1. Характеристики популярных средств разработки кроссплатформенных мобильных приложений

|  |  |  |  |
| --- | --- | --- | --- |
|  | Flutter | React Native | Xamarin |
| Язык | Dart | JS | C# |
| Приложения | Свой UI, свой код | Нативный UI, общий код | Нативный UI, общий код |
| Разработчик | Google | Facebook | Microsoft |
| Open Source | да | да | да + платные пакеты |
| Инструменты создания UI | Собственные виджеты | Нативные компоненты + декларативный UI | Xamarin.IOS/Android + Xamarin.Forms |
| Производительность | Очень высокая | Высокая | Высокая |
| Релиз | 2017 | 2015 | 2011 |
| Платформы | IOS, Android, Web, Desktop | IOS, Android, UWP | IOS, Android, UWP |

С учетом всех перечисленных факторов и характеристик, React Native представляется наиболее удобной платформой для разработки мобильного приложения переводчика с API Yandex Cloud.

Это связано с широкой поддержкой сообщества, множеством готовых библиотек, а также возможностью быстрой разработки и легкой интеграции с различными API. JavaScript, как язык программирования, обеспечивает гибкость и позволяет разработчикам быстро находить решения для различных задач.

Таким образом, React Native предлагает оптимальный баланс между производительностью и удобством разработки для приложения переводчика.

* 1. **Интеграция с API Yandex Cloud**

Для начала работ нужно иметь аккаунт Яндекса. Перейти в https://console.cloud.yandex.ru/.

Создать платежный аккаунт с привязкой к банковской карте, без этого сервисный аккаунт не создается. Создается на странице <https://console.cloud.yandex.ru/billing>

По умолчанию присутствует каталог «default», скопировать его идентификатор и перейти в него.

В рамках каталога создать сервисный аккаунт с присвоением ему роли ai.translate.user (рис. 7).

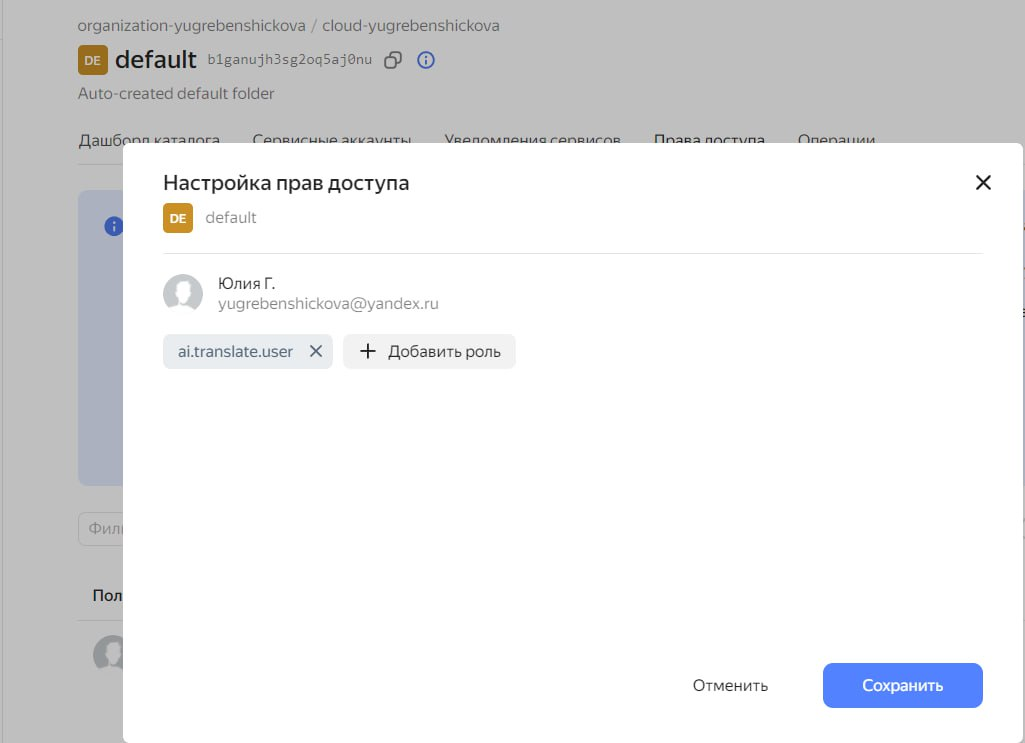


Рисунок 7 – Настройка прав доступа

После добавления перейти в профиль сервисного пользователя и создать API ключ и получить секретный ключ (рис. 8).

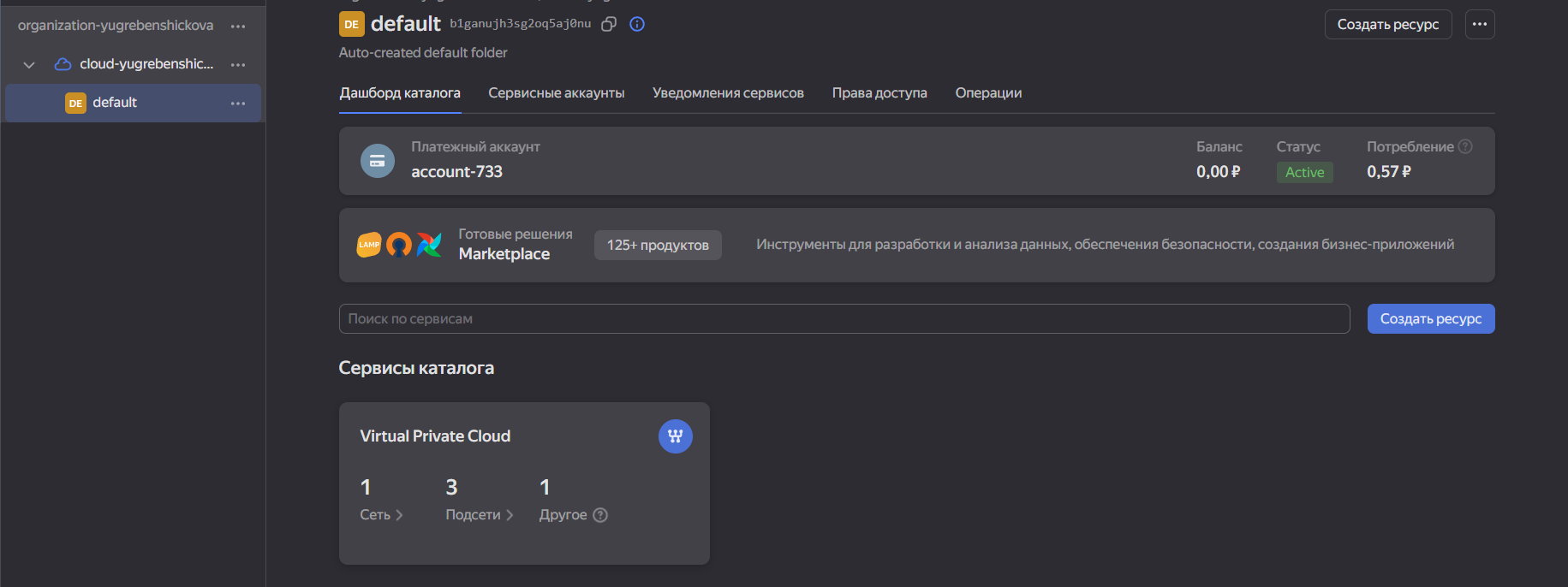


Рисунок 8 – Профиль сервисного пользователя

Теперь необходимо прописать код для файлов: get translator, get language.

Файл get language содержит – файл с языками, которые берется с сервиса Yandex cloud по передаваемым параметрам (заголовкам).

import axios from 'axios'

import { folderID, iamToken } from "./keys";

export type Language = {

  code: string;

  name: string;

};

const url = 'https://translate.api.cloud.yandex.net/translate/v2/languages';

export const getLanguage = async () => {

    try {

        const response = await axios.post<{ languages: Language[] }>(

            url,

            {

                folderId: folderID,

            },

            {

                headers: {

                'Content-Type': 'application/json',

                'Authorization': iamToken,

                },

            }

            );

        return response.data.languages || [];

    } catch (error) {

        console.error('Языки не получены', error);

    } finally {

        // setIsLoading(false);

    }

};

Файл get translator содержит – функцию для перевода текста, типизацию ответа, сам текст целевой и переведенный и возвращаемый объект для перевода и языка перевода.

import axios from 'axios';

import { folderID, iamToken } from './keys';

export interface GetTranslateProps {

  text: string;

  language: string;

}

// Типизация для ответа от API

interface TranslateResponse {

  translations: { text: string; detectedLanguageCode: string }[];

}

// Тип возвращаемого значения (переведенный текст и целевой язык)

export interface TranslationResult {

  translatedText: string;

  language: string;

}

// Функция для получения перевода

export const getTranslate = async (

  text: string,

  language: string

): Promise<TranslationResult | undefined> => {

  const url = 'https://translate.api.cloud.yandex.net/translate/v2/translate';

  try {

    const response = await axios.post<TranslateResponse>(

      url,

      {

        folderId: folderID,

        texts: [text],

        targetLanguageCode: language,

      },

      {

        headers: {

          'Content-Type': 'application/json',

          Authorization: iamToken,

        },

      }

    );

    // Возвращаем объект с переводом и языком

    const translatedText = response.data.translations[0]?.text;

    const detectedLanguage = response.data.translations[0]?.detectedLanguageCode;

    if (translatedText) {

      return {

        translatedText,

        language: detectedLanguage

      };

    } else {

      console.error('Translation not found.');

      return undefined;

    }

  } catch (error) {

    console.error('Error fetching translation:', error);

    return undefined; // Возвращаем undefined в случае ошибки

  }

};

* 1. **Обзор разработанного приложения**

Интерфейс приложения может изменяться в зависимости от потребностей пользователя. Можно уменьшить или увеличить размер шрифта для обеспечения комфортного чтения переведённого текста, а также изменить цветовую гамму для персонализации интерфейса и улучшения восприятия информации. (рис. 9, 10)

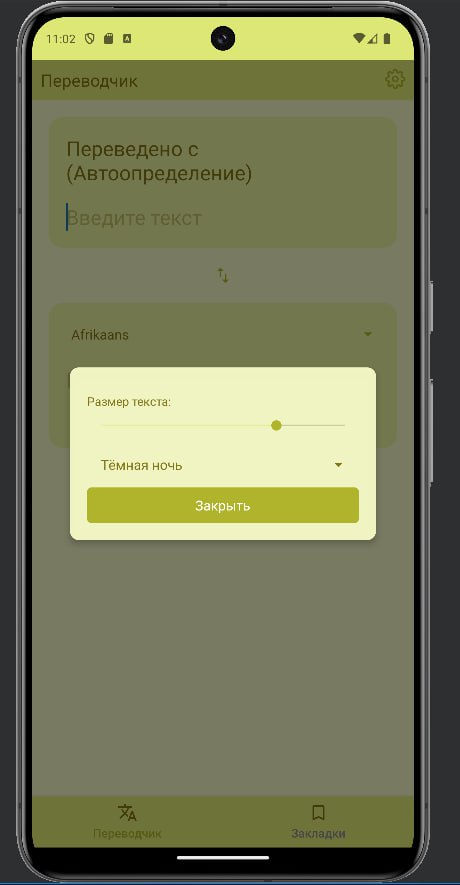


Рисунок 9 – Настройка пользовательского интерфейса

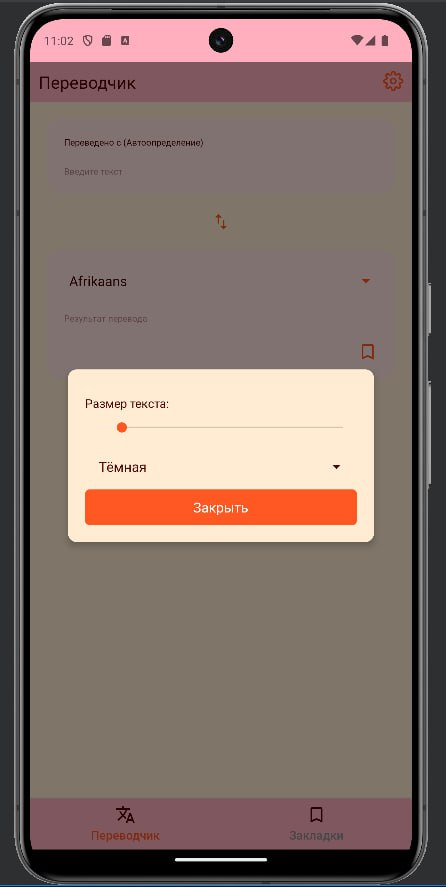


Рисунок 10 – Настройка пользовательского интерфейса

Также пользователь может сохранять перевод, для того, чтобы вернуться к нему или дополнить. (рис. 11)

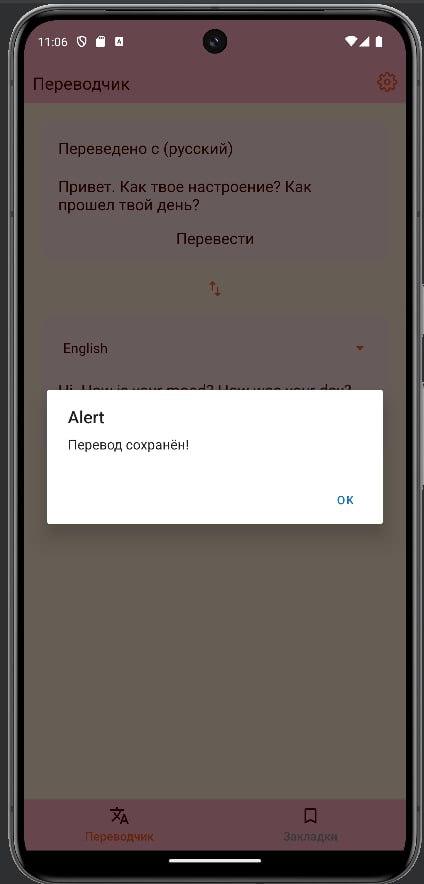


Рисунок 11 – Сохранение переведённого текста

Непосредственно сам перевод текста может быть осуществлен на языки, поддерживаемые сервисом Yandex Cloud. (рис. 12-14)

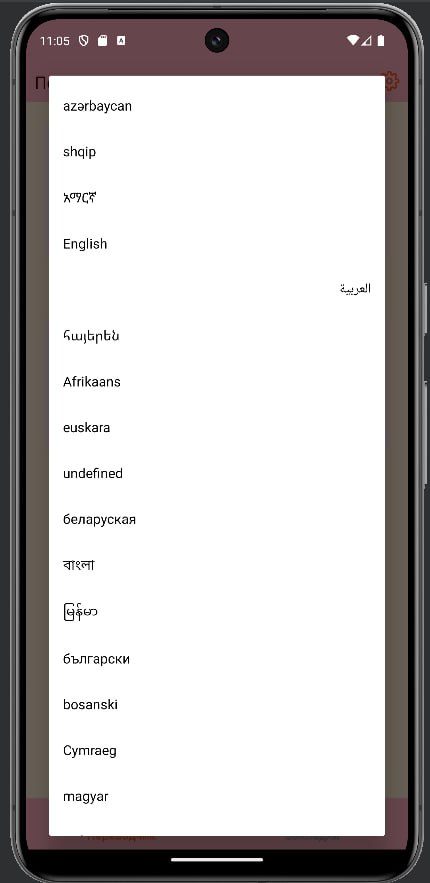


Рисунок 12 – Список языков для перевода

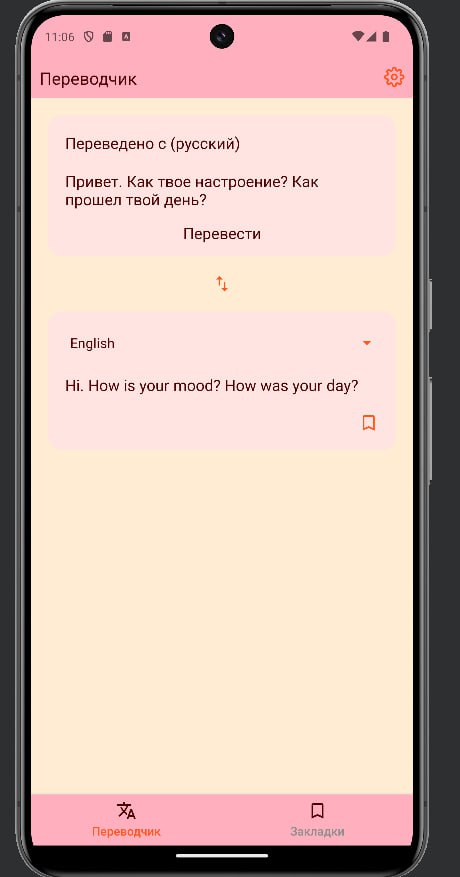


Рисунок 13 – Перевод текста на английский язык



Рисунок 14 – Перевод текста на венгерский язык

Также возможно и удаление сохраненного ранее перевода, если это уже является не актуальным.

Изначально список сохраненных переводов выглядит так (рис. 15):



Рисунок 15 – Сохраненные переводы

Для удаления перевода, необходимо нажать на кнопку удаления, после чего вы получаете уведомление, которое необходимо подтвердить (рис. 16)

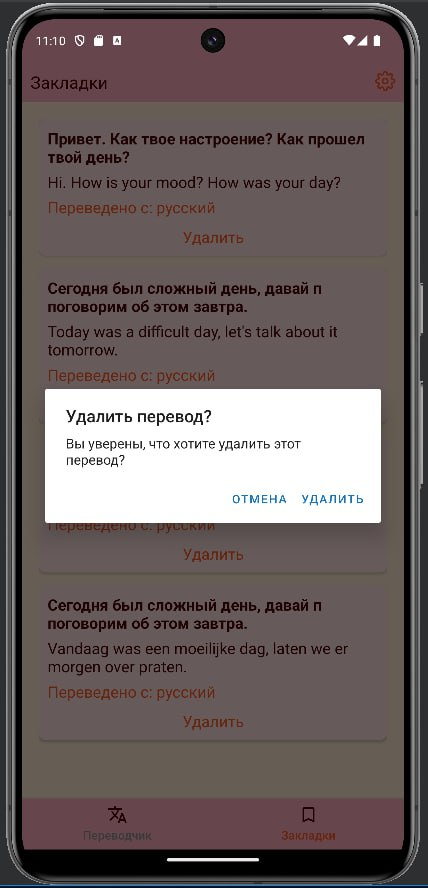


Рисунок 16 – Уведомление о удаление перевода

Проверяем, что перевод был удален (рис. 17)

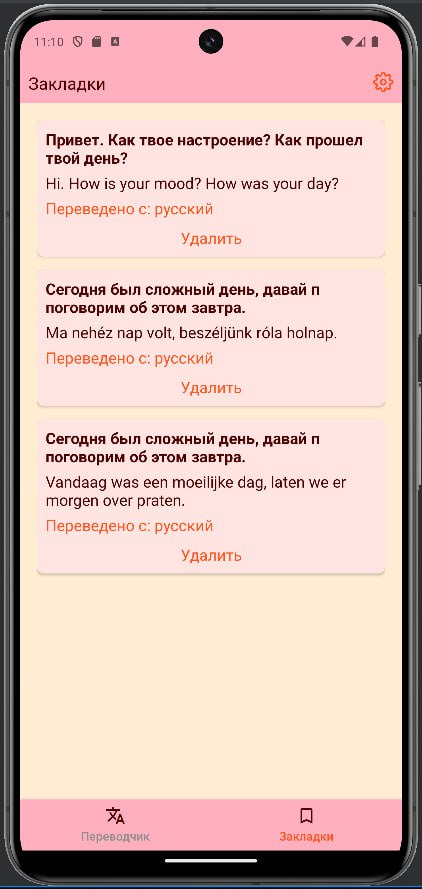


Рисунок 17 – Перевод удален

**Заключение**

В ходе работы было разработано мобильное приложение переводчика.

При выполнении работы были использованы следующие программные средства: API Yandex cloud, платформа разработки React Native на язык JavaScript (node.js). Разработка клиентской и серверной части приложения велась на языках программирования JavaScript (node.js).

В теоретической части курсовой работы рассмотрены роль и место мобильных приложений в современном обществе, а также понятие мобильного приложения и его классификация.

В практической части рассмотрены основные функции, которые были реализованы при разработке мобильного приложения, проанализированы аналоги существующего решения на рынке.

Проведен обзор ОС и платформ, для разработки мобильных приложений.

Проведено обоснование выбора используемых при разработке программных средств с описанием их преимуществ по сравнению конкурентами.

Описана интеграция мобильного приложения с API Yandex Cloud и проведен обзор разработанного приложения.

Было создано мобильное приложение, которое помогает пользователю перевести текст на выбранные языки.

Реализованы следующие функции: настройка размера шрифта и темы пользовательского интерфейса, перевод текста, сохранение и удаление переведенного ранее текста.

# **Список использованной литературы:**

1. Использование мобильных устройств [Электронный ресурс]. – Режим доступа: <http://www.wi-life.ru/stati/wi-fi/marketingovye-stati-2/mobiledevices-use-aruba-research-results/>.– Дата обращения: 25.11.2024.
2. Рынок мобильных приложений в России и мире [Электронный ресурс]. – Режим доступа: <http://www.json.ru/poleznye_materialy/free_market_watches/analytics/rynok_mobilnyh_prilozhenij_v_rossii_i_mire/>. – Дата обращения: 25.11.2024.
3. Федоров, В. Н. Разработка мобильных приложений: от идеи до реализации / В. Н. Федоров. – СПб.: Питер, 2019. – 256 с.
4. Козлов, С. В. API в мобильных приложениях: технологии и тренды / С. В. Козлов. – М.: Эксмо, 2022. – 280 с.
5. Стефанов, И. П. React Native для разработки мобильных приложений / И. П. Стефанов. – М.: БХВ-Петербург, 2021. – 320 с.
6. Яндекс. Облачные технологии. Документация по API [Электронный ресурс]. – Режим доступа: https://cloud.yandex.ru/docs. – Дата обращения: 22.11.2024.
7. Трубин, А. В. Введение в разработку мобильных приложений / А. В. Трубин. — Новосибирск: Сибирское университетское издательство, 2021. — 300 с.
8. Меркулов, Д. С. Node.js. Разработка веб-приложений / Д. С. Меркулов. — Москва: БХВ-Петербург, 2020. — 384 с.
9. Ребров, Е. Д. Основы работы с API / Е. Д. Ребров. — Казань: Казанский федеральный университет, 2022. — 240 с.
10. Дьяков, И. П. Основы проектирования пользовательского интерфейса / И. П. Дьяков. — Москва: Инфра-М, 2020. — 320 с.