**Реферат**

Изм.

№ докум.

Подпись

Дата

Лист

1

ДП 00.00.ПЗ

Разраб.

Журавлёв В.Е.

в В.Е.

В

Провер.

Герман Ю.О.

Консульт.

Герман Ю.О.

Н. Контр.

Жиляк Н.А.

Утверд.

Смелов В.В.

Реферат

Лит.

Листов

1

*БГТУ 84419008, 2017*

У

Дипломный проект содержит в пояснительной записке 70 страниц, 23 рисунка, 11 таблиц, 4 листинга, 12 источников литературы и 2 приложения.

GEOSERVER, UML, СУБД POSTGRESQL, JAVASCRIPT, GO, ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ, ПРОЕКТИРОВАНИЕ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА, СЕБЕСТОИМОСТЬ ПРОГРАММНОГО СРЕДСТВА, МЕРОПРИЯТИЯ ПО ОХРАНЕ ТРУДА

Целью данного дипломного проектирования является создание програмного средства Негорельский лесхоз.

В первом разделе дипломного проекта приведён аналитический обзор предметной области по теме дипломного проекта.

Во втором разделе описаны выбранные программные средства и технологии для разработки программного средства.

В третьем разделе описан процесс разработки и программной реализации приложения.

В четвёртом разделе приведено руководство пользователя и описание процесса тестирования созданного мобильного приложения.

В пятом разделе приводится расчет экономических параметров и себестоимость программного средства.

В шестом разделе дипломного проекта содержатся требования по охране труда и безопасности жизнедеятельности.

Графическая часть включает:

* логическую схему базы данных;
* архитектуру программного средства;
* блок-схему алгоритма обработки клиентом ответа сервера;
* блок-схему алгоритма запуска приложения.

Содержание

Изм.

№ докум.

Подпись

Дата

Лист

1

ДП 00.00.ПЗ

Разраб.

Журавлёв В.Е.

Провер.

Герман Ю.О.

Консульт.

Герман Ю.О.

Н. Контр.

Жиляк Н.А.

Утверд.

Смелов В.В.

Содержание

Лит.

Листов

2

*БГТУ 84419008, 2017*

У

[Содержание 2](#_Toc8309449)

[Введение 4](#_Toc8309450)

[1 Аналитический обзор предметной области 6](#_Toc8309451)

[1.1 Анализ существующих систем 6](#_Toc8309452)

[1.1.1 QGIS 6](#_Toc8309453)

[1.1.2 gvSIG 7](#_Toc8309454)

[1.1.3 ArcGIS 8](#_Toc8309455)

[1.2 Патентный поиск аналогичных прораммных средств 9](#_Toc8309456)

[1.3 Постановка задачи 10](#_Toc8309457)

[Выводы по разделу 1 10](#_Toc8309458)

[2 Проектирование структуры программного средства 11](#_Toc8309459)

[2.1 Анализ требований к проекту 13](#_Toc8309460)

[2.2 Проектирование 13](#_Toc8309461)

[Выводы по разделу 2 16](#_Toc8309462)

[3 Программная реализация 17](#_Toc8309463)

[3.1 Выбор среды разработки и языка программирования для реализации проекта 17](#_Toc8309464)

[3.2 Реализация системы перевода 18](#_Toc8309465)

[3.3 Реализация хранения истории переводов 18](#_Toc8309466)

[3.3 Реализация нечеткого поиска по истории перевода 21](#_Toc8309467)

[Вывод по разделу 3 23](#_Toc8309468)

[4 Тестирование и руководство пользователя системы 25](#_Toc8309469)

[4.1 Тестирование приложения 25](#_Toc8309470)

[4.2 Руководство пользователя 30](#_Toc8309471)

[Вывод по разделу 4 35](#_Toc8309472)

[5 Технико-экономическое обоснование проекта 37](#_Toc8309473)

[5.1 Общая характеристика программного средства 38](#_Toc8309474)

[5.2 Исходные данные 39](#_Toc8309475)

[5.3 Методика обоснования цены 39](#_Toc8309476)

[5.3 Определение объема программного средства 40](#_Toc8309477)

[5.4 Расчет трудоемкости выполняемой работы 41](#_Toc8309478)

[5.5 Расчет основной заработной платы 42](#_Toc8309479)

[5.6 Расчет дополнительной заработной платы 42](#_Toc8309480)

[5.7 Расчет отчислений в Фонд социальной защиты населения 43](#_Toc8309481)

[5.8 Расчет отчислений по обязательному страхованию от несчастных случаев на производстве и профессиональных заболеваний 43](#_Toc8309482)

[5.9 Расчет расходов на спецоборудование 43](#_Toc8309483)

[5.10 Расчет расходов на материалы 44](#_Toc8309484)

[5.11 Расчет расходов на оплату машинного времени 44](#_Toc8309485)

[5.12 Расчет прочих прямых затрат 45](#_Toc8309486)

[5.13 Расчет накладных расходов 45](#_Toc8309487)

[5.14 Расчет суммы расходов на разработку ПС ВТ 46](#_Toc8309488)

[5.15 Расчет расходов на сопровождение и адаптацию 46](#_Toc8309489)

[5.16 Расчет полной себестоимости разработки ПС ВТ 47](#_Toc8309490)

[5.17 Определение отпускной цены на ПС ВТ 47](#_Toc8309491)

[5.18 Результаты и выводы по расчетам 48](#_Toc8309492)

[Выводы 49](#_Toc8309493)

[6 Мероприятия по охране труда и безопасности жизнедеятельности 50](#_Toc8309494)

[6.1 Особенности труда пользователя ПЭВМ 50](#_Toc8309495)

[6.2 Обеспечение санитарно-гигиенических условий 52](#_Toc8309496)

[6.2.1 Освещение 52](#_Toc8309497)

[6.2.2 Параметры микроклимата 53](#_Toc8309498)

[6.2.3 Шум и вибрация 53](#_Toc8309499)

[6.2.4 Электромагнитное и ионизирующее излучения 54](#_Toc8309500)

[6.2.5 Эргономические требования к рабочему месту 56](#_Toc8309501)

[6.2.6 Режим труда 58](#_Toc8309502)

[6.3 Эргономика пользовательских интерфейсов 59](#_Toc8309503)

[6.4 Мероприятия по безопасности жизнедеятельности 63](#_Toc8309504)

[Вывод 65](#_Toc8309505)

[Заключение 66](#_Toc8309506)

[Список используемой литературы 67](#_Toc8309507)

[Приложение А. Программная реализация алгоритма Дамерау-Левенштейна. 68](#_Toc8309508)

[Приложение Б. Диаграмма классов 70](#_Toc8309509)

[Приложение Б 7](#_Toc483396909)0

Введение

Изм.

№ докум.

Подпись

Дата

Лист

1

ДП 00.00.ПЗ

Разраб.

Журавлёв В.Е.

Провер.

Герман Ю.О.

Консульт.

Герман Ю.О.

Н. Контр.

Жиляк Н.А.

Утверд.

Смелов В.В.

Введение

Лит.

Листов

2

*БГТУ 84419008, 2017*

У

В настоящее время всё больше растёт интерес к ГИС — геоинформационным системам, предназначенным для сбора, хранения, анализа и графической визуализации пространственных данных и связанной с ними информации.

Они содержат инструменты, позволяющие пользователям искать, анализировать и редактировать цифровые карты. А также многофункциональные средства анализа сведенных воедино табличных, текстовых и картографических бизнес-данных, демографической, статистической, земельной, адресной и другой информации.

Геоинформационные системы получают все большее распространение в таких областях, как управление природными ресурсами, сельское хозяйство, экология, метеорология, кадастры, городское планирование, землеустройство, экономика, транспорт, оборона.

В качестве систем поддержки принятия решений ГИС помогают улучшить обслуживание клиентов, сохранять высокий уровень конкурентоспособности, повышать прибыльность как коммерческим организациям, чья деятельность зависит от пространственной информации, так и тем, которым анализ геоинформации дает заметные преимущества. ГИС являются эффективным инструментом для выбора мест и определения зон торговли, размещения наружной рекламы и производственных объектов, диспетчеризации и маршрутизации средств доставки, информатизации риэлторской деятельности и т.д.

На сегодняшний день, значительное количество информации состоит из или включает в себя геоданные, то есть различные сведения о распределенных в пространстве или по территории объектах, явлениях и процессах. Работа с такими имеющими координатную привязку характеристиками и является сущностью одной из наиболее бурно развивающихся областей рынка программного компьютерного обеспечения – технологией географических информационных систем.

Целью данного дипломного проекта является создание геоинформационной системы для кафедры лесохозяйствования, которая бы выполняла такие задачи как хранение, загрузка, изменение географической информации Негорельского опытного лесхоза и её представление с помощью наложения на аэрофотографическую карту местности, а также возможность создания маркеров для привязки фотографий местности к конкретным географическим координатам.

Для реализации данной системы используется программное решение Geoserver, база данных PostgreSQL, фреймворк для разработки динамических веб-приложений VueJs, а также библиотеку для работы с картами OpenLayers и библиотеку компонентов VueMaterial. Выбор фремворка для разработки веб-приложения пал в пользу VueJs по причине того, что он позволяет создавать быстродействующие, легко оптимизируемые приложения. Он также предоставляет возможность широкой настройки компонентов и использования синтаксиса jsx, так хорошо известного разработчикам, практикующим использование React. Также фреймворк обзавёлся довольно большим сообществом разработчиков.

База данных PostgreSQL была выбрана, т.к. она является объектно-реляционной ,поддерживает пользовательские объекты и их поведение, включая типы данных, функции, операции, домены и индексы. Это делает Posgres невероятно гибкой и надежной.

Система разрабатывается с использованием языка программирования JavaScript.

При разработке данного приложения был произведен обзор аналогичных продуктов с целью устранения слабых и добавления сильных сторон приложений схожей тематики в Copy Translator. Был спроектирован дизайн-макет и создан прототип приложения. При разработке дизайна использовалась консультационная помощь UX (User eXperience) и UI (User Interface) дизайнеров. Структура приложения спроектирована таким образом, чтобы позволить расширять функционал в таких направлениях как, например, распознавание и перевод текста путём фотографирования, распознавание и перевод голоса, озвучка переведенного текста, коррекция ошибок и создание сервера для сохранения пользовательских настроек, истории перевода и переводов, которые они пометили как «избранные».

1 Аналитический обзор предметной области

Изм.

№ докум.

Подпись

Дата

Лист

1

ДП 01.00.ПЗ

Разраб.

Журавлёв В.Е.

Провер.

Герман Ю.О.

Консульт.

Герман Ю.О.

Н. Контр.

Жиляк Н.А.

Утверд.

Смелов В.В.

Аналитический обзор предметной области

Лит.

Листов

9

*БГТУ 84419008, 2017*

* 1. Анализ существующих систем

На текущий момент разработано множество различных систем перевода текста как платных, так и бесплатных. Лидирующие позиции занимают веб-платформы для перевода и приложения для мобильных устройств. Так как дипломный проект ориентирован на операционную систему Android, то поиск аналогичных программных средств также ориентирован именно на мобильную платформу.

* + 1. QGIS

QGIS – свободная кроссплатформенная геоинформационная система, состоящая из настольной и серверной части. В QGIS можно создавать и редактировать векторные данные, а также экспортировать их в разные форматы. Можно просматривать и накладывать друг на друга векторные и растровые данные в различных форматах и проекциях без преобразования во внутренний или общий формат.

В 2007 году QGIS стал официальным проектом Фонда по открытому геопространственному программному обеспечению (OSGeo), миссия которого состоит в том, чтобы содействовать совместной разработке программного обеспечения с открытым исходным кодом для геоматики. Это означало получение командой Quantum GIS организационной поддержки и новых перспектив для развития.

На сегодняшний день QGIS — это зрелый программный продукт, сравнимый с коммерческими аналогами и поддерживаемый международным сообществом разработчиков и пользователей.

К достоинствам данного приложения можно отнести:

* большой выбор инстпументов для создания и редактирования данных;
* поддержка многих языков;
* поддержка большого количества форматов данных;
* поддержка интеграции с многими базами данных;
* поддержка работа с практически всеми системами координат.

К недостаткам можно отнести:

* огромное количество настроек;
* большой размер приложения;
* медленная работа на слабых устройствах.

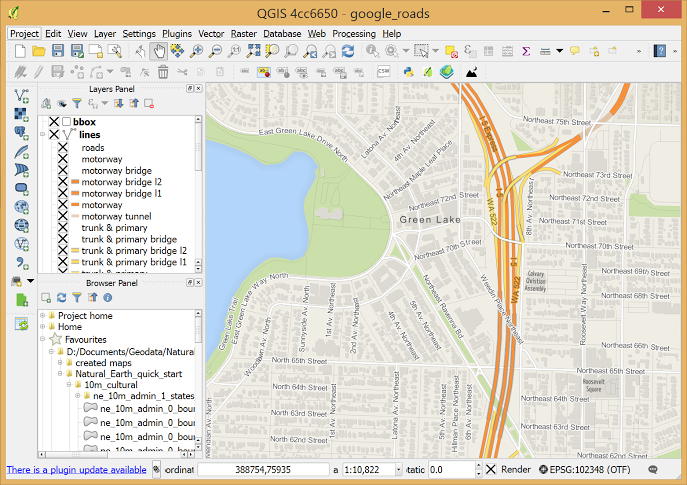


Рисунок 1.1 - Скриншот приложения QGIS

* + 1. gvSIG

gvSIG — настольная географическая информационная система, предназначенная для сбора, хранения, обработки, анализа и развёртывания любой географически привязанной информации для решения комплексных проблем управления и планирования. gvSIG известна дружественным к пользователю интерфейсом, дающим возможность доступа к наиболее распространенным форматам данных, как векторным, так и растровым. gvSIG имеет обширный набор средств для работы с географической информацией (инструменты запросов, создание макетов, геообработка, сетевой анализ и т.д.)

К положительным качествам данного переводчика можно отнести:

* большой набор инструментов для работы с географической информацией;
* работа с рспространёнными форматами данных(векторные и растровые);
* является бесплатной

К отрицательным характеристикам можно отнести следующие:

* отсутствие поддержки некоторых систем координат;
* сложна в обучении.

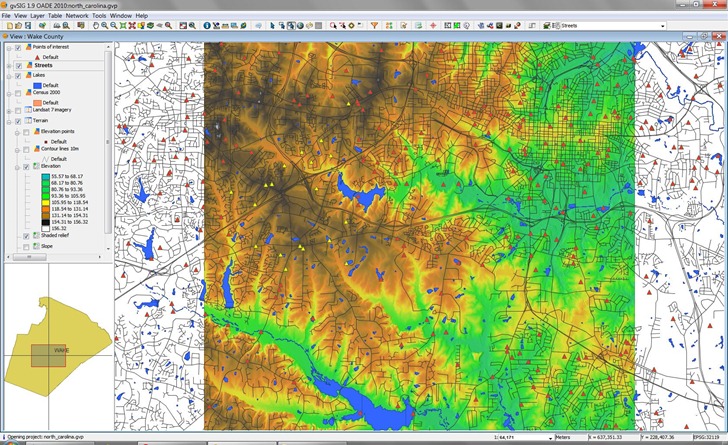


Рисунок 1.2 - Скриншот приложения gvSIG

* + 1. ArcGIS

ArcGIS – это программное обеспечение для построения ГИС любого уровня. ArcGIS помогает использовать географическую информацию для проведения анализа, лучшего понимания данных и принятия более информированных решений.

Применяется для земельных кадастров, в задачах землеустройства, учёта объектов недвижимости, систем инженерных коммуникаций, геодезии и недропользования и других областях

Основной серверный продукт — ArcGIS for Server, предназначен для многопользовательских геоинформационных проектов с централизованным хранилищем и неограниченным числом рабочих мест, публикации интерактивных карт в Интернете. Для публикации больших объёмов растровых данных выпускается продукт Image Server, для хранения пространственных данных в СУБД и интеграции с другими информационными системами предназначен продукт ArcSDE.

Преимущества приложения:

* возможность создания, редактирования и обмена картами;
* возможность создания и управления базой географических данных;
* cоздание приложений на основании карт.

Недостатки приложения:

* довольно высокая цена;
* большая сложность обучения.

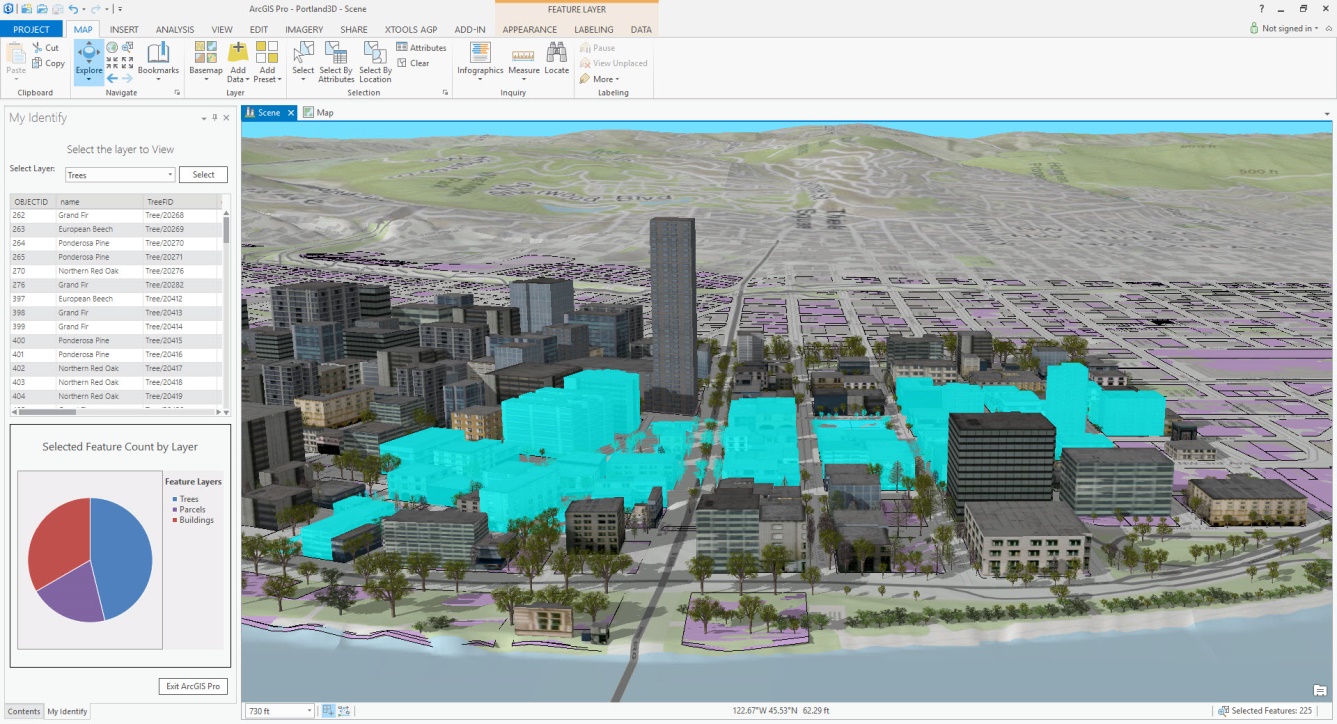


Рисунок 1.3 - Скриншот приложения ArcGIS

* 1. Патентный поиск аналогичных прораммных средств

В соответствии с темой дипломного проекта был проведен патентный поиск в области геоинформационных систем и геоинформационных систем для мониторинга земель. В ходе патентного поиска была изучена научно-техническая и патентная информация:

* описание изобретений к патентам;
* заявки на изобретения;
* рефераты иностранных изобретений.

В результате патентного поиска было найдено два патента. Предметом поиска (объектом исследования или его составной части) была указана “Геоинформационная система”.

Первым найденным патентомв является патент программного комплекса «Геоинформационная система» за номером RU 18586U1 от 13.03.2001, заявителем которого являются Жалковский Евгений Александрович и Страхов Владимир Николаевич. Программный комплекс содержит средства коммуникаций, системное программное и прикладное программное обеспечение, базу метаданных, базы геопространственных данных, средства управления приемом и выполнением заявок потребителей и средства отображения и документирования, отличается тем, что средство управления приемом и выполнением заявок выполнено в виде ядра системы, выходы и входы которого подсоединены к средствам коммуникаций, системному и программному обеспечению, средствам отображения и документирования, а также к базе метаданных и базе геопространственных данных, объединенных в одну подсистему.

Вторым найденным патентом является патент геоинформационной системы мониторинга земель за номером RU 124788 U1 от 06.08.2012, заявителем которого является Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования "Кубанский государственный университет" (ФГБОУ ВПО "КубГУ") . Программа может быть использована для обработки информации для рационального природопользования, а также в области экологии и применяться при проведении мониторинга состояния плодородия почв.

В рассмотренных патентах приведены способы автоматизированного машинного перевода текста.

1.3 Постановка задачи

Целью дипломного проекта является разработка WEB-приложения (интернет сервиса) геоинформационная система Негорельский учебно-опытный лесхоз.

Для достижения цели были поставлены следующие задачи:

* исследовать преимущества и недостатки аналогичных приложений;
* разработать интерфейс приложения;
* разработать прототип приложения на основе интерфейса приложения;
* разработать структуру базы данных;
* разработать архитектуру программной реализации приложения;
* разработать геоинформационную систему Негорельского лесхоза.

Выводы по разделу 1

В данном разделе проведен аналитический обзор существующих геоинформационных систем, выявлены их достоинства и недостатки, приводятся основные понятия, касающиеся современных геоинформационных систем обработки информации. Патентное исследование позволило выяснить, насколько изучена тема обработки географической информации в данное время.

В соответствии с выполненным анализом сформулированы цель и задачи дипломного проека.

2 Проектирование структуры программного средства

Изм.

№ докум.

Подпись

Дата

Лист

1

ДП 02.00.ПЗ

Разраб.

Журавлёв В.Е.

Провер.

Герман Ю.О.

Консульт.

Герман Ю.О.

Н. Контр.

Жиляк Н.А.

Утверд.

Смелов В.В.

Проектирование структуры ПС

Лит.

Листов

5

*БГТУ 84419008, 2017*

Программное средство, разрабатываемое в рамках данного дипломного проекта, в первую очередь, предназначено для создания удобной в использовании геоинфорационной системы Негорельского лесхоза в виде веб-приложения для максимальной совместимости с устройствами. Данное программное средство позволит загружать и просматривать графическую информацию, относящуюся к Негорельскому лесхозу.

Приложение ориентировано на пользователей, которые работают с лесохозяйственной системой негорельского лесхоза, для студентов лесохозяйственного факультета которым нужна информативная визуализация лесных массивов и инструменты для её анализа. Данное программное средство позволит повысить эффективность анализа визуальных данных лесхоза, за счёт стилизации лесных площадей по заданным критериям.

Разработка подобного программного средства в качестве веб-приложения затрагивает достаточно много аспектов разработки программного обеспечения в целом.

*Анализ требований к проекту*. На этом этапе формулируются цели и задачи проекта, выделяются базовые сущности и взаимосвязи между ними. То есть, создается основа для дальнейшего проектирования системы. В рамках данного этапа подбирается оптимальное решение проблем, определяется необходимая степень автоматизации, выявляются наиболее актуальные для автоматизации бизнес-процессы.

*Проектирование*. На основе предыдущего этапа проводится проектирование системы. Этот этап создания программного средства соединяет в себе объектную декомпозицию, приемы представления физической, логической, а также динамической и статической моделей системы.

Во время проектирования разрабатываются проектные решения по выбору платформы, где будет функционировать система языка или языков реализации, назначаются требования к пользовательскому интерфейсу, определяется наиболее подходящая база данных. Разрабатывается функциональная спецификация програмного обеспечения: выбирается архитектура системы, оговариваются требования к аппаратному обеспечению.

*Реализация*. Данный этап разработки программного обеспечения организован в соответствии с моделями эволюционного типа жизненного цикла програмного обеспечения. При разработке применяются экспериментирование и анализ, строятся прототипы как целой системы, так и ее частей. Прототипы дают возможность глубже вникнуть в проблему и принять все необходимые проектные решения еще на ранних этапах проектирования. Такие решения могут затрагивать разные части системы: внутреннюю организацию, пользовательский интерфейс, разграничение доступа и т.д. В результате этапа реализации появляется рабочая версия продукта.

*Тестирование продукта*. Тестирование тесно связано с такими этапами разработки программного обеспечения как проектирование и реализация. В систему встраиваются специальные механизмы, которые дают возможность производить тестирование системы на соответствие требований к ней, проверку оформления и наличие необходимого пакета документации.

Результатом тестирования является устранение всех недостатков системы и заключение о ее качестве.

*Внедрение и поддержка*. Данный этап обычно предусматривает следующие шаги:

* установка системы;
* обучение пользователей;
* эксплуатация.

2.1 Анализ требований к проекту

Основной целью данного программного средства является создание  веб-приложения,  которое позволит отображать картографическую и прочую визуальную информацию, относящуюся к Негорельскому лесхозу. Программное средство должно иметь бэк-энд часть, представляющую REST-сервисы к которым будет обращаться фронт-энд часть приложения. Также должен иметься настроенный геосервер и база данных для хранения географических объектов. Интерфейс пользователя  должен соответствовать основным идеям концепции построения внешнего вида приложений  Google Material Design.

Можно выделить следующие основные требования к программному продукту.

Программное средство должно являть собой веб-приложение, отобржающее Негорельский лесхоз. Программный продукт должен позволять отображать и анализировать гео информацию Негорельского лесхоза а также добавлять новую. Приложение должно быть расширяемым и простым в поддержке.

2.2 Проектирование

Проектирование программного средства в данном дипломном проекте производилось с помощью UML. UML (англ. Unified Modeling Language — унифицированный язык моделирования) – язык графического описания для объектного моделирования в области разработки программного обеспечения, моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

UML представляет собой графическую нотацию, которая предназначена для моделирования и описания всех процессов, протекающих в процессе разработки. Основу UML представляют диаграммы, которые различаются по типам и предназначены для моделирования различных аспектов разработки.

Все диаграммы можно условно разделить на поведенческие и структурные. Поведенческие диаграммы отображают процессы, протекающие в моделируемой среде. Структурные диаграммы отображают элементы, из которых состоит система. При этом одни и те же типы диаграмм могут использоваться как для моделирования бизнес-процессов, так и для непосредственного проектирования архитектуры. UML является языком широкого профиля, это – открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью [9]. UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем. UML не является языком программирования, но на основании UML-моделей возможна генерация кода. UML позволяет также разработчикам программного обеспечения достигнуть соглашения в графических обозначениях для представления общих понятий (таких как класс, компонент, обобщение (англ. generalization), агрегация (англ. aggregation) и поведение) и больше сконцентрироваться на проектировании и архитектуре. В UML используются следующие виды диаграмм (для исключения неоднозначности приведены также обозначения на английском языке):

* структурные диаграммы;
* диаграммы поведения;
* диаграммы взаимодействия.

Общая структура компонентов программного средства в данном дипломном проекте представлена в виде компонентов UML (рисунок 2.1).

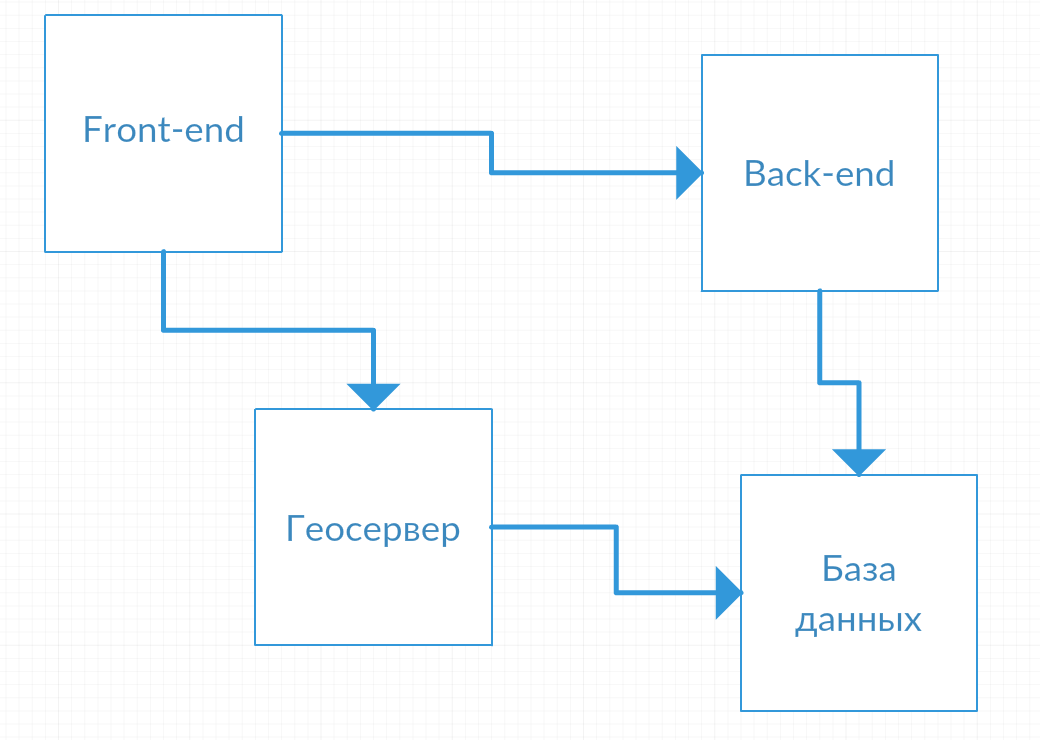


Рисунок 2.1 – Скриншот общей структуры компонентов программного средства

*Front-end* часть представлена веб-приложением, которое созданно для кроссбраузерной работы по отображнию геоданных Негорельского лесхоза. Front-end часть имеет связь с сервисами Back-end части и с REST-сервисами Геосервера.

*Back-end* – это часть системы,представленная REST-сервисами, предоставляющими функциональность авторизации и связующая между базой данных и front-end частью.

Компонент *Геосервер* представляет собой программное средство, предоставляющее для front-end части тайлы, которые преобразуются Геосервером из shape-файлов. Shape-файлы Геосервер получает из базы данных, а отображение производится при помощи front-end части. Также, геосервер предоставляет средства для стилизации тайлов по выбранным признакам,например классам бонитета или возрастом деревьев.

*База данных* – это часть програмного средства ,которая предназаначена для хранения географической информации. Данная часть должна реализовывать примернно следующий функционал: сохранение информации в формате utf-8, возможность «достать» эту информацию при помощи Геосервера или Back-end части. Также даная часть должна поддерживать большой набот типов объектов и желательно быть объекто-ориентированной.

Общая диаграмма последовательности использования приложения отображена на рисунке 2.2.

Последовательность событий начинается с того, что пользователь отткрывет приложение, при этом отображается карта Негорельского лесхоза, также подгружаются доступные для отображения слои с геосервера. Пользователь может выбрать слой для отображения или выбрать стиль для уже отображаемого. При выборе слоя,происходит запрос к геосерверу,после чего геосервер делает запрос в базу для получения shape-файла, который геосервер преобразовывает в тайлы и присылает в ответе. Данные тайлы отображаются в виде слоя. При отображении слоя, можно выбрать стиль тображения. При выборе стиля, делается запрос к геосерверу,после чего геосерве применяет стиль к выбранным слоям и возвращает стилизованные тайлы.

Для того, чтобы спроектировать программную реализацию приложения была создана UML-диаграмма классов, представленная в приложении Б.

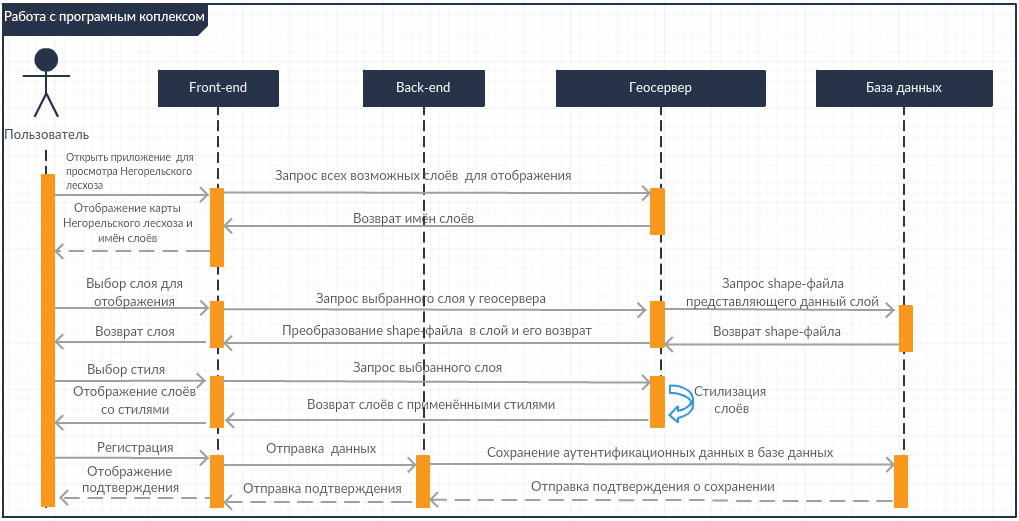


Рисунок 2.2 - Диаграмма последовательности

Ключевыми элементами системы являются Гесервер и back-end части програмного средства,которые язляются так называемым «связующим звеном» между данными, хранящимися в базе данных и пользовательским веб-интерфейсом,который позвляет отобразить данные в удобоваримом для человека виде с целью облегчения анализа.

Для хранения данных используется база данных. Для этих целей была выбрана система управления базами данных PostgreSQL из-за того,что это объектно-ориентированная база данных,которая отлично подходит для хранения данных с любой структурой. Также для PostgreSQL существуют надстройки, позволяющие комфортно работать с shape-файлами и прочей геоинформацией.

Схема таблицы базы данных для хранения истории переводов предствлена на рисунке 2.3 и содержит в себе восемь полей, одно из которых является числовым полем, обозначающим уникальный номер записи, а также представляющее собой ограничение целостности – первичный ключ. Пять полей таблцы являются текстовыми, одно поле отражает дату и время и одно поле – значение Boolean.

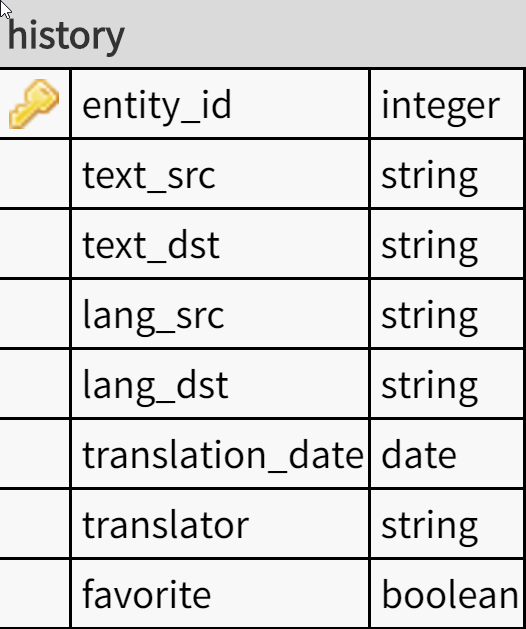


Рисунок 2.3 - Схема таблицы базы данных для хранения истории переводов

Выводы по разделу 2

Исходя из требований разрабатываемого программного средства, можно сделать вывод о том, что хранение данных в приложении лучше всего организовать посредством СУБД PostgreSQL. Данная система организовывает хранение данных в таком виде, который позволит максимально быстро реализовывать вставку, обработку и извлечение данных. Проектирование программного средства производилось в программе Sparx Enterprise Architect с помощью UML-диаграмм. Это позволило составить архитектуру приложения максимально приближенно к реализации программного средства.

3 Программная реализация

Изм.

№ докум.

Подпись

Дата

Лист

1

ДП 03.00.ПЗ

Разраб.

Журавлёв В.Е.

Провер.

Герман Ю.О.

Консульт.

Герман Ю.О.

Н. Контр.

Жиляк Н.А.

Утверд.

Смелов В.В.

Программная реализация

Лит.

Листов

6

*БГТУ 84419008, 2017*

3.1 Выбор среды разработки и языка программирования для реализации проекта

Для разаработки веб-приложения выбран язык JavaScript, так как он является основным языком веб-разработки на текущий момент. Чтобы не увеличивать сложность разработки, был выбран JavaScript фреймворк VueJS.

WebStorm – это одна из самых популярных сред разработки для JavaScript, , разработанная на основе платформы IntelliJ IDEA.

WebStorm обеспечивает автодополнение, анализ кода на лету, навигацию по коду, рефакторинг, отладку, и интеграцию с системами управления версиями. Важным преимуществом интегрированной среды разработки WebStorm является работа с проектами(в том числе, рефакторинг кода JavaScript, находящегося в разных файлах и папках проекта, а также вложенного в HTML). Поддерживается множественная вложенность (когда в документ на HTML вложен скрипт на Javascript, в который вложен другой код HTML, внутри которого вложен Javascript) — то есть в таких конструкциях поддерживается корректный рефакторинг. Данная среда разработки доступна для Windows, OS X и Linux.

Языком программирования для разработки Back-end части был выбран Go. Go – компилируемый многопоточный язык программирования, разработанный внутри компании Google. Разработка Go началась в сентябре 2007 года, его непосредственным проектированием занимались Роберт Гризмер, Роб Пайк и Кен Томпсон, занимавшиеся до этого проектом разработки операционной системы Inferno. Официально язык был представлен в ноябре 2009 года. На данный момент поддержка официального компилятора, разрабатываемого создателями языка, осуществляется для операционных систем FreeBSD, OpenBSD, Linux, macOS, Windows, DragonFly BSD, Plan 9, Solaris, Android. Также Go поддерживается набором компиляторов gcc, существует несколько независимых реализаций. Ведётся разработка второй версии языка.

Язык Go разрабатывался как язык программирования для создания высокоэффективных программ, работающих на современных распределённых системах и многоядерных процессорах. Он может рассматриваться как попытка создать замену языкам Си и C++. По словам Роба Пайка, «Go был разработан для решения реальных проблем, возникающих при разработке программного обеспечения в Google».

Go создавался в расчёте на то, что программы на нём будут транслироваться в объектный код целевой аппаратной и программной платформы и в дальнейшем исполняться непосредственно, не требуя виртуальной машины, поэтому одним из критериев выбора архитектурных решений была возможность обеспечить быструю компиляцию в эффективный объектный код и отсутствие чрезмерных требований к динамической поддержке.

В результате получился язык, «который не стал прорывом, но тем не менее явился отличным инструментом для разработки крупных программных проектов».

Хотя для Go доступен и интерпретатор, практически в нём нет большой потребности, так как скорость компиляции достаточно высока для обеспечения интерактивной разработки.

3.2 Реализация front-end части

Основным элементом этой части является система отображения карт и слоёв. Система использует библиотеку OpenLayers для работы с тайлам и и удобного представления информации в графическом виде.

Интерфейс TranslateEngine является основным интерфейсом системы перевода, от которого должны наследоваться все переводчики, используемые в приложении. Интерфейс состоит из трёх методов: translateSync, translateAsync и getTranslatorMeta.

Метод translateSync принимает на вход текст для перевода, язык исходного текста и язык, на который необходимо перевести текст. Даный метод обеспечивает синхронный перевод текста и возвращает переведенный текст в качестве результата.

Метод translateAsync принимает на вход текст для перевода, язык исходного текста и язык, на который необходимо перевести текст и экземпляр класса, реализующего интерфейс TranslateListener. Даный метод обеспечивает асинхронный перевод текста и вызывает метод onTranslate или onFailure интерфейса TranslateListener в зависимости от результата перевода: в случае успешного перевода – onTranslate, в случае исключения – onFailure.

Метод getTranslatorMeta возвращает экземпляр класса TranslateEngineMeta, который необходим для описания информации о движке перевода. Такой информацией является название движка для перевода для отображения пользователю и уникальное имя движка для уникальной идентификации движка с технической стороны.

3.3 Реализация хранения истории переводов

На данный момент для платформы Android существует несколько решений, позволяющих реализовать ORM-подход для работы с базой данных, но основных два. Это ORMLite и GreenDAO.

ORM – object-relational mapping. Объектно-реляционное отображение означает, что программисту гораздо удобнее оперировать с объектами, которые он использует в своём приложении, нежели чем с таблицами, в которых хранятся данные в реляционных базах данных. Для связи реляционных СУБД с объектной моделью приложения используются ORM-технологии. Так, для решения задач объектно-реляционного отображения в Android используют один из сторонних фреймворков. GreenDAO и ORMLite являются библиотеками с открытым кодом.

Для реализации работы с базой данных в данном дипломном проекте была выбрана ORMLite-система. ORMLite – это популярный фреймворк для Java, имеющий адаптированную версию под Android. Преимуществами данного фреймворка можно назвать мощную и в тоже время простую систему аннотаций, богатый функционал, поддержку многих БД, в том числе SQLite, и архитектуру в соответсвии с принципом KISS. KISS (акроним для «Keep it short and simple») – принцип проектирования, принятый в ВМС США в 1960 год. Принцип KISS утверждает, что большинство систем работают лучше всего, если они остаются простыми, а не усложняются []. Поэтому в области проектирования простота должна быть одной из ключевых целей, и следует избегать ненужной сложности.

Для работы с базой данных с помощью ORMLite необходимо выполнить следующие этапы.

Создать класс, отражающий таблицу в базе данных. Для хранения истории перевода был создан следующий класс:

public class TranslateEntity {

private long entity\_id;

private String text\_src;

private String text\_dst;

private String lang\_src;

private String lang\_dst;

private Date translation\_date;

private String translator;

private boolean favorite;

}

Листинг 3.1 – Исходный код класса для хранения истории перевода

Далее каждый класс, отображающий данные в базе данных, должен быть аннотирован. Стоит заметить, что у класса должен обязательно присутствовать конструктор без аргументов. Пример аннотации, реализованной для данного проекта:

@DatabaseTable(tableName = "translateEntity")

public class TranslateEntity {

@DatabaseField(generatedId = true)

private long entity\_id;

@DatabaseField(canBeNull = false)

private String text\_src;

@DatabaseField(canBeNull = false)

private String text\_dst;

@DatabaseField(canBeNull = false)

private String lang\_src;

@DatabaseField(canBeNull = false)

private String lang\_dst;

@DatabaseField(canBeNull = false)

private Date translation\_date;

@DatabaseField(canBeNull = false)

private String translator;

@DatabaseField(canBeNull = false, defaultValue = "false")

private boolean favorite;

}

Листинг 3.2 – Исходный код класса, отражающего запись в базе данных

Первая аннотация @DatabaseTable(tableName = "translateEntity") указывает название таблицы, куда будут отображены объекты этого класса. Перед каждым полем должна быть аннотация @DatabaseField с различными аргументами. Также можно не указывать аргументов – всё установится по умолчанию, а название столбца в таблице будет совпадать с названием поля. У поля favorite два аргумента: canBeNull = false означает, что в таблице этот столбец не может быть пустым, а defaultValue = "false" принудительно обязывает приводить значение столбца по умолчанию равным false.

Первичным ключем может служить любое поле – достаточно указать у него id = true, но рекомендуется сделать автогенерируемое значение id и поставить ему generatedId = true. ORMLite сама назначит ему уникальный номер. Для индексации столбца в БД указывается index = true. Полное описание всех доступных агументов аннотации полей находится в документации к ORMLite. Также помимо аннотаций ORMLite можно использовать javax.persistence аннотации.

После создания аннотация необходимо реализовать подключение к SQLite в Android. Существует несколько способов получения доступа к данным БД с помощью ORMLite в Android. Можно наследовать каждую activity от ORMLiteBaseActivity, и тогда не придется следить за жизненным циклом соединения с БД, но при этом будет невозможным получить доступ из других классов. Гораздо предпочтительнее подход, рассмотренный далее.

Нужно создать класс, который будет инстанцировать помощника в создании и работе с БД:

public class DBHelper extends OrmLiteSqliteOpenHelper {

private Dao<TranslateEntity, Long> translateEntityDao;

private Context ctx;

public DBHelper(Context ctx) {

super(ctx, Cc.DATABASE\_NAME, null, Cc.DATABASE\_VERSION);

this.ctx = ctx;

}

@Override

public void onCreate(SQLiteDatabase d, ConnectionSource c) {

TableUtils.createTableIfNotExists(c, TranslateEntity.class);

}

public void clearTranslateHistory() {

TableUtils.dropTable(getConnectionSource(), TranslateEntity.class, true);

TableUtils.createTableIfNotExists(c, TranslateEntity.class);

}

public Dao<TranslateEntity, Long> getTranslateEntityDao() {

if (translateEntityDao == null) {

translateEntityDao = getDao(TranslateEntity.class);

}

return translateEntityDao;

}

}

Листинг 3.3 –Вспомогательный класс для работы с базой данных

Обращение к нему будет происходить во время начала и конца жизни приложения, это предотвратит утечки памяти из-за незакрытого соединения с БД.

После создания всех необходимых классов можно производить операции с таблицей базы данных путём использования DAO-объекта, который возвращает функция getTranslateEntityDao. В программном обеспечении data access object (DAO) – это объект, который предоставляет абстрактный интерфейс к какому-либо типу базы данных или механизму хранения. Определённые возможности предоставляются независимо от того, какой механизм хранения используется, без необходимости специальным образом соответствовать этому механизму хранения. Этот шаблон проектирования применим ко множеству языков программирования, большинству программного обеспечения, нуждающемуся в хранении информации и к большей части баз данных, но традиционно этот шаблон связывают с приложениями на платформе Java Enterprise Edition, взаимодействующими с реляционными базами данных через интерфейс JDBC, потому что он появился в рекомендациях от фирмы Sun Microsystems.

3.3 Реализация нечеткого поиска по истории перевода

В разрабатываемом программном средстве реализован поиск по истории переводов, было решено сделать поиск нечётким. Нечеткий поиск является крайне полезной функцией любой поисковой системы. Вместе с тем, его эффективная реализация намного сложнее, чем реализация простого поиска по точному совпадению.

Нечеткий поиск - это поиск информации, при котором выполняется сопоставление информации заданному образцу поиска или близкому к этому образцу значению. Алгоритмы нечеткого поиска используются в большинстве современных поисковых систем, например, для проверки орфографии.

Классическая задача нечеткого поиска может принимать следующий вид: есть текстовая информация определенного размера, пусть пользователь вводит слово или фразу для поиска, необходимо найти в тексте все совпадения с заданным словом с учетом возможных допустимых различий. Например, при запросе пользователя «интернет» нужно найти «интерн», «интернат» и другие слова.

Для того, чтобы оценить сходство двух слов в тексте используются специальные метрики нечеткого поиска. Метрикой нечеткого поиска называют функцию расстояния между двумя словами, позволяющую оценить степень их сходства в данном контексте.

В качестве метрик используют расстояния Хемминга, Левенштейна, Дамерау-Левенштейна. Расстояние Хемминга - это число позиций, в которых соответствующие символы двух слов одинаковой длины различны. У расстояния Хемминга есть один существенный недостаток - сравнивать можно только слова одинаковой длины. Из-за этого данную метрику практически не применяют на практике.

Расстояние Левенштейна или расстояние редактирования - это минимальное количество операций (вставки, удаление одного символа и замены одного символа на другой), необходимых для преобразования одной строки в другую. Данная метрика имеет следующие недостатки: при перестановке слов в предложении расстояние принимает большое значение, расстояние Левенштейна значительно зависит от длины слова.

В качестве метрики, используемой в данной работе было выбрано расстояние Дамерау-Левенштейна. Расстояние Дамерау-Левенштейна - это мера разницы двух строк символов, определяемая как минимальное количество операций вставки, удаления, замены и перестановки соседних символов, необходимых для перевода одной строки в другую. Данная метрика отличается от расстояния Левенштейна только добавлением новой операции (перестановки).

В результате работы алгоритма Дамерау-Левенштейна составляется матрица преобразования символов, где ее значения - это расстояния для преобразования символов. Итоговым значением преобразования одной строки в другую будет элемент с максимальными индексами. Итоговое расстояние между словами «дагестан» и «арестант» будет равно 3.

Расстояние Дамерау-Левенштейна между двумя строками a и b определяется функцией как:

Где это индикаторная функция, равная нулю при и 1 в противном случае.

Каждый рекурсивный вызов соответствует одному из случаев:

*da,b*(*i*-1,*j*)+1 соответствует удалению символа (из a в b),

*da,b*(*i*,*j*-1)+1 соответствует вставке (из a в b),

*da,b*(*i*-1,*j*-1)+1 соответствие или несоответствие, в зависимости от совпадения символов,

*da,b*(*i*-2,*j*-2)+1 в случае перестановки двух последовательных символов.

Псевдокод алгоритма выглядит следующим образом:

int DamerauLevenshteinDistance(S: char[1..M], T: char[1..N]; deleteCost, insertCost, replaceCost, transposeCost: int):

// Обработка крайних случаев

if (S == "")

if (T == "")

return 0

else

return N

else if (T == "")

return M

D: int[0..M + 1][0..N + 1] // Динамика

INF = M + N // Большая константа

// База индукции

D[0][0] = INF

for i = 0 to M

D[i + 1][1] = i

D[i + 1][0] = INF

for j = 0 to N

D[1][j + 1] = j

D[0][j + 1] = INF

lastPosition: int[0..количество различных символов в S и T]

//для каждого элемента C алфавита задано значение lastPosition[C]

foreach (char Letter in (S + T))

lastPosition[Letter] = 0

for i = 1 to M last = 0

for j = 1 to N

i' = lastPosition[T[j]]

j' = last

if S[i] == T[j]

D[i + 1][j + 1] = D[i][j]

last = j

else

D[i + 1][j + 1] = min(D[i][j] + replaceCost, D[i + 1][j] + insertCost, D[i][j + 1] + deleteCost)

D[i + 1][j + 1] = min(D[i + 1][j + 1], D[i'][j'] + (i - i' - 1) \cdot deleteCost + transposeCost + (j - j' - 1) \cdot insertCost)

lastPosition[S[i]] = i

return D[M + 1][N + 1]

Листинг 3.4 – Псевдокод алгоритма Дамерау-Левенштейна

Для использования данного алгоритма в программном средстве, разрабатываемом в рамках текущего дипломного проекта, был написан java-класс, позволяющий рассчитывать расстояние Дамерау-Левенштейна между двумя строками различной длины. Java-код данного класса представлен в приложении А.

Данный алгоритм требует довольно значительных затрат памяти и времени выполнения. Однако он может быть упрощен путем сокращения количества вычислений. Сравнение строк может происходить с учетом морфем, так могут подвергаться трансформации только суффиксы слов или окончания. Другим вариантом снижения затрат может быть использование ограничений по количеству сравнений.

Вывод по разделу 3

В данном разделе обоснованы выбор языка разработки и СУБД для хранения слоёв, а также изучены особенности работы с базой данных при помощи Object-Relational Mapping технологии ORMLite, которая позволяет осуществить безопасную работу с таблицами. Создана база данных для хранения истории переводов.

Изучены особенности программирования под операционную систему Android.

Разработано мобильное Android-приложение, предназначенное для перевода текста на иностранные языки, позволяющее осуществлять перевод, не открывая само приложение, а путём копирования текста из других мобильных приложений.

В процессе разработки были рассмотрены два из наиболее известных методов нечеткого поиска текстовой информации. Каждый из данных методов имеет свои достоинства и свои недостатки. Была реализована работа алгоритма Дамерау-Левенштейна для нечёткого поиска по истории переводов.

4 Тестирование и руководство пользователя системы

Изм.

№ докум.

Подпись

Дата

Лист

1

ДП 04.00.ПЗ

Разраб.

Журавлёв В.Е.

Провер.

Герман Ю.О.

Консульт.

Герман Ю.О.

Н. Контр.

Жиляк Н.А.

Утверд.

Смелов В.В.

Тестирование и руководство пользователя

Лит.

Листов

10

*БГТУ 84419008, 2017*

4.1 Тестирование приложения

В связи с ростом количества смартфонов и устройств под управлением ОС Android все большую популярность набирают мобильные приложения.

Мобильный пользователь ожидает, что устанавливаемые им приложения просты, интуитивно понятны и работают без сбоев. Поэтому качество приложения играет очень большую роль в его популярности.

Качество приложения – это совокупность характеристик данного приложения, относящихся к его способности удовлетворять установленные и предполагаемые потребности пользователей. Оценить уровень качества приложения и выявить различные ошибки в его работе можно с помощью тестирования.

Тестирование – это поиск дефектов или багов в приложении. Термин «баг» используют для обозначения ошибки, из-за которой приложение выдает неожиданное поведение.

Планируя процесс тестирования, тестировщик часто составляет тест-кейсы. В самом простом случае тест-кейс включает в себя:

* краткое описание (то, что тестируем);

– последовательность действий (шаги, которые выполняет тестировщик,

чтобы проверить какую-либо функциональность);

– ожидаемый результат (то, что должно получиться после выполнения действий).

В дальнейшем по данным тест-кейсам тестируется приложение. Выполняя последовательность действий из тест-кейса, тестировщик получает фактический результат, который сравнивает с ожидаемым. Если данные результаты не совпадают, то это означает, что тестировщик нашел баг.

Рассмотрим особенности ручного тестирования Android-приложений. Под ручным понимают тестирование, которое проводится тестировщиком без специальных утилит или это такое тестирование, при котором тестировщик располагается за компьютером (планшетом, телефоном) и производит самостоятельно те же действия, что и пользователь.

Тестирование мобильных приложений существенно отличается от тестирования приложений, предназначенных для использования на персональных компьютерах.

Приведем ряд основных моментов, которые нужно протестировать:

– Установка и запуск приложения, выход из приложения, повторный вход, удаление приложения с мобильного устройства.

– Мультитач и размер экрана. Корректность удаления 2-х элементов или

просмотр двух элементов, нажатием на них одновременно.

– Проверка многократного быстрого нажатия на кнопку – часто при этом может случиться падение приложения. В приложении должны отсутствовать пустые экраны, чтобы пользователь не оказался в ситуации, в которой не очевидно, что делать. Также все элементы должны быть такого размера, чтобы пользователь мог однозначно нажать на них.

– Стабильность. Работа приложения при множестве запущенных приложений, долгое время работы, а также в случае недостатка места для установки или работы приложения. Поведение приложения при отсутствии в некоторых устройствах поддерживаемых приложением функций.

– Адаптация приложения к портретной и альбомной ориентациям устройства.

– Стресс. Реакция приложения на внешние прерывания: входящие и исходящие SMS, MMS, звонки, оповещения других приложений; переход устройства в режим ожидания; выключение устройства, разрядка устройства; зарядка устройства; отключение интернета; переход в другое приложение.

– Интернационализация. Проверка корректности работы приложения на разных языках (если данное приложение мультиязычное).

– Обратная связь с пользователем. Наличие информативных сообщений при попытке выполнить какое-либо действие (например, при удалении важной информации), а также присутствие визуальной индикации хода выполнения функций. У всех нажимаемых элементов должно быть «нажатое состояние» (отклик на действие), благодаря этому пользователь всегда будет видеть, действительно ли произошло нажатие.

– Обновление. Корректность обновления приложения до новой   
версии.

– Орфографические ошибки.

Автоматизированное тестирование программного обеспечения – процесс тестирования программного обеспечения, при котором основные функции и шаги теста, такие как запуск, инициализация, выполнение, анализ и выдача результата, производятся автоматически с помощью инструментов для автоматизированного тестирования.

Планируя процесс тестирования, тестировщик пишет тест-кейсы. Затем он проверяет приложение по данным тест-кейсам. Когда тестировщик находит баги, то он сообщает о них разработчику (документирует баги; часто для хранения багов используются специальные баг-трэкинговые системы). После того, как разработчик исправляет какой-либо баг, тестировщику необходимо заново выполнить тест-кейс, который обнаружил данный баг, т.е. повторно пройтись по всем шагам из этого тест-кейса (ручное тестирование).

Также для надежности необходимо выполнить и все тест-кейсы, на которые мог повлиять найденный баг. В этом случае для ускорения процесса тестирования лучше первоначально автоматизировать тест-кейсы, чтобы потом просто запустить все тесты и посмотреть на результат их выполнения.кну аутентификациишибкение выво о разработано яти таблиц. ложения мобильных объектов на платформе

Как показывает практика тестирования мобильных приложений, наиболее корректной работы приложения можно добиться при ручном тестировании на реальных мобильных устройствах.

Для отображения текстируемых элементов была составлена таблица 4.1 тест-кейсов.

Таблица 4.1 – Функциональные тест-кейсы

|  |  |  |
| --- | --- | --- |
| Описание теста | Ожидаемый результат | Статус |
| Установка приложения | Приложение установлено в систему | Успешно |
| Запуск приложения в первый раз | Приложение успешно установлено, отображён запрос на предоставление разрешения на показ поверх остальных окон | Успешно |
| Предоставление разрешения на показ поверх остальных окон | Скрытие запроса на предоставление разрешения на показ поверх остальных окон | Успешно |
| Запуск приложения после предоставления разрешения на показ поверх остальных окон | Отсутствие запроса на предоставление разрешения на показ поверх остальных окон | Успешно |
| Удаление приложения | Приложение удалено из системы | Успешно |
| Запуск приложения без предоставления разрешения на показ поверх остальных окон | Приложение заблокировано уведомлением о запросе на предоставление разрешения на показ поверх остальных окон | Успешно |
| Выбор системы перевода | Система перевода выбрана, о чём свидетельствует визуальное подтверждение | Успешно |
| Выбор языка перевода источника | Выбран язык перевода источника, о чём свидетельствует визуальное подтверждение | Успешно |
| Выбор языка перевода целевого направленя | Выбран язык перевода целевого направления, о чём свидетельствует визуальное подтверждение | Успешно |
| Поиск по частичному совпадению языка в окне выбора языка | Отображён список языков частично схожих по написанию с введённым в поиск языком | Успешно |
| Ввод текста в главное окно приложения и отправка для перевода | Текст переведен и добавлен в историю переводов | Успешно |
| Ввод текста с помощью голосового ввода и отправка его для перевода | Текст переведен и добавлен в историю переводов | Успешно |
| Продолжение таблицы 4.1 | | |
| Описание теста | Ожидаемый результат | Статус |
| Просмотр истории переводов | Все переводы доступны в истории переводов | Успешно |
| Изменение системы перевода для элемента в истории переводов | Элемент в истории переводов переведен в соответствии с выбранной системой перевода | Успешно |
| Изменение целевого языка для элемента в истории переводов | Элемент в истории переводов переведен в соответствии с выбранным целевым языком | Успешно |
| Выбор исходного текста в элементе в истории переводов | Исходный текст в выбранном элементе перевода выделился визуально | Успешно |
| Выбор переведенного текста в элементе в истории переводов | Переведённый текст в выбранном элементе перевода выделился визуально | Успешно |
| Редактирование текста в элементе в истории переводов | Текст выбранного элемента истории введён в поле ввода для перевода | Успешно |
| Функция “поделиться” текстом в элементе в истории переводов | Текст был передан в чужое приложение с помощью системного функционала Аndroid | Успешно |
| Добавление элементов истории переводов в избранные переводы | Элементы из истории переводов добавлены в избранные переводы | Успешно |
| Включение службы перевода буфера обмена | Приложение начало реагировать на изменения буфера обмена, в меню уведомлений системы Аndroid добавилось уведомление о запущенном переводчике | Успешно |
| Выключение службы переводов буфера обмена | Приложение перестало реагировать на изменения в буфере обмена, в меню уведомлений системы Android удалилось уведомление о запущенном переводчике | Успешно |
| Поиск по истории переводов по точному совпадению | В истории переводов отображен список переводов, исходный или переведенный текст которых точно совпадает с заданным в поиске | Успешно |
| Поиск по истории переводов по неточному совпадению | В истории переводов отображен список переводов, исходный или переведенный текст которых неточно совпадает с заданным в поиске | Успешно |
| Продолжение таблицы 4.1 | | |
| Описание теста | Ожидаемый результат | Статус |
| Поиск по истории переводов несуществующей записи | История переводов не отображает ни одного элемента | Успешно |
| Открытие меню главного окна приложения | Меню главного окна приложения успешно открыто | Успешно |
| Выбор пункта меню «Настройки» | Произошел запуск окна «Настройки» | Успешно |
| Выбор пункта меню «Избранное» | Произошел запуск окна «Избранное» | Успешно |
| Выбор пункта меню «Оценить приложение» | Произошел запуск окна приложения Google Play с открытым приложением Screen Translator. | Успешно |
| Выбор пункта меню «Связь с разработчиками» | Произошел запуск окна почтового приложения | Успешно |
| Выбор пункта настроек очистки истории | Отображено уведомление с возможностью подтверждения очищения истории | Успешно |
| Очистка истории | Количество элементов в истории переводов стало равным нулю | Успешно |
| Удаление из избранных переводов одного перевода | Выбранный элемент списка перестал отображаться в списке «Избранные переводы» | Успешно |
| Удаление из избранных всех переводов | Количество элементов в избранных переводах стало равным нулю | Успешно |
| Функция «поделиться» в окне избранных переводов | Текст был передан в чужое приложение с помощью системного функционала Аndroid | Успешно |
| Ввод текста и отправка его для перевода в окне быстрого перевода | Текст успешно переведён | Успешно |
| Изменение переводчика для последующего перевода текста в окне быстрого перевода | Визуальное отображение изменения переводчика | Успешно |
| Окончание таблицы 4.1 | | |
| Описание теста | Ожидаемый результат | Статус |
| Изменение переводчика для уже переведенного текста в окне быстрого перевода | Визуальное отображение изменения переводчика, текст заново переведён с помощью выбранного переводчика | Успешно |
| Нажатие кнопки «поделиться» в окне быстрого перевода | Текст был передан в чужое приложение с помощью системного функционала Аndroid | Успешно |
| Нажатие кнопки открытия основного окна в окне быстрого перевода | Произошел запуск главного окна приложения | Успешно |
| Нажатие кнопки быстрой смены языков в главном окне приложения | Язык-источник и целевой язык поменялись местами | Успешно |
| Нажатие кнопки быстрой смены языков в окне быстрого перевода | Язык-источник и целевой язык поменялись местами | Успешно |

Тестирование приложения проходило на следующих физических устройствах:

* Xiaomi Redmi 3S под управлением ОС Android 4.1;
* HTC Wildfire S под управлением ОС Android 4.2;
* Doogee X5 под управлением ОС Android 4.1.

А также на виртуальных устройствах под управлением ОС Android 4.1 и выше.

Во время тестирования ошибок обнаружено не было.

4.2 Руководство пользователя

При первом использовании приложения необходимо загрузить APK-файл в память устройства и установить его. После чего приложение станет доступно для запуска на устройстве. Чтобы запустить приложение нужно нажать на его значок, после чего будет запущено главное окно приложения (рисунок 4.1).



Рисунок 4.1 - Значек установленного приложения

В главном окне приложения можно увидеть кнопку главного меню приложения, кнопку запуска службы перевода буфера обмена, кнопку для поиска по истории переводов, кнопки выбора переводчика, кнопки выбора языка, кнопку для голосового ввода текста и кнопку отправки текста для перевода. Данные элементы управления позволяют осуществлять перевод текста на выбранные языки, просмотр истории переводов, добавление элемента истории в избранные переводы, включение и отключение службы фонового перевода, а также доступ к настройкам приложения.

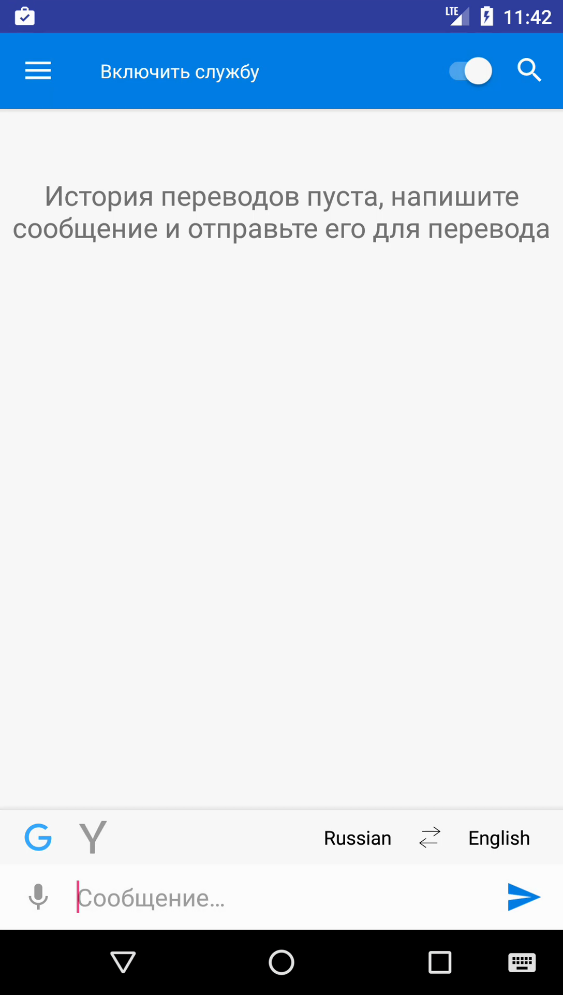


Рисунок 4.2 - Главное окно приложения

Для того чтобы перевести текст, необходимо выбрать язык исходного текста и язык, на который требуется перевести, а затем ввести текст в поле сообщения и нажать кнопку отправки для перевода. После чего текст будет переведён и отобразится в истории переводов.

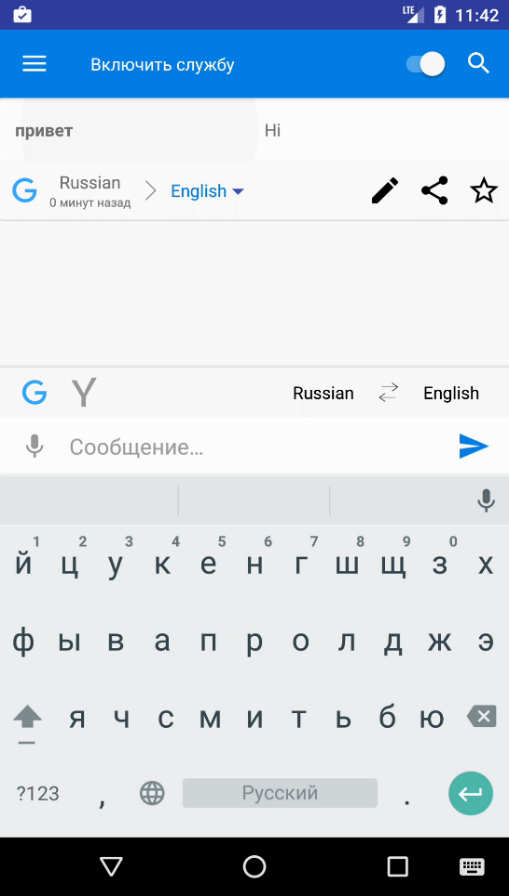


Рисунок 4.3 - Демонстрация переведенного текста

Для того чтобы перевести текст с другого языка, нужно нажать на кнопку смены языков, а затем ввести сообщение для перевода и отправить его.

Голосовой ввод осуществляется нажатием на кнопку со значком микрофона, после чего будет предложено произнести слова или предложения для перевода. Затем произнесённый текст будет введён в поле сообщения.

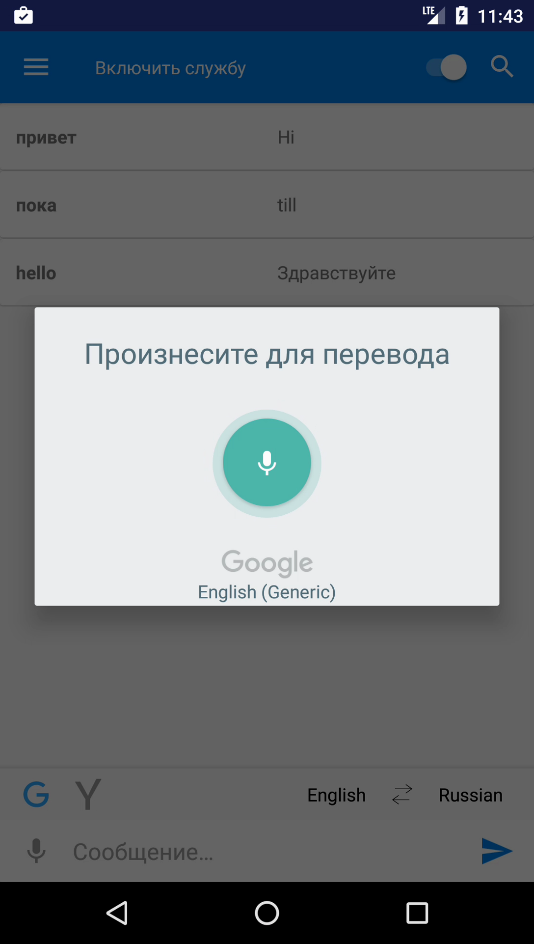


Рисунок 4.4 - Пример работы голосового ввода текста

Для того чтобы осуществить поиск по истории переводов, необходимо нажать кнопку со значком увеличительного стекла и ввести текст для поиска по истории переводов. После чего в истории переводов отображаться только те элементы которые полностью или частично совпадают с заданными в поле поиска текстом.

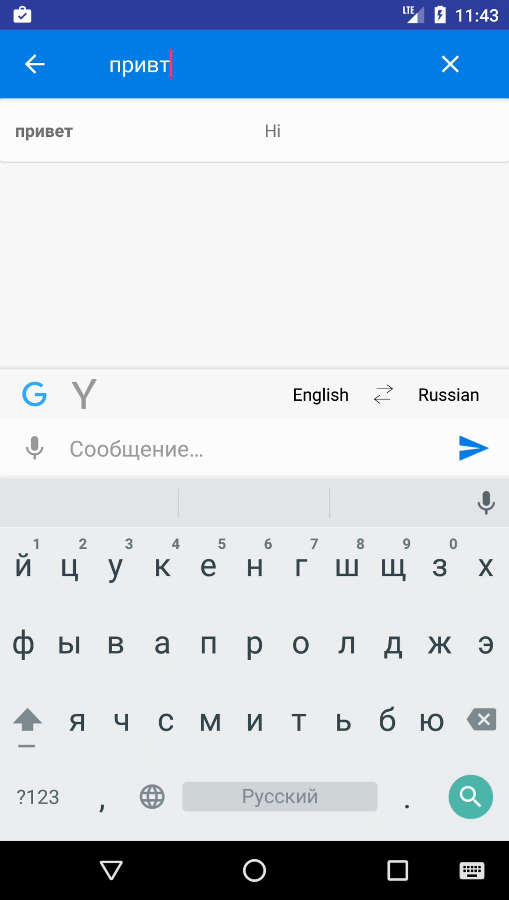


Рисунок 4.5 - Пример работы неточного поиска по истории переводов

Для того чтобы открыть главное меню приложения, необходимо нажать значок в верхнем левом углу приложение. В данном меню доступны элементы настроек приложения, избранных переводов, а также функция оценки приложения и связи с разработчиком.

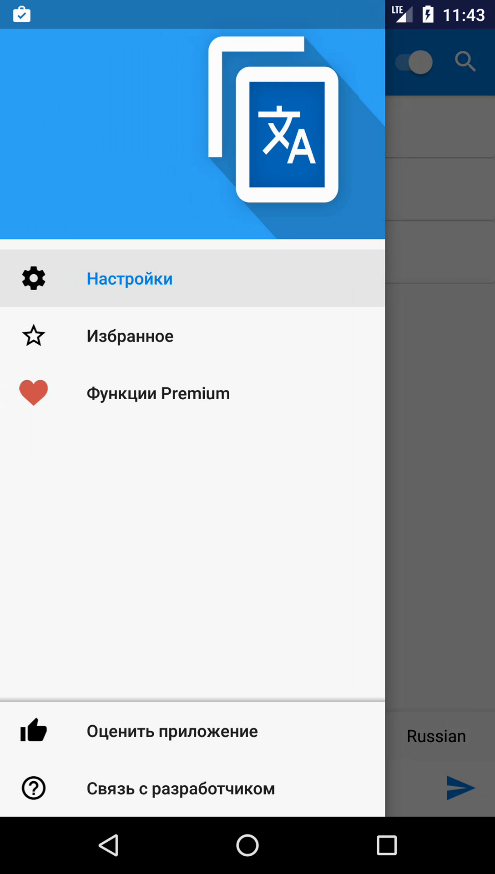


Рисунок 4.6 - Главное меню приложения

Для того чтобы добавить перевод в избранные переводы, необходимо нажать на элемент в истории переводов, чтобы открыть меню, а затем нажать на значок звёздочки,чтобы добавить элемент в список избранных переводов.

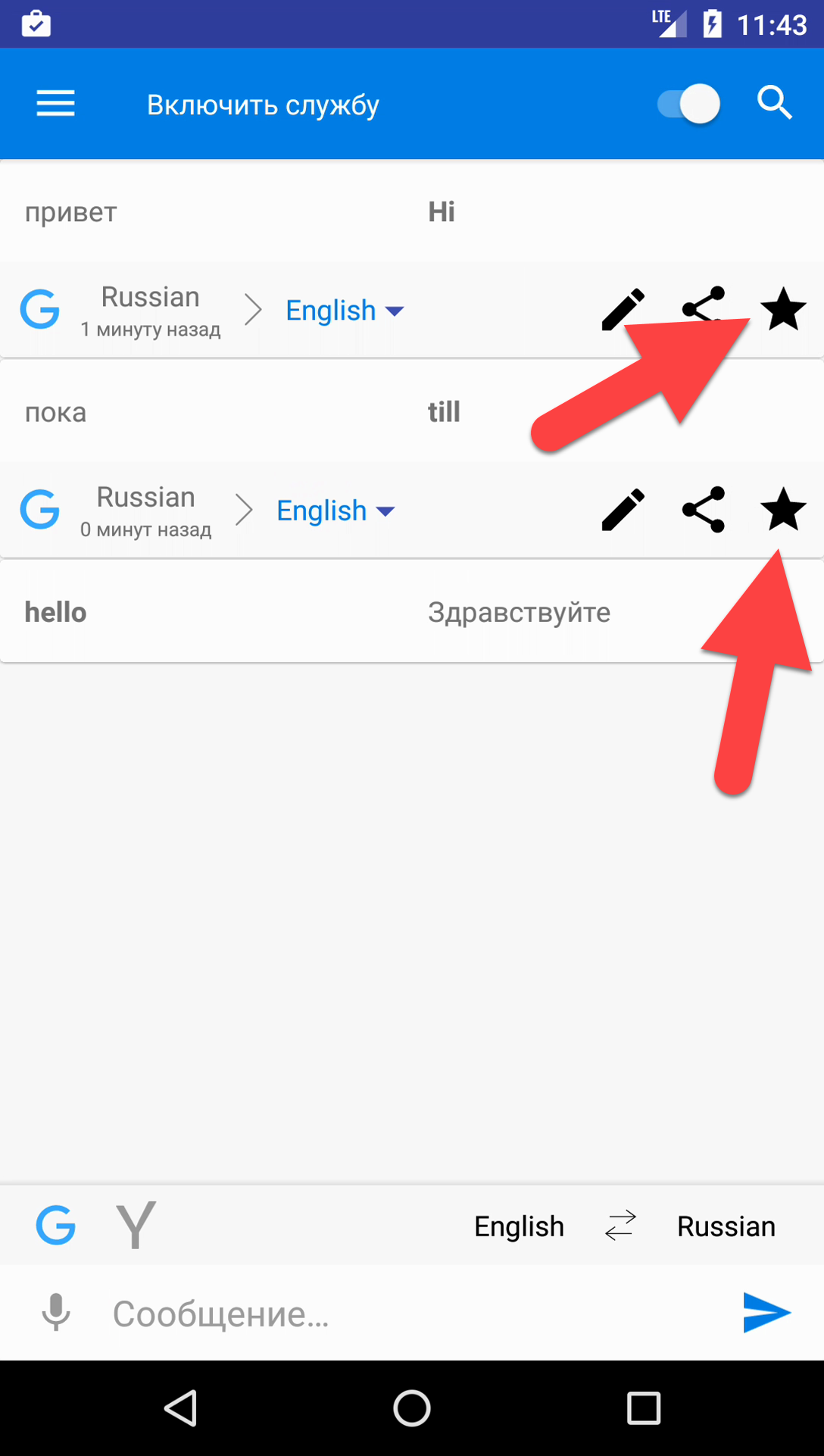


Рисунок 4.7 - Элементы управления для добавления в избранное

Чтобы просмотреть избранные переводы, необходимо открыть главное меню приложения, выбрать пункт меню “избранное”. После чего откроется окно со списком всех избранных переводов.

Избранные переводы имеют тот же фунционал, что и история переводов, за исключением поиска, смены языка и движка перевода существующего элемента.

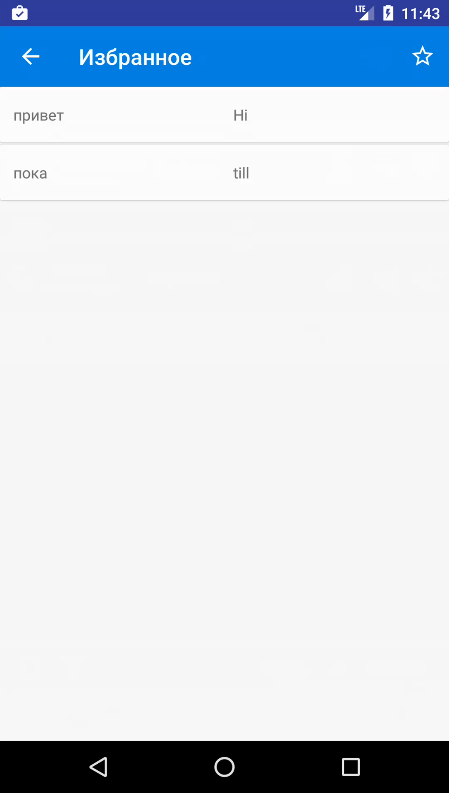


Рисунок 4.8 - Окно с избранными переводами

Для того чтобы очистить историю переводов, необходимо открыть главное меню приложения, выбрать пункт меню “Настройки”, а затем нажать на кнопку “очистить историю”, после чего вся история переводов будет очищена.

Чтобы перевести текст с помощью функции перевода буфера обмена, необходимо включить службу для перевода буфера обмена путем нажатия на кнопку “включить службу” в главном окне приложения, после этого в любом приложении необходимо выделить и скопировать текст, который необходимо перевести.

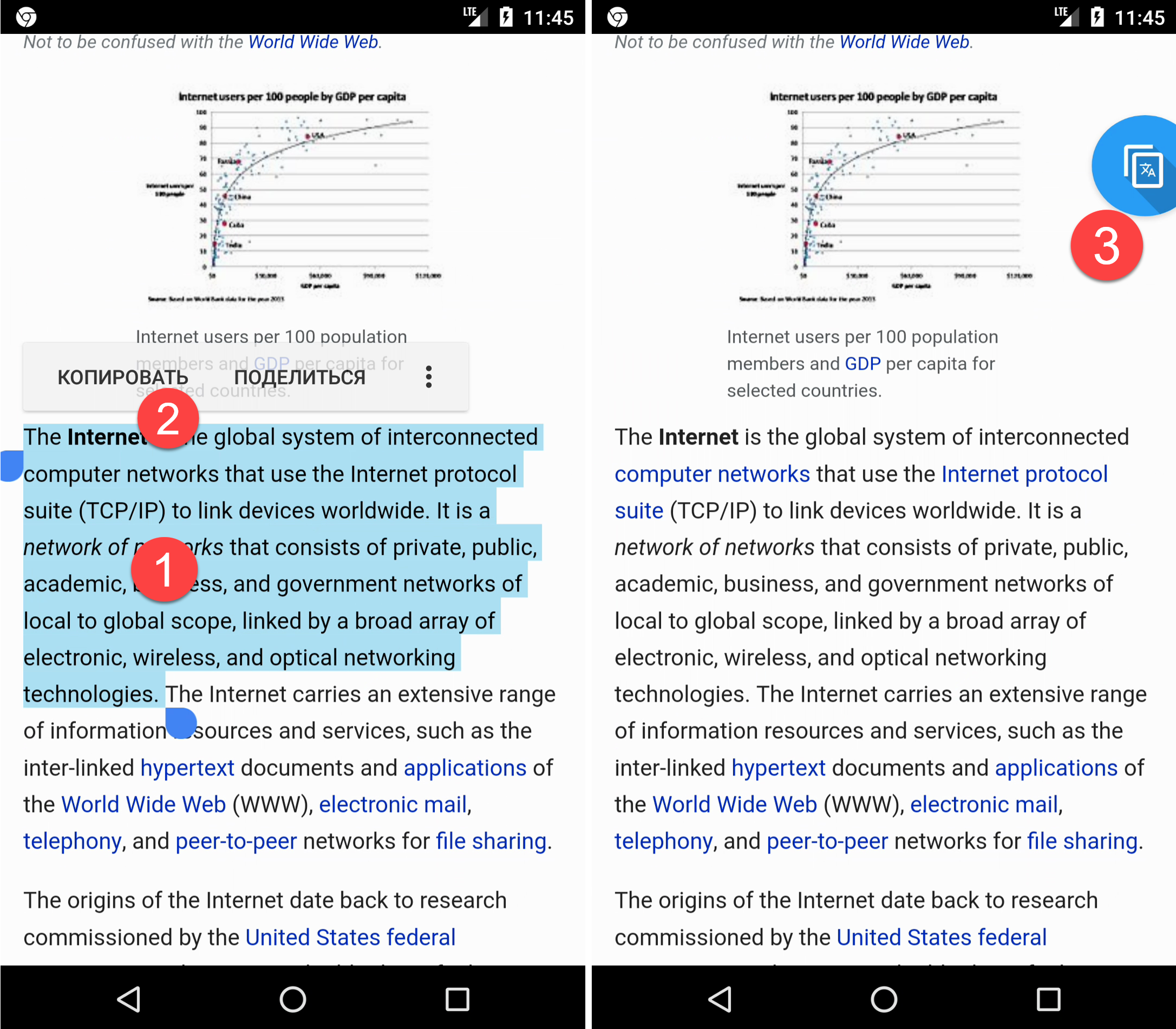


Рисунок 4.9 - Последовательность действий для перевода через буфер обмена

Скопировать текст можно только в приложениях, позволяющих копировать текст.

После того как произведено копирование текста, на экране отобразится изображение синего круга, нажав на который, откроется всплывающее окно с переведённым текстом.

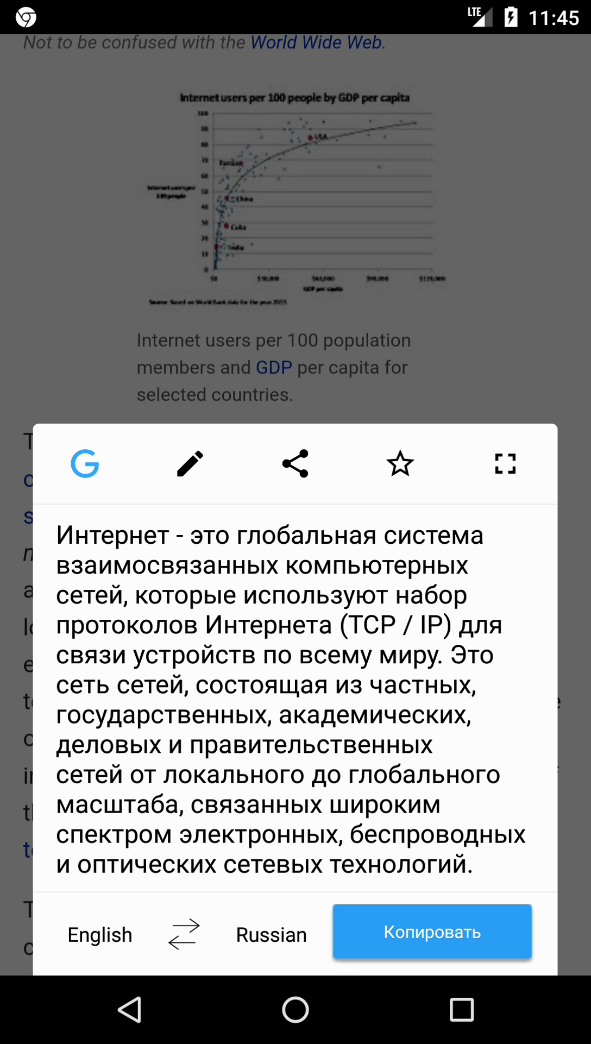


Рисунок 4.10 - Демонстрация переведенного текста через буфер обмена

Вывод по разделу 4

В данном разделе рассмотрены методы тестирования Android-приложений. Было проведено ручное тестирование мобильного приложения. В ходе данного тестирования не было найдено критическиз уязвимостей. Наиболее корректной работы приложения можно добиться при ручном тестировании на реальных мобильных устройствах.

Произведено ручное тестирование приложения на следующих физических устройствах:

* Xiaomi Redmi 3S, под управлением ОС Android 4.1;
* HTC Wildfire S, под управлением ОС Android 4.2;
* Doogee X5, под управлением ОС Android 4.1.

Во время тестирования ошибок обнаружено не было.

Приведено руководство пользователя.

5 Технико-экономическое обоснование проекта

Изм.

№ докум.

Подпись

Дата

Лист

1

ДП 05.00.ПЗ

Разраб.

Журавлёв В.Е.

Провер.

Герман Ю.О.

Консульт.

Соболевсий А.С.

Н. Контр.

Жиляк Н.А.

Утверд.

Смелов В.В.

Технико-экономическое обоснование проекта

Лит.

Листов

7

*БГТУ 84419008, 2017*

Главной целью экономического раздела является экономическое обоснование целесообразности разработки программного средства (ПС), представленного в дипломном проекте. В данном разделе пояснительной записки проводится расчет затрат на всех стадиях разработки и расчет экономии основных видов ресурсов в связи с использованием данного ПС.

Разработка проектов ПС требует разнообразных затрат и нередко значительных объемов ресурсов (трудовых, материальных, финансовых). В связи с этим, разработка и реализация каждого проекта должна быть обоснована, как технически, так и экономически.

5.1 Общая характеристика программного средства

Разработанное в данном дипломном проекте программное средство предназначено для перевода текста на иностранные языки на платформе Android. Программная часть разработана под платформу Android.

Android — операционная система для смартфонов, интернет-планшетов, электронных книг, цифровых проигрывателей, наручных часов, игровых приставок, нетбуков, смартбуков, очков Google, телевизоров и других устройств. В 2015 году появилась поддержка автомобильных развлекательных систем и бытовых роботов. Основана на ядре Linux и собственной реализации виртуальной машины Java от Google. Изначально разрабатывалась компанией Android, Inc., которую затем купила Google.

Приложение разработано с использованием JDK 1.8 и среды разработки Android Studio на объектно-ориентированном языке Java. Одним из достоинств данного языка является полная независимость байт-кода от операционной системы и оборудования.

Другой важной особенностью является гибкая система безопасности, в рамках которой исполнение программы полностью контролируется виртуальной машиной.

На рынке мобильных приложений имеется некоторое количество аналогов представленному в дипломном проекте клиентскому приложению.

Для установки клиентского приложения достаточно скопировать установочный файл в память устройства и запустить его. Для работы приложения требуется наличие подключения к сети интернет.

На основе анализа выполняемых функций и предлагаемых возможностей можно сделать вывод, что программное средство относится по степени сложности к третьей группе программ, а по степени новизны к группе Б. Также необходимо отметить, что ПС относится к программным средствам общего назначения.

5.2 Исходные данные

Исходные данные для расчета представлены в таблице 5.1. Данные взяты из нормативов, действующих на момент написания дипломного проекта.

Таблица 5.1– Исходные данные для расчетов

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование показателя | Единица измерения | Условные обозначения | Норматив |
| Коэффициент изменения скорости обработки информации | ед. | *К*ск | 0,65 |
| Численность разработчиков | чел. | *Ч*р | 1 |
| Тарифная ставка 1-го разряда в организации за месяц | руб. |  | 150 |
| Тарифный коэффициент | ед. | *К*т | 3,5 |
| Фонд рабочего времени | дн. | ФРВ | 251 |
| Коэффициент естественных потерь рабочего времени | ед. | Кп | 1,3 |
| Коэффициент премирования | ед. | Кпр | 1,1 |
| Норматив дополнительной заработной платы | % | Ндз | 15 |
| Ставка отчислений в Фонд социальной защиты населения | % | *Н*фсзн | 34 |
| Ставка отчислений по обязательному страхованию от несчастных случаев на производстве и профессиональных заболеваний | % | *Н*бгс | 1 |
| Цена одного машино-часа | руб. | Цм | 0,050642 |
| Норматив прочих прямых затрат | % | Нпз | 13 |
| Норматив накладных расходов | % | Ннр | 50 |
| Норматив расходов на сопровождение и адаптацию | % | *Н*рса | 10 |
| Уровень рентабельности | % | Урн | 30 |
| Ставка НДС | % | *Н*ндс | 20 |

5.3 Методика обоснования цены

В современных рыночных экономических условиях ПС выступает преимущественно в виде продукции организаций, представляющей собой функционально завершенные и имеющие товарный вид ПС, реализуемые покупателям по рыночным отпускным ценам. Все завершенные разработки ПС являются научно-технической продукцией.

Стоимостная оценка ПС у разработчиков предполагает определение затрат, что включает следующие статьи:

* заработная плата исполнителей – основная и дополнительная;
* отчисления в фонд социальной защиты населения и по обязательному страхованию от несчастных случаев на производстве и профессиональных заболеваний;
* расходы на материалы и комплектующие;
* расходы на специальное оборудование;
* расходы на оплату машинного времени;
* общепроизводственные и общехозяйственные расходы;
* расходы, связанные с установкой и адаптацией (реализационные или коммерческие расходы);
* накладные расходы.

На основании затрат рассчитывается себестоимость и отпускная цена ПС.

Необходимо с предельным вниманием отнестись к экономическим расчетам, поскольку они определят стоит ли разработка затраченных средств и будет ли окупаться проект в будущем.

5.3 Определение объема программного средства

Объем программного средства определяется путем подбора аналогов на основании классификации типов программных средств, каталога функций программных средств и аналогов программных средств в разрезе функций, которые постоянно обновляются и утверждаются в установленном порядке.

На основании информации о функциях разрабатываемого программного средства по каталогу функций определяется объем функций.

Общий объем программных средств рассчитывается по формуле:

, (5.1)



где – общий объем ПС, условных машино-команд;



– объем *i*-ой функции ПС, условных машино-команд;



*n* – общее число функций.

Содержание и объем функций на разрабатываемое программное средство приведены в таблице 5.2.

Таблица 5.2 – Содержание и объем функций на разрабатываемое программное средство

|  |  |  |
| --- | --- | --- |
| № функции | Содержание функции | Объем, условных машино-команд |
| 101 | Организация ввода информации | 150 |
| 102 | Контроль, предварительная обработка и ввод информации | 450 |
| 109 | Организация ввода/вывода информации в интерактивном режиме | 320 |
| 111 | Управление вводом/выводом | 2400 |
| 201 | Генерация структуры базы данных | 4300 |
| 204 | Обработка наборов и записей базы данных | 2670 |
| 207 | Манипулирование данными | 9550 |
| 210 | Загрузка базы данных | 2780 |
| 506 | Обработка ошибочных и сбойных ситуаций | 410 |
| 507 | Обеспечение интерфейса между компонентами | 970 |
| 707 | Графический вывод результатов | 480 |
|  | Итого | 24480 |

В связи с достаточно быстрым изменением вычислительной техники рекомендуется определить скорректированный объем функций по формуле:

*V*o/ (5.2)



где *V*o/ – скорректированный объем ПС, условных машино-команд;

– общий объем ПС, условных машино-команд;



– коэффициент изменения скорости обработки информации.



= 0,65 – по данным, приведенным в таблице 5.1; = 24480 условных машино-команд – подсчитано по формуле (5.1).



*V*o/ = 24480 · 0,65 = 15912 условных машино-команд.

5.4 Расчет трудоемкости выполняемой работы

Определение трудоемкости необходимо для дальнейшего расчета суммы основной заработной платы.

При определении трудоемкости ПС учитываются объем ПС (в тысячах строк условного кода), объем документации (тыс. строк), новизна и сложность ПС, язык программирования, степень использования типовых (стандартных программ).

На основании общего объема программного средства определяется нормативная трудоемкость (Тн). Нормативная трудоемкость устанавливается с учетом сложности ПС. Выделяется три группы сложности, в которых учтены следующие составляющие ПС: языковой интерфейс, ввод-вывод, организация данных, режим работы, операционная и техническая среда. Кроме того, устанавливаются дополнительные коэффициенты сложности ПС.

С учетом дополнительного коэффициента сложности Ксл рассчитывается общая трудоемкость ПС:

** (5.3)



где *Т*о – общая трудоемкость ПС, человеко-дней;

*Т*н – нормативная трудоемкость ПС, человеко-дней;

– дополнительный коэффициент сложности ПС, ед;



*Т*н = 339 -дня – по данным, приведенным в таблице 3 приложения 2 методического пособия; *К*сл = 0,06 – по данным, приведенным в таблице 6 приложения 2 методического пособия.

*Т*о = 339 · (1 + 0,06) = 359,34 человеко-дней.

5.5 Расчет основной заработной платы

В соответствии с «Рекомендациями по применению «Единой тарифной сетки» рабочих и служащих народного хозяйства» и тарифными разрядами и коэффициентами должностей каждому исполнителю устанавливается разряд и тарифный коэффициент.

Месячная тарифная ставка каждого исполнителя определяется путем умножения действующей месячной тарифной ставки 1-го разряда на тарифный коэффициент, соответствующий установленному тарифному разряду, и рассчитывается по формуле:

, (5.4)



где *С*зм – тарифная ставка за месяц, руб.;

– тарифная ставка 1 разряда за месяц, руб.;



*К*т – тарифный коэффициент, ед.

= 150– по исходным данным; *К*т= 3,5 – по исходным данным.



*С*зм = 150 · 3,5 = 525 руб.

Основная заработная плата исполнителей на конкретное ПС рассчитывается по формуле:

, (5.5)



где *С*оз – основная заработная плата, руб.;

*С*зд – тарифная ставка за день, руб.;

*Т*о – общая трудоемкость ПС, человеко-дней;

*К*п – коэффициент естественных потерь рабочего времени, ед.;

*К*пр – коэффициент премирования, ед.

*Т*о = 359,34 человеко-дней – подсчитано по формуле (5.3); *К*п = 1,3 – по исходным данным, приведенным в таблице 5.1; *К*пр = 1,1 – по исходным данным, приведенным в таблице 5.1.

*С*оз = 525/ 21,5 · 359,34· 1,3 · 1,1 = 12547 руб.

5.6 Расчет дополнительной заработной платы

Дополнительная заработная плата на конкретное ПС включает выплаты, предусмотренные законодательством о труде, и определяется по нормативу в процентах к основной заработной плате:

**, (5.6)

где *С*дз – дополнительная заработная плата на конкретное ПС, руб.;

*С*оз – основная заработная плата, руб.;

*Н*дз – норматив дополнительной заработной платы, %.

*С*оз = 12547 руб. – подсчитано по формуле (5.5); *Н*дз = 15% – по исходным данным, приведенным в таблице 5.1.

*С*дз = 12547 · 15 / 100 = 1882 руб.

5.7 Расчет отчислений в Фонд социальной защиты населения

Отчисления в Фонд социальной защиты населения (ФСЗН) определяются в соответствии с действующими законодательными актами по нормативу в процентном отношении к фонду основной и дополнительной зарплаты исполнителей. Общая формула для расчетов имеет следующий вид:

 (5.7)



где *С*фсзн – сумма отчислений в Фонд социальной защиты населения, руб.;

*Н*фсзн – норматив отчислений в Фонд социальной защиты населения, %;

*С*оз – основная заработная плата, руб.;

*С*дз – дополнительная заработная плата на конкретное ПС, руб.

*С*оз = 12547 руб. – подсчитано по формуле (5.5); *С*дз = 1882 руб. – подсчитано по формуле (5.6); *Н*фсзн = 34% – по исходным данным, приведенным в таблице 5.1.

*С*фсзн = (12547 + 1882 ) · 34 / 100 = 4905руб.

5.8 Расчет отчислений по обязательному страхованию от несчастных случаев на производстве и профессиональных заболеваний

Расчет отчислений по обязательному страхованию от несчастных случаев на производстве и профессиональных заболеваний выполняется в соответствии с действующим законодательством:

 (5.8)



где *С*бгс – сумма отчислений по страхованию от несчастных случаев, руб.;

*Н*бгс – норматив отчислений по страхованию от несчастных случаев, %;

*С*оз – основная заработная плата, руб.;

*С*дз – дополнительная заработная плата на конкретное ПС, руб.

*С*оз = 12547 руб. – подсчитано по формуле (5.5); *С*дз = 1882 руб. – подсчитано по формуле (5.6); *Н*бгс = 1% – по исходным данным, приведенным в таблице 5.1.

Сбгс = (12547 + 1882 ) · 1/ 100 = 41,39 руб.

5.9 Расчет расходов на спецоборудование

Расходы включают затраты средств на приобретение вспомогательных специального назначения технических и программных средств, необходимых для разработки конкретного программного средства, включая расходы на их проектирование, изготовление, отладку, установку и эксплуатацию. Затраты по этой статье определяются в соответствии со сметой расходов, которая составляется перед разработкой программного средства. Данная статья включается в смету расходов на разработку ПС только в том случае, когда приобретаются специальное оборудование или специальные программы, предназначенные для разработки и создания только данного программного средства. Расходов на приобретение спецоборудования не предвидится.

5.10 Расчет расходов на материалы

Расходы на материалы определяются с учетом действующих нормативов. По данной статье отражаются расходы на магнитную носители, перфокарты, бумагу, красящие ленты и другие материалы, необходимые для разработки ПС. Нормы расхода материалов в суммарном выражении определяются в расчете на 100 строк исходного кода. Общая сумма затрат на используемые материалы рассчитывается по формуле:

 (5.9)

где *С*м – сумма расходов на материалы, руб.;

*Н*м – норма расхода материалов в расчете на 100 строк исходного кода программного средства, руб.;

*V*o/ – скорректированный объем ПС, условных машино-команд.

*V*o/ = 15912 условных машино-команд – подсчитано по формуле (5.2);  
*Н*м = 380 неденоминированных руб. – по исходным данным, приведенным в таблице 10 приложения 2 методического пособия. С 1.07.2016 прошла деноминация (по принципу 1:10000), поэтому *Н*м = 0,038 руб.

*С*м = 0,038 · 15912 / 100 = 6,046 руб.

5.11 Расчет расходов на оплату машинного времени

Расходы включают оплату машинного времени, необходимого для разработки и отладки ПС, которое определяется по нормативам на 100 строк исходного кода машинного времени в зависимости от характера решаемых задач и типа вычислительной машины. Сумма затрат на оплату машинного времени рассчитывается по формуле:

 (5.10)



где *С*мв – сумма расходов на оплату машинного времени, руб.;

*Цм* – цена одного машино-часа, руб.;

*Н*мв – норматив расхода машинного времени на отладку 100 строк программного кода, машино-часов;

*V*o/ – скорректированный объем ПС, условных машино-команд.

*Ц*м = 0,050642 руб. – по исходным данным, приведенным в таблице 5.1;  
*V*o/ = 15912 условных машино-команд – подсчитано по формуле (5.2).

Информационная система выполняет общесистемные задачи,  
поэтому *Н*мв = 12 машино-часов – по данным, приведенным в таблице 11 приложения 2 методического пособия.

*С*мв = 0,050642· (15912 / 100) · 12 = 96,698 руб.

5.12 Расчет прочих прямых затрат

Расходы на конкретное ПС включают затраты на приобретение и подготовку специальной научно-технической информации и специальной литературы. Определяются по нормативу в процентах к основной заработной плате:

 (5.11)

где *С*пз – сумма прочих затрат, руб.;

*Н*пз – норматив прочих затрат в целом по организации, %;

*С*оз – основная заработная плата, руб.

*С*оз = 12547 руб. – подсчитано по формуле (5.5); *Н*пз = 13% – по исходным данным, приведенным в таблице 5.1.

*С*пз = (12547 · 13) / 100 = 467,896 руб.

5.13 Расчет накладных расходов

Данные затраты, связанные с необходимостью содержания аппарата управления, а также с расходами на общехозяйственные нужды, относятся на конкретное ПС по нормативу в процентном отношении к основной заработной плате исполнителей и определяются по формуле:

 (5.12)



где *С*нр – сумма накладных расходов, руб.;

*Н*нр – норматив накладных расходов в целом по организации, %;

*С*пз – сумма прочих затрат, руб.;

*С*оз – основная заработная плата, руб.

*С*оз = 12547 руб. – подсчитано по формуле (5.5); *Н*нр = 50% – по исходным данным, приведенным в таблице 5.1.

*С*нр= 12547 · 50 / 100 = 6273,5 руб.

5.14 Расчет суммы расходов на разработку ПС ВТ

Общая сумма расходов на ПС рассчитывается по формуле:

 (5.13)

где *С*р – сумма расходов на разработку ПС ВТ, руб.;

*С*оз – основная заработная плата, руб.;

*С*дз – дополнительная заработная плата на конкретное ПС, руб.;

*С*фсзн – сумма отчислений в Фонд социальной защиты населения, руб.;

*С*бгс – сумма отчислений по обязательному страхованию от несчастных случаев на производстве и профессиональных заболеваний, руб.;

*С*м – сумма расходов на материалы, руб.;

*С*мв – сумма расходов на оплату машинного времени, руб.;

*С*пз – сумма прочих затрат, руб.;

*С*нр – сумма накладных расходов, руб.

*С*оз = 12547 руб. – подсчитано по формуле (5.5); *С*дз = 1882 руб. – подсчитано по формуле (5.6); *С*фсзн = 4905 руб. – подсчитано по формуле (5.7); *С*бгс = 41,39 руб. – подсчитано по формуле (5.8); *С*м = 6,046 руб. – подсчитано по формуле (5.9); *С*мв = 96,698 руб. – подсчитано по формуле (5.10); *С*пз = 467,896 руб. – подсчитано по формуле (5.11); *С*нр = 6273,5 руб. – подсчитано по формуле (5.12).

Общая сумма расходов на разработку данного программного средства составит:

*С*р= 12547 + 1882 + 4905 + 41,39 + 6,046 +

+ 96,698 + 467,896 + 6273,5= 26219,53 руб.

5.15 Расчет расходов на сопровождение и адаптацию

Кроме того, организация-разработчик осуществляет затраты на сопровождение и адаптацию ПС, которые определяются по нормативу:

 (5.14)

где *С*рса – сумма расходов на сопровождение и адаптацию ПС ВТ, руб.;

*С*р – сумма расходов на разработку ПС, руб.;

*Н*рса – норматив расходов на сопровождение и адаптацию, %.

*С*р = 26219,53 руб. – подсчитано по формуле (5.13); *Н*рса = 10% – по исходным данным, приведенным в таблице 5.1.

*С*рса = 26219,53 · 10 / 100 = 2621,953 руб.

5.16 Расчет полной себестоимости разработки ПС ВТ

Общая сумма расходов на разработку (с затратами на сопровождение и адаптацию) – полная себестоимость ПС определяется по формуле:

 (5.15)

где *С*п – полная себестоимость ПС ВТ, руб.;

*С*р – сумма расходов на разработку ПС, руб.;

*С*рса – сумма расходов на сопровождение и адаптацию ПС, руб.

*С*р = 26219,53 руб. – подсчитано по формуле (5.13); *С*рса = 2621,953 руб. – подсчитано по формуле (5.14).

*С*п = 26219,53 + 2621,953 = 28840,953 руб.

5.17 Определение отпускной цены на ПС ВТ

Отпускная цена определяется на основании цены разработчика, которая формируется на основе показателя рентабельности продукции. Рентабельность и прибыль создаваемого ПС определяются исходя из результатов анализа рыночных условий, переговоров с заказчиком (потребителем) и согласования с ним отпускной цены, включающей дополнительно налог на добавленную стоимость и отчисления в целевые бюджетные фонды из выручки от реализации продукции.

Прибыль рассчитывается по формуле:

 (5.16)



где *П*пс – прибыль от реализации ПС, руб.;

*У*рн – уровень рентабельности ПС, %;

*С*п – полная себестоимость программного средства, руб.

*С*п = 28840,953 руб. – подсчитано по формуле (5.15); *У*рн = 30% – по исходным данным, приведенным в таблице 5.1.

*П*пс = 28840,953 · 30 / 100 = 8652,2859 руб.

Прогнозируемая цена разработчика ПС без налогов:

, (5.17)



где *Ц*п – прогнозируемая цена разработчика ПС, руб.;

*С*п – полная себестоимость программного средства, руб.;

*П*пс – прибыль от реализации, руб.

*С*п = 28840,953 руб. – подсчитано по формуле (5.15); *П*пс = 8652,2859 руб. – подсчитано по формуле (5.16).

*Ц*п = 28840,953 + 8652,2859 = 37492,2859 руб.

Сумма налога на добавленную стоимость:

 (5.18)



где *НДС* – сумма налога на добавленную стоимость, руб.;

*Н*ндс – ставка *НДС*, %;

*Ц*п – планируемая цена разработчика ПС, руб.

*Ц*п = 37492,2859 руб. – подсчитано по формуле (5.17); *Н*ндс = 20% – по исходным данным, приведенным в таблице 5.1.

*НДС* = 37492,2859 · 20 / 100 = 7498,45718 руб.



Планируемая отпускная цена:

**, (5.19) где *Ц*о – планируемая отпускная цена, руб.

*Ц*п = 37492,2859 руб. – подсчитано по формуле (5.17); *НДС* = 7900 руб. – подсчитано по формуле (5.18).

*Ц*о = 37492,2859 + 7900 = 47400,0177 руб.

5.18 Результаты и выводы по расчетам

В таблице 5.3 приведены данные о затратах, которые понесет разработчик при создании и продаже программного средства.

Таблица 5.3 – Расчет затрат на создание ПС и отпускной цены

|  |  |
| --- | --- |
| Наименование показателя | Сумма, руб. |
| Основная заработная плата | 12547 |
| Дополнительная заработная плата | 1882 |
| Отчисления в Фонд социальной защиты населения | 4905 |
| Отчисления по обязательному страхованию от несчастных и профессиональных заболеваний | 41,39 |
| Расход на материалы | 6,046 |
| Расход на оплату машинного времени | 96,698 |
| Прочие прямые затраты | 467,896 |
| Накладные расходы | 6273,5 |
| Расходы на разработку ПС | 26219,53 |
| Расходы на сопровождение и адаптацию ПС | 2621,953 |
| Полная себестоимость ПС | 28840,953 |
| Прибыль от реализации ПС | 8652,2859 |
| Планируемая цена разработчиков ПС без налогов | 37492,2859 |
| Налог на добавленную стоимость | 7900 |
| Отпускная цена с НДС | 47400,0177 |

С целью повышения производительности труда за счет снижения трудоемкости изделия существует ряд организационно-технических мероприятий: применение современных и энергосберегающих технологий, замена морально и физически устаревшего оборудования на более технологичное и высокопроизводительное, применение спецприспособлений. Предложенные мероприятия позволяют повысить конкурентоспособность данного вида продукции за счет снижения себестоимости и отпускной цены изделия.

Таким образом, из таблицы 5.3 видно, что большую часть отпускной цены разрабатываемого программного средства составляют основная заработная плата, накладные расходы и расходы на его разработку.

В результате расчетов себестоимость ПС составила 28840,953 руб.. прибыль разработчика от реализации составит 8652,2859 руб., прогнозируемая отпускная цена 47400,0177 руб. Все расчеты были проведены исходя из того, что рентабельность продукции составляет 30%. Расходов на приобретение спецоборудования не предвидится.

Выводы

В результате проведенных в экономической части дипломного проекта расчетов была определена производственная программа в натуральном и стоимостном выражении. Для определения производственной программы в стоимостном выражении произведен расчет калькулирования себестоимости изделия, фондов заработной платы по категориям работающих и общий плановый фонд заработной платы промышленно-производственного персонала.

В данном разделе было проведено технико-экономическое обоснование дипломного приложения, произведен расчет предполагаемых расходов, доходов и чистой прибили от реализации программного средства.

Результаты проведенных расчетов показывают, что разработанное программное средство является рентабельным и экономически обоснованным, а отпускная цена, являясь приемлемой для потребителя, обеспечит конкурентоспособность продукции на рынке товаров и услуг.

6 Мероприятия по охране труда и безопасности жизнедеятельности

Изм.

№ докум.

Подпись

Дата

Лист

1

ДП 06.00.ПЗ

Разраб.

Журавлёв В.Е.

Провер.

Герман Ю.О.

Консульт.

Герман Ю.О.

Н. Контр.

Жиляк Н.А.

Утверд.

Смелов В.В.

Мероприятия по охране труда и безопасности жизнедеятельности

Лит.

Листов

15

*БГТУ 84419008, 2017*

6.1 Особенности труда пользователя ПЭВМ

Частая и продолжительная работа за компьютером, не обеспеченная определенными организационно-техническими защитными мерами отрицательно сказывается на здоровье и самочувствии пользователей.

Пользователи ПЭВМ сталкиваются с воздействием таких физически опасных и вредных производственных факторов, как повышенный уровень шума, повышенная температура внешней среды, отсутствие или недостаточная освещенность рабочей зоны, электрический ток, статическое электричество и другие.

Работа на компьютере связана с воздействием таких психофизических факторов, как умственное перенапряжение, перенапряжение зрительных и слуховых анализаторов, монотонность труда, эмоциональные перегрузки.

Воздействие неблагоприятных факторов приводит к снижению работоспособности, вызванное развивающимся утомлением. Появление и развитие утомления связано с изменениями, возникающими во время работы в центральной нервной системе, с тормозными процессами в коре головного мозга.

К наиболее распространенным нарушениям обеспечения безопасных условий труда работающих на компьютерах относятся:

– недостаточные площадь и объем производственного помещения;

– несоблюдение требований, предъявляемых к температуре и влажности рабочих помещений;

– низкий уровень освещенности в помещениях и на рабочих поверхностях аппаратуры;

– повышенный уровень низкочастотных магнитных полей от мониторов;

– произвольная расстановка техники и нарушения требований ор-ганизации рабочих мест;

– несоблюдение требований к режимам труда и отдыха;

– чрезмерная производственная нагрузка работников;

– отсутствие навыков по снижению влияния психоэмоционального напряжения.

При работе на персональном компьютере пользователи могут подвергаться воздействию различных опасных и вредных производственных факторов, основными из которых являются:

– физические: повышенные уровни: электромагнитного, рентгеновского, ультрафиолетового и инфракрасного излучения; статического электричества; запыленности воздуха рабочей зоны; повышенное или пониженное содержаниеаэроионов в воздухе рабочей зоны;

повышенный или пониженный уровень освещенности рабочей зоны и другие;

– химические: содержание в воздухе рабочей зоны оксида углерода, озона, аммиака, фенола, формальдегида и полихлорированных фенилов;

– психофизиологические: напряжение зрения, памяти, внимания; длительное статическое напряжение: большой объем информации, обрабатываемой в единицу времени; монотонность труда; нерацио-нальная организация рабочего места; эмоциональные перегрузки.

Основными видами работ на ПЭВМ с использованием видеодисплейных терминалов (ВДТ) являются: считывание информации с экрана с предварительным запросом; ввод информации; творческая работа в режиме диалога с ПЭВМ. Наибольшая нагрузка на орган зрения имеет место при вводе информации в ПЭВМ. Наибольшее общее утомление вызывает работа в режиме диалога. Наибольшее напряжение вызывает выполнение работы при дефиците времени для принятия решения и особенно если это сопряжено с высокой ответственностью за принятые решения (например, при управлении непрерывными технологическими процессами).

Выполнение многих операций при работе на ПЭВМ требует длительного статического напряжения мышц спины, шеи, рук, ног, что приводит к быстрому развитию утомления. Указанные особенности работы часто усугубляются нерациональной высотой рабочей поверхности стола и сидения, отсутствием опорной спинки и под локотников, неудобными углами сгибания в плечевом и локтевом суставах при выполнении рабочих движений, неправильным углом наклона экрана, отсутствием пространства и подставки для ног.

Неблагоприятное влияние на условия труда работающих с ВДТ оказывают нерациональное естественное и искусственное освещение помещений и рабочих мест, яркие и темные пятна на рабочих поверхностях, засветка экрана посторонним светом, наличие ярких и блестящих предметов.

Важное значение для предупреждения утомления работников имеют также правильный выбор режима работы видеодисплейного терминала, применение защитных фильтров (с обязательным их заземлением), определение оптимальных и допустимых диапазонов визуальных эргономических параметров видеотерминала, использование светозащитных средств.

Использование фильтров-экранов позволяет снизить зрительное утомление и защитить пользователей от электростатического воздействия и частично от воздействия электрической составляющей электромагнитного поля.

Часто при организации рабочих мест для работающих на ПЭВМ не учитывается, что ВДТ генерирует рентгеновское, радиочастотное, видимое ультрафиолетовое излучение, а также имеют место электромагнитные излучения промышленной частоты. Указанные излучения могут оказывать неблагоприятное воздействие и на соседние рабочие места при их нерациональном размещении.

Наличие электростатического поля приводит к уменьшению содержания отрицательных ионов в воздухе помещения и загрязнению экрана в результате притягивания к нему отрицательных ионов и мелких частиц пыли.

Длительная работа компьютера приводит к снижению концентрации кислорода, повышению концентрации озона. Озон является сильным окислителем и концентрация его выше предельно допустимых величин может привести к неблагоприятным обменным реакциям организма, изменяя активность ряда ферментов, способствует нарушению зрения.

Важным фактором, оказывающим воздействие на состояние здоровья работающих на ПЭВМ, является аэроионный состав воздуха. Его нарушение ухудшает состав крови, работу органа зрения, иммунной системы.

6.2 Обеспечение санитарно-гигиенических условий

6.2.1 Освещение

Правильно спроектированное и выполненное производственное освещение улучшает условия зрительной работы, снижает утомляемость, способствует повышению производительности труда, благотворно влияет на производственную среду, оказывая положительное психологическое воздействие на работающего, повышает безопасность труда и снижает травматизм.

Недостаточность освещения приводит к напряжению зрения, ослабляет внимание, приводит к наступлению преждевременной утомленности. Чрезмерно яркое освещение вызывает ослепление, раздражение и резь в глазах.

Неправильное направление света на рабочем месте может создавать резкие тени, блики, дезориентировать работающего. Все эти причины могут привести к несчастному случаю или профзаболеваниям, поэтому столь важен правильный расчет освещенности.

Согласно требованиям нормативных документов помещения с ВДТ и ПЭВМ должны иметь естественное и искусственное освещение.

Естественное освещение должно осуществляться через световые проемы, ориентированные преимущественно на север и северо-восток и обеспечивать коэффициент естественной освещенности (КЕО) не ниже 1,5%.

Искусственное освещение должно осуществляться системой общего равномерного освещения. В случаях преимущественной работы с документами в помещениях эксплуатации ВДТ и ПЭВМ допускается применение системы комбинированного освещения. Освещенность на поверхности стола в зоне размещения документов должна быть 300-500 лк. Местное освещение, при этом, не должно создавать бликов на поверхности экрана и увеличивать освещенность экрана более 300 лк.

В качестве источников искусственного освещения обычно используются люминесцентные лампы, которые попарно объединяются в светильники и располагаются над рабочими поверхностями равномерно.

Источники света, такие как светильники и окна, которые дают отражение от поверхности экрана, значительно ухудшают точность знаков и влекут за собой помехи физиологического характера, которые могут выразиться в значительном напряжении, особенно при продолжительной работе. Отражение, включая отражения от вторичных источников света, должно быть сведено к минимуму.

Окраска помещений и мебели должна способствовать созданию благоприятных условий для зрительного восприятия, хорошего настроения. В зависимости от ориентации окон рекомендуется следующая окраска стен и пола: окна ориентированы на юг: – стены зеленовато-голубого или светло-голубого цвета; пол – зеленый; окна ориентированы на север: – стены светло-оранжевого или оранжево-желтого цвета; пол – красновато-оранжевый; окна ориентированы на восток: – стены желто-зеленого цвета; пол зеленый или красновато-оранжевый; окна ориентированы на запад: – стены желто-зеленого или голубовато-зеленого цвета; пол зеленый или красновато-оранжевый.

Для внутренней отделки помещений должны использоваться диффузно-отражающие материалы с коэффициентом отражения от потолка – 0,7 – 0,8; для стен 0,5 – 0,6; для пола – 0,3 – 0,6.

6.2.2 Параметры микроклимата

Параметры микроклимата могут меняться в широких пределах, в то время как необходимым условием жизнедеятельности человека является поддержание постоянства температуры тела благодаря терморегуляции, т.е. способности организма регулировать отдачу тепла в окружающую среду. Принцип нормирования микроклимата – созданиеоптимальных условий для теплообмена тела человека с окружающей средой.

Вычислительная техника является источником существенных тепловыделений, что может привести к повышению температуры и снижению относительной влажности в помещении. В помещениях, где установлены компьютеры, должны соблюдаться определенные параметры микроклимата. Производственные помещения, в которых установлены компьютеры, должны оборудоваться системами отопления, кондиционирования воздуха или эффективной приточно-вытяжной вентиляцией. Микроклимат должен соответствовать следующим санитарным нормам:

– температура воздуха в теплый период года – не более 23-25 °С, в холодный – 22-24°С;

– относительная влажность воздуха – 40-60%;

– скорость движения воздуха – 0,1 м/с.

Для повышения влажности воздуха в помещениях следует применять увлажнители воздуха, ежедневно заправлять их дистиллированной или кипяченой водой.

Площадь на одно рабочее место с ВДТ и ПЭВМ должна составлять не менее 6,0 м, а объем – не менее 20,0 м (в учебных заведениях не менее 18 м).

В помещениях ежедневно должна проводиться влажная уборка.

6.2.3 Шум и вибрация

Шум ухудшает условия труда, оказывая вредное действие на организм человека. Работающие в условиях длительного шумового воздействия испытывают раздражительность, головные боли, головокружение, снижение памяти, повышенную утомляемость, понижение аппетита, боли в ушах и так далее. Такие нарушения в работе ряда органов и систем организма человека могут вызвать негативные изменения в эмоциональном состоянии человека вплоть до стрессовых. Под воздействием шума снижается концентрация внимания, нарушаются физиологические функции, появляется усталость в связи с повышенными энергетическими затратами и нервно-психическим напряжением, ухудшается речевая коммутация. Все это снижает работоспособность человека и его производительность, качество и безопасность труда. Длительное воздействие интенсивного шума (выше 80 дБА) на слух человека приводит к его частичной или полной потере .Уровни звука приведены в таблице 6.1.

Таблица 6.1 – Уровни звука, эквивалентные уровни звука и уровни звукового давления в октавных полосах частот.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Категория нормы шума | Уровни звукового давления, дБ в октавных полосах со сред­негеометрическими частотами, Гц | | | | | | | | | Уровни звука, эквивалентные уровни звука, ДБА |
| 31,5 | 63 | 125 | 250 | 500 | 1000 | 2000 | 4000 | 8000 |
| I | 86 | 71 | 61 | 54 | 49 | 45 | 42 | 40 | 38 | 50 |
| II | 93 | 79 | 70 | 63 | 58 | 55 | 52 | 50 | 49 | 60 |
| III | 96 | 83 | 74 | 68 | 63 | 60 | 57 | 55 | 54 | 65 |
| IV | 103 | 91 | 83 | 77 | 73 | 70 | 68 | 66 | 64 | 75 |

При выполнении работ с ВДТ и ПЭВМ в производственных помещениях уровень вибрации не должен превышать допустимых значений согласно СанПиН 9-131 РБ 2000.

Для снижения уровня шума стены и потолок помещений, где установлены компьютеры, могут быть облицованы звукопоглощающими материалами. Дополнительным звукопоглощением служат однотонные занавеси из плотной ткани гармонирующие с окраской стен и подвешенные в складку на расстоянии 15-20 см от ограждения. Ширина занавеси должна быть в два раза больше ширины окна.

Уровень вибрации в помещениях вычислительных центров может быть снижен путем установки оборудования на специальные виброизоляторы. Шумящее оборудование (АЦПУ, принтеры и д.р.), уровни шума которого превышают допустимые, должно находиться вне помещения с ВДТ, ЭВМ и ПЭВМ.

6.2.4 Электромагнитное и ионизирующее излучения

Общеизвестно, что компьютерная техника является источником излучений и электромагнитных полей, потенциально опасных для здоровья человека, особенно при неправильном ее использовании. Особо необходимо отметить, что в спектре электромагнитных полей, создаваемых компьютерами, присутствуют низкочастотные электромагнитные колебания от единиц герц до нескольких десятков герц, частоты которых близки к частотам биоритмов человеческого организма. В этом принципиальное отличие ПЭВМ по их потенциальной экологической опасности в сравнении с обычными бытовыми электроприборами и другими излучающими техническими средствами, которые по роду своего использования могут находиться (как и ПЭВМ) в близком контакте с человеком.

При работе ВДТ уровни напряженности, плотности магнитного потока электромагнитного поля, напряженности электростатического поля согласно СанПиН 9-131 РБ 2000 на расстоянии 50 см от экрана, правой, левой, верхней и тыльной поверхности видеомонитора не должны превышать допустимых значений, приведенных в таблице 6.3

Таблица 6.2 – Допустимые значения параметров неионизирующих электромагнитных излучений.

|  |  |
| --- | --- |
| Наименование параметра | Допустимые значения |
| Напряженность электромагнитного поля. Электрическая составляющая, не более: | |
| диапазон частот 5 Гц-2 кГц; | 25,0 В/м |
| диапазон частот 2-400 кГц | 2,5 В/м |
| Плотность магнитного потока, не более: | |
| диапазон частот 5 Гц-2 кГц; | 25° нТл |
| диапазон частот 2-400 кГц | 25 нТл |
| Напряженность электростатического поля, не более | 15 кВ/м |

Допустимые уровни напряженности (плотности потока мощности) электромагнитных полей, излучаемых клавиатурой, системным блоком, манипулятором «мышь», беспроводными системами передачи информации на расстояния и иными вновь разработанными устройствами в зависимости от основной рабочей частоты изделия, не должны превышать значений, приведенных в таблице 6.4.

Таблица 6.3 – Допустимые уровни электромагнитных полей

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Диапазон частот | 0,3­-300 кГц | 0,3-3,0 МГц | 3,0 - 30,0 МГц | 30,0-300,0 МГц | 0,3-300 ГГц |
| Допустимый уровень | 25 В/м | 15 В/м | 10 В/м | 3 В/м | 10 мкВт/см |

Допустимые уровни напряженности электрического поля тока промышленной частоты 50 Гц, создаваемые монитором, системным блоком, клавиатурой, изделием в целом, не должны превышать 0,5 кВ/м.

Допустимые уровни напряженности электростатического поля, создаваемые монитором, клавиатурой, системным блоком, манипулятором «мышь», изделием в целом, не должны превышать 15,0 кВ/м.

Интенсивность ультрафиолетового излучения от экрана видеомонитора не должна превышать в диапазоне 0,28-0,315 мкм 0,1-10-3 Вт/м2; в диапазоне 0,315-0,4 мкм – 0,1 Вт/м2. Излучение в диапазоне 0,2-0,28 мкм не допускается.

Конструкция ВДТ, ЭВМ и ПЭВМ должна обеспечивать безопасный для пользователя уровень мощности экспозиционной дозы рентгеновского излучения в любой точке пространства на расстоянии 0,05 м от экрана и частей корпуса ВДТ, ЭВМ или ПВЭМ при любых положениях регулировочных устройств. Уровень мощности экспозиционной дозы рентгеновского излучения не должен превышать на расстоянии 0,5 м от экрана и частей корпуса ВДТ не должен превышать 7,74-10-12 А/кг, что соответствует мощности эквивалентной дозы, равной 100 мкР/ч (0,03 мкР/с).

Компьютеры с жидкокристаллическим экраном не имеют источников мощного электромагнитного излучения и не наводят статического электричества. Однако при использовании блока питания возникает некоторое превышение уровня на частоте 50­ Гц, поэтому рекомендуется работать больше с использованием аккумулятора.­­­­­­­­­­­­­­­­­­­­

Во всех случаях для снижения уровня облучения монитор рекомендуется располагать на расстоянии не ближе 50 см от пользователя.

Для снижения воздействия этих видов излучения рекомендуется применять мониторы с пониженным уровнем излучения (MPR-II, TCO-92, TCO-99), устанавливать защитные экраны, а также соблюдать регламентированные режимы труда и отдыха.

6.2.5 Эргономические требования к рабочему месту

Проектирование рабочих мест оборудованных ПЭВМ и ВДТ относится к числу важных проблем эргономического проектирования в области вычислительной техники.

Эргономическими аспектами проектирования видеотерминальных рабочих мест, в частности, являются: высота рабочей поверхности, размеры пространства для ног, требования к расположению документов на рабочем месте (наличие и размеры подставки для документов, возможность различного размещения документов, расстояние от глаз пользователя до экрана, документа, клавиатуры и т.д.), характеристики рабочего кресла, требования к поверхности рабочего стола, регулируемость элементов рабочего места.

Рабочее место и взаимное расположение всех его элементов должно соответствовать антропометрическим, физическим и психологическим требованиям. Большое значение имеет также характер работы. В частности, при организации рабочего места пользователя ПЭВМ должны быть соблюдены следующие основные условия: оптимальное размещение оборудования, входящего в состав рабочего места и достаточное рабочее пространство, позволяющее осуществлять все необходимые движения и перемещения.

Рабочие места должны располагаться таким образом, чтобы естественный свет падал сбоку, преимущественно слева.

Расстояние между рабочими столами с мониторами (в направлении тыла поверхности одного монитора и экрана другого) должно быть не менее 2м, а между боковыми поверхностями мониторов - не менее 1,2 м.

Рабочие места при выполнении творческой работы, требующей значительного умственного напряжения или высокой концентрации внимания, следует изолировать одно от другого перегородками высотой 1,5-2 м.

Главными элементами рабочего места пользователя ПЭВМ являются стол и кресло. Основным рабочим положением является положение сидя. Рабочая поза сидя вызывает минимальное утомление.

Зона досягаемости ограничивается дугами, описываемыми максимально вытянутыми руками при движении их в плечевом суставе. Зона легкой досягаемости – часть моторного поля, ограниченная дугами, описываемыми расслабленными руками при движении их в плечевом суставе. Оптимальная зона досягаемости моторного поля – предплечьями при движении в локтевых суставах с опорой.

Рациональная планировка рабочего места предусматривает четкий порядок и постоянство размещения предметов, средств труда и документации. То, что требуется для выполнения работ чаще, расположено в зоне легкой досягаемости рабочего пространства.

Рабочий стул (кресло) должен быть подъемно-поворотным и регулируемым по высоте и углам наклона сиденья и спинки, а также расстоянию спинки от переднего края сиденья. При этом регулировка каждого параметра должна быть независимой, легко осуществляемой и иметь надежную фиксацию. Поверхность сиденья, спинки и других элементов стула (кресла) должна быть полумягкой, с нескользящим, неэлектризующимся и воздухопроницаемым покрытием, обеспечивающим легкую очистку от загрязнений.

Тип рабочего стула (кресла) должен выбираться в зависимости от характера и продолжительности работы с ВДТ, ЭВМ и ПЭВМ с учетом роста пользователя.

Экран видеомонитора должен находиться от глаз пользователя на оптимальном расстоянии 600-700 мм, но не ближе 500 мм с учетом размеров алфавитно-цифровых знаков и символов.

Во всех случаях для снижения уровня облучения монитор рекомендуется располагать на расстоянии не ближе 50 см от пользователя.

Уровень глаз при вертикальном расположенном экране ВДТ должен приходиться на центр или 2/3 высоты экрана. Линия взора должна быть перпендикулярна центру экрана (рисунок 6.1а). При работе на клавиатуре необходимо соблюдать правильное положение рук оператора (рисунок 6.1б).

При неудобной рабочей позе могут появиться боли в мышцах, суставах и сухожилиях.

Клавиатуру следует располагать на поверхности стола на расстоянии не менее чем 300 мм от края, обращенного к пользователю или на специальной, регулируемой по высоте рабочей поверхности, отделенной от основной столешницы.



Рисунок . Правильная позиция оператора (а) и правильное положение рук оператора при работе на клавиатуре (б)

6.2.6 Режим труда

При работе с персональным компьютером очень важную роль играет соблюдение правильного режима труда и отдыха. В противном случае у персонала отмечаются значительное напряжение зрительного аппарата с появлением жалоб на неудовлетворенность работой, головные боли, раздражительность, нарушение сна, усталость и болезненные ощущения в глазах, в пояснице, в области шеи и руках.

Режимы труда и отдыха должны организовываться в зависимости от вида и категории трудовой деятельности.

Виды трудовой деятельности разделяются на три группы: группа А – работа по считыванию информации с экрана ВДТ, ЭВМ и ПЭВМ с предварительным запросом; группа Б – работа по вводу информации; группа В – творческая работа в режиме диалога с ЭВМ. При выполнении в течение рабочей смены работ, относящихся к разным видам трудовой деятельности, за основную работу с ВДТ, ЭВМ и ПЭВМ следует принимать такую, которая занимает не менее 50 % времени в течение рабочей смены или рабочего дня, что обязательно следует соблюдать.

Для обеспечения оптимальной работоспособности и сохранения здоровья профессиональных пользователей на протяжении рабочей смены должны устанавливаться регламентированные перерывы. Время регламентированных перерывов в течение рабочей смены следует устанавливать в зависимости от ее продолжительности, вида и категории трудовой деятельности (таблица 6.5).

Продолжительность непрерывной работы с ВДТ без регламентированного перерыва не должна превышать 2 ч.

При работе с ВДТ, ЭВМ и ПЭВМ в ночную смену (с 22 до 6 ч), независимо от категории и вида трудовой деятельности, продолжительность регламентированных перерывов должна увеличиваться на 60 мин.

Таблица 6.4 – Время регламентированных перерывов.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Категория  работы ВДТ, ЭВМ и ПЭВМ | Уровень нагрузки за рабочую смену при видах работ с ВДТ | | | Суммарное время регламентированных перерывов, мин | |
|  | группа А,  количество  знаков | группа Б,  количество  знаков | группа В, ч | При 8-  часовой  смене | при  12-часовой смене |
| I | до 20000 | до 15000 | до 2,0 | 30 | 70 |
| II | до 40000 | до 30000 | до 4,0 | 50 | 90 |
| III | до 60000 | до 40000 | до 6,0 | 70 | 120 |

При 8-часовой рабочей смене и работе на ВДТ, ЭВМ и ПЭВМ регламентированные перерывы следует устанавливать:

– для I категории работ – через 2 ч от начала рабочей смены и через 2 ч после обеденного перерыва продолжительностью 15 мин каждый;

– для II категории работ – через 2 ч от начала рабочей смены и через 1,5-2 ч после обеденного перерыва продолжительностью 15 мин каждый или продолжительностью 10 мин через каждый час работы;

– для III категории работ – через 1,5-2 ч от начала рабочей смены и через 1,5-2 ч после обеденного перерыва продолжительностью 20 мин каждым или продолжительностью 15 мин через каждый час работы.

При 12-часовой рабочей смене регламентированные перерывы должны устанавливаться в первые 8 часов работы аналогично перерывам при 8-часовой рабочей смене, а в течение последних 4 ч работы, независимо от категории и вида работ, каждый час продолжительностью 15 мин, что обязательно следует соблюдать.

Во время регламентированных перерывов с целью снижения нервно-эмоционального напряжения, утомления зрительного анализатора, устранения и влияния гиподинамии целесообразно выполнять комплексы упражнений.

Работающим на ВДТ, ЭВМ и ПЭВМ с высоким уровнем напряженности во время регламентированных перерывов и в конце рабочего дня показана психологическая разгрузка в специально оборудованных помещениях (комната психологической разгрузки).

Профессиональные пользователи ВДТ, ЭВМ и ПЭВМ должны проходить обязательные предварительные и периодические медицинские осмотры в установленном порядке и не иметь медицинских противопоказаний.

Женщины со времени установления беременности и в период кормления ребенка грудью к выполнению всех видов работ, связанных с использованием ВДТ, ЭВМ и ПЭВМ, не допускаются.

Эффективность перерывов повышается при сочетании с производственной гимнастикой или организации специального помещения для отдыха персонала с удобной мягкой мебелью, аквариумом, зеленой зоной.

6.3 Эргономика пользовательских интерфейсов

Масштабы распространения компьютеров, все возрастающая интенсификация человеческого труда требуют повышения внимания к проектированию интерфейсов, с тем, чтобы по возможности способствовать устранению или уменьшению стресса, который испытывает человек при работе с компьютером.

Под пользовательским интерфейсом (ПИ) понимают совокупность элементов, позволяющих пользователю программы управлять ее работой и получать требуемые результаты. Это система правил и средств, регламентирующая и обеспечивающая взаимодействие программы с пользователем. При проектировании интерфейсов главная задача - облегчить процесс восприятия и переработки информации.

Одним из важных показателей качества программного обеспечения является удобство его использования. Удобство использования пользовательского интерфейса (usability) – показатель его качества, который определяет общую концепцию удобства при использовании программного обеспечения, логичность и простоту в расположении элементов управления. Термин «юзабилити» можно рассматривать как синоним слова «эргономичность» с той разницей, что последняя определяет минимальность конкретных физических усилий при пользовании вещью, а первая – конечную суммарную степень удобства, меру интеллектуального усилия необходимого для получения полезных качеств этой вещи и скорость достижения положительного результата при управлении ею.

Согласно международному стандарту качества интерфейсов ISODIS 9241-11 Эргономические требования к офисной работе с визуальными терминалами, юзабилити определяется как «степень, в которой продукт может быть использован определенными пользователями для достижения поставленных целей эффективно, экономично и с удовлетворением в заданном контексте использования».

Для того чтобы любой пользователь мог легко освоить пользование программой, ее интерфейс должен быть выполнен в соответствии с выработанными эргономическими принципами. Эти принципы разрабатываются на основе опыта проектирования интерфейсов пользователей с учетом их пожеланий, на основе наблюдений за пользователями при их работе с программными средствами. Данные принципы позволяют рассматривать пользователя программного продукта и сам программный продукт как единую систему, между элементами которой происходят взаимодействия. Человек рассматривается как компонент рабочей системы. Правильное проектирование рабочих систем приводит к снижению рисков при работе, а, следовательно, и стрессов, с одновременным увеличением производительности труда.

Для успешного проектирования рабочих систем необходимо точно определить назначение и требования к разрабатываемой системе, в данном случае к программному средству. Интерфейс должен быть интуитивно понятным, не содержать элементов, которые вводят пользователя в заблуждение или заставляют обратиться в специализированные источники информации для уточнения смысла, должен позволять пользователю быстро получать доступ к имеющейся информации и удобно работать с программой.

Важным моментом при проектировании систем является распределение функций между человеком и системой. Для этого необходимо определить ограничения по функциям человека и машины, выбор исполнителя каждой функции и оптимизировать работу человека и машины. Распределение функций признается удовлетворительным, если рабочая нагрузка человека допустима, а работа осмысленна и мотивирована.

Хороший дизайн пользовательского интерфейса подразумевает, что программа соответствует ожиданиям пользователей о том, как она должна себя вести. Основное достоинство хорошего интерфейса пользователя заключается в том, что пользователь всегда чувствует, что он управляет программным обеспечением, а не программное обеспечение управляет им.

Для создания у пользователя такого ощущения «внутренней свободы» интерфейс должен обладать целым рядом свойств.

Естественность интерфейса. Естественный интерфейс – такой, который не вынуждает пользователя существенно изменять привычные для него способы решения задачи. Это, в частности, означает, что сообщения и результаты, выдаваемые приложением, не должны требовать дополнительных пояснений. Использование знакомых пользователю понятий и образов (метафор) обеспечивает интуитивно понятный интерфейс при выполнении его заданий. Метафоры являются своего рода «мостиком», связывающим образы реального мира с теми действиями и объектами, которыми приходится манипулировать пользователю при его работе на компьютере; они обеспечивают «узнавание», а не «вспоминание». Пользователи запоминают действие, связанное со знакомым объектом, более легко, чем они запомнили бы имя команды, связанной с этим действием.

Согласованность интерфейса. Согласованность позволяет пользователям переносить имеющиеся знания на новые задания, осваивать новые аспекты быстрее, и благодаря этому фокусировать внимание на решаемой задаче, а не тратить время на уяснение различий в использовании тех или иных элементов управления, команд и т. д. Обеспечивая преемственность полученных ранее знаний и навыков, согласованность делает интерфейс узнаваемым и предсказуемым. Согласованность важна для всех аспектов интерфейса, включая имена команд, визуальное представление информации и поведение интерактивных элементов.

Согласованность в пределах продукта. Одна и та же команда должна выполнять одни и те же функции, где бы она ни встретилась, причем одним и тем же образом. Надо использовать одну и ту же команду, чтобы выполнить функции, которые кажутся подобными пользователю.

Согласованность в пределах рабочей среды. Поддерживая согласованность с интерфейсом, предоставляемым операционной системой, приложение может «опираться» на те знания и навыки пользователя, которые он получил ранее при работе с другими приложениями.

Согласованность в использовании метафор. Если поведение некоторого программного объекта выходит за рамки того, что обычно подразумевается под соответствующей ему метафорой, у пользователя могут возникнуть трудности при работе с таким объектом.

Дружественность интерфейса (принцип прощения пользователя). На каждом этапе работы эффективный интерфейс должен разрешать только соответствующий набор действий и предупреждать пользователей о тех ситуациях, где они могут повредить системе или данным; также, у пользователя должна быть возможность отменить или исправить выполненные действия. Даже при наличии хорошо спроектированного интерфейса пользователи могут делать те или иные ошибки. Эти ошибки могут быть как «физического» типа (случайный выбор неправильной команды или данных) так и «логического» (принятие неправильного решения на выбор команды или данных). Эффективный интерфейс должен позволять предотвращать ситуации, которые, вероятно, закончатся ошибками. Он также должен уметь адаптироваться к потенциальным ошибкам пользователя и облегчать ему процесс устранения последствий таких ошибок.

Принцип «обратной связи». Всегда должна существовать обратная связь для действий пользователя. Каждое действие пользователя должно получать визуальное, а иногда и звуковое подтверждение того, что программное обеспечение восприняло введенную команду; при этом вид реакции, по возможности, должен учитывать природу выполненного действия. Желательно, чтобы время отклика было сопоставимо с реакцией человека. Типичный пользователь способен вытерпеть только несколько секунд ожидания ответной реакции от своего электронного «собеседника». Задержку часто можно «спрятать» от пользователя через многозадачность, позволяя продолжать выполнять другую работу, пока производятся фоновые вычисления. В любом случае пользователь должен иметь подтверждение того, что программа не «зависла», а продолжает работать.

Простота интерфейса. Простота интерфейса предполагает не упрощенчество, а обеспечение легкости в его изучении и в использовании. Кроме того, интерфейс должен предоставлять доступ ко всему перечню функциональных возможностей, предусмотренных данным приложением. Реализация доступа к широким функциональным возможностям и обеспечение простоты работы противоречат друг другу. Разработка эффективного интерфейса призвана сбалансировать эти цели. Один из возможных путей поддержания простоты – представление на экране информации, минимально необходимой для выполнения пользователем очередного шага задания. Многословные командные имена или сообщения, непродуманные или избыточные фразы затрудняют пользователю извлечение существенной информации. Другой путь к созданию простого, но эффективного интерфейса -размещение и представление элементов на экране с учетом их смыслового значения и логической взаимосвязи. Это позволяет использовать в процессе работы ассоциативное мышление пользователя.

Гибкость интерфейса. Гибкость интерфейса – это его способность учитывать уровень подготовки и производительность труда пользователя. Гибкость предполагает возможность изменения структуры диалога и/или входных данных.

Эстетическая привлекательность. Проектирование визуальных компонентов является важнейшей составной частью разработки программного интерфейса. Корректное визуальное представление используемых объектов обеспечивает передачу весьма важной дополнительной информации о поведении и взаимодействии различных объектов. В то же время следует помнить, что каждый визуальный элемент, который появляется на экране, потенциально требует внимания пользователя, которое, как известно, не безгранично. Следует формировать на экране среду, которая не только содействовала бы пониманию пользователем представленной информации, но и позволяла бы сосредоточиться на наиболее важных ее аспектах.

Качество интерфейса сложно оценить количественными характеристиками, однако более или менее объективную его оценку можно получить на основе приведенных ниже частных показателей.

Время, необходимое определенному пользователю для достижения заданного уровня знаний и навыков по работе с приложением (например, непрофессиональный пользователь должен освоить команды работы с файлами не более чем за 4 часа).

Сохранение полученных рабочих навыков по истечении некоторого времени (например, после недельного перерыва пользователь должен выполнить определенную последовательность операций за заданное время).

Скорость решения задачи с помощью данного приложения; при этом должно оцениваться не быстродействие системы и не скорость ввода данных с клавиатуры, а время, необходимое для достижения цели решаемой задачи. Исходя из этого, критерий оценки по данному показателю может быть сформулирован, например, так: пользователь должен обработать за час не менее 20 документов с ошибкой не более 1%.

Субъективная удовлетворенность пользователя при работе с системой (которая количественно может быть выражена в процентах или оценкой по n-бальной шкале).

В разработке интерфейса часто употребляются метафоры, чтобы помочь пользователю познакомиться с приложением, используя ассоциации с уже имеющимся знанием. Некоторые из наиболее известных – это метафора бухгалтерской книги VisiCalc, метафора рабочего стола, впервые примененная Xerox, и метафора чековой книжки из Quicken.

6.4 Мероприятия по безопасности жизнедеятельности

Для предотвращения производственных аварий и катастроф разрабатывается и осуществляется комплекс мероприятий по обеспечению безопасности жизнедеятельности персонала объекта хозяйствования и населения, который включает в себя два направления. Первое направление состоит в разработке технических и организационных мероприятий, уменьшающих вероятность реализации опасного поражающего потенциала современных технических систем. Второе направление содержит комплекс мероприятий по защите обслуживающего персонала, формирований гражданской обороны и населения при возникновении чрезвычайных ситуаций (ЧС) природного и техногенного характера.

Большое значение имеет своевременное оповещение людей о произошедшей аварии. Для этого создаются системы оповещения, организуются внутренняя система оповещения и связи. Наличие подобных систем имеет большое значение, так как они позволяют в минимально короткие сроки предупреждать работающих об аварии. Ответственность за организацию оповещения населения возлагается на штабы гражданской обороны объектов и районов.

В соответствии с Законом Республики Беларусь «О защите населения и территорий от чрезвычайных ситуаций природного и техногенного характера», чрезвычайная ситуация – это обстановка, сложившаяся в результате аварии, катастрофы, стихийного или иного бедствия, которые повлекли или могут повлечь за собой человеческие жертвы, вред здоровью людей или окружающей среде, значительные материальные потери и нарушение условий жизнедеятельности людей.

Предупреждение чрезвычайных ситуаций – комплекс мероприятий, проводимых заблаговременно и направленных на максимально возможное уменьшение риска возникновения чрезвычайных ситуаций, а также на сохранение здоровья людей, снижение размеров вреда окружающей среде и материальных потерь в случае их возникновения.

Зона чрезвычайной ситуации – территория, на которой возникла чрезвычайная ситуация.

Ликвидация чрезвычайных ситуаций – аварийно-спасательные и другие неотложные работы, проводимые при возникновении чрезвычайных ситуаций и направленные на спасение жизни и сохранение здоровья людей, снижение размеров вреда окружающей среде и материальных потерь, а также на локализацию зон чрезвычайных ситуаций.

По характерам происхождения ситуации, которые могут обусловить возникновение ЧС на территории Республики Беларусь, различают:

ЧС техногенного характера – транспортные аварии (катастрофы), пожары, неспровоцированные взрывы или их угроза, аварии с выбросом (угрозой выброса) опасных химических, радиоактивных, биологических веществ, внезапное разрушение сооружений и зданий, аварии на инженерных сетях и сооружениях жизнеобеспечения, гидродинамические аварии на плотинах, дамбах и других инженерных сооружениях;

ЧС природного характера – опасные геологические, метеорологические, гидрологические явления, деградация грунтов или недр, природные пожары, изменение состояния воздушного бассейна, инфекционная заболеваемость людей, сельскохозяйственных животных, массовое поражение сельскохозяйственных растений и лесных массивов болезнями или вредителями, изменение состояния водных ресурсов и биосферы.

Задачи государственной системы предупреждения и ликвидации чрезвычайных ситуаций определены Законом Республики Беларусь «О защите населения и территорий от ЧС природного и техногенного характера»:

– контроль состояния, природной среды и потенциально опасных объектов;

– прогнозирование ЧС и их последствий;

– проведение комплекса мероприятий по предупреждению ЧС;

– оповещение населения, органов власти и управления о ЧС;

– организация защиты населения и обеспечение экологической;

– подготовка сил и средств ликвидации ЧС и их последствий;

– планирование, организация и проведение спасательных и других неотложных работ по ликвидации ЧС и их последствий;

– оценка материального ущерба от ЧС и подготовка предложений о выделении материальных и финансовых средств на ликвидацию последствий и возмещение ущерба;

– проведение комплекса мероприятий по обеспечению устойчивости работы объектов экономики и систем жизнеобеспечения населения;

– обучение населения действиям по выживанию в ЧС.

Вывод

В разделе подробно описаны вредные и опасные производственные факторы при работе с персональным компьютером в ходе проектирования и разработки программного средства.

Представлены требования к помещениям и рабочим местам, где эксплуатируются ПЭВМ касающиеся освещения, микроклимата, уровня шума и вибрации, излучений и электромагнитных полей, пожарной безопасности. Также описаны общие принципы мероприятий по безопасности жизнедеятельности.

Заключение

Изм.

№ докум.

Подпись

Дата

Лист

1

ДП 00.00.ПЗ

Разраб.

Журавлёв В.Е.

Провер.

Герман Ю.О.

Консульт.

Герман Ю.О.

Н. Контр.

Жиляк Н.А.

Утверд.

Смелов В.В.

Заключение

Лит.

Листов

1

*БГТУ 84419008, 2017*

У

Во введении и аналитическом обзоре литературы был произведен соответсвующий экскурс в историю развития реологии, а также была сформулирована проблема просадки дорожной одежды из-за постоянного и многочисленного автомобильного трафика. Были описаны реологические модели для дорожных одежд. Был произведен патентный поиск аналогов разрабатываемого программного средства. В разделе, посвящённом выбору средств разработки, была обоснована причина выбора конкретной технологии и языка программирования, как наиболее оптимальных для решения поставленной задачи дипломного проекта. В разделе разработки программного средства были представлены расчетные схемы для многослойной и двухслойной дорожной одежды, а также были описаны этапы разработки программы и листинги наиболее важных компонентов системы. В руководстве пользователя показаны последовательности действий пользователей для коректной работы с соответсвующим программным средством. Было произведено тестирование разрабатываемого программного средства в целях выявления и исправления несогласованных и ошибочных ситуаций в работе программы. В разделе по охране труда были рассмотрены важные моменты по организации трудового процесса и контролю здоровья работников. В экономическом разделе была обоснована целесообразность разработки программного средства.

В результате было реализовано программное средство для расчета дорожных одежд с учетом реологических свойств материалов. Данная программа позволяет автоматизировать расчет численных значений просадки верхнего слоя дорожной одежды согласно алгоритму. Важным компонентом системы является приведение многослойной дорожной конструкции к двухслойной модели дорожной одежды. Была выполнена визуализация зависимостей просадки дорожной одежды от времени действия нагрузки и скорости движения автопоезда соответственно.

В настоящее время из-за увеличения количества транспорта на дорогах актуальной проблемой является определение численных характеристик просадки дорожной одежды. Это необходимо для того, чтобы при проектировании дорожных конструкций, можно было таким образом подобрать реологические параметры дорожной одежды, чтобы значение просадки стремилось к нулю, тем самым минимизировать появление неровностей на дороге. Поэтому данное разработанное программное средство имеет важное значение при конструировании дорожных одежд.

По теме диплома были выступления на научно-технических конференциях студентов и магистрантов, также имеется публикация [23].

Список используемой литературы

Изм.

№ докум.

Подпись

Дата

Лист

1

ДП 00.00.ПЗ

Разраб.

Журавлёв В.Е.

Провер.

Герман Ю.О.

Консульт.

Герман Ю.О.

Н. Контр.

Жиляк Н.А.

Утверд.

Смелов В.В.

Список литературы

Лит.

Листов

1

*БГТУ 84419008, 2017*

1. Википедия – свободная энциклопедия [Электронный ресурс] / Google Переводчик – Режим доступа: https://ru.wikipedia.org/wiki/Google\_Переводчик. – Дата доступа: 11.04.2017.
2. Википедия – свободная энциклопедия [Электронный ресурс] / Яндекс Переводчик – Режим доступа: https://ru.wikipedia.org/wiki/Яндекс.Переводчик. – Дата доступа: 12.04.2017.
3. Википедия – свободная энциклопедия [Электронный ресурс] / Microsoft Translator – Режим доступа: https://en.wikipedia.org/wiki/Microsoft\_Translator. – Дата доступа: 15.04.2017.
4. Википедия – свободная энциклопедия [Электронный ресурс] / Approximate string matching – Режим доступа: https://en.wikipedia.org/wiki/Approximate\_string\_matching. – Дата доступа: 14.04.2017.
5. Википедия – свободная энциклопедия [Электронный ресурс] / Расстояние Левенштейна – Режим доступа: https://ru.wikipedia.org/wiki/Расстояние\_Левенштейна. – Дата доступа: 23.04.2017.
6. Википедия – свободная энциклопедия [Электронный ресурс] / Расстояние Дамерау — Левенштейна – Режим доступа: https://ru.wikipedia.org/wiki/Расстояние\_Дамерау\_—\_Левенштейна. – Дата доступа: 11.04.2017.
7. Википедия – свободная энциклопедия [Электронный ресурс] / N-грамма – Режим доступа: https://ru.wikipedia.org/wiki/N-грамма. – Дата доступа: 23.04.2017.
8. Википедия – свободная энциклопедия [Электронный ресурс] / BK-tree – Режим доступа: https://en.wikipedia.org/wiki/BK-tree. – Дата доступа: 23.04.2017.
9. Википедия – свободная энциклопедия [Электронный ресурс] / UML – Режим доступа: https://ru.wikipedia.org/wiki/UML. – Дата доступа: 19.04.2017.
10. Википедия – свободная энциклопедия [Электронный ресурс] / Enterprise Architect (software) – Режим доступа: https://en.wikipedia.org/wiki/Enterprise\_Architect\_(software). – Дата доступа: 23.04.2017.
11. Википедия – свободная энциклопедия [Электронный ресурс] / ORM – Режим доступа: https://ru.wikipedia.org/wiki/ORM. – Дата доступа: 19.04.2017.
12. Каштелян, Т.В. Экономические обоснования дипломных проектов. Методические указания для студентов специальности 1-40 01 02-03 «Информационные системы и технологии» / Т.В. Каштелян. – Мн.: БГТУ, 2013. – 81с.

Приложение А. Программная реализация алгоритма Дамерау-Левенштейна.

**public class** DamerauLevenshteinAlgorithm {  
 **private final int deleteCost**, **insertCost**, **replaceCost**, **swapCost**;  
  
**public** DamerauLevenshteinAlgorithm(**int** deleteCost, **int** insertCost,  
 **int** replaceCost, **int** swapCost) {  
**if** (2 \* swapCost < insertCost + deleteCost) {  
 **throw new** IllegalArgumentException(**"Unsupported cost assignment"**);  
 }  
 **this**.**deleteCost** = deleteCost;  
 **this**.**insertCost** = insertCost;  
 **this**.**replaceCost** = replaceCost;  
 **this**.**swapCost** = swapCost;  
 }  
  
**public int** execute(String source, String target) {  
 **if** (source.length() == 0) {  
 **return** target.length() \* **insertCost**;  
 }  
 **if** (target.length() == 0) {  
 **return** source.length() \* **deleteCost**;  
 }  
 **int**[][] table = **new int**[source.length()][target.length()];  
 Map<Character, Integer> sourceIndexByCharacter = **new** HashMap<Character, Integer>();  
 **if** (source.charAt(0) != target.charAt(0)) {  
 table[0][0] = Math.*min*(**replaceCost**, **deleteCost** + **insertCost**);  
 }  
 sourceIndexByCharacter.put(source.charAt(0), 0);  
 **for** (**int** i = 1; i < source.length(); i++) {  
 **int** deleteDistance = table[i - 1][0] + **deleteCost**;  
 **int** insertDistance = (i + 1) \* **deleteCost** + **insertCost**;  
 **int** matchDistance = i \* **deleteCost** + (source.charAt(i) == target.charAt(0) ? 0 : **replaceCost**);  
 table[i][0] = Math.*min*(Math.*min*(deleteDistance, insertDistance),  
 matchDistance);  
 }  
 **for** (**int** j = 1; j < target.length(); j++) {  
 **int** deleteDistance = (j + 1) \* **insertCost** + **deleteCost**;  
 **int** insertDistance = table[0][j - 1] + **insertCost**;  
 **int** matchDistance = j \* **insertCost** + (source.charAt(0) == target.charAt(j) ? 0 : **replaceCost**);  
 table[0][j] = Math.*min*(Math.*min*(deleteDistance, insertDistance),  
 matchDistance);  
 }  
 **for** (**int** i = 1; i < source.length(); i++) {  
 **int** maxSourceLetterMatchIndex = source.charAt(i) == target.charAt(0) ? 0  
 : -1;  
 **for** (**int** j = 1; j < target.length(); j++) {  
 Integer candidateSwapIndex = sourceIndexByCharacter.get(target  
 .charAt(j));  
 **int** jSwap = maxSourceLetterMatchIndex;  
 **int** deleteDistance = table[i - 1][j] + **deleteCost**;  
 **int** insertDistance = table[i][j - 1] + **insertCost**;  
 **int** matchDistance = table[i - 1][j - 1];  
 **if** (source.charAt(i) != target.charAt(j)) {  
 matchDistance += **replaceCost**;  
 } **else** {  
 maxSourceLetterMatchIndex = j;  
 }  
 **int** swapDistance;  
 **if** (candidateSwapIndex != **null** && jSwap != -1) {  
 **int** iSwap = candidateSwapIndex;  
 **int** preSwapCost;  
 **if** (iSwap == 0 && jSwap == 0) {  
 preSwapCost = 0;  
 } **else** {  
 preSwapCost = table[Math.*max*(0, iSwap - 1)][Math.*max*(0, jSwap - 1)];  
 }  
 swapDistance = preSwapCost + (i - iSwap - 1) \* **deleteCost** + (j - jSwap - 1) \* **insertCost** + **swapCost**;  
 } **else** {  
 swapDistance = Integer.***MAX\_VALUE***;  
 }  
 table[i][j] = Math.*min*(Math.*min*(Math  
 .*min*(deleteDistance, insertDistance), matchDistance), swapDistance);  
 }  
 sourceIndexByCharacter.put(source.charAt(i), i);  
 }  
 **return** table[source.length() - 1][target.length() - 1];  
 }  
}

Приложение Б. Диаграмма классов

